

Managing Linux® Systems with Webmin™

Managing Linux® Systems

with Webmin™

System Administration and Module Development

Jamie Cameron



PRENTICE HALL
PROFESSIONAL TECHNICAL REFERENCE
UPPER SADDLE RIVER, NJ 07458
WWW.PHPTR.COM

Library of Congress Cataloging-in-Publication Data

Cameron, Jamie.

Managing Linux systems with Webmin / Jamie Cameron.

p. cm.

ISBN 0-13-140882-8

1. Linux. 2. Operating systems (Computers). I. Title.

QA76.76.O63 C3545 2003

005.4'32—dc22

2003016330

Editorial and production services: *TIPS Technical Publishing, Inc.*

Cover design director: *Jerry Votta*

Cover design: *Nina Scuderi*

Manufacturing buyer: *Maura Zaldivar*

Executive Editor: *Jill Harry*

Editorial assistant: *Brenda Mulligan*

Marketing manager: *Dan DePasquale*

© 2004 by Jamie Cameron



Published by Pearson Education, Inc.
Publishing as Prentice Hall Professional Technical Reference
Upper Saddle River, New Jersey 07458

This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Prentice Hall PTR offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact: U.S. Corporate and Government Sales, 1-800-382-3419, corpsales@pearsontechgroup.com. For sales outside of the U.S., please contact: International Sales, 1-317-581-3793, international@pearsontechgroup.com.

Company and product names mentioned herein are the trademarks or registered trademarks of their respective owners.

Printed in the United States of America

First Printing

ISBN 0-13-140882-8

Pearson Education LTD.

Pearson Education Australia PTY, Limited

Pearson Education Singapore, Pte. Ltd.

Pearson Education North Asia Ltd.

Pearson Education Canada, Ltd.

Pearson Educación de Mexico, S.A. de C.V.

Pearson Education—Japan

Pearson Education Malaysia, Pte. Ltd.

Contents at a Glance

I INTRODUCTION	
1 Introduction to Webmin	1
2 Installing Webmin	6
3 Securing Your Webmin Server	14
II SYSTEM MODULES	
4 Users and Groups	19
5 Disk and Network Filesystems	39
6 NFS File Sharing	53
7 Disk Quotas	60
8 Partitions, RAID, and LVM	68
9 Bootup and Shutdown	84
10 Scheduled Commands	93
11 Process Management	99
12 Software Packages.....	105
13 System Logs	113
14 Filesystem Backups	121
15 Internet Services	129
16 Network Configuration.....	144
17 Network Information Service	154
18 PPP Server Configuration.....	165
19 Firewall Configuration	173
20 Setting the Date and Time	191
21 Boot Loader Configuration.....	195
22 Printer Administration	205
23 Voicemail Server Configuration.....	215
24 Remote Shell Login	220
25 Running Custom Commands.....	224
26 Webmin's File Manager	232
27 Perl Modules.....	244
28 Status Monitoring with Webmin	250
III SERVER MODULES	
29 Apache Web Server Configuration ..	264
30 DNS Server Configuration	315
31 CVS Server Configuration	354
32 DHCP Server Configuration.....	361
33 Downloading Email with Fetchmail	378
34 Managing Majordomo Mailing Lists	388
35 The MySQL Database	405
36 The PostgreSQL Database.....	428
37 Configuring Sendmail	448
38 Configuring Qmail	476
39 Analyzing Log Files	491
40 The ProFTPD Server	500
41 The WU-FTPD Server.....	525
42 SSH Server Configuration.....	544
43 Windows File Sharing with Samba	554
44 Configuring the Squid Proxy Server.....	577
45 Filtering Email with Procmail	605
46 Creating SSL Tunnels	615
47 Usermin Configuration.....	620
IV CLUSTER MODULES	
48 Cluster Software Management	643
49 Cluster User Management	649
50 Cluster Webmin Configuration	660
V WEBMIN MODULES	
51 Webmin Configuration.....	669
52 Webmin Access Control.....	688
53 Webmin Servers	700
54 Logging in Webmin.....	707
VI DEVELOPER'S GUIDE	
55 Webmin Module Development	710
56 Advanced Module Development.....	721
57 Inside the Scheduled Cron Jobs Module.....	734
58 Creating Webmin Themes.....	741
59 Inside the MSC Theme	747
60 The Webmin API.....	751

Contents

I INTRODUCTION

1	Introduction to Webmin	1
	What is Webmin? 1	
	Who Should Use Webmin? 2	
	How and Why Was it Developed? 3	
	What is this Book About? 4	
	Who Should Read this Book? 4	
	Conventions Used in this Book 5	
	Acknowledgments 5	
2	Installing Webmin	6
	Downloading Webmin for Your System 6	
	Installing the RPM Package 7	
	Installing the tar.gz Package 8	
	Installing the Solaris Package 10	
	The Webmin User Interface 10	
	Uninstalling Webmin 13	
	Summary 13	
3	Securing Your Webmin Server	14
	Network Security 14	
	SSL Encryption 15	
	Requesting a Valid SSL Certificate 17	
	Summary 18	

II SYSTEM MODULES

4	Users and Groups	19
	Introduction to UNIX Users and Groups 19	
	The Users and Groups Module 20	
	Creating a New User 21	
	Editing an Existing User 23	
	Deleting a User 24	
	Creating a New Group 25	
	Editing an Existing Group 26	
	Deleting a Group 27	
	Viewing Recent and Current Logins 27	
	Reading Users' Email 28	
	Creating Users from Batch Files 28	
	Configuring the Users and Groups Module 30	

	Before and After Commands 34	
	Module Access Control 34	
	Other Operating Systems 37	
	Summary 38	
5	Disk and Network Filesystems	39
	Introduction to Filesystems 39	
	The Disk and Network Filesystems Module 40	
	Mounting an NFS Network Filesystem 40	
	Mounting an SMBFS Windows Networking Filesystem 43	
	Mounting a Local ext2 or ext3 Hard Disk Filesystem 44	
	Mounting a Local Windows Hard Disk Filesystem 45	
	Adding Virtual Memory 46	
	Automounter Filesystems 47	
	Editing or Removing an Existing Filesystem 48	
	Listing Users of a Filesystem 48	
	Module Access Control 49	
	Configuring the Disk and Network Filesystems Module 50	
	A Comparison of Filesystem Types 50	
	Other Operating Systems 51	
	Summary 52	
6	NFS File Sharing	53
	Introduction to File Sharing with NFS 53	
	The NFS Exports Module 54	
	Exporting a Directory 54	
	Editing or Deleting an NFS Export 55	
	NFS on Solaris 56	
	NFS on BSD, MacOS X and OpenServer 57	
	NFS on Irix 59	
	Summary 59	
7	Disk Quotas	60
	Introduction to Disk Quotas 60	
	The Disk Quotas Module 61	
	Enabling Quotas for a Filesystem 62	
	Disabling Quotas for a Filesystem 62	

Setting Quotas for a User or Group	63
Copying Quotas to Multiple Users	63
Setting Grace Times	64
Setting Default Quotas for New Users	65
Other Operating Systems	66
Configuring the Disk Quotas Module	66
Module Access Control	66
Summary	67
<hr/> 8 Partitions, RAID, and LVM	68
Introduction to Hard Disk Partitions	68
The Partitions on Local Disks Module	69
Adding and Formatting a New Partition	70
Creating a New Filesystem	70
Partition Labels	71
Deleting or Changing a Partition	72
Module Access Control	73
Other Operating Systems	74
Introduction to RAID	74
The Linux RAID Module	75
Introduction to LVM	77
The Logical Volume Management Module	78
Creating a New Volume Group	79
Adding and Removing a Physical Volume	80
Creating and Deleting a Logical Volume	80
Resizing a Logical Volume	81
Creating a Snapshot	82
Summary	83
<hr/> 9 Bootup and Shutdown	84
Introduction to the Linux Boot Process	84
The Bootup and Shutdown Module	85
Configuring an Action to Start at Bootup	85
Starting and Stopping Actions	86
Adding a New Action	87
Rebooting or Shutting Down Your System	89
Configuring the Bootup and Shutdown Module	89
Other Operating Systems	89
The SysV Init Configuration Module	91
Summary	92
<hr/> 10 Scheduled Commands	93
Introduction to Cron Jobs	93
The Scheduled Cron Jobs Module	93
Creating a New Cron Job	94
Editing a Cron Job	95
Controlling Users' Access to Cron	96
Module Access Control Options	96
Configuring the Scheduled Cron Jobs Module	96
Other Operating Systems	97
The Scheduled Commands Module	97
Creating a New Scheduled Command	98
Summary	98
<hr/> 11 Process Management	99
Introduction to Processes	99
The Running Processes Module	99
Viewing, Killing, or Reprioritizing a Process	101
Searching for Processes	102
Running a Process	103
Module Access Control Options	103
Other Operating Systems	104
Summary	104
<hr/> 12 Software Packages	105
Introduction to Packages	105
The Software Packages Module	107
Installing a New Package	107
Finding and Removing a Package	109
Updating on Debian Linux	110
Updating on Red Hat Linux	111
Other Operating Systems	111
Summary	112
<hr/> 13 System Logs	113
Introduction to Logging	113
The System Logs Module	115
Adding a New Log File	115
Editing or Deleting a Log File	117
Module Access Control	118
Other Operating Systems	119
Summary	120

14	Filesystem Backups	121
	Introduction to Backups with Dump	121
	The Filesystem Backup Module	121
	Adding a New Backup	122
	Making a Backup	124
	Editing or Deleting a Backup	125
	Restoring a Backup	125
	Configuring the Filesystem Backup Module	126
	Other Operating Systems	128
	Summary	128
15	Internet Services	129
	Introduction to Internet Services	129
	The Internet Services and Protocols Module	130
	Enabling an Internet Service	133
	Creating Your Own Internet Service	133
	Creating and Editing RPC Programs	135
	Configuring the Internet Services and Protocols Module	136
	Other Operating Systems	138
	The Extended Internet Services Module	139
	Enabling or Editing an Extended Internet Service	140
	Creating an Extended Internet Service	141
	Editing Default Options	142
	Summary	143
16	Network Configuration	144
	Introduction to Linux Networking	144
	Viewing and Editing Network Interfaces	146
	Adding a Network Interface	147
	Configuring Routing	149
	Changing the Hostname or DNS Client Settings	150
	Editing Host Addresses	151
	Module Access Control	152
	Other Operating Systems	153
	Summary	153
17	Network Information Service	154
	Introduction to NIS	154
	Becoming an NIS Client	155
18	PPP Server Configuration	165
	Introduction to PPP on Linux	165
	Configuring a PPP Server	166
	Managing PPP Accounts	169
	Restricting Access by Caller ID	171
	Module Access Control	172
	Summary	172
19	Firewall Configuration	173
	Introduction to Firewalling with IPtables	173
	The Linux Firewall Module	175
	Allowing and Denying Network Traffic	177
	Changing a Chain's Default Action	181
	Editing Firewall Rules	182
	Creating Your Own Chain	182
	Setting Up Network Address Translation	183
	Setting Up a Transparent Proxy	184
	Setting Up Port Forwarding	185
	Firewall Rule Conditions	186
	Configuring the Linux Firewall Module	189
	Summary	189
20	Setting the Date and Time	191
	The System Time Module	191
	Changing the System Time	192
	Change the Hardware Time	192
	Synchronizing Times with Another Server	193
	Module Access Control	193
	Other Operating Systems	193
	Summary	194

21	Boot Loader Configuration	195
	Introduction to Boot Loaders	195
	The Linux Bootup Configuration Module	196
	Booting a New Kernel with LILO	197
	Booting Another Operating System with LILO	198
	Editing Global LILO Options	199
	The GRUB Boot Loader Module	200
	Booting a New Linux Kernel or BSD with GRUB	201
	Booting Another Operating System with GRUB	202
	Editing Global GRUB Options	202
	Installing GRUB	203
	Configuring the GRUB Boot Loader Module	203
	Summary	203
22	Printer Administration	205
	Introduction to Printing on Linux	205
	The Printer Administration Module	206
	Adding a New Printer	206
	Editing an Existing Printer	209
	Managing Print Jobs	210
	Configuring the Printer Administration Module	211
	Module Access Control	212
	Other Operating Systems	213
	Summary	214
23	Voicemail Server Configuration	215
	The Voicemail Server Module	215
	Configuring Your System as an Answering Machine	216
	Listening to Recorded Messages	218
	Setting a Greeting Message	219
	Summary	219
24	Remote Shell Login	220
	The SSH/Telnet Login Module	220
	Configuring the SSH/Telnet Login Module	220
	The Command Shell Module	222
	The Shell In A Box Module	223
	Summary	223
25	Running Custom Commands	224
	The Custom Commands Module	224
	Creating a New Command	225
	Parameter Types	227
	Creating a New File Editor	229
	Module Access Control	230
	Configuring the Custom Commands Module	231
	Summary	231
26	Webmin's File Manager	232
	The File Manager Module	232
	Navigating Directories and Viewing Files	232
	Manipulating Files	234
	Creating and Editing Files	234
	Editing File Permissions	235
	Creating Links and Directories	236
	Finding Files	237
	Editing EXT File Attributes	237
	Editing XFS File Attributes	238
	Editing File ACLs	239
	Sharing Directories	240
	Module Access Control	242
	Summary	243
27	Perl Modules	244
	Introduction to Perl Modules	244
	Perl Modules in Webmin	245
	Installing a Perl Module	245
	Viewing and Removing a Perl Module	247
	Configuring the Perl Modules Module	248
	Summary	248
28	Status Monitoring with Webmin	250
	The System and Server Status Module	250
	Adding a New Monitor	252
	Monitor Types	253
	Setting Up Scheduled Monitoring	260
	Module Access Control	262
	Configuring the System and Server Status Module	262
	Summary	263

III SERVER MODULES

29	Apache Web Server Configuration	264	Module Access Control 346 Configuring the BIND DNS Server Module 347 The BIND 4 DNS Server Module 347 Summary 353
31	CVS Server Configuration	354	Introduction to CVS 354 The CVS Server Module 354 Setting Up the CVS Server 355 Using the CVS Server 356 Adding and Editing Users 356 Limiting User Access 358 Configuring the CVS Server 359 Browsing the Repository 359 Configuring the CVS Server Module 359 Summary 360
32	DHCP Server Configuration	361	Introduction to the Dynamic Host Configuration Protocol 361 The ISC DHCP Server 362 The DHCP Server Module 363 Adding and Editing Subnets 365 Viewing and Deleting Leases 369 Editing Global Client Options 370 Adding and Editing Fixed Hosts 370 Adding and Editing Shared Networks 372 Adding and Editing Groups 373 Module Access Control 374 Configuring the DHCP Server Module 375 Summary 377
33	Downloading Email with Fetchmail	378	Introduction to Fetchmail 378 The Fetchmail Mail Retrieval Module 379 Adding a New Mail Server to Check 381 Downloading Email 384 Running the Fetchmail Daemon 384 Editing Global Settings 385 Module Access Control 386 Configuring the Fetchmail Mail Retrieval Module 386 Summary 386
30	DNS Server Configuration	315	Introduction to the Domain Name System 315 The BIND DNS Server Module 318 Creating a New Master Zone 321 Adding and Editing Records 322 Record Types 325 Editing a Master Zone 330 Creating a New Slave Zone 332 Editing a Slave Zone 334 Creating and Editing a Forward Zone 336 Creating a Root Zone 337 Editing Zone Defaults 338 Configuring Forwarding and Transfers 340 Editing Access Control Lists 341 Setting Up Partial Reverse Delegation 342 Using BIND Views 344

34	Managing Majordomo Mailing Lists	388	36	The PostgreSQL Database	428
	Introduction to Mailing Lists and Majordomo	388		Introduction to PostgreSQL	428
	The Majordomo List Manager Module	389		The PostgreSQL Database Server	
	Using Other Mail Servers	391		Module	429
	Creating a Mailing List	391		Creating a New Database	431
	Managing List Members	392		Creating a New Table	431
	Editing List Information, Headers, and Footers	393		Adding and Editing Fields	433
	Editing Subscription Options	395		Deleting a Field	433
	Editing Forwarded Email Options	396		Field Types	434
	Editing List Access Control	397		Viewing and Editing Table Contents	436
	Moderating and Maintaining a Mailing List	398		Deleting Tables and Databases	436
	Deleting a Mailing List	399		Executing SQL Commands	437
	Creating a Digest List	399		Backing Up and Restoring a Database	437
	Editing Digest Options	400		Managing PostgreSQL Users	439
	Editing Global Majordomo Options	401		Managing PostgreSQL Groups	441
	Module Access Control	401		Restricting Client Access	441
	Configuring the Majordomo List Manager Module	402		Editing Object Privileges	442
	Summary	402		Module Access Control	443
35	The MySQL Database	405		Configuring the PostgreSQL Database Server Module	444
	Introduction to MySQL	405		Summary	447
	The MySQL Database Server Module	406	37	Configuring Sendmail	448
	Creating a New Database	407		Introduction to Internet Email	448
	Creating a New Table	408		The Sendmail Configuration Module	449
	Adding and Editing Fields	409		Editing Local Domains and Domain	
	Field Types	412		Masquerading	451
	Viewing and Editing Table Contents	412		Managing Email Aliases	452
	Deleting Tables and Databases	416		Configuring Relaying	455
	Executing SQL Commands	417		Managing Virtual Address Mappings	456
	Backing Up and Restoring a Database	417		Configuring Domain Routing	457
	Managing MySQL Users	419		Editing Global Sendmail Options	458
	Managing Database, Host, Table, and Field Permissions	421		Viewing the Mail Queue	460
	Module Access Control	423		Reading Users' Email	461
	Configuring the MySQL Database Server Module	424		Adding Sendmail Features with M4	463
	Summary	427		Creating Autoreply Aliases	465
38	Configuring Qmail	476		Creating Filter Aliases	466
	Introduction to Qmail	476		Sendmail Module Access Control	468
	The Qmail Configuration Module	477		Configuring the Sendmail Configuration	
	Editing Local Domains	478		Module	469
	Managing Email Aliases	479		Summary	475

Configuring Relaying 480	
Managing Virtual Mappings 481	
Configuring Domain Routing 483	
Editing Global Qmail Options 484	
Editing Mail User Assignments 484	
Viewing the Mail Queue 486	
Reading Users' Email 486	
Configuring the Qmail Configuration Module 488	
Summary 490	
<hr/>	
39 Analyzing Log Files	491
The Webalizer Logfile Analysis Module 491	
Editing Report Options 492	
Generating and Viewing a Report 496	
Reporting on Schedule 496	
Adding Another Log File 497	
Editing Global Options 498	
Module Access Control 498	
Summary 499	
<hr/>	
40 The ProFTPD Server	500
Introduction to FTP and ProFTPD 500	
The ProFTPD Server Module 501	
Running ProFTPD from inetd or xinetd 503	
Using the ProFTPD Server Module 504	
Creating Virtual Servers 505	
Setting Up Anonymous FTP 506	
Restricting Users to Their Home Directories 507	
Limiting Who Can Log In 508	
Setting Directory Listing Options 510	
Message and Readme Files 511	
Setting Per-Directory Options 512	
Restricting Access to FTP Commands 514	
Configuring Logging 517	
Limiting Concurrent Logins 519	
Restricting Clients by IP Address 520	
Limiting Uploads 521	
Manually Editing Directives 523	
Configuring the ProFTPD Server Module 523	
Summary 524	
<hr/>	
41 The WU-FTPD Server	525
Introduction to WU-FTPD 525	
The WU-FTPD Server Module 526	
Limiting Who Can Log In 528	
Setting Up Anonymous FTP 529	
Managing User Classes 531	
Denying Access to Files 532	
Setting Up Guest Users 534	
Editing Directory Aliases 535	
Message and Readme Files 536	
Configuring Logging 538	
Limiting Concurrent Logins 540	
Restricting Clients by IP Address 541	
Restricting Access to FTP Commands 541	
Configuring the WU-FTPD Server Module 542	
Summary 543	
<hr/>	
42 SSH Server Configuration	544
Introduction to SSH 544	
The SSH Server Module 545	
Restricting Access to the SSH Server 545	
Network Configuration 547	
Authentication Configuration 549	
Editing Client Host Options 551	
Setting Up SSH for New Users 552	
Configuring the SSH Server Module 553	
Summary 553	
<hr/>	
43 Windows File Sharing with Samba	554
Introduction to SMB and Samba 554	
The Samba Windows File Sharing Module 556	
Managing Samba Users 556	
Adding a New File Share 559	
Adding a New Printer Share 560	
Viewing and Disconnecting Clients 562	
Editing Share Security Options 563	
Editing File Permission Settings 564	
Editing File Naming Options 565	
Editing Other File Share Options 566	
Editing Printer Share Options 567	
Editing Share Defaults 568	
Configuring Networking 568	
Configuring Authentication 571	
Configuring Printers 572	

Accessing SWAT from Webmin 573	Configuring the SSL Tunnels Module 618
Module Access Control 573	Summary 619
Configuring the Samba Windows File Sharing Module 574	<hr/>
Summary 576	47 Usermin Configuration 620
44 Configuring the Squid Proxy Server 577	Introduction to Usermin 620
Introduction to Proxying and Squid 577	The Usermin Configuration Module 621
The Squid Proxy Server Module 578	Starting and Stopping Usermin 621
Changing the Proxy Ports and Addresses 580	Restricting Access to Usermin 622
Adding Cache Directories 581	Changing the Port and Address 623
Editing Caching and Proxy Options 583	Configuring the Usermin User Interface 623
Introduction to Access Control Lists 584	Installing Usermin Modules 624
Creating and Editing ACLs 586	Changing the Default Language 625
Creating and Editing Proxy Restrictions 592	Upgrading Usermin 625
Setting Up Proxy Authentication 593	Configuring Authentication 626
Configuring Logging 595	Editing Categories and Moving Modules 628
Connecting to Other Proxies 596	Changing and Installing Themes 629
Clearing the Cache 598	Turning on SSL 630
Setting Up a Transparent Proxy 599	Configuring Usermin Modules 631
Viewing Cache Manager Statistics 599	Restricting Access to Modules 632
Analyzing the Squid Logs 600	Limiting Who Can Log In 636
Module Access Control 601	About the Usermin Modules 638
Configuring the Squid Proxy Server Module 601	Configuring the Usermin Configuration Module 641
Summary 604	Summary 642
45 Filtering Email with Procmail 605	<hr/>
Introduction to Procmail 605	IV CLUSTER MODULES
The Procmail Mail Filter Module 606	48 Cluster Software Management 643
Setting Up Sendmail 606	Introduction to Webmin Clustering 643
Creating and Editing Actions 608	The Cluster Software Packages Module 644
Creating and Editing Variable Assignments 611	Registering a Server 645
Conditional Blocks and Include Files 612	Installing a Package 646
Filtering Spam with SpamAssassin 613	Searching for Packages 646
Configuring the Procmail Mail Filter Module 614	Deleting a Package 647
Summary 614	Exploring and Removing a Server 647
46 Creating SSL Tunnels 615	Refreshing the Package List 648
Introduction to SSL and STunnel 615	Configuring the Cluster Software Packages Module 648
The SSL Tunnels Module 616	Summary 648
Creating and Editing SSL Tunnels 617	<hr/>
49 Cluster User Management 649	
The Cluster Users and Groups Module 649	
Registering a Server 650	
Creating a New User 651	

Editing an Existing User	652	Changing Your Operating System	675
Deleting a User	653	Editing the Program Path and Environment Variables	676
Creating a New Group	654	Changing Webmin's Language	676
Editing an Existing Group	654	Editing Main Menu Settings	677
Deleting a Group	656	Upgrading Webmin	678
Refreshing User and Group Lists	656	Installing Updates to Webmin	679
Synchronizing Users and Groups	656	Configuring Authentication	681
Listing and Removing a Server	658	Editing Categories and Moving Modules	682
Configuring the Cluster Users and Groups Module	659	Changing and Installing Themes	683
Summary	659	Referrer Checking	684
50 Cluster Webmin Configuration	660	Allowing Unauthenticated Access to Modules	685
The Cluster Webmin Configuration Module	660	Turning on SSL	686
Registering a Server	661	Setting Up a Certificate Authority	686
Creating a New Webmin User	662	Summary	687
Editing or Deleting a Webmin User	662	52 Webmin Access Control	688
Creating a New Webmin Group	664	Introduction to Webmin Users, Groups, and Permissions	688
Editing or Deleting a Webmin Group	664	The Webmin Users Module	689
Editing the User or Group ACL for a Module	665	Creating a New Webmin User	689
Installing a Module or Theme	666	Editing a Webmin User	691
Viewing and Deleting a Module or Theme	667	Editing Module Access Control	692
Refreshing User and Module Lists	667	Creating and Editing Webmin Groups	694
Listing and Removing a Server	668	Requesting a Client SSL Key	695
Configuring the Cluster Webmin Configuration Module	668	Viewing and Disconnecting Login Sessions	697
Summary	668	Module Access Control	697
V WEBMIN MODULES		Configuring the Webmin Users Module	698
51 Webmin Configuration	669	Summary	699
The Webmin Configuration Module	669	53 Webmin Servers	700
Restricting Access to Webmin	669	The Webmin Servers Index Module	700
Changing the Port and Address	670	Adding a Webmin Server	701
Setting Up Logging	671	Editing or Deleting a Webmin Server	703
Using Proxy Servers	672	Using Server Tunnels	703
Configuring the Webmin User Interface	672	Broadcasting and Scanning for Servers	704
Installing and Deleting Webmin Modules	673	How RPC Works	704
Cloning a Webmin Module	674	Module Access Control	705

54 Logging in Webmin	707	57 Inside the Scheduled Cron Jobs Module	734
Introduction to Logging	707	Module Design and CGI Programs	734
The Webmin Actions Log Module	708	The cron-lib.pl Library Script	735
Displaying Logs	708	Module Configuration Settings	737
Summary	709	The lang Internationalization Directory	738
VI DEVELOPER'S GUIDE			
55 Webmin Module Development	710	The acl_security.pl Access Control Script	738
Introduction	710	The log_parser.pl Log Reporting Script	739
Required Files	711	The useradmin_update.pl User Synchronization Script	740
Module CGI Programs	712	Summary	740
Module Configuration	715	58 Creating Webmin Themes	741
Look and Feel	717	Introduction to Themes	741
Design Goals	718	Overriding Images and Programs	743
Online Help	718	Theme Functions	744
Module Packaging	719	Summary	746
Summary and Learning More	720	59 Inside the MSC Theme	747
56 Advanced Module Development	721	Theme Design and Graphics	747
Module Access Control	721	The index.cgi Program	748
User Update Notification	723	The theme_header Function	748
Internationalization	723	The theme_footer Function	749
File Locking	725	Summary	750
Action Logging	726	60 The Webmin API	751
Pre- and Post-Install Scripts	728	API Functions	751
Functions in Other Modules	728	Summary	765
Remote Procedure Calls	730	Index	767
Creating Usermin Modules	732		
Summary	733		

Introduction to Webmin

This chapter explains what Webmin is, why it was written, and what you can expect from this book.

1.1 What is Webmin?

Webmin is a program that simplifies the process of managing a Linux or UNIX system. Traditionally, you have needed to manually edit configuration files and run commands to create accounts, set up web servers, or manage email forwarding. Webmin now lets you perform these tasks through an easy-to-use web interface, and automatically updates all of the required configuration files for you. This makes the job of administering your system much easier.

Some of the things that you can do with Webmin include:

- Creating, editing, and deleting UNIX login accounts on your system
- Exporting files and directories to other systems with the NFS protocol
- Setting up disk quotas to control how much space users can take up with their files
- Installing, viewing, and removing software packages in RPM and other formats
- Changing your system's IP address, DNS settings, and routing configuration
- Setting up a firewall to protect your computer or give hosts on an internal LAN access to the Internet
- Creating and configuring virtual web sites for the Apache Web server
- Managing databases, tables, and fields in a MySQL or PostgreSQL database server
- Sharing files with Windows systems by configuring Samba

These are just a few of the available functions. Webmin lets you configure almost all of the common services and popular servers on UNIX systems using a simple web interface. It protects you

from the syntax errors and other mistakes that are often made when editing configuration files directly, and warns you before potentially dangerous actions.

Because Webmin is accessed through a web browser, you can log in to it from any system that is connected to yours through a network. There is absolutely no difference between running it locally and running it remotely, and it is much easier to use over the network than other graphical configuration programs.

Webmin has what is known as a *modular* design. This means that each of its functions is contained in a module that can generally be installed or removed independently from the rest of the program. Each module is responsible for managing some service or server, such as UNIX users, the Apache Web server, or software packages.

If you have been manually configuring your system up till now, any existing settings will be recognized by Webmin. It always reads the standard configuration files on your system and updates them directly, instead of using its own separate database. This means that you can freely mix Webmin, manual configuration, and other programs or scripts that work in the same way.

Even though this book is written for Linux users, Webmin can be used on many other flavors of UNIX as well, such as Solaris, FreeBSD, and HP/UX. One of its biggest strengths is its understanding of the differences between all these operating systems and the way it adjusts its user interface and behavior to fit your OS. This means that it can often hide the underlying differences between each UNIX variant and present a similar or identical interface no matter which one you are using.

Webmin on its own is not particularly useful though—it is only a configuration tool, so you must have programs installed for it to configure. For example, the Apache module requires that the actual Apache Web server be installed. Fortunately, all of the services and servers that Webmin manages are either included with most Linux distributions as standard, or can be freely downloaded and installed.

1.2 Who Should Use Webmin?

Webmin was written for use by people who have some Linux experience but are not familiar with the intricacies of system administration. Even though it makes the process of creating UNIX users or managing the Squid proxy server easy, you must first have some idea of what a UNIX account is and what Squid does. The average Webmin user is probably someone running it on their Linux system at home or on a company network.

The program assumes that you are familiar with basic TCP/IP networking concepts, such as IP addresses, DNS servers, and hostnames. It also assumes that the user understands the layout of the UNIX filesystem, what users and groups are, and where user files are located. If you use Webmin to manage a server like Apache or Sendmail, you should first have an idea of what they can do and what kind of configuration you want completed.

Webmin itself runs with full UNIX root privileges, which means that it can edit any file and run any command on your system. This means that it is quite possible to delete all of the files on your system or make it un-bootable if you make a mistake when using the program, especially if you are configuring something that you don't understand. Even though Webmin will usually warn you before performing some potentially dangerous action, there is still plenty of scope for causing damage.

Even though it can be used on a system with no connection to the Internet, Webmin does benefit if your Linux system is on a network. It can download new software packages, Perl modules, or even new versions of Webmin for you, if connected. A permanent high-speed connection is best, but even a dial-up connection is good enough for most purposes.

Because Webmin runs with `root` privileges, you must be able to log in to your system as `root` to install and start it. This means that it cannot be used on a system on which you have only a normal UNIX account, such as a virtual web server that is shared with other people. You might, however, be able to get your system administrator to install and configure it for you.

If you are already an experienced UNIX system administrator, Webmin may not feel like the tool for you because using it is generally slower than directly editing configuration files and running commands. However, even the experts can benefit from its automatic syntax checking and the actions that it can perform automatically.

It is also possible to give different people different levels of access to Webmin, so that an experienced administrator can use it to safely delegate responsibility to less-skilled subordinates. For example, you might want someone to be only able to manage the BIND DNS server and nothing else, while giving yourself full access to the system and all of Webmin's functions.

1.3 How and Why Was it Developed?

Webmin, the program, was designed and created by me, Jamie Cameron—the author of this book. I started it back in 1997 and released the first version (0.1) in October of that year. Since that time, its user interface, features, and appearance have changed dramatically, and almost all of the code has been re-written. The basic concept of a web-based administration tool, however, has been the same since that very first release.

I started writing it when I was the administrator for a system running a DNS server and was spending a lot of time updating the server's configuration files to add new host records requested by users. Giving them the `root` password was not an option—they did not have the experience to properly edit the zone files and re-start the server. The solution was a simple web interface that would display existing DNS records and allow them to be edited, created, and deleted. Users could then safely be given access to this interface to make the changes that they needed.

DNS management was just the start though. Once I saw the possibilities for simplifying the configuration of a UNIX system through a web interface, I started adding other features to the program and putting them into modules. Next came modules for UNIX users, Samba, mounting file-systems, NFS, and Cron jobs. I thought up the name Webmin, made it available for anyone to download, and announced it on a few mailing lists. The initial feedback was good, so I kept on writing.

Over the years, the program has gone through three different user interfaces, grown to 83 modules, added support for non-English languages, provided advanced access control, included lots more operating systems, and offered many other features. The Linux distribution companies Caldera and MSC.Linux have supported the project financially, and many users have made contributions of code patches, modules, translations, and suggestions. In addition to the standard modules, over 100 have been written by other people and are available to be added to Webmin on your system once you have installed the program.

1.4 What is this Book About?

This book explains how to install Webmin, how to use almost all of its modules, and how to write your own. The book focuses on the standard modules that come with the Webmin package, not those written by other people. Not all of the 83 standard modules are covered, however, as some are not very useful to the average administrator.

Although this book is written primarily for Linux users, the program behaves almost identically on other operating systems. Each chapter also lists any differences between Linux and other UNIX variants in their “Other Operating Systems” sections. This means that it is still very useful if you are running Webmin on FreeBSD, Solaris, MacOS X, or some other variety of UNIX.

Each chapter in the book covers the use of Webmin for managing a particular service or server, such as NFS exports, Sendmail, or the ProFTPD FTP server. Most chapters only discuss a single module, but some cover two or three that have similar or related purposes. Each chapter is pretty much self-contained, so there is no need to read through the entire book in sequence if you just want to find out how to configure one server. Chapters 2, 3, and possibly Chapter 52, however, should be read first as they explain how to install Webmin, how to secure it, and how to limit what other users can do with a module, respectively.

Each chapter is broken up into sections, and most sections explain how to perform a specific task. A section will generally contain an introduction to the task explaining why you might want to do it, followed by a list of steps to follow in the Webmin user interface to carry it out. At the beginning of each chapter are sections that introduce the server being configured and the concepts behind it, and list the underlying configuration files that get modified when you use the module covered in that chapter.

Chapters 55 to 60 cover the development of your own Webmin modules and themes, and therefore have a different style. The average user does not need to read them, but if you have an idea for a module that is not currently available, they provide all the information that you need to implement it.

1.5 Who Should Read this Book?

This book should be read by anyone wanting to use Webmin to manage their Linux or UNIX systems. It was written for readers with a basic knowledge of UNIX commands and concepts—people who have installed Linux and have used it for a while.

Each chapter starts with an introduction to the service being configured so that readers have some idea of what the DNS protocol is for or how a firewall works. Even so, a complete novice should not try to set up a server until he understands how it works and what he wants it to do. The best way to learn is to use the service on some other system as a user. For example, if you have used a proxy server before on some other network, then you will have the background knowledge needed to use this book to set up the Squid proxy on your own system.

The development chapters, on the other hand, are written for someone who already understands how to write Perl scripts and CGI programs on a UNIX system. This means that they are more complex than the rest of the book, and assume some knowledge of programming and manual system administration. They can be skipped, however, if you just want to learn how to *use* Webmin rather than how to extend it.

1.6 Conventions Used in this Book

The following special text styles are used in this book:

Bold Used for text that appears in Webmin itself, such as error messages, icon names, buttons, and field labels.

Fixed width This style is used for the names of shell commands, UNIX users, directories and files. Also used for text in configuration files, program code and API functions.

Italics Used to indicate example input entered by the user into Webmin, example commands, or directories. Also used in Chapter 60 “The Webmin API” for the names of parameters to functions.

1.7 Acknowledgments

This book could not have been written without the support of Jill Harry and the others at Prentice Hall, Robert Kern for suggesting the idea, my wife Foong Ching for her constant support, and all the members of the Webmin mailing list for their ideas and suggestions over the years.

Installing Webmin

This chapter explains how to download the appropriate Webmin package for your operating system, how to install it, and what you will see after logging in for the first time.

2.1 Downloading Webmin for Your System

The latest version of Webmin can always be downloaded from www.webmin.com/. At the time of writing, the latest release was Version 1.100, but new versions come out frequently. All of the instructions below, however, will use Version 1.100 for the filenames. If you download a later release, the version number in all the filenames and paths will have changed.

Some Linux distributions, such as Mandrake and Caldera, include Webmin as a standard feature, so it may already be installed on your system. The version that they include, however, may not be the latest official version that is available for download. If you are happy with the release that you already have, however, you can skip this chapter.

Other Linux distributions, like Debian and Gentoo, include Webmin as a package that can be downloaded and installed automatically. On Debian, the command `apt-get install webmin` will install the latest version available in the Debian APT repository. This can sometimes be a few versions behind the newest official release, however, so you may want to download from www.webmin.com/ instead. On Gentoo Linux, the command `emerge webmin` will install the latest version from the Gentoo Portage repository, which should be the same as the newest official release.

If you are upgrading from an older Webmin version, the process is exactly the same as installing for the first time. Any changes that you have made to the configuration of Webmin itself, or to other servers like Apache or Sendmail, will be left unharmed by the upgrade.

While Webmin supports a wide variety of UNIX variants, it does not cover all of them. Because it deals with system configuration files that differ in location and format between different

kinds of UNIX operating systems, it has been written to behave differently depending on the type of operating system that it is running on. To see a complete list of supported operating systems, visit the web page www.webmin.com/support.html. If your operating system is not on the list, you cannot use Webmin.

Before downloading Webmin for installation on your system, you have to choose a package format in which to download it. The available formats are:

RPM If you are running Red Hat, SuSE, Mandrake, Caldera, MSC, or any other Linux distribution which supports the RPM packaging format, then the RPM package is your best choice.

tar.gz The tar.gz packaged version of Webmin will work on any operating system, but is slightly harder to install than the RPM and Solaris packages.

Solaris package If you are running Solaris on Sparc or x86, then this is the package format for you.

For instructions on installing your chosen package type, see Section 2.2 “Installing the RPM Package” below.

2.2 Installing the RPM Package

In the top-right corner of every Webmin website page is a link for the RPM package. A link can also be found on the page www.webmin.com/download.html. Once you have downloaded it, you should have a file on your Linux system named something like `webmin-1.1.100-1.noarch.rpm`. To install, run the following command as `root`:

```
rpm -U webmin-1.1.100-1.noarch.rpm
```

The RPM install can only fail if you do not have Perl installed, or if Webmin cannot identify your operating system. If that occurs and your Linux distribution is on the list of supported operating systems, you should install the `tar.gz` version instead. Because all Linux distributions are slightly different, the Webmin install process has to positively identify the exact distribution and version that you are running, such as *Red Hat 7.3*. This can fail if one of the files that contain the distribution name (such as `/etc/issue`) has been modified.

Assuming the RPM install successfully completes, you will be able to login to Webmin immediately. Open a web browser, and go to the URL `http://localhost:10000/` if you are running the browser on the same Linux system on which Webmin was installed, or `http://your-systems-host-name:10000/` if the browser is being run on another PC. Either way, a web form will appear prompting for a username and password, as shown in Figure 2.1.

You should be able to login as `root`, using the same password as the `root` UNIX user on your Linux system. If the password is changed using the command-line `passwd` command or the Users and Groups module, your Webmin password will change too.

If the OpenSSL library and the Net::SSLeay Perl module have already been installed on your system, Webmin will automatically start in SSL mode. This means that you should use a URL starting with `https://` instead of `http://` to connect to it. Attempting to connect with the non-SSL URL will only bring up a page with a link to the `https://` URL on it, which you should follow to log in.



Figure 2.1 The Webmin login page.

2.3 Installing the tar.gz Package

In the top-right corner of every Webmin website page there is a link for the `tar.gz` package. A link can also be found on the page www.webmin.com/download.html. Once you have downloaded it, you should have a file on your system named something like `webmin-1.1.100.tar.gz`. To install the package, follow these steps:

1. Login to your system as `root`.
2. Choose a directory under which you want Webmin installed. This is usually `/usr/local`, but can be `/opt` or any other location that you prefer. The instructions below will use `/usr/local` for simplicity.
3. Copy the `webmin-1.1.100.tar.gz` file to the `/usr/local` directory.
4. Run the following commands to uncompress and extract the `tar.gz` file and run the following setup script:

```
cd /usr/local  
gunzip webmin-1.1.100.tar.gz  
tar xf webmin-1.1.100.tar  
cd webmin-1.1.100  
. ./setup.sh
```

5. After running the `setup.sh` script, you will be asked a series of questions that control the installation process. The questions and their meanings are:

Config file directory [/etc/webmin] This is the directory in which Webmin will store all of its own configuration files. It is best just to hit Enter to accept the default of /etc/webmin. If this directory already exists from an older version of Webmin that you are upgrading from, this is the only question that will be asked.

Log file directory [/var/webmin] This is the directory in which Webmin's log and process ID files will be stored. Just hit Enter to accept the default of /var/webmin for this one as well.

Full path to perl This is the location of the Perl executable on your system. If it is at /usr/bin/perl or /usr/local/bin/perl, then you can just type enter to accept the default. Otherwise, you must enter the full path to the Perl interpreter.

Operating system This question will only be asked if Webmin cannot automatically identify your operating system. You must enter the number next to one of the operating system names that appears in the list before the question.

Version Like the question above, this will only be asked if Webmin cannot identify your operating system. Again, you must enter the number next to one of the version numbers displayed.

Web server port (default 10000) This is the HTTP port on which Webmin listens. It is best to stick with the default, unless you are running some other network server on port 10000.

Login name (default admin) This is asking for the username that you will use for logging into Webmin. admin is the traditional username, but anything can be used.

Login password This is the password that must be entered along with the username. You must enter this twice, to verify that you haven't accidentally made a mistake.

Use SSL (y/n) This question will only be asked if you have already installed the OpenSSL and Net::SSLeay libraries on your system, as explained in Chapter 3. If you enter *y*, Webmin will use SSL right from the start. If you enter *n* now, however, you can still turn it on later.

Start Webmin at boot time (y/n) This question controls whether Webmin will be starting when your system boots up, which means that you do not have to re-start it yourself manually every time you reboot. If you want to have it started at boot, just enter *y*. If not, enter *n*.

6. After all the questions have been answered, the install process will finish, and a message showing the URL that you can use to log in will appear. You can now delete the old webmin-1.1.100.tar file if you no longer need it. Do **not** delete the /usr/local/webmin-1.1.100 directory that was created when the tar file was extracted, however. This contains all the scripts that Webmin needs to run.

Now that the package has been installed, you can open a web browser, and go to the URL <http://localhost:10000/> if you are running the browser on the same Linux system on which Webmin was installed, or <http://your-systems-hostname:10000/> if the browser is being run on another

PC. Either way, a web form will appear prompting for a username and password as shown in Figure 2.1. Log in using the username and password that you chose before in response to the **Login name** and **Login password** questions.

If you answered yes to the SSL question, you should use a URL starting with *https://* instead of *http://* to connect. If Webmin detects a non-SSL connection when it is in SSL mode, it will display a page with a link to the correct URL.

2.4 Installing the Solaris Package

The Solaris version of Webmin is only available for download from www.webmin.com/download.html. Once you have downloaded it, you should have a file on your Solaris system named something like `webmin-1.1.100-1.pkg.gz`. To install, run the following commands as `root`:

```
gunzip webmin-1.1.100.pkg.gz  
pkgadd -d webmin-1.1.100.pkg.gz WSwebmin
```

The Solaris package can only fail if you already have Webmin installed, or if you do not have the Perl executable at `/usr/local/bin/perl`. If you have Perl installed somewhere else on your system, you should create a symbolic link from `/usr/local/bin/perl` to the real location.

Assuming the Solaris package install completes successfully, you will be able to log in to Webmin immediately. Open a web browser, and go to the URL `http://localhost:10000/` if you are running the browser on the same Linux system on which Webmin was installed, or `http://your-systems-hostname:10000/` if the browser is being run on another PC. Either way, a web form will appear prompting for a username and password, as shown in Figure 2.1.

You should be able to login as `root`, using the same password as the `root` UNIX user on your Solaris system. If you change the UNIX `root` password down the road, however, the Webmin `root` user will not change. This is because the package install just copies the current password from the `/etc/shadow` file.

2.5 The Webmin User Interface

Assuming the installation process and login were successful, your browser should show the Webmin main menu with the **Webmin** category selected, as shown in Figure 2.2. You can switch to other categories by clicking on the icons along the top of the page, such as **System**, **Servers**, or **Others**. Every module is a member of one category, and a table of icons for each module in the selected category will appear in the body of the page. To enter a module, just click on its icon.

To log out of Webmin, just click on the **Logout** link that appears in the top-right corner of every page. To send feedback to the author (that's me), click on the **Feedback** link that is next to the **Logout** button. To visit www.webmin.com/, click on the **Webmin** logo in the top-left corner of any page.

If you are using a different theme, the user interface will appear different to the screen, as shown in Figure 2.2. Some versions of Webmin that come with Linux distributions use a different theme by default, such as Mandrake and Caldera. The main menu, however, will still show categories and modules, maybe using different sized icons in a different on-screen layout. All the screen shots in this book were captured using the default theme, so you may want to switch to it now (see Chapter 52 for instructions on how to change the current theme).

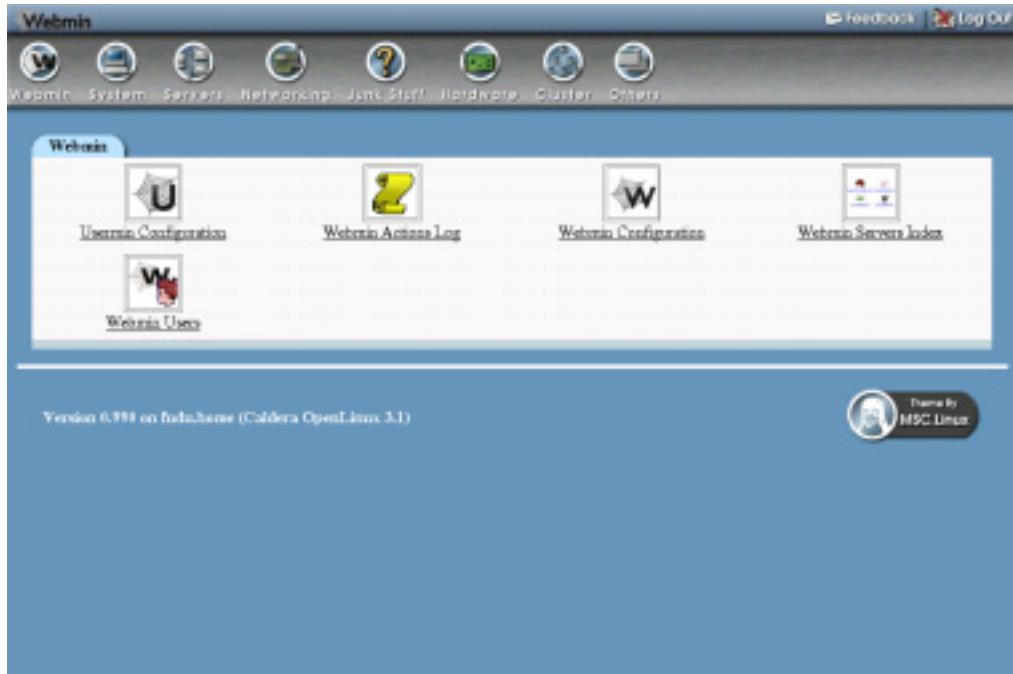


Figure 2.2 Modules in the Webmin category.

All Webmin modules have a common layout and user interface, in order to make navigation easier. When you click on a module icon from the main menu, the main page of the module will appear. For example, Figure 2.3 shows the main page of the Disk Quotas module.

At the top are the category icons that appear on every Webmin page, so that you can easily switch to another module. Below are links for **Help**, **Module Config**, and **Search Docs**. Not every module will display all of these links, but where they appear they have common purposes:

Help This link opens a pop-up window containing an overview of the module and the options available on the main page.

Module Config This link displays a form containing configurable options for the current module. See Figure 2.4 for an example of the options available in the Disk Quotas module. Each module has its own set of options, but all use a similar interface for editing them. In most cases, you will not need to change any of these configuration options for normal use of a module.

Search Docs This link displays a list of UNIX man pages, package documentation, HOWTO files, and websites related to the server or program that the module is configuring. This can be useful for finding out additional information about the underlying configuration files and commands that Webmin is using.

Other pages below the first page in each module also have a common layout. Figure 2.5 shows a sample page from the Disk Quotas module. Below the list of category icons is a link labeled **Module Index**, which will always return you to the module's main page. This can be found on almost every page of every module. Next to it is another **Help** link that pops up a window dis-

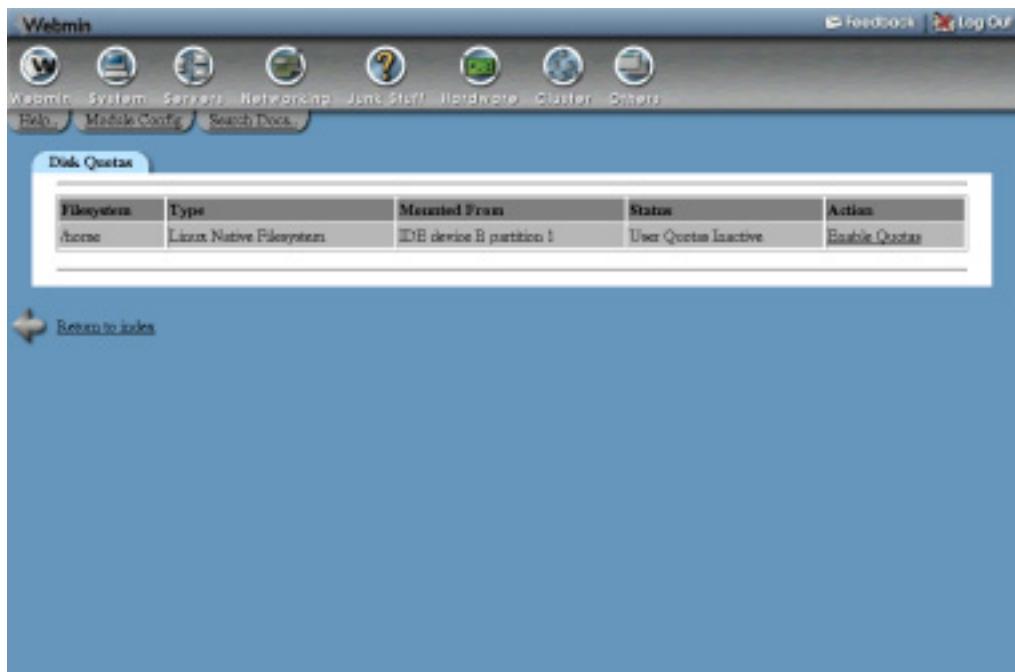


Figure 2.3 The Disk Quotas module main page.

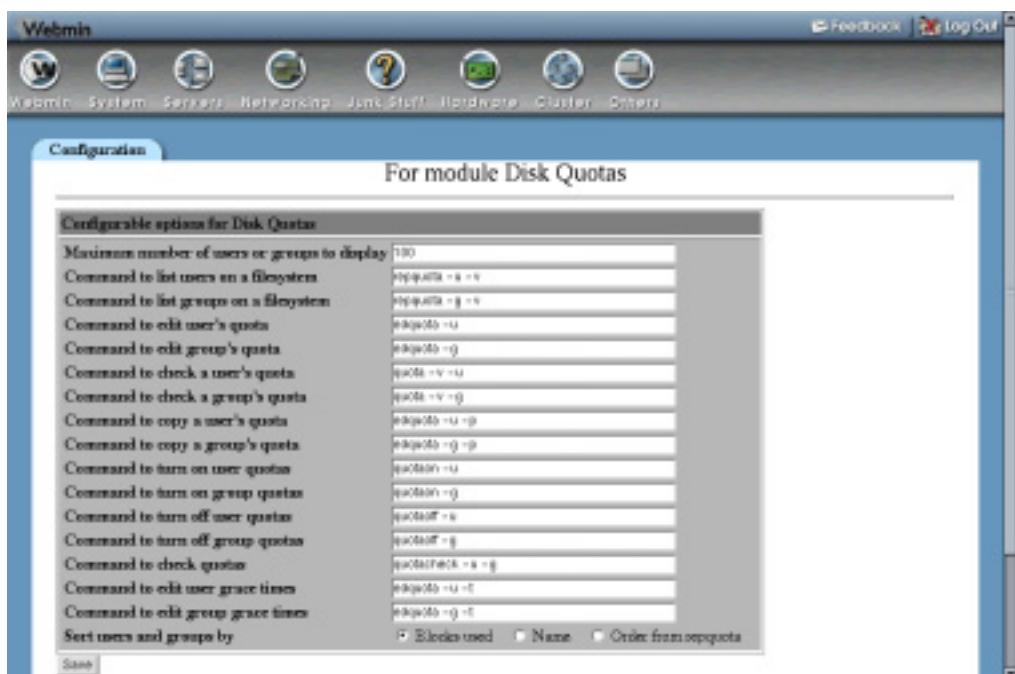


Figure 2.4 The configuration page for the Disk Quotas module.

The screenshot shows the Webmin interface with the title bar "Webmin". Below it is a toolbar with icons for Home, System, Servers, Networking, Junk Stuff, Hardware, Cluster, Others, Feedback, and Log Out. A sub-menu bar includes "Mobile Today" and "Help". The main content area is titled "User Quotas" and displays "All Quotas for jeameron". A table titled "Blocks" lists file system usage for the user "jeameron". The table has columns for Filesystem, Used, Soft Limit, and Hard Limit for both Blocks and Files. The data shows one entry for "/home/jeameron" with Used values of 2285495 and 111738, and Soft and Hard Limits set to Unlimited. Below the table are buttons for "Edit Quota On" and "Home", and a "Copy Quotas" button. At the bottom left is a "Return to filesystem list" link.

Blocks				Files		
Filesystem	Used	Soft Limit	Hard Limit	Used	Soft Limit	Hard Limit
/home/jeameron	2285495	Unlimited	Unlimited	111738	Unlimited	Unlimited

Figure 2.5 An example page from the Disk Quotas module.

playing information on the current page. Not all pages have online help, so this link will not always appear. Finally, at the bottom of the page is a link, whose label starts with **Return to**, that will take you back one level in the module's hierarchy of pages.

2.6 Uninstalling Webmin

If, for some unimaginable reason, you want to remove Webmin from your system, you can just log in as `root` and run the command:

```
/etc/webmin/uninstall.sh
```

This command will ask if you are sure you want to uninstall, and if you do it will delete the Webmin scripts and configuration directories. This means that any configuration you have done to Webmin itself, such as changing IP access control, switching themes, or creating new Webmin users will be lost. There will, however, be no harm done to the configuration of other servers such as Apache or Sendmail, even if they were done using Webmin.

2.7 Summary

After reading this chapter, you should understand how to install Webmin for the first time on a server, or upgrade an existing installation to the latest release. You should also know the differences between the three package formats, and which one is suitable for your operating system. Because this entire book is about Webmin, it should definitely be installed before reading on!

Securing Your Webmin Server

This chapter covers the necessary steps for adding additional security to Webmin on your system once it has been installed. It explains both IP address restrictions and the use of SSL.

3.1 Network Security

Unless you are running Webmin on a system that is never connected to any other network, it is a wise idea to restrict which client network addresses are allowed to log in. Because Webmin is so powerful, anyone who manages to log in will have total control over your system—as though they had `root` shell access. Even though a username and password is always required to log in, it is always good to have an additional layer of security in case an attacker guesses (or somehow discovers) your password. IP access control also protects you from any bugs in Webmin that may show up in future that will allow an attacker to log in without a password—some older releases have had just this problem.

To restrict the IP addresses and networks from which Webmin will accept connections, follow these steps:

1. In the **Webmin** category, click on the icon for the Webmin Configuration module.
2. Click on the icon for **IP Access Control**. The form shown in Figure 3.1 will appear for restricting client IP addresses.
3. Select the option **Only allow from listed addresses**, and enter the IP addresses or hostnames of client systems in the text box from which you will allow access. If you want to allow access from an entire IP network, enter the address of the network with 0 for the final octet. For example, if you wanted to allow all clients with IP addresses from `192.168.1.0` up to `192.168.0.255`, you would enter `192.168.1.0`.

Networks can also be entered in the standard network/netmask format, like `192.168.1.0/255.255.255.0`. You can also grant access from an entire domain by entering a wildcard hostname like `*.foo.com`, assuming that reverse IP address resolution has been set up for that domain.

- When done, click the **Save** button to apply your changes. Webmin will warn you if the restrictions will prevent the client system on which you are currently running your browser from logging in so you do not accidentally lock yourself out!



Figure 3.1 The IP access control form.

3.2 SSL Encryption

If you are accessing your Webmin server over an untrusted network such as the Internet, you should be aware that, by default, an attacker can capture your login and password by listening in on network traffic. This is particularly easy if you are using a non-switched Ethernet network shared by people that you do not fully trust, such as those in offices or universities.

Fortunately there is a solution that is relatively easy to set up—switching Webmin to use SSL so that all network traffic between your web browser and the server is encrypted. The RPM package of Webmin will run in SSL mode by default if the OpenSSL library and Net::SSLeay Perl module are installed. Most systems, however, do not meet these requirements so you will need to follow the steps below to enable SSL:

- Install the OpenSSL library, if you do not already have it. Most recent Linux distributions will include it as standard, but you may have to install it from your distribution CD.

If there are separate packages for `openssl` and `openssl-devel`, make sure both are installed. If your operating system does not come with OpenSSL, you can download it from www.openssl.org/ instead.

2. Install the Net::SSLeay Perl module, if it is not already installed. If your system is connected to the Internet, the easiest way to do this is to enter the **Perl Modules** module of Webmin (under the **Others** category), enter Net::SSLeay into the **From CPAN** field and click the **Install** button.
After the Perl module has finished downloading, click on **Continue with install** to have Webmin automatically compile and install it.
3. Once both are installed, go to the Webmin Configuration module and click on SSL Encryption. The form shown in Figure 3.2 will appear.
4. On the top part of the page, change the **Enable SSL if available?** option to **Yes**, and click **Save**. If all goes well, Webmin will be switched to SSL mode and your browser will connect to it securely.
5. If this is the first time you have connected to Webmin in SSL mode, your browser will display a warning about the certificate being invalid. For now, you can ignore this warning and choose to accept the certificate. For more details, see Section 3.3 “Requesting a Valid SSL Certificate”.
6. From now on, when logging into Webmin you must use a URL starting with `https://` instead of just `http://`. Once in SSL mode, it will no longer accept insecure connections.
7. Go back to the SSL Encryption page and scroll down to the second form. If a warning starting with **Because you are currently using the default Webmin SSL key...** is displayed, you definitely should continue following these steps to create your own private SSL certificate and key. If, however, it does not appear, then a private key was created at installation time and there is no need to go on reading.
8. If your system is always accessed using the same hostname in the URL, enter it into the **Server name in URL** field, such as `www.example.com`. This will cause the generated certificate to be associated only with that hostname. Otherwise select **Any hostname** to allow the certificate to be used with any URL hostname. This is more convenient, but slightly less secure.
9. In the **Email address** field, enter your email address—such as `joe@example.com`.
10. If appropriate, fill in the **Department** field with the name of the department or group within the organization to which this system belongs, such as *Network Engineering*. This can be left blank if inappropriate, such as on a home system.
11. In the **Organization** field, enter the name of the company or organization that owns this system, such as *Foo Corporation*. Again, this can be left blank if it makes no sense.
12. In the **State** field, enter the name of the state that your system is in, such as *California*.
13. In the **Country code** field, enter the two-letter code for the country in which the system resides, such as *US*.
14. Leave the **Write key to file** field unchanged, and the **Use new key immediately** field set to **Yes**.
15. Hit the **Create Now** button to generate a new key and certificate, write them to `/etc/webmin/miniserv.pem` and immediately activate them. Your browser will probably prompt you again to accept the new certificate.

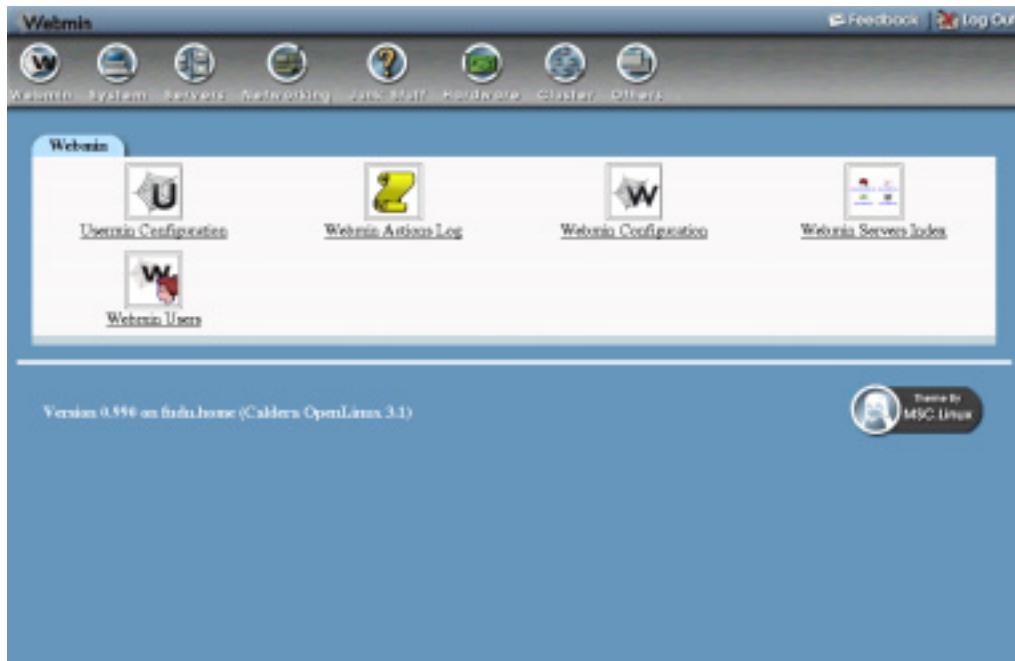


Figure 3.2 The SSL activation form.

Older versions of Webmin just used a fixed SSL key that was included as part of the package. This, however, was completely useless for securing network traffic because anyone with a copy of that key can decrypt the data that is supposedly protected with SSL! For this reason, recent Webmin versions create a new private key at installation time if possible, and warn you if the old fixed SSL key is being used.

3.3 Requesting a Valid SSL Certificate

If you want to use a valid SSL certificate and do not have one for your hostname, it is possible to generate one using the `openssl` command and a certificate authority. A valid certificate is one that is recognized by all browsers because it was signed by a recognized authority. Those created by Webmin itself, by following the steps in Section 3.2 “SSL Encryption”, do not meet this criteria and will trigger a warning in all browsers when they connect to the Webmin server.

Unfortunately, certificate authorities charge money for signing and verifying that the owner of the server in the hostname actually matches the company details in the certificate. For this reason, most people do not bother to use a signed certificate with Webmin, as there is no real advantage in security once you have accepted an unsigned certificate into your browser for the first time.

If you do want to obtain a real valid certificate, however, the steps to follow are:

1. At the shell prompt, run the `openssl genrsa -out key.pem 1024` command. This will create the `key.pem` file, which is your private key.

2. Run the `openssl req -new -key key.pem -out req.pem` command. When it asks for the common name, be sure to enter the full hostname of your server as used in the URL, like `www.yourserver.com`. This will create the `req.pem` file, which is the certificate signing request (CSR).
3. Send the CSR to your certificate authority by whatever method they use. They should send you back a file that starts with `-BEGIN CERTIFICATE-` which can be put in the `cert.pem` file.
4. In Webmin, enter the Webmin Configuration module and click on SSL Encryption.
5. In the SSL Encryption form (shown in Figure 3.2), enter the path to your `key.pem` file into the **Private key file** field, and the path to your `cert.pem` file into the **Certificate file** field.
6. Click the **Save** button to switch to the new certificate.

From now on, your browser should no longer display a warning when connecting to Webmin in SSL mode.

3.4 Summary

Securing your Webmin server to prevent unauthorized access is critical, as there are many potential attackers on the Internet who would love to use it to take over your system. This chapter has covered the two different types of security configuration (IP access control and SSL) that should be performed where possible. Because some versions of Webmin have had remotely exploitable security holes, it is also advisable to always upgrade to the latest version as soon as it becomes available to ensure your system's security.

Users and Groups

This chapter is devoted to the Users and Groups module, which allows you to create and manage UNIX user accounts and UNIX groups.

4.1 Introduction to UNIX Users and Groups

On Linux and other UNIX operating systems, a user is a person who can login to the system via SSH, telnet, FTP or at the console. Users can also receive email and own files on the server's local filesystems. Each user has a login name, a password, and a home directory in which all its files are stored. Users also have several additional attributes, such as a real name, shell (the program that is run when the user logs in), and expiry date.

Each user is a member of at least one group, called a primary group. In addition, a user can be a member of an unlimited number of secondary groups. Group membership can be used to control the files that a user can read and edit. For example, if two users are working on the same project you might put them in the same group so they can both edit a particular file that other users cannot access.

Every system will have several standard user accounts like `root` and `nobody` that are created when the system is installed—although most of these (except for `root`) cannot be used to login. If your server will be used by more than one person, you will need to create an additional user account for each person to keep their files and email separate. Even if you are the only person who uses your machine, it is a good idea to create a user account for yourself that you use to login with instead of using the `root` account.

Depending on your operating system, user and group information will be stored in different files in the `/etc` directory. On modern versions of Linux, `/etc/passwd` and `/etc/shadow` are used to store user details, and `/etc/group` for group details. The Users and Groups module works by directly editing those files, not by calling any external programs or functions. This means that if you are using NIS or storing users in an LDAP server, this module is not for you.

4.2 The Users and Groups Module

The Webmin module Users and Groups that is found under the **System** category (as shown in Figure 4.1) can be used to create, edit, and delete all the UNIX users and groups on your system. You should always be careful when using this module to edit existing system users like `root` and `daemon` because changing or deleting them could stop your system from working. Some users have their home directory set to `/` (the root directory). Deleting such a user would cause all the files on your system to be deleted!

In addition to managing the UNIX users on your system, this module can also affect user settings in other modules. For example, Samba has its own list of users and passwords that should be kept in sync with the UNIX password list. Webmin can handle this for you automatically using the **other modules** option that appears on the user creation, editing, and deletion forms. You must, however, enable this in every other module that you want automatically updated. The module also

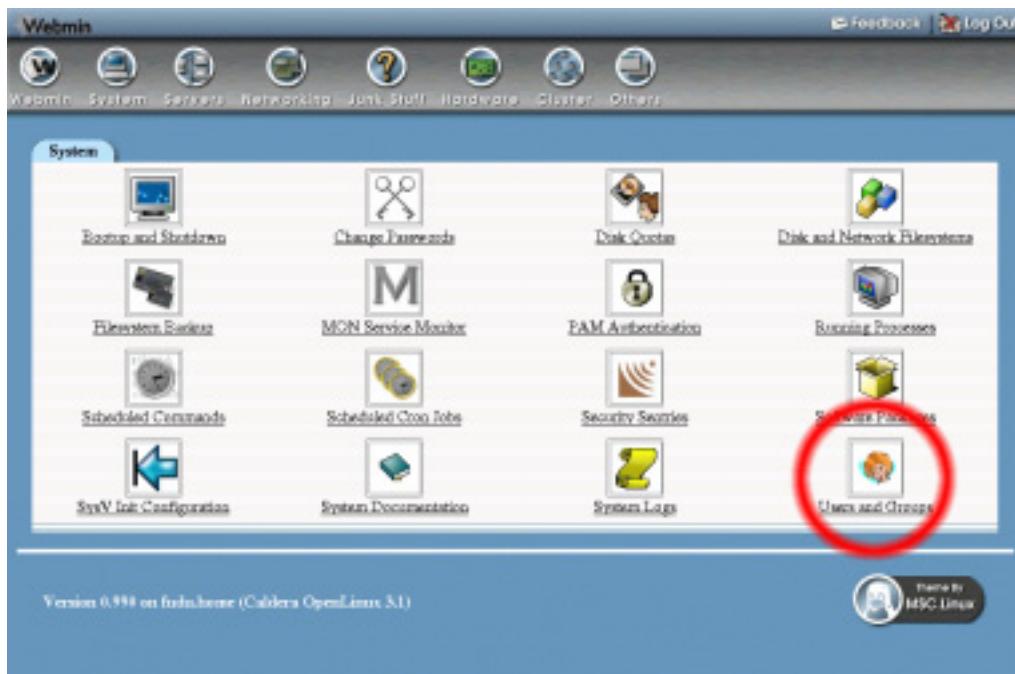


Figure 4.1 The Users and Groups module icon.

has options for synchronizing UNIX groups in a similar way, such as with Samba groups. However, since this feature only works with Samba 3.0, which is still under development, it is not covered in this chapter.

Once you enter the module, the main page lists all the users that currently exist on your system in one table (Figure 4.2), and all the groups in another (Figure 4.3). If there are too many users or groups to sensibly display in a table, then a small form allowing you to search for a user or group will be displayed instead.



The screenshot shows a table titled "Local Users" under the "Users and Groups" section of Webmin. The table has columns for Username, User ID, Real name, Home directory, and Shell. The data includes standard system users like root, bin, daemon, and many user accounts such as emily, foong, games, gdm, goather, halt, hdb, homeless, http, jcameron, jdesk, john.smith, lara, lp, and mail.

Local Users					
Create a new user: Create, modify and delete users from batch file					
Username	User ID	Real name	Home directory	Shell	
adm	3	adm	/root/adm		
bin	1	bin	/root/bin		
daemon	2	daemon	/root/daemon	/bin/false	
emily	3004	Emily Cameron	/home/emily	/bin/bash	
foong	3002	Foong Ching Chan	/home/foong	/bin/bash	
ftp	14	FTP User	/home/ftp		
games	12	games	/usr/games		
gdm	58	GDM user	/	/bin/false	
goather	13	gopher	/var/lib/gopher-data		
halt	?	halt	/sbin/halt	/bin/false	
hdb	3005	HDB Notes User	/home/hdb	/usr/bin/hdb	
homeless	3009	Homeless User	/dev/null	/bin/false	
httpd	55	HTTP Server	/	/bin/false	
jcameron	3001	Jamie Cameron	/home/jcameron	/bin/bash	
jdesk	3006	Ideak User	/home/jdesk	/usr/bin/kiosk	
john.smith	3007	John Smith	/home/john.smith	/bin/bash	
lara	3021	Lara Cameron	/home/lara	/bin/bash	
lp	4	lp	/var/spool/lpd		
mail	8	mail	/var/mail/mail		

Figure 4.2 List of existing users.

4.3 Creating a New User

To create a new UNIX user, complete the following steps:

1. Click on the **Create a new user link** above or below the table of existing users. A form for entering the details of the new user will appear, as shown in Figure 4.4.
2. At this point you have to decide on a username for the new user, which should be something simple without spaces in it—like *jcameron* or *jamie*—and not used by any other user. If your server is receiving email, the username determines the part of the user's email address to the left of the @. Enter your choice in the **Username** field.
3. The **User ID** field should generally be left unchanged, as it is worked out for you by Webmin. If you set it to the same user ID as another user, they will be able to access each other's files. This is generally not a good idea.
4. In the **Real name** field, you should enter the user's full name, such as *Jamie Cameron*.
5. Every user has a home directory, in which the user stores his personal documents and preference files. In the **Home directory** field, you should enter a directory that does not exist yet, such as */home/jcameron*. When the user is created, this directory will be created and its ownership granted to the new user.
If Webmin on your system offers an **Automatic** option for the home directory, it is generally best to stick with that.
6. The user's shell is a program that is run when he makes a text mode login of some kind (via SSH, for example), or opens a shell prompt after logging in graphically at the con-

Local Groups		
Create a new group		
Group name	Group ID	Members
adm	4	adm daemon
bin	1	bin daemon
bind	19	
daemon	2	bin daemon
database	37	
dip	40	
disk	6	
dmesg	3066	
lp	50	
mail	20	
gnome	58	
gopher	30	
inetd	101	
halt	62	
inetd	55	
ing	1000	jcameron fenton hdb squid saching
kern	9	
lp	7	daemon lp
mail	3004	
mailnull	32	
mailnull	36	
mailnull	41	mail

Figure 4.3 List of existing groups.

sole. The shell is responsible for running the commands that you type (such as `ls` and `cat`), running scripts on login and logout, and providing an interface for command editing. Shells like `bash` and `tcsh` are easier for users to use, because they allow the up and down arrows to be used to scroll through previous commands, and the tab key to auto-complete commands and filenames.

In some cases, you might not want a user to be able to make a shell login at all, as in when the user is only meant to be able to read and send email. In that case, his shell should be set to `/bin/false`, which is a program that does nothing and exits immediately.

You should select whatever shell you want the user to have from the list in the **Shell** field. If your choice is not on the list, select the **Other** option and enter the path to the shell in the field below.

7. For the **Password** field, you have four choices:

No password required The user can login without needing to enter any password.

No login allowed The user can never login.

Normal password You get to enter the user's password.

Pre-encrypted password You must enter a password that is already encrypted, such as one taken from the `/etc/shadow` file on another system.

Generally you will want to use the **Normal password** option. Note that on many operating systems, only the first eight characters of the password are actually used.

8. On most systems, a set of inputs under the heading **Password options** will be available. The first of these is the **Expiry date**—if you want the user to be unable to login after a particular date, fill in this field.
9. The **Minimum days** field is the number of days after the user is created or the password is last changed that the user must wait before changing it again. Leave it blank to allow changing as soon as the user wants.
10. The **Maximum days** field is the number of days after the user is created or the password is last changed that the password will expire and need to be changed. A user with this option set will be forced to change his password periodically, which is good for system security. Leave it blank to prevent the password from ever expiring.
11. The **Warning days** field is the number of days before the password expiry date that the user will be warned at login that his password is about to expire. If left blank, the user will not know that his account has expired until he tries to log in and is forced to choose a new password.
12. The **Inactive days** field is the number of days after the password expires that the entire account will be disabled if the user has not chosen a new password. If left empty, the account will never expire.
13. For the **Primary group**, either select an existing group or enter the name of a new one that Webmin will create for you.
14. If you want the user to be a member of more than one group, select some of the groups from the **Secondary group** list.
15. If you want the user's home directory to be created, select the **Create home directory?** option. If the directory does not already exist, you should select this as well as **Copy files to home directory?** so that the user gets a basic set of preference files like `.profile` and `Desktop`.
16. To create the user in other modules that you have configured for such action, select **Create user in other modules?** It is possible to set up the Samba module to automatically create a user in its user list, and the MySQL module to create a new database user, among others.
17. To create the user, click the **Create** button. After a short delay, you will be returned to the list of existing users, which should include your newly created user.

Once the **Create** button has been clicked, the new user will be able to login via SSH, telnet, or whatever other services you have set up

4.4 Editing an Existing User

You can change any of the details of any user that already exists on your system by following these steps:

1. Click on the user you want to edit from the existing list. A form containing all the details of the user will appear, as shown in Figure 4.5.
2. Change any of the details that you want to modify, including the username. The fields have the same meanings as described in Section 4.3 “Creating a New User”.
3. If you have modified the **User ID** or changed the **Primary group**, files owned by the user may need to be updated to use the new IDs. The options at the bottom of the page

Figure 4.4 The user creation form.

labeled **Change user ID on files?** and **Change group ID on files?** control which directories will be searched for files with the old IDs.

4. If you have changed the user's home directory, you can have Webmin rename it to the new path. However, if the new home directory already exists, this may not always be what you want. The **Move home directory if changed?** option determines if it is moved or not.
5. To have the user updated in other modules where this has been set up, select **Modify user in other modules?** If you are changing the username, this will also rename the user's Sendmail mail file and Cron jobs.
6. Click the **Save** button to have Webmin update the user. Once it is complete, you will be returned to the lists of users and groups.

4.5 Deleting a User

You should always be careful when deleting a user, as important files in the user's home directory may be lost. It is generally never a good idea to delete any of the users that are created when your system is first installed—especially `root`! Even normal users that you have created can be disabled by editing the user and setting the password option to **No login allowed**.

If you still want to go ahead and delete a user, follow these steps:

1. Click on the user you want to edit from the existing list. A form containing all the details of the user will appear, as shown in Figure 4.5.

Figure 4.5 The user editing form.

2. Click the **Delete** button at the bottom of the page. This will bring up a form asking you to confirm the deletion, with buttons to delete just the user or his home directory as well. The amount of disk space used by the user's home directory will be shown.
3. Select the **Delete user in other modules?** option if you want the user to be deleted from other modules in which deletion has been set up. Any Cron jobs belonging to the user will be deleted, as will his Sendmail mail file.
4. Click either the **Delete User** or **Delete User and Home Directory** button to delete the user. A page showing the progress of the deletion will be displayed while it is taking place.

4.6 Creating a New Group

A new UNIX group can be added by following these steps:

1. Click on the **Create a new group** link at the top or bottom of the existing list of groups. A form for entering the details of the group will appear, as shown in Figure 4.6.
2. Choose a name for the new group, and enter it into the **Group name** field. The name must not be used by any other group, and should be short and contain no spaces.
3. The **Group ID** field should be left alone, as it is automatically determined by Webmin. If for some reason you change it, make sure that it is not the same as any existing group's ID.
4. The **Password** field can be ignored, as group passwords are never used.

5. In the **Members** field, enter the names of any existing users that you want included in this group. You can use the button to the left of the field to pop up a selection window of all existing users.
6. Click the **Create** button to have Webmin create the new group. Once it is complete, you will be returned to the lists of users and groups.

Once the new group has been created, you can edit users to make it their primary group or one of their secondary groups.

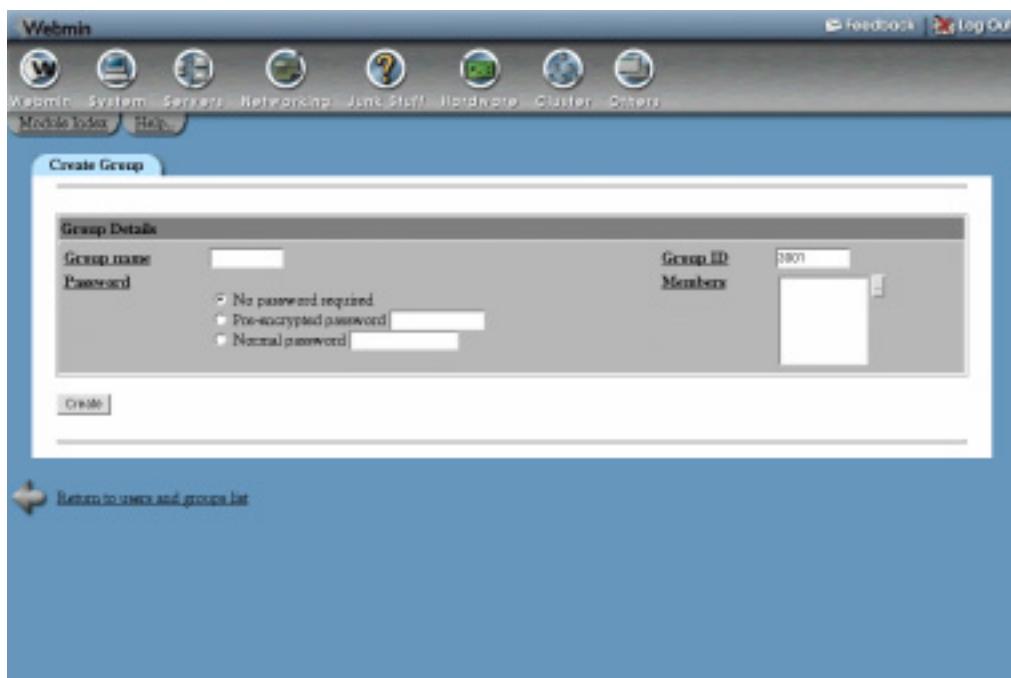


Figure 4.6 The group creation form.

4.7 Editing an Existing Group

You do not often need to edit an existing group, as users can be added to or removed from it by editing them directly. However, if you do want to edit a group, follow these steps:

1. Click on the name of the group that you want to edit from the list of existing groups. This will bring up the group editing form, as shown in Figure 4.7.
2. Change any of the details such as the group ID or member list. It is not possible to change the name of an existing group.
3. If you are changing the group ID, files owned by the group may need to be updated to use the new ID. Use the **Change group ID on files?** option to control which directories will be searched for files that need updating.
4. Click on the **Save** button to make the changes active. Once they are complete, you will be returned to the lists of users and groups.



Figure 4.7 The group editing form.

4.8 Deleting a Group

You can safely delete a group at any time, but Webmin will only let you do so if there are no users who have selected it as their primary group. To delete, follow these steps:

1. Click on the name of the group you want to delete from the list of existing groups. This will bring up the group editing form as shown in Figure 4.7.
2. Click the **Delete** button at the bottom of the page. A page asking if you really want to delete the group will appear.
3. Click the **Delete Group** button to confirm the deletion. A page showing the progress of the deletion will be displayed.

4.9 Viewing Recent and Current Logins

All UNIX systems keep track of recent logins made by users using SSH, telnet, or at the console. Some also track FTP logins as well. You can display recent user logins that include the date, time, and source address by following these steps:

1. Below the lists of users and groups, enter the username of the one you want to track into the **Display logins by** field, and click the button. If you want to see logins by ALL users, just leave the field blank.

2. A page listing recent logins by the user or users will be displayed. The list may not cover all logins from the date your system was first installed, as many operating systems automatically truncate the log file periodically in order to save disk space.

It is also possible to display a list of users who are currently logged in by clicking the **Logged In Users** button below the lists of users and groups. If a user is logged in graphically at the console, he may be listed multiple times—once for each shell window he has open.

4.10 Reading Users' Email

When editing a user, you can view mail in the user's mailbox by clicking on the **Read Email** button at the bottom of the page. This will take you directly to the mailbox viewing page of either the Sendmail, Qmail, or Postfix module, depending on what you have chosen for the **Display user email from** option in the module configuration. For more documentation on using the mail interface, see Chapter 37.

4.11 Creating Users from Batch Files

Sometimes you may want to create a large number of users at once without having to go through the process of filling out the user creation form over and over again. You will often have the details of these users in a text file of some kind containing their usernames, passwords, and real names. Fortunately, Webmin has a feature that automates this task for you.

If you click on the **Create, modify and delete users from batch file** link above or below the list of existing users, a form will appear that allows you to upload a file containing the details of users to create, as shown in Figure 4.8. Your file must contain one line of text for each user that you want to create, and the format of each line must match the format shown on the batch file page.

The exact file format depends on what information your system stores about each user, but on most systems each line must follow this format:

```
create:username:passwd:uid:gid:realname:homedir:shell:min:max:warn:inactive:expire
```

An example line to create a user with the user ID automatically assigned by Webmin would be:

```
create:jcameron:mysecret::3001:Jamie Cameron:/home/jcameron:/bin/bash::::
```

As you can see, the line is made up of a series of fields, each separated by a colon (:). When creating a user, the first field must be the `create` field. The meanings of the other fields are shown in Table 4.1.

Once you have created a file containing the details of users to create, select it using either the **Upload batch file** or **Local batch file** fields, and click the **Execute batch** button. A page displaying each user created and any errors encountered will be displayed. The most common error is a missing field in one of the lines—each must have exactly the right number of fields, and even if a field is blank the colon separator next to it must still be included.

This form allows you to create, modify or delete many users at once from an uploaded or local text file. Each line in the file specifies one action to take, depending on its first field. The line formats are:

- `create username [passwd] [uid] [gid] [realname] [home dir] [shell] [min/max login time] [inactive] [expire]`
- `modify [oldusername] [newusername] [passwd] [uid] [gid] [realname] [shell] [min/max login time] [inactive] [expire]`
- `delete [username]`

In create lines, if the `uid` field is left empty, Webmin will assign a UID automatically. If the `gid` field is empty, Webmin will create a new group with the same name as the user. The `username`, `homedir` and `shell` fields must be supplied for every user - all other fields are allowed to be empty. If the `passwd` field is blank, no password will be assigned for the user. If it contains just the letter `x`, the account will be locked. Otherwise, the text in the field will be taken as the cleartext password and encrypted.

In modify lines, an empty field will be taken to mean that the corresponding user attribute is not to be modified.

Upload batch file

Local batch file

Create, modify or delete users in other modules? Yes No

Create home directories for created users? Yes No

Copy files to home directories of created users? Yes No

Rename home directories of modified users? Yes No

Change UID on files of modified users? No Home directory All files

Change GID on files of modified users? No Home directory All files

Delete home directories of deleted users? Yes No

Passwords are already encrypted? Yes No

Figure 4.8 The batch file execution form.**Table 4.1** Batch File Fields and Their Meanings

username	The user's login name. This cannot be left blank.
passwd	The user's password. If this field is left blank, then no password will be needed for the user. If it contains just the letter x, then the user will be locked and no login allowed.
uid	User ID for the new user. This should be left blank, so Webmin can assign one automatically.
gid	ID of the user's primary group. This cannot be a group name, and cannot be left blank. If more than one GID is entered, the user will be added as a secondary member to all of those listed after the first one as well.
realname	The user's real name. Not mandatory, but should not be left blank.
homedir	A directory that is created with ownership assigned to the user. You can leave this blank if the module has been configured to assign home directories automatically.

Table 4.1 Batch File Fields and Their Meanings (Continued)

shell	The user's login shell. This field cannot be left blank.
min	The number of days after the user is created or the password is last changed that the user must wait before changing it again. Can be left blank to allow changing as soon as the user likes.
max	The number of days after the user is created or the password is last changed that the password expires and must be changed again. If left blank, the password will never expire.
warn	The number of days before the password expiry date that the user will be warned at login that his password is about to expire. If left blank, the user will not know that his password has expired until it happens.
inactive	The number of days after the password expires that the entire account will be disabled, if the user has not chosen a new password. If left empty, the account will never expire.
expire	The date on which this account will expire. Unfortunately, you must enter this as a number of days since January 1, 1970!

4.12 Configuring the Users and Groups Module

Like other Webmin modules, Users and Groups has several options that can be configured by clicking on the **Module Config** link above the lists of users and groups, as shown in Figure 4.9. The options that you can safely change and their meanings are shown in Table 4.2.

Table 4.2 Module Configuration Options

Command to run before making changes	Whatever shell command you enter into this field will be run just before any action is performed, such as adding, deleting, or modifying a user or group. It can be useful for doing things like making a backup copy of the <code>/etc/passwd</code> file before Webmin makes any changes. The command can determine exactly what Webmin is about to do by checking environment variables, as explained in the Section 4.13 “Before and After Commands”.
Command to run after making changes	Like the above option, but this command is run <i>after</i> any action is performed. It can be very useful if you want to have a command run after a user is created in order to setup additional files for that user.
Permissions on new home directories	The octal file permissions on newly created home directories, in the same format as used by the <code>chmod</code> command.

Table 4.2 Module Configuration Options (Continued)

Copy files into new home directories from	Directories or files to copy into the home directory of newly created users, assuming the Copy files to home directory? option is turned on. If any of the paths you enter is a directory, all files and subdirectories in that directory will be copied. This option is usually set to <code>/etc/skel</code> by default, which is a system directory containing files like <code>.cshrc</code> and <code>.profile</code> .
Automatic home directory base	The directory under which users' home directories are usually created. If this option is set, an Automatic option will appear for the Home directory field in the user creation form. If chosen, the home directory will be determined by this option and the Automatic home directory style below.
Automatic home directory style	This option controls the path to a new user's home directory under the base. The most common default option of <code>home/username</code> will make it just a subdirectory under the base, with the same name as the username. So if you were creating a user called <code>jcameron</code> and the home directory base was set to <code>/home</code> , then the resulting home directory would be <code>/home/jcameron</code> . Other options create subdirectories using the first one or two letters of the username. They can be useful if you have a very large number of users on your system, and want to avoid having thousands of entries in <code>/home</code> .
Lowest UID for new users	When Webmin automatically chooses a user ID for a new user, it will never pick one that is lower than specified in this option. On most systems, normal users have user IDs above 500, and system users have IDs below that.
Lowest GID for new groups	Like the option above, but for group IDs.
Create new group for new users?	If this option is set to Yes when creating a new user, the default action is to create a group of the same name and make it the user's primary group.
Assign same ID to new user and group?	This option only works if the previous one is enabled. If set to Yes when a new group is created for a new user, Webmin will make sure that their UID and GID are the same. This doesn't actually make any difference, but some administrators like it.
Don't use MD5 passwords if missing perl MD5 module?	This option should only be changed to Yes if you run into an error when creating a new user caused by a missing MD5 Perl module.
Check for send-mail alias clashes?	If set to Yes when creating or renaming a user, Webmin will check if there is a Sendmail alias of the same name. This can be useful to prevent the creation of users who would be unable to receive mail due to an alias redirecting it all to another address.
Only delete files owned by user?	If set to Yes when deleting a user, files in the user's home directory that do not belong to him will not be deleted.

Table 4.2 Module Configuration Options (Continued)

Maximum user and group name length	The maximum allowed length for a user or group name. If this is set by default, it is not a good idea to adjust it because your operating system will not recognize longer usernames.
Default group for new users	The default primary group on the new user creation form.
Default secondary groups for new users	A space separated list of secondary groups that will be selected by default on the new user creation form.
Default shell for new users	The default shell on the new user creation form.
Default minimum days for new users	The default number of days before which password changing is not allowed.
Default maximum days for new users	The default number of days after which the password must be changed.
Default warning days for new users	The default number of days before password expiry that the user is warned.
Default inactive days for new users	The default number of days after password expiry that the user is disabled.
Maximum number of users to display	If the number of users or groups on the module's main page exceeds this number, the table of users or groups will be replaced by a search form. You may want to adjust this if the number of users on your system is just over the default limit.
Sort users and groups by	This option controls the ordering of users and groups on the module's main page.
Number of previous logins to display	This option limits the number of recorded logins to display so the table does not become too large on systems that keep an unlimited login history.
Display users and groups by	By default, users and groups are shown on the module's main page in a table with one row per user or group. However, if you change this option to Name only then only the username of each appears, saving a lot of screen space if you have a large number of users. Changing to Primary group categorized also displays users by username only, but categorized by their primary group.
Conceal plain-text password?	If set to Yes when editing or creating a user, the Normal password field will show only stars instead of the actual password that you enter. Useful if you are worried about people looking over your shoulder when creating users.

Table 4.2 Module Configuration Options (Continued)

Get user and group info from	Even though the module reads and edits system user, group, and password files directly, there will in some cases be users and groups on your system that come from another source, such as NIS. When displaying a user's primary group or the users who are members of a group, Webmin will use the <code>getpw</code> family of system calls by default to get a list of users and groups, instead of reading the user and group files directly. This is normally the right thing to do, but in some cases it will not work properly or will be very slow. You should only change this option to Files if you are sure that you want the module to never use the <code>getpw</code> functions.
Generate password for new users?	If this option is set to Yes when creating a new user, Webmin will generate a random password for you by default.
Show office and phone details?	Normally, a user's Real name field only contains his name. However, it can also contain additional information such as his office location, home phone, and work phone. These extra fields are displayed by the <code>finger</code> command, and are stored by the system in the real name field of the <code>/etc/passwd</code> file separated by commas. If you want to be able to edit this additional information separately, set this option to Yes . It will not work well if usernames on your system contain commas in them—like <i>Cameron, Jamie</i> .
Display user email from	This option controls which module is used when the Read Email button is clicked on the user editing page. You should make sure it is set appropriately depending on the mail system you are using because Sendmail and Qmail use different locations and file formats for user mailboxes.
Minimum password length	If set, you will not be able to create or edit users whose plain-text passwords are shorter than this length. This option and the three below also effect the Change Passwords and Cluster Users and Groups modules. They can be useful if you want to delegate user management to someone else, and don't trust the quality of his passwords.
Prevent dictionary word passwords?	If this option is set, passwords that exactly match any word from the dictionary will not be allowed.
Perl regexp to check password against	If set, passwords must match this Perl regular expression. For example, you could enter <code>/0-9/</code> for this option to force all passwords to contain at least one digit.
Prevent password words containing username?	When this option is set to Yes , passwords that exactly match or contain the user's username will not be allowed.

The other options under the **System configuration** heading control the files Webmin reads and writes user and group information from and to. Because they are set automatically based on the type of operating system you use, they should not be changed unless you know what you are doing.



Figure 4.9 Configuration options for Users and Groups.

4.13 Before and After Commands

As Section 4.12 “Configuring the Users and Groups Module” explains, you can specify shell commands to be run before and after any action is taken in the module. Because these commands are called for every addition, modification, or deletion of a user or group, they need some way of telling exactly what action is being performed. They can do this using environment variables that are set before the command is run. The available environment variables are shown in Table 4.3.

If you wanted to send out email when a user is created, for example, you could set the **Command to run after making changes** option to:

```
[ "$USERADMIN_ACTION" = "CREATE_USER" ] && echo "Added user  
$USERADMIN_USER ($USERADMIN_REAL)" | mail -s "Added new user"  
you@yourdomain.com
```

4.14 Module Access Control

It is possible to grant a Webmin user or group access to only a subset of features in the Users and Groups module. This is most commonly used to allow a subadministrator the right to edit only

Table 4.3 Environment Variables for Before and After Commands

USERADMIN_ACTION	Indicates which action is being taken. Possible values are: CREATE_USER MODIFY_USER DELETE_USER CREATE_GROUP MODIFY_GROUP DELETE_GROUP
USERADMIN_USER	The username of the user being created, modified, or deleted. Not set when a group action is being performed.
USERADMIN_UID	The user ID of the user being created, modified, or deleted.
USERADMIN_GID	The group ID of the user.
USERADMIN_REAL	The real name of the user, including any office and phone information.
USERADMIN_SHELL	The shell of the user.
USERADMIN_HOME	The home directory of the user.
USERADMIN_PASS	The plain text password of the user, if available.
USERADMIN_SECONDARY	A comma-separated list of any secondary groups to which the user belongs.
USERADMIN_GROUP	The name of the group being added, modified, or deleted. Not set when a user action is being performed.

selected users and groups on the system, and to change their attributes in only limited ways. In a virtual hosting environment, for example, you may want to give a Webmin user the ability to create and edit up to 10 users with UIDs in a limited range, and home directories under a fixed directory. These privileges give the user no way to gain **root** access and affect users that do not belong to him.

Chapter 52 explains how to create additional Webmin users and edit their module access control in more detail. The following steps cover just the parts of the process that grant the kind of limited access that is specific to the Users and Groups module:

1. In the Webmin Users module, click on **Users and Groups** next to the name of the user that you want to edit. This will take you to the access control form shown in Figure 52.3.
2. Change the **Can edit module configuration?** field to **No**.

3. The **UNIX users who can be edited** field controls the users that can be changed by this Webmin user. You would typically set it to **Users with UIDs in range** and enter maximum and minimum UIDs into the fields next to it, such as 5000 and 5010.
4. To allow the addition of new UNIX users, set the **Can create new users?** field to **Yes**.
5. Set the **Can view batch file form?** option to **No**. This will prevent the Webmin user from creating and editing users from a batch script, which is not normally necessary. Allowing it, however, does not grant the user any additional privileges and is not a security risk.
6. For the **UIDs for new and modified users** fields, enter the same UIDs as in Step 4.
7. Deselect the **More than one user can have the same UID** option, but leave the **UIDs of existing users can be changed** option selected. An untrusted subadministrator should not normally be allowed to create multiple users with the same UID due to the problems that this can cause.

When UID clashes are prevented, the Webmin user will not be able to create any more UNIX users than fit in his allowed UID range.
8. In the **Allowed groups for new or modified users** field, you would typically select the **Only groups** option and enter the names of any groups of which new users can be primary or secondary members. Normally you would just enter a single group like *users*. Leaving this field set to **All groups** is a very bad idea, because it would allow the creation of users who are members of the `root` or `bin` groups, and who can thus edit important system files and executables. The **Groups with GIDs in range** option can be useful if this Webmin user is allowed to create multiple groups of his own within the same GID range.
9. To restrict the shells that a new user can be assigned, set the **Allowed shells for new or modified users** to **Listed** and enter their paths into the text box below. This can be useful to allow the creation of only mail-only users who always have the shell `/bin/false`.
10. Set the **Home directories must be under** field to a directory that will only be used for accounts created by this Webmin user. Setting it to `/home` is a bad idea, because this would allow the subadministrator to rename or delete directories belonging to other users that are under `/home`. Instead, enter something like `/home/subadmin`.

To force every user's home directory to be based on his username (such as `/home/subadmin/username`), check the **Home directory is always same as username** box.
11. To stop the Webmin user from deselecting some of the options at the bottom of the user creation, editing and deletion forms, deselect the matching **Allowed on save options**. Any that are not chosen will effectively always be turned on.
12. Assuming you just want the Webmin user to create and edit UNIX users, set the **UNIX groups who can be edited** field to **No groups**.
13. If you want to restrict the user from viewing recent logins, change the **Can display logins by** field. Any user who can login with telnet or SSH can run the last command anyway to display logins, so setting this option to **No users** does not usually make your system any more secure.
14. Finally, click **Save**. You will be returned to the module's main page and the new access control restrictions will be immediately applied to the Webmin user.

Be careful when granting a Webmin user access to certain UNIX users, as a mistake may allow him to edit the `root` user or create a new user who is equivalent to `root`. There are also many

other users like bin, uucp, and httpd that own important system files or are used for running server and daemon processes. Someone who can edit or login as one of these users could gain **root** privileges on your system or access files that he is not supposed to.

Often the access control in the Disk Quotas and Scheduled Cron Jobs module is set up to allow editing of the quotas and Cron jobs of the same UNIX users as those that can be edited and created in this module. All modules support the UID range and primary group access control options, which can be set in the same way.

It is also possible to use the Users and Groups access control form to allow a user to edit or create selected UNIX groups, though this is not generally as useful. Granting an untrusted user the rights to edit all groups on the system is a bad idea, as he would make himself a member of the root or bin group and so be able to read or write critical files.

4.15 Other Operating Systems

Different operating systems store different information about users than Linux does. This is due to the different files and file formats used for storing user information. Some, for example, do not have an /etc/shadow file, meaning that information about password change and expiry times does not exist. The list below explains the major differences between other supported operating systems and Linux:

FreeBSD, OpenBSD and NetBSD All these operating systems use the /etc/master.passwd file for storing user information, which combines /etc/passwd with some fields from /etc/shadow. When editing or creating a user, you can enter a **Password change time** which is the date and time after which the password must be next changed, and an **Account expiry time** after which an account can no longer be used. Each user can also have a **Login class**, which is used in conjunction with the /etc/login.conf file to determine memory, CPU, and other limits.

Sun Solaris and SCO UnixWare Both these operating systems use the same files and formats as Linux, and so have all the same options.

HP/UX, SGI Irix, and Compaq Tru64/OSF1 Because none of these systems use an /etc/shadow file by default, none of the options related to password and account expiration are available when editing or creating a user.

Apple MacOS X OSX does not store user and group information in files at all—instead, it uses a network database called NetInfo, which Webmin manipulates using the nidump and niutil commands. This database, however, stores the same information as the BSD master.passwd file, so when editing or creating a user the same fields are available as for FreeBSD.

IBM AIX AIX uses the files /etc/passwd and /etc/security/passwd for storing user information. Therefore, when editing or creating users on AIX there are some options that do not exist on other operating systems. The **Expiry date** field can be used to set the date and time after which the account cannot be used. The **Minimum weeks** and **Maximum weeks** fields are very similar to the **Maximum days** and **Minimum days** fields on Linux, but deal with weeks instead of days. The **Warning days** field has exactly the same meaning as on Linux, and deals with days

not weeks. The unique **Account flags** field sets special options whose meanings are explained on the form.

SCO OpenServer OpenServer uses `/etc/passwd` and `/etc/shadow` files, but the shadow file stores slightly different information than on Linux. This means that when editing a user, the **Expiry date** field is replaced with an option to control whether the user is prompted for a password at their next login, and the **Warning days** and **Inactive days** fields are not available.

Those few operating systems that are not listed above cannot use the Users and Groups module, as their file formats are not currently known to Webmin.

4.16 Summary

This chapter has explained how to create and manage users and groups on a UNIX system. Because they are used to enforce file security, to protect processes from each other, and as mailboxes, user management is one of the most important tasks on a multi-user server system. This means that the module covered in this chapter is one of the most commonly used in Webmin, and also one of the most powerful.

Disk and Network Filesystems

The chapter explains how to mount filesystems, either from partitions on your system's hard disks or from other file servers.

5.1 Introduction to Filesystems

On a UNIX system, all files exist in a tree of directories under the root / directory. Drive letters used by other operating systems (like Windows) to identify different hard disks or network drives do not exist. Instead, different hard disks, CD-ROMs, floppy disks, and network drives are attached to the directory tree at different places, called **mount points**. For example, /home may be a mount point for a different hard disk on your system, and /usr/local may be the mount point for files that are shared from another server. The root directory is also a mount point, almost always for a partition on a hard disk in your machine. The set of files that is actually mounted at a mount point is called a **filesystem**.

All operating systems divide each hard disk up into partitions, each of which can be a different size. Each filesystem is normally stored on one partition of one disk, so it is possible to have multiple filesystems of different types on the same hard disk—one for Linux and one for Windows, for example. If you have multiple hard disks in your system, you will normally need to mount at least one filesystem from each in order to make use of them.

UNIX systems support many different kinds of filesystems—some for files stored on local hard disks and some for files on networked file servers. On Linux, the filesystems on your hard disks will probably be in ext 2 or ext 3 format. Many other local filesystem types exist, such as iso-9660 for CD-ROMs, vfat for Windows partitions, and xfs and reiserfs for high performance file access. Every local filesystem type uses a different format for storing data on disk, so if a partition has been formatted as a filesystem of a particular type, then it must be mounted as that type.

There are also filesystem types for different methods of accessing file servers across a network. If the file server is running UNIX, then an nfs filesystem is usually mounted to access its files. How-

ever, if it is running Windows, an `smbfs` filesystem must be used instead. These different filesystem types correspond to different network protocols for accessing files on another system.

Other special filesystem types contain files that do not actually exist on any disk or file server. For example, a `proc` filesystem contains files that contain information about currently running processes. Different UNIX variants have different types of special filesystems, most of which are automatically mounted by the operating system and do not need to be configured.

No explanation of filesystems can be complete without also covering **virtual memory**. Often a UNIX system will be running processes that take up more memory than is actually installed. This is made possible by the operating system automatically moving some of those processes out of real memory and into virtual memory, which is stored in a file or a local hard disk. Because filesystems and virtual memory are both stored on disk and can be mounted and unmounted, the Disk and Network Filesystems Webmin module also manages with virtual memory.

Depending on your operating system, the files `/etc/fstab` or `/etc/vfstab` contain a list of filesystems that are known to your system and mounted at boot time. It is also possible for a filesystem to be temporarily mounted using the `mount` command without being stored in the `fstab` file. Webmin directly modifies this file to manage filesystems that are mounted at boot time, and calls the `mount` and `umount` commands to immediately activate and deactivate filesystems.

5.2 The Disk and Network Filesystems Module

The Disks and Network Filesystems module is found under the System category, and allows you to configure which filesystems are mounted on your computer, where they are mounted from and what options they have set. The main page of the module (shown in Figure 5.1) lists all the filesystems that are currently mounted or available to be mounted.

For each filesystem, the following information is displayed:

Mounted As The mount point directory for this filesystem, or the message Virtual Memory.

Type A description of the filesystem type, followed by the actual short type name.

Location The disk, filesserver, or other location from which this filesystem was mounted. For `nfs` mounts, this column will be in the form `servername:remotedirectory`. For `smbfs` mounts, it will be similar to `\\\servername\sharename`.

In use? Yes or No, depending on whether the filesystem is currently mounted. For most filesystems, you can click on this field to mount or unmount immediately.

Permanent Yes or No, depending on whether the filesystem is permanently recorded so that it can be mounted at boot time.

5.3 Mounting an NFS Network Filesystem

Before you can mount a filesystem from another UNIX server, that server must be configured to export the directory that you want to mount using NFS. For details on how to export a directory using Webmin, see Chapter 6.

Assuming the directory that you want to mount has been exported properly, you can follow the following steps to mount it on your system:

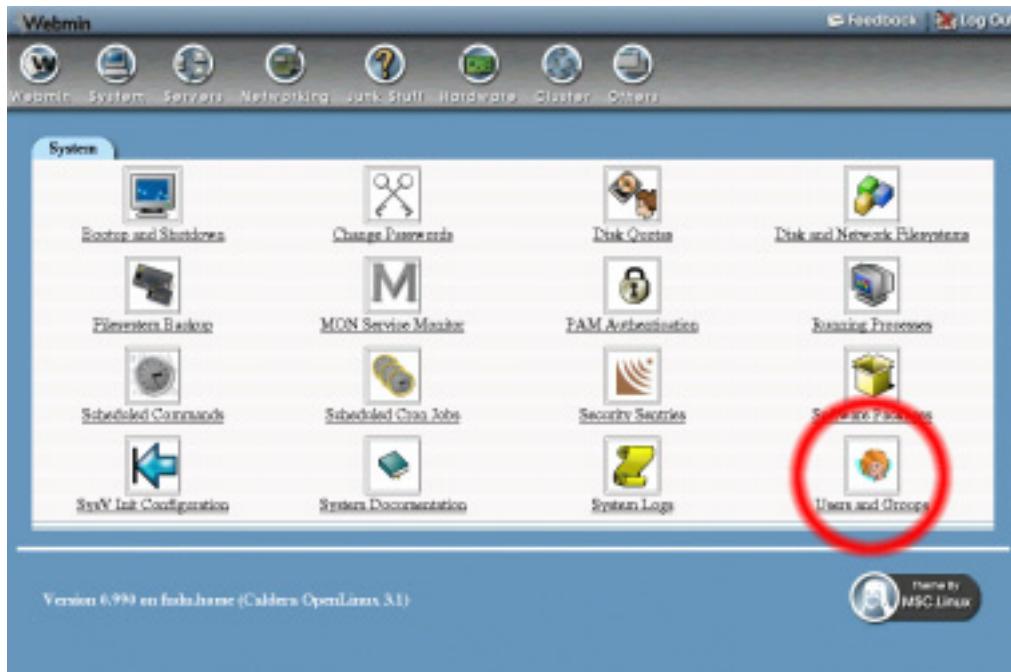


Figure 5.1 The list of existing filesystems.

1. On the main page of the Disk and Network Filesystems module, select **Network Filesystem** from the dropdown box of filesystem types, and click the **Add mount** button. A form will appear, as shown in Figure 5.2.
2. In the **Mounted As** field, enter the directory on which you want the filesystem to be mounted. The directory should be either nonexistent or empty, as any files that it currently contains will be hidden once the filesystem is mounted.
3. If you want the filesystem to be mounted at boot time, select **Save and mount at boot** for the **Save Mount** option. If you want it to be permanently recorded but not mounted at boot, select **Save**. Select **Don't save** if this is to be only a temporary mount.
4. For the **Mount now?** option, select **Mount** if you want the filesystem to be mounted immediately, or **Don't mount** if you just want it to be recorded for future mounting at boot time. It makes no sense to set the **Save and mount** option to **Don't save** and the **Mount now?** option to **Don't mount**, as nothing will be done!
5. In the **NFS Hostname** field, enter the name or IP address of the fileserver that is exporting the directory that you want to mount. You can also click on the button next to the field to pop up a list of NFS servers on your local network.
6. In the **NFS Directory** field, enter the exported directory on the fileserver. If you have already entered the NFS server's hostname, click on the button next to the field to pop up a list of directories that the server has exported.
7. Change any of the options in the bottom section of the form that you want to enable. Some of the most useful are as follows:



The screenshot shows a web-based interface titled "Users and Groups". A sub-tab "Local Users" is selected. Below it, a sub-instruction "Create a new user: Create, modify and delete users from batch file" is visible. The main area displays a table of local users with the following columns: Username, User ID, Real name, Home directory, and Shell. The data is as follows:

Username	User ID	Real name	Home directory	Shell
adm	3	adm	/var/adm	/bin/false
bin	1	bin	/bin	/bin/false
root	0	Root	/	/bin/sh
daemon	2	daemon	/sbin	/bin/false
exuser	3004	Emily Camerson	/home/exuser	/bin/bash
ftchao	3002	Faang Ching Chan	/home/ftchao	/bin/bash
ftp	14	FTP User	/home/ftp	/bin/false
games	12	games	/usr/games	/bin/false
gin	58	GDM user	/	/bin/false
gopher	13	gopher	/usr/lib/gopher-data	/bin/false
halt	7	halt	/sbin	/bin/halt
hdb	3005	HDB Notes User	/home/hdb	/bin/bash/hdb
homeless	3009	Homeless User	/dev/null	/bin/false
httpd	55	HTTP Server	/	/bin/false
jcameron	3001	Jamie Cameron	/home/jcameron	/bin/bash
jdesk	3006	Desk User	/home/jdesk	/bin/false/jdesk
johnsmith	3007	John Smith	/home/john.smith	/bin/bash
lara	3021	Lara Curwens	/home/lara	/bin/bash
lp	4	lp	/var/spool/lpd	/bin/false
mail	8	mail	/var/spool/mail	/bin/false

Figure 5.2 Mounting a network filesystem.

Read-only? If set to Yes, files on this filesystem cannot be modified, renamed, or deleted.

Retry mounts in background? When an NFS filesystem is mounted at boot time, your system will normally try to contact the fileserver forever and ever if it is down or unreachable, which can prevent the boot process from completing properly. Setting this option to Yes will prevent this problem by having the mount retried in the background if it takes too long.

Return error on timeouts? The normal behavior of the NFS filesystem in the face of a fileserver failure is to keep trying to read or write the requested information until the server comes back up again and the operation succeeds. This means that if the fileserver goes down for a long period of time, any attempt to access files mounted from the server will get stuck. Setting this option to Yes changes this behavior so that your system will eventually give up on operations that take too long.

8. To mount and/or record the filesystem, click the **Create** button at the bottom of the page. If all goes well, you will be returned to the filesystems list, otherwise an error will be displayed explaining what went wrong.

Once the NFS filesystem has been successfully mounted, all users and programs on your system will be able to access files on the fileserver under the mount point directory. If users can log in to both your system and the remote fileserver, any files that they own on one machine should be

owned on the other because the NFS protocol supports UNIX file permissions and file ownership information. This depends, however, on every user having the same user ID on both servers. If this is not the case, you may end up in a situation in which user jcameron owns a file on the files server, but the file appears to be owned by user fred when it is mounted and accessed on your system.

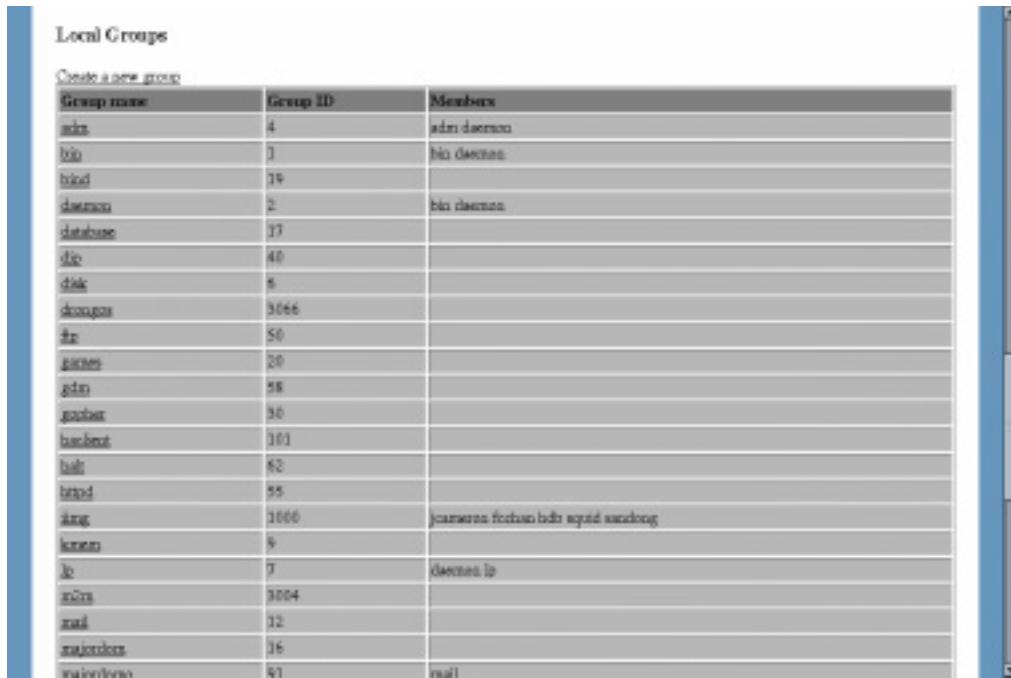
The best solution to this problem is to make sure that user IDs are in sync across all servers that share files using NFS. The best ways to do that are using NIS (as explained in Chapter 17), or Webmin's own Cluster Users and Groups module (as explained in Chapter 49).

5.4 Mounting an SMBFS Windows Networking Filesystem

smbfs is the protocol used by Windows systems to share files with each other. If you have files on a Windows system that you want to be able to access on your Linux system, you must first share the directory and assign it a share name using the Windows user interface.

Once that is done, follow these steps to mount the share on your UNIX system:

1. On the main page of the Disk and Network Filesystems module, select **Windows Networking Filesystem** from the drop-down box of filesystem types and click the **Add** button. A form will appear, as shown in Figure 5.3.
2. In the **Mounted As** field, enter the directory on which you want the filesystem to be mounted. The directory should be either nonexistent or empty because any files that it currently contains will be hidden once the filesystem is mounted.
3. If you want the filesystem to be mounted at boot time, select **Save and mount at boot** for the **Save Mount** option. If you want it to be permanently recorded but not mounted at boot, select **Save**. Select **Don't save** if this is to be only a temporary mount.
4. For the **Mount now?** option, select **Mount** if you want the filesystem to be mounted immediately, or **Don't mount** if you just want it to be recorded for future mounting at boot time.
5. In the **Server Name** field, enter the hostname or IP address of the Windows server. The button next to the field will pop up a list of Windows servers on your network, requested from the domain or workgroup master set in the module configuration.
6. In the **Share Name** field, enter the name of the share. This will be something like `movies`, not the full path on the Windows server like `c:\files\movies`. If you have entered the server name, clicking on the button next to the field will pop up a list of available shares.
7. If the Windows server requires a username and password to access the file share, fill in the **Login Name** and **Login Password** fields. If no authentication is needed, these fields can be left blank.
8. Because Windows networking has no concept of UNIX users, all files from the files server will be owned by a single UNIX user and group when the filesystem is mounted. That user is `root` by default, but you can change this by filling in the **User files are owned by** and **Group files are owned by** fields.
9. Click the **Create** button at the bottom of the page to mount and/or record the filesystem. If all goes well, you will be returned to the filesystems list, otherwise an error will be displayed explaining what went wrong.



The screenshot shows a table titled "Local Groups" with a "Create a new group" button at the top. The table has three columns: "Group name", "Group ID", and "Members". The "Members" column lists users separated by commas. A vertical scroll bar is visible on the right side of the table.

Local Groups		
Create a new group		
Group name	Group ID	Members
adm	4	adm, daemon
bio	3	bio, daemon
bind	19	
daemon	2	bio, daemon
database	37	
dip	40	
disk	6	
documentation	3066	
ftp	50	
games	20	
gdm	58	
gopher	30	
hadoop	101	
halt	62	
inetd	55	
icing	1000	jcameron, fenton, hdb, squid, www, www
irc	9	
lp	7	daemon, lp
mail	3004	
mailman	32	
mailnull	36	
mailnews	41	mail

Figure 5.3 Mounting a windows networking filesystem.

Windows networking filesystems can also be exported by UNIX servers using Samba, as explained in Chapter 43. This means that you could share files between two UNIX servers using the Windows file sharing protocol. As you might guess, however, this is not usually a good idea because file permissions and ownership information will not be available on the mounting server.

5.5 Mounting a Local ext2 or ext3 Hard Disk Filesystem

Before you can mount a new filesystem from a local hard disk, a partition must have been prepared and formatted with the corrected filesystem type. For details on how to do this, see Chapter 8. If you have a choice, `ext3` (called the New Linux Native Filesystem by Webmin) should be used instead of `ext2` (the Linux Native Filesystem) because of its support for journaling. See Section 5.13 “A Comparison of Filesystem Types” for more details on the advantages of `ext3`.

To mount your local filesystem, follow these steps:

1. On the main page of the Disk and Network Filesystems module, select **Linux Native Filesystem** or **New Linux Native Filesystem** from the drop-down box of filesystem types, and click the **Add mount** button. A form will appear for entering the mount point, source, and options.
2. In the **Mounted As** field, enter the directory on which you want the filesystem to be mounted. The directory should be either nonexistent or empty because any files that it currently contains will be hidden once the filesystem is mounted.

3. If you want the filesystem to be mounted at boot time, select **Save and mount at boot** for the **Save Mount** option. If you want it to be permanently recorded but not mounted at boot, select **Save**. Select **Don't save** if this is to be only a temporary mount.
4. For the **Mount now?** option, select **Mount** if you want the filesystem to be mounted immediately, or **Don't mount** if you just want it to be recorded for future mounting at boot time.
5. If the **Check filesystem at boot?** option exists, it controls whether the filesystem is validated with the `fsck` command at boot time before mounting. If your system crashes or loses power, any `ext2` or `ufs` filesystems that were mounted at the time will need to be checked before they can be mounted. It is generally best to set this option to **Check second**.
6. For the **Linux Native Filesystem** field, click on the **Disk** option and select the partition which has been formatted for your new filesystem. All IDE and SCSI disks will appear in the menu.

If any of the partitions on your system are labeled, you can mount one by selecting the **Partition labeled** option and choosing the one you want. Labels are explained further in Chapter 8.

If your system has any RAID devices configured (as also explained in Chapter 8), you can select the **RAID device** option and choose the one you want to mount from the menu.

If you are using LVM (Logical Volume Management, covered in Chapter 8), a list of all available logical volumes will appear next to the **LVM logical volume** option for you to select from.

You can also click on the **Other device** option and enter the path to the device file for your filesystem, like `/dev/hda2`.
7. Change any of the options in the bottom section of the form that you want to enable. Some of the most useful are:
 - Read-only?** If set to Yes, files on this filesystem cannot be modified, renamed, or deleted.
 - Use quotas?** If you want to enforce disk quotas on this filesystem, you must enable this option. Most filesystem types will give you the choice of user quotas, group quotas, or both. To complete the process of activating and configuring quotas, see Chapter 7.
8. Click the **Create** button at the bottom of the page to mount and/or record the filesystem. If all goes well, you will be returned to the filesystems list, otherwise an error will be displayed explaining what went wrong.

5.6 Mounting a Local Windows Hard Disk Filesystem

If your system has a Windows partition on one of its hard disks, you can mount it using Webmin so that all the files are easily accessible to UNIX users and programs. Windows 95, 98 and ME all use the older `vfat` format by default, called a Windows 95 filesystem by Webmin. Windows NT, 2000, and XP, however, use the more advanced `ntfs` filesystem format (called Windows NT filesystem) which only a few Linux distributions support.

1. On the main page of the Disk and Network Filesystems module, select either **Windows 95 Filesystem** or **Windows NT Filesystem** from the drop-down box of filesystem types, and click the **Add mount** button. A form will appear for entering the mount point, source, and options.

2. In the **Mounted As** field, enter the directory on which you want the filesystem to be mounted. The directory should be either nonexistent or empty because any files that it currently contains will be hidden once the filesystem is mounted.
3. If you want the filesystem to be mounted at boot time, select **Save and mount at boot** for the **Save Mount** option. If you want it to be permanently recorded but not mounted at boot, select **Save**. Select **Don't save** if this is to be only a temporary mount.
4. For the **Mount now?** option, select **Mount** if you want the filesystem to be mounted immediately, or **Don't mount** if you just want it to be recorded for future mounting at boot time.
5. For the **Windows 95 Filesystem** or **Windows NT Filesystem** field, click on the **Disk** option and select the partition that has been formatted for your new filesystem. All IDE and SCSI disks, RAID devices, and LVM logical volumes will appear in the list.
You can also click on the **Other device** option and enter the path to the device file for your filesystem, like `/dev/hda2`.
6. Select any options that you want to enable. Some useful ones are:
 - User files are owned by** Because the vfat filesystem format has no concept of users and groups, all files in the mounted filesystem will, by default, be owned by `root`. To change this, enter a different UNIX username for this option.
 - Group files are owned by** Like the previous option, this controls the group ownership of all files in the mounted filesystem.
 - File permissions mask** The binary inverse in octal of the UNIX permissions that you want files in the mounted filesystem to have. For example, entering `007` would make files readable and writable by their user and group, but totally inaccessible to everyone else. This option is not available for Windows NT filesystems.
7. Click the **Create** button at the bottom of the page to mount and/or record the filesystem. If all goes well, you will be returned to the filesystems list, otherwise an error will be displayed explaining what went wrong.

Because Windows 95 filesystems have no concept of file ownership, and Windows NT filesystems have ownership information that is unsupported by Linux, it is impossible to change the user, group, or permissions on files in a mounted filesystem.

5.7 Adding Virtual Memory

As explained in the introduction, virtual memory is used when the processes running on your system need to use more memory than is physically installed. Because not all processes run at the same time, those that are inactive can be safely swapped out to virtual memory and then swapped back in again when they need to run. Because disks are far slower than RAM, however, the constant swapping in and out (known as thrashing) will slow the system to a crawl if processes on your system use up too much memory.

Files in an existing local filesystem as well as entire partitions can be used for virtual memory. Using a partition is almost always faster, but can be inflexible if you have no free partitions on your hard disk. A system can have more than one virtual memory file or partition, so if you are running out of virtual memory it is easy to add more.

The steps for adding additional virtual memory are:

1. On the main page of the Disk and Network Filesystems module, select **Virtual Memory** from the drop-down box of filesystem types, and click the **Add mount** button. A form will appear for entering the source and other options.
2. If you want the virtual memory to be added at boot time, select **Save and mount at boot** for the **Save Mount** option. Otherwise, select **Don't save** if this is to be only a temporary addition.
3. For the **Mount now?** option, select **Mount** if you want the virtual memory to be added immediately, or **Don't mount** if you just want it to be recorded for future addition at boot time.
4. If you want to add an entire partition as virtual memory, select **Disk** for the **Swap File** option and select the partition from the list. Otherwise, select **Swap File** and enter the path that you want to use as virtual memory. If you enter a path to a file that already exists, that file will be overwritten when the virtual memory is added.
5. Click the **Create** button at the bottom of the page. If you are adding a swap file which does not exist yet, you will be prompted to enter a size for the file, and Webmin will create it for you. If all goes well, the browser will return to the list of filesystems on the main page.

Once the new virtual memory has been added, your system's available memory should increase by the size of the partition or swap file. Use the memory display of the Running Processes module (explained in Chapter 11) to see how much real and virtual memory is available.

5.8 Automounter Filesystems

Before you can access files on any filesystem using Linux, it must first be explicitly mounted. This is fine for hard disks that are mounted at boot time, but is not so convenient for removable media like CD-ROMs, floppy disks, and Zip disks. Having to mount a floppy before you can read or write files on it, and then unmount it when done, is not very user friendly—especially compared to other operating systems like Windows.

Fortunately, there is a solution—the automounter filesystem. This system does not contain any files of its own, but automatically creates temporary directories and mounts filesystems when needed. An automounter filesystem mounted at `/auto` would normally be configured to mount a floppy disk at `/auto/floppy` as soon as a user tries to `cd` into that directory. When the floppy's filesystem is no longer being used, it will be automatically unmounted so that the floppy can be safely ejected.

Automounter filesystems can be created, viewed and edited in Webmin. Each has a configuration file that specifies which devices it will mount and which subdirectories on which they will be mounted. The editing of these configuration files cannot be done within Webmin, however—you can only choose which one to use. Most modern Linux distributions come with an automounter filesystem at `/auto` or `/media` set up by default, and configured to allow access to floppy and CD-ROM drives.

Another common use for the automounter is to provide easy access to NFS servers. Often an automounter on the `/net` directory is set up so that accessing the `/net/hostname` directory will

mount all the exported directories from *hostname* under that directory. This is all done using another automounter configuration file.

5.9 Editing or Removing an Existing Filesystem

After mounting a filesystem, you can go back and change the mount directory, source, and options at any time. Even most filesystems that were set up as part of your operating system's installation process can be edited. Some special filesystem types like **proc** and **devfs**, however, cannot be edited though Webmin because changing them would probably break your system.

The only catch is that filesystems currently in use cannot be immediately edited. If any user or process is accessing any file or is in any directory on a filesystem, it is considered busy and cannot be unmounted and remounted by Webmin in order to change it. Because the **root** filesystem is always in use, making immediate changes to it is impossible. Fortunately, there is an alternative—changing only the permanent record of a filesystem so the new options are applied when your system reboots.

The steps to follow for editing a filesystem are:

1. From the list of filesystems on the main page, click on the mount point directory in the **Mounted as** column. A form containing the current settings will appear, as shown in Figure 5.4.
2. Change any of the settings, including the **Mounted As** directory, the device or server from which the filesystem is mounted, or the mount options.
3. If you want to unmount the filesystem while still keeping it recorded for future mounting, change the **Mount now?** option to **Unmount**. If you want to mount a filesystem that is permanently recorded, however, change the option to **Mount**.
4. Click the **Save** button to make your changes active. If all goes well, the browser will return to the list of filesystems on the main page.

If you are changing a mounted filesystem that is busy, you will be given the option of having your changes applied to the permanent list only. If you are trying to enable quotas on a Linux native filesystem, having the option applied to the permanent list is usually all that is needed.

To totally remove a filesystem, just edit it and set the **Save Mount?** option to **Don't save**, and the **Mount Now?** option to **Unmount**. Assuming it is not in use, it will be unmounted and removed from the list of recorded filesystems and will no longer show up in the list on the module's main page.

5.10 Listing Users of a Filesystem

If you cannot unmount or edit a filesystem because it is busy, you may want to kill the processes that are currently using it.

To find which processes are using a filesystem, follow these steps:

1. From the list of filesystems on the main page, click on the mount point directory in the **Mounted as** column. The form shown in Figure 5.4 will appear.
2. Click the **List Users** button in the bottom-right corner of the page. This will display a list of all processes that are reading, writing, or in any file or directory in the filesystem.

The screenshot shows the 'Create User' dialog box. The 'User Details' section includes fields for Username, Real name, Shell (set to /bin/sh), Home directory, User ID (set to 1011), and Password options (radio buttons for 'No password required', 'No login allowed', 'Normal password', and 'Pre-encrypted password'). The 'Password Options' section includes fields for Password changed, Expiry date (set to 3 Jan 2011), Minimum days, Maximum days, Warning days, and Inactive days. The 'Group Membership' section includes fields for Primary group (radio buttons for 'New group' and 'Existing group') and Secondary groups (a dropdown menu listing adm (4), bin (1),晓得 (2), daemon (3), and diskhouse (17)). The 'Upon Creation' section includes checkboxes for Create home directory? (Yes), Copy files to home directory? (Yes), and Create user in other modules? (Yes). A 'Create' button is at the bottom.

Figure 5.4 Editing an existing filesystem.

3. To kill them, click the **Kill Processes** button at the bottom of the page. You should now be able to return to the Disk and Network Filesystems module and unmount successfully.

5.11 Module Access Control

A Webmin user can be given limited access to this module so he can only edit the settings for certain filesystems or only mount and unmount. Allowing an untrusted user to mount any filesystem is a bad idea because he can gain complete control of your system by mounting an NFS or floppy disk filesystem containing setuid-root programs. Giving someone the rights to only mount and unmount certain filesystems that have their options set to prevent the use of setuid programs, however, is quite safe. This can be useful if your system has a floppy or CD-ROM drive and you are not using an automounter.

Once a user has been given access to the module (as explained in Chapter 52), you can limit him to just mounting or unmounting selected filesystems by following these steps:

1. In the Webmin Users module, click on Disk and Network Filesystems next to the user's name to bring up the access control form.
2. Change the **Can edit module configuration?** field to **No** to stop him from configuring the module to use a different fstab file or mount commands.
3. In the **Filesystems that can be edited** field, select **Under listed directories** and enter a list of mount points into the adjacent text box. For example, you might enter `/mnt/floppy /mnt/cdrom`. It is also possible to enter a directory like `/mnt` to allow access to all filesystems under it.

4. Change the **Can add new filesystems?** field to **No**.
5. Change the **Only allow mounting and unmounting?** field to **Yes**, so that the user cannot actually edit filesystem details.
6. Hit the **Save** button to activate the new restrictions.

On Linux systems, the **Allow users to mount this filesystem?** field can be used to allow the use of the command-line mount and umount programs. Other tools like the Gnome mount panel applet and Usermin also make use of this feature, which may be a better way to give normal users mount and umount privileges.

5.12 Configuring the Disk and Network Filesystems Module

Like other modules, this one has a few options that you can change. To see them, click on the **Module Config** link in the top-left corner of the main page. This will take you to the standard configuration editing page, on which the options shown in Table 5.1 are available under the **Configurable options** header.

Table 5.1 Module Configuration Options

Server to request browse list from	When mounting an smbfs filesystem, the button next to the Server name field will pop up a list of Windows servers on your network. This list is fetched from the server specified by this option, which should be the domain or workgroup master.
Show long filesystem type names	If this option is set to No , the main page will display only short filesystem type names (like EXT2). By default, it is set to Yes and long filesystem types are displayed (like Linux Native Filesystem).

None of the other options on the configuration page should be changed, as they are set automatically by Webmin based on your operating system type.

5.13 A Comparison of Filesystem Types

Unlike other operating systems, Linux supports several different types of filesystems that fully support UNIX file permissions and ownership information. Originally, `ext2` was the only choice, but newer kernel versions and distributions have added support for `ext3`, `reiserfs`, and `xfs`. This list explains the benefits of each of these alternative filesystem types.

New Linux Native Filesystem (`ext3`) Very similar to `ext2`, but with support for journaling. This means that if your system crashes or loses power without having a chance to properly unmount its filesystems, there is no need for the lengthy `fsck` check of the entire `ext3` filesystem that would be needed with `ext2`.

Because `ext3` filesystems are so similar to `ext2`, they are stored on disk in almost exactly the same format. This means that it is relatively simple to convert an existing filesystem to `ext3` by creating a special journal file.

Rieser Filesystem (`reiserfs`) ReiserFS is a totally new filesystem designed to be faster and more efficient than `ext2`. It supports journaling like `ext3` does, and

deals much better with large numbers of small files than other filesystems. It is probably not as mature as `ext3` or `xfs`, however, and does not support quotas.

SGI Filesystem (`xfs`) XFS was originally developed by SGI for its IRIX operating system, and if you are running Webmin on IRIX you can mount `xfs` filesystems as well. It supports journaling and includes native support for ACLs (access control lists) and file attribute lists. The ACL support in particular is very useful, because it allows you to grant access to files in ways that would be impossible with the normal UNIX user/group permissions. XFS has been used for several years on IRIX, so it should be reasonably mature and reliable.

IBM Journaling Filesystem (`jfs`) JFS was originally developed by IBM for use on its AIX and OS/2 operating systems, but has recently been ported to Linux. It supports journaling and large (64-bit) file sizes, but does not currently support quotas or ACLs. Because JFS has been used for years on IBM operating systems, it should be reasonably mature. It is quite new on Linux, however, and so may not be as well tested.

To see which of these filesystem types are supported by your system, go into the Partitions on Local Disks module (covered in Chapter 8) and select an unused partition of type **Linux**. At the bottom of the page will be a form that you can use to create a new filesystem on the partition in one of the types that is available on your system. Most new Linux distributions will support `ext3`, some will support `reiserfs`, but only a few include `xfs` support.

Linux also supports several older filesystem types such as `ext`, `xiafs`, and `minix`. You will never need to use these unless you have an old disk formatted with one of them.

5.14 Other Operating Systems

The Disk and Network Filesystems module supports several other operating systems in addition to Linux, using basically the same user interface. The main differences lie in the filesystem types supported by each operating system, and the type used for hard disk UNIX filesystems. Only Linux, Solaris, and Irix display a drop-down menu of available partitions when adding a hard disk filesystem—on other systems, you must enter the IDE or SCSI controller and drive numbers manually.

The operating systems on which the module can be used, and the major differences between each of them and Linux, are:

Sun Solaris Solaris uses **ufs** (called the Solaris UNIX Filesystem by Webmin) as its standard filesystem type for local hard disks. It has many of the same options as **ext2** on Linux, but does not support group quotas, only user quotas. Adding virtual memory is also supported, in exactly the same way as on Linux.

The NFS filesystem type on Solaris is also similar to Linux, but supports mounting from multiple NFS servers in case one goes down. When entering servers into the **Multiple NFS Servers** field, they must be comma separated like `host1:/path,host2:/path,host3:/path`. Solaris systems can only mount Windows Networking Filesystems if the `rumba` program has been installed. They can only be mounted temporarily, however, not recorded for mounting at boot time.

One interesting filesystem type that only Solaris supports is the RAM Disk (`tmpfs`). Files in a filesystem of this type are not stored on disk anywhere, and so will be lost

when the system is rebooted or the filesystem is unmounted. By default, Solaris uses `tmpfs` for the `/tmp` directory.

FreeBSD FreeBSD also uses `ufs` as its standard local hard disk filesystem type, although it is called the FreeBSD UNIX Filesystem by Webmin. It has most of the same options as Linux, and supports user and group quotas. Virtual memory is also supported on FreeBSD, but with the catch that once added it cannot be removed without rebooting. NFS is supported with similar options to Linux, but Windows networking filesystems are not.

OpenBSD OpenBSD uses the `ffs` filesystem type for local hard disk, which is called the OpenBSD UNIX Filesystem by Webmin. Like FreeBSD, it supports virtual memory and NFS, but not Windows networking filesystems.

HP/UX HP's UNIX variant uses `hfs` (HP UNIX Filesystem) as its standard local hard disk filesystem type, but also supports the superior, journaled `vxf`s, called HP Journaled UNIX Filesystem by Webmin. Both have an option for disk quotas, but for users only. Virtual memory is supported and can be added and removed at any time, but is always mounted at boot if permanently recorded. NFS is also available, with similar options to Linux, but there is no Windows networking filesystem type.

SGI Irix Newer versions of Irix use `xfs` (SGI Filesystem) as their standard hard disk filesystem type, which supports all the same options as `xfs` on Linux, including user quotas, ACLs, and file attributes. The `efs` (Old SGI Filesystem) type is also available, but should only be used if you have old partitions that are already formatted for it, or are running an old version of Irix. Irix supports NFS with similar options to Linux, but does not support Windows networking. AppleTalk and Netware filesystems can also be mounted using command-line tools, but they can not yet be mounted or edited from within Webmin.

The operating system also has standard virtual memory support, but with the peculiarity that the first swap partition on the first hard drive is always added as virtual memory automatically, using the special `/dev/swap` device file.

SCO UNIXWare UNIXWare has very similar filesystem support to Solaris, but also adds support for the hard disk based `vxf`s (Veritas Filesystem) type.

If your operating system is not on the list above, then it is not supported by the Disk and Network Filesystems module. In some cases, this is because the code has not yet been written, such as with AIX or Tru64/OSF1. MacOS X, on the other hand, mounts all hard disk partitions at boot time and automatically mounts network filesystems when requested by the user through the GUI. It therefore has no need for a Webmin module for managing filesystems.

5.15 Summary

Unlike other operating systems, Linux and other variants of UNIX do not automatically make files on hard disks and removable media available to you. This chapter has explained what filesystems and partitions are and how to mount them to gain access to the data that they contain. It has also covered the client-side part of file sharing between two or more systems over a network.

NFS File Sharing

This chapter explains how to export files to other UNIX systems by setting up an NFS server.

6.1 Introduction to File Sharing with NFS

NFS is the most common protocol for sharing files between UNIX systems over a network. NFS servers export directories from their local hard disks to NFS clients, which mount them so that they can be accessed like any other directory. Unlike other file sharing protocols, such as Windows networking, Netware, and AppleShare, NFS was designed to support client systems that have multiple users. This means that a client never logs into a server, and that the server almost completely trusts the client to authenticate users. The down side is that NFS is not a good protocol for sharing files with client systems that are not fully trusted.

Instead of using usernames and passwords for authentication, NFS uses the IP address of the client. Only trusted clients are allowed to mount directories from the server so that it is not vulnerable to unauthorized file access from any client on the network. Some additional security can be gained by restricting the access of particular UNIX users on a client, or treating all requests from a client as a single user.

On Linux, the `/etc/exports` file contains a permanent list of directories exported by NFS and the clients to which they are exported. Typically, this file is read at boot time by the `nfsd` and `mountd` programs that run in the background to service NFS requests. When you change or create exports using Webmin, the `exports` file is directly updated.

This chapter covers only the sharing of directories from a server using NFS. For details on how to mount an NFS exported directory on a client, see Chapter 5. If you want to share files with Windows clients, you should read Chapter 43 (which covers Samba) instead, as NFS support is not widely available for Windows.

6.2 The NFS Exports Module

On Linux, NFS server configuration is done using the NFS Exports module, which can be found under the Networking category. After entering the module, the main page will display a list of exported directories and the clients that are allowed to access them, as shown in Figure 6.1.

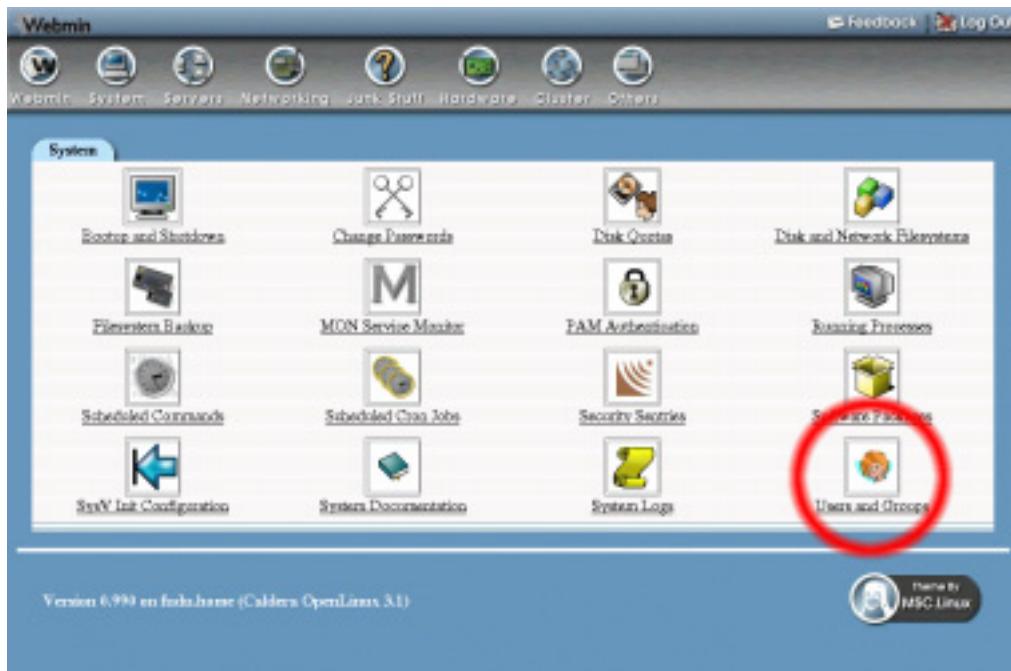


Figure 6.1 The NFS Exports module.

Most Linux distributions come with the programs required for NFS file sharing installed by default. If Webmin detects that they are missing from your system, however, an error message will be displayed when you enter the module. If that happens, you will need to install the `nfs-server` or `nfs` package from your distribution CD or website.

6.3 Exporting a Directory

Only directories on local filesystems can be exported via NFS, so it is not possible to re-export files that have been mounted from another NFS server. It is also not possible to export directories from non-UNIX filesystems such as `vfat`, `ntfs` or `iso9660`. If an exported directory has mount points under it, files under those mount points will not be accessible by NFS clients. So if you exported the root directory `/` and it has a separate filesystem mounted at `/home`, you would need to also export `/home` and clients would need to mount it in order to see the files under it.

The steps for exporting a directory are:

1. Click on the **Add a new export** link on the main page of the module. This will take you to a form for entering the details of the export, as shown in Figure 6.2.

2. Enter the directory that you want to share in the **Directory to export** field.
3. Unless you want the export to be unavailable, make sure the **Active?** option is set to **Yes**.
4. The **Export to** option allows you to choose which clients will have access to the directory. The possible choices are:

Everyone Any system that can connect to yours over the network will be able to mount the directory. Be very careful with this choice, as it may allow anyone on the Internet to access your files.

Host(s) Only the single specified host or IP address will be allowed. You can also enter a wildcard hostname like `*.foo.com` for this option to allow all hosts from a domain. However, if you want to export a directory to several specific client hosts then the only solution is to create multiple exports of the same directory, each with a different hostname in this field.

WebNFS clients WebNFS is a rarely used protocol for accessing NFS exports over the Internet. Don't use this option unless you know what you are doing, as it may allow anyone to access your files.

Netgroup A netgroup is a list of hosts that is defined on an NIS server. Unfortunately, your system must be an NIS client for this to be useful.

Network and Netmask All hosts on the specified network will be allowed to connect. To allow all hosts with IP addresses from `192.168.1.0` to `192.168.1.255`, you would enter `192.168.1.0` for the network, and `255.255.255.0` for the netmask.

5. If you want to prevent clients from modifying or creating files in the exported directory, set the **Access mode** option to **Read only**.
6. If exporting only to trusted systems, set the **Trust remote users** option to **Everyone**.
If you want to ensure that clients only have the permissions of a single UNIX user, however, set **Trust remote users** to **Nobody** and enter the user and his primary group into the **Treat untrusted users as** and **Treat untrusted groups as** fields respectively. This can be very useful if exporting to a client workstation that is used by single user.
7. Click the **Create** button to save the export. If you have made any mistakes in any of the fields, an explanatory error message will be displayed. Otherwise, the browser will return to the list of exports.
8. Click the **Apply Changes** button to make your new export active.

Allowed clients should now be able to mount the exported directory. If not, check your system's error logs for messages from the NFS server processes that explain why the client is being rejected.

6.4 Editing or Deleting an NFS Export

All the details of any existing NFS export can be edited at any time by following these steps:

1. On the main page of the module, click on the client under the **Exported to** column that you want to edit. If a single directory is exported multiple times to different clients, each one must be edited individually.
2. On the export editing form (which is almost identical to Figure 6.2), change any of the options including the directory to share.



The screenshot shows a web-based administration interface titled "Users and Groups". A sub-tab "Local Users" is selected. Below it, a table lists various system and user accounts. The columns are: Username, User ID, Real name, Home directory, and Shell. The table includes entries like "adm", "bin", "root", "daemontools", "exim4", "fengchao", "ftpc", "games", "gdm", "gopher", "halt", "hdb", "homeless", "httpd", "jcameron", "jdesk", "jsmith", "lcameron", "lp", and "mail". Most users have "/home/" as their home directory and "/bin/bash" as their shell.

Local Users				
Create a new user: Create, modify and delete users from batch file				
Username	User ID	Real name	Home directory	Shell
adm	3	adm	/var/adm	
bin	1	bin	/bin	
root	0	DNS Server	/	/bin/false
daemontools	2	daemontools	/sbin	
exim4	3004	Emily Cameron	/etc/exim4	/bin/bash
fengchao	3002	Feng Ching-Chan	/home/fengchao	/bin/bash
ftpc	14	FTP User	/home/ftp	
games	12	games	/usr/games	
gdm	58	GDM user	/	/bin/false
gopher	13	gopher	/usr/lib/gopher-data	
halt	7	halt	/sbin	/bin/halt
hdb	3005	HDB Notes User	/home/hdb	/bin/bash/hdb
homeless	3009	Homeless User	/dev/null	/bin/sh
httpd	55	HTTP Server	/	/bin/false
jcameron	3001	Jamie Cameron	/home/jcameron	/bin/bash
jdesk	3006	Desk User	/home/jdesk	/bin/bash/jdesk
jsmith	3007	John Smith	/home/jsmith	/bin/bash
lcameron	3021	Lara Cameron	/home/lara	/bin/bash
lp	4	lp	/var/spool/lpd	
mail	8	mail	/var/spool/mail	

Figure 6.2 The new NFS export form.

3. If you want to delete the export, click the **Delete** button at the bottom right of the page. Otherwise, click **Save** to save your changes. Either way, your browser will return to the module's main page.
4. Click the **Apply Changes** button to make the changes active.

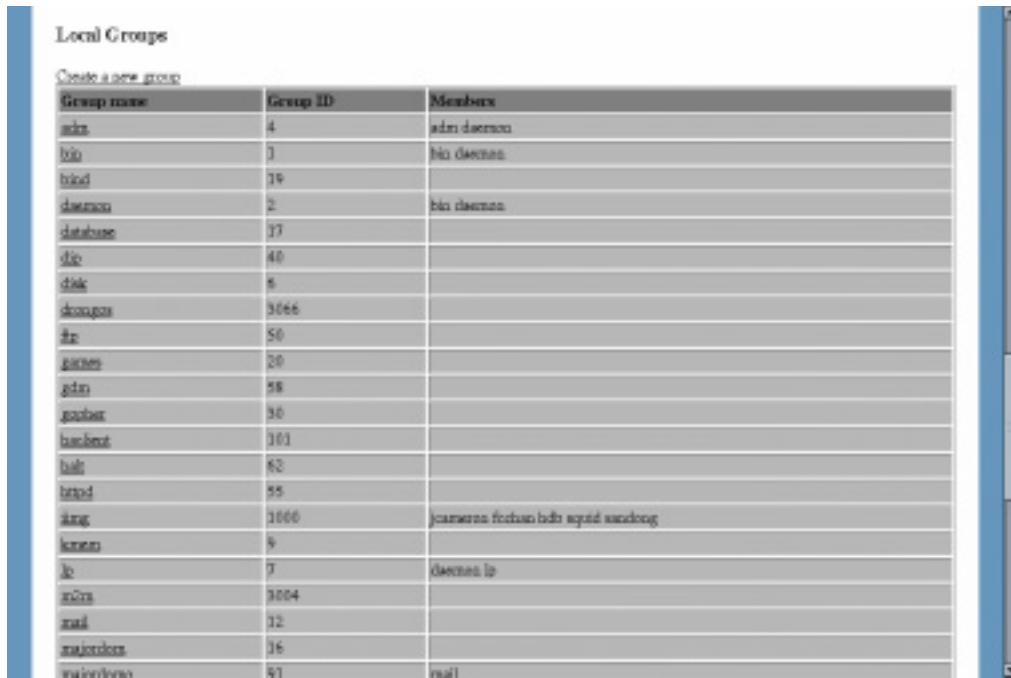
6.5 NFS on Solaris

On Solaris, NFS exports are managed by the separate NFS Shares module. Because Solaris uses a different file (`/etc/dfs/dfstab`) and file format for storing exports, the module's user interface is different to that of the Linux module. Figure 6.3 shows the main page of the NFS Shares module. As you can see, exports are configured by directory instead of by client.

To add a new NFS export on Solaris, follow these steps:

1. Click on the **Start sharing a new directory** link, which will take you to a form for entering the details of the new export.
2. Enter the directory that you want to share in the **Directory** field.
3. Fill in the **Read-only access** and **Read-write access** fields with the hostnames of clients to which you want to grant access. As the names suggest, a host in the **Read-only** field will not be able to write to or modify files on the server.

In addition to hostnames, you can also enter networks using the format `@192.168.1` or `@192.168.1/24`, NIS netgroups or even DNS domains like `.foo.com` (the leading dot indicates an entire domain).



The screenshot shows a web-based interface for managing local groups. At the top, there's a header 'Local Groups' and a sub-header 'Create a new group'. Below is a table with columns: Group name, Group ID, and Members. The table lists various system groups like adm, bin, daemon, database, dip, disk, docgroup, ftp, games, gdm, gather, bucket, halt, hload, ping, known, lp, nroff, mail, mailerlock, and null. The 'Members' column for the 'ping' group contains the text 'jcameron fenton hdb squid sasdcng'.

Local Groups		
Create a new group		
Group name	Group ID	Members
adm	4	adm daemon
bin	3	bin daemon
daemon	2	bin daemon
database	17	
dip	40	
disk	5	
docgroup	3066	
ftp	50	
games	20	
gdm	58	
gather	30	
bucket	101	
halt	62	
hload	55	
ping	1000	jcameron fenton hdb squid sasdcng
known	9	
lp	7	daemon lp
nroff	3004	
mail	32	
mailerlock	36	
null	41	null

Figure 6.3 The Solaris NFS Shares module.

4. By default, the `root` user on clients will have only limited access to files on the server. To give `root` on some clients' full file access privileges, enter their hostnames, networks, netgroups, or domains into the **Root access** field.
5. Click the **Save** button at the bottom of the page to create the export. Unless you have made a mistake on the form, you will be returned to the list of exported directories.
6. Click the **Apply Changes** button to make your new export active.

Existing NFS exports can be edited by simply clicking on a directory on the main page of the module. The same form used for creating an export will appear, allowing you to change any of the options. If you want to delete the export, click the **Delete** button at the bottom of the page. Be sure to click **Apply Changes** again after making any changes so that they will become active.

6.6 NFS on BSD, MacOS X, and OpenServer

FreeBSD, NetBSD, OpenBSD, OS X, and OpenServer all use the `/etc/exports` file for storing NFS exports, but its format is different in different operating systems. This means that they use a different NFS Exports module that has its own unique user interface, as shown in Figure 6.4. Exports are configured by directory instead of by client, and you can specify options and allowed clients for multiple directories at once.

To add a new NFS export on one of these operating systems, follow these steps:

1. Click on the **Add a new export** link on the main page of the module. A form for entering the details of the new NFS export will appear.

Figure 6.4 The BSD NFS Exports module.

2. Enter the directories that you want to share into the **Directories to export** field. Be aware that multiple directories on the same filesystem cannot be exported to the same client separately.
3. If you want to allow clients to mount subdirectories as well, select the **Export subdirectories?** option. If this is enabled, however, only one directory can be entered in the **Directories to export** field and it must be the root of a filesystem.
4. To give all clients read-only access, set the **Read only?** option to **Yes**.
5. To limit access to a single host or list of hosts, select **Hosts / netgroups** for the **Clients** option and enter the hostnames, IP addresses, or netgroups that you wish into the field. To limit access to an entire network, select the **Network** option and entire network address (like *192.168.1.0*) and netmask (like *255.255.255.0*) into the respective fields.
6. Click the **Save** button to create the export and you will be returned to the list of exports on the main page.
7. Click the **Apply Changes** button to make your new export active.

Existing NFS exports can be edited by simply clicking on a directory on the main page of the module. The same form as is used for creating an export will appear, allowing you to change any of the options, or click the **Delete** button to get rid of it. Be sure to click **Apply Changes** again after making any changes so that they will become active.

6.7 NFS on Irix

Irix has its own unique format for the `/etc(exports` file that is similar to the BSDs, but not quite the same. It therefore also has its own special version of the NFS Exports module with a slightly different user interface. The main page of the module lists the directories being exported and the hosts they are exported to, in a very similar layout to the BSD NFS Exports module shown in Figure 6.4.

To add a new NFS export on Irix, follow these steps:

1. Click on the **Add a new NFS** export link on the main page, which will take you to a form for entering the new export's details.
2. Enter a directory into the **Directory to export** field.
3. Enter the hostnames, IP addresses, and netgroups of clients that you want to grant access to into the **Export to hosts/netgroups** field. If this field is left empty, any host will be allowed to mount the exported directory.
4. To prevent all clients from modifying exported files, set the **Read-only?** option to **Yes**.
5. If you want to give read/write access to some clients and read-only access to others, enter the hostnames or IP addresses of the read/write clients into the **Read/write access** field.
6. By default, the `root` user on clients will have only limited access to files on the server. To give `root` on some clients full file access privileges, enter their hostnames or IP addresses into the **Root file access** field.
7. Click the **Save** button to create the export and you will be returned to the list of exports on the main page, as long as there are no errors in the form.
8. Click the **Apply Changes** button to make your new export active.

Existing NFS exports can be edited or deleted by clicking on their directory on the module's main page. If you make any changes, you must click the **Apply Changes** button to make them active.

6.8 Summary

In this chapter the NFS file sharing protocol has been explained, and the steps to take to share files from one system to others have been documented. You should now know how to export data from a system running any of the supported UNIX variants to any client that can mount NFS filesystems. You should also understand the security implications of sharing files with NFS, and know that it should not generally be used to share files with untrusted clients.

Disk Quotas

In this chapter, the use of disk quotas to limit the amount of space that individual users can consume is explained.

7.1 Introduction to Disk Quotas

On a system with multiple users, it is often necessary to limit how much disk space each user can take up. Quotas are the mechanism used by UNIX systems to enforce limits on the amount of disk space and the number of files each user (and possibly group) can own. Each file counts towards the quota of the user who owns it, and if group quotas are being used the file counts towards the quotas of its group owner as well. Once a user exceeds his quota, he will not be able to create or enlarge any files until some are deleted.

Quotas are set up on a per-filesystem basis, so that you can have different quotas for different directories on your system. This means, however, that if two directories are both on the same filesystem then they must share the same quotas. Only UNIX filesystems like `ext2`, `ext3`, and `xfs` on local hard disks support quotas—although if your system NFS mounts a remote directory that has quotas enabled, they will be enforced on the server.

Each user or group has two different quotas, one for blocks and one for files. The blocks quota controls how much disk space the user can use and is specified in disk blocks that are typically 1 kB in size. The files quota controls how many separate files the user can create, and is necessary because UNIX filesystems often have a limit on how many files can exist at one time. Without a files quota, a user could create millions of empty files until the filesystems limit was reached and so prevent other users from creating any files at all.

Both the blocks and files quotas have what are called soft and hard limits. The soft limit is the point at which the user is warned that he is close to exceeding his quota, but is still allowed to continue using up disk space. The hard limit is the number of blocks or files that can never be exceeded, and any attempt to do so will result in an error. Both limits are optional, so that you can

have only a hard limit and give the user no warning that he is approaching his quota, or only a soft limit and only warn users of quota violations instead of actually enforcing them.

If a user stays above his soft limit but below the hard limit for more than a set period of time (called the grace period), the system will treat him as though he has exceeded the hard limit and prevent the creation or enlargement of any files. Only when the user deletes enough files to drop his usage below the soft limit will it revert to just a warning level.

At the shell prompt, quotas can be viewed using the repquota and quota commands, and edited using the edquota command. The files aquota.user and aquota.group in the mount directory of each filesystem contain the actual records of how much disk space is allocated to each user or group and how much they are currently using. When displaying and setting quotas, Webmin calls the quota commands and parses their output. It does not use system calls or attempt to edit the quota files directly.

7.2 The Disk Quotas Module

Webmin's Disk Quotas module is found under the System category. When you enter the module, a list of all filesystems on which quotas could be or are active is displayed, along with their current active status and whether quotas are configured for users, groups, or both. See Figure 7.1 for an example.

On most systems that have never used quotas before, none of your filesystems will be listed. This is because quotas must first be enabled in the Disk and Network Filesystems module, as explained in Chapter 5.

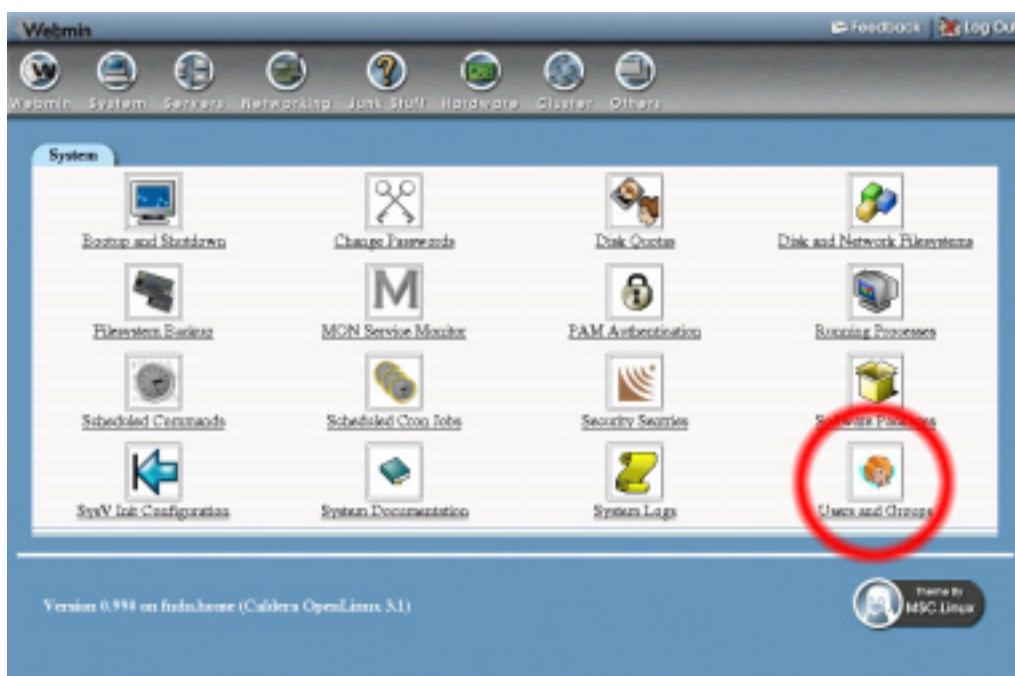


Figure 7.1 The Disk Quotas module.

If your system does not have the quota manipulation commands installed, Webmin will display an error message on the main page of the module and you will not be able to activate or edit any quotas. All Linux distributions should have a package on their CD or website containing the quota commands.

7.3 Enabling Quotas for a Filesystem

If the main page of the module shows **User Quotas Active** (or **Group Quotas Active**) under the **Status** column for the filesystem, then quotas have already been enabled. If not, to configure and turn on quotas for an `ext2` or `ext3` filesystem, follow these steps:

1. If the filesystem already appears in the list on the main page of the module, quotas have already been configured and you can skip to Step 5.
2. Go to the Disk and Network Filesystems module and click on the filesystem on which you want to enable quotas.
3. Change the **Use Quotas?** option to either **User only**, **Group only**, or **User and Group** depending on which kinds of quota you want to enforce.
4. Click the **Save** button. If an error appears saying that the filesystem is already in use, just click the **Apply to Permanent List** button. Quotas can usually be enabled without needing to reboot, and will be automatically reenabled when the system is next rebooted. However, if the next step fails you will need to reboot your system to activate them—this is necessary on some newer versions of Linux.
5. Back in the Disk Quotas module, your filesystem should now be visible. Click on the **Enable Quotas** link to activate quotas now.
6. Assuming all goes well, the browser will return to the list of quotas after a short delay and the **Status** column will change to **User Quotas Active**.

For an `xfs` filesystem, the procedure is slightly different. You must first enable user and/or group quotas in the Disk and Network Filesystems module, and then either reboot or unmount and remount the filesystem. Quotas will be automatically activated at mount time, so there is no need to enable them in the Disk Quotas module.

7.4 Disabling Quotas for a Filesystem

To permanently deactivate quotas for an `ext2` or `ext3` filesystem, follow these steps:

1. On the main page of the module, click on **Disable Quotas** under the **Action** column for the filesystem.
2. To prevent quotas from being reactivated at boot time, go to the Disk and Network Filesystems module and click on the filesystem from the list.
3. Change the **Use Quotas?** option to **No**.
4. Click the **Save** button. If an error saying that the filesystem is already in use appears, just click the **Apply to Permanent List** button.

For an `xfs` filesystem, Step 1 is not necessary (or possible) as quotas are only enabled when the filesystem is mounted. In Step 4, however, when saving the quota settings for the filesystem, it must be unmounted and remounted cleanly for the deactivation to take effect.

7.5 Setting Quotas for a User or Group

The quotas for a user or group can be set or changed at any time on a filesystem that currently has quotas enabled of the correct type. By default, any user or group whose quotas have not yet been set will have no limits at all and thus be able to use up all the disk space on the filesystem.

To set quotas for a user, follow these steps:

1. From the list of filesystems on the main page of the module, click on the mount point of one on which you want to edit quotas. This will take you to a page listing the quotas for all users on the filesystem, as shown in Figure 7.2.
2. Click on the name of the user you want to edit under the User column, or enter the username into the **Edit Quota For** field and press the button. Both will take you to a form containing the user's current quota settings and blocks and files used, as shown in Figure 7.3.
3. Set the **Soft Block Limit** and **Hard Block Limit** fields to the number of blocks to which you want to limit the user, or select **Unlimited** to not impose any limit. On most filesystems, each block will be 1 kB in size, but this not necessarily always the case.
4. Set the **Soft File Limit** and **Hard File Limit** fields to the number of files that you want to limit the user to owning.
5. Click the **Update** button. The new quota settings will take effect immediately.

The procedure for setting group quotas is almost identical. If a filesystem has both user and group quotas enabled, the main page of the module will have two links for each filesystem—one for users, and one for groups.

7.6 Copying Quotas to Multiple Users

If you have a large number of users on your system and want them to all have the same quotas, there is an easier solution than setting each user individually. Instead, you can set the quotas that you want for one user and duplicate his settings to as many other users as you want. The only down side is that quotas are copied on all filesystems, not just a single one.

The steps for copying quotas like this are:

1. Set the quotas for a single *source* user, as explained in Section 7.5 “Setting Quotas for a User or Group”.
2. On the main page of the module, enter the username of the *source* user into the **Edit User Quotas** page and press the button.
3. On the page that appears listing the user's quotas on all filesystems, click the **Copy Quotas** button. This will take you to a form for choosing to which users the quota settings will be copied.
4. Choose which *target* users to copy quotas to by selecting one of the options on the form:
 - All users on your system** Every single user on your system will have the same quota settings. You may want to set quotas for `root` back to unlimited after doing this.
 - Selected users** Only the users entered into the field next to this option will have their quotas set.
 - Members of selected groups** All primary and secondary members of the groups entered into the field next to this option will have their quotas set.

The screenshot shows a table titled "Local Users" with the following data:

Username	User ID	Real name	Home directory	Shell
adm	3	adm	/root/adm	/bin/false
bin	1	bin	/root/bin	/bin/false
daemon	2	daemon	/root/daemon	/bin/false
etc	3004	Emily Cunwoo	/home/etc	/bin/false
ftchao	3002	Foong Ching Chan	/home/ftchao	/bin/bash
ftp	14	FTP User	/home/ftp	/bin/false
games	12	games	/usr/games	/bin/false
gdm	58	GDM user	/	/bin/false
goopher	13	goopher	/usr/lib/goopher-data	/bin/false
halt	7	halt	/root	/sbin/halt
hald	3005	HDB Notes User	/home/hald	/bin/false
homeless	3009	Homeless User	/dev/null	/bin/false
httpd	55	HTTP Server	/	/bin/false
jcameron	3001	Jamie Cameron	/home/jcameron	/bin/bash
jdesk	3006	Ideak User	/home/jdesk	/bin/false
john.smith	3007	John Smith	/home/john.smith	/bin/false
lara	3021	Lara Cameron	/home/lara	/bin/false
lp	4	lp	/usr/sbin/lpd	/bin/false
mail	8	mail	/usr/local/mail	/bin/false

Figure 7.2 The list of users and their quotas.

- Click the **Copy** button to copy the quotas for the *source* user on all filesystems to all *target* users.

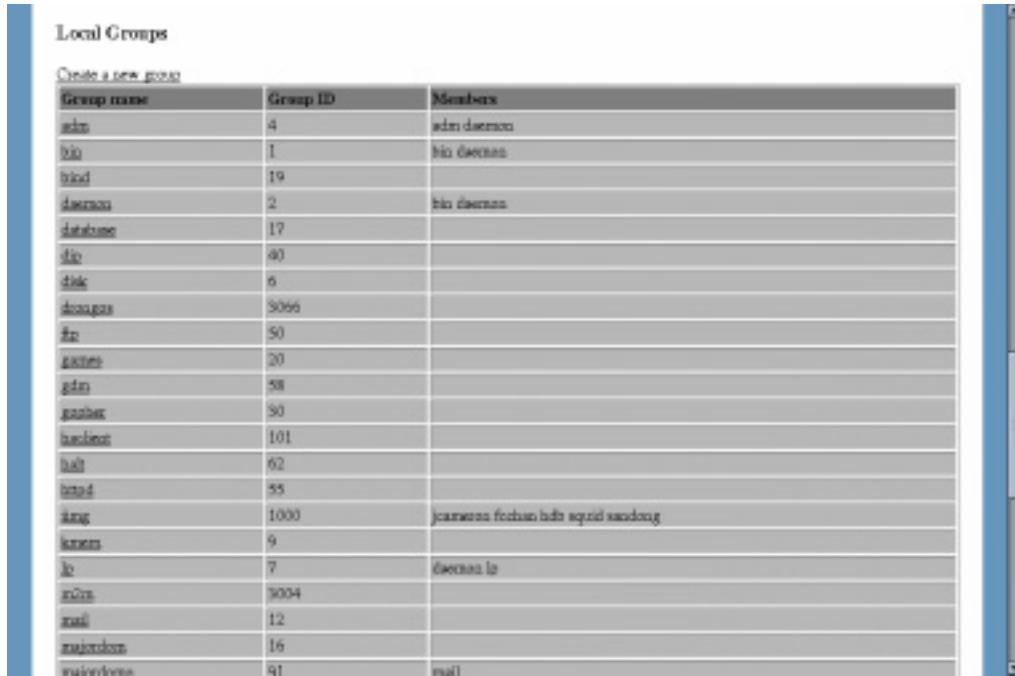
If you are using group quotas, it is also possible to copy the settings for one group to multiple other groups. The options for choosing which groups to copy to, however, are slightly different. The **Selected users** option is replaced with **Selected groups**, and the **Members of selected groups** option is replaced with **Groups containing users**. The latter option will copy to all groups that have one of the entered users as a member.

7.7 Setting Grace Times

When a user exceeds his soft blocks or files limit, he will still be able to use up disk space up to the hard limit for a certain period of time—the grace period. There are separate periods for the blocks quota and the files quota on each filesystem. Once the period has expired, it will be as though he had reached the hard limit. No more blocks of disk space can be used if it was the blocks quota that was exceeded, or no more files can be created if it was the files quota. Grace periods can also be set for group quotas, and if a filesystem has both user and group quotas enabled, each has their own separate periods.

To set the grace periods for all users on a particular filesystem, follow these steps:

- Click on the mount point from the list of filesystems on the main page of the module. This will take you to the list of all users and their quotas, as shown in Figure 7.2.
- Click the **Edit Grace Times** button, which will bring up a form for editing the periods.



The screenshot shows a table titled "Local Groups" with a header "Create a new group". The table has three columns: "Group name", "Group ID", and "Members". The data rows are as follows:

Group name	Group ID	Members
adm	4	adm daemon
bin	1	bin daemon
bind	19	
daemons	2	bin daemon
database	17	
dip	40	
disk	6	
diskgas	3066	
ftp	50	
games	20	
gin	58	
gopher	90	
mail	101	
mail	62	
mail	55	
mail	1000	jcameron fochan hahn squid xandong
news	9	
lp	7	daemon lp
nntp	3004	
null	12	
network	16	
network	91	null

Figure 7.3 The user quota form.

3. For both the blocks and files quotas, select the period and units. When done, click the **Update** button to save your settings and put the grace periods into immediate effect.

The process for editing the group grace times on a filesystem is almost exactly the same. If a filesystem has both user and group quotas enabled, the main page of the module will have two links for each filesystem—one for users and one for groups.

7.8 Setting Default Quotas for New Users

If a filesystem has user quotas enabled, you can configure the blocks and files quotas that will be assigned to new UNIX users created using Webmin's Users and Groups module. As explained in Chapter 4, any time a user is added other modules will be notified so that they can perform additional actions. In the case of the Disk Quotas module, that action can be the setting of an initial quota for the user on multiple filesystems.

To set the default quota for new users on a particular filesystem, follow these steps:

1. On the module's main page, click on the mount point of the filesystem for which you want to set the default. This will take you to the list of users and their quotas, as shown in Figure 7.2.
2. At the very bottom of the page is a form in which you can set the default hard and soft blocks and files quotas. When you are done filling it in, click the **Apply** button.

The same options can be set for UNIX groups created in the Users and Groups module on the group quotas page.

7.9 Other Operating Systems

As disk quotas work in a very similar way across all versions of UNIX, this module appears almost identical on all supported operating systems. The biggest difference is that some UNIX variants do not support group quotas. Some (like Solaris) do not need quotas to be enabled in the Disk and Network Filesystems module before activating them in this module. If there is a quotas option for the filesystem, it determines whether they are enabled at boot time or not.

7.10 Configuring the Disk Quotas Module

The Disk Quotas module has only a few options that can be changed to configure its user interface. To edit them, click on the **Module Config** link on the main page, which will take you to the standard configuration editing page. The available settings under the **Configurable options** header are displayed in Table 7.1.

Table 7.1 Module Configuration Options

Maximum number of users or groups to display	When you click on a filesystem on the main page of the module, a full list of users or groups with quotas will be displayed. However, if the number of users exceeds this option, a text box for entering a username to view and set quotas for will be displayed instead.
Sort users and groups by	Normally the list of users with quotas on a filesystem is ordered by disk usage, but by changing this option you can have them ordered by username or just displayed in the order that the <code>repquota</code> command uses.

None of the other options on the configuration page should be changed, as they are set automatically by Webmin based on your operating system type.

7.11 Module Access Control

As described in Chapter 52, it is possible to give a Webmin user access to only part of the functionality of a module. In the case of the Disk Quotas module, you can limit which users and groups quotas can be edited, and on which filesystems they can be edited. This can be useful if there is a person in your organization who is allowed to edit some or all quotas, but not perform any other administrative tasks.

Assuming you have already created a user with access to the module, the steps to follow to set this up are:

1. In the Webmin Users module, click on **Disk Quotas** next to the name of the user that you want to restrict.
2. Set the **Can edit module configuration?** field to **No**, so that the user cannot change the commands used for setting and getting quotas.

3. To restrict the filesystems on which quotas can be assigned, change the **Filesystems this user can edit** field to **Selected** and choose them from the list below.
4. Set the **Can enable and disable quotas?** field to **No**, unless the user is responsible for all user and group quotas on the allowed filesystems—otherwise he will be able to turn off quotas for users that he is not allowed to edit.
5. Change the **Can configure quotas for new users?** field to **No**, so that he cannot change the quotas that are assigned to users created in the Users and Groups module. Only if the Webmin user is allowed to edit all quotas on a filesystem should this be left set to **Yes**.
6. If you do not want this Webmin user to change grace times, set the **Can edit user grace times?** and **Can edit group grace times?** fields to **No**.
7. To stop the user from handing out massive disk quotas, set the **Maximum grantable block quota** and **Maximum grantable file quota** fields to the maximum blocks and files that can be granted to any one user, respectively. There is nothing, however, to stop him granting quotas to multiple users that add up to more than these limits.
8. To restrict the UNIX users whose quotas can be edited, change the **Users this user can edit quotas for** field from **All users** to one of the other options. The most useful is **Users with UID in range**, which restricts access to those users whose UIDs lie within the minimum and maximum numbers entered into the fields next to it. It is usually a bad idea to allow the editing of the `root` user's quotas, as setting it too low may prevent the system from creating important PID, mail, and lock files. You can prevent this by selecting **All except users** and entering `root` into the field next to it, assuming that you want to allow the editing of every other user. To stop the Webmin user editing any user quotas at all, select the **Only users** option and enter nothing into the field next to it.
9. Similarly, you can limit the groups whose quotas can be edited by changing the **Groups this user can edit quotas for** field. Naturally, this only has an effect on filesystems that have group quotas enabled.
10. When done, click the **Save** button to have the restrictions applied immediately.

7.12 Summary

After reading this chapter you should understand what disk quotas are useful for, and how to enable them on any UNIX system. You should also know the difference between block and file quotas, the connection between quotas and filesystems, and the effects of this on a server with multiple filesystems containing user data. Finally, the differences between and effects of user and group quotas should be clear.

Partitions, RAID, and LVM

This chapter explains how hard disks are partitioned and how filesystems are created on those partitions. It also covers the use of RAID and LVM to combine multiple partitions into one large filesystem.

8.1 Introduction to Hard Disk Partitions

All hard disks used by Linux and other operating systems on PC hardware are divided into one or more non-overlapping regions called partitions. Sometimes an entire hard disk will be taken up by one partition, but usually your system will have at least two partitions on the primary disk—one for the root filesystem, and one for virtual memory (also known as swap space). As explained in Chapter 5, each partition can be used for either a single filesystem or for virtual memory.

Every partition has a type which identifies the kind of data that it stores. There is a type for Linux filesystems, a type for Linux swap space, a type for Windows filesystems, and many more. Almost every kind of operating system that runs on PC hardware has its own partition type for its own filesystems. When adding new partitions on your system, however, you will very rarely use any types other than those specifically for Linux.

On PC systems, each hard disk can only contain four **primary** partitions. Because this is often not enough, it is possible for one of those four to be a special **extended** partition that can contain an unlimited number of **logical** partitions. If you make use of an extended partition, there is effectively no limit on the number that your hard disk can contain.

Every hard disk is divided into equal-sized cylinders, which represent concentric circles on the surface of the disk. Larger hard disks generally have more cylinders, but due to different drive geometries this is not always the case. Each partition has a starting and ending cylinder and occupies all the space on the disk between them.

Be very careful when changing or reformatting any existing partitions on your system. Because they contain filesystem data, deleting or modifying one could wipe out all your files or make your system unbootable. Webmin tries to prevent this, but it is still possible to do a lot of damage with only a few mouse clicks! Normally you should only need to create or edit partitions when adding a new hard disk to your system.

8.2 The Partitions on Local Disks Module

All disk partition management in Webmin is done using the Partitions on Local Disks module, which can be found under the Hardware category. When you enter the module, a page showing all hard disks and partitions found on your system will be displayed, as shown in Figure 8.1.

All IDE and SCSI disks are shown, along with their manufacturers and model numbers. If your system has a hardware RAID controller that is supported by the module, the RAID devices will be shown instead of the actual underlying hard disks that make them up. Disks and partitions used for software RAID will be shown, but not the logical or virtual drives that they have been combined into.

For each disk, all partitions on it will be listed showing their type, start and end cylinders and current mount point, or other use. If the partition contains a filesystem, the amount of free disk space will be displayed as well. If a partition is being used for software RAID, the raid device that it is part of will be shown. Similarly, if a partition is part of an LVM volume group, the group name will be displayed under the **Use** column.

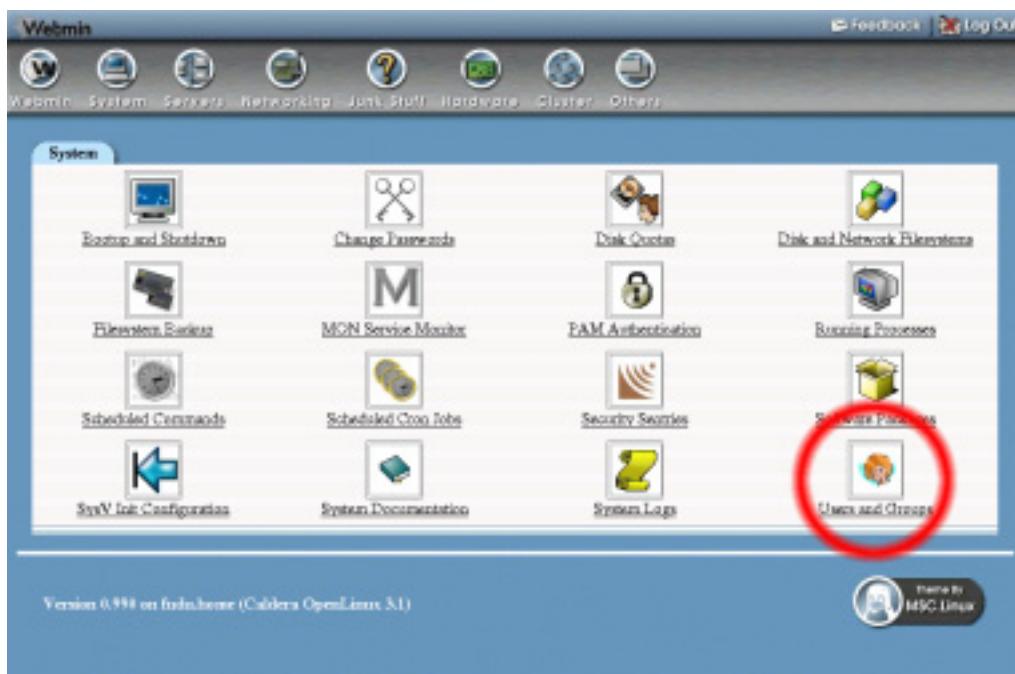


Figure 8.1 The Partitions on Local Disks module.

8.3 Adding and Formatting a New Partition

If you have just added a new hard disk to your system and want to make use of it with Linux, you must first partition it and then format the partition as the filesystem type of your choice. The steps for this process are:

1. Locate your new hard disk in the main page of the Partitions on Local Disks module. It will probably not have any partitions on it, but it may have been set up with one large partition by the manufacturer.
2. Assuming no partitions exist yet, click the **Add primary partition** link next to your new hard disk. This will take you to the creation form shown in Figure 8.2 for entering the details of the new partition.
3. If the new partition takes up the entire hard disk, the **Extent** fields can be left unchanged as they are always automatically filled in to cover all the free space left on the disk. If you want to create more than one partition, however, adjust the extent so that it takes up only part of the disk.
4. If this partition is for an `ext2`, `ext3`, `reiserfs`, or `xfs` filesystem, set the **Type** field to **Linux**.
If it is to be for virtual memory, set the **Type** to **Linux swap**.
If it is for software RAID, set the **Type** to **Linux raid**.
If it is for LVM, set the **Type** to **Linux LVM**.
If you are creating the filesystem for some other operating system to use, set the **Type** field to whatever is appropriate for that OS.
5. Click the **Create** button to add the partition. Assuming no errors were detected, you will be returned to the list of disks and partitions on the main page of the module, which should now include the new partition.
6. If the new partition is to have a Linux filesystem created on it, you must follow the steps in Section 8.4 “Creating a New Filesystem”. Virtual memory partitions can be added immediately in the Disk and Network Filesystems module. Partitions for use with RAID can be also be used immediately in the Linux RAID module but you must have created all the partitions that will make up a RAID device before creating it. Partitions that will be part of an LVM volume group can be added immediately using the Logical Volume Management module.

8.4 Creating a New Filesystem

Before a newly created partition can be used to store files, it must first have a filesystem created on it. Filesystems can also be created on partitions that have been used before, perhaps by another operating system. However, be very careful when formatting a partition with a new filesystem, as any files that it used to contain will be lost forever.

The steps for creating a new filesystem are:

1. On the main page of the module, click on the number of the partition that you want to reformat. This will take you to the partition editing form, as shown in Figure 8.3.
2. Near the bottom of the page is a button labeled **Create Filesystem** with a menu of supported filesystem types next to it. See Section 5.13 “A Comparison of Filesystem Types”



The screenshot shows a software interface titled "Users and Groups". Below it, a section titled "Local Users" contains a table with the following data:

Username	User ID	Real name	Home directory	Shell
adm	3	adm	/root/adm	/bin/false
bin	1	bin	/root/bin	/bin/false
daemon	2	daemon	/root/daemon	/bin/false
empty	3004	Emily Cunwoo	/home/empty	/bin/false
foochan	3002	Foong Ching Chan	/home/foochan	/bin/bash
ftp	14	FTP User	/home/ftp	/bin/false
games	12	games	/usr/games	
gdm	58	GDM user	/	/bin/false
goopher	13	goopher	/usr/lib/goopher-data	
halt	?	halt	/sbin	/bin/false
hald	3005	HDB Daemon User	/home/hald	/usr/bin/hald
homeless	3009	Homeless User	/dev/null	/bin/false
httpd	55	HTTP Server	/	/bin/false
jcameron	3001	Jamie Cameron	/home/jcameron	/bin/bash
ideak	3006	Ideak User	/home/ideak	/usr/bin/ideak
johnsmith	3007	John Smith	/home/john.smith	/bin/false
lara	3021	Lara Cameron	/home/lara	/bin/false
lp	4	lp	/usr/sbin/lpd	
mail	3	mail	/usr/local/mail	

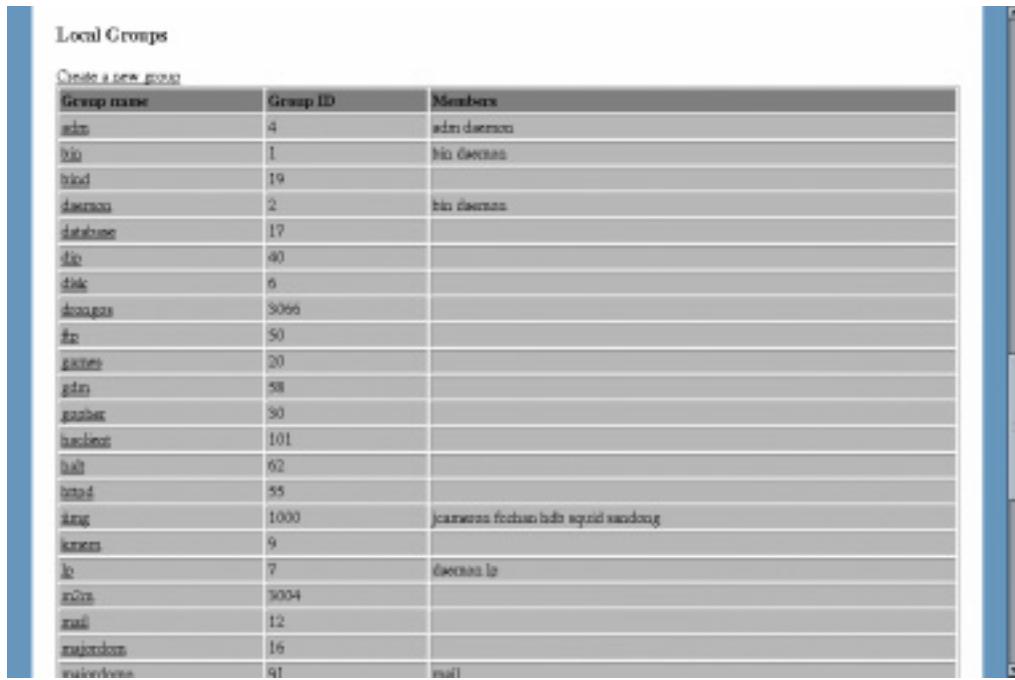
Figure 8.2 The partition creation form.

for information on the pluses and minuses of each type. When you have made a selection, click the button and it will take you to a form for selecting options for the new filesystem.

3. Depending on the type of filesystem chosen, different creation options are available. For `ext2` or `ext3` filesystems, the only one that you might want to change is **Reserved blocks**, which determines the amount of disk space reserved for the exclusive use of the root user. The default is 5 percent, which I think is rather wasteful.
4. Click the **Create Filesystem** button to format the partition. A page showing the progress of the new filesystem's creation will be displayed, which can take some time for large hard disks.
5. Assuming that the formatting is successful, you can now use the Disk and Network Filesystems module to mount the new filesystem.

8.5 Partition Labels

Labels are a feature of newer versions of Linux that allow a partition to be identified in the `/etc/fstab` file by a short name rather than its IDE or SCSI device file, such as `/dev/hdb3`. Device files can change if you change an IDE drive from one controller to another, change the ID of a SCSI drive, or even add a new SCSI drive with an ID lower than an existing drive. Any of these changes could cause a partition to fail to mount at boot time—possibly making your system unbootable. Partitions with labels can be referred to by label name, however, which does not change even if the device file does.



The screenshot shows a table titled "Local Groups" with a header "Create a new group". The table has three columns: "Group name", "Group ID", and "Members". The data is as follows:

Group name	Group ID	Members
adm	4	adm daemon
bin	1	bin daemon
bind	19	
daemon	2	bin daemon
database	17	
dip	40	
disk	6	
dousgas	3066	
ftp	50	
games	20	
gin	58	
gopher	90	
hacinet	101	
halt	62	
inetd	55	
icing	1000	jcameron fochan hala squid xandong
irc	9	
lp	7	daemon lp
mail	3004	
mailnull	12	
mailxmas	16	
mailxmas	91	mail

Figure 8.3 The partition editing form.

Some newer Linux distributions use labels by default for filesystems that you create at install time. If you use the Disk and Network Filesystems module on such a system, the **Location** column for these filesystems will be something like **Partition labeled /home**.

Only partitions with ext 2, ext 3 or xfs filesystems on them can be labeled, as the label is stored in the filesystem rather than the partition table. To label an existing filesystem, follow these steps:

1. On the main page of the module, click on the number of the partition that you want to label. This will take you to the partition editing form, as shown in Figure 8.3.
2. Assuming the partition is not currently in use, you will be able to enter the new label into the **Partition label** field. It must be at most 16 characters long—for example */home* or *root*.
3. After you have entered the label, click the **Save** button. It will be stored in the filesystem, and the browser will return to the module's main page.
4. At this point, the Disk and Network Filesystems module can be used to mount the labeled filesystem by label name, as explained in Chapter 5.

8.6 Deleting or Changing a Partition

Once a partition has been created, its size or position on the hard disk cannot be changed using Webmin. The only things you can do are change its type or delete it. Neither are possible, however, if a filesystem on the partition is listed in the Disk and Network Filesystems module—that is, if it is currently mounted or recorded for mounting at boot time.

Changing the type of a partition will not harm the data on it in any way. It may, however, make it unusable by some operating systems or for some purposes.

The steps to change its type are:

1. On the main page of the module, click on the number of the partition that you want to change. This will take you to the partition editing form.
2. As long as the partition is not in use, you will be able to select a new type from the **Type** field and click the **Save** button.
3. Once the change has been made, the browser will return to the list of disks and partitions.

Deleting a partition should be done only if you are sure that you want to lose all the data on it. It is the only way to make some changes to the partition table in Webmin, however, such as replacing two small partitions with one larger one. If you are sure that you want to go ahead with the deletion, use the following process:

1. On the main page of the module, click on the number of the partition that you want to delete, which will take you to the partition editing form.
2. Click the **Delete** button, which will only appear if the partition is not in use. This will take you to a page for confirming the deletion.
3. If you are really sure you want to go ahead, click the **Delete Now** button. Once the job is done, you will be returned to the main page of the module.

It is theoretically possible to restore a deleted partition by creating a new one with the exact same size and extents.

8.7 Module Access Control

Surprisingly, it is possible to limit the access that a Webmin user has to certain disks in the Partitions on Local Disks module. This could be useful if your system has a removable drive (like a Zip or Jaz drive) that you want users to be allowed to partition with Webmin, while preventing them from reformatting the primary hard disk.

Once a user has been granted access to the module, the process to restrict the disks that he can access includes the following steps:

1. In the Webmin Users module, click on Partitions on Local Disks next to his username. This will bring up the module access control form.
2. Change the **Disk this user can partition and format** field to **Selected**, and choose the disks that the user should be allowed to partition and create filesystems on from the list below.
3. To stop the user seeing disks on the main page that he cannot manage, change the **Can view non-editable disks?** option to **No**.
4. Finally, click the **Save** button to activate the access control restrictions.

Just being able to partition and format a disk is not particularly useful, unless it can be mounted as well. The Disk and Network Filesystems module has no support for access control restrictions, because giving a user the rights to mount a filesystem would open up several security

holes. A better solution is to set up an automounter filesystem so that removable devices can be mounted by just entering a special mount-point directory.

8.8 Other Operating Systems

Solaris is the only other operating system that has a module for managing disks and partitions, however there are several differences between Linux and Solaris:

- Every Solaris disk has exactly 8 partitions, some of which may have no extent if they are not being used. Partitions never need to be created or deleted and there are no extended or logical partitions.
- When editing a Solaris partition, its extents can be changed without needing to delete and recreate it. This will, however, almost certainly result in the loss of data on the partition.
- Every partition has a type that indicates what it is used for. The `root` type is usually for the root directory filesystem, the `swap` type is for virtual memory, the `usr` type is for other filesystems, and the `unassigned` type is for empty partitions.
- Each partition has two flags—**Mountable** and **Writable**—which indicate whether it can be mounted or written to, respectively.
- The only filesystem supported on Solaris partitions is `ufs`—the native UNIX filesystem type.
- Partition labeling is not supported on Solaris.
- When editing the module access control, there is no **Can view non-editable disks?** option.

The RAID and LVM modules explained below are not available on Solaris or any other operating system.

8.9 Introduction to RAID

RAID (which stands for Redundant Array of Inexpensive Disks) is a method for combining multiple partitions on different disks into one large virtual device, also known as a RAID array. This has several advantages:

- You can create a single filesystem that is as big as all your existing hard disks, instead of needing to mount each one separately at a different mount-point directory.
- In most cases, reading to and writing from a RAID device is faster than accessing a single disk, because the data being read or written is spread across multiple drives.
- With the right configuration, data on a RAID device can survive even if any one of the hard disks fails. This is done by spreading redundant information across all drives, and comes at the cost of some disk space.

The different types of RAID configuration are called *levels*. The levels supported by Linux are:

Concatenated or Linear In this mode, all the partitions in the RAID array are combined end-to-end into one large virtual device. Data written to the device will fill up the first disk and then go on to the second disk, and so on. Linear mode does not generally make data access any faster, as all the blocks of a file being read or written are likely to be next to each other on the same disk.

RAID 0 or Striped As in linear mode, multiple partitions in striped mode are also combined into one large device. Data written to the array, however, will be spread evenly across all disks so that reading or writing a single large file is much faster. Ideally, if you had 5 disks in your striped RAID array, then accessing data would be 5 times faster. The only problem with this mode is that it does not deal well with disks that are not all the same size—any space on a disk that is larger than the rest will still be used, but only at its normal speed.

RAID 1 or Mirrored In mirrored mode, every partition in the array contains exactly the same data. This means that in the event of a disk failure, your data is safe even if only one disk survives. The down side is that under normal conditions most of the disks are wasted and the usable space on the array is only as big as the smallest partition. Reading from a mirrored array is as fast as reading from a striped array, but writing will be as slow as the slowest disk due to the need to write all data to all disks simultaneously.

RAID 4 or Parity Parity mode is rarely used, as it offers no real advantage over RAID 5. It provides protection against a single disk failure and increases read speed but not write speed. A RAID 4 array can survive the loss of any one disk because it dedicates one disk to the storage of parity information, which can be used to reconstruct data on other disks if one of them fails. Because all writes to the array cause a write to this disk, it becomes a bottleneck that slows down the entire array.

RAID 5 or Redundant This is the most useful RAID mode as it provides protection against a disk failure, increases read and write speeds, and combines multiple partitions into one large virtual device. A RAID 5 array can survive the loss of any one disk without the loss of all data, but at the expense of sacrificing some space on all the disks for storing redundant information. It is faster than linear mode, but not quite as fast as striped mode due to the need to maintain redundancy.

This chapter only covers the RAID configuration software on Linux. If your system has a separate RAID controller card or external array, you will need special software to set it up. Virtual RAID devices on hardware controllers will show up in the Partitions on Local Disks module for partitioning, just like any real hard disk would. They will not be visible or configurable in the Linux RAID module.

8.10 The Linux RAID Module

This module allows you to create, format, and delete RAID arrays on your Linux system. Like the other hard-disk related modules, it can be found under the Hardware category. When you

enter the module, the main page will display existing RAID devices (if any), as shown in Figure 8.4.

If Webmin detects that the commands used to set up RAID are missing from your system, an error message will be displayed on the main page of the module. Most Linux distributions, however, should have a package on their CD or website containing the RAID commands. A different error will be displayed if your Linux kernel has not been compiled with RAID support. In this case, you may have to recompile the kernel with RAID support turned on.

Assuming all the necessary packages have been installed, adding a new RAID device is relatively easy. The steps to follow are:

1. In the Partitions on Local Disks module, create a partition on each disk that you want to use for RAID. Existing partitions can also be used, as long as they do not contain any data that you do not want overwritten. A disk that is partially used for some other purpose can also have a new partition added for RAID use, although this may negate some of the performance benefits.

Every partition that is going to be part of the RAID array should have its type set to **Linux raid**. Unless you are using linear mode, all partitions should be the same size so that space on the larger partitions is not wasted.

2. At this point, it may be necessary to reboot your system. Some Linux kernels can only detect new partitions at boot time. If you do not reboot and the partition is not detected, the creation of the RAID device will fail.

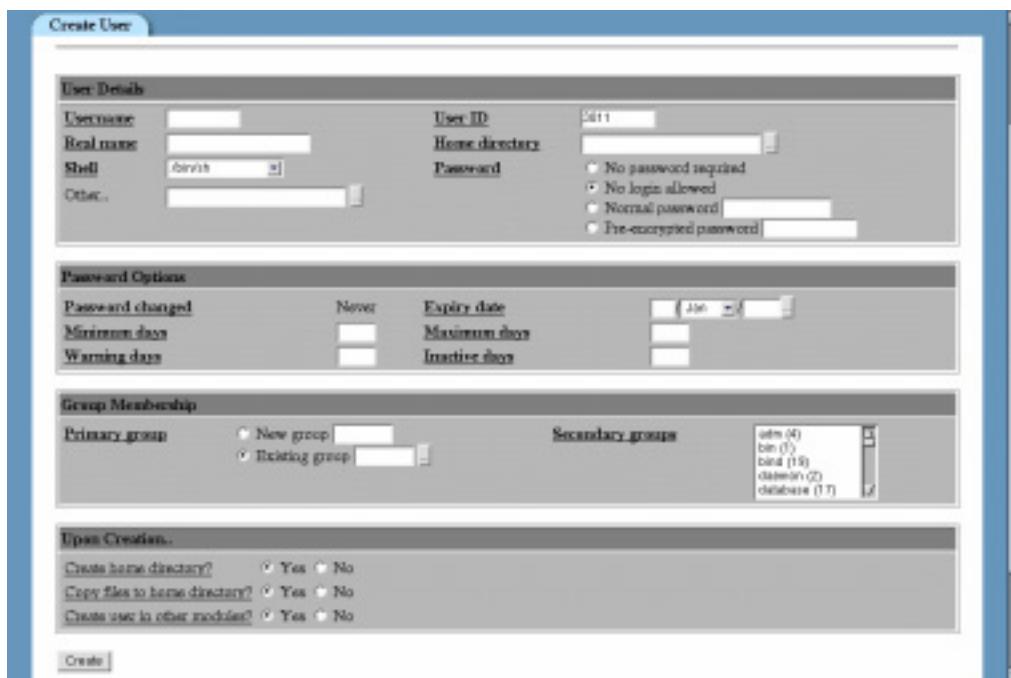


Figure 8.4 The Linux RAID module.

3. On the main page of the module, select the RAID level that you want to use and click the **Create RAID device of level button**. This will take you to a form for selecting the partitions to be part of the array and other options, assuming Webmin detects at least one unused partition on your system.
4. The **Partitions in RAID** option will list all hard disk partitions that are not currently in use for possible inclusion in your RAID device. It will also list any other RAID devices that are not in use, allowing you to theoretically create an array that contains other arrays. Select all the partitions that you want to be part of your new RAID device.
5. The **Force initialization of RAID?** option should be set to **Yes** if any of the selected partitions have been used before for other purposes. Otherwise, the creation of the new array will fail if a filesystem is detected on any of the partitions.
6. Click the **Create** button to set up the new array. If everything is successful, you will be returned to the main page of the module, which should now include your new RAID device.
7. If you want to create a filesystem on the new device so that it can be mounted, click on its icon to go to the device status page. If the RAID device is to be used for virtual memory, as part of an LVM volume group or as part of another RAID array, then this is not necessary.
8. Select the type of filesystem you want to create from the menu at the bottom of the page and click the **Create filesystem of type** button.
9. Select any options for the new filesystem, as explained in Section 8.4 “Creating a New Filesystem”. When done, click the **Create** button. A page showing the progress of the new filesystem’s creation will be displayed, which can take some time for large arrays.
10. Assuming that the formatting is successful, you can now use the Disk and Network Filesystems module to mount the new filesystem.

Existing RAID devices that are not in use can be deleted or deactivated by clicking on their icon on the main page of the module, and pressing the **Delete** button. Deleting a device will cause any data stored on it to be lost forever.

8.11 Introduction to LVM

LVM (Logical Volume Manager) is a powerful Linux feature that adds a layer of abstraction between the physical partitions on your system and the filesystems that they store. Partitions managed by LVM are called a *physical volumes*, which are combined together to form *volume groups*. From each volume group *logical volumes* can be created, on which filesystems are actually stored. The size of each volume group is the sum of the sizes of all its physical volumes. This space can be handed out to as many logical volumes as will fit into it, so that it could contain many small logical volumes or one huge one that spans multiple physical volumes (and thus partitions).

At first glance, LVM may not seem any more powerful than RAID, which can also combine multiple partitions into one large filesystem. It does, however, give you far more freedom to carve up disks into separate filesystems that may take up part of a disk, several disks, or anything in between. The only down side is that LVM does not support redundancy as RAID does in Levels 1 and 5.

The most useful feature of LVM is the ability to resize logical volumes and the filesystems within them up to the amount of free space in the volume group. Additional physical volumes

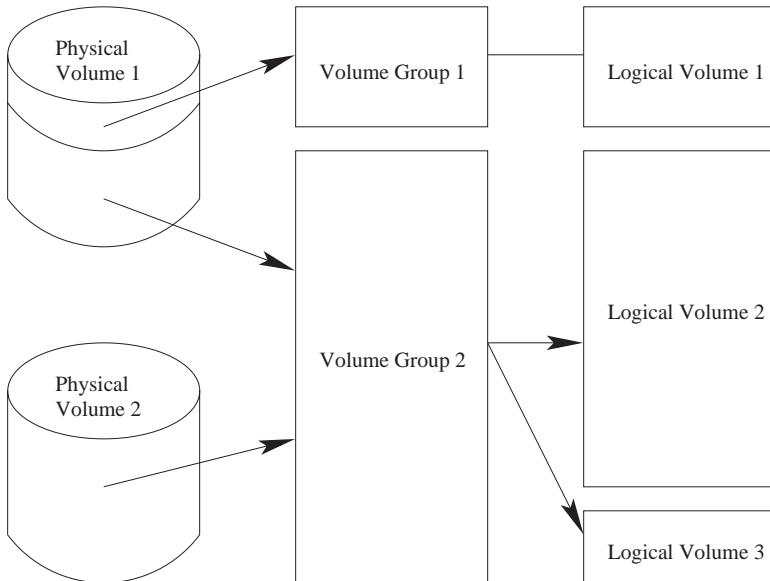


Figure 8.5 An overview of the logical volume manager.

(such as newly installed hard disk partitions) can be added to an existing volume group, subsequently increasing the amount of free space. For example, if your system has two hard disks whose partitions are combined to form a volume group, you might have a filesystem on a logical volume that is as big as both disks combined. If you begin to run out of disk space and want to enlarge the filesystem, you can install a new hard disk, add it to the volume group, and then enlarge the logical volume to make use of all the new free space! This is far more convenient than mounting the new hard disk as a subdirectory somewhere under the existing filesystem.

Physical volumes can also be removed from an LVM volume group, as long as there is enough free space in the group to store data that used to be on the physical volume. This means that you could theoretically remove a small hard disk from your system and replace it with a larger one, without having to manually copy files around.

8.12 The Logical Volume Management Module

Webmin's Logical Volume Manager module allows you to perform almost all of the tasks that can be done using the command-line LVM tools. When you enter the module from the Hardware category, the main page shows all existing volume groups and their physical and logical volumes, as shown in Figure 8.6.

Because the module depends upon the LVM tools such as `vgcreate`, the main page will display an error message if they are not found. They should, however, be available on your distribution CD or website if you are running a reasonably recent version of Linux. It also checks to see if your kernel supports LVM by looking for the `/proc/lvm` directory. If support is missing, you will need to load the `lvm-mod` kernel module with the command `modprobe lvm-mod` or recompile your kernel with LVM support.

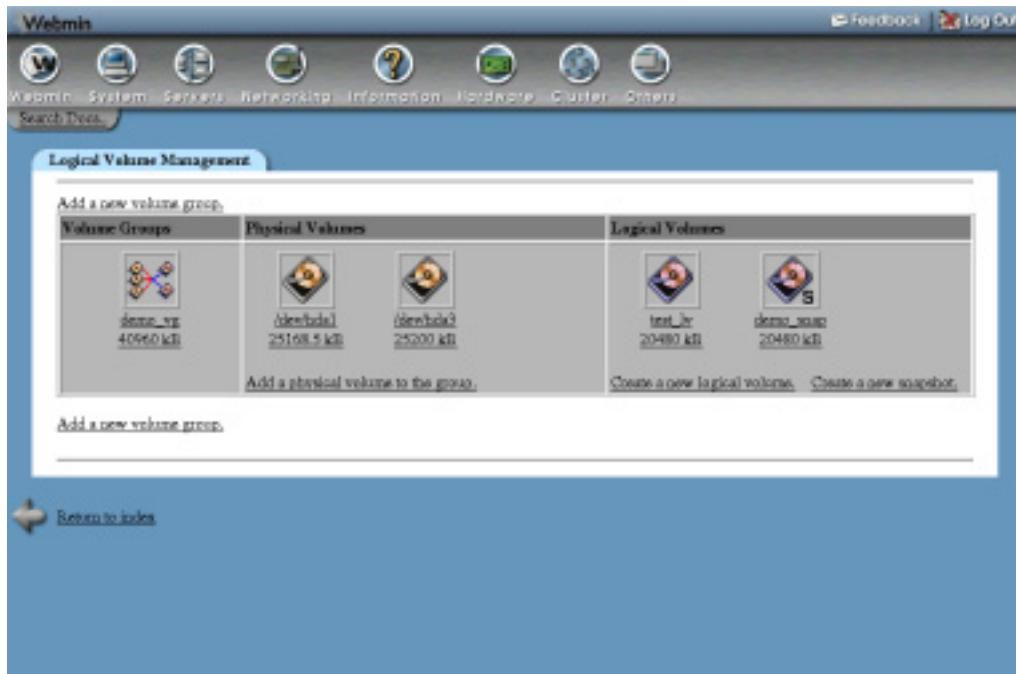


Figure 8.6 The Logical Volume Management module.

8.13 Creating a New Volume Group

Assuming you have at least one partition free for use by LVM, setting up a new volume group is easy. The process to follow is:

1. In the Partitions on Local disks module, change the types of any partitions that you want to include in the volume group to **Linux LVM**. Trying to use partitions of any other type will fail.
2. Back in the Logical Volume Manager module, click on the **Add a new volume group** link, which will take you to the volume group creation form.
3. Enter a name for your new volume group in the **Volume group name** field. This should be short and contain no spaces, like *data_vg*.
4. Select the initial partition to be included in your volume group with the **Initial physical device** field. Only partitions or RAID devices that Webmin determines are not in use will appear in the list. You also specify a partition by device file name by selecting the **Other** option and entering the file name into the field next to it.
If **Other** is the only option available, Webmin has not detected any partitions free for use by LVM.
Be aware that any data on the partition or device that you select will be lost forever, even if the volume group is not actually used.
5. Click the **Create** button. If all goes well, you will be returned to the main page of the module and your volume group with its initial physical volume will be displayed.

6. To add more physical volumes to your new volume group, see Section 8.14 “Adding and Removing a Physical Volume”.

8.14 Adding and Removing a Physical Volume

Once a volume group has been created with its initial physical volume, you can add new partitions or RAID devices to it at any time. This will increase the amount of free space in the volume group, and allow you to create more logical volumes or extend existing ones. To add a physical volume, follow these steps:

1. If you are adding a disk partition, use the Partitions on Local Disks module to change its type to **Linux LVM**.
2. On the main page of the Logical Volume Management module, click on **Add a physical volume to the group** inside the section for the appropriate volume group. This will take you to a page for selecting the partition or RAID device to add.
3. Choose the one that you want to add from the list in the **Disk device** field, or select the **Other** option and enter a device file manually. Only partitions that Webmin thinks are not in use elsewhere will be available for selection.
Be aware that any data on the partition or device that you select will be lost forever.
4. Click the **Add to volume group** button to add the physical volume. If successful, you will be returned to the main page of the module.

It is also possible to remove a physical volume from a volume group, as long as there is enough free space in the group to store all the data that was previously on the physical volume. The steps for doing this are:

1. On the main page, click on the icon for the physical volume that you want to remove.
2. Click the **Remove from volume group** button. Assuming that removal is possible, there may be a delay as data is shifted to other physical volumes.
3. Once the removal is complete and the browser returns to the list of volume groups, you can immediately use the partition for some other purpose. It can be formatted with a file-system and mounted, included in a RAID group, or even added to another volume group.

8.15 Creating and Deleting a Logical Volume

As long as a volume group has some free space, you can add a logical volume to it at any time. A logical volume can be any size, but the size will always be rounded up to a multiple of the allocation block size used by the volume group (4 MB by default). You can see the current block size, blocks allocated, and total blocks by clicking on a volume group’s icon on the main page of the module.

The steps for adding a new logical volume are as follows:

1. On the list of volume groups, click on the **Create a new logical volume** link next to the volume group to which you want to add it.
2. In the **Volume name** field, enter a name for your new logical volume. This should be short and contain no spaces, like *data_lv*.

3. For the **Volume size** field, enter the number of kilobytes to allocate to this volume. Whatever you enter will be rounded up to the nearest **Allocation block size** shown below. By default, this field will be set to the total amount of free space in the volume group.
4. If the **Allocation method** option is set to **Contiguous**, all space reserved for this logical volume will be in one large block on disk. This can speed up access to the data but is inflexible if you are adding and removing logical volumes causing the volume group to become fragmented. Therefore, it is usually best to leave the option set to **Non-contiguous**.
5. The **Volume striping** option controls how data for the logical volume is laid out on disk. The **Disabled** option is similar to linear mode in RAID, while the **Stripe across** option is similar to striped mode. See Section 8.9 “Introduction to RAID” for a more detailed explanation.
6. When all the fields are set to your satisfaction, click the **Create** button. As long as all fields have been filled in properly and there enough free space in the volume group, the browser will return to the main page of the module and a new icon for your logical volume should be visible.
7. Assuming you want to mount the new logical volume somewhere, you will first need to create a filesystem on it. To do this, click on its icon on the main page of the module that will take you to the logical volume editing page.
8. Select the type of filesystem you want to create from the menu at the bottom of the page, and click the **Create filesystem of type** button.
9. Select any options for the new filesystem, as explained in Section 8.4 “Creating a New Filesystem”. When done, click the **Create** button. A page showing the progress of the new filesystem’s creation will be displayed, which can take some time for large volumes.
10. Assuming that the formatting is successful, you can now use the Disk and Network Filesystems module to mount the new filesystem.

Existing logical volumes can be deleted from their volume group to free up space or reduce the volume group size. Before you can delete a logical volume, it must have been unmounted in the Disk and Network Filesystems module. When it is deleted, any data that it contained will be lost forever.

To remove a logical volume, follow these steps:

1. Click on its icon on the main page of the module, which will take you to the logical volume editing form.
2. Click the **Delete** button. This will bring up a page asking if you are really sure about deleting it.
3. Click **Delete Now** to confirm. Once it has been removed from the volume group, your browser will return to the main page of the module. The space freed up can be reused for another logical volume immediately.

8.16 Resizing a Logical Volume

One of the most powerful features of LVM is its ability to enlarge or reduce existing logical volumes, even if they contain a filesystem. Webmin, however, only supports the resizing of `ext2`, `ext3`, `reiserfs` and `jfs` filesystems at the moment—logical volumes formatted with other filesystem types (such as `xfs`) cannot be resized without losing data. You must also unmount a

logical volume before resizing it, and then remount afterwards—there is no way to resize a filesystem that is currently in use.

As would be expected, a logical volume can only be enlarged by the amount of free space in its volume group. When shrinking a logical volume containing a supported filesystem, its size cannot be reduced to less than the space occupied by files on the filesystem. Currently, `jfs` filesystems cannot be shrunk at all—they can only be enlarged.

The steps to follow for resizing a logical volume are:

1. In the Disk and Network Filesystems module, make sure the logical volume is unmounted.
2. On the main page of the Logical Volume Management module, click on its icon. This will take you to the volume editing form.
3. Enter a new size in kB in the **Volume size** field. The size cannot be increased by more than the amount of free space in the volume group, or reduced to less than the space occupied by files on the filesystem, unless you plan to recreate the filesystem.
4. Click the **Save** button. When resizing a volume containing an `ext2`, `ext3`, `reiserfs`, or `jfs` filesystem, you will be returned to the main page of the module as long as no problems are encountered.

If the filesystem could not be shrunk below the amount of space occupied by its files, however, an error page will appear offering you the option of resizing anyway. Clicking the **Resize Logical Volume** button will force a resize, but any files on the volume will be lost and you will need to re-create the filesystem.

If you are resizing a logical volume containing some other type of filesystem (such as `xfs`), or one whose contents are unknown to Webmin, a page asking you to confirm the resize will appear. If you click the **Resize Logical Volume** to go ahead, any filesystem on the volume will be lost and need to be created again.

5. If the filesystem was resized successfully, you can remount it in the Disk and Network Filesystems module. Otherwise, you will need to recreate it as explained in Section 8.15 “Creating and Deleting a Logical Volume”.

8.17 Creating a Snapshot

A snapshot is a special kind of logical volume that is actually a temporary, read-only copy of another volume. When a snapshot is created, it appears to contain a copy of all the data in the source volume, so that if the source is changed, the snapshot remains the same. In order to save on disk space, the snapshot really only stores data that has changed on the original logical volume since it was created. This makes it possible to create a snapshot copy of a 100 MB of volume even if the volume group has less than 100 MB of free space.

Snapshots are useful for quickly freezing a filesystem at some point so that it can be safely backed up. A snapshot can even act as a kind of backup itself, to which you can revert if something goes wrong with files on the original volume. The only down side is that a snapshot can only be safely created when the source logical volume is unmounted, as a mounted filesystem will not be in a valid state for copying.

To create a snapshot, follow these steps:

1. In the Disk and Network Filesystems module, unmount the filesystem on the original logical volume, if necessary.
2. Back in the Logical Volume Management module, click on the **Create a new snapshot** link in the same volume group as the original volume.
3. On the snapshot creation form, enter a short name without spaces into the **Volume name** field—*data_snap*, for example.
4. For the **Volume size**, enter the amount of disk space (in kB) that you want to allocate to this snapshot for storing differences made to the original logical volume after the snapshot was created. If the amount of space is too small and too many changes are made to the logical volume, I/O errors will start to occur when reading files in the snapshot filesystem.
5. For the **Snapshot of logical volume** field, select the logical volume of which you want to make a copy.
6. Click the **Create** button to create the snapshot and return to the main page. An icon for your new snapshot will appear among the other logical volumes in its volume group.
7. In the Disk and Network Filesystems module, remount the filesystem on the original logical volume. You can mount the filesystem on the snapshot separately here as well.

Once created, a snapshot can be resized in the same way that you would resize a normal logical volume. This does not, however, resize the filesystem on the snapshot—instead, it changes the amount of space available for storing differences between the snapshot and original volume group. A snapshot can also be deleted, assuming the filesystem on it has been unmounted first. Any data in the snapshot will be lost, but since it is just a copy of another volume this isn't likely to matter much.

8.18 Summary

This chapter has covered the three low-level devices that can be used by Linux systems to store filesystems. The simplest are regular disk partitions, which are just a single section of a hard disk. RAID devices are more complex, as they combine multiple partitions into single, large virtual partitions. LVM, the most complex and powerful of all, can be used to create volumes that cover multiple partitions and that contain filesystems that can be expanded as more space is added. All of these device types appear the same to users when they have been initialized with a filesystem and mounted on a directory somewhere.

After reading this chapter, you should understand the differences between them in terms of simplicity, reliability, and flexibility. You should be able to choose the best one to use for your own system based on the number of hard drives that you have and the importance of your data.

Bootup and Shutdown

This chapter explains methods for starting servers and services at boot time, and tells you how to use Webmin to run your own commands at startup.

9.1 Introduction to the Linux Boot Process

The very first thing that happens when a PC starts up is the loading of the BIOS from ROM. The BIOS (Basic Input/Output System) performs memory and other hardware checks, and then loads a tiny piece of code from the first part of one of the system's hard disks—known as the master boot record or MBR. This piece of code is called a *boot loader*, and is responsible for displaying a menu of operating systems to the user and loading one of them. There are several boot loaders available for Linux, such as LILO and GRUB, but they all do basically the same thing.

Once the kernel has been loaded, it mounts the `root` filesystem and runs the `init` program, which is responsible for managing the rest of the boot process. It reads the `/etc/inittab` file and executes the commands it specifies—the most important of which begins execution of bootup scripts. Each of these scripts is responsible for a single task, such as initializing network interfaces, starting a web server, or mounting other filesystems. The scripts have a fixed order in which they must execute because some of the later scripts are dependant on earlier ones. For example, network filesystems cannot be mounted until network interfaces have been enabled.

At shutdown time, a series of scripts is also run to shut down servers and unmount filesystems. These scripts also have a fixed order so that the deactivation of networking and other basic services happens last. If requested and supported by the hardware, the last step in the shutdown process will be the powering off of the system by the kernel.

When a Linux system starts up, different scripts are executed depending on the *runlevel* in which it is starting. The runlevel can be set by the boot loader or by the `/etc/inittab` file.

The commonly used runlevels are:

5 – Graphical mode All servers and services will be started, and X started to display a graphical login prompt on the console.

3 – Multi-user mode All servers and services are started, but only the normal text login is available on the console.

2 – Multi-user mode without NFS Almost all servers and services are started, but NFS filesystems are not mounted.

1 – Single user mode Only the most basic system initialization is done, and a root shell opened on the console. This runlevel is useful if some bootup script is failing and making your system unbootable.

See Section 9.9 “The SysV Init Configuration Module” for information on how to change the bootup runlevel.

The directory `/etc/rc.d/init.d` is usually used to store the actual bootup shell scripts. The scripts that are started or stopped in each runlevel are determined by symbolic links from the `/etc/rc.d/rcX.d` directory, where *X* is the runlevel number. Each symbolic link has a name like `sYYscriptname`, in which *YY* is the order that the script is started in the boot process—the lower the number, the earlier the script starts. So `/etc/rc.d/rc5.d/S10network` would be run in runlevel 5 before `/etc/rc.d/rc5.d/S80sendmail`.

Not all Linux distributions use these directories for their bootup scripts. Some use `/etc/init.d` for the actual script files, while others (such as older versions of SuSE) put everything in the `/sbin` directory. Fortunately, `/etc/rc.d` seems to be becoming the standard base directory in newer distributions. Of course, if you are using Webmin you don’t have to worry about the locations of any of these directories as it always knows where they are.

9.2 The Bootup and Shutdown Module

This module allows you to create and edit the scripts that are run at bootup and shutdown, called *actions* by the module. It can be found under the System category in Webmin, and when you enter it, the main page will display a list of all available actions, whether or not they are started at boot, and a short description for each. See Figure 9.1 for an example.

Each Linux distribution has its own set of standard action scripts, so on one system the script `httpd` may start the Apache Web server, but on another it may be called `apache`. You should, however, be able to get a good idea of what each script does from its description.

9.3 Configuring an Action to Start at Bootup

If some server on your system such as Apache or Squid is not currently being started at boot time, you can use this module to change that. On most Linux distributions, every server that comes with the distribution will have its own bootup action script, but not all will be enabled by default. To configure an action to start at boot time, the steps to follow are:

1. On the main page of the module, click on the name of the action that you want to enable. This will take you to the action editing page, as shown in Figure 9.2.
2. Change the **Start at boot time?** option from **No** to **Yes**.

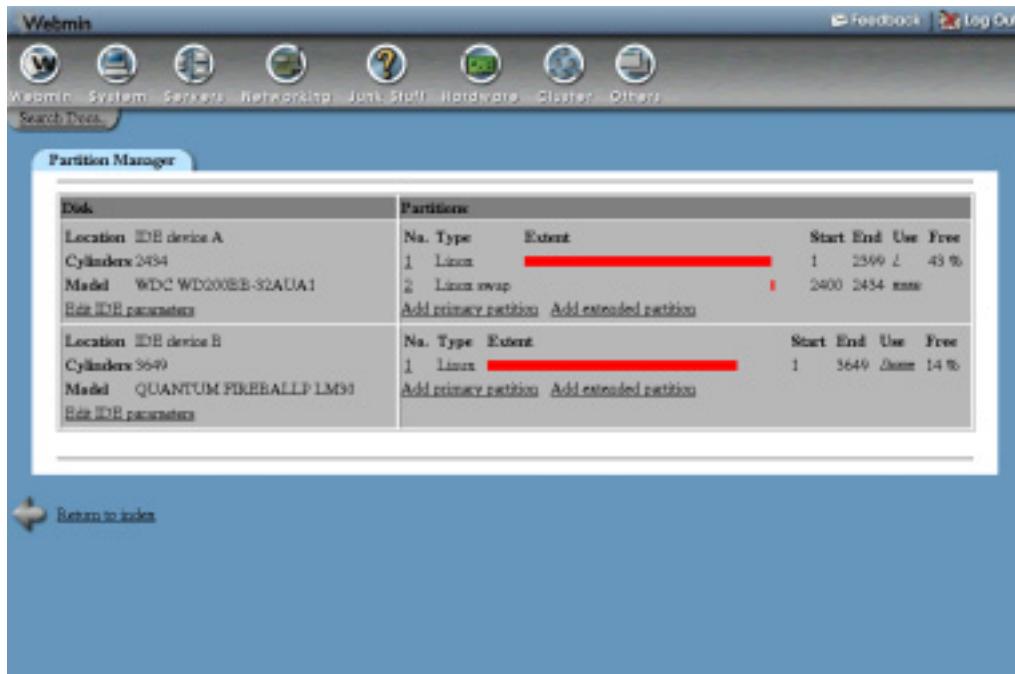


Figure 9.1 The Bootup and Shutdown module.

3. Click the **Save** button, and your browser will return to the list of actions on the main page.

If there is a server that is currently being started at boot time that you want to disable, just follow the same steps but set the **Start at boot time?** option to **No** instead.

9.4 Starting and Stopping Actions

Even though action scripts are normally started at boot time and stopped at shutdown, you can start or stop them at any time using Webmin. Many action scripts can also perform additional functions, such as showing the status of a server or reloading its configuration. To start or stop an action, do the following:

1. On the main page of the module, click on the name of the action. This will take you to the action editing form shown in Figure 9.2.
2. At the bottom of the page (in the middle) will be a row of buttons, each for running an action script to perform some function. Depending on the script there may be different buttons available, but some of the most common are:
 - Start Now** Immediately starts the server or service. On some versions of Linux, this will do nothing if the action has already been started and the server is already running.
 - Stop Now** Stops the server or service. In some Linux versions, this will do nothing unless the action has already been started.

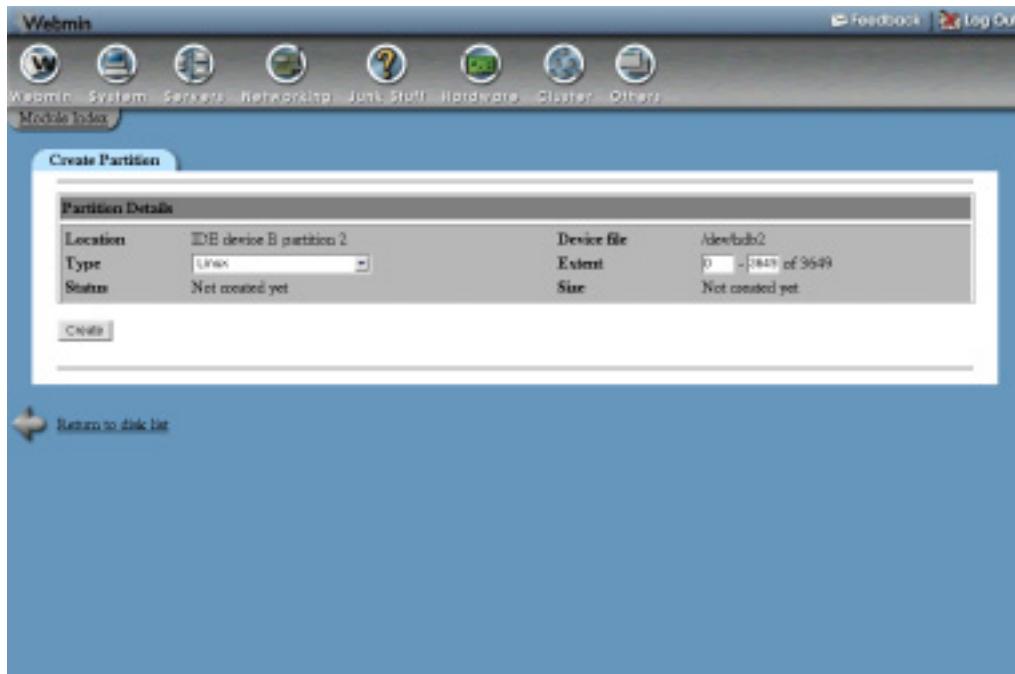


Figure 9.2 The action editing form.

Restart Now Stops and restarts the server. In many cases, this will do nothing if the action has not been started yet.

Reload Now Where available, this function tells the server started by the action to reread its configuration files.

Show Status Just displays a message telling you if the server is running or not, and if so what its PID is.

3. After you click the button for the function that you want to perform, a page showing the output from the action script will appear. This should indicate whether the action was performed successfully or not.

9.5 Adding a New Action

If you have a command that you want run at boot time, creating a new action script is the best way to set it up. Servers like Apache or Qmail that have been compiled and installed manually do not have actions, so you will need to create one that runs whatever command is necessary to start the server.

To create your own action, follow these steps:

1. On the main page of the module, click the **Create a new bootup and shutdown action** link above or below the list of existing actions. This will take you to the form shown in Figure 9.3 for entering the code for your new action script.



Figure 9.3 The action creation form.

2. In the **Name** field, enter a short name for the action like *qmail*. Every action must have a unique name.
3. In the **Description** field, enter a few lines of text to describe your action—maybe something like *Start the Qmail mail server*. This will show up on the main page of the module under the **Description** column.
4. The **Bootup commands** field must be filled in with the shell commands that you want to run when your action is started at boot time. For example, if you wanted to start Qmail you might enter `/var/qmail/rc`.
5. The **Shutdown commands** field should be filled in with commands that you want to run when your action is stopped. For example, to stop Qmail you might enter `killall -9 qmail-send`.
6. Assuming you want your action to run at boot time, set the **Start at boot time?** option to **Yes**.
7. Finally, click the **Create** button to save the new action. Webmin will create a script in the `/etc/rc.d/init.d` directory combining the commands you entered with a standard wrapper to make a valid action script. Your action will be set to run in the current runlevel, with order number 99 so that it is run last. If you want to control exactly which runlevels and in what order your action is run, see the **Allow selection of individual runlevels** option in Section 9.7 “Configuring the Bootup and Shutdown Module”.

After an action has been created, you can edit the start and stop commands by following this process:

1. On the main page of the module, click on the name of your action. This will take you to the action editing form shown in Figure 9.2.

2. In the **Action script** text box, look for a line like 'start'). The commands that are run at boot time will come after it, down to the a line containing just ;;. Edit them as you wish, but leave the surrounding code alone.

Similarly, the commands that are run when the action is stopped are between 'stop') and ;;. Changing any other part of the script is a bad idea unless you know what you are doing.

3. Click the **Save** button to apply your changes.

Any of the existing action scripts can be edited using Webmin, not just your own creations. Be careful editing them, as they may have a format totally different to the scripts created by Webmin.

9.6 Rebooting or Shutting Down Your System

Linux systems should always be rebooted or shut down using the appropriate commands, rather than simply turning off the power or hitting the reset button. If not, you may lose data on your local hard drives and will certainly have to wait through a lengthy filesystem check with fsck at boot time if using a non-journaling filesystem.

To reboot, simply do the following:

1. At the bottom of the main page of the Bootup and Shutdown module, click the **Reboot System** button. This will take you to a page confirming if you really want to reboot.
2. Click the **Reboot System** button on the confirmation page. The shutdown process will start immediately, and if you are logged in at the console your session will be logged out. After all the shutdown scripts have been run, the system will bootup again as explained in the introduction.

The process for shutting down is almost identical—just use the **Shutdown System** button at the bottom of the page instead.

9.7 Configuring the Bootup and Shutdown Module

Like most modules, Bootup and Shutdown can be configured by clicking on the **Module Config** link on the main page. This will take you to the standard configuration editing page, on which the settings in Table 9.1 are available under the **Configurable options** header.

None of the other options on the configuration page should be changed, as they are automatically set by Webmin based on your operating system type.

9.8 Other Operating Systems

Many other UNIX operating systems—but not all of them—use the system of bootup scripts used by Linux. Even those that do use it have some slight differences in their implementation, and almost all use different directories for storing the actual scripts and links.

Sun Solaris, HP/UX, SCO UnixWare, SCO OpenServer, CompaqTru64/OSF1, and SGI Irix All these operating systems use action scripts that are very similar to Linux, but are stored in different directories. Because those that come with the

Table 9.1 Module Configuration Options

Allow selection of individual runlevels	If set to Yes when editing or creating an action, you will be able to enter the exact order number for the action in each runlevel. Because this is rather complex, this option is set to No by default so that Webmin only displays a single Start at boot? option when creating or editing an action.
Display actions with descriptions	If this option is set to Yes, and show all runlevels , the main page will show exactly which runlevels each action is started in, along with the description. If set to Yes , each action will only show whether it is started at boot or not, and its description. This is the default on most systems. If set to No , the main page will display only a table of action names with no descriptions or boot information. This can be useful if you have a lot of actions or your operating system does not include descriptions in the action scripts.
Show boot order of actions?	If Yes is selected, the main page will include the boot order of each action in the current runlevel, or in some other runlevel. The default No option hides this information.
Show current status of actions	On some Linux distributions, the standard action scripts can report the status of the servers that they start. This option allows Webmin to display this status information in various places. If set to No , the status of actions is never displayed unless you click the Show Status button when editing an action. If set to On action page only , Webmin will display the current status of an action when you edit it by selecting it from the main page of the module. If set to On index and action pages , the main page of the module will display the current status of every single action. This provides a lot of information, but can be very slow.
Sort actions by	When Boot order is chosen, actions on the main page are listed in the order that they will be started by your system in the current runlevel. The default of Name causes them to be sorted by name instead.

system do not have descriptions, the main page of the module will just display action names by default.

FreeBSD, NetBSD and OpenBSD The BSD family of operating systems does not use action scripts at all, relying instead on a fixed set of scripts that are run at

boot time. One of these scripts (`/etc/rc.local`) allows system administrators to add their own commands to be run at boot time.

On any of these operating systems, the main page of the module will just display a form for editing the `rc.local` file, above the **Reboot System** and **Shutdown System** buttons. To add any commands that you want run at boot time, just enter them into the text box and click the **Save** button.

IBM AIX AIX is very similar to the BSD operating systems in that it does not have action scripts. Instead, the file `/etc/rc` can be edited to add additional commands to be run at boot time, using the form on the main page of the module.

Apple Mac OS X Apple's version of UNIX uses a totally different set of files for storing actions to be run at boot time than any other supported operating system. Separate action scripts still exist, but the user interface in this module for viewing and editing them is quite different.

If your operating system is not in this list, then the Bootup and Shutdown module does not support it at all, therefore the module icon will not appear in Webmin.

9.9 The SysV Init Configuration Module

As explained in the introduction to this chapter, the very first file read by the system to determine which commands to run at boot time is `/etc/inittab`. It is read by the `init` program, which is the first process to be run after the Linux kernel finishes loading, and remains running until the system is shut down. The `inittab` file specifies which runlevel to boot into, the commands to be run to start all of the action scripts, processes to begin displaying text and graphical login prompts, and commands to run in the case of an impending power failure.

The SysV Init Configuration module, found under the System category in Webmin, can be used to edit any of these commands. As they are critical for ensuring that your system boots up properly, however, editing them is a bad idea unless you really know what you are doing. The only thing that you might want to change is the bootup runlevel so your system does not display an unnecessary graphical login prompt if it is not needed.

To change the initial runlevel, follow these steps:

1. Enter the SysV Init Configuration module. The main page will display a list of commands and the runlevels and situations in which they are executed, as shown in Figure 9.4.
2. Click on the entry in the **ID** column for the row in which the **Action** is **After system boot**. This will take you to a form for editing the `inittab` file entry.
3. For the **Bootup runlevel** option, de-select whichever level is currently selected and choose a new one. Make sure that you choose exactly one level, such as **3** (for text login mode) or **5** (for graphical login mode). See the explanation in the introduction to this chapter for details on what each runlevel means.
4. Click the **Save** button to have your change written to the `inittab` file. The browser will return to the main page of the module.
5. If you like, you can reboot the system now using the Bootup and Shutdown module.

The module is also available on the Solaris, HP/UX, UnixWare, OpenServer, AIX, and Irix operating systems. Its basic structure and purpose is the same on all systems, but the actual default commands will differ significantly. The previous instructions for changing the bootup runlevel, however, will work on all operating systems.

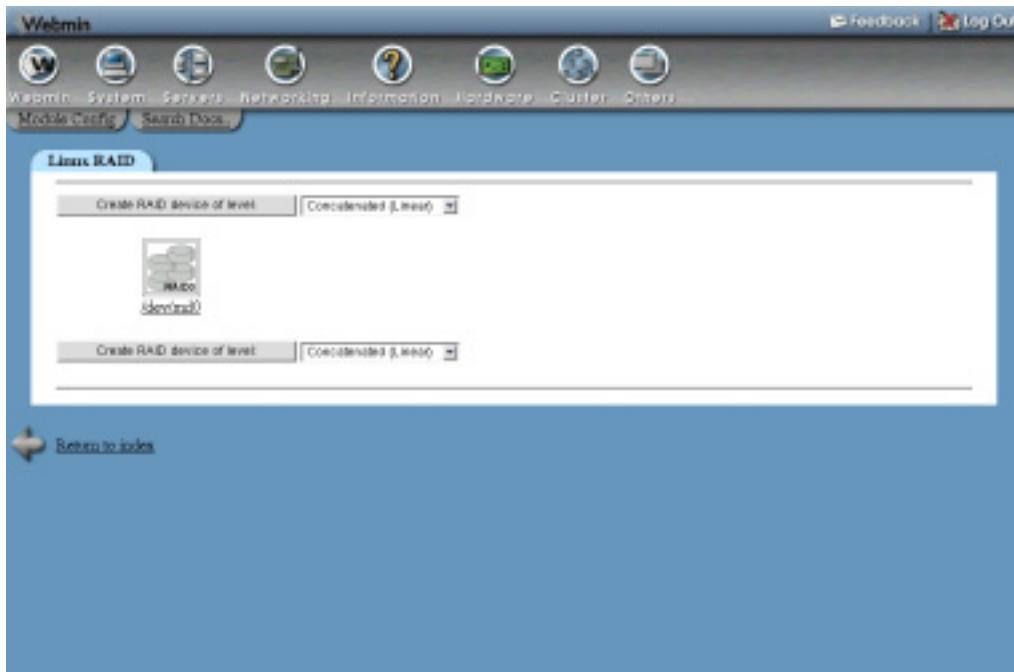


Figure 9.4 The SysV Init Configuration module.

9.10 Summary

This chapter has covered the bootup scripts used on Linux and many other UNIX variants, and shown how to manage them using Webmin. It has also explained where they fit into the overall Linux boot process along with the BIOS, boot loader, and `init` program. Finally, it has covered runlevels and the use of the SysV Init Configuration module to select the runlevel into which your system boots.

Scheduled Commands

In this chapter, you can learn about the UNIX Cron and At facilities, used for running commands on a regular schedule or once at some future time.

10.1 Introduction to Cron Jobs

A *Cron job* is a UNIX term for a command that is run on a regular schedule by the cron daemon. Each job is owned by a UNIX user, and runs with the permission of that user. Each has a set of minutes, hours, days, months, and days of weeks on which it runs, allowing considerable flexibility in scheduling. For example, a job may run every 10 minutes, or at 3 a.m. every day, or at 5 p.m. Monday to Friday in January, February, and March.

Cron jobs are very useful for performing regular system tasks, such as cleaning up log files, synchronizing the system time, backing up files, and so on. Most Linux distributions will have several Cron jobs that were set up by default as part of the operating system install process for doing things like removing unneeded kernel modules, updating the database used by the locate command, and rotating log files.

The actual Cron job configuration files are stored in different places, depending on whether they are part of a package included in your Linux distribution or created by a user. The `/var/spool/cron` directory is for jobs created manually by users, and contains one file per UNIX user. The `/etc/crontab` file and the files under the `/etc/cron.d` directory contain jobs that are part of packages, such as those that are part of your distribution.

10.2 The Scheduled Cron Jobs Module

The Webmin module for editing Cron jobs can be found under the System category. When you enter it, the main page displays a table of all the existing jobs on your system, as shown in Figure 10.1. For each action, the owner, active status, and command are listed. If the command for a job is too long, it will be truncated for display on the page.

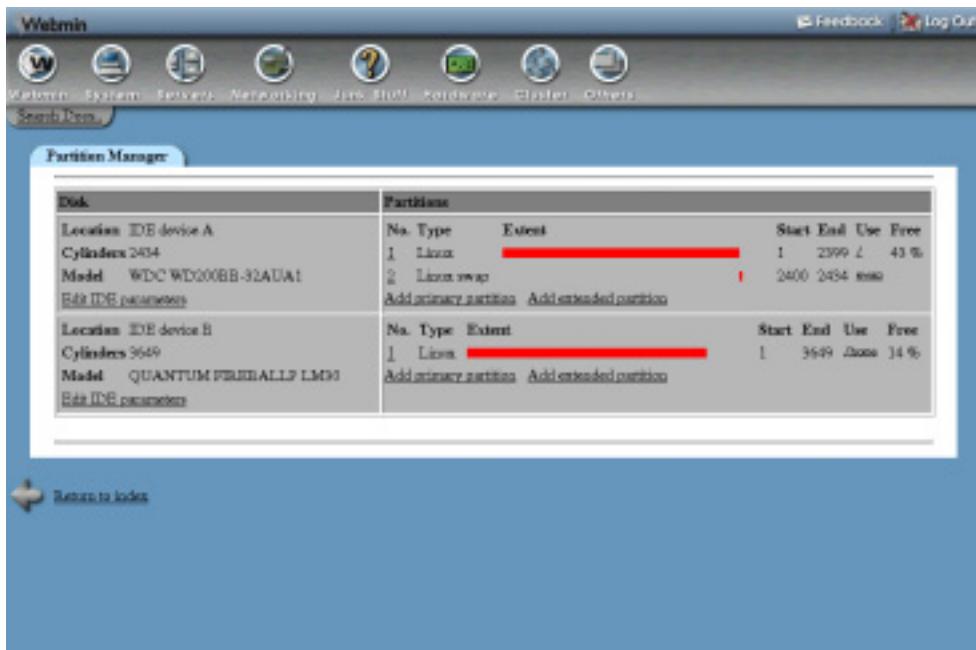


Figure 10.1 The Scheduled Cron Jobs module.

10.3 Creating a New Cron Job

Using Webmin, you can easily create a new Cron job that will execute as any UNIX user on your system. To achieve this, follow these steps:

1. On the main page of the module, click on the **Create a new scheduled cron job** link above or below the list of existing jobs. This will take you to the job creation form shown in Figure 10.2.
2. In the **Execute cron job as** field, enter the name of the UNIX user you want the job to execute as. The command executed by the job will run in the user's home directory with his full permission.
3. The **Active?** field can be set to **No** if you don't want this new job to actually be executed. This is useful for creating jobs to be enabled at a later date.
4. In the **Command** field, enter the shell commands that you want the Cron job to run. Just as at the shell prompt, multiple commands can be entered separated by a semicolon (;), and all the normal shell operators such as >, <, and && can be used.
By default, any output from the command will be emailed to the owner of the Cron job. If you don't want this to happen, make sure that output is redirected to a file or /dev/null.
5. Anything that you enter into the **Input to command** field will be fed to the command as input when it is run. If for example your command was mail foo@bar.com, anything entered into the field would be sent to that email address.
6. In the **When to execute** section, choose the times and dates on which you want the command to execute. For each of the **Minutes**, **Hours**, **Days**, **Months**, and **Weekdays** options you can choose multiple times or dates, or select the **All** option.

For example, to have a job executed at *3:15a.m.* every *Monday* and *Friday*, you should change the **Minutes** option to **Selected** and select *15*, change the **Hours** option to **Selected** and select *3*, and the **Weekdays** option to **Selected** and select *Monday* and *Friday*. The **Days** and **Months** options would remain on **All**.

7. Click the **Create** button to add the new Cron job. Assuming there are no errors in your selections, you will be returned to the main page of the module and your new job should appear next to its owner.



Figure 10.2 The Cron job creation form.

10.4 Editing a Cron Job

Existing Cron jobs, including those created by users through Webmin or included with your operating system, can be edited and rescheduled using this module. Be careful when editing jobs that came as part of your distribution though, as some perform important tasks like truncating web server, mail, and login log files so that they do not use up all of your disk space.

To edit an existing job, follow these steps:

1. On the main page of the module, click on the command for the job that you want to edit. This will take you to the module editing form, which is similar to Figure 10.2.
2. Change any of the details of the job, including the user, command, active status, and execution times and dates.
3. When done, click the **Save** button, and you will be returned to the main page of the module. Existing Cron jobs can be deleted by following these steps, but clicking the **Delete** button instead of **Save**. You can also force the immediate execution of a job by clicking the **Run Now** button on the edit page, which will execute the command and display any output that it produces.

10.5 Controlling Users' Access to Cron

The Scheduled Cron Jobs module can also be used to control access to the `crontab` command by UNIX users at the command line. This can be useful if you allow untrusted users to log in to your system, and want to prevent some of them from setting up Cron jobs to run commands and take up CPU time when they are not logged in. Usually by default, all users have the ability to create Cron jobs, but you can change that by following these steps:

1. At the bottom of the module's main page, click on the **Control user access to cron jobs** link. This will take you to a form for entering the usernames of users who can or cannot use Cron.
2. To grant access to all users, select the **Allow all users** option.
To grant access to only some users, select the **Allow only listed users** option and enter their usernames into the text field.
To give access to all but a few users, select the **Deny only listed users** option and enter the usernames of the people to whom you want to deny access into the text field.
3. When done, click the **Save** button.

If a user has been denied access to Cron, you will no longer be able to use the module to create, edit, or delete jobs belonging to him. Existing jobs, however, may continue to execute!

10.6 Module Access Control Options

As described in Chapter 52, it is possible to use the Webmin Users module to control for which UNIX users a Webmin user can edit Cron jobs. To set this up, you must first grant the user access to the module, then follow these steps:

1. In the Webmin Users module, click on **Scheduled Cron Jobs** next to the name of the user that you want to restrict.
2. Change the **Can edit module configuration?** field to **No**, so the user cannot edit the commands that Webmin calls to create and edit jobs.
3. Switch the **Can edit cron jobs for** field from **All users** to one of the other options. The most commonly used is **Users with UID in range**, which allows you to enter a minimum and maximum UID into the fields next to it.
Never allow an untrusted user access to the Cron jobs of system users like `root` or `bin`, as this will clearly give him full access to your system and defeat any other Webmin access control.
4. Set the **Can control user access to cron?** field to **No**, so that the Webmin user cannot stop users outside his control using Cron.
5. Click the **Save** button at the bottom of the page to make the access control active.

10.7 Configuring the Scheduled Cron Jobs Module

Most of the module settings that you can view by clicking on the **Module Config** link on the main page are set by default to match the installed operating system, and vary rarely need to be changed. However, there is one field that effects the module's user interface, shown in Table 10.1.

Table 10.1 Module Configuration Options

Maximum command length to display	This field determines how many characters of each Cron job command will be shown on the module's main page. You can either select Unlimited to show them all (which may cause lines to be wrapped), or enter a maximum number of characters. The default is 80.
--	--

10.8 Other Operating Systems

Cron is available on almost all UNIX systems, with very similar capabilities. That means that this module appears almost identically on all operating systems, with only a couple of minor differences. On some, there is no **Input to command** field available for when creating or editing a job. On others, when controlling which users have access to Cron, the default **Allow all users** option will be replaced with **Allow all users except root** or **Deny all users**.

Internally, other operating systems use different directories for storing Cron jobs—Solaris for example uses `/var/spool/cron/crontabs` instead of `/var/spool/cron` on Linux. Most other systems do not have an `/etc/crontab` file or `/etc/cron.d` directory, either. When using Webmin, however, you do not have to bother about these differences as it knows about the paths used by other UNIX variants and displays all Cron jobs using the same interface, no matter which file they are stored in.

10.9 The Scheduled Commands Module

At jobs (called Scheduled Commands by Webmin) are similar to Cron jobs, but instead of executing repeatedly on a schedule, they run only once at a specified date and time. Unlike Cron jobs, they can be configured to execute in a specific directory instead of the user's home directory. Scheduled commands also keep track of the environment variables that were set when they were created, and make them available to the command when it runs.

Normally the `at` command is used to create *At* jobs, the `atq` command is used to list them, and the `atrm` command is used to remove them. On Linux, the directory `/var/spool/at` is used to store jobs—one per file. The daemon process `atd`, which runs all the time in the background, checks these files and runs them at the appropriate times. If the *At* daemon is not running, no commands will be run, which may be the case if it is not configured to start in the Bootup and Shutdown module. After a job is run, it is automatically deleted, as it is no longer needed.

The Webmin module for creating and deleting *At* jobs is called Scheduled Commands, and can be found under the System category. When you enter it, the main page will display a list of commands that are waiting to be run (assuming there are any), and a form for adding a new command. Figure 10.3 shows an example.

Any of the commands shown on the main page can be viewed in more detail by clicking on its **Job ID**. This will take you to a page that shows the full shell script that will be run when the command executes, including all environment variables. For this page, you can cancel the command before it gets a chance to run by clicking the **Cancel this command** button.

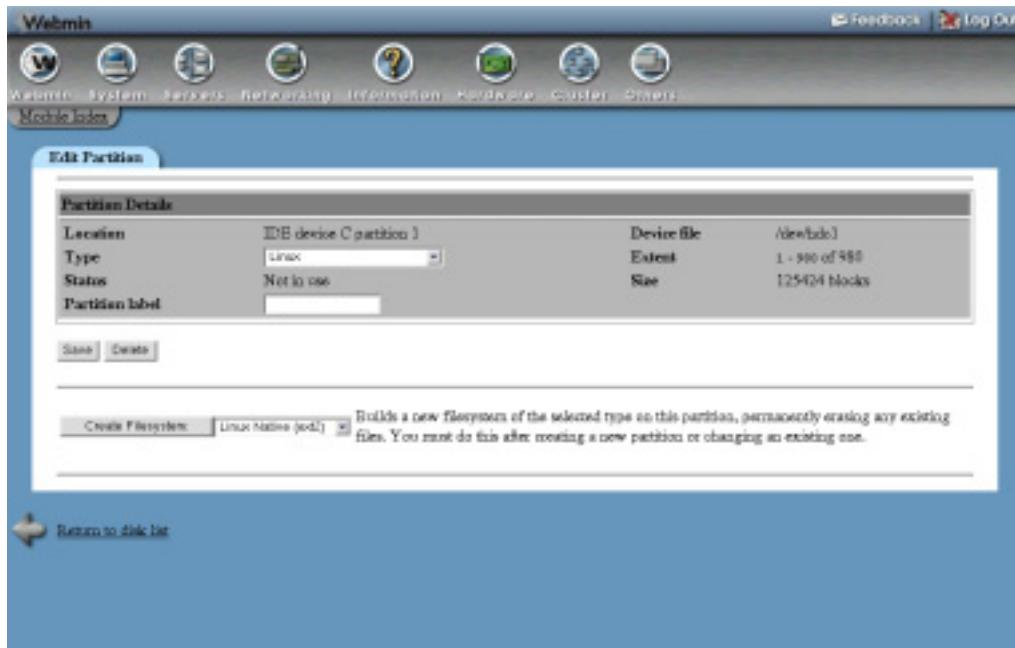


Figure 10.3 The Scheduled Commands module.

10.10 Creating a New Scheduled Command

A new command that executes at the time and as the user of your choice, can be created by following these steps:

1. Enter the name of the user that you want the command to run as into the **Run as user** field on the main page of the module in the **New scheduled command** form.
2. Fill in the **Run on date** and **Run at time** fields with the date and time on which the command is to run.
3. Set the **Run in directory** field to whatever directory in which you want the command to run.
4. In the **Commands to execute** text box, enter as many shell commands as you want—one per line.
5. When done, click the **Create** button. The page will be refreshed and your new command will appear on the list at the top of the page.

Scheduled commands created from within Webmin will use environment variables set by Webmin itself, which are not be the same as the variables that would have been set if the command was created by its owner at the shell prompt.

10.11 Summary

Cron and At are the two services used by UNIX systems to schedule tasks to be run in the future. This chapter has explained how they work, what the differences between them are, and how to use Webmin to configure them. It has also covered the use of Webmin's access control features to restrict access to the modules for managing these services, and the limitations of this kind of access control.

Process Management

This chapter explains how to manage running processes on your system using Webmin.

11.1 Introduction to Processes

Every program, server, or command running on a Linux system is a *process*. At any time, there are dozens of processes running on your system, some for programs that you are interacting with graphically, some for commands that you have started at a shell prompt, some for servers running in the background, and some that perform system tasks. Every time you type a command like `ls` or `vi` at the shell prompt, a new process is created, only to exit as soon as its job is done.

Each process is identified by a unique ID known as the PID, or process ID. Each is owned by a single user and is a member of multiple groups, which determine the privileges that the process has. And each has a priority (also known as the nice level), which controls how much CPU time the process can use up on a busy system. Almost every process has a parent, which is the process that started it, and from which it inherits ownership, priority, and other settings.

A process will run until it chooses to exit, or until it is killed by a signal from another process.

11.2 The Running Processes Module

This module can be used to view, kill, re-prioritize and run processes on your system. When you enter it for the first time from the **System** category, the main page will display a tree of processes as shown in Figure 11.1.

The module has several different ways of viewing all the processes on your system, selectable by the **Display** links at the top of the main page. They are:

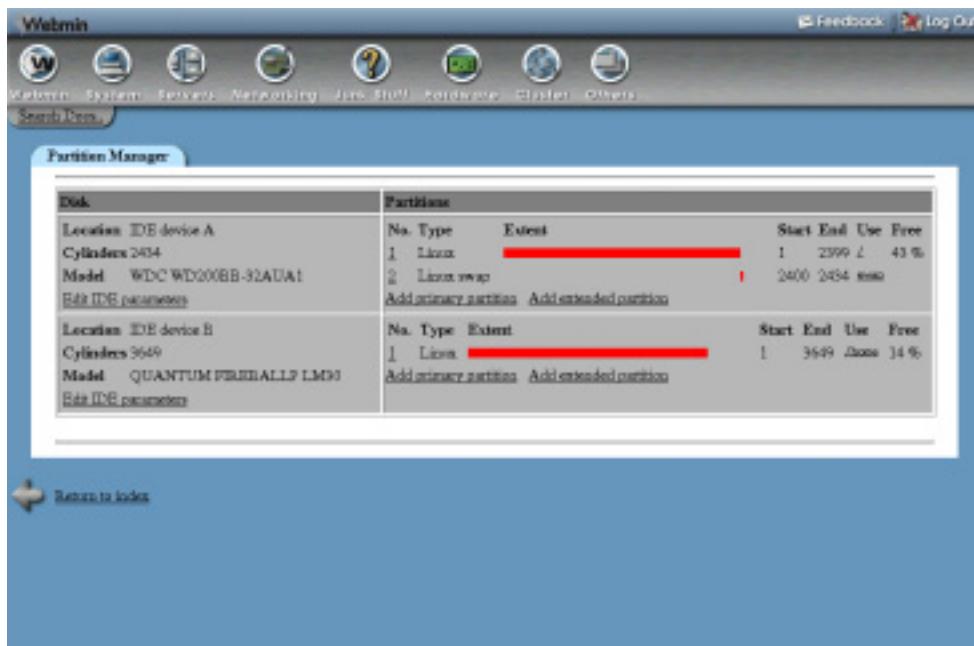


Figure 11.1 The Running Processes module tree display.

PID In this display mode each process is shown indented under its parent, forming a tree of all the processes running on your system. At the top of the tree is the `init` command, which is started by the kernel at boot time and so has no parent.

User This mode groups processes by their owner. It can be useful on systems with many users for seeing at a glance what each user is running.

Memory In this mode, processes are ordered by the amount of memory they are using up, with those using the most memory shown at the top of the page. A process's memory usage is not always indicative of the amount of memory it is really using, because processes often share memory with each other.

In addition, the total and free amount of real and virtual memory on your system is displayed above the process list.

CPU This display mode orders processes by their current CPU usage, with the heaviest user appearing first. Sometimes the Webmin command that generates the page will appear near the top of the list, but it can be safely ignored.

The system load averages will be displayed at the top of the page, to give some idea of how busy the system has been over the last 1, 5, and 10 minutes. An average of 0 means no activity at all, 1 means the CPU is fully utilized, and anything above 1 means that there are more processes wanting to run than the system has CPU time for.

The **Search** and **Run** options are for searching for processes and running new ones, respectively. See the following sections for more details.

11.3 Viewing, Killing, or Reprioritizing a Process

You can see the full details of any running process by clicking on its **Process ID** column entry in any of the displays on the main page. This will take you to the process information page, shown in Figure 11.2.

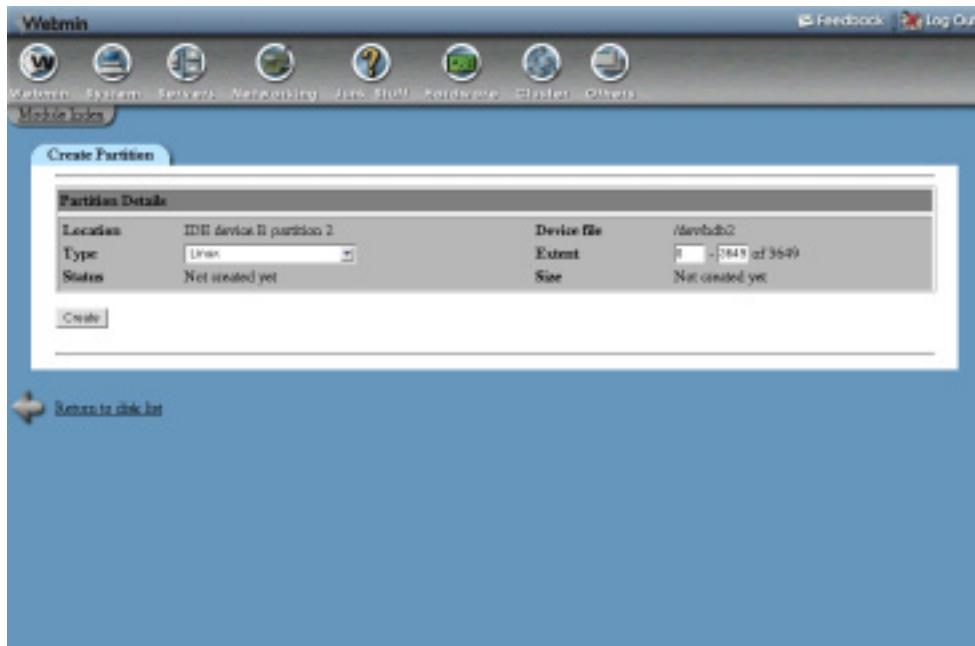


Figure 11.2 Detailed information on a process.

The page displays all available information about the process, including its full command line, parent command, and any sub-processes. You can go to the information page for the parent by clicking on its command, or to the page on any of the sub-processes by clicking on its process ID. A list of files that the process has open and network connections that it is currently using can be viewed by clicking the **Files and Connections** button.

The process can be stopped using a **TERM** signal by clicking the **Terminate Process** button. Because this can be ignored by some commands, the **Kill Process** button can be used to send a **KILL** signal if the termination fails. Unless the process is hung inside a kernel system call, killing it is guaranteed to succeed.

Other signals can be sent by selecting the type of signal from the list next to the **Send Signal** button before pressing it. Some of the more useful signals are:

HUP For many server processes, this signal will cause them to reread their configuration files.

STOP Suspends the process until a **CONT** signal is received.

CONT Resumes a process that has been suspended by a **STOP** signal.

The information page can also be used to change the nice level of a running process, giving it a higher or lower priority. To change a process's priority, select a new level from the **Nice level** list, and then click the **Change** button. Lower levels mean higher priorities, so a process with a nice level of -10 will get more CPU time than one with level 5.

On a system with multiple users, long-running processes that take up a lot of CPU time should be given a higher nice level so that they do not slow down processes that are interacting with users. Alternately, you can speed up a process at the expense of others by giving it a lower nice level. You should be careful when setting an extremely low level (such as -20), as all other processes may become starved of CPU time, making the system unresponsive.

11.4 Searching for Processes

If you have a large number of processes running on your system and want to find one or more to kill or view, the Running Processes module's search feature makes it easy. To find processes, follow these steps:

1. On the main page of the module, click on the **Search** display mode link. This will take you to a search form as shown in Figure 11.3.
2. The form shows several different criteria for finding processes, of which you can choose one by selecting the radio button next to it. The criteria are:

Owned by Processes owned by the user whose name you enter next to this option will be found.

Matching Finds processes whose command or arguments contains the text that you enter next to this option.

Using more than Finds processes using more than the specified percentage of CPU time.

Using filesystem Processes whose current directory is on the chosen filesystem or are accessing any file on it will be found. Useful if you cannot unmount a filesystem because it is busy.

Using file Finds processes that have the entered file open for reading or writing. If you enter a directory, any process that has it as its current directory will be found.

Using port Finds processes that are sending, receiving, or listening for network traffic on the entered port using the chosen protocol. Useful if you know the port number a server is listening on, and want to find the server process.

3. To filter the Webmin search processes from the results, select the Ignore search processes in result option. This can be useful when searching by CPU usage, as the Webmin processes use up a lot of CPU time.
4. After you have selected the search criteria, click the **Search** button. Any matching processes will be displayed below the form.
5. If you want to see additional information about a process, change its priority, or send it alone a signal, click on its **Process ID** in the results.
6. To kill all matching processes, click the **Terminate Processes** or **Kill Processes** button. You can also send any signal to all processes by selecting it from the list next to the **Send Signal** button. A page will be displayed listing each process ID and whether it was signaled or killed successfully.

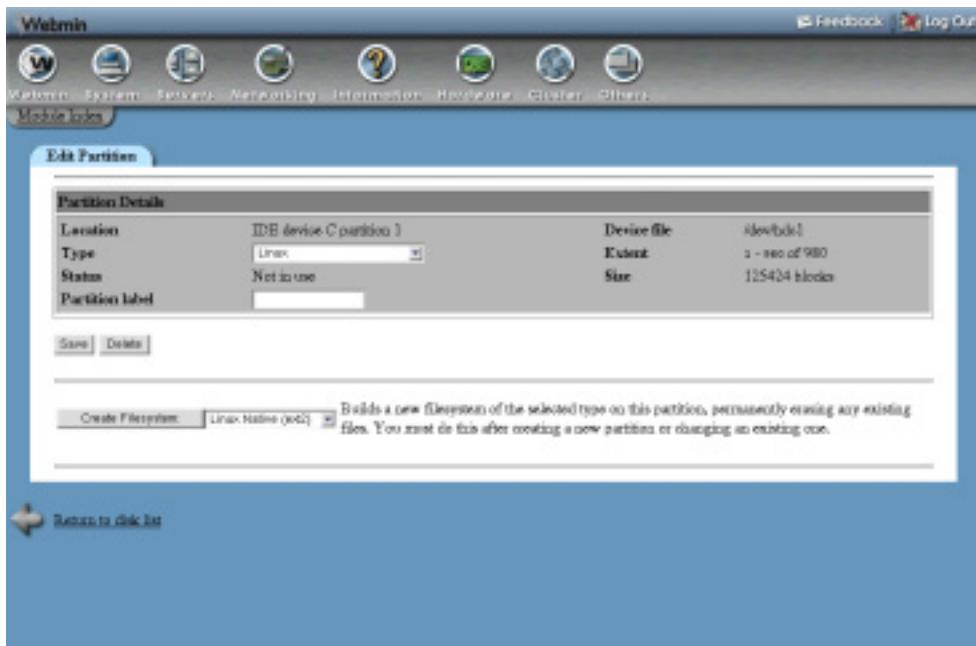


Figure 11.3 The process search form.

11.5 Running a Process

The module can also be used to run simple commands, either in the foreground so that their output is displayed, or in the background as daemons. This can be useful if you just want to run a command without having to log in via telnet or SSH (or if a firewall is preventing a telnet or SSH login). The steps to follow are:

1. On the main page of the module, click on the **Run** link next to the display mode options. This will take you to the form for starting a new process.
2. Enter the command that you want to run into the **Command to run** field. Shell operators and special characters like ;, <, >, and & can be used.
3. If the command is something that will take a long time to run, you can set the **Run mode** option to **Run in background** to have Webmin automatically put it in the background. However, if you want to see the output from the command, leave the option set to **Wait until complete**.
4. Enter any input that you want fed to the command into the **Input to command** field.
5. Click the **Run** button to run it. If the **Wait until complete** option was selected, any output from the command will be displayed.

11.6 Module Access Control Options

By default, any Webmin user with access to this module can manage all processes running on the system, as though he were logged in as `root`. However, using the Webmin Users module, you can limit a user's access so that he can only kill or re-nice processes owned by a particular

UNIX user. It is also possible to restrict a user to read-only mode, allowing him only to see processes but not change them in any way or start new ones.

You should read Chapter 52 first to learn more about module access control and how to grant a user access to the Running Processes module. Once you have done that, the steps to follow to edit a Webmin user's access to this module are:

1. In the Webmin Users module, click on **Running Processes** next to the name of the user or group that you want to restrict.
2. Change the **Can edit module configuration?** field to **No**.
3. To give the Webmin user access to only those processes owned by a particular UNIX user, enter the username into the **Manage processes as user** field. If the UNIX and Webmin users have the same name, you can select **Current Webmin user** instead. This can be useful when setting up module access control for a group in which you want each member to be able to manage only his own processes.
4. To put the user into read-only mode, set the **Can kill and re-nice processes?** and **Can run commands?** fields to **No**. If this is done, it doesn't really matter what username you enter in Step 3 because no process management can be done.
5. Click the **Save** button to have your changes activated.

To restrict the processes that a Webmin user can manage, the module code simply switches to run as the UNIX user specified in Step 3. Because a UNIX user cannot kill or re-prioritize any process that he does not own, switching users like this causes the operating system to automatically enforce process access control for Webmin.

11.7 Other Operating Systems

Because processes exist on all versions of UNIX with almost identical attributes, this module appears almost exactly the same on all supported operating systems. The only minor differences are:

- When viewing detailed information about a process, different information may be available on other operating systems. The range of nice levels may also be different, but lower levels still mean a higher priority and vice-versa.
- When searching for a process, the **Using filesystem**, **Using file**, or **Using port** criteria may not be available. These options depend on the fuser and lsof commands that are not available for or installed by default on all systems.

11.8 Summary

After reading this chapter you should have a good understanding of what processes are, and their importance on a UNIX system. You should also understand how to use Webmin to view processes running on your system, search for those matching some criteria, and kill, signal, or re-prioritize one or more of them. Finally, you should know how to configure the module to restrict the capabilities of certain Webmin users, if required for your system.

Software Packages

This chapter covers the installation and management of software on your system using packages. It also covers the differences between the various UNIX package formats, such as RPM, DPKG, and Solaris.

12.1 Introduction to Packages

All Linux systems use some kind of software packaging system to simplify the process of installing and removing programs. A package is a collection of commands, configuration files, man pages, shared libraries, and other files that are associated with a single program like Apache or Sendmail, combined into a single package file. When it is installed, the package system extracts all the component files and places them in the correct locations on your system. Because the system knows which package every file came from, when you want to remove a package it knows exactly which files to delete.

On almost all versions of Linux, packages generally contain compiled programs that will only work on the CPU architecture that they were compiled for. Because Linux supports many different CPU types (x86, Alpha, and IA64 to name a few), some programs have packages compiled for several different CPUs. A package can only be installed on a system with the right CPU architecture—unless it is architecture-independent, in which case it will install on any system type. Programs written in languages like Perl (such as Webmin) or packages that contain only documentation are usually CPU-independent.

When a Linux distribution is installed, almost every file that is placed on the hard disk is a member of one of the distribution's packages. This makes it easy to remove unwanted software that was installed by default, or add additional software from the distribution CD or website.

Because some programs depend on other programs to operate, packages can have dependencies as well. Certain packages may fail to install unless you have installed another package first, and some packages may not be removable if others depend upon them. This system of dependencies protects the user from installing software that will not work due to a missing shared library or command.

Because the package system knows exactly which files are in each package, it can use that information to validate the files after installation. All package systems also keep track of the MD5 checksum for each file, so that any manual modifications to files in a package can be detected. This can be very useful for detecting unauthorized modifications, such as by an attacker who has cracked your system and replaced important commands like `ls` and `find` with modified versions. Unfortunately, on Linux there is more than one package system. The most common is RPM, which stands for Red Hat Package Manager. It is used by Red Hat, Caldera, SuSE, Mandrake, MSC, and a few other Linux distributions. It works well, and there is more software available in RPM format than any other package system. Installation, querying, and deletion of RPM packages is done using the `rpm` shell command.

The biggest competitor to RPM is Debian's DPKG package format. It is technically superior in many ways, particularly when it comes to dependencies—however, only Debian and a few other distributions use it. The `dpkg` and `deselect` commands are used at the shell prompt to manage Debian packages.

Another packaging system is Gentoo's Emerge, which is only available on Gentoo Linux. The biggest difference between Emerge and other package systems is that almost all packages contain source code, which is compiled when the package is installed. All Gentoo package installation and management is done using the `emerge` command.

Even though these package systems are internally different and use incompatible file formats, they all offer basically the same features. All allow multiple files related to the same program to be combined into one package file for easy installation and removal, and all support dependencies. Unfortunately, once you have chosen your Linux distribution it is very difficult to change to another packaging system, so you are stuck with what the distribution uses.

On most distributions that use RPM, packages are either installed from a distribution CD or downloaded from various sites on the Internet. Debian Linux, however, includes a command called `apt-get` that can automatically download and install packages from a repository run by the distribution maintainers. If the package depends on others that are not yet installed on your system, they will be automatically downloaded and installed as well. Because all packages in the repository are created and maintained by the same people, incompatibilities between them are reduced and dependencies easily resolved. The repository also contains a package for almost every free software program that you might want to install, so there is no need to search the Internet for the package that you want.

The Debian repository can also be used to update all the packages on your system to the latest version. Because new versions of packages come out frequently (especially when using the `unstable` or `testing` Debian releases), an update is an easy way of ensuring that you are running the latest version of everything. This can take a long time if you do not have a fast connection to the Internet though, as many new packages may be downloaded for each update.

Gentoo Linux's Emerge system also has a repository from which packages can be automatically downloaded and installed using the `emerge` command. Like Debian's `apt-get`, it automatically downloads and installs packages needed to fulfill dependencies.

Red Hat systems also have access to a package repository as part of the Red Hat Network. This allows updated packages to be selected on the Red Hat website and installed automatically or on request on multiple systems. Unlike the Debian and Gentoo repositories, it is not generally used for installing new packages.

12.2 The Software Packages Module

The Software Packages module provides a consistent interface for installing, searching, and removing packages, independent of the actual packaging system being used. Its icon can be found under the **System** category, and clicking on it will take you to the main page shown in Figure 12.1.

Depending on your Linux distribution, the page may look slightly different—additional buttons and fields for installing from a repository may be visible. However, the top section for finding packages, the middle section for installing a package, and the lower section for identifying a file will always be there.

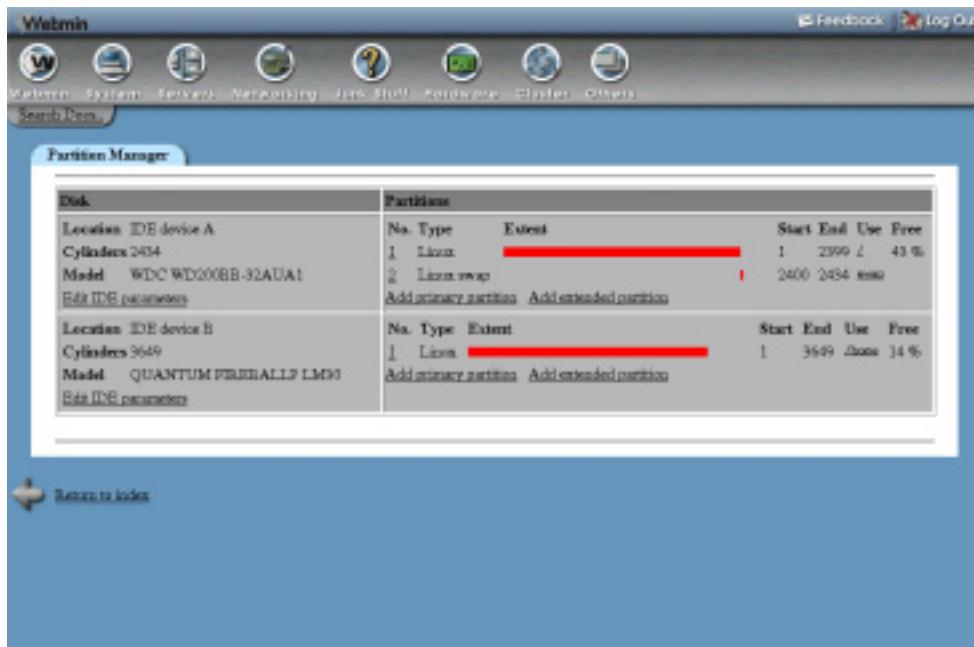


Figure 12.1 The Software Packages module main page.

12.3 Installing a New Package

Before you can install a new program using this module, you first have to locate a package file for it that is in the correct format. For RPM-based distributions like Red Hat, the best places to look are the distribution CDs or the [rpmpfind.net](#) website. If you are using Debian Linux, it is best to try installing from the APT repository as it contains almost all available packages. Either way, the steps for installing a package are similar:

1. On the main page of the module, scroll down to the **Install a New Package** form, which will be used to select the package and start the install process.

If the package file is on the system running Webmin, select the **From local file** option and enter the full path to the package file. If your system uses RPM packages, you can

enter a directory containing multiple *.rpm* files or a wildcard like */tmp/*.rpm* as well. This can be used to install several packages at once.

If the package is on the computer your browser is running on, select the **From uploaded file** option and click on the **Browse** button to select the package file. If you are running your browser at the console of your Webmin system, there is no difference between this option and the previous one.

If the package is on a website somewhere, select the **From ftp or http URL** option and enter or paste the URL into the text field next to it. Webmin will do the download for you before starting to install. If your system uses RPM packages and you have the *rpmpfind* command installed, the **Search rpmfind.net** button next to the URL field can be clicked to pop up a window for searching the RPM database at *rpmfind.net/*.

If running Debian Linux, you can select the **Package from APT** option and enter the package name into the text field next to it. Click the **Search APT** button to find the package name if you don't know exactly what it is called.

If running Red Hat Linux, the **Package from Red Hat Network** option can be used to install one of the packages that you have available for downloading. The **Search RHN** button can be used to display all those that are available.

If you are running Gentoo Linux, the **From Portage repository** option and **Search** buttons can be used to install from the repository. In fact, very few Gentoo packages can be found outside the repository.

2. Once the package source has been entered, click the **Install** button.

If you chose to install from a repository (such as APT, Red Hat Network, or Portage), the download and installation process will start immediately. Webmin will display output from the install command, and if successful, a list of packages that were installed. No other steps are necessary to complete the install process.

If any other install source was chosen, the package will be downloaded if necessary and the installation options form displayed.

3. The installation options available differ depending on your package system, but the defaults will work fine for upgrading or installing a package with no dependency problems. RPM-based systems have several options, the most useful of which are:

Ignore dependencies? If a package is failing to install due to dependency errors that you know are incorrect, set this option to **Yes**. It can also be useful if you are going to install packages to solve the dependency problems later.

Replace new version with old? If you want to downgrade a package to an older version, this option must be set to **Yes**.

Overwrite files? If a package cannot be installed due to conflicts with files from another package, enable this option.

4. When you are done selecting install options, click the **Install** button. If everything goes well, a page showing the details of the new package and all the files that it contains will be displayed. However, if the install fails, an error message explaining why will be displayed. In that case, you can use the browser's back button to return to the install options form and try again with different choices.

12.4 Finding and Removing a Package

A typical Linux system has hundreds of installed packages, most of which were added as part of the distribution install process. Because there are so many, it is difficult to simply browse through them to find one that you want to remove or view the details of. To find a package or packages, follow these steps:

1. On the main page of the module, enter a search keyword into the **Search For Package** field. This will be matched against the names and descriptions of all packages, so you can enter something like *apache* to find all that are related to Apache.
2. Click the **Search For Package** button, which will either display a list of all matching packages, take you to the details of the package if only one is found, or show an error message if none were found. If a list appears, click on one of the package names to see its full details.
3. The package details page (shown in Figure 12.2) will display all available information, including a full description. If you want to see all the files that it contains, click the **List Files** button. This will take you to a page showing the path, type, owner and group, and validation status for each file. The status is particularly useful, as it allows you to see if a file has been changed manually since the package was installed.

Packages can also be browsed manually by clicking on the **Package Tree** button on the main page. On most operating systems, each package is a member of a class such as **Development** or **Administration/Networking**. The package tree page uses this class information to display all installed packages in a hierarchy, much like a directory tree. You can open classes by clicking on their folder icons until you get to the package level. Clicking on a package icon will take you to the same details page as described in the steps above.

If you know the name of a command or file and want to find the package that it belongs to, the **Identify a File** form on the main page can be used. Enter either a full path (like */etc/rc.d/init.d/httpd*) or a command (like *apachectl*) into the **Search For** field, and hit the button. If the file or command is known to the package system, information on it will be displayed along with a list of packages that it belongs to. Clicking on one of the package names will take you to the information page described above.

Once a package has been found by searching or browsing the tree, you can delete it from your system by following these steps:

1. On the package details page, click the **Uninstall** button. This will take you to a confirmation page showing the number of files in the package and the amount of disk space that they occupy.
2. If using the RPM packaging system, the **Ignore dependencies?** option can be set to **Yes** to force an uninstall even if some other packages depend upon the one being removed.
3. Click the **Delete** button to remove the package. If something goes wrong, an error message will be displayed. If successful, the browser will return to the module's main page or to the package search results list, if you found the package using a search.

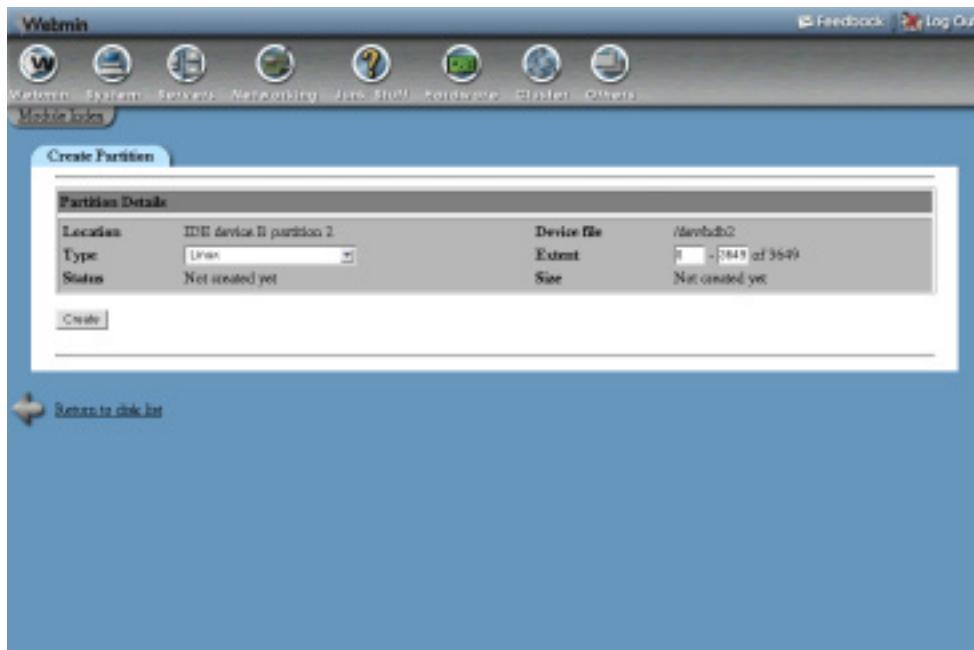


Figure 12.2 The package details page.

12.5 Updating on Debian Linux

If you are running Debian Linux, at the bottom of the main page of the module there will be a form headed **Upgrade All Packages**. This form has three options, which are:

Resynchronize package list If this option is set to **Yes**, the Debian package repository will be queried to retrieve the latest list of packages available for download. This should be done before any upgrade, so that your system knows which URLs to download from when installing packages from the APT repository.

The actual command used to synchronize the package list is `apt-get update`.

Perform distribution upgrade When this option is set to **Yes**, your Debian system will be upgraded to the latest distribution release when the form is submitted. With the default **No** selection, it will simply be updated so that all packages installed are the latest version. Unless you have a fast network connection and really want to upgrade, it is advisable to leave this option set to **No**.

When **Yes** is selected, the command `apt-get upgrade-dist` will be run. For **No**, `apt-get upgrade` will be used instead.

Only show which packages would be upgraded If set to **Yes**, nothing will actually be installed when the form is submitted—instead, a list of packages that would be updated or installed will be displayed. This can be useful if you want to see exactly what would happen when doing an update before going ahead for real.

After you have made your choices, click the **Upgrade Now** button. Webmin will run the appropriate `apt-get` commands and display their output, so that you can see which packages are downloaded and updated.

12.6 Updating on Red Hat Linux

Red Hat offers a service to users of its Linux distribution, called the Red Hat Network. One of its features allows you to have updated RPM packages automatically installed on your system, to fix bugs or security holes that are found in the packages supplied with the distribution. If you are running Red Hat Linux, there will be a form at the bottom of the main page under the heading **Red Hat Network Options** that you can use to configure the automatic installation of updated packages. Before it can be used, you must have signed up with the Red Hat Network and registered the system you are running Webmin on.

The form actually serves two purposes—changing the settings for the update daemon that periodically checks for new packages and forcing an immediate update. The fields on the form are:

Automatically check for updates? If this option is set to **Yes**, the `rhnscd` daemon that checks for updates will be configured to start at boot time. It will also be started immediately when the **Save and Apply** button is clicked, if it is not currently running. Setting it to **No** will stop the daemon and prevent it from being started at boot time.

Checking interval When the automatic update daemon is enabled, the number of minutes between checks for new packages is determined by this option.

Proxy server URL for downloading If your system cannot connect directly to the Red Hat website, you will need to set this option to the URL of a web proxy server. It must be formatted like this: `http://proxy.company.com:8000/`.

Skip packages matching This option is for entering a list of patterns for package names that you do not want automatically updated. By default it prevents kernel updates from being automatically installed.

The **Save and Apply** button will save your settings and start or stop the `rhnscd` daemon as necessary. The **Save and Check Now** button will do the same thing, but will also run the `up2date -u` command to immediately check for and download new packages. All output from the command will be displayed so that you can see which packages are being updated.

12.7 Other Operating Systems

Linux is not the only version of UNIX that uses packages to simplify the process of installing and removing software. The operating systems listed below can also use the Software Packages module, with an almost identical user interface. However, each has its own packaging format that is incompatible with Linux or any other variety of UNIX.

The differences between each UNIX's package system and RPM are explained below:

Sun Solaris, SCO OpenServer, and SCO UNIXWare

- All of these operating systems use the same basic System V package format, but packages from one cannot be installed on any of the others.
- Package files are usually named *something.pkg* or *something.pkg.gz*. If a package file is compressed, Webmin will uncompress it automatically.
- Files can contain multiple packages, all of which will be installed when using Webmin.
- No package repository or search service exists for System V packages.
- Directories like **/usr/bin** are often shared between multiple system packages.

FreeBSD, NetBSD, and OpenBSD

- Package files have names like *something.tgz*, and are actually just specially formatted *tar* files.
- Webmin does not support any repository for BSD packages.

HP/UX

- HP/UX uses its own unique Depot package format.
- Package files are usually named like *something.depot* or *something.depot.gz*. If a package is compressed, Webmin will automatically uncompress it for you.
- Webmin does not support any repository for HP/UX packages.

12.8 Summary

Package management is one of the most useful features of Linux and the other UNIX variants that support it. This chapter has explained what packages are, what kinds of package management systems exist, and how you can use Webmin to install and manage packages on your system. It also covered other package-related services, such as automatic updates and repositories.

System Logs

In this chapter, the UNIX `syslog` service that controls where logs are written to is explained, and the Webmin module for configuring it is documented.

13.1 Introduction to Logging

Many Linux servers and daemons generate log messages for errors, warnings, requests, and diagnostic information. In most cases, these logs are not written directly to a file—instead, they are passed to the UNIX logging program `syslog` which decides what to do with each log message. Logs can be written to a file, sent to another server, passed to another program via a pipe, or even broadcast to all users logged into the system. Different types of messages from different servers can be logged using each of these methods.

Normally logs are written to files in the `/var/log` directory. On most Linux distributions the file `/var/log/messages` contains general information, error and warning messages, the file `/var/log/mail` records incoming and outgoing mail, and `/var/log/secure` records successful and failed logins. However, your system may have a totally different `syslog` configuration and so use different logfiles.

Each log message that is sent to `syslog` has three attributes—the program that it comes from, a *facility*, and a *priority*. The facility classifies the message, indicating which part of the system it is coming from. Facilities that are recognized on Linux are seen in Table 13.1.

The priority or log level associated with each message indicates how serious it is. Many servers generate messages with low priorities that contain only diagnostic or debugging information, which can safely be ignored. However, messages with higher priorities indicate a serious problem with a server or possibly the entire system. The recognized priorities on Linux (in order from least to most serious) are seen in Table 13.2.

Table 13.1 System Logging Facilities and Their Sources

auth or authpriv	All messages related to successful or failed authentication attempts will use this facility.
cron	Used for log messages from the Cron and At daemons.
daemon	Used for messages from other daemons, such as the NFS, NIS, and DHCP servers.
kern	For error, warning, and informational log messages that come from the kernel.
lpr	For messages from the printer server and print commands.
mail	For email delivery logs, and error messages from Sendmail or Postfix.
news	For messages from news servers like INN.
syslog	Used for log messages generated by the <code>syslog</code> daemon itself.
user	For generic user-level messages. Not often used.
uucp	For messages from the UUCP (UNIX to UNIX Copy) server programs, which are hardly ever used anymore.
local0 to local7	These facilities are reserved for local use, such as by a server that can be configured to use a different facility.

Table 13.2 System Logging Priorities and Their Meanings

debug	Debugging information only, which can be safely ignored.
info	An information message indicating that something has occurred, but nothing serious.
notice	Indicates a normal but significant event.
warning	A warning about some potential problem.
err	An error message indicating that something has failed.

Table 13.2 System Logging Priorities and Their Meanings (Continued)

crit	Indicates a critical problem of some kind.
alert	An extremely serious problem that must be looked into immediately.
emerg	Indicates imminent or actual system failure.

The file `/etc/syslog.conf` contains the `syslog` configuration that controls which messages are logged to which files and destinations. Webmin reads and modifies this file directly to change your system's logging settings, and reads from the files in `/var/log` to display log messages.

Not all logs generated by all programs are controlled by `syslog`. For example, the Apache Web server writes directly to a log file that records every HTTP request that it receives. Other programs like Squid and Qmail also have their own private log files that are not under the control of `syslog` and so cannot be configured using the System Logs Webmin module. Some of these servers can be configured to log via `syslog`, but this is never the default and is usually a bad idea for programs that generate large numbers of log messages, such as Apache. See Chapters 29, 38, and 44 for more information on configuring logging in these servers.

13.2 The System Logs Module

If you want to view log files on your system and configure where log messages are recorded, the System Logs module under the System category is the place to go. The main page of the module lists all files and other destinations that `syslog` is currently logging to, as shown in Figure 13.1. For each log destination, its active status and the facilities and priorities that are logged to it are displayed.

Even if you don't want to change existing logging settings, you can use the module to view a log file by clicking on its **View** link. This will take you to a page showing the last 20 lines of the file, with a **Refresh** button at the bottom to reload the page or increase the number of lines displayed. You can also limit the display to only certain types of log messages with the **Only show lines with text** field. Only logs written to normal files can be viewed—those sent to another server, to users, to a named pipe, or to a device file cannot be read by Webmin.

13.3 Adding a New Log File

Because the messages written to each log destination have no effect on other destinations, you can add a new log file without affecting any of the existing ones. This can be useful if there is some information which you want to see but which is not currently being recorded, or if you want to separate out messages of a particular facility or priority into a different file from the one that they are currently being logged to.

To add a new log file or destination, the steps to follow are:

1. On the main page of the System Logs module, click on the **Add a new system log** link above or below the list of existing log files. This will take you to the form shown in Figure 13.2 for entering the details of the new log destination.

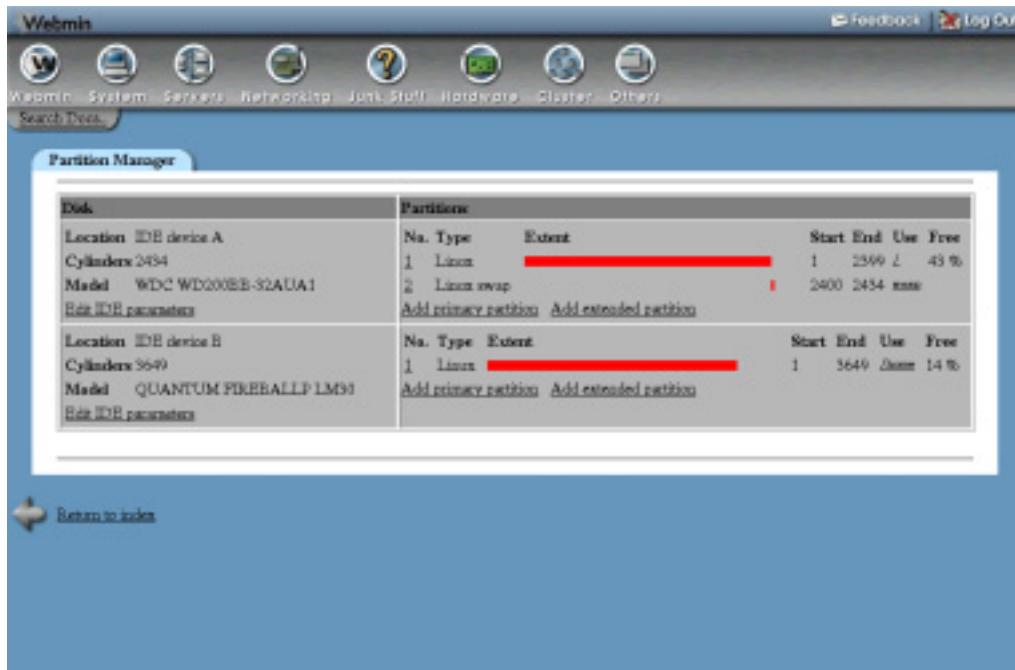


Figure 13.1 The System Logs module.

2. Select one of the five choices in the **Log to** field, which controls where messages are written to. The choices are:

File If this option is selected, you must enter into the text field the name of a file to write logs to. Log lines will be appended to the file, which will be created if it does not exist. To ensure that syslog forces each line to be written to disk after adding it, select the **Sync after each message?** option. Unless you are trying to reduce hard disk activity on your system (such as on a laptop), it is wise to leave this option selected.

It is possible to create more than one log that writes to the same file. This can be done safely without worrying that messages from one will overwrite another.

Named pipe A named pipe is a special file that can be written to by one program and read by another. If you want log messages to be written to a pipe, first create it and then enter its path into the field next to this option.

Syslog server on This option can be used to pass some or all of the log messages from your system to another server, assuming it is running `syslog` as well. If selected, the hostname or IP address of the remote server must be entered into the text field next to the option. Unlike local log files, logs written to a remote server are safe from attackers who break into your system.

Local user If this option is selected, log messages will be broadcast to any of the users listed in the text box next to the option. Users must be logged in via SSH, telnet, or at the console to receive log messages.

All logged-in users This is like the previous option, but messages will be sent to all logged-in users. In order to avoid annoying people, this option should only be used for logging really serious errors.

3. The **Logging active?** field determines whether this log is enabled or not. If set to **No**, the `syslog.conf` entry for the log will be commented out and nothing will be sent to the chosen destination.
4. The **Message types to log** section controls which messages are written to the log destination. It is composed of two parts: **Facilities** and **Priorities**. A message will only be logged if it matches both the selected facilities and the selected priorities.
For the **Facilities**, you can either select a single facility from the menu, select the **All** option to include all of them, or enter a list of facilities separated by spaces into the **Many** text field.
5. For the **Priorities**, you can select **None** to indicate that no messages of the select facilities will be logged, select **All** to log messages of any priority, or choose one of the range options from the menu (**At or above**, **Exactly**, **Below** or **All except**) and choose a priority from the final menu. This last option limits logging to messages of one or more priorities depending on your range type and priority selection.

When creating a new log, you can select only one set of facilities and one range of priorities. However, after saving, if you re-edit the log you can add an additional row specifying facilities and priorities so that more than one type of message is logged. It is even possible to use the **None** option under **Priorities** to exclude some facilities that were included by a previous row.

6. When done making your selections on the form, click the **Create** button. As long as there are no errors, you will be returned to the main page of the module.
7. Click the **Apply Changes** button to make your new log destination active.

13.4 Editing or Deleting a Log File

Any of the existing logs shown on the main page of the module can be edited or deleted using Webmin. However, you should be careful when changing destinations that were included in the system's default configuration, as important messages may no longer be logged. Even changing the filename that logs are written to could cause problems, as many Linux distributions include software to automatically truncate the standard log files to prevent them from taking up too much disk space.

To change a log, the steps to follow are:

1. On the main page of the module, click on the destination of the log that you want to edit. This will take you to an editing form that is almost identical to the creation form shown in Figure 13.2.
2. Change any of the existing settings, such as the destination type, log file, or active status. You can also change which facilities and priorities are logged by adding to or editing the rows in the **Message types to log** section. There will always be one blank row for selecting new facilities and a new priority range, as explained in Section 13.3 “Adding a New Log File”.

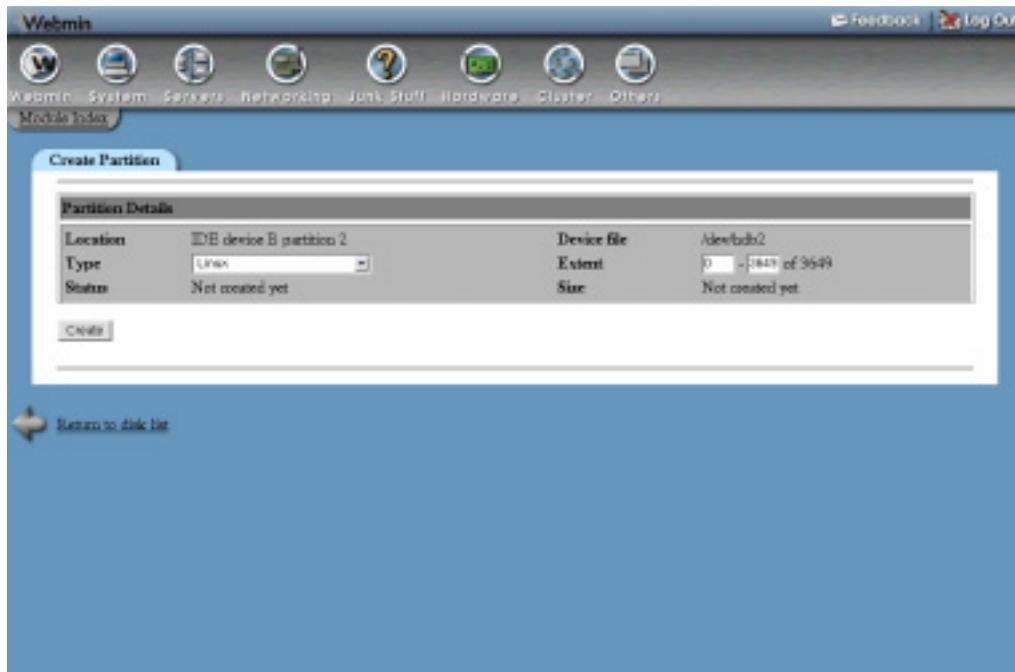


Figure 13.2 The new log destination form.

3. When done, click the **Save** button. As long as you have made no errors in the form, your browser will return to the module's main page.
4. Click the **Apply Changes** button to make your changes active.

To delete a log, follow these steps:

1. On the main page of the module, click on the destination of the log that you want to delete.
2. Click the **Delete** button at the bottom of the page. This will stop logging to the destination, but it will not delete any log files that have already been written—you can do that manually if you wish.
3. Back on the main page, click the **Apply Changes** button to make the change active.

13.5 Module Access Control

The System Logs module can be restricted so that a Webmin user can use it only to view log files instead of being able to create and edit them. As explained in Chapter 52, you must first create or edit a user who has access to the module. Once that is done, follow these steps to limit him to viewing log files only:

1. In the Webmin Users module, click on System Logs next to the name of the user that you want to restrict.

2. Change the **Can edit module configuration?** option to **No**, so that he will not be able to reconfigure the module to use a fake `syslog.conf` file.
3. Change the field **Can only view logs?** to **Yes**. When this is set, the only thing that the user will be able to do on the module's main page is click on the **View** link next to a log file entry.
4. To limit the log files that the user can view, select **Only listed files and those under listed directories** in the **Can view and configure log files** field and enter a list of filenames into the adjacent text box. This can be useful if some of the logs on your system contain sensitive or secure information.
5. Click the **Save** button to make the changes active.

13.6 Other Operating Systems

Almost all versions of UNIX use `syslog` to control the destinations that log messages are written to, so the System Logs module is available on most operating systems. It has similar capabilities on all systems, so the user interface is generally the same. However, there are some differences, as explained below:

Sun Solaris, Apple MacOS X, HP/UX, SCO UNIXWare, SCO OpenServer, and IBM AIX

- On Solaris, the first time you use the module, Webmin may ask if you want to convert `syslog.conf` from M4 format. Unless you have made manual changes that use M4 macros, this is safe to do.
- Logging to named pipes is not supported.
- There is no option to sync after each write to a log file.
- When selecting the priorities of messages to write to a log, the **At or above**, **Exactly**, **Below**, and **All except range types** are not available. Instead, all messages with priorities at or above the one you select will be logged.

FreeBSD, OpenBSD, and NetBSD

- On FreeBSD, logging to named pipes is not supported.
- On OpenBSD and NetBSD, logs can be sent directly to the input of a command instead of to a named pipe.
- There is no option to sync after each write to a log file.
- When selecting the priorities of messages to write to a log, the **At or above**, **Exactly**, **Below**, and **All except range types** are replaced with `>=`, `=`, `<`, `<>`, which have similar meanings.
- Each log destination can be associated with a program, set using the optional **Only for program** field. If set, only log messages from the entered server or daemon will be written to this log file.

SGI Irix

- Logging to named pipes is not supported. Instead, logs can be sent directly to the input of a command.
- There is no option to sync after each write to a log file.
- Logs can be written to a UNIX domain socket file.
- When selecting the priorities of messages to write to a log, the **At or above**, **Exactly**, **Below**, and **All except** range types are not available. Instead, all message with priorities at or above the one you select will be logged.

If your operating system is not on the list above, then it is not supported by the System Logs module.

13.7 Summary

The log files on a UNIX system contain a wealth of useful information, such as the details of email received, attempted and successful logins by users, hardware error reports from the kernel, and much more. This chapter explains how to configure the files and other destinations that are used for various types of messages, and how to view those files though Webmin. An important part of system administration is keeping a look out for log messages indicating serious errors or potential security violations.

Filesystem Backups

This chapter explains how to backup and restore files with the `dump` command, using Webmin's Filesystem Backup module.

14.1 Introduction to Backups with Dump

There are many ways of backing up a UNIX system—you can just copy files to another directory, use the `tar` command to create an archive file or write to a tape device, or use the `dump` family of commands. Although copying or using `tar` is easier, only `dump` can preserve all file types (such as named pipes and symbolic links) and file information (such as ACLs and attributes). It can do this because it has a more sophisticated knowledge of the underlying filesystem than other backup programs.

Another unique advantage of the `dump` program is its support for backup levels. If you are regularly backing up the same directory, instead of writing all files to the backup device every time, you can choose to save only those files that have changed since the last backup of a lower level. For example, you could do a full backup (level 0) every week, and a much faster partial backup (level 9) each day. The only down side is that if data needed to be restored, the weekly backup and all the daily backups for the week so far would need to be read.

Using `dump` to make backups has some problems that other backup tools do not. The data that it writes to a file or tape device is not compressed, although this is not a problem with most tape drives as they compress data automatically. Another problem is that it cannot backup files mounted via NFS from another server, as it reads directly from the disk, unlike the `tar` and `cp` commands.

14.2 The Filesystem Backup Module

This module allows you to backup directories on your local filesystems, either on demand or on a fixed schedule. The appropriate command for the filesystem type being backed up is used—for

example, `xfsdump` on `xfs` filesystems or `dump` on `ext2` or `ext3`. The module also supports the restoration of backups, either to their original location or to a different path.

When you enter the module from the System category, the main page will display all backups that you currently have configured, as shown in Figure 14.1. Of course, if this is the first time you have used the module there will be none to display.

If Webmin detects that you do not have any of the necessary backup commands installed on your system, an error message will be displayed on the main page instead. All Linux distributions should include a package containing the `dump` program on their CD or website.

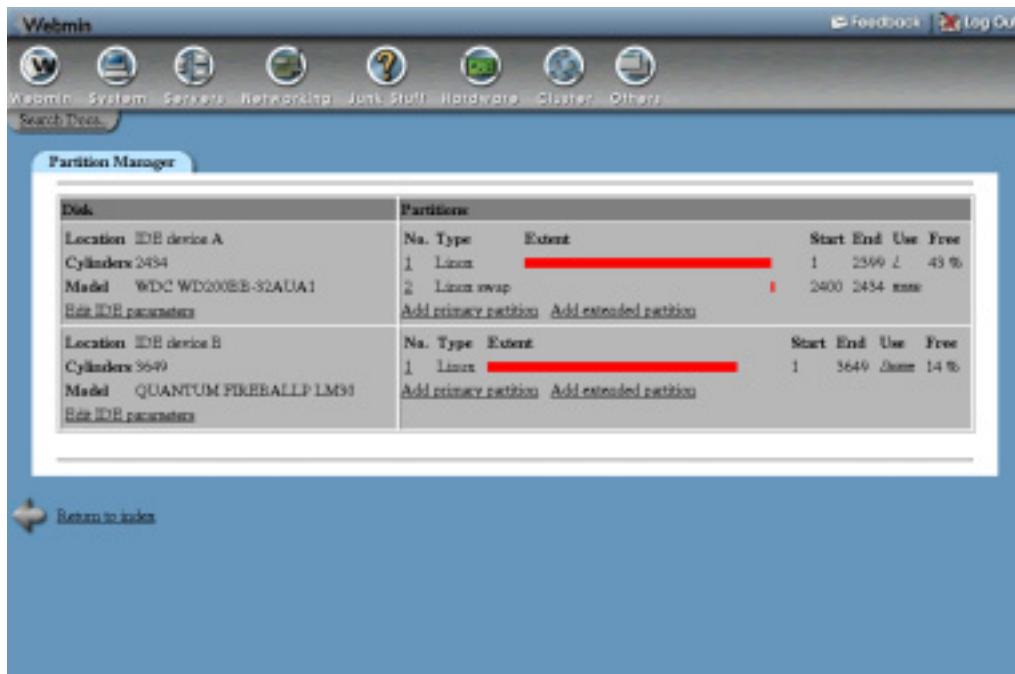


Figure 14.1 The Filesystem Backup module main page.

14.3 Adding a New Backup

If you want to backup a directory, either just occasionally or on a regular schedule, you first need to add a new backup configuration. This specifies a directory to backup, a set of options to use, and the times at which it should be scheduled to run. The steps to follow to create a new configuration are:

1. On the main page of the module, enter the path to the directory that you want to backup into the field next to the **Add a new backup of directory** button. When you click the button, Webmin will determine what type of filesystem the directory is in (`ext2`, `ext3`, or `xfs`) and display a backup creation form with options for that filesystem type. Figure 14.2 shows the form for an `ext2` or `ext3` backup.

2. The path you entered will appear in the **Directory to backup** field. You can still change it if you wish, as long as the new directory that you enter is still on the same filesystem.
3. If backing up to a local file, set the **Backup to** field to the **File or tape device** option and enter the file that you want the backup written to into the text field next to it.

Backing up to a tape drive is similar to writing to a file, but instead of entering a filename into the **File or tape device** field you must enter the device file for the tape drive. For example, `/dev/st0` would be the device file for the first SCSI tape drive on your system.

If backing up to another server, you must select the **Host** option for the **Backup to** field and enter a hostname, remote username, and file or device name on the remote server. The server must have the `shell` service enabled in its Internet Services module, as explained in Chapter 15. An appropriate `.rhosts` file must also be set up for the target user, to allow the `dump` command to connect without needing to supply a password.

4. If your backup is being written to a local file that you do not want to be larger than a certain size, set the **Split across multiple files?** option to **Yes** and enter the maximum size in kilobytes into the **Tape size** field. This can be useful if the backup is going to be later saved to multiple CDs or Zip disks.
5. If you are doing multiple backups at different levels as explained in the introduction, change the **Dump level** field to something other than **Full backup**. However, if you want each backup to contain all files in the source directory, leave it unchanged at level 0.
6. If you are backing up to a tape, it is a good idea to set the **Tape size** field to the number of kilobytes that can fit on your tape. Otherwise, the `dump` command may underestimate the amount of data that can be written and fail to complete the backup.
7. The `chattr` command can be used to mark a file to be skipped when making backups, which can be useful if the directory contains huge and useless files that you would rather not save. However, when doing a level 0 backup such files will be included, unless the **Always exclude marked files?** field is set to **Yes**.
8. If you are familiar with the `dump` command used on your operating system, the **Extra command-line parameters** field can be used to enter extra options to be passed to the program, such as `-A /tmp/archive`. Otherwise, leave it blank.
9. To have commands run before and after the `dump` command is executed, fill in the **Command to run before backup** and **Command to run after backup** fields. These commands will be run as `root` when the backup is made, either as scheduled or manually through Webmin. They can be useful for loading and unloading tapes, or copying files into the backup directory before it is saved.
10. If you want the backup to be run on a regular schedule, set the **Scheduled backup enabled?** option to **Enabled** and select the times and days for it to run from the lists at the bottom of the page. The user interface for date and time selection is exactly the same as the one used by the Scheduled Cron Jobs module, explained in Chapter 10.
11. To have a status report of the scheduled backup emailed to you, enter your email address into the **Email scheduled output to** field.
12. By default, the subject of the backup email will be something like *Dump of /etc*. If you are using this module on multiple systems, you may want to customize the subject line so that the host the email is coming from and the data that has been backed up is more obvi-

ous. To do this, de-select the **Default** option for the **Email message subject** field, and enter a new subject line into the text field next to it.

- Finally, click the **Create** button to save the details of the new backup configuration. If there are no errors in the form, you will be returned to the modules main page.

Alternatively, you can begin a backup immediately by clicking **Create and Backup Now**. This will take you to the page showing its progress, as explained in Section 14.4 “Making a Backup”.

Apart from `ext2` and `ext3`, the only other filesystem type that has a similar backup command is `xfs`. Because its `xfsdump` command has slightly different options, the fields on the new backup form are not quite the same as described above. One important difference is that the **Directory to backup** must be the mount point of a filesystem, not just any directory within it.

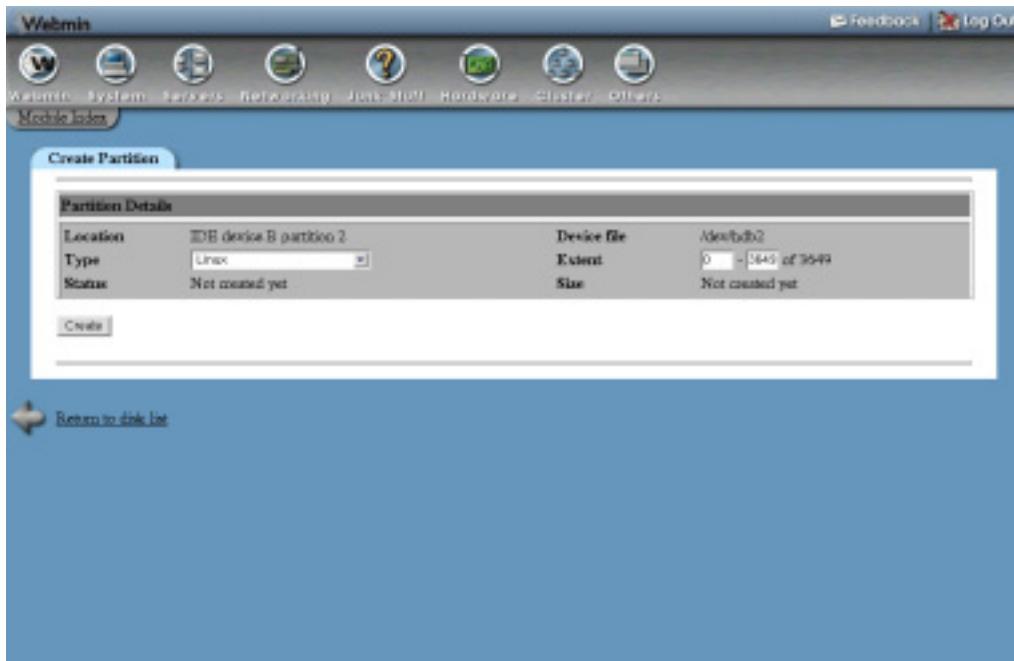


Figure 14.2 The new backup configuration form.

14.4 Making a Backup

Once you have created a backup configuration that appears on the main page of the module, you can use Webmin to manually execute it at any time. To do this, the steps to follow are:

- From the list of configurations on the module’s main page, click on the directory you want to backup. This will take you to a form showing all the details of the backup configuration.
- Click the **Backup Now** button at the bottom of the page. Webmin will execute the appropriate `dump` command and display its output as it writes to the backup file or tape device. The output from any commands that are run before or after the backup will be shown as well.

3. If all goes well, the message **backup complete** will be displayed at the bottom of the page. However, if something goes wrong, **backup failed** will be displayed instead—check the output of the `dump` command to see exactly what the problem was.

One limitation of the Filesystem Backup module is that it cannot create backups that span multiple tapes. Usually the `dump` command would prompt the user to load a new tape when necessary, but that is not possible when it is being run from Webmin.

14.5 Editing or Deleting a Backup

The backup configurations shown on the module’s main page can be edited at any time. To change one, do the following:

1. Click on the directory of the backup configuration that you want to change. This will take you to the editing form, which is similar to the creation form shown in Figure 14.2.
2. Change any of the options, including the directory to backup, destination, or schedule. You cannot change the directory to a path on a different type of filesystem though, as it may have different options or not be supported at all.
3. Click the **Save** button to record your changes, or click **Save and Backup Now** to immediately begin a backup with the new options.

To delete an existing backup configuration, the steps to follow are:

1. On the main page, click on the directory of the backup configuration that you want to delete, which will take you to the editing form.
2. Click the **Delete** button at the bottom of the page. The backup configuration will be immediately removed and you will be returned to the main page of the module. The actual backup files created by the configuration will not be touched though.

14.6 Restoring a Backup

Backups made using Webmin (or by running the `dump` command manually) can be restored using this module as well. If you have been creating backups of different levels, they must be restored in ascending level order starting with the complete backup (level 0). A backup can be restored to any directory, not just the one that it was originally saved from. However, some file information such as ACLs and attributes will be lost if the restore directory’s filesystem does not support them.

To restore a backup, the steps to follow are:

1. On the main page of the Filesystem Backup module, select the type of filesystem that the backup was made from, using the list next to the **Restore backup of filesystem type** button. Because there are different programs for restoring different types of filesystems, the restore options will vary depending on the type you choose.
2. Click the **Restore backup of filesystem type** button, which will take you to the restore options page. Figure 14.3 shows the page for restoring an `ext2` or `ext3` filesystem backup.

3. If restoring from a local file, set the **Restore from file or device** field to the **File or tape device** option and enter the file that you want the backup read from into the text field next to it.

Restoring from a tape drive is similar to reading from a file, but instead of entering a filename into the **File or tape device** field you must enter the device file for the tape drive: `/dev/st0`, for example.

If restoring from another server, you must select the **Host** option and enter a hostname, remote username, and file or device name on the remote server. As explained in Section 14.3 “Adding a New Backup”, the server must have been configured correctly to allow remote access.

4. By default, everything in the backup will be restored. To extract only some files, set the **Files to restore option** to **Listed files** and enter paths to the files you want to restore into the field next to it. To restore multiple files, the filenames must be separated by spaces.

Because the paths used in the backup are sometimes relative to the mount point of the filesystem that they were originally on, it is often a good idea to use the **Only show files in backup?** option to see what the correct filenames are.

5. In the **Restore to directory** field, enter the base directory under which you want the restored files to be saved.
6. If the original backup spanned multiple files, set the **Backup is split across multiple files?** option to **Yes**.
7. If you just want to view the contents of the backup without extracting any files, set the **Only show files in backup?** option to **Yes**.
8. If you are familiar with the `restore` command used on your operating system, the **Extra command-line parameters** field can be used to enter extra options to be passed to the program, such as `-A /tmp/archive`. Otherwise, leave it blank.
9. When you are ready to restore, click the **Restore Backup Now** button. If extracting files for real, a page showing the output of the appropriate `restore` command will be displayed. If you chose to just view the files in the backup, the page will display a list produced by the `restore` command instead.

When restoring a backup from an `xfs` filesystem, different options are available on the restore form. The **Files to restore** option does not exist, so all files in the backup will be extracted. However, there is an **Overwrite existing files** option that can be set to **Never** to protect existing files, or **Unless newer than backup** to protect files that have been modified since the backup was made.

One problem with using Webmin to restore is that it cannot cope with backups that span multiple tapes. Normally the `restore` command would prompt the user to eject the first tape and insert the second, but that is not possible when it is being run by Webmin.

14.7 Configuring the Filesystem Backup Module

Clicking on the **Module Config** link in the top-left corner of this module’s main page will bring up a form for setting options that control how it behaves. The available settings and their meanings are see in Table 14.1.

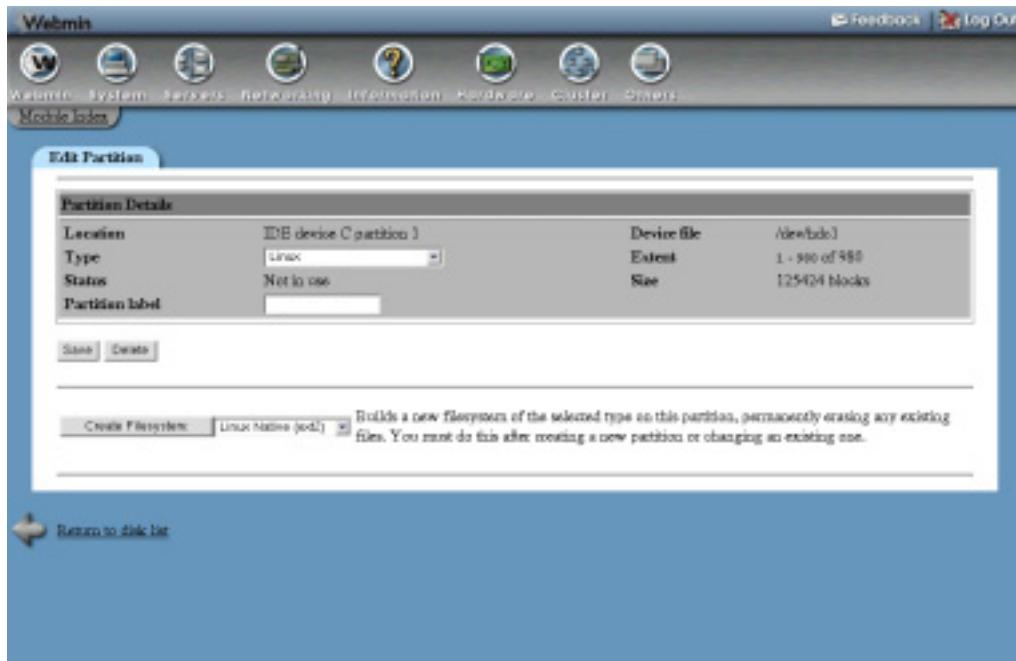


Figure 14.3 The backup restoration form.

Table 14.1 Module Configuration Options

Do strftime substitution of backup destinations?	If Yes is selected in this field, the backup destination path will have any special codes starting with % replaced with components of the current time and date. For example, %m will be replaced with month number, %d with the day of the month and %y with the year. These are the same substitutions that the standard UNIX <code>strftime</code> function uses. This option is useful if you want the backup to be written to a different file each day, instead of over-writing the same file every time. The default option is No , which turns off this behavior.
Send mail via SMTP server	When Local sendmail executable is selected, the output from scheduled backups will be sent by running the <code>sendmail</code> command on your system, the path to which is taken from the Sendmail Configuration module. However, you can tell the module to send email by making an SMTP connection to some other host instead, which may be necessary if your system does not run a mail server at all, or runs one other than Sendmail. Just select the second radio button and enter a hostname into the text box.

14.8 Other Operating Systems

Many UNIX operating systems have similar `dump` and `restore` commands to Linux, and several of them are supported by this Webmin module. However, the options available differ slightly, so the backup and restore forms on different systems will not be exactly the same as Linux.

The currently supported systems and their differences are:

Sun Solaris On Solaris, `ufs` filesystems can be backed up and restored using the `ufsdump` and `ufsrestore` commands. When creating a backup, the **Split across multiple files?** and **Tape size** options are not available—instead, there are **Verify data after backup?** and **Eject tape after backup?** options whose meanings should be obvious. Solaris also supports the backing up of multiple directories at once, by entering multiple paths separated by spaces into the **Directory to backup** field.

For restoring on Solaris, the options are essentially the same as on Linux.

FreeBSD and Apple MacOS X Both these operating systems have almost identical `dump` commands and available options in Webmin, due to the BSD ancestry of MacOS X. When making a backup, the **Split across multiple files?** and **Dump label** fields are not available, but **Tape size** is. The only filesystem type that can be backed up is `ufs` and a backup must be of an entire filesystem, not just any directory. Unfortunately, on MacOS X almost all filesystems are in Apple's native `hfs` format.

When restoring, the only difference is an additional **Just test backup?** option which when set causes the `restore` command to do everything except write to disk.

SGI Irix On Irix, the only filesystem type that can be backed up with Webmin is `xfs`, even though there is a `dump` command for older `efs` filesystems. As with the `xfs` filesystem on Linux, only entire filesystems can be backed up, not arbitrary directories. The **Tape size** option is not available, but instead you can limit the size of files to include with the **Maximum file size to include** option, and turn off the backing up of attributes with the **Include file attributes?** option.

When restoring a backup on Irix, there is no option to specify which files to extract—instead, everything in the backup will be restored. However, there is an **Overwrite existing files?** option to protect existing files, or existing files that are newer than the backup, from being overwritten.

Due to the low-level nature of backups made using the `dump` family of commands, a backup created on one operating system will not be restorable on any other.

14.9 Summary

After reading this chapter, you should be able to use Webmin to create backups of data on your system's local hard disks, and restore those backups if needed. You should also understand the difference between the `dump` backup format used by the module covered here, and those created by commands like `tar` and `cpio`. On a system running important servers or hosting vital data, proper backups are vital—and Webmin can help you create them.

Internet Services

This chapter covers the super servers `inetd` and `xinetd`, which are responsible for starting servers for protocols like telnet and FTP when needed.

15.1 Introduction to Internet Services

Heavily used network services such as email, proxying, and web serving are handled by server processes that run continually and have their own complex configuration files and Webmin modules. However, there are other services like telnet, finger, and POP3 that do not need any configuration and do not need their own permanent server process. Instead, their servers are run when needed by a super server like `inetd` or `xinetd` which listens for network connections on multiple ports. Only when it receives a connection does it start the appropriate process to communicate with the client, which exits when the connection is closed. This saves memory by limiting the number of processes running at any one time, but makes the handling of new connections slightly slower.

Every service has a short name like `telnet` or `pop3`, a port number like 23 or 110 and a protocol like TCP or UDP. The file `/etc/services` lists all the service names and their corresponding ports numbers that your system knows about, only a few of which may have a super server or other server listening on them.

The most commonly used super server is `inetd`, which is used by almost all Linux distributions and UNIX variants. All server settings are stored in the configuration file `/etc/inetd.conf`. In addition to starting servers in response to the TCP and UDP connections, it can also handle RPC (remote procedure call) function calls in a similar way. One major shortcoming of `inetd` is its inability to reject connections depending on the client IP address. However, this can be overcome by using an intermediate TCP-wrappers server program, which has its own IP access control configuration file.

Another super server that is gaining in popularity and has more features is `xinetd`, which uses the `/etc/xinetd.conf` configuration file and sometimes other files under the `/etc/xinetd.d` directory. Like `inetd`, it can launch server processes in response to TCP and UDP connections, but does not support RCP. Its major advantage is built-in support for restricting connections to certain client IP addresses without the need for a separately configured program. It can also re-direct an incoming connection on certain ports to another host and port by making its own client connection and forwarding data back and forth.

Because `inetd` and `xinetd` have totally different configuration files and file formats, there is a separate Webmin module for configuring each of them. Most Linux distributions will ship with one or the other, but in some cases both can be installed and co-exist peacefully. The only limitation is that they cannot both listen on the same port at the same time.

15.2 The Internet Services and Protocols Module

This module deals with the configuration of `inetd`, and can be found under the Networking category in Webmin. If the icon is not visible, Webmin has detected that it is not installed. This could be because your distribution is using `xinetd` instead, in which case you should skip down to Section 15.8 “The Extended Internet Services Module”. If neither module is visible, check your distribution CD or website for an `inetd` or `xinetd` package.

The module’s main page (shown in Figure 15.1) displays two tables, one for **Internet Services** that respond to TCP or UDP connections, and one for **RCP Programs**. In the **Internet Services** section, the names and protocols of all services are shown—in some cases, the same service may be recognized for more than one protocol. Each service can be in one of three states, indicated by the font its name is shown in:

Enabled (bold) A server program has been assigned to this service, and it is currently active.

Disabled (bold-italic) A server program has been assigned, but it is not active. This corresponds to a commented-out entry in the `inetd.conf` file.

Unassigned (normal) No server program has been assigned to this service, meaning there is no `inetd.conf` entry for it.

If the module configuration option **Show services with no program** has been set to **No**, services in the unassigned state will not be displayed. This is the default on some operating systems, due to the large number of services that the system knows about.

Most Linux distributions ship with almost all services in the disabled state by default. This limits the number of unnecessary services that your system allows connections to, and thus reduces the chance of a security hole in one of the server programs being exploited by an attacker.

Because each service is shown with only a short name like `telnet` or `chargen`, it is not obvious to an inexperienced administrator what each of them do. Some of the more commonly used services and their purposes are seen in Table 15.1.

The `daytime`, `echo`, and `chargen` services for both TCP and UDP protocols are handled internally by `inetd` when enabled, not by a separate server program.

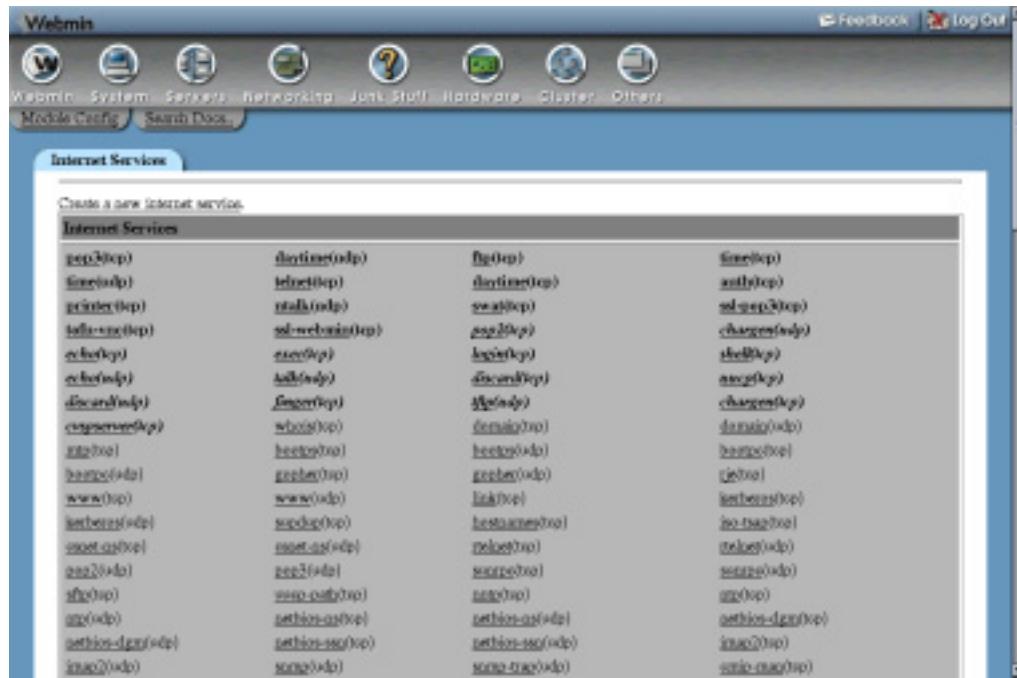


Figure 15.1 The Internet Services and Protocols module main page.

Table 15.1 Common Services and Their Purposes

Service name	Protocol	Purpose
telnet	TCP	Remote login using the telnet command. Because a telnet connection is unencrypted, any username or password sent over it can theoretically be captured by an attacker. On modern systems, the secure SSH protocol is usually used instead.
pop3	TCP	Mail retrieval using almost any mail client program, such as Outlook, Eudora or Netscape. If you want users to be able to pick up mail from your system, this protocol should be enabled.
imap	TCP	A superior mail retrieval protocol that supports folders and server-side mail storage. However, fewer mail clients support it.
finger	TCP	Remote user lookup using the finger command.
ntalk	TCP	Person-to-person chat using the UNIX talk program.

Table 15.1 Common Services and Their Purposes (Continued)

Service name	Protocol	Purpose
ftp	TCP	File upload or download using an FTP client. There are several different FTP server programs available, the most common being <code>wu-ftpd</code> and <code>proftpd</code> . Because they have many options that are configured separately, each has its own Webmin module as covered in Chapters 40 and 41.
shell	TCP	Unauthenticated remote login using the <code>rsh</code> command. Because the <code>shell</code> protocol validates users by client IP address only, it is not considered secure— <code>ssh</code> is a far better alternative. However, you may have to enable it for remote backups from the Filesystem Backup module.
login	TCP	Remote login using the <code>rlogin</code> command. Because this can be configured to validate users by client address only, it is considered insecure and rarely used.
exec	TCP	Remote command execution using the <code>rexec</code> program. Rarely used on modern systems, due to the superiority of <code>ssh</code> .
daytime	TCP	Upon connection, displays the server time in a human-readable format.
daytime	UDP	Like the TCP <code>daytime</code> service, but sends the human-readable time back in a UDP packet.
time	TCP	Upon connection, displays the system time as a 4-byte binary number.
time	UDP	Like the TCP <code>time</code> service, but sends the binary time back in a UDP packet.
echo	TCP	Sends back any data that is sent to it.
echo	UDP	Sends back any packets that it receives.
chargen	TCP	Produces an endless stream of data containing printable ASCII characters for as long as a client is connected.
chargen	UDP	Like the TCP <code>chargen</code> service, but sends back a single UDP packet of ASCII characters in response to each one received from a client.

15.3 Enabling an Internet Service

If you want to allow users to fetch mail from your system using the POP3 protocol or login via telnet, it is necessary to turn on the appropriate Internet service if it is not currently enabled. To do this, the steps to follow are:

1. On the main page of the module, click on the name of the service that you want to enable in the **Internet Services** table. This will take you to the page shown in Figure 15.2 for editing its details.

If unassigned services are not displayed on your system, you can enter the service name and select the protocol in the fields next to the **Edit service** button. Clicking the button will take you to the editing form, assuming the service name is recognized.

2. The **Service name**, **Port number**, **Protocol**, and **Aliases** fields should be left unchanged unless you want to rename the service or change the port it is listening on. For services that you did not create yourself, changing any of these fields is a bad idea, as it may prevent programs on your system connecting to other servers.
3. To enable the service in the **Server program** section, select the **Program enabled** option. If **Program disabled** was selected previously, then all the other settings in the section should be correct and will not need to be changed.

However, if **No program assigned** was selected before, then you will need to choose a server program and a user for the server to run as. Select the **Program** field **Command** option and enter the full path to the server program into the field next to it, such as `/usr/sbin/in.ftp` for an FTP server. In the **Args** field, enter the server command again and any arguments that it needs, such as `in.ftp -l -a`. Even though the program path is in the **Command** field, the program name must appear in the **Args** field as well.

You will need to enter into the **Execute as User** field a username for the server program to run as. For almost all servers, this will be `root`. One of the **Wait Mode** options must be set as well—unless the server runs and executes very quickly, choose **Don't wait**.

Some services such as `daytime`, `echo`, `chargen`, and `discard` are handled internally by `inetd`. If you are enabling one of them, just select the **Internal to inetd**. No program or arguments need to be entered, and the user the server executes as is irrelevant.

4. When you are done, click the **Save** button. As long as there are no errors and the chosen server program actually exists, the browser will return to the list of services on the main page.
5. Click the **Apply Changes** button at the bottom of the page to make your changes active.

In some cases, you will not be able to enable a service because the corresponding server program is not installed yet. If this is the case, use the Software Packages module to install it from your Linux distribution CD or website.

If you want to disable a service, just follow the same steps but select the **Program disabled** option instead. This is better than choosing **No program assigned** as it is easy to turn the service back on again without having to re-enter the server program details.

15.4 Creating Your Own Internet Service

In some situations, you may want to add a new server to your system that listens on a port not assigned to anything else. You might want to run a telnet server on some non-standard port, or



Figure 15.2 Editing an Internet service.

redirect traffic from one port on your system to another server using a program like nc. If you are just trying to turn on some standard service like `ftp` or `imap`, the instructions in this section are not for you—see Section 15.3 “Enabling an Internet Service” instead.

The steps to follow to create a new service are:

1. On the main page of the module, click the **Create a new internet service** link. This will take you to the service creation form, which is similar to the editing form in Figure 15.2.
2. Fill in the **Service Name** field with a unique name for your service.
3. Enter the port number you want the service to be associated with into the **Port Number** field.
4. Select the protocol from the **Protocol** list. This will almost always be TCP, but in some cases you may need to use UDP.
5. In the **Aliases** field, enter any alternate names that you want the service to be referred to by.
6. Assuming you want to have a server program associated with this service, choose the **Program enabled** option in the **Server Program** section. Otherwise all that will be created is an association between a service name and port number.
7. For the **Program** field, select the **Command** option and enter the full path to the server program into the field next to it—for example `/usr/local/bin/someserver`. In the **Args** field, enter the program name and any command line arguments that it should be run with, such as `someserver -foo`.

To give another example, if you wanted to create a service that displayed all the processes running on your system to anyone who connected via telnet, you could set the

- Command** to `/bin/ps` and the **Args** to `ps auxwww`. (This would be a bad idea from a security point of view, though.)
8. If the server program is going to take more than a second to run, or if it accepts any input, set the **Wait mode** field to **Don't wait**. Otherwise `inetd` will stop handling new network connections until the program has finished. The only advantage of this **Wait until complete** mode is a slight reduction in memory usage.
 9. In the **Execute as User** field, enter the username of the UNIX user that the server program should run as. This is usually `root`, but can be anyone.
 10. To limit the rate at which `inetd` will accept connections for your service, enter a number into the **Max per Minute** field. If the limit is exceeded, subsequent connections will be refused until the next minute.
 11. By default, the group that the server program runs as is the primary group of the user set in the **Execute as User** field. To change this, enter a group name into the **Execute as Group** field.
 12. Click the **Create** button to create your service. As long as there are no errors in the form, you will be returned to the list of services on the main page.
 13. Click the **Apply Changes** button to make the service active.

Once a service has been created, you can test it by running `telnet localhost portnumber` at the shell prompt on your system. You can edit your service at any time by clicking on its name on the main page, and changing any of the options before clicking **Save**—or **Delete** if you want to get rid of it. After making any modifications, the **Apply Changes** button must be used to make them active.

15.5 Creating and Editing RPC Programs

RPC is a protocol and data format that is the basis for other protocols like NFS and NIS. RPC clients make function calls to RPC servers, passing parameters and getting back results. To the client or server, making a remote procedure call is no more difficult than calling a normal library function, which makes writing programs that use RPC much easier than creating your own protocol from scratch.

An RPC program is a set of functions that are handled by a server. Each program has a unique number, similar to the port of an Internet service. Programs are not associated with a particular protocol, as they can generally accept connections and function calls via UDP or TCP. Nor do they have a fixed port, as they are assigned dynamically when needed.

RPC servers (like the NIS and NFS servers) that handle a large amount of traffic have their own processes that run all the time. However, some servers that need to be run only occasionally can be executed by `inetd` only when needed—just like infrequently used Internet services. Some of the more commonly used RPC programs are:

On some systems, these RPC programs may be handled by servers that are not run from `inetd` but instead as stand-alone processes. In that case, the Bootup and Shutdown module (explained in Chapter 9) is the place to activate or de-activate them. Due to the small number of common RPC programs and their limited usefulness, many Linux distributions do not have any programs enabled or disabled in the `inetd` configuration by default. However, this is not the case on other operating systems like Solaris.

Table 15.2 Common RPC Programs and Their Purposes

Program name	Purpose
rquotad	Remote disk quota retrieval. If a system is NFS-exporting a filesystem with quotas, this program can be used by the <code>quota</code> command on the client to display used and available blocks and files on the NFS-mounted filesystem.
rusersd	Requesting the list of users logged into a system. The <code>rusers</code> command can be used to display users logged into one or more servers.
walld	Broadcasting a message to users on a server. Like the <code>wall</code> command for sending a message to local users, the <code>rwall</code> command sends to users on another system by calling this RPC program.

If you want to make use of an RPC protocol which is not currently enabled, you can use this module to turn it on. Of course the appropriate RPC server program must be installed first, and the `inetd` on your system must support RPC programs. If so, the steps to follow are:

1. On the main page of the module, click on the program name from the **RPC Programs** table. This will take you to the program editing form shown in Figure 15.3.
2. Under the **Server Program** section, select the **Program enabled** option. If **Program disabled** was selected previously, then all the other settings in the section should be correct and will not need to be changed. However, if **No program assigned** was checked, you will need to fill in several other fields.

The **RPC Versions** field should be set to the range of versions that the server program supports, such as *1-3*.

The **Socket Type** field should be set to **Datagram**, and the **Protocol** field set to only the **udp** option.

For the **Server Program** field, enter the full path to the RPC program, such as `/usr/sbin/rpc.rusersd`. For the **Command** field, enter the program name and any arguments, such as `rpc.rusersd -a`.

For the **Wait Mode**, select **Don't wait**.

For the **Execute as User** field, enter the username you want the server program to run as—usually `root`.

3. When done, click the **Save** button. As long as there are no errors in your input, you will be returned to the main page of the module where the RPC program should appear as enabled.
4. Click the **Apply Changes** button to make the program active.

15.6 Configuring the Internet Services and Protocols Module

To access the configurable options of the Internet Services module, click on the **Module Config** link in the top left corner of its main page. This will take you to the standard configuration form, on which you can change the options shown in Table 15.3.

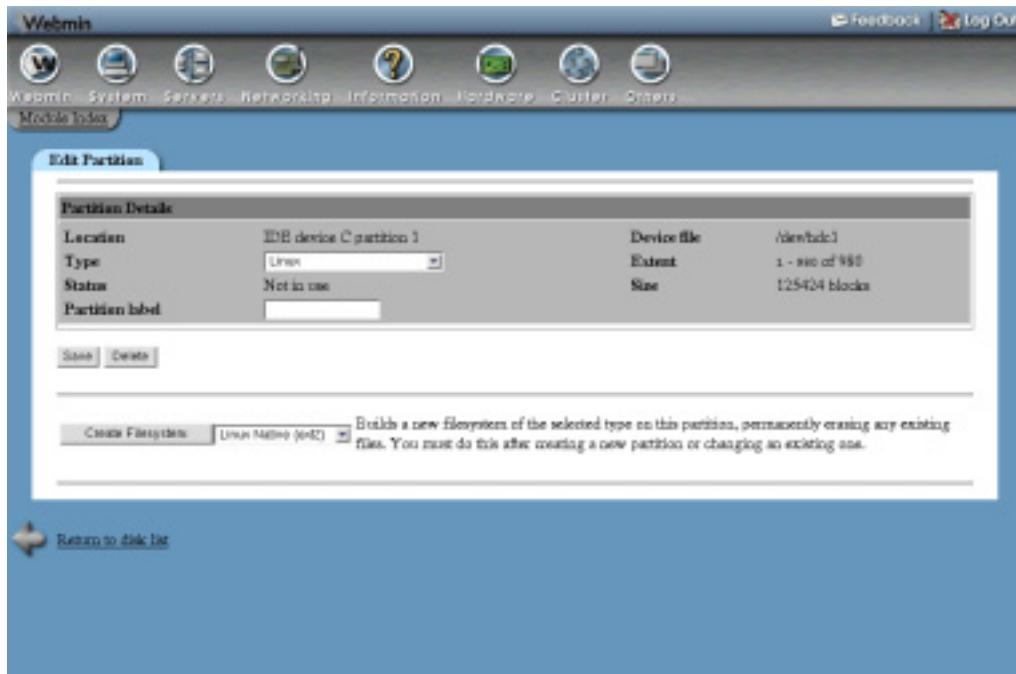


Figure 15.3 The RPC program editing form.

Table 15.3 Module Configuration Options

Show services with no program	If set to Yes , the main page will show only Internet services that have an enabled or disabled server program assigned. On most distributions the default is No , but some have so many known services that this option has to be turned on to limit the size of the services list.
Sort services and programs by	Controls the ordering of Internet and RPC services on the main page. If Name is selected, they will be ordered by service name. If Assignment is selected, all those that have an enabled server program will be listed first, followed by those with a disabled program and finally those that are unassigned. If Order in file is chosen, the services will be displayed in the same order as they are stored in /etc/services, which is usually by port number.

The rest of the module configuration options under **System configuration** are set automatically by Webmin based on your operating system type, and so should not be changed.

15.7 Other Operating Systems

Almost all versions of UNIX include `inetd` as standard, and use it to launch infrequently run server programs in the same way that Linux does. However, its configuration file format and capabilities are slightly different on other operating systems, which means that the module's user interface will not be exactly the same. The main page will always show lists of Internet and RPC services, but when editing or creating a service, different fields and options will be available depending on the UNIX variant you are running:

Sun Solaris

- When editing an Internet service, the **Max Per Minute** and **Execute as Group** fields are not available.
- Solaris versions 8 and above support IPv6 TCP and UDP protocols, as well as the standard IPv4 that Linux uses.
- Many RPC services exist in the disabled state by default, for things like NFS quotas and locking.

FreeBSD

- RPC services cannot have programs assigned. All you can do is edit the service names and program numbers.
- When editing or creating a service, you can control the number of server programs that can be active at any one time with the **Max Child Processes** field.
- Also when editing, you can set the login class that the server program runs as with the **Execute as Login Class** field.

NetBSD

- As on FreeBSD, the **Max Child Processes** and **Execute as Login Class** fields are available when editing or creating a service.
- As with Solaris, Internet services can use IPv6 TCP and UDP protocols.

OpenBSD, Compaq Tru64/OSF1, IBM AIX, SCO OpenServer, and SCO UnixWare

- As on Solaris, the **Max Per Minute** and **Execute as Group** fields are not available.

SGI Irix

- The **Max Per Minute** and **Execute as Group** fields are not available when editing a service.
- There is an additional checkbox below the server program **Command** field labeled **Command may not exist?** If this is set, it tells `inetd` to ignore the service if the server program is not installed. By default, this is turned on for many services related to Irix packages that are not installed by default.

HP/UX

- On HP/UX, the module has exactly the same options as on Linux.

Apple MacOS X

- As on Solaris, the **Max Per Minute** and **Execute as Group** fields are not available.
- RPC services cannot have programs assigned, as on FreeBSD.
- Instead of being stored in the `/etc/services` file, service names and ports are in a NetInfo table. Webmin dumps and re-loads this table to read and edit services.

15.8 The Extended Internet Services Module

This module allows you to configure `xinetd`, a super server that is similar in purpose to `inetd` but has several additional features. Like the Internet Services and Protocols module, this one can also be found under the Networking category. However, its icon will appear only if Webmin detects that `xinetd` is installed, which it does by looking for the `/etc/xinetd.conf` file. If you have compiled and installed it manually, you may need to create a symbolic link to the real location of `xinetd.conf`.

The main page lists all services that have server programs assigned, their port numbers, protocol, program, and active status—see Figure 15.4 for an example. Services with no programs are never shown, unlike in the Internet Services module.

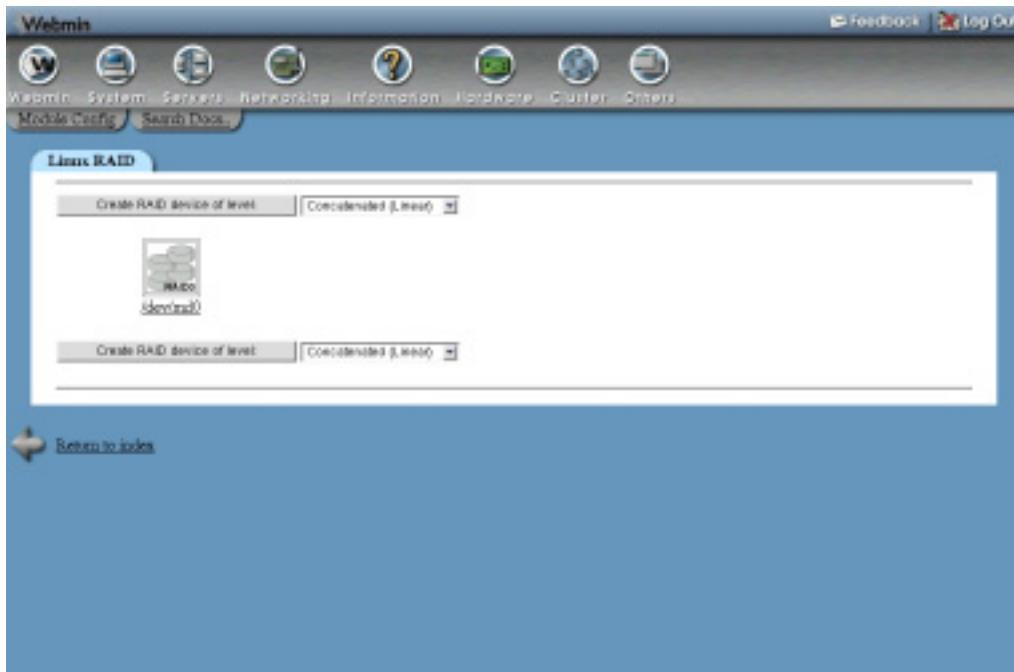


Figure 15.4 The Extended Internet Services module.

On Linux distributions that use `xinetd`, most server program packages include a file that adds an appropriate service to the list shown on the main page. These are generally disabled by default, so that services are not unexpectedly enabled the moment you install them.

If you are using a different operating system on which you have installed `xinetd`, the user interface will be exactly the same as on Linux. However, server program packages will probably not set up services when installed.

15.9 Enabling or Editing an Extended Internet Service

If you want to allow users to fetch mail from your system using the POP3 protocol or login via telnet it is necessary to turn on the appropriate service in this module, assuming it is listed on the main page. If not, you will need to first install the appropriate package from your distribution website or CD, which should add an entry for the service. If not, see Section 15.10 “Creating an Extended Internet Service”.

Existing services can also be changed in other ways—for example, to restrict the allowed client IP addresses or number of concurrent connections. To edit a service, the steps to follow are:

1. On the main page of the Extended Internet Services module, click on the name of the service that you want to edit. This will take you to the form shown in Figure 15.5.
2. The **Service name**, **Socket type** and **Protocol** options should all be left unchanged. The **Port** field should be changed only if you know what you are doing.
3. To turn on the service, set the **Service enabled?** field to **Yes**. Or if it is already enabled and you want to turn it off, select **No**.
4. If you want the service to be accessible only via a single IP address on your server, enter it into the **Bind to address** field. This can be useful if you have multiple virtual IP interfaces on your system and want different servers to listen on different addresses.
5. Most of the fields under **Server program options** can be left unchanged, unless you want to limit the amount of load the service puts on your system. If so, you can set the **Max concurrent servers** field to the maximum number of server processes that should be allowed to run at any one time. The **Maximum connections per second** and **Delay if maximum is reached** fields can be set to limit the rate at which clients are allowed to connect and the amount of time that the service is disabled if that rate is exceeded.
6. To control which addresses clients are allowed to connect from, use the fields in the **Service access control** section. If **Allow access from** is set to **Only listed hosts**, only the IP addresses (like *192.168.1.55*), hosts (like *server:foo.com*) and networks (like *192.168.1.0/24*) entered will be allowed. If **Deny access from** is set to **Only listed hosts**, the hosts, IP addresses, and networks entered will be prevented from connecting.
If a client matches an entry in both lists, the most specific entry will be used to determine whether access is allowed or denied. For example, if *192.168.1.10* was allowed and *192.168.1.0/24* was denied then a client with IP address *192.168.1.10* would be able to connect.
7. If you want to limit the times at which the service can be used, fill in the **Allow access at times** field. It must be in the format `HH:MM-HH:MM`, such as `9:00-17:00` to allow access during normal working hours.

8. Click the **Save** button when you are done making changes. As long as you haven't made any mistakes, the browser will return to the module's main page.
9. Click the **Apply Changes** button to make your modifications active.

If you want to delete a service totally, you can click the **Delete** button on the editing form instead. However, it is usually better to simply disable it so that it can be easily turned back on later.

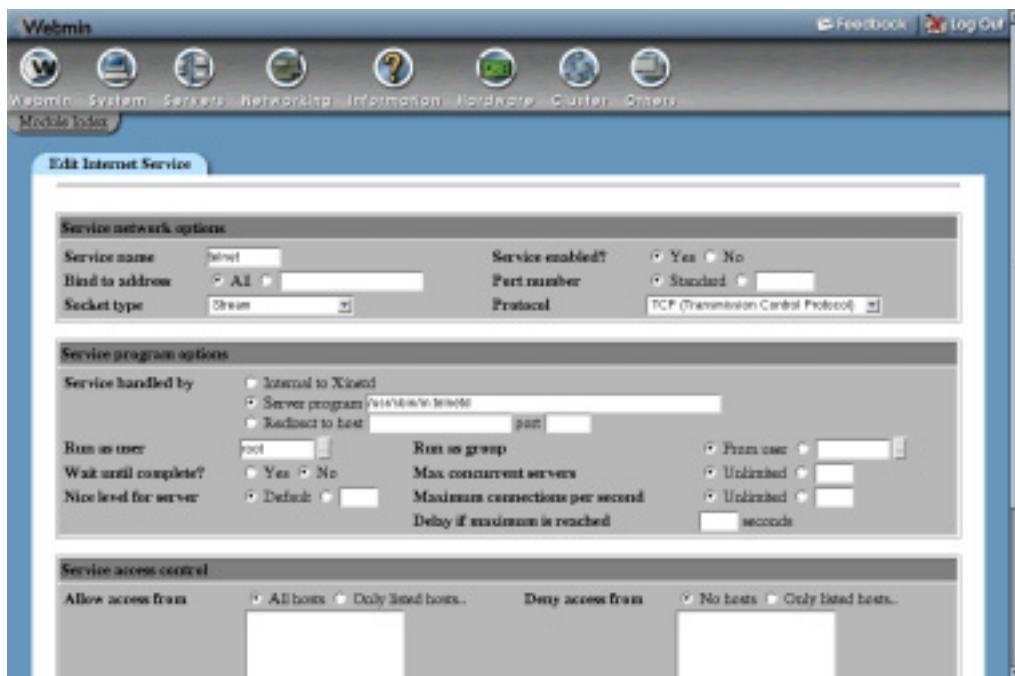


Figure 15.5 Editing an extended Internet service.

15.10 Creating an Extended Internet Service

If you want to enable a protocol that is not in the list on the main page, or redirect traffic from a particular port to another host, then you will need to create a new service using this module. The appropriate server program for the service must be installed first, unless you are setting a redirection. The steps to follow are:

1. Click on the **Create a new internet service** above or below the list on the main page. This will take you to the creation form, similar to the one in Figure 15.5.
2. If the service is for a standard protocol like telnet or finger, enter its name in the **Service name** field. The **Port number** can then be left set to **Standard**. Otherwise, enter a unique name into the **Service name** field and set the **Port number** to the port you want the service to listen on.

3. If you want the service to be accessible only via a single IP address on your server, enter it into the **Bind to address** field. This can be useful if you have multiple virtual IP interfaces on your system and want different servers to listen on different addresses.
4. Set the **Protocol** field to the protocol you want the service to use, usually TCP. The **Socket type** field should be set to **Stream** for TCP protocol services, or **Datagram** for UDP services.
5. If your service is going to use a server program, set the **Service handled by** option to the **Server program** option and enter its command and any arguments into the field next to it—for example, `/usr/sbin/in.telnetd -a`.
If the service is just redirecting traffic to another host, select the **Redirect to host** option and enter the destination hostname and port in the corresponding fields. Redirection can be useful for making services on an internal network system available to the rest of the Internet, if your firewall or gateway host is running xinetd.
6. In the **Run as user** field, enter the name of the UNIX user that the server program will be run as. This is not necessary for redirection services.
7. Unless the server program always completes very quickly, set the **Wait until complete** field to **No**. If you leave it set to **Yes**, xinetd will not process any more connections until the program finishes.
8. To limit the rate at which clients can connect, set the **Max concurrent servers** and **Maximum connections per second** fields as explained in Section 15.9 “Enabling or Editing an Extended Internet Service”.
9. To limit the addresses from which clients can connect or the times at which connections are allowed, set the fields under **Service access control** as explained in the list above.
10. When done, click the **Create** button. If there are no errors in the form, you will be returned to the main page on which your new service should now be listed.
11. Click the **Apply Changes** button to make the service active.

Once a service has been created, you can test it by running `telnet localhost portnumber` at the shell prompt on your system. You can edit or delete your service at any time by following the instructions in Section 15.9 “Enabling or Editing an Extended Internet Service”.

15.11 Editing Default Options

There are several global options that apply to all services handled by xinetd, for logging and IP access control. To edit these options, the steps to follow are:

1. Click the **Edit Defaults** button at the bottom of the module’s main page, which will take you to the default options form.
2. To restrict the addresses from which clients can connect to any service, fill in the **Allow access from** and **Deny access from** fields. They accept the same input as the fields of the same name on the service form, as explained in Section 15.9 “Enabling or Editing an Extended Internet Service”.

Any IP access controls configured for an individual service will override the default settings that you enter on this form.

3. To have xinetd log to syslog, set the **Xinetd logging mode** field to **Log to syslog facility** and choose the facility and priority that it should use. Chapter 13 explains in detail how to configure the log file that messages from xinetd will be written to, based on the selected priority and facility. Normally, this is the default and best option.

If you want xinetd to log directly to a file, select the **Log to file** option and enter the log file path into the field next to it. To have a warning message logged when the file becomes too big, enter a file size in bytes into the **Soft file limit** field. To set a file size limit that will never be exceeded, fill in the **Hard file limit** field. If the soft limit is set but the hard limit is not, it will default to 1 percent more than the soft limit. If neither is set, the log file will grow forever—which could cause all your disk space to be consumed by an attacker making millions of connections to xinetd.

To turn off logging altogether, set the **Xinetd logging mode** field to **Disable logging**.

4. To control which events are logged, choose the appropriate options from the **On successful connection log** and **On failed connection log** fields.
5. When done, click the **Save** button. As long as there are no errors in your input, you will be returned to the module's main page.
6. Click the **Apply Changes** button to make the new defaults active.

15.12 Summary

This chapter has explained the purposes of the two common UNIX super server programs—inetd and xinetd—and the differences between them. After reading it, you should know how to enable standard services started by either of these programs, and how to create and edit your own services, if necessary. Several other chapters refer back to the modules covered in this one, especially those on FTP, CVS, and SSL tunnel servers.

Network Configuration

This chapter explains how to set your system's IP address, hostname, DNS servers, and other network settings. It covers both Linux and other UNIX variants.

16.1 Introduction to Linux Networking

A Linux system can be connected to a network or the Internet in several different ways—for example, via an Ethernet network card, a token ring card, or a PPP (Point-to-Point Protocol) connection over a dial-up modem. If your system is never connected to a network, then this chapter is not for you. However, if you do need to set up network connections (especially to a local area network), then read on.

Every Ethernet network card, PPP connection, wireless card, or other device in your system that can be used for networking is known as an *interface*. Interfaces are usually associated with a piece of hardware (like a network card), but they can also be dynamically created (like PPP connections). For an interface to be used, it must first have an IP address assigned, which may be fixed and set from a configuration file on your system or dynamically assigned by a server. An Ethernet interface for a desktop PC on a company or home network would usually have a fixed address, whereas a PPP connection interface to an ISP would have its address dynamically assigned by a server at the other end.

PPP interfaces are configured in a very different way to Ethernet and other fixed hardware interfaces. Before one can be activated, a modem must be used to dial an ISP at a particular phone number and log in with a username and password. Only after the login is successful will the PPP interface have an IP address assigned by the ISP's access server. Other network settings on your system such as the DNS server addresses and default gateway will be assigned by the ISP as well. An Ethernet interface, however, can have an IP address set and start working at any time, and a system connected via Ethernet usually uses fixed DNS server and gateway addresses.

Sometimes an Ethernet interface will have its addresses dynamically assigned as well. If so configured, the system will broadcast a request for an address using the DHCP (Dynamic Host Configuration Protocol) when the interface is activated at boot time. This will be answered by a DHCP server, which supplies the IP address and possibly default gateway and DNS server addresses as well. DHCP is often used on large networks with many systems that frequently connect and disconnect (such as laptops), in order to avoid manually configuring each system with a fixed IP address.

One special network interface that is always available is the loopback interface. It always has the IP address `127.0.0.1`, which is mapped to the hostname `localhost`. This interface cannot be used to communicate with other systems, just your own—for example, running the command `telnet localhost` will bring up the login prompt of your own system (assuming a telnet server is active).

Every interface has a name, like `eth1` or `ppp0`. All Ethernet interfaces start with `eth`, PPP interfaces with `ppp`, loopback with `lo` and token ring with `tr`. The number tells you which network card of that type the interface is related to—if your system had two Ethernet cards, the first would be `eth0` and the second `eth1`.

If your system is connected to a network any bigger than a small home LAN, one of the computers on the network will be the gateway. This is a server (or more likely a router) that knows how to route traffic to other networks or the Internet, perhaps by a PPP link, broadband connection, or other network card. For your system to communicate with those other networks, it must be configured with the IP address of the gateway.

All communication on an IP network is done using IP addresses like `192.168.1.10` or `210.23.128.117`. Because addresses like this are not too easy for the average person to remember, they can have names associated with them as well, like `server.foo.com`. Any time a system needs to lookup an IP address for a hostname (or vice versa) it queries a DNS server which will supply the needed information, either from its own records or by querying other DNS servers on the network or Internet. For your system to be able to query a DNS server, it needs to be configured with the IP address or addresses of nearby servers and a default domain name to append to any hostnames.

Not all IP addresses are looked up from a DNS server though—some are stored in the `/etc/hosts` file on your system so that they can be found even when networking is not active. Typically the IP addresses for `localhost` and your system's hostname will be stored in this file, because they rarely change.

As would be expected, the Network Configuration module can be found under the Networking category in Webmin. The main page shows one icon for each of the four configuration categories—**Network Interfaces, Routing and Gateways, DNS Client, and Host Addresses**. All the editable forms and options in the module are under one of those four categories.

This module was designed mainly for configuring networking on systems with permanent network connections, such as Ethernet or token ring cards. If your system has only a dial-up PPP connection to the Internet, it will not be much use to you. Instead, you should use Webmin's PPP Dialup Client module, which allows you to set phone numbers, usernames, and passwords for dial-up connections.

The forms in this module only allow you to set up your system as a DNS and DHCP client. If you want to run your own DNS server on your network, see Chapter 30. To learn how to set up your own DHCP server, see Chapter 32.

16.2 Viewing and Editing Network Interfaces

To view the interfaces that are currently active on your system, click on the **Network Interfaces** icon on the main page of the module. This will take you to the page shown in Figure 16.1, which lists interfaces on your system in two categories. At the top under **Interfaces Active Now** are those that are currently enabled and have an IP address assigned. All loopback, Ethernet, and PPP interfaces will be shown, although not all will be editable using Webmin. At the bottom under **Interfaces Activated at Boot Time** are those which have been configured to be activated at boot. The two lists will not necessarily be the same, as some types of interface (such as PPP) are not activated at boot time and so will not appear in the second list.

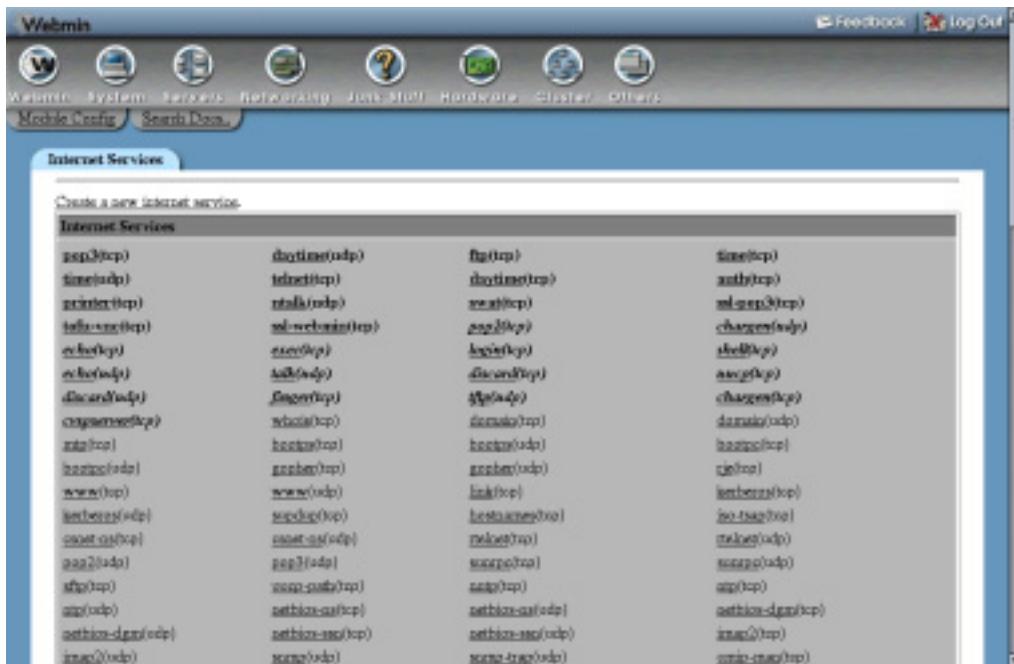


Figure 16.1 The network interfaces page.

The steps to follow to change the IP address, active status, or other details of an interface are:

1. If the interface appears under both **Interfaces Active Now** and **Interfaces Activated at Boot Time** (as most editable ones do), click on its name in the lower list. This will take you to a form for editing its settings, shown in Figure 16.2.
 2. To assign a different address, enter it into the **IP Address** field. Or select the **From DHCP** option if you want the address to be dynamically assigned by a DHCP server on your network.
 3. If necessary, change the **Netmask** field. If it or the IP address is changed, you will also need to set the **Broadcast** address field based on the new netmask and IP.
 4. When editing an active interface, the **MTU** and **Hardware address** fields will be available. You should leave the MTU alone unless you really know what you are doing, as

changing it could reduce network performance or cut your system off from the network altogether. The hardware address should be changed only if you want to give your network card a different Ethernet address, which is rarely necessary.

5. If editing a boot-time interface, make sure the **Activate at boot?** field is set to **Yes** so that the interface is brought up when the system starts.

If editing an active interface, make sure the **Status** field is set to **Up** so that it can be used immediately.

6. When done editing a boot-time interface, click the **Save and Apply** button to save your changes for use at bootup time, and to make them immediately active.

If you are editing an active interface, just click **Save** to activate your changes.

After changing any of your system's IP addresses, be sure to update any host address entries associated with them as well. See Section 16.6 "Editing Host Addresses" for details on how to do this. You may also need to update records in your DNS server as well.

An active interface can be shut down by clicking the **Delete** button on its editing form instead. Similarly, a boot-time interface can be removed (for example, if you have removed a network card) so that it will not be activated at startup by clicking the **Delete** button on its form.

16.3 Adding a Network Interface

There are two situations in which you might want to add a new network interface—if your system has just had a network card installed, or if you are adding an additional virtual IP address to an existing interface. In the latter case, the new virtual interface is not associated with its own separate network card, but instead adds an additional IP address to an existing Ethernet card. Virtual addresses are often used on systems hosting multiple websites, so that each site can have its own IP address.

Before an interface for a new network card can be configured, you must make sure that it is recognized by the Linux kernel and the appropriate kernel module loaded. There is no support in Webmin for doing this at the moment, but most distributions include a graphical tool for loading kernel modules, or a configuration file in */etc* that specifies which modules to load. Once the interface is recognized, the steps to configure it are:

1. On the main page of the module, click the **Add a new interface** link under **Interfaces**. **Activate at Boot Time**. This will take you to the creation form, which is similar to the editing form in Figure 16.2.
2. Enter the interface name (such as *eth1* or *tr0*) into the **Name** field. This must correspond to whatever name has been assigned by the kernel.
3. In the **IP Address** field, either enter an address or select the **From DHCP** option for it to be dynamically assigned.
4. Enter the netmask for the network the interface is on into the **Netmask** field, such as *255.255.255.0*.
5. Set the **Broadcast** field based on the address and netmask. For example, if the IP was *10.1.2.3* and the netmask was *255.0.0.0*, then the broadcast address would be *10.255.255.255*.
6. If you want the interface to be brought up at boot time, set the **Activate at boot?** field to **Yes**.

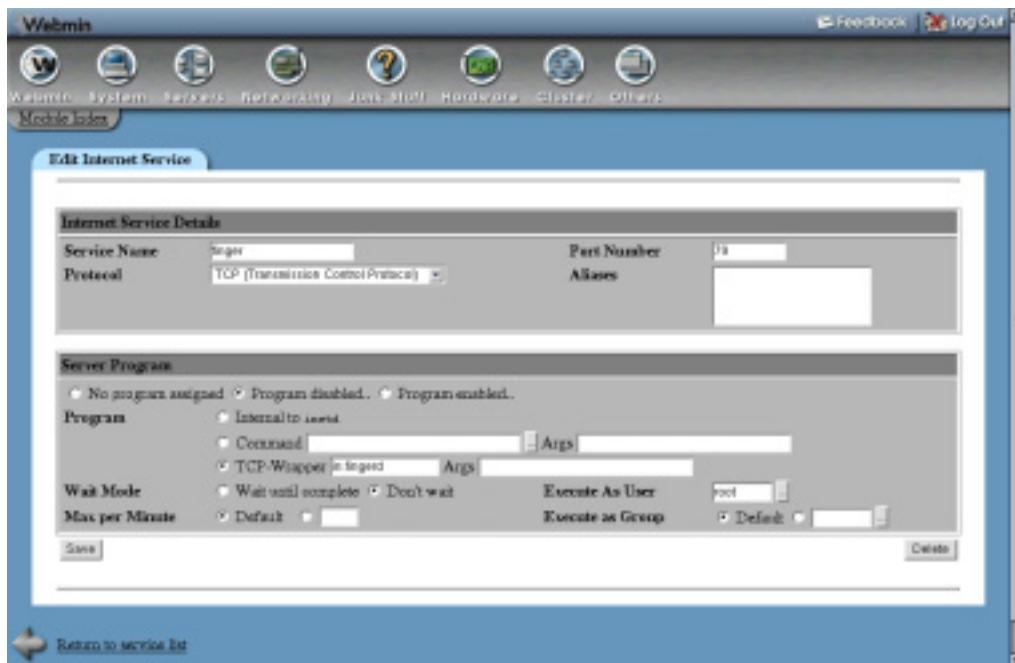


Figure 16.2 The interface editing form.

7. Finally, click the **Create** button. Assuming there are no errors in your input, you will be returned to the list of interfaces.
8. To make the interface active now, click on its name from the **Interfaces Activate at Boot Time** list. Then on the editing form, click the **Save and Apply** button. If any error occurs during activation (such as the interface not being recognized by the kernel) Webmin will display an error message.

A virtual interface adds an additional IP address to an existing real interface. Virtual interfaces have names like *eth0:1*, where *eth0* is the name of the real interface and *1* is the virtual number. To add one, the steps to follow are:

1. On the main page of the module, click on the real interface that you want to add a virtual address for, under **Interfaces Activate at Boot Time**.
2. On the editing form, click the **Add virtual interface** link. This will take you to a creation form, similar to Figure 16.2.
3. In the **Name** field, enter a number for the virtual interface. This must not be used by any existing virtual interface on the same real network card.
4. Fill in the **IP Address** field with the address that you want to assign to the virtual interface.
5. The **Netmask** and **Broadcast** fields should be set to the same addresses as the real interface. They would only be different if the virtual interface was on a different IP network that was sharing the same LAN as the network for the real interface.

6. Assuming you want the virtual interface to be created at boot time, set the **Activate at boot?** field to **Yes**.
7. Hit the **Create** button. As long as there are no errors in your input, you will be returned to the list of interfaces. Your new virtual interface will appear under its real parent in the **Interfaces Activate at Boot Time** section.
8. To activate the virtual interface immediately, click on its name, and then on the editing form click the **Save and Apply** button.

16.4 Configuring Routing

Any system attached to a large network needs to know the address of a default gateway, as explained in the introduction. In some cases, the system itself may be a gateway as well—perhaps forwarding data between a local area network and a dial-up or broadband connection. In this case, it must be configured to forward incoming packets that are destined for some other address.

In some cases, traffic destined for certain networks may have to be sent via another router instead of the default gateway. Or if more than one IP network shares the same LAN, traffic for any of those networks must be sent using the correct interface. If either of these are the case on your network, static or local routes need to be configured so that the system knows where to send packets for certain destinations.

To change the default gateway used by your system or enable packet forwarding, the steps to follow are:

1. On the Network Configuration module's main page, click the **Routing and Gateways** icon. This will take you to a form for configuring routing, which is unfortunately slightly different on each Linux distribution due to differences in the underlying configuration files.
2. Enter the IP address of the default gateway into the **Default router** field.
On Red Hat Linux versions 8 and above, this field and the one mentioned in the next step is replaced by a table of default routers and interfaces instead. This can be used to define a different gateway for each interface, which can be useful if your system does not always use the same one. In most cases, however, you should just select **Any** from the menu under **Interface** and enter the router's IP address into the field under **Gateway**.
3. Enter the name of the network interface that must be used to reach the default router into the **Default route device** field. On some Linux distributions this field is optional, meaning that the system will work it out automatically. On others, there is a **Gateway** field next to the **Default router** input.
4. To enable routing, set the **Act as router?** field to **Yes**.
5. On Red Hat, Mandrake, MSC and Turbo Linux, you can set up static routing using the **Static routes** table. For each static route, you must enter one row containing the following information:
 - In the **Interface** column, enter the interface that will be used to reach the router, such as *eth0*.
 - In the **Network** column, enter the address of the remote network, such as *192.168.5.0*.
 - In the **Netmask** column, enter the network's netmask, such as *255.255.255.0*.
 - In the **Gateway** column, enter the IP address of a router that knows how to forward data to the network, such as *192.168.4.1*.

6. On those same distributions, you can set up routing to additional IP networks on connected LANs using the **Local routes** table. For each route, you must enter one row containing the following details:
 - In the **Interface** column, enter the name of the interface that the LAN is connected to, such as *eth1*.
 - In the **Network** column, enter the address of the additional IP network, such as *192.168.3.0*.
7. Click the **Save** button when done. Any changes will not be activated immediately—instead, they will take effect when your system is next booted. On some Linux distributions, the module's main page will have a button labeled **Apply Configuration** at the bottom, which if clicked will activate the routing settings immediately. It will also make any boot-time network interfaces immediately active as well, so be careful when using it from a web browser on another system—your network connection may be cut off if the network configuration is incorrect.

If your system's primary network connection is via PPP dialup, then the default gateway will be assigned automatically when you connect and removed when you disconnect. Therefore there is no need to set it up using this form.

16.5 Changing the Hostname or DNS Client Settings

Every UNIX system has a hostname, which appears in the login prompt, system logs, outgoing email and on every Webmin page. Normally the hostname is the same as or part of the DNS name for the system's IP address, but this does not have to be the case, especially if the system is not connected to a network or only connects occasionally via dialup. However, for permanently connected systems the hostname should be the hosts fully qualified DNS name (like *server1.foo.com*), or just the first part (like *server1*). Anything else is likely to cause confusion and possibly network problems.

When a Linux system is first set up, you get to choose the hostname as part of the distribution's installation process. However, it can be changed at any time, either using Webmin, a GUI tool provided by the distribution, or the `hostname` command. To make the change in Webmin, the steps to follow are:

1. On the main page of the Network Configuration module, click the **DNS Client** icon. This will take you to the form for editing the hostname and DNS options shown in Figure 16.3.
2. Enter the new hostname (composed of letters, numbers, underscores, and dots) into the **Hostname** field.
3. Click the **Save** button to have it immediately changed. Your browser will be returned to the module's main page.
4. Change the host address for your old hostname to the new one, as explained in Section 16.6 “Editing Host Addresses”.
5. If you are running a DNS server, don't forget to update the entry for your system there as well.

As explained in the introduction to this chapter, in order to look up hostnames and IP addresses your system will almost certainly need to know the addresses of DNS servers on the network.

To change the system's DNS settings, follow these steps:

1. Click on the **DNS Client** icon on the main page of the module, which will take you to the form shown in Figure 16.3.
2. Enter the addresses of up to three servers into the **DNS servers** field. If the first is not available, your system will try the second, and finally the third. Most networks will have at least a primary and secondary DNS server to increase reliability in case one fails.
3. The **Resolution order** field can be used to control where your system will look when resolving hostnames and IP addresses. Generally the defaults are reasonable, with **Hosts** (the `/etc/hosts` file) listed first and **DNS** later. However, if you are using NIS for hostname resolution you will need to make sure it is selected somewhere in the order.
4. In the **Search domains** field, enter any domain names that you want your system to automatically append to resolved hostnames. For example, if `foo.com` was listed and you ran the command `telnet server1` then the IP address for `server1.foo.com` would be looked up.
5. When done, click the **Save** button. Any changes will take effect immediately in all programs running on your system.

If your system's only network connection is via dialup, the DNS servers may be assigned automatically by your ISP depending on your PPP configuration.

16.6 Editing Host Addresses

Host addresses are mappings between an IP address and one or more hostnames that are stored in the `/etc/hosts` file on your system. Because they are stored locally, they can be looked up at any time, even when a DNS server is not accessible. On a small network with only a few systems, you may choose not to run a DNS server at all, but instead keep the addresses of every system in the `hosts` file on each system. In fact, this is what was done in the early days of the Internet before DNS was developed.

To view the addresses on your system, click the **Host Addresses** icon on the module's main page. There will always be an entry for `localhost`, and probably one for your system's hostname as well. If your system's IP address or hostname has been changed, the host addresses list will probably not reflect the change, which could cause problems. To change a host address, the steps to follow are:

1. Click on its IP address from the list, which will take you to an editing form.
2. Enter the new address into the **IP Address** field.
3. Enter any hostnames into the **Hostnames** field. It is always a good idea to enter both the short and long forms of any hostname, such as `server1.foo.com` and `server1` so that both can be used.
4. Click the **Save** button, and if there are no errors in the form your browser will return to the list of hosts and addresses.

You can add extra host addresses by clicking the **Add a new host address** link above or below the link and filling in the same form. There are no restrictions on the same hostname being associated with two different IP addresses, or the same IP address appearing twice in the list.

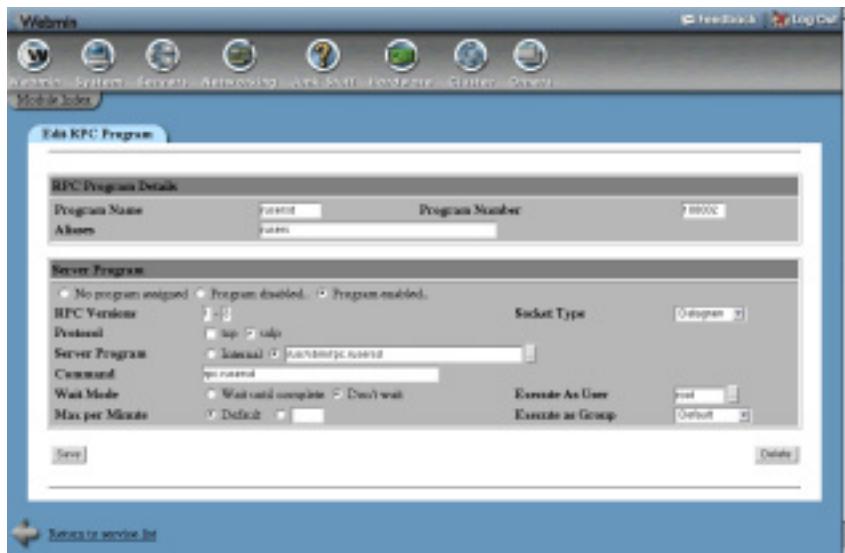


Figure 16.3 The DNS client and hostname form.

16.7 Module Access Control

As Chapter 52 explains, it is possible to limit the features of this module that a particular Webmin user or group can access. For example, you may want to allow a user to edit only the host addresses list, or to be able to view settings only instead of editing them. To do this, create or edit a Webmin user who has access to the module, and then follow these steps:

1. In the Webmin Users module, click on Network Configuration next to the name of the user or group that you want to restrict. This will bring up the module access control form.
2. Change the **Can edit module configuration?** field to **No**, so that the user cannot configure the module to edit a host addresses file other than /etc/hosts.
3. The **Can edit network interfaces?** field determines which interfaces the user can see and edit. Setting it to **Yes** allows editing of all of them, while choosing **No** prevents the Network Interfaces page from being accessed at all.
If **View only** is chosen, all interfaces will be visible but the user will not be able to change any of their attributes. If **Only interfaces** is chosen, only those whose names (separated by spaces) are entered into the field next to it will be editable. All others will be only viewable.
4. If the **Can edit routing and gateways?** field is set to **Yes**, the user will be able to set up the default router and static routes as normal. If **No** is chosen, the Routing and Gateways page will not be accessible at all, or if **View only** is chosen the current settings will be visible but not changeable.
5. Similarly, the **Can edit DNS client settings?** and **Can edit host addresses?** fields can be set to **Yes**, **View only** and **No** to control access to the DNS Client and Host Addresses pages respectively.
6. When you are done making selections, click the Save button to have the new restrictions immediately activated.

Be very careful giving an untrusted user the rights to edit any network configuration in this module, as he may be able to figure out a way to gain `root` access or disrupt other users by changing routes, host addresses, or interface settings.

16.8 Other Operating Systems

The Network Configuration module is also available on several other operating systems, with options fairly similar to those of Linux. Due to the different features supported by network configuration files on other versions of UNIX, in some sections the user interface is quite different. The supported systems and the variations between them and Linux are:

Sun Solaris and SCO UnixWare

- When editing a boot-time network interface, all that can be changed is the IP address.
- The boot-time settings for the loopback interface cannot be edited at all. Both operating systems always enable it at boot with the IP address `127.0.0.1`.
- On the routing and gateways page, multiple default routers can be entered. There is no need to specify a default route device though, as it is always worked out automatically.

FreeBSD and NetBSD

- There is no option to use DHCP to automatically assign an address for an interface at boot time.
- On the routing and gateways page, there is no default route device field. However, there is an additional **Start route discovery daemon?** option.
- The hardware address of an active interface cannot be changed.
- When creating a virtual interface, the netmask must be entered as `255.255.255.255`.

OpenBSD

- On the routing and gateways page, there is no default route device field. However, there is an additional **Start route discovery daemon?** option.
- The hardware address of an active interface cannot be changed.

Mac OS X

- On the routing and gateways page, only options for setting the default router and controlling it if this system acts as a router are displayed.
- On the DNS client page, no list of resolution orders appears, because the hosts file is always checked first, followed by DNS.

16.9 Summary

This chapter has explained how to perform basic network configuration on your Linux or UNIX system. After reading it, you should know how to specify IP addresses for network interfaces, define routes and choose an DNS server. Unless your system has only a single dial-up connection to the Internet, the information presented here is important if you want to connect your machine to a network.

Network Information Service

NIS is a protocol for sharing users, groups, and other information between multiple systems. This chapter explains how NIS works, and how to set up your system as either a client or server using Webmin.

17.1 Introduction to NIS

NIS was originally developed by Sun Microsystems, but is now available on Linux and many other UNIX operating systems. Its original name was YP (Yellow Pages), which is why many of the NIS commands start with `yp`.

On a network with many systems, users may be allowed to log in to any of those systems. Typically, to avoid having to create and update users on each system separately, NIS can be used to distribute a master list of users and groups to all hosts. Although distributing user and group information is the most common use of NIS, it can also be used to share hostnames and IP addresses, automounter maps, Internet services, and netgroups.

An NIS server is a system that stores tables of user, group, and other information. A client system connects to a server and queries it for stored information, usually by looking up usernames, hostnames, and so on. Normally a server system is also one of its own clients, so that it has access to the users and other data in its own tables.

Each server is responsible for a single NIS domain, and each client is a member of a domain. A domain has a short name, like `marketing` or `foo.com`, which is not necessarily the same as the network's DNS domain. When NIS is started on a client system, it can either broadcast for any server on the network for its domain, or connect to specific server IP addresses. A single network may have multiple NIS servers for different domains, each of which supplies different tables.

In order to reduce the load on the NIS server, a network may contain multiple servers that all have copies of the same tables. One is the master server and the rest are slaves, which just

receive information from the master whenever it is changed. A client can then connect to either the master or a slave and query the same tables.

In recent years, a new version of the old NIS protocol has been developed, called NIS+. It solves many problems with the original protocol, the biggest being lack of security. However, it is more complex to configure and not as widely available. For these reasons, Webmin supports only the configuration of NIS clients and servers.

The file `/var/yp/Makefile` is usually the primary configuration file for an NIS server, as well as a `make` script that generates binary format table data from source text files. The server also reads the files `/var/yp/securenets` and `/etc/ypserv.conf` to control which clients are allowed to connect, and which tables they can query. Webmin directly updates all of these files, along with the table source files, when you are configuring NIS. The primary NIS server program is called `ypserv`, but others such as `yppasswd` (for processing password change requests from clients) and `ypxfrd` (for sending tables to slaves) may be run as well.

On client systems, the file `/etc/yp.conf` stores the domain name and NIS server IP addresses. Information about which services to query NIS for is stored in `/etc/nsswitch.conf`. All clients run the program `ypbind`, which passes queries for user, group, and other information from local programs to the NIS server.

The NIS Client and Server Webmin module allows you to set up your system as an NIS client and/or server. When you enter it from the Networking category, the main page simply shows five icons for the different areas of client and server configuration. If Webmin detects that the NIS client programs are missing from your system, the main page will instead display an error message—if this happens, check your Linux distribution CD or website for a package named something like `ypbind`.

The module is not supported on all versions of Linux. At the time of writing, only Red Hat, Mandrake, OpenLinux, Debian, SuSE, UnitedLinux, and MSC.Linux could use it. Because each distribution uses slightly different configuration files for NIS, there may be some differences in the user interface and default settings between different distributions, in particular on the client services and NIS server pages.

17.2 Becoming an NIS Client

To set your system up as an NIS client, there must already be an NIS server running on your network. If not, see Section 17.3 “Setting Up an NIS Master Server” for information on how to start one. Assuming there is an NIS server running and you know its NIS domain name, the steps to become a client are:

1. On the module’s main page, click the **NIS Client** icon. This will take you to a form for entering the domain name and NIS server IP addresses.
2. In the **NIS domain** field, enter the name of your network’s NIS domain.
3. If you do not know the IP address of an NIS server, set the **NIS servers** option to **Find by broadcast**. This will work only if the server is on the same LAN as your system—if not, the broadcast will not be able to reach it.

If you do know the address of an NIS server, select the **Listed below** option and enter all the master and slave server addresses into the text box. The more you enter the better, because your system will try to query each of them in turn when NIS is enabled.

However, it is best to enter the nearest server first so that a more distant and thus slower server is not always queried.

4. Click the **Save and Apply** button to have your settings saved and immediately activated. If your system cannot contact a server for the NIS domain, an error message will be displayed—otherwise, the browser will return to the module's main page.
5. Now that you are connected to an NIS server, you must configure the system to actually query it for users, groups, and other information. To do this, click on the **Client Services** icon which will take you to the form shown in Figure 17.1.
6. Each row of the client services form controls what your system will query when looking something up for a particular service. For each, you can select several sources that will be checked in order until one finds a match. The available sources are:

Files Local configuration files, such as `/etc/passwd` or `/etc/hosts`.

NIS This NIS server that your system is currently connected to.

NIS+ The NIS+ server that your system is connected to. Configuring NIS+ is not supported by Webmin.

NIS and Files This option only works for the **UNIX users** and **UNIX groups** services. If chosen, special lines in `/etc/passwd` and `/etc/group` starting with + or - can be used to indicate that some or all NIS users should be included. This is actually more flexible than just choosing the **NIS** source, as special + and - lines can be used to bring in only some users and groups, or change the attributes of those that are included.

DNS This option makes sense only for the **Host addresses** source. It tells the system to query a DNS server when looking up hostnames, which is almost always what you want to do.

Typically, you should set each of the services for which you want to use NIS (such as **UNIX users** and **UNIX groups**) to **Files** and **NIS**. Everything else should be left set to just **Files**, or in the case of **Host addresses** just **Files** and **DNS**. Your system will then look in the local system configuration file first (such as `/etc/passwd`) and then query the NIS server.

7. When done, click the **Save** button. Your changes will take effect immediately in all programs, and any NIS users should be able to log in just as local users would.

Once you have used Webmin to make your system an NIS client, it will attempt to connect to a server at boot time. Failure to connect could cause the system to hang part way through the boot process, waiting for the server to become available. If the server goes down while your system is connected, any program that looks up user information may hang as well.

To stop your system from being an NIS client, the steps to follow are:

1. On the main page of the module, click the **NIS Client** icon to go to the client options page.
2. Set the **NIS domain** field to **None (NIS disabled)**.
3. Click the **Save and Apply** button. The system will no longer use NIS to look up any information, and will not connect at boot time. Any services that are configured to use an NIS source on the **Client Services** page will simply skip that source, and most likely use only local files instead.

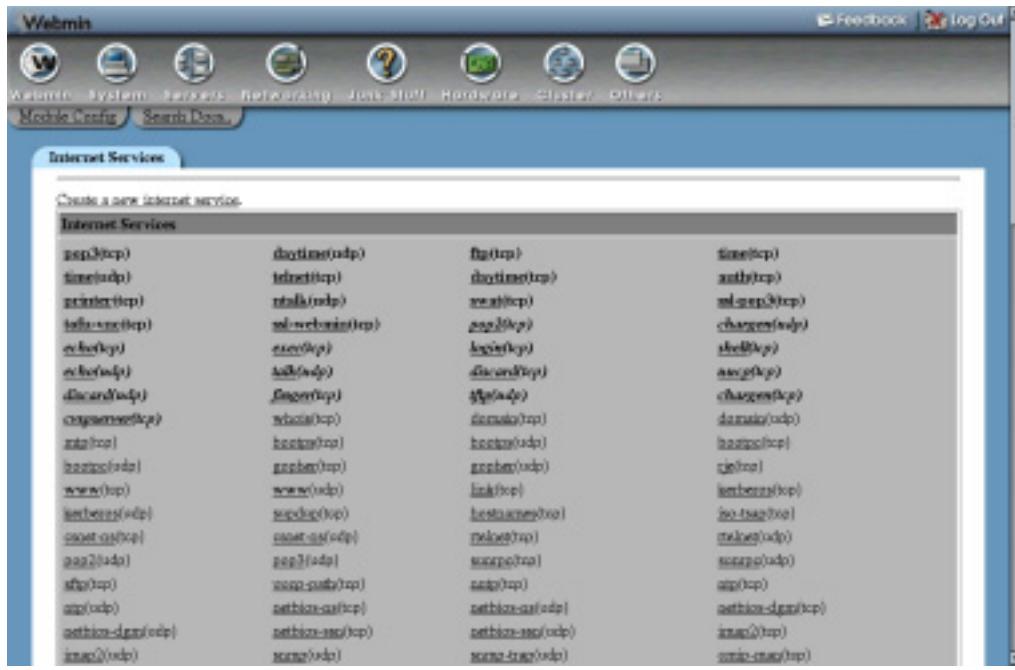


Figure 17.1 The NIS client services form.

17.3 Setting Up an NIS Master Server

Before your system can become an NIS, the appropriate server programs must first be installed—if they are not, when you click on the **NIS Server** icon an error message will be displayed. Check your Linux distribution CD or website for a *ypserv* or *nis-server* package, which should contain all the needed commands and files.

The first step in setting up an NIS server is deciding on a domain name. Typically, this will be the same as your Internet domain (such as *foo.com*), but anything made up of letters, numbers, and dots is allowed. After deciding, the steps to follow are:

1. On the module's main page, click on the **NIS Server** icon. This will take you to a form for enabling the server and configuring other options, as shown in Figure 17.2. The form will look the same on most Linux distributions, but on Caldera's OpenLinux it will have far fewer options.
2. Set the **Enable NIS server?** option to **Yes**. When the form is saved, the server processes will be started immediately and at each subsequent reboot.
3. Enter your chosen domain into the **Serve NIS domain** field. This is better than choosing the **Same as client** option, even if they are going to be the same.
4. Leave the **Server type** set to **Master server**. To set up a slave server, see Section 17.6 “Setting Up an NIS Slave Server”.

5. If NIS clients are incapable of looking up hosts and addresses in DNS themselves, turn on the **Lookup missing hosts in DNS?** option to have the master server do lookups for them. Only very old client operating systems like SunOS 4 need this.
6. In the **NIS tables to serve** field, select all the tables that you want to make available to clients. Some of the most commonly used tables and their contents are:
 - passwd** UNIX users, as stored in the `/etc/passwd` file. Normally this contains passwords as well, instead of having them stored in a separate shadow table.
 - group** UNIX groups, as normally found in the `/etc/group` file.
 - hosts** Hosts and IP addresses, as found in the `/etc/hosts` file. Even though NIS can be used to store and lookup hostnames and addresses, it is almost always better to set up a DNS server instead.
 - shadow** Additional user information, including passwords. If this table and **passwd** are selected, depending on your NIS `Makefile` configuration you may be able to edit extended user information, such as expiry and warning dates.
 - netgrp** Netgroups, which are groups of hosts. These can be used when exporting directories via NFS, as explained in Chapter 6.
7. If your network will have slave servers, it is advisable to set the **Push updates to slaves?** option to **Yes**. This way whenever a change is made to one of the NIS tables, all slave servers will be notified immediately so that they are in sync.
8. Enter the IP addresses of any slaves (separated by spaces) into the **Slave servers** field.
9. In the **Master NIS files** section, you can choose which files will be used as the sources for the NIS tables. Often by default the normal user, group, host, and other configuration files in `/etc` will be used, such as `/etc/passwd`, `/etc/group`, and `/etc/hosts`. This is not a good idea, though—instead, you should change the files for the tables that your server is using to similar filenames in the `/var/yp` directory, such as `/var/yp/passwd` and `/var/yp/group`. Once the server is running, it can be configured to become one of its own clients and so have access via NIS to any records in these files, instead of accessing them locally.
10. When done, click the **Save and Apply** button. The NIS server will be started on your system, and be configured to start at boot time in future.

Now that the server is running, you can test it by configuring some other system as an NIS client for the chosen domain. Server settings on the form can be changed at any time by simply repeating the same steps, and they will become effective immediately.

To shut down your NIS server, the steps to follow are:

1. Make sure any clients are no longer using your system as a server, either by turning off NIS on them altogether or having them use a different server.
2. On the module's main page, click on the **NIS Server** icon to go to the server options form.
3. Set the **Enable NIS server?** field to **No**.
4. Click the **Save and Apply** button. The server processes on your system will be shut down, and prevented from starting at boot time in future.

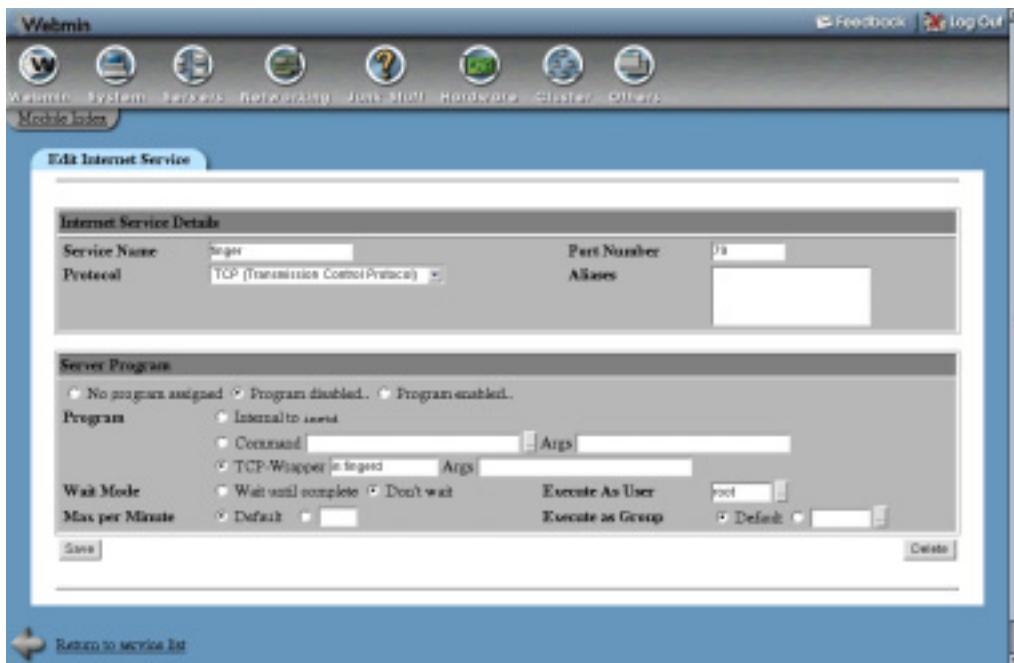


Figure 17.2 The NIS server configuration form.

17.4 Editing NIS Tables

Once your system is running as an NIS master server, you can use this Webmin module to edit records in the tables that it is serving. To see the editable tables, click on the **NIS Tables** icon, which will take you to a page with a menu of all tables and the contents of one displayed. Other tables can be shown by selecting one of them from the list and clicking the **Edit NIS table** button.

For most table types, Webmin will parse the contents of their files and display them as a table on the page, with one record per row. You can edit any record by clicking on its name in the first column, or add a new one by clicking the **Add a new record** link. However, some tables are in a format unknown to Webmin and so will be shown as raw text in a text box instead. If you know the correct format, the table can be manually edited and saved with the **Save and Apply** button. You can also switch any table to manual mode by clicking the **Edit table manually** link, if you prefer to work with the raw text.

The fields that exist in each record and the form for editing them are different for each type of table. The instructions below explain how to add, delete, and modify records in several frequently used tables. One commonality is that any changes will cause the NIS table to be automatically rebuilt from the changed source files, and pushed out to slave servers if configured to do so.

To create a new UNIX user for NIS clients, the steps to follow are:

1. Select the **UNIX users** table from the menu and click the **Edit NIS table** button.
2. Click the **Add a new record** link above or below the table of existing users, which will take you to the user creation form.

3. Enter the user's name into the **Username** field, and an ID number for the new user into the **User ID** field. Unlike in the Users and Groups module, the ID will not be automatically chosen for you, so make sure it is unique.
4. Enter the user's full name into the **Real name** field.
5. Enter a home directory into the **Home directory** field. Unlike in the Users and Groups module, this will not be created for you and files will not be copied into it.
6. Select a shell from the **Shell** menu, or select the **Other** option and enter the path to the shell program into the field below.
7. Select the **Normal password** option for the **Password** field, and enter the new user's password into the text field next to it.
8. Enter the numeric ID of the user's group into the **Primary group ID** field.
9. If the shadow NIS table is enabled, you can set the optional **Expiry date**, **Minimum days**, **Maximum days**, **Warning days**, and **Inactive days** fields. These all have the same meanings as in the Users and Groups module, covered in Chapter 4.
10. When done, click the **Create** button to have the new user added to the table.

Existing UNIX users can be edited by clicking on their names in the table, which will take you to an editing form with all the same fields as described above. Change any of the fields, and click the **Save** button—or to delete the user, click the **Delete** button at the bottom of the form. When deleting, the user's home directory will not be touched, so you may need to delete it manually.

To create a new UNIX group in NIS, the process is as follows:

1. Select the **UNIX groups** table from the menu and click the **Edit NIS table** button.
2. Click the **Add a new record** link above or below the table of existing groups, which will take you to the user creation form.
3. Enter a name for the new group into the **Group name** field, and a numeric ID into the **Group ID** field. Make sure that the ID is not used by any other existing group.
4. The **Password** field can be left untouched, as group passwords are almost never used.
5. Fill in the **Group members** field with the usernames of users who will be members of the group, one per line.
6. When done, click the **Create** button to have the new group added to the table.

As with users, you can edit a group at any time by clicking on its name from the table, which will take you to an editing form. Make any changes that you want, and click the **Save** button to save them—or use the **Delete** button to remove the group. Note that no checking will be done to see if it is the primary group of any existing users.

As the instructions for editing users and groups show, the process for editing any of the supported tables is quite similar. Currently, you can edit **UNIX users**, **UNIX groups**, **Host addresses**, **Networks**, **Services**, **Protocols**, **Netgroups**, **Ethernet addresses**, **RPC programs**, **Netmasks** and **Aliases** using forms in Webmin. All other tables must be edited manually.

17.5 Securing Your NIS Server

By default, an NIS server allows any client to connect to it and query tables, as long as the client knows the domain name. If your system is connected to the Internet, an attacker could guess the NIS domain and request a list of all NIS users. Even though their passwords are stored in

encrypted format, it is still possible for obvious or dictionary word passwords to be discovered by a brute-force attack on the password encryption.

For this reason, it is wise to limit the addresses of clients that connect to the server to only those UNIX systems that are really clients. To set this up, the steps to follow are:

1. On the main page of the module, click on the **Server Security** icon, which will take you to the form shown in Figure 17.3.
2. The rows in the **Allowed clients** table control which clients are allowed to connect. You can modify any of the existing entries, or use the empty row at the bottom to add a new one. To add more than one row, you will have to add them one at a time, saving and opening the form for each one.

To grant access to a single host, under the **Netmask** column select **Single host**, and enter its IP address under **Network/host address**.

To grant access to an entire IP network, select **Netmask** and enter a netmask (such as **255.255.255.0**) into the field next to it, and the network address under the **Network/host address** column.

To grant access to all clients, just select the **Any host** option under the **Netmask** column.

3. When done, click the **Save and Apply** button. The new restrictions will take effect immediately, and you will be returned to the module's main page.

It is a good idea to let only clients on your own network connect, and deny all others. An even more secure alternative would be to allow only those systems that you know are NIS clients, assuming they have fixed IP addresses and do not change often.

Even if you restrict access to only trusted client systems, users who can log in to those systems via SSH or telnet may still be able to get a list of all NIS users and their encrypted passwords. To prevent this, it is possible to configure the server to allow only clients using trusted ports to access certain tables or fields within tables. Because on UNIX systems only the **root** user can create TCP or UDP sockets with port numbers below 1024, these low ports are considered trusted and safe from use by regular users.

It is also possible to prevent certain clients from accessing some NIS tables, but still allow them access to others. For example, you might want to give all client systems access to the **Host addresses** table, but only a trusted few the rights to the **UNIX users** table.

To restrict access to tables on your server, the steps to follow are:

1. On the main page of the module, click on the **Server Security** icon to get to the form shown in Figure 17.3.
2. The **Client map restrictions** table controls which NIS tables can be accessed by certain client systems, and who on those systems can access them. Each row specifies a rule that applies to some or all clients, and can either allow access, block it entirely or filter the a queried table. The fields and their meanings are:

Hosts An IP address or partial IP address (like **192.168.1.**) that this restriction applies to. Entering ***** will make the restriction apply to all clients.

NIS tables Select the **All** option to have the restriction apply to all tables, or enter a single table name. Internally, the NIS server appends something like **.byname** or

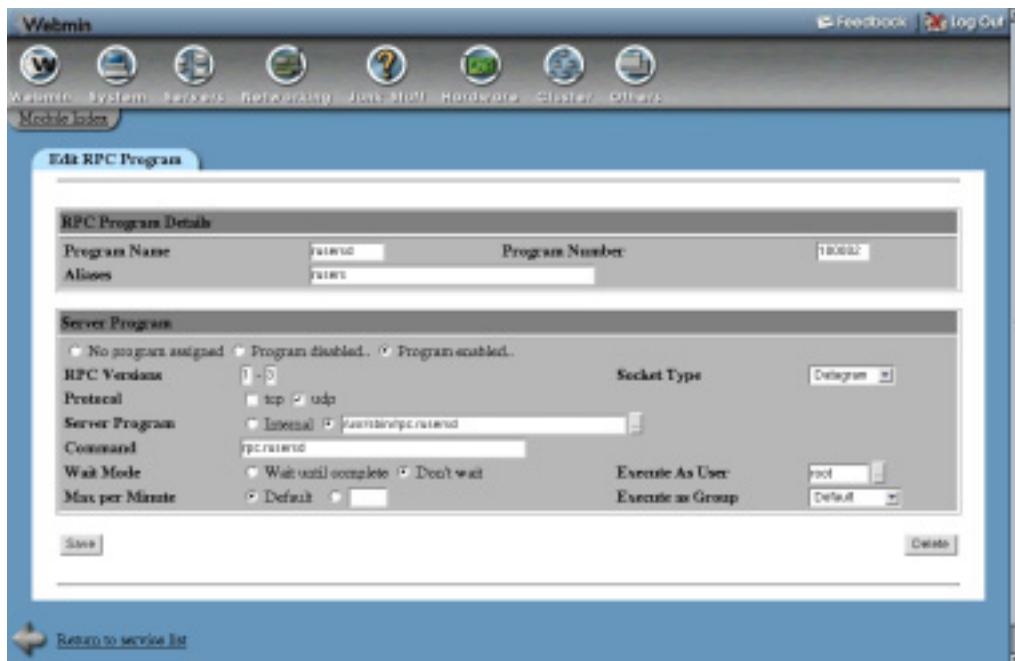


Figure 17.3 The server security form.

.byuid to table names to indicate what they are indexed by. The table name that you enter must use this internal name, such as *passwd.byuid* or *hosts.byaddr*.

Restriction This field controls what the server does if a client request matches. Select **None** to allow the request, **Deny access** to block it altogether, or **Trusted port** to block if the client is using an untrusted port.

Mangle field If using the **Trusted port** restriction, you can use this option to hide only a single field of the requested table from the client. Selecting **None** will block access to the table altogether, but entering a field number will cause its contents to be replaced with an x. The only practical use of this option is hiding passwords in the *passwdbyname*, *passwd.byuid* or *shadowbyname* tables, which are in field 2.

New restrictions can be added using the empty row at the bottom of the table. To add more than one restriction, you will need to save and re-open the form multiple times.

When a client requests a table, the NIS server will find the first row in the table that matches and use the restriction defined. For this reason, you must make sure that any new row you add is before the one that grants access to all clients and tables, which usually exists by default.

3. When done, click the **Save and Apply** button. The browser will return to the module's main page, and any changes to restrictions will take effect immediately.

The most useful use of the **Client map restrictions** table is to add a row for all clients on the *passwdbyname* table, with the restriction set to **Trusted port** and the **Mangle field** option set

to 2. Then add another row for the `passwd/byuid` table, with all other options the same. These will prevent non-root users from seeing encrypted passwords, while still allowing programs running as root, such as the telnet or SSH server.

If you are using the separate shadow table to store passwords and expiry information, the restriction should be on the `shadow/byname` table instead. On many Linux distributions, a restriction like this exists by default.

17.6 Setting Up an NIS Slave Server

Slave NIS servers are used in a similar way to secondary DNS servers—they keep a copy of the tables held by the master server, and they can be used by clients if the master fails or is slow to respond. If you are using NIS on a very large network that has multiple LANs connected by slow links, it may also make sense to put a slave server on each LAN so that clients can use it instead of the master.

On OpenLinux, there is no way to setup a slave server using Webmin, due to the unique NIS configuration files used by the distribution. On all other versions of Linux, the steps to set up a system as a slave server are:

1. On the module's main page, click on the **NIS Server** icon. This will take you to the server configuration form, shown in Figure 17.2.
2. Set the **Enable NIS server?** field to **Yes**.
3. Enter the master server's domain into the **NIS domain** field.
4. Change the **Server type** to **Slave of server**, and enter the IP address of the master into the field next to it. None of the other fields need to be touched, because they all relate to running a master server.
5. Click the **Save and Apply** button. The server should be started immediately, and configured to start at boot time.

Make sure that the master server has the address of this slave entered into the **Slave servers** field on the server configuration form. It should also have the **Push updates to slave servers?** option enabled, so that any changes to tables will be immediately sent to the slaves. If not, you can use the `yppush` command to send the contents of an NIS table to some or all slave servers.

17.7 Configuring the NIS Client and Server Module

The module has a few configurable options that can be changed by clicking on the **Module Config** link in the top left corner of the main page. The ones that you can safely change shown in Table 17.1.

The other options on the module configuration page are set automatically based on your operating system, and generally do not need to be changed.

17.8 NIS on Solaris

The only other operating system that Webmin allows you to configure NIS on is Sun's Solaris. On Solaris, the **NIS Client** and **Client Services** page are identical to those on Linux, and work in the same way. However, the **NIS Server** and **Server Security** forms are slightly different:

Table 17.1 Module Configuration Options

Maximum number of records to display	If the number of records in an NIS table exceeds this number, then the NIS Tables page will not display all of the records. Instead, it will show a search form for finding records based on a field and search term.
Automatically rebuild NIS maps?	Normally, any time you add, update, or delete a record in a table file, the NIS Makefile is run to rebuild all tables. Because this can be slow on systems with many records or slave servers, setting this option to No will turn off the automatic rebuild. Instead, you can force a build whenever you want by clicking the button labeled Rebuild NIS tables on the NIS Tables page.

- On the **NIS Server** page, whatever domain you enter will also be used for the NIS client as well. This is a limitation of Solaris—unlike Linux, where a system can be a server for one domain and a client of another.
- On the server page, you cannot specify the paths to individual table files directly. Instead, the **NIS source files directory** and **NIS password source files directory** fields control which directories they are stored in, usually `/var/yp`.
- There is no **Client map restrictions** table on the **Server Security** page, and so no way to control which tables and fields clients can request. However, you can still allow or deny certain hosts and networks entirely using the **Allow clients** table.

Solaris systems include client and server support for NIS+ as standard. However, because that protocol is not supported by Webmin, attempting to use this module to reconfigure a system that is already running as an NIS+ client or server will not work, and may even cause problems with its configuration.

17.9 Summary

NIS is the standard way of sharing user, group, and other information between UNIX systems, and this chapter has explained how to set up and configure it using Webmin. After finishing it, you should know how to make your system a client of an existing NIS server on your network. You should also understand how to run a master NIS server of your own to serve other clients, and how to set up a slave to act as a backup for master.

PPP Server Configuration

This chapter covers the process of setting up a Linux system with an attached modem as a dial-in server, so that other computers can dial up to it and access connected networks.

18.1 Introduction to PPP on Linux

Any Linux system with a modem attached can be configured so that other computers can dial up to it and start a PPP session, giving them TCP/IP access to the system and any networks that it is connected to. This allows it to act like a miniature ISP, and in fact some small ISPs have been run using Linux systems with multiple serial port cards as access servers.

Two separate programs are responsible for different parts of the dial-in service. The first is `mgetty`, which communicates on a serial port with an attached modem and instructs it to answer the phone. Once the server and client modems are connected, `mgetty` displays a text login prompt and waits for either a username or the start of a PPP session. A client can log in using text mode and get a UNIX shell prompt without needing to start a PPP session at all, but this is rarely done these days. Once the client disconnects or logs out, `mgetty` hangs up the modem and waits for a new connection.

Because most clients start a PPP session as soon as they connect, `mgetty` is usually configured to run the separate `pppd` program if it detects a PPP connection. This creates a `ppp` network interface on the server, authenticates the client, assigns an IP address, and starts sending and receiving data using the PPP protocol. The assigned IP address and other configuration options are usually set on a per-serial port basis, so that you can have multiple modems and support several simultaneous clients with different addresses.

The PPP Dialin Server module allows you to setup both `mgetty` and `pppd` so that clients can dial in and start PPP sessions. When you enter it from the Networking category, the main page simply shows four icons, under which are the actual configurable options.

Currently, the PPP Dialin Server module can be used only on Linux and Solaris systems, even though `mgetty` is available on some other versions of UNIX. If neither of the programs that it configures are installed, the main page will display an error message—however, all Linux distributions include packages for `pppd` and `mgetty` on their CDs or websites. If only `mgetty` is installed, you can use the **Serial Port Configuration** and **Caller ID Access** features. Conversely, if only `pppd` is installed, you can access only the **PPP Options** and **PPP Accounts** pages.

When you use the module to set up `mgetty` to answer calls on a serial port, an entry is added to the `/etc/inittab` file so that `init` will run the `mgetty` process at boot time, and re-run it as necessary. You will be able to see this entry in the SysV Init Configuration module (covered in Chapter 9), but you should not edit it there unless you know what you are doing.

Even though this chapter was written with Linux in mind, the module behaves almost identically on Solaris. The only difference is the names of the serial port device files—whereas `/dev/ttys0` is the first serial port on Linux, Solaris would use `/dev/term/a` instead.

18.2 Configuring a PPP Server

Before you can set a system up to allow clients to connect with PPP, it must either have a modem attached to a serial port, or be connected via a null-modem cable to another machine. Internal modems that emulate a serial port can be used as well, although they are not recommended as they do not have easily visible LEDs to indicate if the modem is connected, transmitting, and so on. USB modems should work, as long as they are recognized by the kernel—however, they will probably use a special device file. Modems that require special drivers to operate (commonly known as Winmodems) cannot be used at all, unless there is a driver for the modem available for Linux.

Naturally, any modem must be connected to a phone line. Because your system will be configured to answer the phone after a few rings, the phone line should not be used for anything else—otherwise, voice callers will have their calls answered by the modem, which is not very friendly.

Once all the hardware is ready, the steps to set up your system as a PPP server are:

1. On the main page of the module, click on the **Serial Port Configuration** icon. This will take you to a page listing any existing ports that have been configured for PPP or voicemail.
2. Click on the **Add a new serial port** link, which will bring up the port configuration form shown in Figure 18.1.
3. Set the **Serial device** to the port on which your modem or null-modem cable is connected. **Serial port 1** corresponds to the device file `/dev/ttys0`, and so on. For modems on serial devices not starting with `/dev/ttys` (such as USB modems), select the **Other device** option and enter the full device file path into the text field next to the menu.
4. Set the **Type** option to either **Direct connection** (for a system connected via null-modem cable), or **Modem** (for an actual dial-in modem).
5. The **Port** speed field should be set to the baud rate that the modem or null-modem connection will use. This must be one of the standard speeds, such as 57600 or 33600.
6. In the **Answer** after field, enter the number of rings that you want `mgetty` to wait for before answering the phone. If the phone line your modem is on will be also used for receiving voice calls, you could set this to something large like 20 to give yourself plenty of time to answer the phone before the modem does.

Naturally, this option has no meaning for null-modem connections.

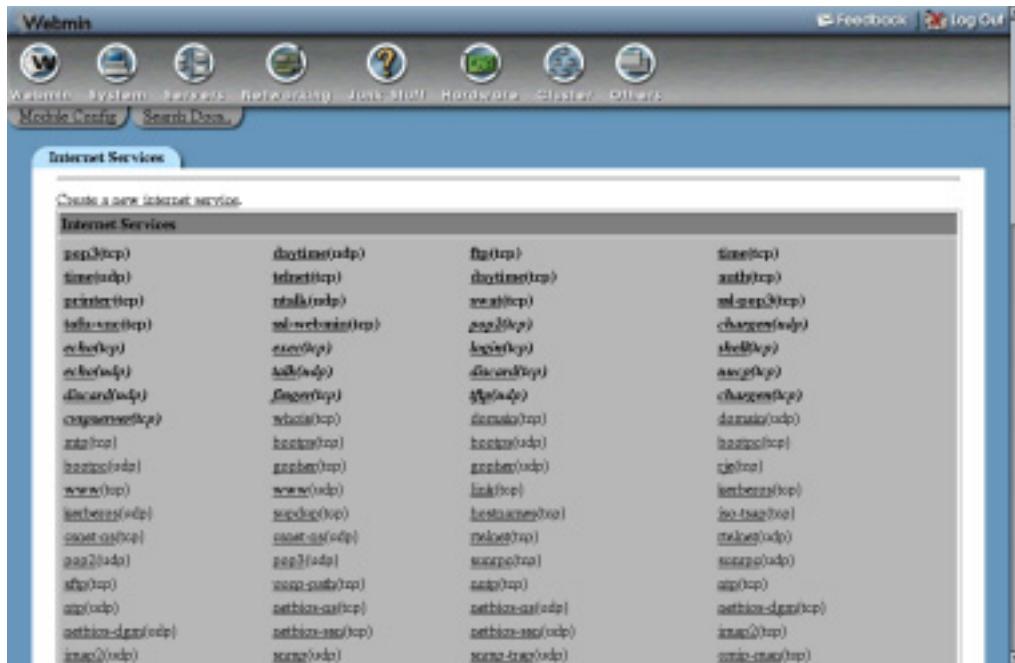


Figure 18.1 The serial port configuration form.

7. Click the **Create** button. A new entry will be added to the `/etc/inittab` file, and you will be returned to the serial ports list.
8. Click **Apply Configuration** to activate `mgetty` on the new port. Phone calls to the line your modem is on should now be answered after the configured number of rings.
If you care only about text-only clients, then nothing more needs to be done—they will be able to dial up, authenticate at the login prompt and execute shell commands.
9. To set up PPP, click on the **PPP Options** icon back on the main page. This will take you to the form shown in Figure 18.2, where you can set options that will apply to all PPP connections.
10. Unless you want clients to log in using text mode and start the `pppd` command manually, it is best to set the **Automatically detect PPP connections on serial ports?** option to **Yes**. With this enabled, `mgetty` will detect that the client wants to start a PPP session when the server is waiting for a log-in prompt, and run `pppd` automatically.
11. In the **PPP IP Address** fields, enter the IP address that you want the server's end of the connection to use (the **Local IP**) and the address for the client's end of the connection (the **Remote IP**). Normally these addresses will not be on your local LAN, but on a different subnet. Other systems on the network should be configured to route traffic for the client's address to your system, so that they can communicate.
If no addresses are specified, then the PPP server will use whatever addresses are supplied by the client. This might make sense when connecting two machines via null-modem, but will not work with most dial-up clients.

It is possible to assign the client an IP address that is within the range of the local LAN, by turning on the **Create proxy ARP entry?** option. If this is enabled, enter an unused LAN IP address into the **Remote IP** field and your system's current Ethernet IP into the **Local IP** field.

12. Set the **Control lines mode** field to **Local** for a null-modem connection, or **Modem** if there is a real modem connected to the serial port.
 13. Unless you are setting up a null-modem connection, clients should be forced to authenticate to prevent potential attackers from connecting. To turn on authentication, set the **Require authentication?** field to **Yes**. To turn it off totally for null-modem use, set the field to **No**.
- To set usernames and passwords for clients to authenticate against, see Section 18.3 “Managing PPP Accounts”.
14. To disconnect clients that have been idle for a long period, enter a number of seconds into the **Idle time before disconnect** field.
 15. Enter the IP addresses of any DNS servers on your network into the **DNS servers for clients** field. Client operating systems like Windows will use them automatically, which simplifies their configuration.
 16. Finally, click the **Save** button. Clients should now be able to dial in, establish a PPP session and access your system and network.

If your system is going to have multiple simultaneous PPP clients connected, then you will need to set different options for each serial port. In particular, each client must have a different remote IP address, although the local address can be re-used.

To set up different PPP options for each serial port, the steps to follow are:

1. On the module's main page, click on the **PPP Options** icon. Change the **PPP IP Addresses** field back to **From client**, and change any other options that you want set on a per-port basis back to their defaults as well.
2. Go back to the main page, click on **Serial Port Configuration** and then on the **Edit** link under **Port PPP Config** for the serial port that you want to set options for. This will take you to the per-port options page, which is very similar to the global PPP options form shown in Figure 18.2.
3. Enter remote and local IP addresses to which you want PPP clients connecting on this port to be assigned, and change any other options that have not been set on the global PPP options page.
4. When done, click the **Save** button. Clients connecting on the configured port will use the new options from now on.

The easiest way to stop your system from acting as a PPP server is simply to remove the serial port configuration entry for your modem. If you have multiple modems attached, the steps below can be used to disable one without any effect on the others:

1. On the main page, click on **Serial Port Configuration** and then on the device name of the port with the attached modem.

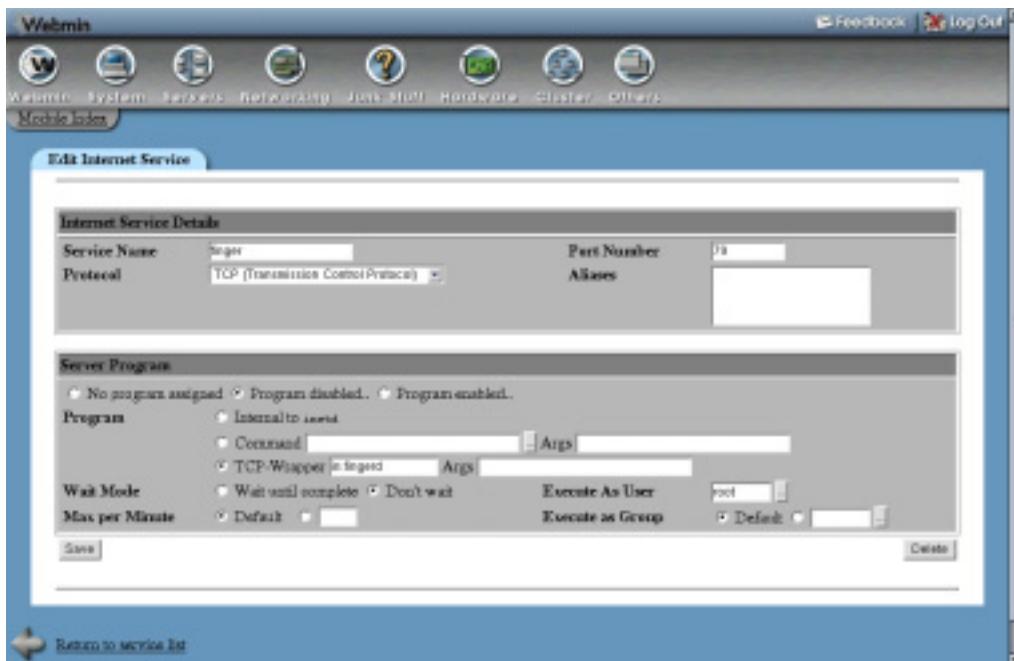


Figure 18.2 The PPP configuration form.

2. On the port options page, click the **Delete** button in the lower-right corner. The appropriate entry will be removed from the `/etc/inittab` file, and you will be returned to the list of enabled ports.
3. Click the **Apply Configuration** button to make the change active. From now on, your system will no longer answer incoming phone calls or communicate with another computer attached by a null-modem cable.

18.3 Managing PPP Accounts

If you enable dialin access to your system, you should force all clients to authenticate themselves by turning on the **Require authentication?** option on the **PPP Options** page. Even if you think that your sever doesn't need to authenticate clients because only you know the phone number of the line your modem is on, it is still a good idea to enable it in case someone stumbles across the number by accident—or in case a “war dialer” trying out hundreds of phone numbers in search of insecure servers finds it.

Once authentication is enabled, you can add a new account that is allowed to log in by following these steps:

1. On the main page of the module, click on the **PPP Accounts** icon. This will take you to a page listing all existing accounts, including those that have been created for dialing out to other servers.
2. Follow the **Create a new PPP account** link, which will bring you to the account creation form shown in Figure 18.3.

3. Enter a login name into the **Username** field, and make sure its **Any** option is not selected.
4. Make sure the **Server** field is set to **Any**. If you set it to something else, then the user-name will be accepted only when the client's hostname matches whatever you enter.
5. Select the **Set to** option in the **Password field**, and enter a password for the account into the text field next to it. It is also possible to have the PPP server read the password from a separate file, by selecting the **From file** option and entering a file name into its text field. Or you can remove the need for a password to be supplied at all, by selecting **None**—however, this isn't a very good idea from a security point of view.
6. Assuming that all clients are being assigned IP addresses, set the **Valid Addresses** field to **Allow any**. However, if no addresses are specified in the **PPP Options** page, you may want to select **Allow listed** and enter acceptable addresses into the text box below it.
7. Finally, click the **Save** button and the new PPP account will be created. It can be used immediately by connecting clients.

To edit an existing PPP account, just click on its username from the accounts list. This will bring you to the account editing form, which is almost identical to the creation form shown in Figure 18.3. Change the username, password, or any other options, and click the **Save** button to save your changes and make them immediately active. Or click the **Delete** button on the editing form to remove the account instead.

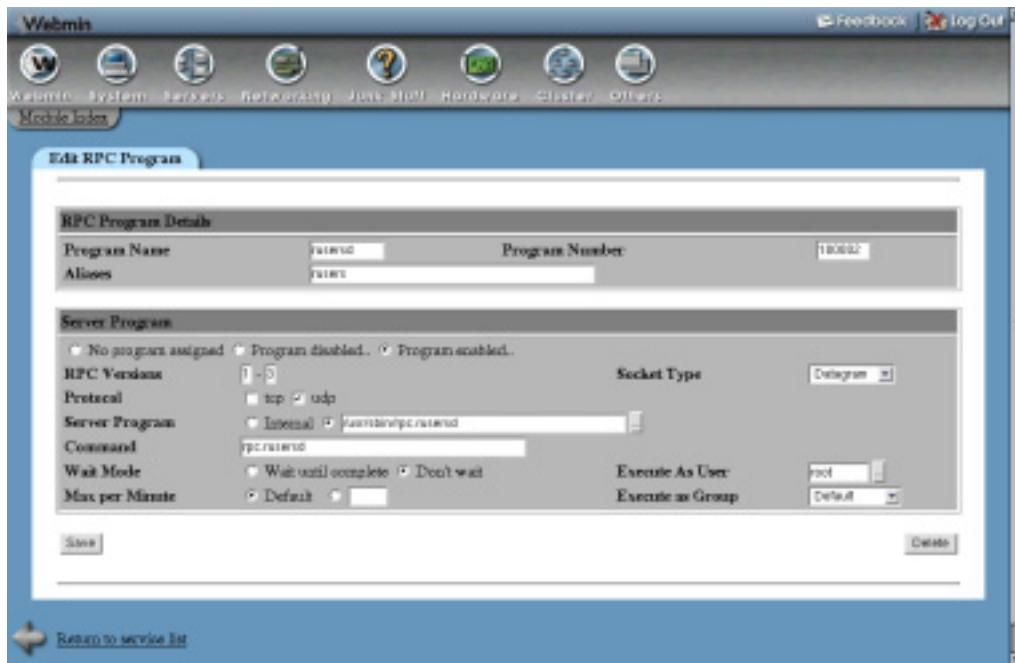


Figure 18.3 Creating a new PPP account.

By default, Webmin will add new users to the `/etc/ppp/pap-secrets` file. This is only read by the PPP server when doing PAP authentication, which is used by default. If you have manually configured your system to authenticate clients using the more secure CHAP protocol instead, you will need to configure Webmin to edit the `chap-secrets` file instead. This can be done by clicking on the **Module Config** link in the top left corner of the main page, and changing the **PAP secrets file** field to `/etc/ppp/chap-secrets`.

18.4 Restricting Access by Caller ID

If your phone line has caller ID enabled and your modem supports it, `mgetty` can be configured to block certain callers based on their phone numbers. By default, any caller will be allowed to connect—but you can change this so that only a few numbers are allowed by following these steps:

1. On the main page of the module, click on the **Caller ID Access** icon. This will take you to a form listing restricted numbers, which will probably be empty if you have not added any yet.
2. Click on the **Add a new caller ID number** link, which will take you to a form for entering the new number.
3. Set the **Phone number** option to **Numbers starting with**, and in the field next to it enter a partial or complete phone number that you want to allow. If you enter something like just 555, any caller whose phone number starts with 555 (such as 555-1234) will be allowed.
4. Set the **Action** field to **Allow**.
5. Click the **Create** button, which will save the number and return you to the list of those that are allowed and denied.
6. To add another allowed number, repeat Steps 2 through 5.
7. Finally, click on **Add a new caller ID number** again and on the creation form set **Phone number** to **All numbers** and the **Action** to **Deny**.
8. Click the **Create** button to have this final deny entry added to the list. From now on, only the phone numbers that you explicitly allowed will be able to connect.

Because the system checks each entry in the list in order and stops when it finds one that matches, any entry that denies (or allows) all callers must appear at the bottom of the list—otherwise, those after it will never be processed. If you want to allow a new phone number in the future, you must add it, then use the arrows in the **Move** column to move it above the final entry that denies everyone.

Because some clients may not provide caller ID information, the **Unknown numbers** option for the **Phone number** field can be used to match their calls. Allowing all unknown callers is not a good way to block known attackers though, as they may just disable the sending of caller ID information on their phone line.

Caller ID restrictions should never be the only form of security on your dial-in server, as caller numbers are supplied by the phone company and thus not totally under your control. PPP authentication should be enabled as well, so that all clients are forced to log in.

18.5 Module Access Control

Like others, this module has several options that you can set in the Webmin Users module to control which of its features are available. They are most useful for disabling parts of the module that are no use on a particular system—for example, you may want only the **PPP Accounts** page to be visible for a certain user.

To edit access control options in this module for a user or group, the steps to follow are:

1. In the Webmin Users module, click on **PPP Dialin Server** next to the name of a user who has been granted access to the module.
2. For the **Available pages** field, de-select those icons on the module's main page that you don't want the user to be able to access. If **PPP Options** is de-selected, the user will not be able to edit the options that apply to a single serial port either.
3. If the user is granted access to only a single page, setting the **Go direct to one page?** field to **Yes** will cause the browser to jump directly to that page when the module is entered.
4. Click the **Save** button to make the access control settings active.

18.6 Summary

After reading this chapter, you will be able to set up any Linux system with an attached modem as a dial-in server to which PPP clients can connect. You should understand the security implications of doing this, and how access can be restricted to only trusted systems. You should also know how a dial-up connection like this fits in with, and is accessible from, the rest of your network.

Firewall Configuration

If your system is connected to the Internet, it should be protected with a firewall to prevent unauthorized access. This chapter covers the process of setting up and configuring a firewall with Webmin and IPtables.

19.1 Introduction to Firewalling with IPtables

A firewall is a system that protects itself and other hosts on a network from attackers on untrusted networks, such as the Internet. It can block packets and connections based on a variety of criteria, such as the source address, destination address, port, and protocol. Typically a firewall is also a router, forwarding packets between a secure local network and the untrusted Internet—however, it is also possible for a system to protect just itself.

A firewall system can also be configured to hide multiple hosts behind a single IP address, using a process known as NAT (Network Address Translation). Typically, the hidden hosts are on an internal LAN using a private IP network (such as 192.168.0.0) and the firewall has a single Internet IP address. NAT allows these internal hosts to communicate with others on the Internet, even though they do not have real public IP addresses.

The Linux kernel has included several different firewall implementations over the years, such as IPfwadm and IPchains. The 2.4 series of kernels include the IPtables firewall, which is more powerful and flexible than its predecessors. All Linux distributions that use the 2.4 kernel have IPtables support enabled, and include the commands needed to configure it. This chapter and the Linux Firewall module cover only the setting up of a firewall using IPtables, not any of the older implementations like IPchains or IPfwadm.

All IP network traffic is broken up into packets, which are chunks of data with a source, destination, and protocol information. Even a continuous flow of data such as the download of a large file is broken into packets when sent, and re-assembled at its destination. Because the IPta-

bles firewall operates at the IP level, all of its rules and chains evaluate and operate on individual packets, not TCP connections or HTTP requests.

An IPtables firewall is made up of three different kinds of objects—tables, chains, and rules. Each of the three tables contains two or three standard chains, and possibly many user-defined custom chains. Each chain contains zero or more rules, which are applied to packets received by or sent out from the firewall to determine their fate. The three tables and their standard chains are:

Packet filtering (filter) The INPUT, OUTPUT, and FORWARD packets chains in this table apply to packets received by, sent out from, or forwarded by the firewall, respectively. If the firewall system is acting as a router, only the FORWARD chain applies to routed packets. Network traffic destined for the system itself is processed by the INPUT chain, and traffic sent out by local process by the OUTPUT chain.

For a system that is an ordinary router and not doing any masquerading, or a system that needs a firewall only to protect itself, this is the only table that rules need to be added to.

Network address translation (nat) This table is used only for packets that start a new connection. The rules in its PREROUTING chain are applied to packets as soon as they are received by the system for routing, and the POSTROUTING for packets about to leave after routing. The OUTPUT chain rules are applied to locally generated packets for modification before routing.

Rules are typically added to this table to set up masquerading, transparent proxying, or some other kind of address translation.

Packet alteration (mangle) This table is used only for specialized packet alteration. It contains two chains—PREROUTING for modifying packets before routing, and OUTPUT for modifying locally generated packets.

This table is rarely used in a typical firewall configuration.

When a network packet is processed by a chain, each rule in the chain is executed in order. Every rule has a set of conditions that determine whether the rule matches or not, and an action that is taken in the case of a match. This action may be to immediately accept the packet, immediately drop it, perform some modification, or continue execution. If the end of a chain is reached, its default action will be taken instead, which is usually to allow the packet through.

Figure 19.1 shows the tables and chains that a packet passes through, and the order in which they are checked. Packets coming in from the network enter the diagram at the top, and are processed by both the PREROUTING chains. At this point, a decision is made—packets destined for the local system go to the left, while those being forwarded to some other destination take the right hand branch. Those that go left are processed by the incoming packets chain before being delivered to local processes, such as servers. Forwarded data is processed by the Forwarded packets and After routing chains before being sent on to its destination.

The firewall can also affect packets sent out by processes on the local system. These are checked against the three Output chains and the After routing chain before being transmitted via the appropriate network interface to their destinations. This means that an IPtables firewall can be used to limit the addresses that local processes can connect to, and the protocols that they can use.

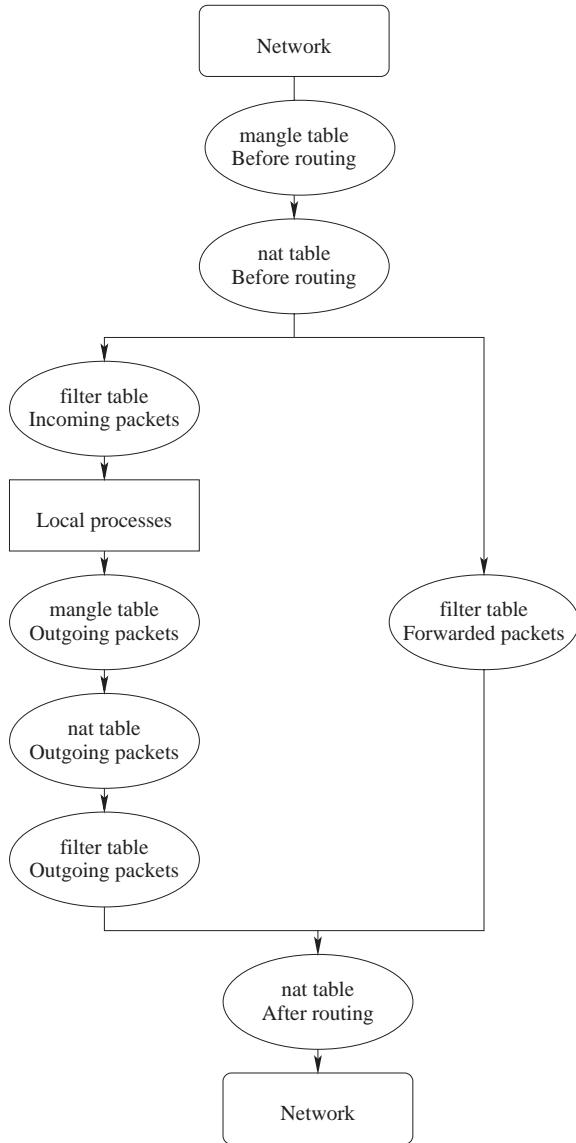


Figure 19.1 An overview of IPtables.

19.2 The Linux Firewall Module

This module can be used to set up a firewall on a Linux system with IPtables enabled, or to edit any part of an existing firewall. It stores the firewall configuration in a save file created and read by the `iptables-save` and `iptables-restore` commands, not in a shell script containing calls to the `iptables` command. Red Hat, Debian, and Gentoo Linux all use a save file like this as standard, which Webmin knows about and will work with.

If you have manually created a firewall using a shell script and want to use this module to edit it from now on, it will have to be converted to an IPtables save file so that Webmin can edit it. Fortunately, the module can do this for you automatically—all you have to do is stop your custom script from being run at boot time, and tell the module to create its own firewall setup script instead.

This also applies to firewalls created by tools such as YaST or fBuilder, which write out shell scripts of `iptables` commands. Unless the tool can also edit an IPtables save file (such as knetfilter), it should not be used alongside Webmin's Linux Firewall module, or they will probably overwrite each other's settings.

When you enter the module from the Networking category, the main page will usually display a list of all chains and rules in the first table that contains any (usually **Packet filtering**), as shown in Figure 19.2. However, if Webmin detects that the `iptables` or `iptables-save` commands are not installed, an error message will be displayed instead—check your distribution CD or website for a package containing them.

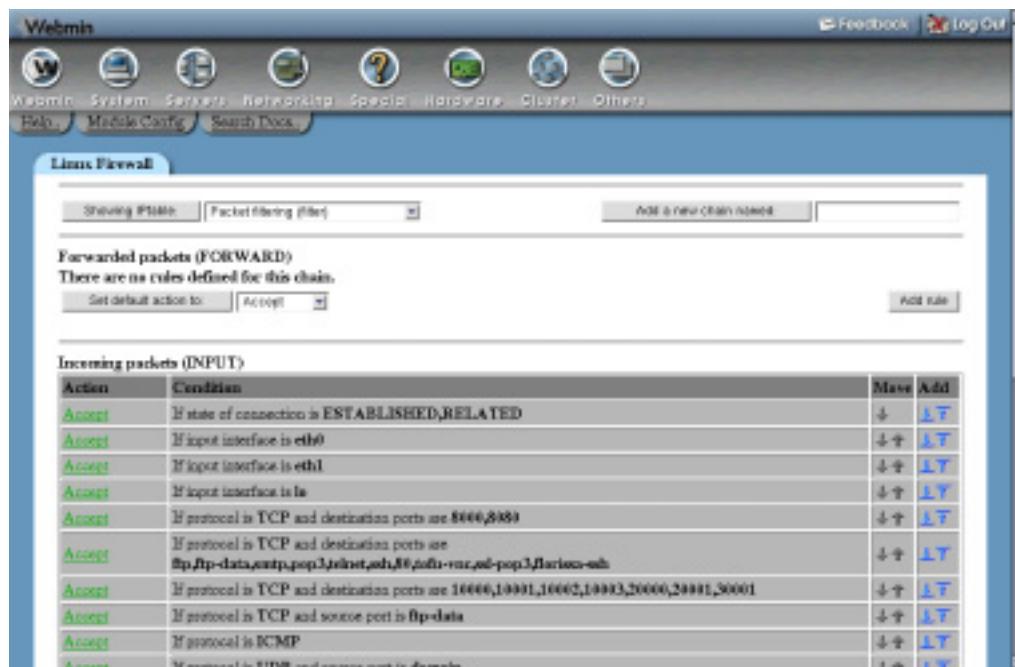


Figure 19.2 The Linux Firewall module.

If this is the first time you have used the module and no firewall has been set up on your system yet, the main page will instead display a form to simplify the initial firewall creation. Three options will be displayed—select one and click the **Setup Firewall** button to set it up. If necessary, Webmin will also display an **Enable firewall at boot time?** checkbox, which if selected will cause a boot-up script to be created so that the firewall is enabled at boot time as well.

The firewall setup options are:

Allow all traffic If this is selected, the firewall will be created “empty” and all traffic allowed through.

Do network address translation on external interface The firewall will be set up for NAT, so that hosts on an internal LAN can access the Internet via a host with a single public IP address. You must select the network interface that is connected to the Internet from the list next to this option, such as *ppp0*.

Block all incoming connections on external interface If this is chosen, the firewall will be set up to block all traffic coming into your system on the selected network interface, except for established connections, DNS replies, and harmless ICMP packets. The interface you select should be the one connected to the Internet, such as *ppp0*.

Block all except SSH and IDENT on external interface Similar to the previous option, but SSH and IDENT protocol traffic will be allowed as well.

Block all except SSH, IDENT, ping, and high ports on interface Similar to the previous option, but ICMP pings and connections to ports above 1024 will be allowed as well.

If this is the first time the module has been used and Webmin detects that a firewall already exists on your system, its rules will be displayed and you will be prompted to convert it to a save file so that the module can be used to edit it. If you choose to do this by clicking the **Save Firewall Rules** button, all existing tables, chains, and rules will be safely recorded. An **Enable firewall at boot time?** checkbox will also be displayed if necessary, which if selected will cause Webmin to create a boot script to activate the saved firewall rules at boot time.

If you choose to convert an existing manually created firewall configuration, be sure to disable any existing script that sets it up at boot time. Otherwise both the old script and the one created by Webmin will be run, possibly causing the rules set up in this module to be cancelled out by the older manual configuration.

19.3 Allowing and Denying Network Traffic

To restrict the types of connections and packets that your firewall will accept or forward, you need to create additional firewall rules. The best place for these rules is the **Packet filtering** table, in either the **Incoming packets** or **Forwarded packets** chain. If your firewall is acting as a router and you want to protect systems on the secure network that it is attached to but not the firewall itself, the **Forwarded packets** chain should be used. However, if you want to protect both the firewall and other systems that it routes to, rules should be added to the **Incoming packets** chain.

It is also possible to restrict data being sent out by your system, which may come from local processes or be forwarded from other hosts. To do this, you can add rules to the **Outgoing packets** chain. This can be useful for limiting what addresses and ports local users can connect to, if you desire.

To create a new rule to block traffic, the steps to follow are:

1. On the main page of the module, select **Packet filtering** from the list next to the **Show IPtable** button, and then click it to switch to the filtering table.
2. To add a rule that applies to all incoming traffic, click the **Add Rule** button in the **Incoming packets** section. If you want to restrict only forwarded traffic, click the button under **Forwarded packets** instead.
Either way, you will be taken to the rule creation form, shown in Figure 19.3.
3. Change the **Action to take** to **Drop**, so that packets matching this rule are silently discarded by the firewall.
4. In the **Rule comment** field enter a short explanation for this rule if you wish.
5. In the **Condition details** section, select the conditions that determine which packets will be matched and thus dropped. Only packets matching all conditions that are not set to <Ignore> will be dropped.

Some examples of the conditions to select to block certain kinds of traffic are:

Blocking all connections to a certain TCP port Set the **Network protocol** field to **Equals** and select **TCP**. To block a port, a protocol must always be selected.

Set the **Destination TCP or UDP port** to **Equals** and enter a port number into the **Port(s)** field next to it. You can block several ports by entering a list of numbers separated by commas into the **Port(s)** field, or block an entire range by selecting **Port range** and entering the start and end ports into the fields next to it.

Blocking all traffic from a particular address Set the **Source address or network** to **Equals** and enter the IP address to block into the field next to it. You can also block an entire network by entering a network/prefix pair like *130.194.164.0/24* into the field.

Set the **Connection state** to **Does not equal** and select **Existing connection** from the menu next to it. This step will allow your system to connect to the blocked addresses, but not vice-versa.

Blocking traffic to a particular address Set the **Destination address or network** to **Equals** and enter the IP address or network to block into the field next to it.

Because this will effectively stop the blocked system from connecting to yours as well, it may be a good idea to set the **Connection state** to **Does not equal** and select **Existing connection** from the menu next to it.

In all cases, it is usually a good idea to set the **Incoming interface** to the network interface that is connected to the Internet (such as `ppp0`), so that the restriction does not apply to connections from your local LAN.

6. When you are done selecting conditions, click the **Create** button. As long as there are no errors in your input, you will be returned to the module's main page on which the new rule will be listed.
7. To make the new rule active, click the **Apply Configuration** button at the bottom of the page.

The rules in each chain are evaluated in order from top to bottom, and the action taken is determined by whichever one matches first. If none match, then the chain's default action is taken,



Figure 19.3 The rule creation form.

which is usually to accept the packet. You can make use of this evaluation order to create a rule that allows a single IP address, followed by a rule to deny an entire network. The final effect will be that every host within the network is denied except one.

Because the ordering of rules is important, you may sometimes want to add a rule in the middle of an existing chain. To do this, use one of the arrow buttons under a chain's **Add** column on the module's main page to create a new rule either before or after an existing one.

The most common actions and their meanings are listed below. Not all are available in all chains and tables.

Do nothing If a rule with this action is matched, nothing will be done and processing will continue to the next rule.

Accept Matching packets will be immediately accepted, and no further processing will be done in the chain. However, rules in other tables may still affect the packet.

Drop Matching packets will be silently discarded, as though they were never received at all. No further processing will take place in this chain or any other.

Userspace Packets will be passed to a normal userspace process. This action is rarely used.

Exit chain Jump immediately to the end of the chain, and execute its default action instead. If this is used in a user-defined chain, processing will return to the rule that called it.

Masquerade Matching packets will have their source address changed to appear to come from the firewall system, and no further rules in the chain will be processed. When this action is selected, you can use the **Source ports for masquerading** field to control which ports the firewall will use for masqueraded connections. See Section 19.7 “Setting Up Network Address Translation” for more details.

The **Masquerade** option is available only in the **Network address translation** table, in the **Packets after routing** chain.

Source NAT Similar to the **Masquerade** option, but better suited to systems that have a fixed Internet IP address. If selected, you can use the **IPs and ports for SNAT** field to control which addresses and ports are used for NAT, as explained in Section 19.7 “Setting Up Network Address Translation”.

This option is only available in the **Network address translation** table, in the **Packets after routing** chain.

Destination NAT Matching packets will have their destination address and port modified based on the IPs and ports for **DNAT** field. This is the basis for transparent proxying, so to learn more, see Section 19.8 “Setting Up a Transparent Proxy”.

This action is available only in the **Network address translation** table, in the **Packets before routing** and **Output** chains.

Redirect This action redirects all matching packets to a port or ports on the firewall box, specified by the **Target ports for redirect** field. It can also be used for transparent proxying, although **Destination NAT** is more flexible.

The redirect action is available only in the **Network address translation** table, in the **Packets before routing** and **Output** chains.

You can also choose the **Run chain** option for the **Action to take**, which will pass the packet on to the user-defined chain or custom target entered into the field next to it. See Section 19.6 “Creating Your Own Chain” for more information on user-defined chains. Some of the targets available are LOG (for logging packets to `syslog`), MIRROR (for reflecting packets back to their sender), and MARK (for marking a packet for later conditions).

For each condition, the options **<Ignored>**, **Equals**, and **Does not equal** can be selected. The first means that the condition is not used at all when checking if a packet matches the rule. The second means that a packet must match the condition for it to match the entire rule, and the third means that the packet must NOT match the condition for the rule to be executed. If for example the **Incoming interface condition** was set to **Does not equal** and `eth0` selected, the rule would match only packets coming in on any interface except the primary Ethernet card.

Because almost all network protocols involve traffic flowing in two directions, attempting to block just incoming traffic from some address using the **Source address or network** condition will also block connections to the address as well, because reply packets that are part of the connection will be dropped. The same goes for blocking incoming data on a particular port using the **Destination TCP or UDP port** condition—if in the unlikely case that the randomly chosen source port of a connection from your system matches the blocked port, any replies to it will be dropped. For these reasons, it is usually a good idea when creating deny rules to set the **Connection state** condition to **Does not equal** and select **Existing connection** from the menu next to it.

This will cause IPtables to keep track of outgoing connections made by your server, and not block them.

As you can see, there are many different conditions available which can be combined to create quite complex rules. To learn more about what each of the available conditions do, see Section 19.10 “Firewall Rule Conditions”. Because there are so many conditions, Webmin allows you to create new rules that are almost identical to existing ones. To do this, click on an existing rule to edit it and use the **Clone rule** button at the bottom of the page to go to the rule creation form, with all conditions and actions set based on the original rule.

19.4 Changing a Chain's Default Action

Packets that do not match any rule in a chain will be processed using the default action, which is usually to accept the packet. On the module's main page, the default action for each chain is shown next to the **Set default action to** button. To change it, the steps to follow are:

1. Select the new action from the menu next to the **Set default action to** button. Only the **Accept**, **Drop**, **Userspace**, and **Exit chain** actions are available—see Section 19.3 “Allowing and Denying Network Traffic” for their meanings. Typically, only **Allow** and **Drop** make sense as default actions.
2. Click the **Set default action to** button to save the new default.
3. If changing to **Drop**, add any additional firewall rules needed so that your system can still access other servers and supply important services.
4. When done, click the **Apply Configuration** button to make the new default active.

Just changing the default action to **Drop** for incoming packets is an easy way to totally cut your system off from the network, and possibly make it unusable. Before you do so, make sure you allow at least the following kinds of traffic:

- **All established connections** Create an **Allow** rule with the **Connection state** set to **Equals** and **Existing connection** chosen.
- **Connections related to those that are established, such as FTP data connections** Create an **Allow** rule with the **Connection state** set to **Equals** and **Related connection** chosen.
- **All traffic on the loopback interface** Create an **Allow** rule with **Incoming interface** set to **Equals** and **lo** chosen.
- **Traffic from your system to itself on its primary network interfaces** For each interface create an **Allow** rule with both the **Source address or network** and **Destination address or network** set to the interface IP address.
- **Safe ICMP types** Create four **Allow** rules with the **ICMP packet type** set to **Equals** and **echo-reply**, **destination-unreachable**, **source-quench**, and **time-exceeded** chosen.

Changing the default action for forwarded packets to **Drop** will not cause as many problems—it will just be the equivalent of turning off forwarding altogether. Changing the default action for outgoing packets to **Drop** is a bad idea as it will cut off all network access, and probably makes very little sense in most cases.

19.5 Editing Firewall Rules

Webmin can be used to edit any of the existing firewall rules that have been created manually, in another program or using this module. Even though the module does not support all of the available IPtables condition and action options, you can still use it to safely edit rules containing unknown options. Only those known to Webmin can be changed, and others will be left untouched.

To edit a rule, the steps to follow are:

1. On the main page of the module, select the table the rule is in from the list next to be **Showing IPtable** button before clicking it.
2. Click on the action of the rule you wish to change in the table for its chain. This will take you to an editing form, which is identical to the creation form shown in Figure 19.3.
3. Change the action or any of the conditions, and click the **Save** button to return to the list of chains and rules. Or to delete the rule altogether, click the **Delete** button.
4. To make the changes active, click on **Apply Configuration**.

Rules can be moved up and down within their chain using the arrows under the **Move** column on the main page. Because rules are evaluated in order by the firewall, changing their ordering can affect which traffic is allowed or denied. Whenever you create a new rule, it will be added to the end of its chain, so it may be necessary to move it up to the correct position to get the desired effect.

19.6 Creating Your Own Chain

It is possible to create your own custom chains of rules in addition to the standard ones. The difference is, they will only be executed if a rule in one of the standard chains has its action set explicitly to jump to a custom chain. When execution of a custom chain finishes (or a rule with the **Exit chain** action is matched), evaluation will return to the calling chain. This means that custom chains can be used to define rules that are shared by several standard chains, instead of repeating the same rules in multiple places. In a way, a custom chain is like a subroutine in a programming language.

To create your own chain, the steps to follow are:

1. On the main page of the module, select the table you want the chain to be in from the menu next to **Showing IPtable**, and click the button. Custom chains can only be called from other chains in the same table.
2. Enter the name of your new chain into the text box next to the **Add a new chain named** button, and then click the button to create it. Chain names must be unique, and are generally composed of only lower-case letters and numbers.
3. Once the new chain has been created, it will appear at the bottom of the page. You can use its **Add rule** button to append rules to it, just as with one of the normal chains.

Custom chains do not have a default policy, so they have no **Set default action to** button on the main page. If execution of the chain reaches the end, control will always return to the caller. Custom chains can be deleted though, using the **Delete chain** button underneath their tables of rules.

A custom chain can contain rules that jump to other custom chains. However, a chain cannot jump to itself, nor can you create loops by jumping to another chain that jumps back to the first. Even if this were possible, it would be a very bad idea!

19.7 Setting Up Network Address Translation

If you have several systems in your home or office connected by a LAN and only one Internet IP address, network address translation can be used to give all those systems almost complete Internet access. NAT hides the addresses of all systems on the internal LAN behind a single Internet address, converting addresses and ports back and forth as needed. This allows all internal systems to make connections to any host on the Internet, such as web servers, DNS servers, POP3 servers, and so on. The only limitation is that internal systems cannot receive connections from other Internet hosts, which can cause some protocols (such as Internet telephony and network games) to fail.

Because of this limitation, internal systems are protected from most attacks from other hosts on the Internet, just as if you were to block all forwarded packets coming in on the external interface. NAT also makes IP address assignment easier, as there is no need to worry about running out of real Internet addresses to assign to internal hosts that do not really need them. For these reasons, it may make sense to set up NAT in your organization even if it is not totally necessary from a networking point of view.

NAT works by modifying the source address and port of packets sent by internal hosts and routed through the firewall. The source address is always changed to the external IP address of the firewall system, and the source port to a randomly chosen unused port. When a reply packet comes back, its destination port is used to determine the original internal client IP address and port to which the packet should be forwarded.

To set up NAT, all you really need is a system with two network interfaces—one for the internal LAN, and one that is connected to the Internet via dialup, ISDN, ADSL, or cable modem. Once you have this, the steps to follow are:

1. On the internal LAN, every system's Ethernet interface should be assigned an address on a private IP network such as 192.168.0.0, including the gateway system.
2. Set the default router on all internal systems to the LAN IP address of the gateway system.
3. Make sure that the gateway has IP forwarding enabled in the Network Configuration module under **Routing and Gateways**. See Chapter 16 for more information on how to do this.
4. On the main page of the Linux Firewall module on the gateway system, select **Network address translation** from the list next to the **Showing IPtable** button. Then click the button to display chains in the NAT table.
5. Click the **Add rule** button in the **Packets after routing** section, which will take you to the rule creation form.
6. Set the **Action to take** to **Masquerade**.
7. To control which ports the firewall will use for masqueraded connections, set the **Source ports for masquerading** option to **Port range** and enter starting and ending port numbers into the fields next to it. Usually just selecting **Any** to let the firewall use any available port will work fine.

8. Change the **Outgoing interface** condition to **Equals** and select the external network interface from the list next to it, such as *ppp0*.
9. Click the **Save** button at the bottom of the page to return to the list of chains and rules.
10. Click on **Apply Configuration** to make the new rule (and NAT) active.

It is possible to combine NAT with other firewall rules in the **Packet filtering** table to block connections to the firewall host itself. You can also prepend deny rules to the **Packets after routing** chain to stop certain internal hosts from accessing the Internet, or limit the ports to which they can connect.

The instructions above will work on any network that has a gateway system with a single Internet IP address. However, if your gateway's address is static it is better to select **Source NAT** in Step 6 instead of **Masquerade**. When using masquerading, any connections being forwarded by the firewall will be lost if the external network interface goes down, even if it comes back up again with the same IP address. If the external interface has a dynamically assigned address, this doesn't matter as the connections would be lost anyway. But when using a static IP address, it is possible for a connection to be maintained even through a short network outage.

To use it, in Step 6 set the **Action to take** to **Source NAT**. Then set the **IPs and ports for SNAT** to **IP range** and enter your system's static external IP address into the field next to it. All other steps in the NAT setup process are the same.

19.8 Setting Up a Transparent Proxy

Many networks use proxy servers like Squid to cache commonly accessed websites and thus cut down on the amount of bandwidth used by web browsing clients. Normally, each client must be configured to use the proxy server instead of making direct connections to websites. On a large network with many client systems or at an ISP where they are owned by many different people, this individual configuration can be difficult. It is made worse by each browser having its own proxy server settings, so if a user installs a new browser it will probably default to not using a proxy at all.

Fortunately, there is a solution—transparent proxying. If all client systems access the Internet through a gateway running an IPtables firewall, it can be configured to redirect connections to port 80 (used by most websites) to a proxy server on some other system. This means that clients do not need to be configured to access a proxy, as any HTTP requests that they make will be transparently sent to the proxy server without their knowledge.

To set up transparent proxying, the steps to follow are:

1. On the main page of the Linux Firewall module on the gateway system, select **Network address translation** from the list next to the **Showing IPtable** button, then click the button.
2. In the **Packets before routing** section, click on **Add rule** to go to the rule creation form. The rule being added will redirect all traffic on port 80 forwarded by the firewall system to a proxy server.
3. Set the **Action to take** to **Destination NAT**.
4. In the **IPs and ports for DNAT** field, select **IP range** and enter the address of the proxy server system into the field next to it. If the proxy is running on the same system, enter its Ethernet IP address (not 127.0.0.1).

In the field next to **Port range**, enter the port the proxy server is running on, such as *8080*.

5. Set the **Incoming interface** to **Equals** and select the internal LAN interface, such as *eth0*.
6. Set the **Network protocol** to **Equals** and select TCP.
7. If the proxy is on another system that is also on the internal LAN, make sure that its connections on port 80 will not be proxied by the firewall as well! To do this, set the **Source address or network** condition to **Does not equal** and enter the IP address of the proxy server into the field next to it.
If the proxy is on a different LAN or is the firewall system, this is not necessary.
8. Set the **Destination TCP or UDP** port to **Equals** and enter 80 into the **Port(s)** field.
9. Click the **Create** button to save the rule and return to the module's main page.
10. Click on **Add rule** under **Packets after routing** to bring up the rule creation form again. This rule will forward packets back in the other direction from the proxy to the client. If your firewall system is also running the proxy server, this rule is not necessary and you can skip to Step 16.
11. For the **Action to take**, select **Source NAT**.
12. In the **IPs and ports for SNAT** field, select **IP range** and enter the LAN IP address of the firewall server into the field next to it.
13. Set the **Destination address or network** to **Equals** and enter the IP address of the proxy server into the field next to it.
14. Set the **Network protocol** to **Equals** and select TCP.
15. Click the **Create** button to add the new rule.
16. Back on the main page, click the **Apply Configuration** button. All packets on port 80 forwarded by your firewall will now be sent to the proxy server instead.
17. Assuming you are running the Squid proxy server (version 2.4 or above) on the proxy system, you can use Webmin to configure it. Otherwise, you will need to set it up manually to accept transparent proxy connections, and there is no point reading beyond this step.
18. On the proxy system, enter the Squid Proxy Server module and click on **Miscellaneous Options**.
19. Set the **HTTP Accel Host** field to **Virtual**, and the **HTTP Accel Port** to **80**.
20. Set both the **HTTP Accel With Proxy** and **HTTP Accel Uses Host Header** fields to Yes.
21. Finally, click **Save** to return to the main page of the Squid module, and click the **Apply Changes** link near the top of the page to activate the new configuration.

From now on, any HTTP requests on port 80 forwarded by your firewall will be sent to the proxy server for processing. Transparent proxying can be safely used at the same time as conventional NAT by creating a **masquerade** rule in the **Packets after routing** chain, as explained in Section 19.7 “Setting Up Network Address Translation”.

19.9 Setting Up Port Forwarding

On a network that uses NAT to hide internal systems from the Internet, outside hosts cannot connect directly to those on the internal network. This is great for security, but can be annoying if there is some internal service that you do want to make available to the outside world. For example, your mail server system may not be the firewall host, which would normally make it inaccessible from the Internet. Fortunately, there is a solution to this problem—port forwarding.

This lets you redirect all connections to some port on the firewall system to a different host and port on your internal network. For a mail server, all data received on port 25 might be sent to the same port on the host that is actually being used for user email. Of course, this would make it impossible for your firewall system to receive email itself.

To set up port forwarding, follow these steps:

1. On the main page of the Linux Firewall module on the gateway system, select **Network address translation** from the list next to the **Showing IPtable** button, then click the button.
2. In the **Packets before routing** section, click on **Add rule** to go to the rule creation form. The rule being added will redirect all external traffic received by the firewall to some internal address.
3. Set the **Action to take to Destination NAT**.
4. In the **IPs and ports for DNAT** field, select **IP range** and enter the address of the internal host into the adjacent text box, such as *192.168.1.10*. In the **Port range** box, enter the port number on the internal host to which data should be sent, such as 25 for SMTP, 110 for POP3 or 80 for HTTP.
5. Set the **Network protocol** to **Equals** and select **TCP**.
6. In the **Destination TCP or UDP port** field, select **Equals** from the menu and enter the external port number for which forwarding should be done into the adjacent text field. Typically this will be the same as the port entered in Step 4.
7. Hit the **Save** button to create the rule and return to the main page, and then click the **Apply Configuration** button.

The only problem with this method is that connections from inside your network to the firewall system will not be forwarded to the other host.

19.10 Firewall Rule Conditions

When creating a firewall rule, you can select many different conditions to control which packets the rule matches. A rule's action will only be executed if all the conditions are matched. Each condition can be in one of three states, chosen by the menu next to it on the rule creation form:

<Ignore> The condition will be totally ignored when deciding whether the rule matches or not.

Equals The rule will only match if the packet matches the address, port, interface, or whatever was selected for this condition.

Does not equal The rule will only match if the packet does NOT match whatever was selected for this condition.

The available conditions and what each one matches are listed in Table 19.1. Note that some are not available in all tables and chains.

Remember that each condition is applied on a per-packet basis, and that a single TCP connection may involve multiple packets flowing in both directions.

Table 19.1 Firewall Conditions

Condition	Matches
Source address or network	The IP address, host, or network that the packet was sent from. When entering a network, you can use the network/prefix notation (like <code>192.168.0.0/16</code>) or the network/netmask notation (like <code>192.168.0.0/255.255.0.0</code>).
Destination address or network	The IP address, host, or network that the packet is going to. As with the source address, you can use both the network/prefix and network/netmask notations.
Incoming interface	The network interface on which the packet entered the firewall server. See the discussion of interface types and names in Chapter 16 for more details.
Outgoing interface	The network interface on which the packet is being sent out by the firewall server.
Fragmentation	When an IP packet is too large for the physical network it is being sent over, it will be broken into multiple fragments. If Is fragmented is chosen, the rule will apply only to fragments after the first one. If Is not fragmented is chosen, the rule applies only to the first fragment, or packets that were not fragmented at all. Because fragments after the first do not contain any protocol or port information, rules that have protocol, port, TCP, state, or type of service conditions will never match a fragment.
Network protocol	The network protocol of the data carried by the packet. TCP is used by HTTP, FTP, telnet, SSH, SMTP, POP3, and many other higher level protocols. UDP is used by the DNS, NFS, and NIS protocols. ICMP is used by commands like <code>ping</code> and <code>traceroute</code> .
Source TCP or UDP port	The port that a TCP connection or UDP packet came from. For packets sent by a client to a server, the source port is usually randomly assigned and thus useless for firewalling. But for packets sent back from the server to the client, the source port is the same as the port that the client connected to. If the Port(s) option is selected, you can enter one or more ports into the field next to it, separated by commas. If Port range is selected, you must enter a starting and ending number to cover all ports between them. This condition can be used only if your Network protocol is set to TCP or UDP.
Destination TCP or UDP port	The port that a TCP connection or UDP packet is going to. Instead of entering a port number, you can enter a name from the <code>/etc/services</code> file that is associated with a port, such as <code>telnet</code> or <code>http</code> . As with the Source TCP or UDP port condition, a list or range of ports can be entered, and the Network protocol must be set to either TCP or UDP.

Table 19.1 Firewall Conditions (Continued)

Condition	Matches
Source and destination port(s)	For a condition of this type to match, both the source and destination ports must be in the comma-separated list of port names or numbers entered into the field next to it. This condition has never seemed particularly useful to me.
TCP option number is set	Matches if the entered TCP option number is set.
TCP flags set	The flags set on a TCP packet. The selections in the second row determine which flags the firewall will look at, while those in the first row indicate whether a particular flag must be set or not. This condition can be used to detect TCP packets that are part of an existing connection. However, the Connection state condition is a far superior and simpler way of doing the same thing. For this condition to be used, the Network protocol must be set to TCP.
ICMP packet type	For ICMP packets, this condition matches if the packet type matches whatever is chosen from the menu next to it. Some types such as echo-request and echo-reply are sent by the ping command, while others are used for low-level network flow control. Because ICMP packets are usually harmless and sometimes important, it is not necessary to block them. As would be expected, the Network protocol must be set to ICMP for this condition to be used.
Ethernet address	The MAC address (usually Ethernet) of the packet sender. If the packet was forwarded by another router after being sent by the original host, its MAC address will be that of the router. Ethernet addresses must be formatted like <i>00:D0:B7:ID:FB:AA</i> , as displayed by the ifconfig command.
Packet flow rate	Matches packets up to the rate entered (if Below is chosen), or above the rate entered (if Above is chosen). This condition cannot be used for limiting the amount of traffic a host can send—rather, it is useful for logging with the LOG target only a fraction of the packets that match some rule.
Packet burst rate	The maximum initial number of packets to match. This number gets recharged by one every time the Packet flow rate is not reached, up to the number entered.

Table 19.1 Firewall Conditions (Continued)

Condition	Matches
Connection state	Matches packets depending on their connection status and the options chosen from the menu. You can select more than one to match packets with any of the chosen statuses. The available options and the packets they match are: New connection Matches packets that are part of a new TCP connection. Existing connection Packets that are part of a connection that has already been established. Related connection Packets in a connection that is related to one already established, such as an FTP data connection. Not part of any connection Packets that do not fit in with any new or existing connection at all.
Type of service	Matches packets whose IP type-of-service field is the same as the type selected from the menu next to this condition.
Sending unix user	Packets sent by a local process owned by the chosen UNIX user. This condition (and the three below) make sense only in the Outgoing packets chain.
Sending unix group	Packets sent by a local process owned by the chosen UNIX group.
Sending process ID	Packets sent by a local process with the specified PID.
Sending process group	Packets sent by a local process with the specified process group ID.
Additional parameters	This field can be used to enter additional parameters to a rule that cannot be set through the module's user interface, such as— <i>log-level warn</i> . It should only be used if you are familiar with the <code>iptables</code> command.

19.11 Configuring the Linux Firewall Module

This module has several configurable settings, reachable by clicking on the **Module Config** link on the main page. It is shown in Table 19.2.

19.12 Summary

Any system that is directly connected to the Internet is potentially vulnerable to attacks by hackers and other malicious people. After reading this chapter, you should know how to limit the kinds of traffic that your system will accept, making it much harder for attackers to break in. You should also know how to set it up as a masquerading gateway that protects hosts on an internal LAN which still allows them access to the Internet. Finally, you should know how to set up a transparent proxy and configure port forwarding, if required, on your network.

Table 19.2 Module Configuration Options

Directly edit firewall rules instead of save file?	Normally, this field is set to No , which tells the module to edit firewall rules in a save file that can be applied by hitting the Apply Configuration button. Selecting Yes switches the module to a different mode, in which all changes are made directly to the active firewall rules. The user interface in this mode is similar, but the apply, revert, and boot-time buttons on the main page are no longer displayed, as they do not make any sense. Directly updating the firewall rules makes sense if some other program on your system is editing them as well, such as PortSentry. However, all rules will be lost when your system is re-booted, unless you have manually created a script to save them at shutdown time with the <code>iptables-save</code> command, and restore them at boot time with <code>iptables-restore</code> .
IPtables save file to edit	This field can be used to specify an alternate file for the module to read and update IPtables rules in. You should only change it if your system is using some custom save file, perhaps created by another firewall tool.
Display comment?	This field determines if the comment for each rule is shown on the module's main page along with the condition and action.
Display condition?	This field determines if each rule's condition is shown on the module's main page.

Setting the Date and Time

This chapter explains how to set the system and hardware clocks on your server.

20.1 The System Time Module

All UNIX systems have an internal clock to keep track of the current time, even when the system is powered off. Linux systems effectively have two clocks—one that is maintained by the kernel when the system is running, and a separate hardware clock that runs all the time. The kernel’s system time is set based on the hardware time when the kernel is loaded, so they should be synchronized. However, if one of the clocks is slower than the other it is possible for the hardware and system times to fall out of sync on a system that has been running for a long time.

All UNIX systems store the time internally as the number of seconds since January 1, 1970 GMT. For display, this is converted to a human-readable local time based on the system’s configured time zone. All Linux distributions allow you to choose your time zone at install time, and include a tool for changing it later. However, there is no support in Webmin for choosing a time zone.

The system and hardware times can be displayed and set using the `date` and `hwclock` commands respectively. Only the `root` user can change the system time, and only `root` can use the `hwclock` command to display the hardware time.

You can adjust both the system and hardware times using the System Time module, which can be found under the Hardware category. The module really has only one page, which is shown in Figure 20.1. Both times on the page are updated every five seconds, so that they remain correct even if the page has been displayed in your browser for a long period.



Figure 20.1 The System Time module.

20.2 Changing the System Time

The system time can be brought forwards or sent backwards at any time using this Webmin module. Generally, this is quite safe—however, large changes may confuse some programs that do not expect to see the current time go backwards or jump forwards by a huge amount.

To change the system time, the steps to follow are:

1. On the main page of the module, select a new **Day**, **Month**, **Year**, **Hour**, **Minute**, and **Second** in the **System Time** section.
2. Click the **Apply** button below the fields. The new date and time will be set, and the page will be redisplayed.

It is also possible to force the system time to be set to the current hardware time, by clicking the **Set system time to hardware time** button. Either way, any change will immediately be visible to all programs running on your system, such as desktop clocks, syslog and mail clients.

20.3 Change the Hardware Time

Because the hardware time is read by the kernel only when the system boots, it can be changed without having any effect on programs that are currently running. To change it, follow these steps:

1. On the main page of the module, select a new **Day**, **Month**, **Year**, **Hour**, **Minute**, and **Second** in the **Hardware Time** section.

2. Click the **Apply** button below the fields. The new date and time will be set, and the page will be redisplayed.

You can also synchronize the hardware time with the system time by clicking the **Set hardware time to system time** button. It is a good idea to do this every now and then on a system that hasn't been rebooted for a long time, so that the times do not drift too far out of sync.

20.4 Synchronizing Times with Another Server

The System Time module can also be used to set the system or hardware time based on the system time of another server. The other server must be either running an NTP (Network Time Protocol) server, or have the `time` protocol enabled in `inetd` (as explained in Chapter 15). For your system to use NTP for synchronization you must have the `ntpdate` NTP client program installed.

To synchronize the time, the steps to follow are:

1. Enter the hostname or IP address of the other server into the **Host/Address** field in the **Time Server** section. It is always better to choose a server that is close by, so that the effect of network latency is minimized.
2. Click either the **Sync system time** or **Sync system time/Sync hardware time** button, to set just the system time or both the system and hardware times, respectively. If the server cannot be contacted or does not support the NTP or time protocols, an error message will be displayed. Otherwise the time or times will be set and the page redisplayed.

20.5 Module Access Control

As with many other modules, it is possible to restrict what a Webmin user or group can do in the System Time module. However, the available restrictions are very basic due to module's limited functionality, and do not really make it any more secure for use by untrusted users.

Once a Webmin user has been granted access to the module as described in Chapter 52, you can limit what he can do by following these steps:

1. In the Webmin Users module, click on **System Time** next to the name of the user or group that you want to restrict.
2. Change the **Can edit module configuration?** field to **No**, so that the user cannot change operating-specific settings.
3. To stop the user changing the system time, set the **User can edit system time** field to **No**.
4. To prevent the user from changing the hardware time, set the **User can edit hardware time** field to **No**.
5. When done, click the **Save** button at the bottom of the page to make the new restrictions active.

20.6 Other Operating Systems

Linux is the only operating system supported by the System Time module that has separate hardware and system times. Solaris, Irix, HP/UX, and OpenServer have only a single system time, which can be set in exactly the same ways as on Linux. FreeBSD, NetBSD, and Mac OS X like-

wise support only system time, which can also be set in the same ways, but only to the nearest minute. Other operating systems cannot use this module at all.

20.7 Summary

You should now know how to manually adjust the two clocks on your Linux or UNIX system. You should also now have a firm grasp of the process for synchronizing your system's clock with another server.

Boot Loader Configuration

This chapter covers the Linux boot process and the LILO and GRUB boot loaders. It explains how to run different operating systems or load different kernels at boot time.

21.1 Introduction to Boot Loaders

When a Linux system running on PC hardware is started, the first code to be run is the BIOS (Basic Input Output System) which is loaded from ROM. After it has finished testing the system's memory and discovering what hardware is installed, it attempts to pass control to an operating system boot loader to continue the boot process. The boot loader is a tiny program that may prompt the user to choose which OS to run, and then loads the rest of the operating system kernel from a hard drive, floppy disk, or some other source.

On a normal system, the boot loader is loaded by the BIOS from the first block on the primary hard drive, called the master boot record or MBR. The BIOS, however, may (depending on its configuration) check the floppy drive or CD-ROM for a boot loader first, so the system can be booted off a removable disk. This is usually only done when installing a new operating system—for normal everyday use, almost every system boots from hard disk.

There are several boot loaders available for Linux, but the two most common (and the two which will be covered in this chapter) are LILO and GRUB. Both work only on x86-compatible PC hardware, so if you are running Linux on an Apple, SPARC, or Alpha system, this chapter will not be much use to you. Each non-PC hardware platform has its own specialized Linux boot loader designed to deal with the particular quirks and requirements of the platform.

Other operating systems (such as Windows, FreeBSD, and Solaris) have their own boot loaders, which do basically the same thing as LILO or GRUB, but are designed to load the kernel of a different OS instead. Webmin does not support the configuration of any non-Linux boot loader, so if you are running a different version of UNIX, this chapter can be skipped.

On a Linux system, the boot loader's primary responsibility is the loading of the kernel. Once the kernel has been loaded into memory and control has been transferred to it, the boot loader's job is done. The kernel then mounts the root filesystem, initializes drivers, and finally runs the `init` program to continue the boot process as explained in Chapter 9.

The boot loader can also start the process of loading a totally different OS on systems that have more than one installed. It does this by loading the other operating system's boot loader from the first block of a partition or other hard disk, and then transferring control to it. The other OS then loads exactly as it would if its boot loader were run directly by the BIOS. Being able to decide which operating system to load at boot time makes it practical to have two or more installed on the same system, such as Windows and Linux.

Both LILO and GRUB can be configured to display a menu of boot options when they are loaded, allowing the user to select which particular kernel to load or another operating system to load. Being able to choose from several different kernels can be particularly useful when you have installed a new one and want to have the option of booting into both the new and the old. It is even possible to have several boot options that all load the same kernel version but with different command-line options.

All Linux distributions will give you the option of automatically setting up a boot loader at installation time. You can usually choose to boot other installed operating systems as well—for example, if you are adding Linux to a system with Windows already installed on a different partition. If this default configuration is working for you, be very careful when changing the LILO or GRUB configuration manually or through Webmin. A single mistake may render your system unbootable, and necessitate the use of a rescue disk to recover.

LILO's configuration is stored in the file `/etc/lilo.conf`. The boot loader itself, however, does not actually read this file—instead, it reads from a separate map file that is built from `lilo.conf` whenever the `lilo` command is run. This map file contains the actual on-disk block locations of the kernel files, which allows LILO to load a kernel without having to understand the format of the filesystem on which it is mounted. Any time `lilo.conf` is changed or a kernel is recompiled or installed, the `lilo` command must be rerun to update the map file so the boot loader knows where to look on disk.

One major limitation of LILO is that on systems with older BIOSs, it can only boot kernels that lie within the first 8 GB of a hard disk. With drives over 100 GB in size becoming common, this can be a serious problem unless the disk is partitioned properly. Typically, the `/boot` directory in which kernels are stored is mounted from a separate partition that is located at the start of the disk, and the root directory is mounted from a partition that takes up the rest.

GRUB usually uses the configuration file `/boot/grub/menu.lst`, but unlike LILO it does understand the format of `ext2`, `ext3`, and `vfat` filesystems and can read the `menu.lst` and kernel files without the need for a block map. For this reason, and because GRUB can load a kernel stored anywhere on the hard disk, it is usually considered to be a superior boot loader and is slowly overtaking LILO on modern Linux distributions.

21.2 The Linux Bootup Configuration Module

This module allows you to configure LILO—the most common Linux boot loader. It can be found under the Hardware category, and when you enter it, the main page displays a table of

icons as shown in Figure 21.1. Each icon represents a boot-time menu option, which can be either a Linux kernel or another operating system.

If Webmin detects that you do not have LILO installed, the main page will display an error message to that effect. If this is the case, your distribution probably set up GRUB as its boot loader—see Section 21.6 “The GRUB Boot Loader Module” instead.

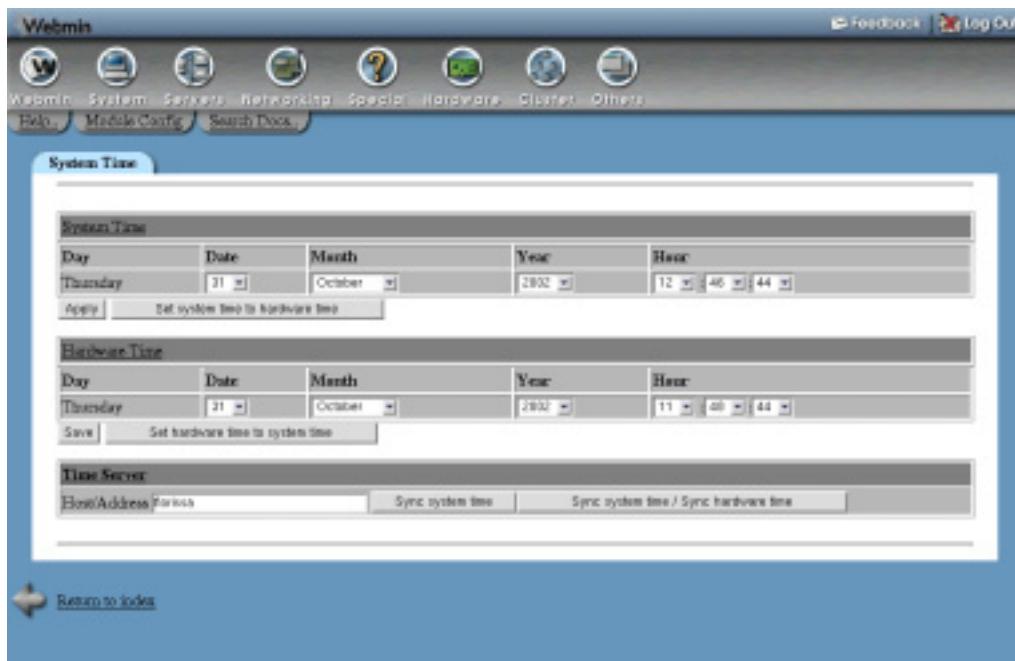


Figure 21.1 The Linux Bootup Configuration module.

Some Linux systems have both GRUB and LILO installed, even though only one can actually be used as a boot loader at any one time. If your system uses GRUB, you should probably not use this module even though it will work correctly. Any time the **Apply Configuration** button on the main page is clicked, LILO will be installed on the disk or partition configured on the global options, possibly overwriting GRUB.

21.3 Booting a New Kernel with LILO

If you have just compiled a new kernel and want to be able to use it, you will need to add a new LILO boot kernel entry. To do this, follow these steps:

1. After compiling the kernel, copy its compressed kernel image file (usually found under the source directory at `arch/i386/bzImage`) to the `/boot` directory. Normally it should be renamed to `vmlinuz-xx.yy.zz`, where `xx.yy.zz` is the kernel version number.
2. On the main page of the **Linux Bootup Configuration** module, click on the **Create a new boot kernel** link to go to the kernel creation form.

3. Enter a unique name for your new kernel into the **Name** field, such as *linux-xx.yy.zz*. Whatever you enter will appear in the LILO menu at boot time.
4. In the **Kernel to boot** field, enter the full path to the kernel file that you copied to the */boot* directory.
5. To pass extra options to the kernel, set the **Kernel options** field to **Add options** and enter them into the text field to its right. Most of the time, no additional options are needed though.
6. Set the **Boot device** field to **Device**, and choose the partition that contains your system's root filesystem from the menu next to it.
7. If the **root** directory on your system is mounted from a device that is not compiled into the kernel (such as a SCSI disk or hardware RAID controller), you will need to create an initial RAM disk containing the kernel modules needed to access the root filesystem. The simplest way of checking to see if this is necessary is to look at other existing boot kernel configurations. To create an initial RAM disk file under the */boot* directory for kernel version *xx.yy.zz*, you will need to run a command like

```
mkinitrd /boot/initrd-xx.yy.zz xx.yy.zz
```

and then set the **Initial ramdisk file** option to the path of the newly created file.

8. Click the **Create** button to create the new LILO boot kernel and return to the module's main page. An icon for the kernel should now be visible.
9. Click **Apply Configuration** at the bottom of the page to have LILO reinstalled on your hard disk with the new kernel in its map file. A page showing the output of the *lilo* command and any errors encountered will be displayed so you can see if the installation was successful or not.
10. To use the new kernel, you will need to reboot. Depending on the LILO configuration, it will either display a menu of options at boot time, or prompt you to enter an option name. Either way, select your new kernel to have it loaded and started. Be sure to watch the debugging output and error messages that the kernel displays while booting so that if anything goes wrong you can diagnose the problem. If there is a problem, you may need to reboot and select the old kernel option, then use Webmin to fix the LILO configuration.

An existing boot kernel can be edited by clicking its icon on the main page, which will take you to an editing form. Any of the fields can be edited and the changes saved by clicking the **Save** button, or the kernel can be removed by clicking **Delete**. Always be careful when editing any kernel configurations that you did not create yourself, as a mistake may make the system unbootable.

21.4 Booting Another Operating System with LILO

If your system has multiple operating systems installed on different partitions or hard disks, you can use LILO to select which one to load at boot time. To add a new operating system that you can select at boot time, follow these steps:

1. On the main page of the Linux Boot Loader module, click on the **Create a new boot partition** link to bring up the partition creation form.

2. Enter a unique name for your new boot option into the **Name** field, such as *windows98*. Whatever you enter will appear in the LILO menu at boot time.
3. Select the partition on which the operating system you want to boot from the **Partition to boot** menu. The selected partition must have an appropriate boot loader or boot sector installed. Windows, for example, does by default, but other operating systems like FreeBSD may need a boot loader to be installed separately.
4. Set the **Pass partition table to OS** field to **Yes**, and select the drive on which the operating system's partition is located.
5. Click the **Create** button, and if you have not made any errors on the form you will be returned to the module's main page.
6. Click **Apply Configuration** at the bottom of the page to have LILO reinstalled on your hard disk with the new boot option in its map file. A page showing the output of the `lilo` command and any errors encountered will be displayed so you can see if the installation was successful or not.
7. You should now be able to reboot and select the new OS from the LILO menu.

Once you have created a new operating system boot option, you can edit or delete it at any time by clicking on its icon on the module's main page. If you make any changes, remember to click **Apply Configuration so they can be used at boot time**.

21.5 Editing Global LILO Options

LILO has several configurable options that apply to all bootable kernels and operating systems. The steps to edit them are:

1. Click on **Global Options** on the module's main page to go to the global options configuration form.
2. To change the drive or partition on which LILO is installed, select it from the **Write boot loader to** menu. Generally you will not need to change this if LILO is already being correctly run at boot time.
3. Normally LILO will give the user a chance to select a kernel or operating system to load at boot time. To disable this, set the **Display LILO prompt?** option to **No**. To give the user the opportunity to select an OS, set it to **Yes**.
4. To change the kernel or OS that is loaded automatically at boot time if no other selection is made, adjust the **Default kernel/partition** field.
5. To adjust the amount of time that LILO will wait for user input before loading the default kernel, enter a new time into the **Time to wait at LILO prompt** field.
6. To prevent untrusted users from booting the system, enter a password into the **Default boot password** field. Unless a boot option has the **Password needed for** field set to **Booting kernels with extra options**, it will not be loadable unless the password is entered.
7. On systems with modern BIOSs, LILO can be configured to load a kernel located anywhere on the hard disk. To enable this, set the **Allow booting from beyond 1024 cylinders?** field to **Yes**.
8. Click the **Save** button to save your changes to the global options and return to the module's main page.

9. Click on **Apply Configuration** so LILO will use the new options at the next boot time. If the disk or partition on which LILO is to be installed was changed, it will be written to the new location now.

21.6 The GRUB Boot Loader Module

As the name suggests, this module allows you to set up GRUB. Like the LILO module, you enter it from the Hardware category and the main page shows a list of icons—one for each boot-time option. Figure 21.2 shows an example.



Figure 21.2 The GRUB Boot Loader module.

The module's icon will only appear if Webmin detects that GRUB is installed on your system. If it is not visible or if an error is displayed when you enter the module, GRUB is not installed. If so, LILO is probably being used instead and you should use the Linux Boot Loader module.

One peculiarity of GRUB is that internally it refers to all hard disks by their BIOS disk number. The first drive identified by the BIOS is `hd0`, and `hd0,0` is the first partition on that drive. On a system with only IDE hard drives, this numbering is quite simple—BIOS disk 0 is the primary master, or `/dev/hda` on Linux. Disk 2 (called `hd1` by GRUB) is the primary slave, and so on. On a system with SCSI and IDE drives, however, things get more complex. IDE disks usually come before SCSI in the BIOS ordering, but this may be reversed on some systems. Fortunately, the GRUB module in recent versions of Webmin can automatically detect the relationship between Linux device files and BIOS disk numbers.

21.7 Booting a New Linux Kernel or BSD with GRUB

If you have just compiled a new kernel and want to be able to use it, you will need to add a new GRUB boot option.

A similar process should be followed if you have both Linux and FreeBSD, NetBSD, or OpenBSD installed on your system and want to be able to choose one of them at boot time. To set this up, follow these steps:

1. To boot a Linux kernel after compiling, copy its compressed kernel image file (usually found under the source directory at `arch/i386/bzImage`) to the `/boot` directory. It should normally be renamed to `vmlinuz-xx.yy.zz`, where `xx.yy.zz` is the kernel version number.
2. On the main page of the **GRUB Boot Loader** module, click on the **Add a new boot option** link to go to the option creation form.
3. Enter a unique name for your new kernel into the **Option title** field, such as `linux-xx.yy.zz`. Whatever you enter will appear in the GRUB menu at boot time.
4. Set the **Boot image partition** field to **Selected** and choose the partition that contains your kernel from the list next to it. If the partition does not appear in the menu, you will need to choose **Other** instead and enter the disk and partition into the field next to it, in the `hdX,Y` format used by GRUB. For example, `hd2,1` would be the second partition on BIOS drive 3.
5. For **Operating system to boot**, select **Linux kernel** and enter the path to the kernel's compressed image file into the field next to it. To pass additional arguments to the kernel, enter them into the **Kernel options** field below it. For FreeBSD, you must also select **Linux kernel** and enter `/boot/loader` into the field. No additional kernel arguments are allowed. For NetBSD or OpenBSD, select **Linux kernel** as well and enter this:

```
-type=netbsd /netbsd-elf
```

6. If the root directory on your system is mounted from a device that is not compiled into the Linux kernel (such as a SCSI disk or hardware RAID controller), you will need to create an initial RAM disk containing the kernel modules needed to access the root file-system. The simplest way of determining if this is necessary is to look at other existing boot kernel configurations. To create an initial RAM disk file under the `/boot` directory for kernel version `xx.yy.zz`, you will need to run a command like

```
mkinitrd /boot/initrd-xx.yy.zz xx.yy.zz
```

and then set the **Initial ramdisk file** field to the path to the newly created file.

7. Finally, click the **Create** button. As long as there were no errors detected in your input, you will be returned to the module's main page, which will now contain an additional icon for the new kernel.
8. To boot into the new kernel, you will need to restart your system. When GRUB loads at boot time, it will display a menu of available boot options, from which you can select the newly added kernel. Be sure to watch the debugging output and error messages that the kernel displays while booting, so that if anything goes wrong you can diagnose the problem. If there is a problem, you may need to reboot and select the old kernel option, then use Webmin to fix the GRUB configuration.

Once you have created a new kernel boot option, you can edit it by clicking on its icon on the module's main page. On the editing form, any of the fields can be edited and the changes saved by clicking the **Save** button, or the kernel can be removed by clicking **Delete** instead. Always be careful editing any kernel configurations that you did not create yourself, as a mistake may make the system unbootable.

21.8 Booting Another Operating System with GRUB

If your system has another operating system installed on a different hard disk or partition, you can configure GRUB so it can be chosen and started at boot time instead of Linux. If you want to boot FreeBSD, NetBSD, or OpenBSD, see Section 21.7 “Booting a New Linux Kernel or BSD with GRUB” instead—but for Windows, UNIXWare, or any other OS, follow these steps:

1. On the module's main page, click on **Add a new boot option** to bring up the boot option creation form.
2. Enter a unique name into the **Option title** field, such as *windows*.
3. Set the **Boot image partition** field to **Selected** and choose the partition that contains the other OS from the list next to it. If the partition does not appear in the menu, you will need to choose **Other** instead and enter the disk and partition into the field next to it, in the *hdX, Y* format used by GRUB.
4. Change the **Operating system to boot** to **Other OS**.
5. Normally, GRUB will simply run the boot loader in the first sector of the chosen partition. There may not always be a boot loader there if, for example, the operating system normally writes its loader to the master boot record.
If the other operating system is Windows, select the **From chainloader file** and enter *+1* into the field next to it. You must also check the **Make root partition action?** option.
If booting SCO UNIXWare, you need to also select the **From chainloader file** and enter *-force +1* into the field next to it. The **Make root partition action?** option must also be selected.
6. Click the **Create** button to have the new OS added. Your browser will return to the module's main page, which will now include an icon for your new boot option.
7. To boot into the other operating system, restart your system and select it from the GRUB menu at boot time.

As with boot options for Linux kernels, you can edit or delete the option for another operating system by clicking on its icon on the module's main page. Any changes will take effect immediately, to be used when the system is next rebooted.

21.9 Editing Global GRUB Options

GRUB has several options that apply to all bootable kernels and operating systems. To edit these global options, follow these steps:

1. Click the **Edit Global Options** button on the module's main page, which will take you to the options form.

2. To control which kernel is booted automatically if the user does not choose one from the GRUB menu within the configured time limit, change the **Default boot option** field. If the option you choose cannot be loaded, GRUB will fall back to whatever is selected in the **Fallback boot option** field.
3. To change the amount of time that GRUB will wait for the user to choose a boot option before it uses the default instead, edit the **Timeout before using default** field.
4. The GRUB boot menu allows users to do things like change kernel parameters and read arbitrary files on Linux filesystems. To prevent this, enter a password into the **Boot password** field. This will limit users to the available boot options unless the password is entered. Furthermore, boot options in which the **Password locked?** field has been set will not be selectable either.
5. When done, click the **Save** button and you will be returned to the module's main page.

21.10 Installing GRUB

If you have been using the LILO boot loader and want to switch to GRUB, you will need to install it on the same master boot record or partition that LILO is currently using. This only has to be done once, unlike LILO which has to be effectively reinstalled every time its configuration is changed.

Follow these steps to install GRUB:

1. On the module's main page, click on the **Edit Global Options** button.
2. From the **Install GRUB on disk/partiton** menu, select the disk or partition onto which you want GRUB installed. This will typically be the first hard drive on your system.
3. Click the **Save** button to return to the module's main page.
4. Click on the **Install GRUB** button to have it written to the drive or partition chosen in Step 2.
5. Create any necessary kernel boot options as explained in the Section 21.7 “Booting a New Linux Kernel or BSD with GRUB” so your system can be booted into Linux from now on. If you reboot before doing this, it will be impossible to start Linux again!

21.11 Configuring the GRUB Boot Loader Module

This module has only one configuration setting that you might want to change. The rest are related to the location of GRUB and its configuration file on your system, and generally do not need to be modified. The setting, which you can edit by clicking on the **Module Config** link on the main page, is shown in Table 21.1.

21.12 Summary

This chapter has explained the purpose of boot loaders on a Linux system, and shown how to set up and configure the two most common loader—LILO and GRUB. After reading it you should know how to add an option to start a different operating system at boot time, or use a different Linux kernel. Because a mistake can render the system unbootable, you should also understand the risks involved in reconfiguring your boot loader.

Table 21.1 Module Configuration Options

File for device name mappings	When Get from GRUB is selected, the module will use the command <code>grub --device-map</code> to obtain a list of BIOS disk numbers and their associated Linux device files. This works perfectly, but can be very slow on some systems. The alternative is to have the module read a file containing the disk mappings, usually found at <code>/boot/grub/device.map</code> . This is faster, but if a new hard disk is added to your system, it may not get added to this file depending on how often your Linux distribution updates it. This means that a new disk may not show up in the menus in this module.
--------------------------------------	---

Printer Administration

This chapter shows you how to use Webmin to set up printers and printer drivers on your system. It covers the many different print systems in use, such as CUPS, LPRng, and the Solaris print server.

22.1 Introduction to Printing on Linux

Like other operating systems, Linux can print directly to attached printers or indirectly to printers connected to another system on a network. Any program that wishes to print runs a command like `lpr` to submit a job to the print server daemon, which adds the job to a queue for the specified printer. When the printer is ready, the daemon opens the appropriate parallel port or USB device file and sends it the print job data. If the printer is attached to another system on the network, the daemon connects using the appropriate protocol and sends it the job for queuing and printing.

Almost all Linux programs submit print jobs in one of two formats—plain text, or PostScript. Because most consumer-grade printers do not support PostScript, the print server daemon must convert the submitted PostScript to a format that the printer does recognize. This is done using a driver program or script, most of which are based around the freely available `ghostscript` PostScript rendering program.

Almost every different printer manufacturer (and even different models by the same manufacturer) has its own data format in which it accepts print jobs. All manufacturers supply driver software for Windows with their printers, but very few include drivers for Linux. This means that the job of writing drivers has to be done by free software enthusiasts who cannot always keep up with the rate at which new printers with new data formats are released. Some newer printer models may not be supported on Linux until a while after their release, and some models for which driver information is not available may never be supported.

Several different print system packages exist for Linux, such as LPR, LPRng, and CUPS. All perform basically the same task but have different capabilities and are configured in different ways. Most modern Linux distributions include either LPRng or CUPS, but some older versions may just include LPR.

There are also several different packages of printer drivers, many of which were created by Linux distribution vendors. All have the same purpose of converting postscript into the data format accepted by a printer, but have different configuration files and capabilities. The best are the CUPS drivers, because they have been designed for and well integrated with the CUPS print server.

22.2 The Printer Administration Module

No matter which kind of print server is installed on your system, it can be configured using Webmin's Printer Administration module. The module attempts to provide a similar user interface regardless of the print system and drivers being used, while still allowing you to use all of their capabilities. By default, the module assumes that you are using the driver and printer daemon packages that are installed as standard by your Linux distribution. If you have installed a different print server (such as the superior CUPS), then you will need to tell Webmin which print system you are using. See Section 22.6 "Configuring the Printer Administration Module" for details.

All of the instructions in this chapter are written with the CUPS print system and drivers in mind, and all of the screenshots are taken from a system using CUPS. This is because I believe it to be the best print system available for Linux, and because it is used by default on many modern Linux distributions.

When you enter the module from the Hardware category, the main page will list all printers installed on your system, as shown in Figure 22.1. On Red Hat Linux versions 7.0 and above, only printers that have been created using Webmin will be shown. Those added by other programs (such as Red Hat's `printconf` tool) will not be listed, as they cannot be edited using this module.

When the print server daemon is running, there will be a button labeled **Stop Scheduler** at the bottom of the main page. If clicked, the daemon will be stopped, causing all printing to cease. To start it again, click the **Start Scheduler** button that will appear in its place.

If Webmin detects that the currently configured print system is not installed, an error message will appear on the main page instead. This indicates that either print software has not yet been installed on your server, or the wrong system was chosen on the module configuration page.

22.3 Adding a New Printer

If you have just connected a printer to your system or want to access one connected to another system on a local network, you must add it to the printer daemon's configuration before any program on your Linux system can print to it. To do this, follow these steps:

1. Click on the **Add a new printer** link on the module's main page. This will take you to the printer creation form shown in Figure 22.2 (the screenshot is taken from a system using the CUPS print system, so the user interface may not be the same on your Linux machine).



Figure 22.1 The Printer Administration module.

A screenshot of the Webmin interface titled "Create Printer". It contains three main configuration sections:

- Printer Configuration:** Fields for "Name" (empty), "Description" (empty), "Print banner?" (radio buttons for "Yes" and "Optional"), "Accepting requests?" (radio buttons for "Yes" and "No - Why not" with a text input field), "Printing enabled?" (radio buttons for "Yes" and "No - Why not" with a text input field), and "Default printer?" (radio buttons for "Yes" and "No").
- Print Destination:** Options for "Local device" (selected, "Parallel Port 1" dropdown), "Local file" (disabled), "Remote Unix server" (disabled), "Direct TCP connection" (disabled), and "Remote Windows server" (disabled). Sub-fields include "Printer", "Type" (dropdown set to "LPR"), "Port", "Printer", "User", "Password", and "Workgroup". A checkbox "Check if remote server is up?" is also present.
- Printer Driver:** Options for "None" (selected, note "(Remote or raw print)"), "Program" (disabled), and "CUPS-driver" (disabled). A dropdown menu shows printer models: "Alps MD-1080, Formatic + m20", "Alps MD-1280, Formatic + m20", and "Alps MD-2080, Formatic + m20".

Figure 22.2 The Printer Creation form.

2. Enter a unique name for the new printer (such as *epson* or *hp_laser*) into the **Name** field. This will be the name by which the printer is specified when using the **lpr** command or printing from other programs.
3. Enter a short description into the **Description** field, such as *Office Epson Stylus 740*.
4. If you want every print job to be preceded by a banner page containing the name of the file being printed and the user who printed it, set the **Print banner?** field to **Yes**. This is usually a waste of paper unless the printer is being used by a large number of people in a large organization.
5. To make this the default printer that will be used if no printer name is specified in the **lpr** command line, set the **Default printer?** option to **Yes**. This option is not available for all print systems.
6. When using some print systems, you can control the maximum size of a job that can be submitted to the printer using the **Max print job size** field. For a printer on your own personal machine, this should be set to **Unlimited**, but on a network with many users it may make sense to enter a lower number of 1 KB blocks.
7. If your system is using the Linux or LPRng print systems, you can enter multiple space-separated aliases for the printer into the **Alternate printer names** field. To make the printer the default, enter *lp* as one of the aliases.
8. If the printer is connected directly to your system, select the **Local device** option in the **Print Destination** section and select the parallel or USB port that it is on from the menu next to it.
If the device is not on the list, select **Local file** instead and enter the device path into its field, such as */dev/ttyS5*. You could also enter a filename to which to print, as long as it already exists and is writable by the print server daemon.
9. If the printer is attached to another system on a network (or is directly connected to the network itself), you must choose a protocol to print to it.
For a printer attached to a UNIX system, select **Remote Unix server** and enter the hostname of the server and the name of the printer on that server into the fields next to it. For most UNIX systems, the **Type** can be set to **BSD**, but if the remote server is running CUPS you can select **IPP** instead.
For a printer on a Windows system, select **Remote Windows server** and enter the hostname and printer name into the appropriate fields next to it. If the server requires clients to log in before printing, fill in the **User** and **Password** fields with a valid login for the Windows system. If you have multiple workgroups on your network, you may need to fill in the **Workgroup** field as well.
For some printers that can be plugged directly into the network, you must select the **Direct TCP connection** option and enter a hostname and port number into the fields next to it.
If the **hpnp** command is installed on your system, the option **Remote HPNP server** will be available so that you can print to HP network printers that use that protocol. If selected, you must enter a hostname and port number into the fields next to it.
10. To have Webmin check if the remote printer can actually be reached using the chosen protocol, click the **Check if remote server is up?** box.

11. If the printer supports PostScript, select the **None** option in the **Printer Driver** section. You should also select this option when printing to a remote UNIX server, as conversion from PostScript to the correct data format will be done on the server.
For printers that do not support PostScript and are connected directly to your system or accessed over the network using the **Direct TCP connection** or **Remote Windows server** options, you must select a printer driver (explained in the next step).
12. If your printer does not use PostScript and needs a driver, select the **CUPS driver** option. When using another print system, this option may be labeled **Webmin driver** or **Redhat driver** or **COAS driver** instead. Either way, next to it will be a list of printer models from which you can select the make and model of your printer.
If it does not appear in the list, try selecting the entry with the same manufacturer and closest model number that you can find (for example, if you have a *Footronic 810* and only *Footronic* models 800 and 1000 appear, select the model 800).
13. With print systems, other options such as DPI and paper size may be available under the printer model list. Select those that are appropriate for your printer.
14. Finally, click the **Create** button. If anything goes wrong (such as an inability to contact the remote print server or a failure to create the printer), an error message will be displayed. Otherwise, you will be returned to the module's main page which will now list your new printer.
15. If you are using the CUPS print system and have set a driver for the printer, click on the name of your new printer on the list to go to the printer editing form. At the bottom, below the printer model list, will be an additional set of fields for configuring things like the paper size, print quality, and paper type. Because the fields are dependent on the type of printer chosen, they are not displayed on the printer creation form. Set the paper size, DPI, and so on to whatever is appropriate for your system. The defaults will usually produce fast low-quality output, so if you want to use your printer's photo-quality mode on glossy paper, you will need to change them. When you are done changing the printer-specific options, click the **Save** button at the bottom of the page.
16. The newly created printer can now be printed to using the `lpr` command or any program that supports printing.

22.4 Editing an Existing Printer

Any printer created using Webmin or any other tool can be edited using the Printer Administration module. You can also temporarily disable a printer so that it no longer accepts jobs or sends them to the printer. To do this, follow these steps:

1. Click on the name of the printer on the module's main page. This will take you to an editing form, which is similar to the creation form shown in Figure 22.2.
2. To prevent users from submitting new jobs to the printer, set the **Accepting requests?** field to **No**. You can enter a reason why the printer is unavailable into the **Why not** field, which will be displayed to users who try to use the `lpr` command. This field, however, may not be available with some print systems.
3. To stop the printer from printing or sending jobs to a remote server, set the **Printing enabled?** field to **No**. This can be useful if the printer is going to be taken offline for

maintenance, as the queue will still accept jobs to be printed when the field is set back to **Yes** again. You can also enter a reason into the **Why not** field, which will be displayed when the print queue is displayed with the `lpq` command.

4. All other fields on the page can be changed, as explained in Section 22.3 “Adding a New Printer”. The only exception is the printer name, which cannot be modified after the printer is created.
5. When you are done changing the printer’s details, click the **Save** button. The changes will be made immediately and you will be returned to the module’s main page.

Existing printers can also be deleted by clicking the **Delete** button on the editing form. Any jobs in the printer’s queue will be deleted as well.

22.5 Managing Print Jobs

When a job is submitted to a printer, it is placed in the printer’s queue. It is removed only when it has been successfully printed or sent to a remote server. On a system with many users or a slow printer, the queue can grow quite large if jobs are being submitted faster than they can be printed.

You can use this Webmin module to list jobs in the queue for a printer, view their contents, or delete them. To manage these tasks, complete the following steps:

1. On the module’s main page, click on the **list** link under the **Jobs** column for the printer whose queue you want to manage. This will take you to a page listing all jobs currently being printed or waiting to be printed.
2. To view the contents of a print job, click on its size. Because most jobs are submitted in PostScript format, your browser must have a plug-in or helper application that can handle the format.
This is not possible for remote printers or on some print systems.
3. To delete a print job, click on its ID in the first column. Or to remove all the jobs in the queue, click the **Cancel all print jobs** button.

The print jobs page can also be used to submit a test page to the printer for verifying from within Webmin that it is working. To do this, follow these steps:

1. On the module’s main page, click on the **list** link under the **Jobs** column for the printer on which you want to print a test page.
2. Click on the **Print Test Page** button.
3. Select either the **Black and white Postscript page**, **Colour Postscript page**, or **Plain ASCII text** option to use one of Webmin’s built-in test pages. Or, select **Any uploaded file** and use the field next to it to choose a file on your system for printing.
4. Click the **Print page** button to submit the chosen page to the printer. A web page showing the output from the `lpr` command will be displayed so that you can see if any immediate errors occurred.

22.6 Configuring the Printer Administration Module

Like many Webmin modules, Printer Administration has several options that can be configured by clicking on the **Module Config** link on the main page. The options that you can safely change are listed under **Configurable options** on the configuration page, as shown in Table 22.1:

Table 22.1 Module Configuration Options

Show enabled and accepting status instead of driver?	If this option is set to Yes , the module's main page will display for each printer whether it is currently printing and if it is accepting new jobs. If set to No (the default), each printer's driver will be shown instead.
Seconds to wait before refreshing print queue	If Don't refresh is not chosen, then the print jobs page will be periodically refreshed automatically if it is displayed in your browser. The number entered for this option is the number of seconds between each refresh.
Sort printers by	If Name is selected, the list of printers on the module's main page will be ordered by name. Otherwise if Order in system is selected (which is default), they will most likely be displayed in the order that they were created.
On main page show	Normally the module's main page contains a lot of information about each printer, such as its destination, driver, and description. On some systems that have a large number of printers, this can be very slow—particularly on Solaris. If Just printer names is chosen for this option, only each printer's name will be shown on the main page, speeding up the display and reducing the page size.
Show number of jobs in queues on main page?	This field can be used to have the size of each printer's queue shown on the module's main page. However, it can make the page much slower to display if you have a large number of printers.

If you upgrade the print server daemon on your system, you will need to change some of the other module configuration options under **System configuration** so that it can be managed by Webmin. The most common upgrade is to CUPS, which can be installed on almost any Linux or UNIX system and is available as an optional package for many distributions. If you upgrade to CUPS, follow these steps to change the module configuration:

1. Click on the **Module Config** link on the module's main page to get to the configuration form.
2. Select CUPS from both the **Printer configuration style** and **Printer driver style** menus.
3. Change the **Printers file** field to **None**, as it is not needed by Webmin when using CUPS.

4. Set the **Directory containing interface programs** to the base directory under which all of CUPS's .ppd driver files can be found. This is usually `/usr/share/cups/model`, but may be different depending on how it was installed.
5. Set the **Command to start scheduler** to the init script command needed to start `cupsd`, such as `/etc/init.d/cups start`. If there is no such init script, just leave it set to **Determined by printer style**.
6. Similarly, set the **Command to stop scheduler** to the init script command that stops `cupsd`, such as `/etc/init.d/cups stop`.
7. Make sure the **Command to run after making changes** should be set to **None**.
8. Click the **Save** button to return to the module's main page. You should now be able to create and edit printers using CUPS.

Another popular print system that you may want to upgrade to is LPRng, particularly if you are running a UNIX variant with a poor print server daemon. If you do, the module must be reconfigured using the following steps:

1. Click on the **Module Config** link on the module's main page to get to the configuration form.
2. Set the **Printer configuration style** to LPRng.
3. Set the **Printer driver style** to Webmin. Make sure that **ghostscript** is installed on your system, as Webmin uses it to create drivers for non-PostScript printers.
4. Set the **Printers file** to `/etc/printcap`.
5. Change the **Directory containing interface programs** to **None**.
6. Enter the full paths to the `smbclient` and `gs` commands into the **Path to smbclient** and **Path to ghostscript** fields, respectively.
7. If `ghostscript` was compiled and installed manually, you may need to set the **Ghostscript font directories** and **Ghostscript library directories** options to colon-separated lists of directories that contain PostScript font files. These options are used to set the `GS_FONTPATH` and `GS_LIB` environment variables, respectively.
8. Set all the remaining options to their default values.
9. Finally, click the **Save** button to return to the module's main page.

22.7 Module Access Control

It is often useful to give a user the rights to view print queues and delete jobs, but not create or edit printers. This can be done using the Webmin users module, once you have created a user with access to the Printer Configuration module or edited an existing user to give him access. Chapter 52 explains how to do this in more detail.

Once a user with access to the module exists, you can limit which printers he can manage and what he can do to them by following these steps:

1. In the Webmin Users module, click on **Printer Administration** next to the name of the user or group.
2. Set the field **Can edit module configuration?** to **No**, so the user cannot change the print system or paths to configuration files.

3. You can limit the printers for which a user can edit the destination, driver, and other attributes by changing the **Printers this user can configure** field to **Selected** and choosing them from the list. This will not stop him from managing jobs on those printers though—the option in Step 4 controls that.

To prevent the user from managing any printers, choose **Selected**, but do not select any printers from the list. Be aware that a user who can edit or create a printer can gain `root` access by specifying his own driver program (which is typically run as `root`), or having the printer write to a system file such as `/etc/passwd`.

4. To limit the printers on which the user can manage print jobs, change the **Can cancel print jobs?** field to **Only on selected printers** and choose them from the list below. Or, select **No** to stop him from canceling or viewing the contents of jobs on any printer.
5. It is also possible to further restrict the jobs that can be managed using the **Manage print jobs owned by** field. By default, jobs submitted by any user on allowed printers can be cancelled. If the last option in this field is selected and a username entered into the field next to it, however, only jobs owned by that user can be managed.
You can also select the **Current Webmin user** option, which will limit the user to jobs submitted by a UNIX user with the same name as the Webmin user.
6. To prevent the Webmin user from creating new printers, set the **Can add new printers?** option to **No**. This should be done if he is not allowed to edit existing printers.
7. Because there is no reason why the user should need to stop or restart the scheduled print process, change the **Can stop or start scheduler?** field to **No**.
8. To hide printers on the main page on which the user is not allowed to edit or manage print jobs, set the **Show non-configurable printers?** option to **No**.
9. To stop the user from printing pages through Webmin, change the **Can print test pages?** option to **No**.
10. Finally, click the Save button to have your new restrictions activated.

22.8 Other Operating Systems

In addition to Linux, the Printer Configuration module is also available on several other UNIX operating systems. Because each has its own unique print system, the module's user interface is slightly different—just as there are differences between the Linux print systems such as CUPS and LPRng.

The supported operating systems and their differences are:

Sun Solaris and SCO UnixWare Solaris and Unixware have a very similar print systems to CUPS, so the Printer Configuration module has an almost identical user interface. One difference is the addition of a **Driver accepts** field on the printer creation form, which tells the print system what data format the driver program can handle. In most cases you should just select **postscript**—or if you want all data to be passed directly through to the printer without filtering, select **other** and enter *binary* into the field next to it.

Another unique feature is the ability to control which users can use each printer, using the **Access control** field on the creation form. The biggest omission on these UNIX variants is the **Direct TCP connection** destination type.

HP/UX and SGI Irix The print systems on these operating systems lack many options available on Linux, such as the **Description** field and **Banner** options. Once a printer has been created, it is impossible to change its destination or driver. Printing via a direct TCP connection is not supported either.

FreeBSD, NetBSD, OpenBSD and Apple MacOS X The print system on these operating systems is very similar to LPRng on Linux. You cannot, therefore, designate a printer as the default or enter a reason why a printer is unavailable or offline. It is possible, however, to specify a maximum print job size and enter alternative names for a printer.

Because none of these listed operating systems include printer drivers, Webmin has to create its own using the ghostscript package. If the module detects that the `gs` command is not installed, you will not be able to choose a driver when creating or editing a printer. Similarly, to be able to print to Windows servers, the Samba `smbclient` program must be installed and its path set in the module configuration page.

For all of these listed operating systems, the module will, by default, use their standard print systems. Therefore, if you have installed a different package such as CUPS or LPRng, the module configuration will need to be changed so that Webmin can configure it correctly. See Section 22.6 “Configuring the Printer Administration Module” for details.

22.9 Summary

This chapter has explained how to configure your Linux system with Webmin to print to either a locally connected printer or one attached to a print server of some kind. After completing it, you should understand how printing on UNIX works, how drivers are set up, and what differences exist between the various available print systems. You should also know how to restrict access to the module so that certain Webmin users can only manage particular printers or jobs.

Voicemail Server Configuration

If your system has a modem with voice capabilities attached, read this chapter to learn how to set up it up as an answering machine using the vgetty program.

23.1 The Voicemail Server Module

If you have a modem attached to your Linux system that supports voice recording and playback, the Voicemail Server module can be used to turn your computer into a powerful answering machine. Not all modems support voice, so check your user manual to make sure that yours does before trying to use this module. Some modems that require special drivers (often called Win-modems) are not generally usable on Linux at all, and so cannot be used with this module. Almost all modern external modems that attach to a serial port, however, will work fine. Some internal modems that emulate a serial port can be used as well.

A Linux system running as an answering machine is far more flexible than a traditional machine. The number of messages that you can store is limited only by hard disk space—messages can be viewed and listened to from any host on the network, and actions can be taken when a message is received (such as emailing it to an address). Like any answering machine, your system can be configured to answer the phone after a certain number of rings so that you have a chance to pick up the phone before the answering system kicks in.

The underlying software that makes all this possible is called vgetty, which is a modified version of the mgetty modem control program covered in Chapter 18 “PPP Server Configuration”. Not all Linux distributions include it, but it can be downloaded from rpmfind.net/ or the developer’s website at alpha.greenie.net/mgetty/. Webmin adds entries to the /etc/inittab file so that vgetty will be started at boot time and listen on the appropriate serial ports. This is exactly the same as the method used to set up mgetty, as explained in Chapter 18.

The Voicemail Server module can be found in Webmin under the Hardware category, and when you enter it the main page simply displays four icons. If the module detects that vgetty is not installed, the main page will display an error message instead, telling you that you need to install it before the module can be used. All of the actual configuration forms and pages can be reached by clicking on the appropriate icons.

23.2 Configuring Your System as an Answering Machine

Assuming you have a modem attached to a serial port on your system and plugged into a phone line, and that it supports voice recording and playback, you can set up your system as an answering machine by following these steps:

1. On the main page of the Voicemail Server module, click on the **Serial Port Configuration** icon. This will take you to a page listing any existing ports that have been configured for PPP or voicemail.
2. Click on the **Add a new serial port** link, which will bring up the port configuration form shown in Figure 23.1.



Figure 23.1 The serial port configuration form.

3. Set the **Serial device** to the port to which your modem or null-modem cable is connected. **Serial port 1** corresponds to the device file /dev/ttys0, and so on. For modems on serial devices not starting with /dev/ttys (such as USB modems), select the **Other device** option and enter the full device file path into the text field next to the menu.

4. If the **Rings before answering** field is visible, you can deselect **Global default** and enter the number of rings that your system will wait for before answering a call. This applies only to the modem on this serial port, however. If the field is not visible, or you want to use the same setting on all modems, you can set the number of rings in Step 6, instead.
5. If the **Answer mode** field is visible, just leave it set to **Global default**.
6. Click the **Create** button at the bottom of the page to return to the list of serial ports.
7. Return to the module's main page and click on the **Voicemail Server Options** icon. This will take you to the form shown in Figure 23.2.
8. Enter the number of rings that the server should wait for before picking up a call into the **Rings before answering** field, unless it was already set in Step 2. If you have multiple modems and want to set a different number of rings for each of them, check the **Can be set for each serial port** box.
9. In the **Answer mode** field, make sure that at least one of the menus is set to **Voice**. If this phone line is only going to be used for answering voice calls, you should set the first menu to **Voice** and leave the other two blank.
10. To limit the length of a message than can be left on your system, change the **Maximum message length** field. Entering too high a number could cause all of your disk space to be consumed by extremely long messages.
11. To stop very short messages from being saved, change the **Minimum message length** field. If a caller hangs up before the time specified in this field is elapsed, the recorded message will not be saved to a file.
12. The **Silence threshold level** field determines the percentage volume level below which vgetty treats recorded audio as silence. If the **Remove silence from end of messages?** field is set to **Yes**, any audio at the end of a message that falls below the threshold will be truncated.
13. To set the volume levels for recorded messages and for greeting messages played by the modem, set the **Recording volume level** and **Playback volume level** fields, respectively. Both can be set to either **Default** or to a volume percentage. Not all modems, however, support playback and recording volume configuration.
14. To have newly recorded messages emailed to you, change the **After recording message** field to **Email in WAV format to**, and enter your address into the field next to it. You can also select **Run command on message file** and enter the path to a program into its field. Whenever a message is recorded, the program will be run with the message file in RMD format as its first command-line argument.
15. Click the **Save** button to record your new configuration settings and return to the module's main page.
16. Click the **Apply Configuration** button to tell vgetty and init to use the new configuration. You can now try calling your phone number to test and see if the call is answered and a message recorded. Because no greeting message has been set yet, you will only hear a beep when the system is ready to record.

There are several things that can go wrong that cannot be detected until vgetty tries to communicate with your modem and answer a call. Fortunately, detailed logs are written to the file `/var/log/vgetty.ttynname` so that you can see what is going wrong. If your modem is on the first serial port, the log file will be `/var/log/vgetty.ttyS0`. Log in as root and use the `tail -f` command on it to monitor it when a call comes in, so you can see what is happening.

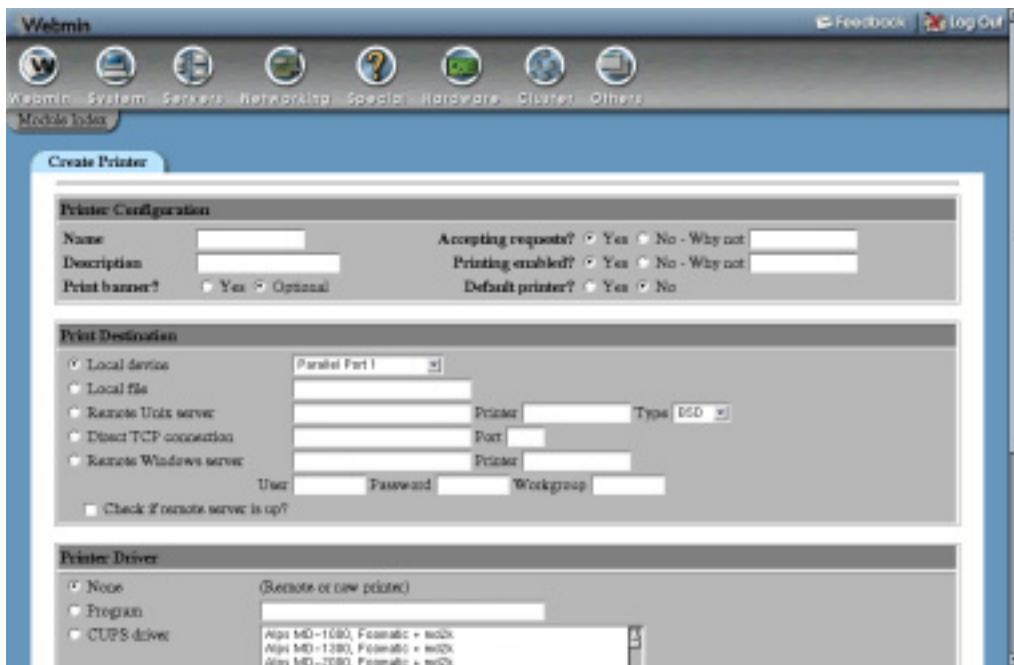


Figure 23.2 The voicemail server options form.

If your modem does not support voice playback and recording, an appropriate error message will be written to the log as soon as the **Apply Configuration** button is clicked. If this happens, there is nothing you can do apart from buying a new modem. Another common problem is a failure to play the greeting message, due to the same rate or compression format not being supported by your modem. See Section 23.4 “Setting a Greeting Message” for details on how to resolve this.

23.3 Listening to Recorded Messages

Every time a message is received, it is written to a file in the `/var/spool/voice/incoming` directory in RMD format. Fortunately, these files can be easily converted to more useful formats, like WAV, using commands like `rmdtopvf` and `pvftowav` that come with `vgetty`. Webmin does this for you automatically when you use it to listen to a message.

To view and manage recorded messages, follow these steps:

1. On the module’s main page, click on the **Received Messages** icon. This will take you to a page listing all available recorded messages, their sizes, and audio formats. The most recently recorded message is shown at the top of the table.
2. To listen to a message, just click on the date and time under the **Received at** column. Webmin will convert it to WAV format, and if your browser has been configured to play audio files in this format you should hear it immediately.
3. To delete messages, check the box to the left of each message in the table and click the **Delete selected messages** button.

Even if a message has been emailed to some address or had a program run on it when received, it will still be displayed on this page.

23.4 Setting a Greeting Message

When your system answers the phone, it can play a greeting message so callers know who they have reached. After the message, vgetty will play a short beep as well, so the caller knows when to start talking. By default, however, there is no greeting message, so callers will just hear a beep. Because this is not very friendly, you can use Webmin to set up one or more messages to be played when the call is answered. If multiple messages are set up, vgetty will choose one of them at random for each call.

To add a greeting message, follow these steps:

1. On the module's main page, click on the **Greeting Messages** icon. This will take you to a page listing all existing messages, if there are any.
2. Record a message in WAV format using some other program, such as the Windows Sound Recorder application. Make sure that the sample rate is the same as the rate used by recorded messages shown on the Received Messages page. If it is not, your modem will probably not be able to play it.
3. Back in Webmin, select the WAV file using the **Upload message** field. From the menu next to it, select the same audio format and number of bits as is used by recorded messages. Once again, the chosen format must be correct for your modem to be able to play the message.
4. Click the **Upload** message button to have the file converted to RMD format and added to the greeting messages list.

Existing greeting messages can be listened to by clicking on their filename from the list, which will cause Webmin to convert the chosen audio file back to WAV format before sending it to your browser. If you don't want to use some of the greeting messages anymore, just select the checkbox next to them and click the **Delete selected messages** button.

There is another way to create a greeting message that avoids any format or sample rate problems and does not require a sound card or microphone to be attached to your computer. An existing recorded message can be converted to a greeting by following these steps:

1. Call your own answering machine and leave the greeting message as a recording.
2. On the module's main page, click on the **Received Messages** icon and find the new recording on the list.
3. Select the checkbox next to it and click the **Convert selected to greetings** button. This will remove the message from the list, and add it to the list on the Greeting Messages page.

The only problem with converting messages like this is that they may be of poor audio quality or have silence or telephone sounds at the beginning and end.

23.5 Summary

This chapter has explained how any Linux system with a suitable modem attached can be converted into a simple voicemail server or answering machine. It has also covered the playing back of messages, recording of greetings, and configuration of automatic email notification.

Remote Shell Login

This chapter explains the various ways that you can make a remote shell log in to your system through Webmin.

24.1 The SSH/Telnet Login Module

If your system is running either an SSH or telnet server (as most do by default), you can use Webmin’s built-in SSH and telnet client module to do a normal shell login. When you use it to log in, the connection is coming from the client host on which your web browser is running—just as if you ran a normal telnet client program—because the module uses a Java applet. This means that if there is any firewall blocking telnet or SSH access from the client, this Webmin module will not be able to get around it.

To use the module, click on its icon under the Others category in Webmin. The main page simply contains a Java applet that will, by default, make a telnet connection to the server running Webmin. If your browser does not support Java, an error message will be displayed instead. If the applet loads and is able to connect, it should display a login prompt—just click on the applet and enter your username and password to log in. Figure 24.1 shows an example.

Not all versions of Linux have a telnet server running by default. Many new distributions include an SSH server instead, which means that the applet will be unable to make a telnet connection. If this happens, you need to reconfigure it as explained in Section 24.2 “Configuring the SSH/Telnet Login Module”.

24.2 Configuring the SSH/Telnet Login Module

This module has several configuration options that control its user interface and connection. To edit them, click on the **Module Config** link in the top-left corner of the main page. The editable options that will be displayed are shown in Table 24.1.

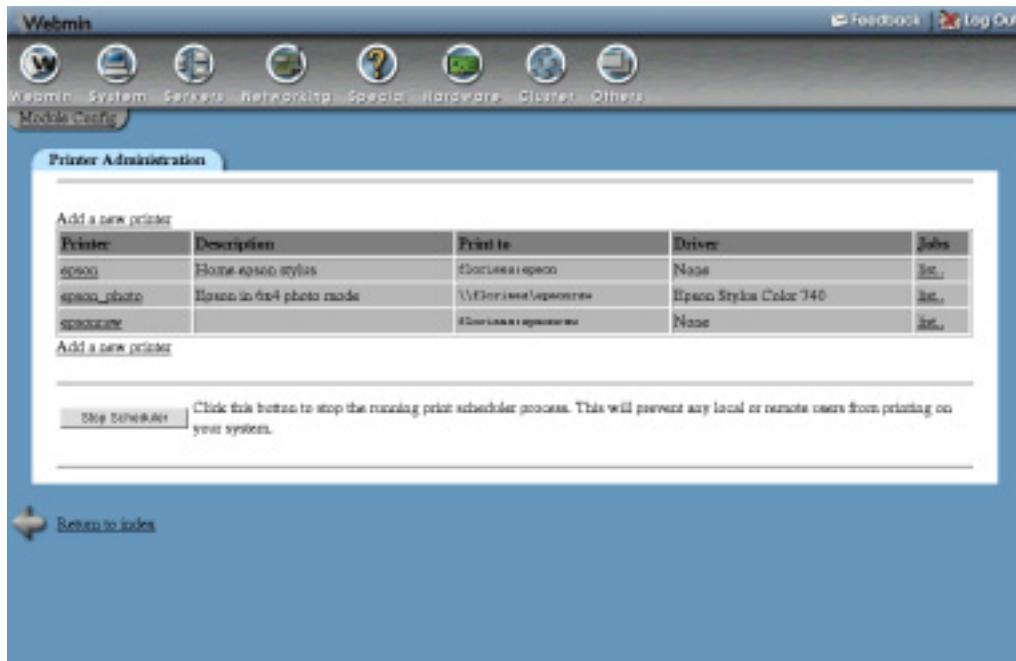


Figure 24.1 The SSH/Telnet Login module.

Table 24.1 Module Configuration Options

Hostname to connect to	Normally this field is set to Automatic , which tells the SSH/telnet applet to connect to the Webmin server from which it was loaded. You can enter a different hostname or IP address instead to log in to a different system, but this will usually not work due to the restrictions on where a Java applet can connect.
Port to connect to	When this field is set to Default , the applet will connect to either port 23 for telnet or 22 for SSH. These are the standard ports for those servers so this field can almost always be left alone. If they have been changed on your system, however, this module configuration option will have to be changed as well so that the applet can connect.
Connection type	For the applet to make an SSH connection, select Secure Shell . To stick with telnet, choose Telnet instead. If SSH mode is chosen, your system must be running an SSH server that supports version 1 of the protocol, as the applet does not support version 2.

Table 24.1 Module Configuration Options (Continued)

Applet size	This field has three options: 80x24 characters —The applet will be sized to 80 characters wide and 24 high, the same as a standard telnet or shell window. Maximum —The applet will take up all available space in the browser. Custom size —The size is determined by the values entered into the next field. If the Separate window mode option is enabled, this field is irrelevant.
Custom width x height	If the Applet size field is set to Custom size , you must enter a width and height separated by an x into this field, such as <i>640x480</i> .
Font size in points	The size of the font used by the applet. If this is changed from the Default setting of 11 points, more or fewer rows and columns will fit into the applet.
Separate window mode	If set to Yes , all that will appear on the module's page is a button labeled Connect . Only when it is clicked will a separate window be opened to log in via SSH or telnet. The window can be resized manually as soon as it is opened.
Test telnet or SSH server	Normally this field is set to Yes , which causes Webmin to check to see if there really is an SSH or telnet server running on your system. Because this test can sometimes fail incorrectly due to a firewall preventing your system from connecting to itself, you can set it to No to disable the test.

24.3 The Command Shell Module

One problem with the SSH/Telnet Login module is its inability to connect if there is a firewall of some kind blocking telnet or SSH connections to your system. Even though the rest of Webmin may work fine using HTTP connections, the ports used by the applet may not be available. Even though it is possible to do almost everything in Webmin that you can do at the command line, it is sometimes useful to have a shell prompt for executing UNIX commands.

To get around firewall restrictions that prevent an SSH or telnet connection, you can use the Command Shell module, found under the Others category. It allows you to enter shell commands into the field next to the **Execute command** button that are run when the button is clicked or the return key pressed. All output from the command is displayed in the **Command history** section at the top of the page.

You can rerun old commands by selecting them from the menu next to the **Execute previous command** button, and then clicking it. If the command history becomes too large, it can be

wiped clean using the **Clear history** button. This will not affect the menu of previously run commands.

The module's biggest limitation is that interactive commands like `vi`, `passwd`, and `telnet` cannot be run. There is no support for providing input to a command once it has started, so you are limited to noninteractive programs like `cp`, `ls`, and `rm`.

24.4 The Shell In A Box Module

This module combines the best features of both SSH/Telnet Login and Command Shell—it allows you to make a fully interactive login that is tunneled though an HTTP connection, thus avoiding any firewall restrictions. It is not included as one of the standard Webmin modules, but you can download it from www.webmin.com/download/modules/shellinabox.wbm.gz. See Chapter 51 for instructions on how to install it.

When you enter the module, its main page is taken up entirely by a Java applet. To start the login process, click the **Connect** button in the lower right-hand corner. A normal `login:` prompt should appear at the top of the window, allowing you to enter a username and password to log in and get a shell prompt. When you are done, just click the **Disconnect** button to log out.

The module's biggest disadvantage is that it uses compiled Linux x86 code, and so cannot be run on other UNIX systems or on non-PC hardware. It also uses up a lot of CPU time on the server due to the high frequency of HTTP requests that it makes.

24.5 Summary

This simple chapter explains the ways that Webmin can be used to log in to your system via SSH or telnet, even when you do not have a normal client for either of these protocols available. It also mentions modules for executing simple commands from a web interface, and logging in remotely even when SSH or telnet connections are blocked.

Running Custom Commands

This chapter covers Webmin’s Custom Commands module, which can be used to create buttons for running frequently used shell commands.

25.1 The Custom Commands Module

Most system administrators like to create shell scripts to perform common tasks, such as backing up a database or adding a new user of some kind. Because every system and organization is different, there will always be tasks that a generalized tool like Webmin cannot do as easily as a simple, customized script. Unfortunately, scripts run at the command line are not easy for an inexperienced user to use.

The Custom Commands module allows you to create simple web interfaces for shell scripts and commands so they can be run from within Webmin at the click of a button. It also allows you to define the parameters of various types for each command that can be entered by the user and substituted into the shell command. This can be used to provide additional arguments or input to the scripts that are run, depending on selections made by the user before running it.

Another feature of the module is the ability to define file editors so frequently changed files can be edited through Webmin’s web interface. You can also define commands to be run before and after the file is edited so it can be validated, copied, or backed up before editing.

Possibly the most useful feature of the module is its access control support. You can grant other Webmin users the rights to use some or all of the commands and editors, while giving only yourself and other trusted administrators permissions to create and edit commands. This means that the other users can only execute the scripts and edit the files that you allow them to, but with full root privileges.

Unlike most other modules, this one does not deal with the configuration of some separate server or service, therefore it has the exact same user interface and functionality on all versions of UNIX on which Webmin can run.

When you enter the module from the Others category, its main page shows all existing custom commands and file editors, along with their parameters. Figure 25.1 shows an example from a system with one file editor and eight commands defined—two of which have a parameter. If you have not used the module before, however, the page will be empty.

You can run any command shown on the main page by just clicking its button. If the command has parameters fields or choices, however, you must fill them in or make the appropriate selections before running it. When the button is clicked, you will be taken to a page showing all output from the command so you can see if it succeeded or failed.

To use a file editor, just click on its button on the main page. This will take you to an editing form showing the current file contents, which you can change freely. When done, click the **Save** button below the text box to write out the new file contents.



Figure 25.1 The Custom Commands module.

25.2 Creating a New Command

To create a new command that can be run using a button on the module's main page, follow these steps:

1. Click on the **Create a new custom command** link above or below the existing buttons. This will bring up the creation form shown in Figure 25.2.
2. Enter a short description for your command into the **Description** field. Whatever text you enter will appear on the command's button on the main page. You can also enter additional text (including HTML tags) into the larger text box below it, to be displayed underneath the button.

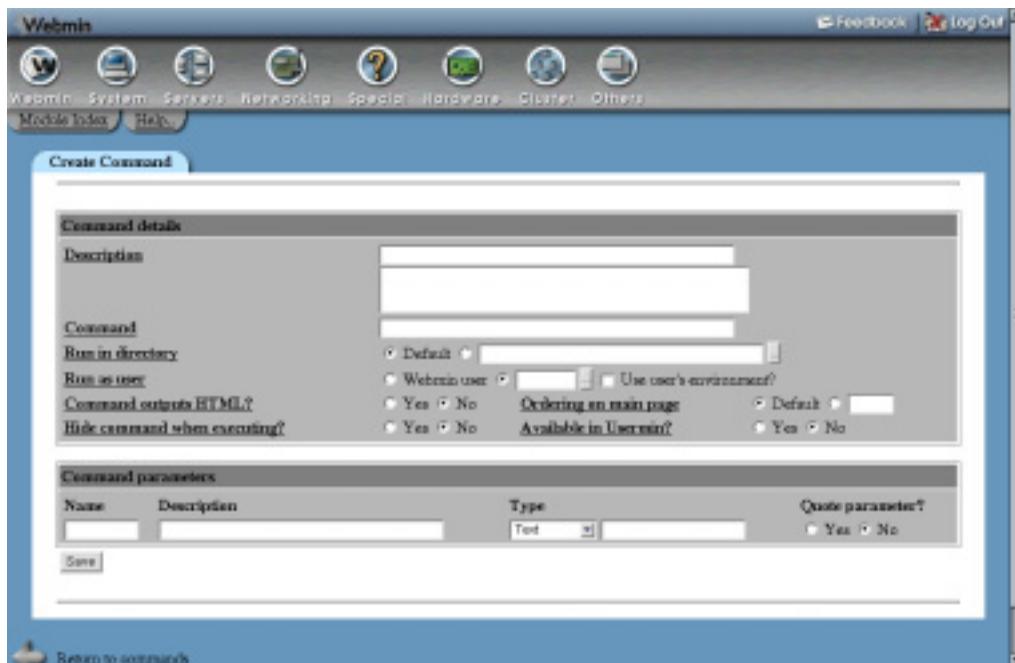


Figure 25.2 Creating a new custom command.

3. In the **Command** field, enter the shell script or command that you want to execute. All standard shell metacharacters are supported, such as |, &, <, and >. To enter multiple commands, separate them with ; or &&.

If your command has parameters (see Step 10) they will be converted into environment variables when the command is run. So, if you have a parameter called `foo`, all occurrences of `$foo` in the command string will be replaced with whatever the user enters for that parameter. For example, a command that allows the user to finger any user on the system might look like `finger $user`.

4. By default, the command will run in the Webmin directory for this module. To change this, deselect **Default** for the **Run in directory** field and enter a different path into the text box next to it.
5. In the **Run as user** field, enter the name of the UNIX user that the command should run as. You can select Webmin user instead, which will cause it to run as the UNIX user with the same name as the Webmin user who runs it.

When the command is executed, it will not normally have access to the same environment variables that the UNIX user would have if he logged in via telnet or SSH. If you check the **Use user's environment** option, however, then all variables set in the user's `.profile`, `.cshrc`, and other login files will be available. Webmin runs the command with `su`, which switches to the user, executes his shell and then executes the command.

6. If your command produces HTML output that you want to appear in the browser when it is run, change the **Command outputs HTML?** field to **Yes**. Otherwise Webmin will

escape all HTML tags in the output, which is the correct thing to do for commands that produce just normal text.

7. To control the placement of the new command on the module's front page, enter a number for the **Ordering on main page** option. Commands are ordered so that those with the highest number appear first. If **Default** is chosen, the ordering number is taken to be zero.

If you do not set the ordering number for any of your custom commands, they will be displayed in the order that they were created.

8. To prevent the user from seeing the actual shell command being run when its button is clicked, set the **Hide command when executing?** field to **Yes**. This is a good idea if your command contains passwords or other sensitive information you want to hide from the user.
9. To have the command appear in Usermin's Custom Commands module, change the **Available in Usermin?** field to **Yes**. See Chapter 47 for more information on how to install and configure Usermin.

10. If you want your command to have parameters that the user can set on the main page, you need to fill in the **Command parameters** section. Each row in Table 25.1 defines one parameter, and for each parameter the following information must be entered:

Name A short, unique name for this parameter, which can be used in the **Command** field (prefixed with a \$) to indicate where the value entered by the user should be substituted. The name should be made up of only letters, numbers and the underline (_) character.

Description The text that will label the parameter on the module's main page. This can contain any characters including HTML tags, but should not be too long.

Type This menu controls how the parameter is displayed on the module's main page, and what inputs are allowed. The most common choice is **Text**, but all available options and their meanings are covered in Section 25.3 "Parameter Types".

Quote parameter? If set to **Yes**, the value entered by the user will be enclosed in quotation marks (" ") before substitution.

When creating a new command, only one empty row for entering a single parameter is available. To add more, you will need to re-edit the command after saving it.

11. Finally, when you are done entering the details of your new command, click the **Create** button. As long as there are no errors in the form, you will be returned to the module's main page on which the new command button should be visible

Once a command has been created, you can edit it by clicking on the **Edit command** link below it on the module's main page. All the fields described above can be changed, and an additional parameter added. Once you are done making changes, click the **Save** button at the bottom of the page. To get rid of the command, click the **Delete** button in the bottom-right corner, instead.

25.3 Parameter Types

For each parameter in a command, you can choose a type from its menu under the **Type** column. The available options and their meanings are listed in Table 25.1.

Table 25.1 Custom Command Parameter Types

Text	The parameter is a text field, into which any string can be entered.
User	The parameter is a small text field with a user selection button next to it. Only valid UNIX users can be entered or selected from the pop-up user window.
UID	Like the User option, but the username entered will be converted to a UID for substitution into the command when it is run.
Group	The parameter is a small text field with a group selection button next to it. Only valid UNIX groups can be entered or selected.
GID	Like the Group option, but the group name entered will be converted to a GID for substitution into the command when it is run.
File	A text field with a file chooser button next to it. No validation is done to check that an actual file or valid filename is entered.
Directory	Like the File option, but the chooser button pops up a directory chooser instead.
Option..	The parameter is displayed as a pair of radio buttons, labelled Yes and No . If Yes is chosen, the text entered in the field next to the type menu on the command creation form will be substituted into the command string. If No is chosen, an empty string will be substituted instead. This type can be useful for optional shell command arguments—for example, in a command like <code>rm \$force /some/directory</code> . In this example, the <i>force</i> parameter would use the Option type and have <i>-f</i> entered into the text field next to the type menu.
Password	Like the Text type, but an HTML password field is used instead to hide the text entered by the user.
Menu..	If this type is chosen, the parameter is displayed as a drop-down menu in which the choices are taken from the file entered in the field next to the type menu. Each line in the file defines one menu entry. If the line contains a comma, the text after the comma is what appears to the user in the menu, while the text before it is the actual value to which the parameter is set when the command is run. An example file might contain the following lines: <pre>jcameron,Jamie Cameron emily,Emily Cameron</pre> The menu that appears on the module's main page would contain the choices Jamie Cameron and Emily Cameron , but the actual parameter passed to the command would be either <code>jcameron</code> or <code>emily</code> .
Upload	This type displays a file upload input that the user can use to select a file on his PC. When the command is run, the file is uploaded to the server and placed in a temporary file. The full path to this file is then used as the parameter when the command is run, so that it can be copied to some directory, converted to a different format, or whatever you like. When the command completes, the temporary file will be deleted.

Table 25.1 Custom Command Parameter Types (Continued)

Text box	A parameter of this type is shown as a text box into which anything can be entered. However, any newline characters are replaced with spaces before the parameter is passed to your command.
-----------------	--

25.4 Creating a New File Editor

To add a new button to the module's main page for editing a file, follow these steps:

1. Click on the **Create a new file editor** link above or below the existing buttons. This will bring up the editor creation form shown in Figure 25.3.
2. Enter a short description for the file to be edited into the **Description** field. Whatever text you enter will appear on the editor's button on the main page. You can also enter additional text (including HTML tags) into the larger text box below it, to be displayed underneath the button.
3. Enter the full path of the file to be edited into the **File to edit** field. The file does not necessarily have to exist yet.
4. To have the file's owner changed when it is saved, set the **File ownership** field to **User** and enter a UNIX username and group name into the fields next to it. This is especially useful when editing a file that does not yet exist, so the ownership of the newly created file is set properly.
If you leave the field set to **Leave as is**, the file's ownership will not be changed when it is saved. Newly created files will be owned by `root`.
5. To have the file's access permissions changed when it is saved, set the **File permissions** field to **Set to octal** and enter the permissions (like `700` or `664`) into the field next to it. If you select **Leave as is**, the file's permissions will not be changed when it is saved. The permissions on newly created files depend on the Webmin processes' umask.
6. To have a command run just before the file is saved by the user, fill in the **Command to run before saving** field. This could be useful for making a backup copy, checking the file out of RCS, or anything else that you can come up with.
7. Similarly, to have a command run just after the file is saved fill in the **Command to run after saving** field. This can be useful for validating the file's contents, copying it to another system, or checking it back into RCS.
8. To control the placement of the new editor's button on the module's front page, enter a number for the **Ordering on main page** option. Commands and editors are ordered so that those with the highest number appear first. If **Default** is chosen, the ordering number is assumed to be zero.
If you do not set the ordering number for any of your file editors, they will be displayed in the order in which they were created.
9. To have the editor appear in Usermin's Custom Commands module, change the **Available in Usermin?** field to **Yes**. See Chapter 47 for more information on how to install and configure Usermin.
10. Finally, click the **Save** button. If there are no errors in the form, you will be returned to the module's main page which will include a button for the new editor.

Once an editor has been created, you can edit it by clicking on the **Edit file editor** link on the module's main page. Once you are done making changes, click the **Save** button at the bottom of the page. To get rid of the editor, click the **Delete** button in the bottom-right corner instead.



Figure 25.3 Creating a new file editor.

25.5 Module Access Control

The access control options in the Custom Commands module are designed to allow a master Webmin user to give some other users the rights to run selected commands, but not edit or create them. From a security point of view, it makes no sense to give an untrusted user permissions to create his own custom commands because that would allow him to run any command as `root` and so compromise the security of the entire system. Similarly, you can restrict the file editors that a Webmin user can use, and prevent him from creating new editors.

Once you have created a user or group with access to the Custom Commands module (as explained in Chapter 52), the steps to follow to limit his access are:

1. In the Webmin Users module, click on **Custom Commands** next to the name of the user or group to which you want to grant access. This will bring up the access control form for the module.
2. Change the **Can edit module configuration?** field to **No**.
3. Unless you want the user to be able to run all commands and use all editors, set the **Commands this user can run** field to **Selected** and choose those that he should be allowed to use from the list provided. You can also choose **All except selected** and select from the list the commands that the user should not be allowed to use. All others will be available.

4. Change the **Can create and edit commands?** field to **No**.
5. Click the **Save** button. The access control settings will be activated and you will be returned to the main page of the Webmin Users module.

If you want to grant access to selected custom commands and editors to a large number of users, a better solution may be to install Usermin, which allows any UNIX user to log in. Any command for which the **Available in Usermin?** field is set to **Yes** will be visible in Usermin's Custom Commands module and work in exactly the same way. See Chapter 47 for more information on Usermin and how it can be configured to limit which UNIX users can run custom commands.

25.6 Configuring the Custom Commands Module

This module has several configuration options (shown in Table 25.2), which you can edit by clicking on the **Module Config** link on its main page.

Table 25.2 Module Configuration Options

Main page shows	When this option is set to All commands and parameters , the module's main page will behave as documented in this chapter. Every command and editor and their parameters will be shown. However, if Links to commands is chosen the page will only display a table of commands and their descriptions. To actually set parameters and run a command, you must first click on it to go to separate page. This mode is useful if you have a large number of commands with lots of parameters, and want to keep the size of the module's main page down.
Width of file editor window	This field can be used to change the width of the text box used by file editors.
Height of file editor window	This field can be used to change the height of file editor's text boxes.
File editor wrap mode	This option controls the text wrapping mode that affects lines longer than the width of the text box used by file editors. The default of Soft will cause lines to be wrapped for display, but not when they are actually saved. The Hard option will also wrap lines in the saved file. The Off option turns off wrapping altogether and forces the use of the scrollbar to view long lines.

25.7 Summary

After reading this chapter, you should be able to create your own custom command buttons, which run shell commands and can take multiple inputs from the user as parameter. You will also be able to create file editor buttons, for easily editing common files through a web interface. Finally, you should understand how to restrict access to these commands and editors, so that certain Webmin users (or Usermin users) can run them without being able to define their own.

Webmin’s File Manager

This chapter documents the File Manager module and its features, such as copying and pasting, ACL and EXT attribute editing, and file sharing.

26.1 The File Manager Module

Under the Others category in Webmin is a module that is quite different from any of the others. Instead of configuring some server or service, it allows the user to view and manipulate files on the server through a Java applet file manager. The user interface is similar to the old Windows explorer—on the left is a tree of directories, and on the right is a list of files in the current directory. At the top is a row of buttons on a toolbar that is used for carrying out various operations on selected files. Figure 26.1 shows an example.

Unlike other modules, this one only has a single page that is taken up entirely with the Java applet. To return to Webmin’s main menu, you have to click on the **Index** arrow in the top-left corner. Naturally, if your browser does not support Java then the applet cannot be used.

The File Manager module’s user interface is almost exactly the same on all versions of UNIX. The only differences are that some of the **EXT**, **ACL**, and **Attr** buttons (described in Section 26.10 “Editing File ACLs”) may not exist on some operating systems. This is because the filesystems on those UNIX variants do not support the extended attributes that the buttons allow you to configure.

26.2 Navigating Directories and Viewing Files

When you first load the file manager, the right-hand pane will display the contents of the root directory on your system. To enter another directory, just double-click on it in the list. To go back up a directory, double-click the **..** link at the top the current directory’s listing.

You can also view the contents of a directory by clicking on it in the tree in the left-hand pane. Double-clicking will open the directory in the tree, causing any subdirectories under it to

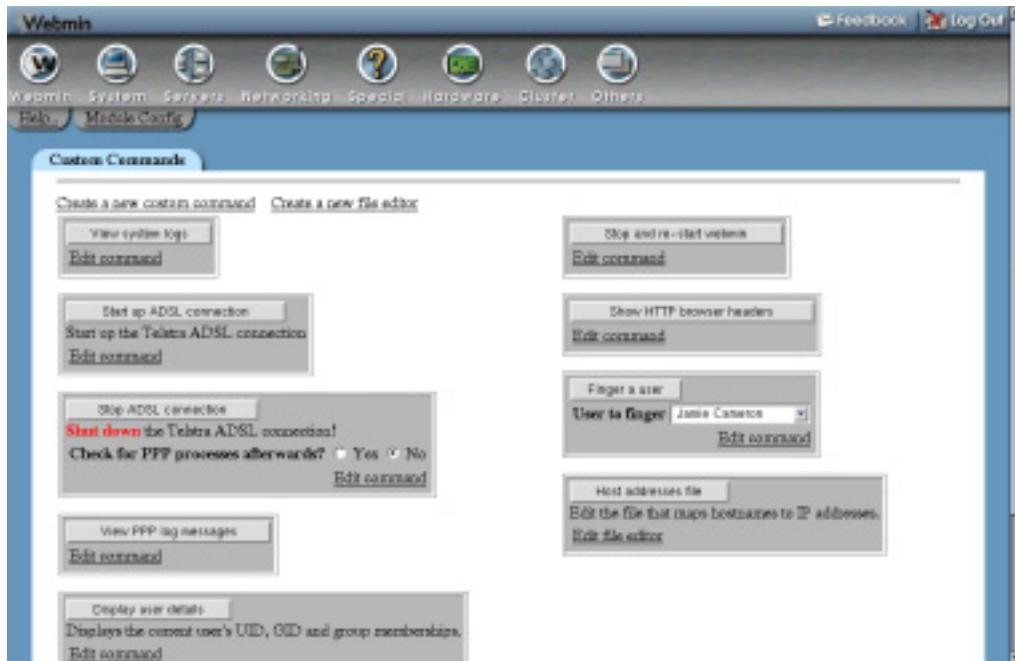


Figure 26.1 The File Manager module.

appear. Double-clicking again will close it. Whenever you enter a directory using the right-hand pane, it will be opened in the tree on the left as well. Similarly, when the .. link is double-clicked to go back to the parent, the old directory will be closed in the tree.

It is also possible to jump to any directory on your system by entering its path into the text field above the right-hand directory listing. Assuming that it actually exists, Webmin will open all parent directories in the tree and displays its contents in the list on the right.

To speed up the user interface, the file manager caches the contents of all directories that you view while using it. This means that if a file is created, modified, or deleted on the server, it will not be reflected in the directory listing until you click the **Refresh** button on the toolbar.

The contents of any file on your system can be displayed by double-clicking on it in the right-hand pane. A separate browser window will be opened and the contents of the file will be displayed by your browser. Any file type that the browser supports, therefore, can be viewed using the file manager.

If you want to download a file from your Webmin system to the host that your browser is running on, hold down the shift key while double-clicking on the file. The browser should prompt you to save the file instead of opening a window to display its contents. You can also force a download by selecting a file from the right-hand pane and clicking the **Save** button on the toolbar at the top of the file manager window.

26.3 Manipulating Files

The File Manager module allows you to rename, move, and copy files in the just the same way that any other file manager would. To select the file that you want to manipulate, just click on it in the right-hand pane. To select multiple files, hold down the control key while clicking, or hold down the shift key to select an entire range.

To move files to a different directory, select one or more and click the **Cut** button on the toolbar. Then navigate to the destination and click the **Paste** button. If a file with the same name already exists, Webmin will prompt you to rename the pasted file to avoid the clash. If you choose not to rename, the file in the destination directory with the same name will be overwritten.

To copy files, select them in the right-hand pane and click the **Copy** button. Then go to the directory to which you want them to be copied, and click **Paste**. As when moving files, you will be prompted to rename any that clash with files that already exist in the destination directory. Multiple copies of a file can be made by pasting in different directories. To create a copy of a file in the same directory, just select it, hit **Copy** and then **Paste**, and enter a new filename.

You can delete one or more files and directories by selecting them and clicking the **Delete** button on the toolbar. Before they are actually removed, a confirmation window listing all chosen files will be displayed. When the **Delete** button in the window is clicked, all chosen files, directories, and their contents will be permanently deleted.

A single file can be renamed by selecting it in the right-hand pane and clicking the **Rename** button on the toolbar. This will bring up a window containing the current filename and a text box for entering a new name. If the new name is the same as an existing file in the same directory, it will be overwritten when the **Rename** button in the window is hit.

26.4 Creating and Editing Files

The File Manager module offers two methods for creating new files—you can either create a text file from scratch, or upload data from the host on which your web browser is running. To create a new empty text file, click on the **New** document button on the toolbar to the right of the **Delete** button. This will bring up a window in which you can enter the full path to the file and its contents. When you are done editing, click the **Save** button at the bottom of the file creation window.

To upload a file from the PC on which your browser is running, click the **Upload** button on the toolbar. This will open a small browser window with two fields. The **File to upload** field is for selecting a file on your PC, while the **Upload to directory** field is for entering the directory to which the file will be uploaded. When both fields have been filled in, click the **Upload** button to have the file sent to your Webmin server. Once the upload is complete, the directory list will be updated to show the new file.

Because many people run their web browsers on the Windows operating system, which uses a different text file format from UNIX, there is an option in the upload window to convert the uploaded file to the correct format. This **Convert DOS newlines?** field should only be set to **Yes** when uploading a text file from a Windows system. Enabling conversion when uploading binary files will cause them to be corrupted.

The file manager can also be used to edit existing text files on your system. To do this, select a file in the right-hand pane and click the **Edit** button on the toolbar. A window showing its current contents will be displayed, allowing you to edit the file as you wish. When done, click the

Save button to have it written back to the server. Do not attempt to edit and save nontext files, as their contents will be corrupted.

Any existing file can be renamed simply by selecting it in the right-hand pane and clicking the **Rename** button on the toolbar. This will bring up a window displaying the current filename and prompting for a new one. Click the **Rename** button in the window after entering a new name to have it changed.

26.5 Editing File Permissions

Each file or directory on a UNIX filesystem is owned by a single user and group and have a set of permissions that determines who can access it. Normally these are changed by the chown and chmod commands, but you can edit them in the file manager as well. To do this, select a single file from the right-hand pane and click the **Info** button on the toolbar. This will bring up the permissions window shown in Figure 26.2.



Figure 26.2 The file permissions window.

The **File** section of the window displays its full path, size, type, and last modification date. The **Permissions** section contains checkboxes that control which users can read, write, and execute the file. These are the same permissions that you can change at the command line with the chmod command. As they are selected and deselected, the octal permissions that would normally be used with chmod are shown in the **Octal** field below.

To change the file's owners, enter new user and group names or IDs into the **User** and **Group** fields in the **Ownership** section of the window. For executables, you can also control which user the program runs as using the **Execute as user** and **Execute as group** fields. Because these options correspond to chmod permissions, changing them will update the **Octal** field as well.

When editing a directory, the checkboxes available are slightly different. The execute permission is replaced with list, an **Only owners can edit files** box is added, and the **Execute as** checkboxes are replaced with **Files inherit group**. These all correspond to standard UNIX file permissions that any system administrator should already be familiar with.

If changing the permissions and ownership of a directory, you can also choose to change those of any subdirectories and files that it contains. The **Apply changes to** menu determines to which files and directories the permissions are applied, and has three options:

This directory only The ownership and permissions chosen will be set on the selected directory only.

This directory and its files The ownership and permissions will be set on the chosen directory and all files that it contains. Subdirectories and their files will not be effected.

This directory and all subdirectories Ownership and permissions will be set on the chosen directory and all files and subdirectories that it contains.

If the file that was selected when the **Info** button was clicked is actually a symbolic link, the window will contain an additional **Link to** field that can be changed if you want to edit the link destination. Changing the permission and ownership fields is pointless, as they cannot be edited for symbolic links on UNIX systems.

26.6 Creating Links and Directories

The file manager can be used to create a new symbolic link in the current directory by following these simple steps:

1. Navigate to the directory in which you want the link to be created, and click the **New link** button on the toolbar.
2. In the window that appears, enter the path of the new link file in the **Link from** field.
3. Enter the path to an existing file or directory that you want the link to point to into the **Link to** field.
4. Click the **Create** button to have it created on the server and added to the directory listing.

New directories can also be created using the following steps:

1. Navigate to the directory under which you want the new subdirectory to be listed, and click the **New directory** button on the toolbar.
2. Enter the full path to the directory into the **New directory** field.
3. Hit the **Create** button to create it.

26.7 Finding Files

The file manager can be used to search for files or directories on your system that match certain criteria. This can be useful if you know the name of a file but not the directory in which it is located, or if you want to find files owned by a particular user or larger than a certain size. To search for files, follow these steps:

1. Click on the **Find** icon on the toolbar, which will bring up a search window.
 2. In the **Search directory** field, enter the directory under which the files you are looking for are listed. To search the entire system, just enter `/`. This may take a long time, however, on a server with large filesystems.
 3. To search by filename, enter a pattern into the **For files matching** field. This can be something like `*.txt` or `foo?.c`. If the field is left blank, filenames will not be included in the search criteria.
 4. To find files owned by a particular user, enter the username or ID into the **Owned by user** field.
 5. Similarly, to find files owned by group, enter its name or ID into the **Owned by group** field.
 6. To limit the search to normal files, directories, or some other type of file, select it from the **File type** field.
 7. If you want to find files larger than a particular size, change the **File size** field to **More than** and enter the minimum size in bytes into the adjacent field. To find those smaller than a particular size, select **Less than** and enter the maximum size into the field next to it.
 8. To prevent filesystems mounted under the search directory from being checked, change the **Search past mounts** option to **No**. This can be useful if you want to avoid searching NFS filesystems, which can be much slower than those mounted from local disks.
 9. Finally, click the **Search Now** button. When the search is complete, all files and directories that match all of the chosen criteria will be displayed in the window under the **Search results** tab. You can double-click on one to have the file manager automatically navigate to the directory that contains it, and select it in the right-hand pane.
- To do another search, click back on the **Search criteria** tab and follow the preceding steps again.

In the background, the file manager's search function uses the UNIX `find` command to locate files matching the criteria that you enter. All of the available options correspond to command-line options to `find`, such as `-name`, `-user`, and `-group`.

26.8 Editing EXT File Attributes

Several UNIX filesystem types support special attributes on files beyond those that can be set with the normal `chmod` and `chown` commands. On Linux `ext2` and `ext3` filesystems, each file has several special options that are normally set with the `chattr` command. Assuming your system has at least one filesystem of this type, you can change the EXT attributes for files that it contains by following these steps:

1. Select the file that you want to modify in the right-hand pane, and click the **EXT** button on the toolbar. This will bring up a window showing attributes that are currently set, assuming that the file is on an `ext2` or `ext3` filesystem.
2. To stop the file's last access time being updated very time it is read, turn on the **Do not update access times** option. This can prevent a lot of useless disk writes on files that are read frequently.
3. To stop processes modifying the contents of a file, check the **Can only append to file** option. This is useful for logfiles that you want to save from truncation or overwriting.
4. To have the kernel automatically and transparently compress the contents of a file, turn on the **Compress data on disk** option. This will only have an effect if your kernel supports transparent file compression.
5. To stop a file being read by the `dump` backup command (explained in Chapter 14), turn on the **Do not backup with dump** option.
6. To prevent a file from being modified or deleted, check the **Do not allow modification** option.
7. To have the kernel overwrite the disk blocks containing the file when it is deleted, turn on the **Zero blocks when deleting** attribute.
8. To force any writes to the file to be written to disk immediately, turn on the **Always sync after writing** option. Normally, the kernel buffers data for writing to disk when it is most convenient.
9. To have the kernel save the contents of the file when it is deleted, turn on the **Save contents for undeletion** option.
10. Finally, click the **Save** button to have your changes applied to the file.

Because all the preceding attributes can be changed at the shell prompt using the `chattr` command, making a file unchangeable or setting it to append-only mode does not provide any protection against someone who has `root` access to your system.

26.9 Editing XFS File Attributes

On Linux and IRIX `xfs` filesystems, files have totally different kinds of attributes. Every file or directory can have an unlimited number, each of which is simply a mapping between a text name and value. Normally, the `attr` command is used for editing attributes, but the file manager can be used as well by following these steps:

1. Select the file that you want to modify in the right-hand pane, and click the **Attrs** button on the toolbar. This will bring up a window listing existing attributes, unless the filesystem on which the file is located does not support them.
2. To create a new attribute, click the **Add Attribute** button at the bottom of the window. This will open another window for entering its name and value that can contain several lines.
3. Click the **Save** button in the new attribute window to add it to the list.
4. To edit any existing attribute, just double-click on it. This will bring up a window like the one used for creating a new attribute but with an additional **Delete** button.
5. When you are done creating and editing attributes for the file, click the **Save** button below the list. Only then will they actually be applied to the file on the server.

Attributes are generally used for storing metadata about files, such as a description, character set, or icon. See the manual page for the `attr` command for more information on what attributes can be used for.

26.10 Editing File ACLs

Standard UNIX file permissions and ownership are a simple way of controlling who can access a file, but are not very flexible. A superior alternative that is available on many operating systems is POSIX ACLs. POSIX is a set of standards that applies to many UNIX systems, and ACL stands for Access Control List. By setting up an ACL for a file, you can grant permissions to additional users or groups in addition to the normal owner and group. When editing the ACL for a directory, defaults for newly created files in that directory can be set as well.

The `xfs` filesystem type on Irix and Linux includes ACL support, as do `ufs` filesystems on Solaris. If you have the right kernel patches installed, `ext2` and `ext3` filesystems on Linux can support ACLs as well. Fortunately, they are implemented in an almost identical way on all operating systems, so the user interface in Webmin for editing them is the same.

An access control list contains at least four entries, each of which grants some permissions to a user or group. The permissions granted by each entry are the same as those set by the `chmod` command—read, write, and execute/list. The default ACL for a file contains entries for its owner user, owner group, and other UNIX users. These are exactly the same as the permissions granted to user, group, and others by `chmod` and the **Info** window in the file manager.

One special entry that appears in all ACLs is the mask, which defines the maximum permissions that can be granted to the group owner and to any other users (except the file's owner). Because the mask limits the permissions that can be granted by other entries, you will often need to change it to achieve the desired effect from your ACL. Exactly one mask entry must exist in every ACL.

The most commonly used ACL entry is one that grants permissions to a UNIX user other than the owner. Similarly, entries that grant permissions to another group can also be defined. There is no limit on the number of such entries that can be created.

The ACL for a directory can include several special default entries that determine the initial ACL of any file created in the directory. Default user, group, and mask entries can be created, and the default user and group can apply to either a specific user or the owner of the file. On most operating systems, if you create any defaults you must create entries for at least the default user owner, default group owner, and default mask.

At the shell prompt, the commands `getfacl` and `setfacl` are used on Linux and Solaris to view and change ACLs, respectively. On Irix, the `ls -D` command is used to display ACLs and the `chac1` command is used to set them. Webmin will call these commands on the server whenever the file manager is used to view or change the ACL of a file.

To edit the ACL for a file or directory, follow these steps:

1. Select the file from the list in the file manager's right-hand pane, and click the **ACL** button on the toolbar. This will bring up a window listing all existing ACL entries, as shown in Figure 26.3.
2. To add a new entry, select its type from the menu next to the **Add ACL of type** button before clicking it. This will bring up another window for entering the user or group to which the entry applies, and the permissions that they are granted. An ACL can only

have one mask or default mask entry, so if either is chosen when one already exists, an error message will be displayed.

3. For user or group ACL entries, you must fill in the **Apply to** field with the name of the user or group to which the permissions are being granted. For default user or default group entries, the **Apply to** field can be set to the **File owner** option, or the name of a user or group can be entered. In the former case, the permissions will apply to the owner or group of any new file created in the directory. In the latter, they will be granted to the entered user or group. For mask ACL entries, there is no field for choosing to whom they apply.
4. In the **Permissions** field, check those permissions that you want granted to the user or group. These have the same meaning as those set by the `chmod` command in the window described in Section 26.5 “Editing File Permissions”.
5. Click the **Save** button to have the new ACL entry added to the list in the ACL window. It will not, however, be saved to the server yet.
6. To edit an existing ACL entry, just double-click on its row in the list. You can change the user or group to which it applies (if any) and the permissions, but not the type. Click the **Save** button to keep your changes or the **Delete** button to remove the entry from the list. Not all types of ACL entry can be deleted—only those that grant permissions to a specific user or group or the various default types for a directory.
7. Finally, click the **Save** button at the bottom of the ACL window to have the ACL applied to the file on the server. Because not all combinations of entries are valid on all operating systems, an error message may be displayed if your ACL is incorrect in some way. If this happens, either fix the problem or use the **Cancel** button to discard your changes.

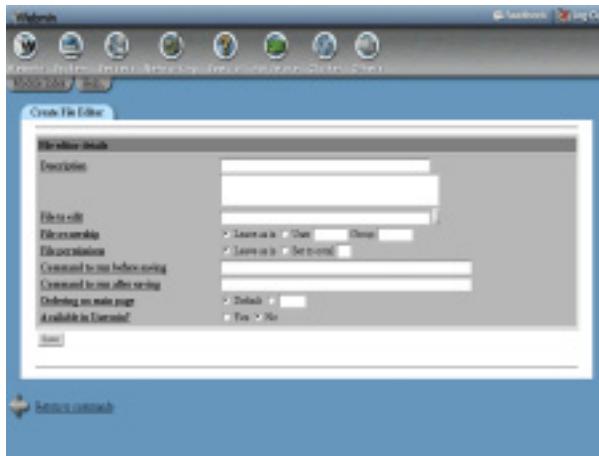


Figure 26.3 The ACL window.

26.11 Sharing Directories

If you have Samba installed on your system (covered in Chapter 43), it is possible to use the file manager to share directories to Windows clients. In addition, if you are running Linux or Solaris, the file manager can be used to export directories via NFS (as explained in Chapter 6). When sharing directories, the file manager has very few options compared to the modules designed

specifically for the purposes of configuring Samba and NFS. It does, however, provide a much simpler user interface.

Assuming the Samba is installed and working on your system, the following steps should be used to share a directory with Windows clients:

1. Select the directory that you want to share in the right-hand pane and click the **Sharing** button on the toolbar. This will bring up a window with two tabs, labeled **Windows** and **NFS**.
2. Under the first tab, turn on the **Windows file sharing enabled** option.
3. Enter a short description for this directory into the **Comment** field.
4. Unless you want the share to be temporarily disabled, make sure the **Currently active?** field is set to **Yes**.
5. To stop clients writing to the directory, change the **Writable** field to **No**. Otherwise, leave it set to **Yes**.
6. To allow clients to access this share without needing to log in, set the **Guest** option to **Yes**. If you set it to **Only**, clients will be treated as guests for the share even if they do login to the server. If you select **No**, clients will not be able to access it at all without logging in.
7. Click the **Save** button to make your new share active. On the server, an entry will automatically be added to the Samba configuration file. From now on, when the directory appears in the file manager, its icon will have the letter **S** on it to indicate that it is shared.

In the same way, directories that are already shared via Samba can be modified using the file manager. Any options that have been set in Webmin or manually will not be affected by editing the share in this module, even though only a few of them are visible under the **Windows** tab. To turn off the sharing of a directory to Windows clients, just select the **Windows file sharing disabled** option and hit **Save**. This will cause the entire share to be deleted from the Samba configuration, including all options.

If you are running Linux and the NFS server software is installed on your system, you can export a directory to UNIX clients by following these steps:

1. Select the directory that you want to share in the right-hand pane and click the **Sharing** button on the toolbar. In the window that appears, select the **NFS** tab.
2. Turn on the **NFS file sharing enabled** option.
3. The **NFS export options** section contains a table of hosts to which the directory is shared, and the options that apply to those hosts. When setting up sharing for the first time, only one empty row is available, so if you want to add multiple rows you must save the export and re-edit it.

In the field under the **Hosts** column, enter the hostname, IP address, or netgroup to which you want the directory to be exported.

From the menus under the **Options** column, you can control whether clients are allowed to write to the directory, and how client UNIX users are treated by the server. Chapter 6 explains the meanings of these menu options in more detail.

4. Click the **Save** button to have the export settings written back to the server and the NFS server automatically restarted. Allowed UNIX clients will be able to access the directory immediately.

5. To add another host to the directory, click the **Sharing** button on the toolbar again and repeat Steps 3 through 5.

On Solaris, the steps for sharing a directory via NFS are not quite the same due to the different options that are available on that operating system. Those steps are:

1. Select the directory that you want to share in the right-hand pane and click the **Sharing** button on the toolbar. In the window that appears, select the **NFS** tab.
2. Turn on the **NFS file sharing enabled** option.
3. Enter a short description for this export into the **Description** field, if you like.
4. To give some hosts read-only access to the directory, change the **Read-only hosts** field to **Listed** and enter their hostnames, IP addresses, or netgroups into the field below, separated by spaces. You can specify an entire network by preceding it with an @, such as @192.168.1.
To give all hosts read-only access, select the **All** option instead. This means that any system that can connect to yours over the network will be able to mount the directory and read the files that it contains.
5. To give hosts read-write access to the directory, change the **Read-write hosts** field to **Listed** and enter their hostnames, IP addresses, netgroups, or networks into the field below it.
If you select **All**, any system that can connect to yours will be able to read and write files in the directory, which is probably a bad idea from a security point of view.
6. By default, even those hosts that have read or write access will not be able to access files as the `root` user. To grant this to some hosts, change the **Root access hosts** field to **Listed** and enter their hostnames, addresses, netgroups, or networks into the field below. See Chapter 6 for more details on what `root` access means in relation to NFS.
7. Finally, click the **Save** button to have your new NFS export saved and made active.

On both Linux and Solaris, once a directory is shared via NFS its icon in the file manager's right-hand pane will be marked with the letter S. Directories that have been shared manually or by Webmin's NFS module will also be similarly indicated, and you can edit their settings by selecting them and hitting the **Sharing** button. Any NFS options that are not configurable in the file manager will be unaffected.

In the sharing window, you can turn off the NFS exporting of a directory by selecting the NFS file sharing disabled option and clicking the Save button. All entries in the NFS configuration file for the directory will be deleted, and the NFS server restarted to make the changes immediately active.

26.12 Module Access Control

Like other modules, the file manager can be configured in the Webmin Users module (covered in Chapter 52) to restrict the access that a user has to it. Specifically, you can limit a Webmin user to particular directories and allow him to access files with the rights of a non-root UNIX user. The directory limitation feature is particularly powerful, as a user can be given `root` access within that directory but prevented from seeing or touching any files outside of it.

Once you have created a Webmin user with access to the module, the steps for restricting his access to it are as follows:

1. In the Webmin Users module, click on **File Manager** next to the name of the user or group for which you want to edit access control restrictions.
2. To change the UNIX user that files are accessed as, enter a new name into the Access files on server as field. Alternatively, you can select the Same as Webmin login option, in which case the Webmin user will have the same privileges as the UNIX user with the same name.
Anyone who uses the module with non-root privileges will not be able to use its file sharing features, as this would open up a large security hole. Webmin users who do not have access to the Samba or NFS modules will also not be able to configure file sharing.
3. The **Umask for new files** field controls the permissions that are set on newly created files and directories. It contains an octal number which is the binary inverse of the number used in the chmod command to set permissions. For example, a umask of 022 would give new files 755 permissions, while a umask of 077 would give them permissions of 700.
4. To prevent the user from creating or editing symbolic links and to force all links to appear as the file that they are linked to, change the **Always follow symlink?** field to **Yes**. This should be done when restricting a user to a directory so he cannot create links to files outside of the directory and then edit or view them in the file manager.
5. To stop the Webmin user from editing or changing any files, set the **Read-only mode?** field to **Yes**.
6. To restrict him to only certain directories, enter them into the **Only allow access to directories** text box. By default, this field contains the root directory, which you must remove if the restrictions are to make any sense. When the user opens the file manager, it will appear as though directories other than those that have been allowed do not exist. The full path to each directory, however, will still be visible.
To automatically include the home directory of the UNIX user with the same name, check the **Include home directory of Webmin user** option. To have the file manager navigate to the first accessible directory automatically, leave the **Open first allowed directory?** option checked.
7. Finally, click the **Save** button to have the new restrictions activated.

If you want to give a large number of users access to the file manager, it may be better to install Usermin (covered in Chapter 47), instead. It includes an identical file manager that always runs as the UNIX user logged into Usermin, and can be restricted to the user's home directory.

26.13 Summary

Even though Webmin's file manager should be relatively intuitive if you have ever used a similar program on Linux or other operating systems, it does have some extra features that you may not have seen before. After reading this chapter you should be able to navigate your system's directories, and perform basic operations like editing, copying, and deleting files. If necessary and supported by your filesystem, you will also be able to share files with NFS or Samba, and edit the ACLs and extended attributes of files.

Perl Modules

This chapter explains how to install new Perl modules onto your system using Webmin, and how to view or delete modules that are already installed.

27.1 Introduction to Perl Modules

The Perl programming language has many of its functions in separate modules, which are loaded only when they need them by Perl scripts. The standard distribution of Perl includes many modules, but there are far more available that can be installed separately. Modules exist for a wide variety of purposes, such as connecting to databases, creating images, using network protocols, and parsing data formats.

All Perl modules have short names like `GD` or `Net::Telnet`. All those that have multi-part names separated by double-colons are part of a family of related modules, which are often packaged together. Modules are distributed in `.tar.gz` files which need to be extracted and compiled before they can be installed. Often, a single distribution file will contain multiple modules that must all be installed together.

The best source of Perl modules is CPAN (the Comprehensive Perl Archive Network), located at www.cpan.org/. It has a vast database of almost every third-party module available, and is easily searchable. Webmin can install a Perl module for you directly from CPAN if you know the name of the module that you want.

Because Webmin is itself written in Perl, it can make use of some optional modules. For example, to run Webmin in SSL mode (as explained in Chapter 2), it is necessary to install the `Net::SSLeay` module. To reliably connect to and manage MySQL and PostgreSQL databases, you need to install the `DBD::mysql` and `DBD::Pg` modules, respectively.

27.2 Perl Modules in Webmin

Under the Others category in Webmin is a module called Perl Modules that can be used to view, install, and remove Perl modules from your system. When you enter it, the main page lists all the modules that are currently installed, as shown in Figure 27.1. For each module, the name, a short description, installation date, and number of submodules is shown. Submodules are Webmin's term for Perl modules that are included in the distribution tar.gz file along with a primary module.

Because Perl behaves the same on all versions of UNIX, this Webmin module has the same user interface and functionality on all operating systems. The only problem that you may encounter on non-Linux systems is the lack of a C compiler, which is often needed when installing Perl modules. All versions of Linux include the `gcc` compiler as standard, but many commercial UNIX variants do not come with a free C compiler.

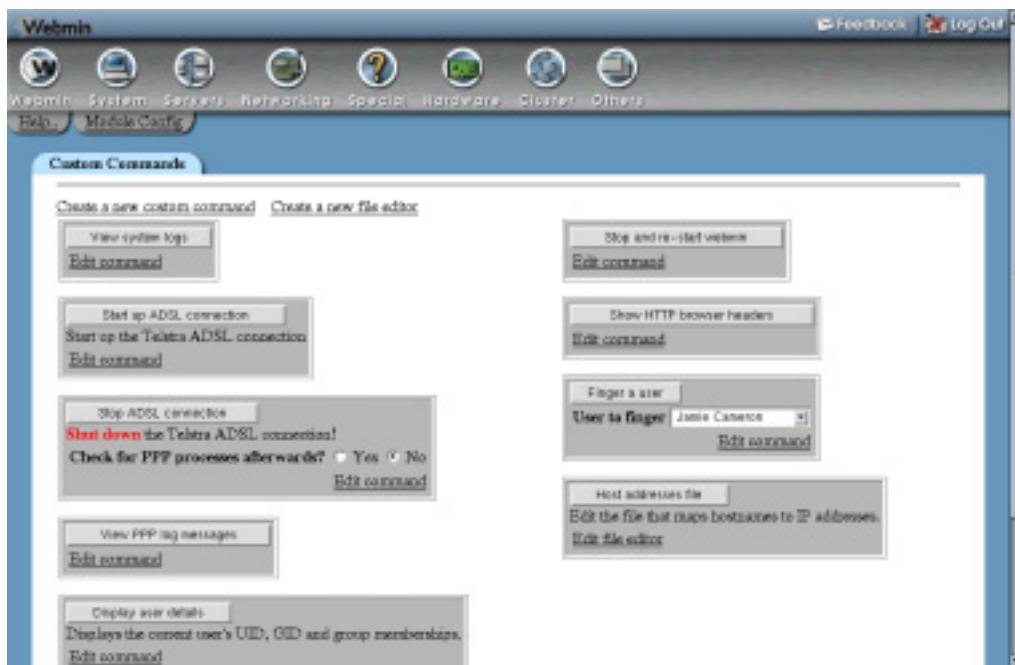


Figure 27.1 The Perl Modules main page.

27.3 Installing a Perl Module

If you need to install a new Perl module for use by Webmin or for developing your own scripts, it can easily be done using this Webmin module. The steps to install a Perl module are:

1. At the bottom of the module's main page is the installation form. It offers four options for types of source from which to install a module in tar.gz distribution file format, but the most common and useful is **From CPAN**. Just select it and enter the name of the module (such as `Net::Telnet`) into the adjacent text field.

If the module file is already on your system, you can choose the **From local file** option and enter the path to the `.tar.gz` file into the field next to it. Or if you have the file on the system on which you are running your browser, click on **From uploaded file** and select the file using the **Browse** button.

The final source from which a module can be installed is a URL on another server. To have Webmin download it for you, select the **From ftp or http URL** option and enter the URL into the field next to it.

2. If the **From CPAN** option was chosen and this not the first module that you have installed from that source, the **Refresh module list from CPAN** checkbox next to the module name field will be visible. If checked, Webmin will again download the complete list of modules and the URLs on which they can be found from the CPAN website. Otherwise, it will use a local cache of the list from the previous download.

The module list should be downloaded periodically to ensure that the local copy remains up-to-date. For this reason, the box will be checked by default every 30 days, or whatever period you have set in the Webmin module's configuration.

3. When you have selected the source, click the **Install** button. This will take you to a page showing the progress of the downloaded CPAN module list and the module file itself, if necessary. If the Perl module cannot be found on CPAN or the select `.tar.gz` file does not appear to be in the correct format, an appropriate error message will be displayed.

If the module file was downloaded and successfully verified, however, an installation options form like the one in Figure 27.2 will be displayed.

4. The **Install action** field determines which steps of the module installation process will be carried out by Webmin. The available options are:

Make only The file will be extracted, its `Makefile` generated with the command `perl Makefile.PL`, and then the `make` command run to build the modules it contains. No actual installation will take place.

Make and test Like the **Make only** mode, but compiled module will be tested with the `make test` command as well. Almost all Perl modules include test code to verify that they have been compiled properly.

Make and install The module file will be extracted, the modules it contains built, and then the `make install` command will be run to copy the compiled files to the appropriate Perl directories on your system. Once they have been installed, the modules will be usable by other Perl scripts and programs (like Webmin).

Make, test, and install Like the **Make and install** mode, but the `make test` command will be run on the compiled modules before they are installed to verify that they were built correctly. This is the default mode, but for some modules it may not be appropriate if the testing phase is prone to failing incorrectly.

5. For some Perl modules, additional parameters may need to be passed to the `perl Makefile.PL` command for them to be built correctly. If so, you can enter them into the **Makefile.PL arguments** field. The `Net::SSLeay` module, for example, requires the path to the OpenSSL directory to be given as a parameter, if it has not already been installed in the standard directory. Generally though, you will not need to fill in this field.

6. Some Perl modules need certain environment variables to be set before `perl Makefile.PL` is run. If that is the case with the module you are trying to install, fill in the **Makefile.PL environment variables** table with the names and values of those that need to be set. The average module does not require any special variables.
7. To have Webmin carry out the compile and installation steps chosen in Step 4, click the **Continue with install** button at the bottom of the form. This will take you to page showing each command run to build the module and any output or error messages that it produces. Only if everything is successful will a message like **Make, test, and install of Net::SSLeay successful** appear at the bottom of the page.
If something goes wrong, check the error messages for clues. Many Perl modules provide an interface to some C library and require that the included files for that library be installed. On many Linux distributions, these are in a different package to the library itself. For example, `Net::SSLeay` uses the OpenSSL C library, whose included files are often in a separate `openssl-devel` package. See Chapter 12 “Software Packages” for instructions on how to install packages on your system.
8. Assuming everything works and you choose to install the module, you can now return to the main page. The new module should be listed there and will be usable in Perl scripts and programs.

Some Linux distributions include various Perl modules in RPM format. They must be installed using the Software Packages module, not this one. Be warned that if you have upgraded Perl from the version included with your distribution, these RPMs will not work. For this reason, it is almost always better to install Perl modules using this Webmin module.

27.4 Viewing and Removing a Perl Module

The main page of this Webmin module displays all non-core Perl modules installed on your system for which a `.packlist` file can be found. Unfortunately, some modules do not create a `.packlist` file, especially those installed from an RPM package. Modules like this will still be usable in Perl scripts, but cannot be viewed or uninstalled by Webmin.

Most Perl modules include documentation on their API for programmers who want to make use of them in scripts. To view a module’s documentation, follow these steps:

1. On the main page, click on the module name under the table’s **Module** column. This will bring you to a page showing its complete documentation, as generated by the `perldoc` command. Not all modules have documentation, so in some cases none will be displayed.
2. If the module has submodules, they will be listed as well. Each may have additional documentation that you can view by clicking on its name.

Webmin can also be used to delete Perl modules from your system, as long as they have properly formatted `.packlist` files. The process should be used to remove a module:

1. On the main page, click on the module’s name to go to the documentation page.
2. If the **Uninstall module and submodules** button exists, click on it. If the button is not displayed, then Webmin cannot remove this Perl module.

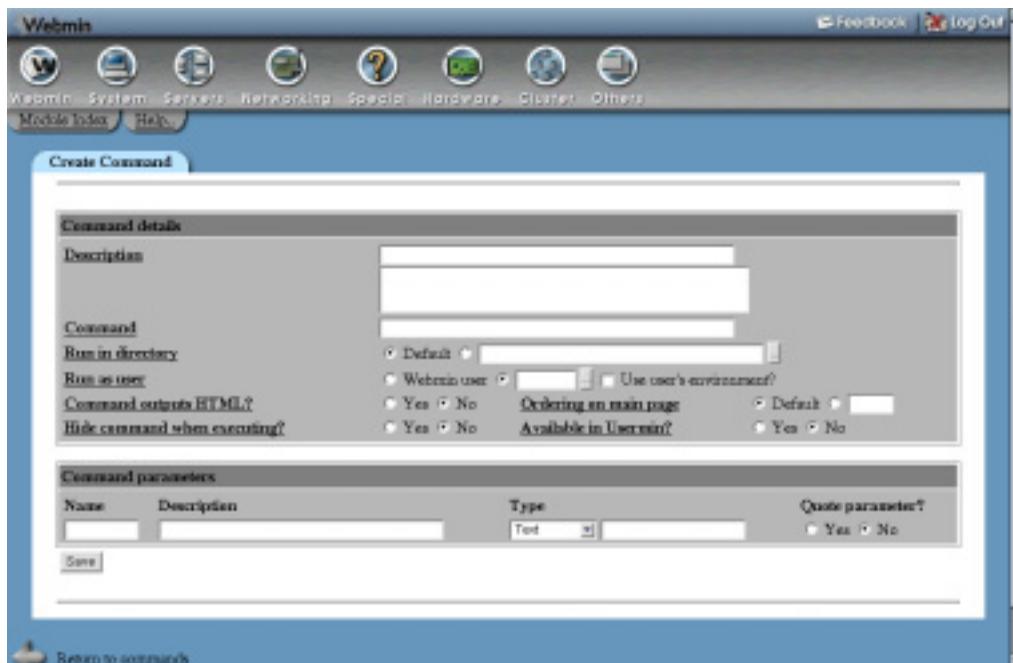


Figure 27.2 The module install options form.

3. Once you click on the button, a page listing all the files to be deleted is displayed. To go ahead with the uninstall, click the **Uninstall now** button at the bottom of the confirmation page. All the module's files will be removed, and you will be returned to the main page.

As mentioned in Section 27.3 “Installing a Perl Module”, some Perl modules are installed from RPM packages. To remove one of these, use the deletion feature of the Software Packages module instead.

27.5 Configuring the Perl Modules Module

This Webmin module has one configurable option that you might want to change and two others that should only be modified if using a different repository for Perl modules than the normal CPAN website. All of the options listed in Table 27.1 can be found by clicking on the **Module Config** link in the top-left corner of the main page.

27.6 Summary

This chapter has explained what Perl modules are and how to use Webmin to install new modules into your system. It has also covered the viewing of documentation and other information for existing modules and explained how to remove those that you no longer need.

Table 27.1 Module Configuration Options

Days before refreshing CPAN module list	This field determines the number of days that Webmin will wait before recommending that the CPAN module list be downloaded again, as explained in Section 27.3 “Installing a Perl Module”. It is a good idea to refresh occasionally as the URLs it contains may become out-of-date when a new version of a Perl module is released.
CPAN perl modules list	These fields determine where Webmin downloads the list of CPAN modules from, and where it downloads actual module files from. The defaults will work perfectly well, but because there are many CPAN mirror sites around the world you may want to change them to use a site closer to you. If so, the CPAN perl modules list field must be set to the URL of the <code>02packages.details.gz</code> file on the mirror server.
CPAN modules base URL	The CPAN modules base URL field must contain the URL of a directory under which module files are categorized by author, which will typically end with <code>authors/id</code> .

Status Monitoring with Webmin

This chapter covers the use of Webmin’s System and Server Status module, which can be used to check for and report down systems, failed servers, network outages, and other problems.

28.1 The System and Server Status Module

This module allows you to monitor the status of various servers and daemons running on your system, so you can easily see which ones are running properly and which are down. It can also be configured to check the status of servers on a regular schedule, and to email you or run a command if something goes down. This can be useful if your system runs critical servers that other people depend upon, such as web or DNS servers.

The module can also monitor servers running on other hosts. This can be done in two ways—by making a TCP or HTTP connection to the port on which the server runs or by communicating with the Webmin server on the remote host and asking it to check the status of the server. The latter method is more powerful because it can be used to monitor things such as disk space and daemons that do not accept any network connections.

Each server or service that you want to watch, using the module, must have a *monitor* defined. Every monitor has a type that indicates what kind of server it is supposed to check, such as Apache or BIND. Monitors also have additional parameters, some of which are specific to their type. The module allows you to create many different types of monitors, for things like checking to see if Sendmail or Squid is running, watching for excessive network traffic or a shortage of disk space, or pinging or connecting to some host.

A monitor can run either on the system on which you are using the module or another server running Webmin. In the latter case, the server must be defined in the Webmin Servers Index module, explained in Chapter 53. You can also check another system that does not have Webmin installed using the remote TCP, HTTP, and ping monitor types.

Many monitors use other Webmin modules to find the locations of the servers and daemons that they check. For this reason, those other modules must be configured and working properly for the associated monitor to work as well. For example, if you have compiled and installed Apache in a different directory from the standard for your Linux distribution, the module configuration for Apache Web server will have to be adjusted to use the correct paths. If not, this module will not know where to look for the Apache PID file.

When you enter the System and Server Status module from the Others category on the Webmin menu, its main page will display a table of all configured monitors. Several monitors for common servers and services will be defined by default, but you can edit, delete, or add to them as you wish. Figure 28.1 shows an example of the module's main page.



Figure 28.1 The System and Server Status module.

For each monitor, a description, the Webmin server that it runs on, and its current status are shown. A monitor can be in one of the following three states:

Up Means that the monitored server or service is running correctly. This state is indicated by a green tick on the main page.

Down Means that the monitored server is down. This state is indicated by a red "X" on the module's main page.

Webmin down Means that Webmin on the remote system is down, and so the monitor cannot be run. Indicated by a red "W" on the module's main page.

Timed out Means that the monitor did not return a result within 60 seconds, perhaps because it ran a command that never completed. Indicated by a red clock symbol on the main page.

Not installed Means that the server being monitored is not installed on your system. This state is indicated by a black circle with a line through it.

By default, the status of every monitor is queried every time you view the module's main page. Because this may take a long time, if you have many monitors or are checking the status of servers on remote hosts, there is a module configuration option that can be used to display the status from the last scheduled check instead.

28.2 Adding a New Monitor

To have Webmin check on the status of a new server or service, you must add an additional monitor in this module. Before you can do this, you must decide on the monitor's type, which is determined by the type of service that you want it to check. See Section 28.3 "Monitor Types" for a list of all those that are available, their purposes, and optional parameters.

Once you have chosen a type, you can add the monitor by following these steps:

1. Select the type from the menu next to the **Add monitor of type** button on the module's main page. When you click the button, the browser will display a form for adding a new monitor, as shown in Figure 28.2.

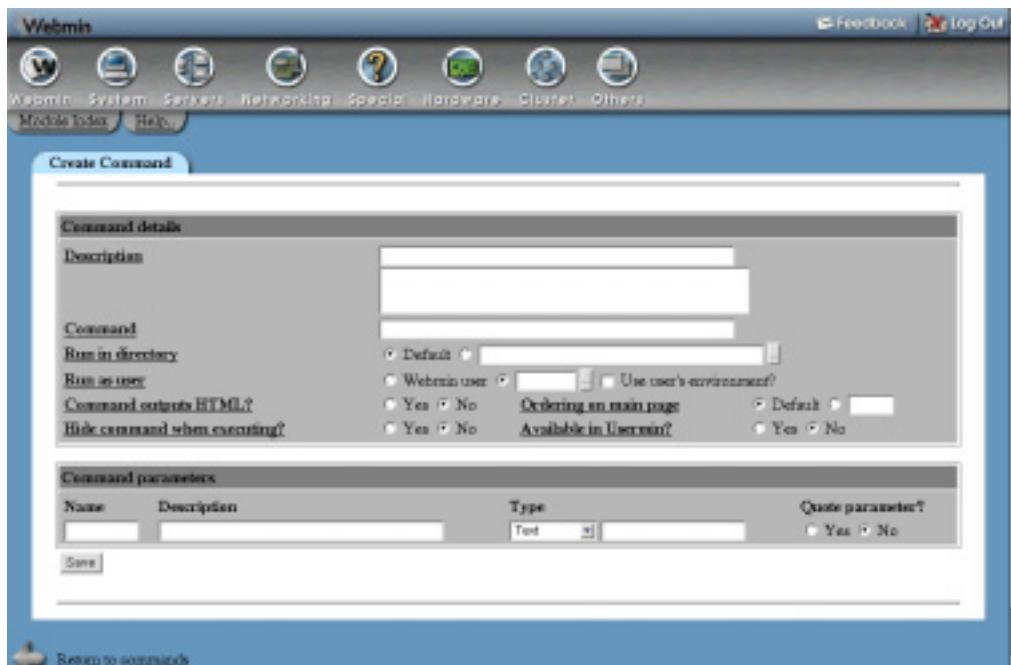


Figure 28.2 Adding a new HTTP monitor.

2. Fill in the **Description** field with a short description of this monitor, such as **Office web server**. This will appear on the main page and in any status emails.
3. To have the monitor executed on another Webmin server, select it from the **Run on host** menu. If you have no servers defined in the Webmin Servers module (covered in Chapter 53), no menu will appear.
4. If you have scheduled monitoring enabled and want this service to be regularly checked by it, make sure the **Check on schedule?** field is set to **Yes**. If it is set to **No**, scheduled checking will be turned off for this particular monitor.
The other options starting with **Yes** allow you to control when email is sent if the monitor goes up or down. They correspond to the options for the **Send email when** field, explained in Section 28.4 “Setting Up Scheduled Monitoring”.
5. To have a command executed when a scheduled check determines that the monitor has gone down, enter it into the **If monitor goes down, run command** field. This could be used to attempt to restart the monitored server, or to notify a system administrator by some method other than email.
6. You can also fill in the **In monitor comes up, run command** field with shell commands to execute when a scheduled check determines that the service has come back up again.
7. If the **Run on host** field is set to another Webmin server, you can choose whether the up and down commands in the previous two steps are run on this system or the remote server. This is controlled by the **Run commands on** field.
8. If the monitor is being run locally and is checking a server configured in another Webmin module for which multiple clones exist, the **Module to monitor** field will appear on the form. This menu can be used to choose from which of the clones the monitor should get its configuration. So, for example, if you had two versions of Apache installed on your system and two Apache Webserver modules set up to configure them, you would be able to choose which one should be checked when creating an Apache Webserver monitor.
See Chapter 51 for more information on how module clones work.
9. Depending on the type of monitor being created, there may be several additional options that you can set on this form. See Section 28.3 “Monitor Types” for the details.
10. When done, click the **Create** button to have the monitor created and added to the main page. Its status should be immediately displayed.

Existing monitors can be edited by clicking on their description on the main page. When editing, all the same fields as described above are available, in addition to a **Current status** field that indicates whether the service is up or down. For some monitor types, additional information is displayed when it is up, such as the time that the server being checked was started.

After you have finished editing a monitor, click the **Save** button at the bottom of the page to record your changes. To get rid of a monitor, use the **Delete** button instead. Either way, the changes will be applied immediately.

28.3 Monitor Types

The System and Server Status allows you to monitor many different kinds of servers and daemons, using different monitor types. All types perform some kind of check, and either succeed or fail depending on whether the check passes or not. In some cases, a monitor can return a third

result indicating that the server being checked is not installed or that the check that it is trying to perform is impossible.

The available types, their purposes, and additional parameters are listed in Table 28.1:

Table 28.1 Monitor Types and Their Options

Monitor	Description	Parameters
Apache Web server	Determines if Apache is running by looking at its PID file, using the configuration set in the Apache Webserver module.	None
BIND 4 DNS Server	Checks if BIND version 4 is running by looking at its PID file, using the path set in the BIND 4 DNS Server module.	None
BIND DNS Server	Checks if BIND version 8 or 9 is running by looking at its PID file, using the configuration from the BIND DNS Server module.	None
Check File	<p>Can be configured to make sure that some file exists, does not exist, or that its size is smaller or larger than a certain number of bytes.</p> <p>This monitor can be useful for detecting log files that have become too large, critical files that have gone missing, or indicator files created by other programs.</p>	<p>File to check The full path to the file whose existence or size should be checked by the monitor.</p> <p>Test to perform If File must exist is chosen, the monitor will fail if the file does not exist.</p> <p>If File must not exist is chosen, the monitor will fail if the file does exist.</p> <p>If File must be bigger than is selected, the monitor will fail if the file is smaller than or equal to the size entered next to this option.</p> <p>If File must be smaller than is selected, the monitor will fail if the file is bigger than or equal to the size entered.</p>

Table 28.1 Monitor Types and Their Options (Continued)

Monitor	Description	Parameters
Check Process	Checks if the process matching some pattern is running or not. This can be useful for ensuring that servers and daemons are running, or for detecting suspicious processes.	<p>Command to check for A Perl regular expression to search the list of running processes for, such as <i>httpd</i> or <i>nfsd</i>.</p> <p>Fail if process is Determines if the monitor checks to make sure that the command is running, or if it does the opposite and makes sure that it is not running.</p>
Configuration Engine Daemon	Checks to see if the CFengine daemon is running.	None
DHCP Server	Checks to see if the ISC DHCP server is running by looking at its PID file, taken from the configuration of the DHCP Server module.	None
Disk Space	Makes sure that a filesystem has at least a certain amount of free disk space. A monitor of this type can be used to give you an early warning of an impending shortage of disk space.	<p>Filesystem to check For this parameter, you must select the filesystem whose size should be checked from its menu. Or you can choose the Other option and enter the mount point into the text field next to it.</p> <p>Minimum free space (in kB) If the amount of disk space free on the filesystem is less than the number of kilobytes entered for this parameter, the monitor will fail.</p>
Execute Command	Executes an arbitrary shell command and checks its exit status. This is the most flexible monitor, as you can use it to run your own custom scripts that perform checks that none of the built-in monitor types can.	<p>Command to check exit status of A shell command or commands that the monitor will run as <i>root</i> and check to see if it succeeds or fails, based on its exit status. All the usual shell metacharacters such as <i>;</i>, <i>&&</i>, and <i> </i> can be used.</p>
Extended Internet Server	Checks to see if the <i>xinetd</i> Extended Internet Server is running by looking at its PID file.	None

Table 28.1 Monitor Types and Their Options (Continued)

Monitor	Description	Parameters
File Change	Determines if a file has changed since the last time the monitor was run. Useful for detecting changes to critical files or log files that record serious error messages.	File to monitor The full path to a file or directory whose last modification time will be checked. The monitor will report a failure if the time has changed since the monitor was last queried.
Free Memory	Makes sure that the amount of free memory available on your system is not less than a certain amount.	Minimum free memory (in kB) If the amount of available memory is less than the number of kilobytes entered for this parameter, the monitor will fail. On Linux, free memory is defined as the sum of free RAM, free swap, and the memory used for buffers and caches.
Hostsentry Daemon	Checks to see if the Hostsentry daemon is running on your system.	None
Internet and RPC Server	Checks to see if the <code>inetd</code> Internet Server is running by looking at its PID file.	None
Jabber IM Server	Checks if the Jabber Instant Messaging Server is running by looking at its PID file.	None
Load Average	Monitors the system load average, and checks to see if it has exceeded some limit.	Load average to check As explained in Chapter 11, all UNIX systems keep track of the average system load over the last 1, 5, and 15 minutes. This parameter controls which average the monitor will check. Maximum load average The average above which the monitor will fail. On a single-CPU machine, an average of 1.0 means that the system is fully loaded.

Table 28.1 Monitor Types and Their Options (Continued)

Monitor	Description	Parameters
MON Service Monitor	Checks to see if MON is running by looking at its PID file, which is taken from the configuration of the MON Service Monitor module.	None
MySQL Database Server	Checks to see if MySQL is running by attempting a test connection, using the username and password set in the MySQL Database Server module.	None
NFS Server	Checks to see if the <code>nfsd</code> process exists on your system, indicating that the NFS server is running.	None
Network Traffic	<p>This monitor type checks the number of bytes that have passed through a network interface, and fails if the data rate exceeds some limit. Because the rate is just the number of bytes that have passed since the last check divided by the number of seconds since the check, a monitor of this type must be run every few minutes on schedule.</p> <p>This can be useful for warning you of a denial of service attack launched from or at your system.</p>	Interface to monitor The network interface that will be checked for excessive traffic. Maximum bytes/second The data rate above which the monitor will report a failure. Direction to monitor This parameter determines whether incoming, outgoing or traffic on both directions is counted towards the data rate.
Portsentry Daemon	Checks to see if Postsentry is running by looking at its PID file, which is taken from the configuration of the Security Sentries module.	None
Postfix Server	Checks to see if Postfix is running by looking at its PID file, the location of which is taken from the Postfix Configuration module.	None

Table 28.1 Monitor Types and Their Options (Continued)

Monitor	Description	Parameters
PostgreSQL Database Server	Checks to see if PostgreSQL is running by attempting a test connection, using the username and password set in the PostgreSQL Database Server module.	None
ProFTPD Server	Checks the ProFTPD PID file to see if the server process is running. This monitor can only be used if the FTP server is running in stand-alone mode, not from a super server like <code>inetd</code> or <code>xinetd</code> .	None
QMail Server	Checks to see if Qmail is running by looking for the <code>qmail-send</code> process.	None
Remote HTTP Service	Attempts a connection to an HTTP server running on some host, and requests a page. If something goes wrong, the monitor will fail. This type is useful for testing web servers running on systems that do not have Webmin installed, or for checking to see that critical pages are available.	<p>URL to request An HTTP or HTTPS URL for the monitor to download when it is run. FTP URLs are not supported.</p> <p>Connection timeout The number of seconds that the monitor will wait to make the connection and request a page. If the process takes longer than this time, a failure will be reported. The default is 10 seconds.</p> <p>Login as If the URL is password protected, this parameter can be used to specify a username and password with which to log in.</p>
Remote Ping	Sends and listens for ICMP packets to determine if some host is up or down. This can be useful for testing network connectivity and server availability.	<p>Host to ping The IP address or hostname of a system to check. If the host fails to respond to ICMP echo-request packets, the monitor will fail.</p> <p>Time to wait for response The number of seconds that the monitor should wait for an ICMP echo-reply response.</p>

Table 28.1 Monitor Types and Their Options (Continued)

Monitor	Description	Parameters
Remote TCP Service	Attempts a TCP connection to a host and port to ensure that the server listening on that port is running. If the connection is successful, it will be closed immediately.	<p>Host to connect to The IP address or hostname of the system to which the monitor should open a TCP connection.</p> <p>Port to connect to The TCP port on the host on which the connection should be made.</p> <p>Connection timeout The number of seconds to wait for the TCP connection to succeed before the monitor gives up and reports a failure. The default is 10 seconds.</p>
Samba Servers	Checks to see if both the <code>smbd</code> and <code>nmbd</code> Samba server processes are running, unless they have been configured to run from <code>inetd</code> or <code>xinetd</code> .	None
Sendmail Server	Checks to see if the Sendmail server is running by looking at its PID file, which is taken from the configuration of the Sendmail Mail Server module.	None
Squid Proxy Server	Checks to see if the Squid server is running by looking at its PID file, which is taken from the configuration of the Squid Proxy Server module.	None
Usermin Web server	Checks to see if the Usermin HTTP server is running by looking at its PID file, the location of which is taken from the Usermin Configuration module.	None
Webmin Web server	Checks to see if Webmin itself is running. This monitor type is only really useful when run on schedule.	None

Not all monitors are available on all operating systems. Because they use Linux-specific files in `/proc`, the Free Memory and Network Traffic monitors are only available on that OS. The Load Average type can only be used on systems that support the Running Processes module, and the Disk Space monitor will only work on systems to which the Disk and Network Filesystems module has been ported.

In addition, many monitors depend upon other Webmin modules. For example, if the Apache Webserver module has been deleted from your Webmin installation, you will not be able to use the Apache Webserver monitor type. If you attempt to add a new monitor that depends upon a module that is not installed or will not work on your operating system, an error message will be displayed when the **Create** button is clicked.

28.4 Setting Up Scheduled Monitoring

The monitors that you can configure using this module are most useful when they are run on schedule, so that you can be automatically notified via email if a monitored server or daemon goes down. When scheduled checking is enabled, all your monitors will be run at a periodic interval, just as they are all run when you visit the module's main page.

To set up scheduled monitoring, the steps to follow are:

1. Click on the **Scheduled Monitoring** button found on the module's main page below the table of monitors. This will take you to the form shown in Figure 28.3.
2. Change the **Scheduled monitoring enabled?** field to **Yes**.
3. The **Check every** field controls when the scheduled check is run. The first lets you set the period, such as every 1 hour or 5 minutes, while the second part controls how many hours or minutes into the period it is run. For example, to have the monitors checked at 3:00 a.m. every day, you would set the **Check every** field to *1 days*, and the **with offset** field to *3*.
4. To limit the check to only certain hours of the day, deselect those hours on which you don't want it to run from the **Run monitor during hours** list. This does not make much sense if the scheduled check is being run only once per day.
5. Similarly, to limit the check to certain days of the week, deselect the days that you don't want it to run from the **Run monitor on days** list.
6. The **Send email when** field determines which events will cause an email message to be sent by the scheduled check.

If **When a service changes status** is chosen, email will be sent when a service goes down or up.

If **When a service goes down** is chosen, email will only be sent when a service goes down.

If **Any time service is down** is chosen, email will be sent as long as any service is down, and will be sent again at each check until service comes back up.

It is possible to override this field on a per-monitor basis using the **Check on schedule** field on the monitor creation form.

7. To receive email when a service goes down, enter your address into the **Email status report** to field. If it is left set to **Nobody**, then no email will be sent.
8. To set the source address of the status email, change the **From: address for email** field. The default is just `webmin@yourhostname`.



Figure 28.3 The scheduled monitoring configuration form.

9. By default, any status email will be sent by running the `sendmail` program on your system. To have it sent via an SMTP server on another system, change the **Send mail via** field to **SMTP server** and enter the hostname of the mail server into the field next to it.
10. If you want to receive an email for each monitor that goes down, change the **Send one email per service?** field to **Yes**. Otherwise, all services that are determined to have failed by a single check will be reported in a single email.
11. If you have a pager command set up and working on the module's configuration, you can enter a pager number into the **Page status report to number** field. It will receive a shortened version of the message that is sent via email.
12. Click the **Save** button at the bottom of the page to activate scheduled monitoring. Webmin will automatically set up a Cron job that runs a script on the chosen schedule.

Once scheduled monitoring is active, you should begin receiving email messages notifying you when services go down and come back up. If a service is down when scheduled checking is first enabled, however, and you have chosen to be only notified when services go down or come up, you will not receive a message about it.

To modify any of the scheduled monitoring options, just repeat the preceding steps again. To turn it off altogether, change the **Scheduled monitoring enabled?** field to **No** and click **Save**. If you want to change the monitoring schedule, it is best to do it in this module instead of in the Scheduled Cron Jobs module that is covered in Chapter 10.

28.5 Module Access Control

You can grant a Webmin user the right to only see the current status of configured monitors but not create or edit them. This can be done in the Webmin Users module, which is covered in Chapter 52. Once you have created a user who has access to the module, follow these steps to give him read-only access:

1. In the Webmin Users module, click on **System and Server Status** next to the name of the user or group that you want to restrict.
2. Change the **Can edit module configuration?** option to **No** to prevent him changing display options.
3. Set the **Can create and edit monitors?** field to **No** so he can only view the status of existing monitors.
4. Set the **Can change scheduled monitoring?** field to **No**.
5. Click the **Save** button to make the module access control restrictions active.

28.6 Configuring the System and Server Status Module

This module has several configuration options, mostly related to the way the main page is displayed. The options, which you can edit by clicking on the **Module Config** link on the main page, are listed in Table 28.2.

Table 28.2 Module Configuration Options

Status to display in list	By default, this field is set to Current status which will cause the main page to query and display the current status of all configured modules. If you change it to From last scheduled check, however , the status of each monitor at the time the last scheduled check was run will be displayed instead. This option will make the main page load much faster, especially if you have a large number of monitors. It doesn't make much sense, however, if you do not have scheduled monitoring enabled.
Command to send message to pager	When the scheduled check needs to send a message to a pager number, it will invoke this command with two parameters. The first is the number to which to send the message, and the second is the actual message text. The freely available <code>yaps</code> works well with these parameters.
Seconds between page refreshes	When Don't refresh is deselected and a number entered, the main page of the module will be periodically reloaded by any browser viewing it. The time between refreshes is the number of seconds entered.

Table 28.2 Module Configuration Options (Continued)

Display monitors sorted by	This field controls the ordering of monitors on the module's main page. The Order created option will show them in the order that they were added. The Description option will sort them by their description. The Host option will sort them by the remote Webmin server on which each runs.
-----------------------------------	---

28.7 Summary

By the time you have finished this chapter, you should understand how Webmin's System and Server Status module can be used to easily monitor various servers and services on one or more systems. You should know how to add and edit monitors, how to set up scheduled checking, and how to configure outage notification via email, pager, or SMS. If you are looking for a more advanced monitoring tool, try MON, which can be configured using the MON Service Monitor module.

Apache Web Server Configuration

This chapter explains how to use Webmin to configure the Apache Web server. It covers virtual hosts, IP access control, password restrictions, and much more.

29.1 Introduction to Apache

Apache is the Internet's most popular HTTP server, due to its zero cost, wide availability, and large feature set. All Linux distributions include it as a standard package, and it can be installed on or compiled for every other UNIX variant supported by Webmin. It has a very large number of option directives defined in a text configuration file, however, and so can be hard for an inexperienced administrator to set up.

Over the years since it was first introduced, many versions of Apache have been released. Starting with 1.0 and moving through to the current 1.3 and 2.0 series, each version has included more features and options. The basic web server functionality and configuration file layout has remained the same throughout, even though the internal implementation has changed significantly.

Apache has a modular design, in which each module is responsible for some part of its overall feature set. There are several standard modules that are included with almost every install of Apache, and many more that are optional or have to be downloaded separately. Modules can be compiled into the web server executable, or dynamically loaded from shared libraries at runtime. This modular architecture can be used to save memory by avoiding the need to load modules that do not provide any useful functionality for a particular system.

Apache takes its configuration from multiple text files, each of which contains a series of directives, usually one per line. Each directive has a name and one or more values and sets an option such as the path to a log file or the MIME type for some file. The directives that Apache recognizes are dependant on the modules in use. Most modules add support for several directives to configure the functions that they provide.

Often, you will want to host more than one website on a single server. Apache can be configured to use a different configuration depending on the website that is requested by a browser. Each one of these sites is called a virtual host and is defined in the configuration file with a special `<virtualhost>` section. All directives inside this virtual host section apply only to requests that match their IP address or hostname.

Similarly, `<Directory>` and `<Files>` sections can be defined in the configuration file to contain directives that apply to only a certain directory or to files matching a particular pattern. These are often used to deny access to certain files on your system, to password protect them, or to control the way that they are displayed to clients.

Another method of creating directives that apply to only a single directory is to put them in a special configuration file named `.htaccess` that resides in the directory itself. Often these files will be created by regular users so they can configure their websites without needing full access to the master configuration file. This is very useful on a system that hosts multiple sites that are each owned by a different UNIX user, rather than on a system with only one website that is set up by the server's owner.

29.2 The Apache Webserver Module

This is one of the most complex and powerful Webmin modules, as it allows you to configure almost every feature of Apache. It can determine the version of Apache that is installed on your system and the modules that it uses, and adjusts its user interface accordingly so that you can edit only those directives that the web server understands. The interface, however, is generally the same for all versions of Apache.

Because there are so many directives and the module attempts to allow configuration of all of them, it groups directives into categories like Processes and Limits, Networking and Addresses, and CGI Programs. These categories are represented by icons that will appear when you open a virtual server, directory or options file in the module. In all cases, you can view and edit the settings under each category by clicking on its icon.

Apache has a large number of standard modules and an even larger number of separate modules that were developed by other people. Webmin does not support the editing of directives in most of these non-standard modules, such as `mod_perl` and `mod_php`. It will safely ignore any configuration file directive that it does not understand, however, so any settings for unsupported modules that you make manually will not be harmed.

The Apache Webserver module can be found under the Servers category on the Webmin main menu. When you enter it for the first time the main page will display a list of all Apache modules that it knows how to configure, with those available on your system selected. Figure 29.1 shows an example of this.

In almost every case, the default selections will be correct for your system and you can just click the **Configure** button to begin using the module. If, however, you have a complex Apache configuration file that Webmin cannot parse properly to find dynamically loaded modules, the default selections may be incorrect. If so, you will need to change them so that the module does not attempt to set directives that are not supported on your system.

Once you have submitted the module configuration form, the main page will be redisplayed as shown in Figure 29.2. From this point on, this page will be displayed immediately whenever you

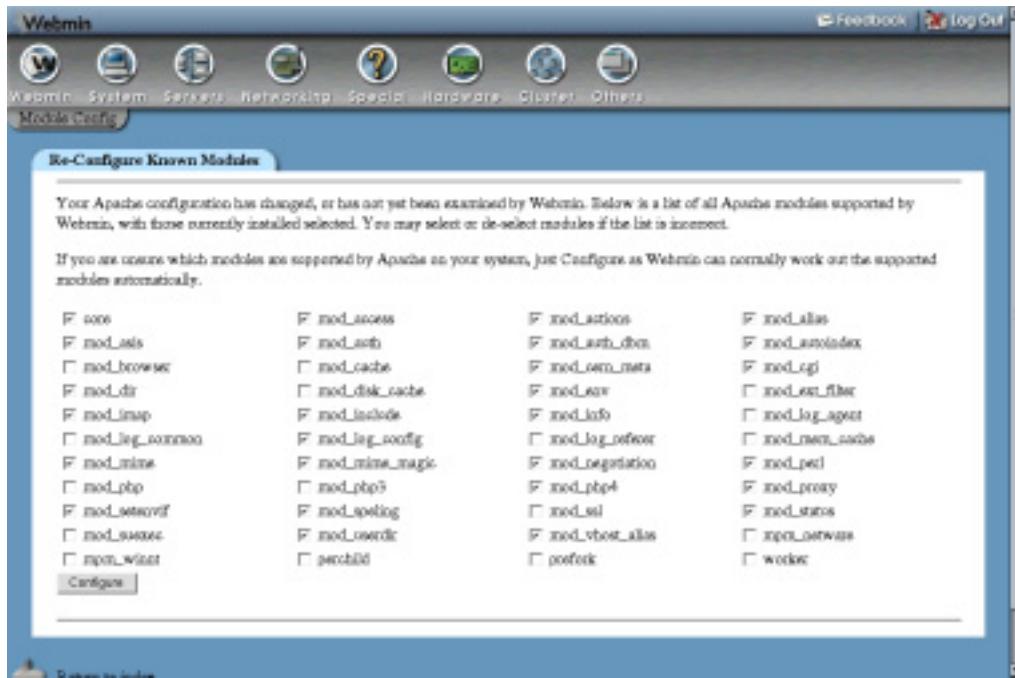


Figure 29.1 Selecting available Apache modules.

enter the module, unless Webmin detects that a new version of Apache has been installed on your system.

At the top of the main page are icons for the various categories of global options, as well as a few extra features. Below them is a list of all current virtual servers, followed by a form for adding a new virtual host. If you have a very large number of virtual servers on your system (more than 100 by default) a search form for finding servers will be displayed instead. The first server will always be the special **Default Server**, which contains directives that apply to all other virtual servers and handles requests that other servers do not.

Naturally, the Apache module will not work if you do not have Apache installed on your system. If this is the case, the main page will display an error message instead of the module configuration form or list of virtual servers. All Linux distributions include a package or packages for it on their CD-ROM or website, so install it from there using the Software Packages module (covered in Chapter 12) before continuing.

Because the module assumes that the Apache executable and configuration files will be in the locations used by your distribution's package, it will report the same error about the software not being installed if you have compiled and installed it manually. If this is the case, click on the **Module Config** link and adjust the paths to the correct locations for your system. The instructions in Section 29.22 “Configuring the Apache Webserver Module” explain how to do this in more detail.

On versions of UNIX that do not include Apache by default, Webmin assumes that it will be installed from the standard source distribution at www.apache.org. If you have installed the web



Figure 29.2 The Apache Webserver module main page.

server from an optional package that has been made available for your OS, then the main page will complain that it is not installed and you will need to adjust the module's configuration.

The module's user interface is quite complex and has a large number of pages, forms, and sub-pages due to the complexity and power of the Apache configuration files. There are, however, elements of the interface that are repeated on many pages throughout the module, such as:

Category icons When you click on the icon for a virtual server, directory, or options file, a table of icons with names like **MIME Types** and **CGI Programs** is displayed at the top of the page. Under each of these icons are fields and tables for configuring options related to the label of the icon they are under. This commonly used layout breaks down the vast number of editable Apache options into categories, as there are far too many fields to display on a single page. The exact icons that appear and the fields under them differ depending on the part of the web server configuration you are editing, and the version of Apache that is installed. Their basic layout, however, is always the same.

Tables fields On many forms, some fields use tables for entering multiple values such as MIME types and their associated file extensions. There is no limit on how many rows each table can have, but Webmin will only display a single empty row in each table at any one time. This keeps down the size of forms that have lots of tables, but means that you can only add one new row to a table at a time. To add more than one, you will need to save the form and then re-enter it, which will cause a new blank row to be displayed below the one you just filled in.

The following sections explain in more detail exactly which icons to click and which tables to fill in when you are doing things like enabling CGI scripts and setting MIME types.

29.3 Starting and Stopping Apache

Before browsers can connect to the Apache Web server on your system, Apache's server process must be started. You can check to see if it is currently running by looking at the top of any of the pages in the module. If links labelled **Apply Changes** and **Stop Apache** appear, then it is currently active. If only the link **Start Apache** appears, however, it is not yet running.

To start it, click the **Start Apache** link. If all goes well, the page that you are currently on will be redisplayed and the links at the top should change to indicate that it is now running. Otherwise, an error message will appear explaining what went wrong. Most likely the cause will be an error in the configuration file.

To stop the web server once it is running, click the **Stop Apache** link on any of the module's page. In the unlikely event that Webmin is unable to stop the server, an error message page will be shown. If it is successfully stopped, the same page will redisplay with the links at the top changed to show that it is no longer running.

When Apache is active, every page will have an **Apply Changes** link at the top that can be used to signal the web server to reload its current configuration. After you make any changes in this module (except those in .htaccess files), this link must be clicked to make them active. Unlike other Webmin modules that have an **Apply** button on the main page, this one has it on every page so you do not have to return to the index every time you make a change.

29.4 Editing Pages on Your Web Server

This section explains how to find and edit the files on your system that are displayed when a client connects to your Apache Web server. If you already know how to do this, feel free to skip it and move on to Section 29.5 "Creating a New Virtual Host".

When Apache is first installed from a package or from source, its initial configuration will typically not have any virtual servers set up. Instead, just the default server will exist, serving pages to any client that connects on port 80. You can view the default pages by running a web browser and going to the URL `http://yourhostname/`, or `http://localhost/` if you are running the browser on the same system on which Webmin resides. The page that appears will probably just be one supplied with Apache or your Linux distribution.

The document root directory out of which Apache serves files will be shown on the module's main page next to the **Default Server** icon. On Red Hat Linux for example, this directory is `/home/httpd/html` by default. The files in this directory can be edited by logging in as `root` or by using Webmin's File Manager module. Any changes that you make will immediately be reflected on the website.

If your system is just going to host a single static website, it may not be necessary to configure any other aspects of Apache. You can just upload or copy HTML, images, and other files to the directory and its subdirectories to create the site that you want. The most important file is `index.html`, which is served by Apache whenever a browser does not request a specific page. Because most people will go to `http://yourserver/` first, the `index.html` page will be the first one that they see.

To make editing easier, you may want to change the ownership of the document root directory and all its files to a non-root user. You must make sure, however, that they are still readable by the user as whom the Apache server process runs, which is typically named `httpd`. The easiest way to do this is to make all files and directories world-readable and world-executable.

29.5 Creating a New Virtual Host

If you want to host multiple websites on your system, then you will need to create an Apache virtual host for each one. Before you can add a site, its address must first be registered in the DNS, either on a DNS server on your system or on another host. If the site's files are to be owned by a different UNIX user than the one who owns the document root directory, then he must be created first as well.

The entire process for adding a virtual server is:

1. Decide on a hostname that will be used in the URL for the new website, such as `www.example.com`.
2. Decide if your new site is going to be IP-based, or name-based. A name-based site will work fine with all except for old browsers and so is, by far, the best choice these days. An IP-based site will work with any browser, but needs its own separate IP address to be added to your system. Because IP addresses are often scarce, this only makes sense if you need to set up a virtual FTP or POP3 server for the domain as well.
3. If your site is going to be IP-based, use the Network Configuration module (covered in Chapter 16) to add a new virtual IP address to the external network interface on your system. Make sure that it will be activated at boot time and is active now. If your system has only a single static internet IP address assigned by your ISP, then any extra virtual IP addresses that you add to it will not work. In that case, you will have to use a name-based virtual server instead, or request that your ISP assign you multiple addresses.
4. If the `example.com` domain already exists on a DNS server, add a record for `www.example.com` with the external IP address of your system (for a name-based site) or the address chosen in the previous step (for an IP-based site).
If the domain does not yet exist, you will need to add it to a DNS server and register it with a DNS registrar like Network Solutions. Either way, Chapter 30 explains how to add records and domains in detail.
5. If the site will belong to a different person, add a UNIX user account for him in the Users and Groups module covered in Chapter 4. It is a much better idea for the files for each site to be owned by separate users than a single one if they are going to be managed by different people.

When you create the user account, make sure it has a valid home directory such as `/home/example`. Then create a subdirectory called `www` under the home and make sure that it is owned by the new user. This can be done automatically for new users by creating a `www` sub-directory under `/etc/skel`, or wherever default files for new users are stored on your system.

6. If the site is going to use the standard HTTP port 80 (which is almost always what you want), then you can skip to Step 8. Otherwise, click on the **Networking and Addresses**

icon on the Apache Webserver module's main page to bring up the form shown in Figure 29.3.

7. In the empty row in the **Listen on addresses and ports** table, select **All** under the **Address** column and deselect **Default** under the **Port** column. Then, enter the TCP port number for your website in the field next to it and click the **Save** button at the bottom of the page.
8. On the module's main page, scroll down to the **Create a New Virtual Server** form below the list of existing virtual hosts.
9. If you are setting up an IP-based virtual server, you should enter the virtual IP address in the **Specific address** field that was added in Step 3. If setting up a name-based virtual server, enter the external IP address of your system into the field instead. If your Apache server has been configured to accept name-based connections on any IP address, you can select the **Any address** option for this field instead. See the following explanation for more details.

If your new virtual server is going to use a port other than 80 and will be the only server on that port, you can select the **Any address** option as well so it handles all requests that come in on the port.
10. If you are setting up an IP-based virtual server, deselect the **Add name virtual server address** checkbox. For name-based servers, it should be left enabled.
11. If the new virtual host is going to use a nonstandard port, select the last option for the **Port** field and enter the number into the field next to it.
12. In the **Document Root** field, enter the full path to the directory that will contain files for this website. For example, this might be `/home/example/www`.
13. In the **Server Name** field, enter the hostnames that clients will use to refer to this website such as `www.example.com`. You can enter more than one name, such as `web.example.com` and `example.com`, if this is going to be a name-based server that should be accessible at several different URLs.
14. Unless you have a separate file on your system that contains all virtual hosts, leave the **Add virtual server to file** field set to **Standard httpd.conf file**. Otherwise, you can choose **Selected file** and enter the path into the field next to it. Make sure that the chosen file is actually used by Apache (such as by an `Include` directive in `httpd.conf`) or the virtual server will be useless and will not appear in Webmin.

If you always use the same separate file for storing virtual hosts, the **File to add virtual servers to** field explained in Section 29.22 “Configuring the Apache Webserver Module” may be useful. When this configuration field is set, an option for creating the virtual host in the chosen file is added to the **Add virtual server to file** field.

15. To have Webmin copy all of the directives from another virtual server to the one that you are creating, select it from the **Copy directives from** menu. This can be useful if all of your virtual hosts have a similar configuration.
16. When you are done filling in the form, click the **Create** button. The new virtual server will be added to the Apache configuration file and to the list of servers on the main page.
17. Click on the icon for the new virtual server, which will take you to its options page, shown in Figure 29.4.

18. Scroll down to the form under **Per-Directory Options** and enter the document root directory that you chose in Step 11 in the **Path** field. Make sure the **Type** is set to **Directory** and the **Regexp?** field to **Exact match**.
19. Click the **Create** button to add a new section to the configuration file for the directory. This is necessary for granting clients the rights to browse files contained in the directory, because the default Apache directory configuration will deny access.
20. Click on the new icon for the directory that has been added to the virtual server options page. This will take you to the directory options page shown in Figure 29.5.
21. Click on the **Document Options** icon and change the **Directory options** field to **Selected below** on the form that appears. Under the **Set for directory** column, change the entry for **Generate directory indexes** to **Yes**. Then click the **Save** button at the bottom of the page.
22. To make all your changes active, click the **Apply Changes** button at the top of any page.
23. You or the user who owns the virtual server can now start adding files to the document root directory. You can test it out by opening the URL (such as *www.example.com/*) in your web browser to make sure that everything is working properly.

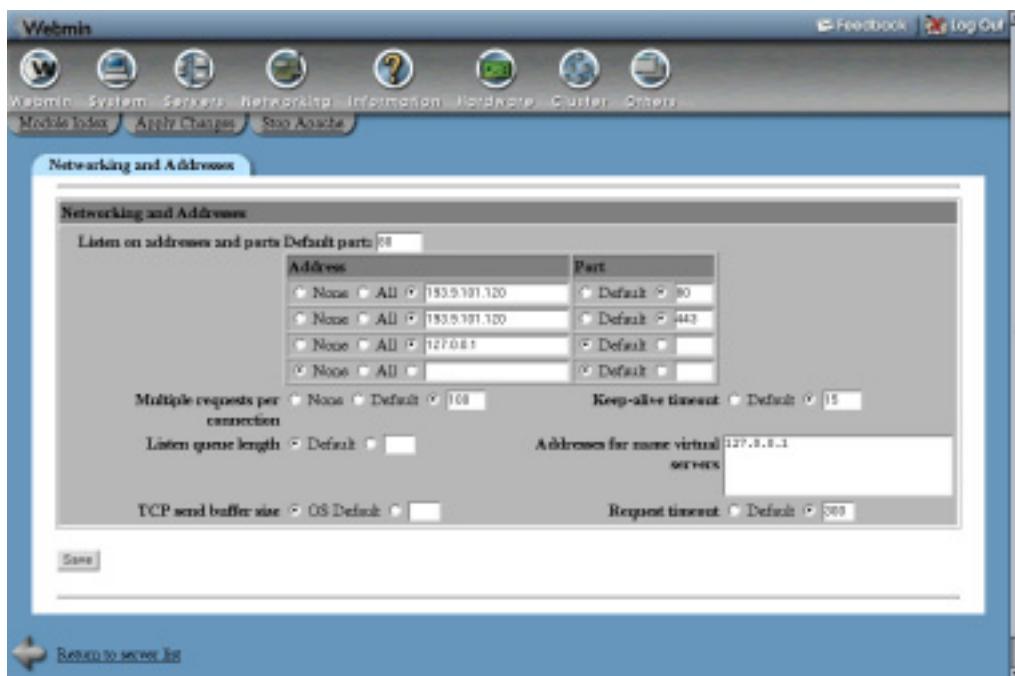


Figure 29.3 The global networking and addresses page.

When Apache receives an HTTP request, it must first work out which virtual server the request is for. It will first look for a name-based virtual server whose hostname matches the host requested by the client, and whose address and port are the same as the ones to which the client

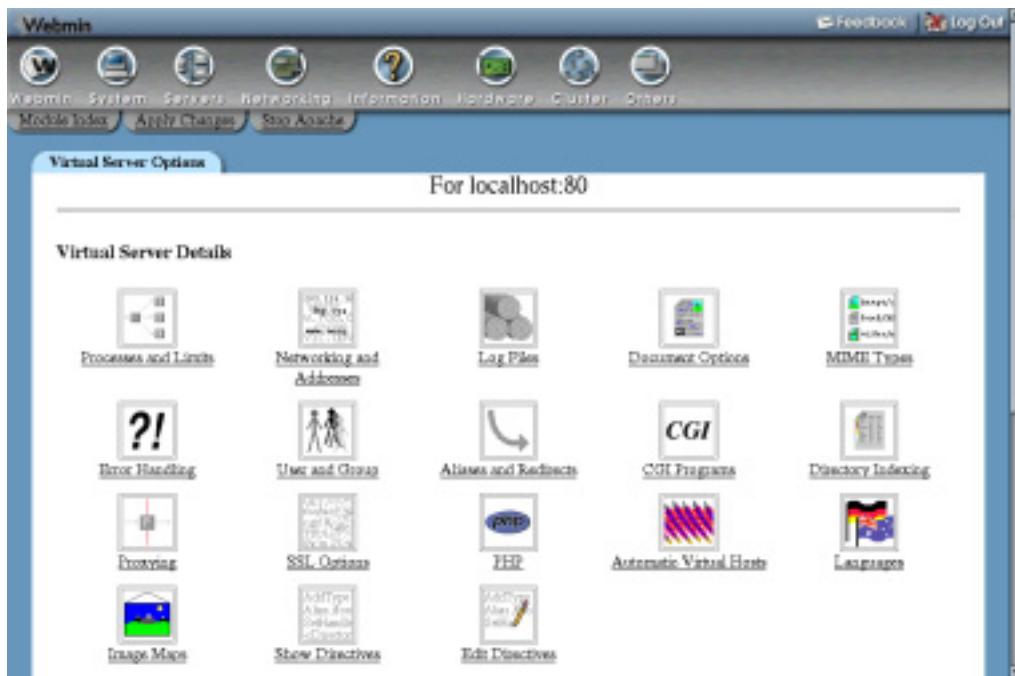


Figure 29.4 The virtual server options page.

connected. If none is found, the first defined virtual server for the address and port will be used instead, or if there are none then the request will be handled by the default server.

Name-based virtual servers can only be used on addresses listed in the **Addresses for name virtual servers** field on the global Networking and Addresses page. If you follow the instructions above, an address will be added to this list automatically when you create a new virtual server. If all the virtual servers on your system are going to be name-based, you can open this page, select the **Include all addresses** option, and click **Save** so that Apache will handle such requests on any IP address. This also makes sense if your system has a dynamically assigned IP address and you want to serve multiple virtual hosts.

Once a virtual server has been created, you can edit its settings or delete it by following these steps:

1. On the module's main page, click on the virtual server's icon. This will take you to the server options page shown in Figure 29.4.
2. Scroll down to the **Virtual Server Details** form at the bottom of the page.
3. Change the **Address**, **Port** and other fields to whatever you want and click the **Save** button. These fields have the same meanings as on the virtual server creation form. If the address is changed on a name-based virtual server, however, you may need to change it on the global Networking and Addresses page as well.

Or if you want to get rid of the virtual server and all the configuration directives that it contains, click the **Delete Virtual Server** button instead.

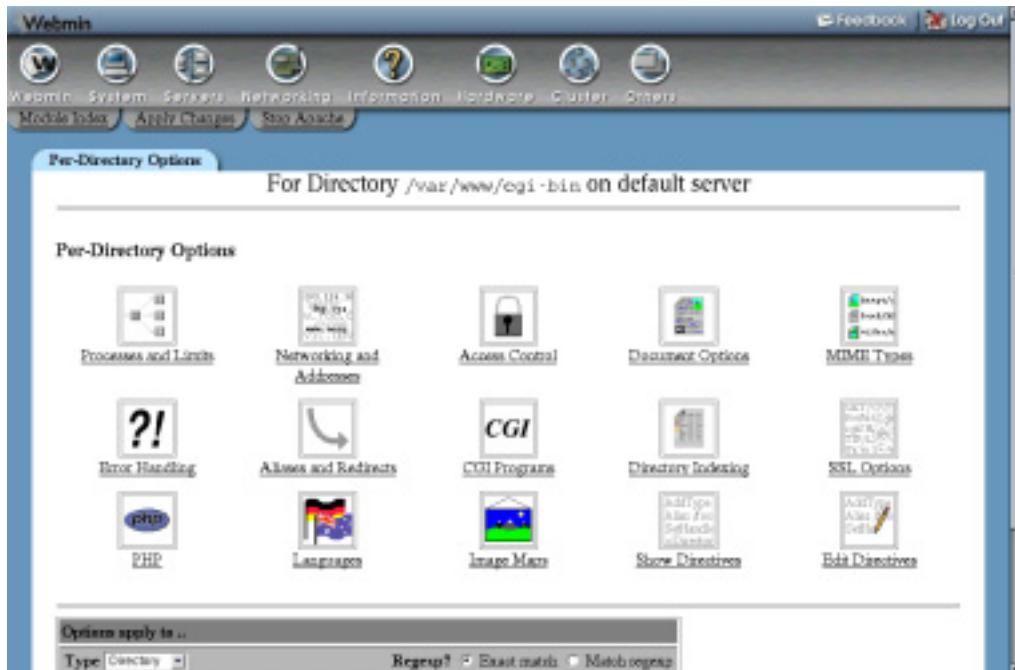


Figure 29.5 The directory options page.

4. Back on the module's main page, click on the **Apply Changes** link to make the new settings active.

You cannot change the settings for the default server, nor can you delete it.

29.6 Setting Per-Directory Options

Apache allows you to specify different options for certain directories—either for all virtual servers or just a single one. Including directories, you can actually set options that apply to three types of objects on your Apache server:

Directory The options apply to a specified directory and all files in it or in the subdirectories that it contains.

Files The options apply to files with a specified name in any directory.

Location The options apply to any files or directories requested by a URL whose path starts with the specified location. For example, in the URL `www.example.com/foo` the path would be `/foo`.

Whenever Apache processes a request, it checks for the options that apply to it in a fixed order. Those from directory sections and `.htaccess` files are read in order so that the most specific directories are checked first. They are then followed by files and then location sections. Options from the virtual server to which the request was made (if any) are then read, and finally options from the default server.

This means that options set for a directory will override the same options set in a higher level directory, or in the virtual server of which it is a member. To set options for a directory, files, or a URL location, follow these steps:

1. Even though the options you are going to set apply to a directory, they must be defined under one of the virtual servers or the default server. If they are under a virtual host, they will apply only to requests to that server for files in the chosen directory or URL location. If they are under the default server, however, requests to any virtual host for files in the directory will be affected.

On the module's main page, click on either the **Default Server** icon or the icon for a virtual server to which you want the directory options to be limited. For directories, it is usually simplest to put their options under the default server as each virtual host typically has its own separate document root directory. URL location options, however, should be put under the virtual server to which they are related because the same URL path may be used in different ways on more than one virtual host. The same goes for file options.

2. On the server options page that appears (shown in Figure 29.4), scroll down to the **Create Per-Directory, Files, or Location Options** form.
3. From the **Type** menu, choose one of the options previously described.
4. If you are setting options for a directory, enter it into the **Path** field such as `/home/example/www/images`. You can also enter a wildcard path such as `/home/example/w*`, which will cause the options to apply to all directories that match.

If the options are being set for a URL location, enter the part of the URL after the hostname into the **Path** field, such as `/images`. You can also use shell wildcard characters like `*` and `?` in the URL as well. If setting options for files, enter a filename into the **Path** field such as `secret.html`. Once again, wildcard characters can be used in the filename, for example `secret*`.

5. If you want to be able to use complex regular expressions in the directory, filename, or URL location, set the **Regexp?** field to **Match regexp**. This will allow you to use Perl regular expression characters like `[`, `]`, `+`, `.` and `*` in the path.
6. Click the **Create** button to add the new directory section to the Apache configuration. The virtual server options page will be displayed again, but with a new icon for the directory.

Now that you have created a new icon for a directory, URL location, or filename, you can set options that apply to it. One of the most common per-directory changes is configuring how files are listed when a browser requests a directory with a URL like `www.example.com/images/`. By default, if there is an `index.html` file in the directory it will be displayed. If not, a page listing all files that it contains will be shown instead.

If you want to change the name of the index file, the style of the directory listing, or any other settings related to indexing, follow these steps:

1. Click on the icon for the directory that you want to configure on the virtual server options page. This will take you to the directory options page shown in Figure 29.5.
2. Click on the **Directory Indexing** icon to bring up a form for setting indexing and listing options.

3. To change the appearance of directory listings, set the **Directory index options** field to **Selected below** and change the fields in the box provided. The defaults will generate a very plain list of files, but you can enhance it by setting the following options:

Display fancy directory indexes If enabled, the list of files will include their icon, size, and modification date.

Display HTML title as description If enabled, the description for HTML files will be taken from their `<title>` tags.

Icon height This option allows you to change the height of icons included in the directory listing. If it is set to **Default**, the height of the standard Apache options will be used.

Icon width Like the previous option, this one allows you to specify the width of icons in the directory listing.

Allow user sorting of column When this is enabled, users will be able to sort the list of files by clicking on the column headings, assuming they are being displayed.

Show file descriptions If enabled, the directory listing will include a description for each file taken from its MIME type or HTML title.

Output HTML header tags When enabled, the directory listing will include the normal `<html>` and `<head>` tags that should begin every HTML page. You will only want to turn it off if you are providing your own header and footer files.

Show last modified times When enabled, the directory listing will include the last modified date for each file.

Show file sizes When enabled, the listing will include the size of each file.

Include icon in link If this option is enabled, the icon in the listing will be a link to the file itself. Otherwise, only the filename is a link.

Filename width This option controls the length of the filename column in the directory listing. You can either enter a number of characters or * to size the column to the length of the longest filename.

Description width This option controls the length of the description column in the directory listing, if any. You can either enter a number of characters or * to size the column to the length of the longest description.

Display directories first If enabled, the listing will show any directories above any files, regardless of any other files.

The options that are available depend on the version of Apache that you have installed on your system. Those listed above are valid for version 1.3.19, but if you have a newer release more options may be available.

4. If you want Apache to return a file other than the default (usually `index.html`) when a browser requests the directory, enter a list of filenames into the **Directory index files** field. More than one can be entered and the first that is found will be used. If none of the index files are found, a directory listing using the options chosen in Step 3 will be returned to the browser instead.
5. To have the web server ignore certain files when generating a list of files in the directory, enter their filenames into the **Files to ignore in directory index** field. You can use shell wildcards in the regular expressions, such as `*.doc`.

6. To have an HTML file inserted at the start of the directory listing, enter its filename (relative to the directory) into the **Directory index header file** field.
7. Similarly, to have a filename added at the end of the directory listing, enter its name into the **Directory index footer file** field.
8. To control the default ordering of the directory, deselect **Default** in the **Sort directory index by** field and select an order and column to sort on from the two menus next to it.
9. You can set descriptions for files by filling in the **Directory index descriptions** table. In the table's empty row, enter a short message describing the file in the **Description** column and a list of filenames or wildcard names in the **Filenames** column.
Because only one empty row is shown at a time, you will need to revisit this page after adding each description if you want to enter more than one.
10. Finally, click the **Save** button at the bottom of the page to store your changes and return to the directory options page. To activate them, click the **Apply Changes** link anywhere in the Apache module.

Most of these options can be set for an entire virtual server by clicking on the Directory Indexing icon on the Virtual Server Options page as well. In this case, they will apply to all files requested from the virtual host unless overridden by options for a directory or URL location.

On the directory options page, there are many more icons on which you can click to set options that apply only to that directory, URL path, or filename. Some of these are explained in later sections of this chapter, such as Section 29.7 “Creating Aliases and Redirects” and Section 29.13 “Password Protecting a Directory”.

You can change the directory, filenames, or URL location to which settings apply by using the **Options apply to** form at the bottom of the Directory Options page. It has the exact same fields as the creation form described at the start of this section. If you make any changes, click the **Save** button to update the Apache configuration and then the **Apply Changes** link to make them active. You can also click on **Delete** to remove the directory configuration and all its options.

29.7 Creating Aliases and Redirects

Normally, there is a direct relationship between the path in the URL and the file that is returned by the web server. For example, if a browser requests `www.example.com/images/foo.gif` and the document root for `www.example.com` is `/home/example/www`, the file `/home/example/www/images/foo.gif` will be read by the web server and returned to the client.

This can be changed, however, by using what Apache calls *aliases*. An alias maps a particular URL path to a file or directory, which does not necessarily have to be under the document root. In the previous example, the `/images` URL path might actually be an alias to the directory `/www/images`, which would cause the file `/www/images/foo.gif` to be read instead.

Aliases can be defined globally or in a virtual server. To create one, follow these steps:

1. On the module's main page, click on the icon for the virtual server under which you want to create the alias. If you want it to apply to all virtual servers (or you don't have any), click on the **Default Server** icon instead.
2. On the virtual server options page that appears next (shown in Figure 29.4), click on the **Aliases and Redirects** icon. This will take you to the page shown in Figure 29.6.

3. Fill in the empty row in the **Document directory aliases** table with the URL path (under **From**) and the file or directory to which it should map (under **To**). If you are editing the default server, there may already be several entries in this table that are part of the standard Apache configuration.

There will always be exactly one empty row in the table. If you need to add more than one alias, you will need to revisit this page after filling in the row and saving.

4. Click the **Save** button to have your new alias stored in the Apache configuration. The browser will return to the virtual server options page.
5. To make the alias active, click on the **Apply Changes** link at the top of the page.



Figure 29.6 The aliases and redirects form.

Existing aliases can be editing by just changing the entries in the **Document directory aliases** table and then clicking **Save**. You should not change the alias for `/icons` in the default server, though, as this is used by Apache when it generates icons for directory listings. If you want to delete an alias, just delete the contents of both its fields in the table.

Aliases that use Perl regular expressions can also be created to match more complex URL paths. These must be entered into the **Regexp document directory aliases** table on the **Aliases and Redirects** form, which has the same columns as the **Document directory aliases** table described above. The difference is that any regular expression can be entered into the **From** field, such as `^/images/(.*).gif$`. The **To** field can take a string that refers to bracketed sections in the expression, such as `/images/$1.jpg`. This would convert any request for a GIF file into one for the JPEG with the same name.

Redirects are similar to aliases, but have a different purpose and work in a different way. Whenever a client requests a URL path that has been redirected, Apache will tell it to go to another URL (possibly on another server) instead. For example, you might redirect all requests to `www.example.com/webmin/` to `www.webmin.com/`. Unlike the way aliases behave, if a browser requests a page like `/webmin/foo.gif` it will not be redirected to `www.webmin.com/foo.gif`—it will just go to the URL `www.webmin.com/` instead.

Redirects are implemented by the web server sending the special 302 status code to the browser, which tells it to go to a new location. It is quite possible for the new URL to be a redirect itself, and you can even create a loop of redirects—not that this is a good idea.

To set up redirection for a path on your server, follow these steps:

1. On the module's main page, click on the icon for the virtual server under which you want to create the redirect. If you want it to apply to all virtual servers, click on the **Default Server** icon instead.
2. On the virtual server options page that appears, click on the **Aliases and Redirects** icon to go to the page in Figure 29.6.
3. In the empty row of the **URL redirects table**, enter the URL path on your server under the **From** column, such as `/webmin`. Under the **To** column, enter the URL to which requests should be redirected, such as `www.webmin.com/`.

The **Status** field is optional, but can be filled in if you want to change the HTTP status code that will be used for this redirect. The default is 302, which indicates a temporary redirection. You can, however, 301 to tell browsers that the direction is permanent or 303 to tell them that the original content has been replaced.

There will always be exactly one empty row in the table. If you need to add more than one redirect, you will need to revisit this page after filling in the row and saving.

4. Click the **Save** button to have your new redirect stored in the Apache configuration. The browser will return to the Virtual Server Options page.
5. To make the redirection active, click on the **Apply Changes** link at the top of the page.

As with aliases, existing redirects can be edited by just changing the entries in the **URL redirects** table and then clicking **Save**. To delete a redirect, just delete the contents of all of its fields in the table.

You can also create regular expression redirects that behave in a similar way to *regexp* aliases, using the **Regexp URL redirects** table on the same page. Under the **From** column you can enter a URL path expression such as `^/webmin/(.*)$`, and under the **To** column you can enter a URL that can refer to bracketed parts of the path, such as `http://www.webmin.com/$1`. In this example, a request by a client for a page under `/webmin` would be redirected to the same file at `www.webmin.com`.

Also on the Aliases and Redirects page are two more tables labeled **Permanent URL redirects** and **Temporary URL redirects**. The first behaves exactly like a normal redirection, but with the status code always set to 301, indicating a permanent redirection. The second also behaves like a normal redirect, but always uses a status code of 302 (temporary redirection). This option is really quite useless, as normal redirections default to using status 302 if one is not specified.

Redirects can also be defined in the options for directories, URL locations, filenames, and `.htaccess` files. When editing the options for one of these (described in Section 29.6 “Setting

Per-Directory Options”), the exact same icon and table are available as when setting up aliases for a virtual server. Naturally, a redirect in a directory only makes sense if the URL path being redirected actually refers to some file or subdirectory that it contains. The same goes for redirects in URL locations—the path being redirected must start with the location’s path.

If Apache on your system has been compiled with or dynamically loads the proxy module (covered in Section 29.18 “Configuring Apache as a Proxy Server”), tables labeled **Map locale to remote URLs** and **Map remote Location: headers to local** will appear on the **Aliases and Redirects** form under the virtual server options page. These allow you to specify a URL path that, when requested, will cause Apache itself to request pages from another website and return them to the browser. Although the URL that the user is accessing is on your server and their browser is connecting only to your system, the content is actually being loaded from elsewhere.

To set up this URL mapping, follow these steps:

1. On the module’s main page, click on the icon for the virtual server that you want to create the mapping under. If you want it to apply to all virtual servers, click on the **Default Server** icon instead.
2. On the virtual server options page that appears, click on the **Aliases and Redirects** icon to go to the page shown in Figure 29.6.
3. In the empty row in the **Map locale to remote URLs** table, enter a URL path on your server (like `/webmin`) into the first field, and the full URL from which you want the pages to be requested into the second (like `http://www.webmin.com/`).
4. In the empty row in the **Map remote Location: headers to local** table, enter the same full remote URL into the first field and the URL path on your server into the second. This second table controls the conversion of redirects issued by the remote server and should almost always be set. If it is not set, the browser will end up connecting directly to it instead of to your server whenever the remote server issues a redirect.
5. Click the **Save** button to have your new mapping stored in the Apache configuration. The browser will return to the virtual server options page.
6. To make the mapping active, click on the **Apply Changes** link at the top of the page.

You can test it out by going to the mapped URL path on your system. You should see pages that have been requested from the remote server. The process is not totally transparent though, because it does not convert HTML files in any way. If in the previous example the remote server contained an HTML page with a link like ``, following it would take the browser to `/foo.html` on your system, not `/webmin/foo.html` as you might expect. There is no solution to this problem, apart from making sure that the remote server always uses relative links and image paths.

29.8 Running CGI Programs

CGI stands for Common Gateway Interface and is a standard method for web servers to run external programs, pass them details of a browser’s request, and read back any content that the program generates. CGI programs are one of the simplest ways of adding dynamic pages to your web server, and are relatively easy to set up and develop. Server-side includes (covered in Section 29.9 “Setting Up Server-Side Includes”) are even simpler, but very limited in what they can do.

A CGI program can be written in any language as long as it follows certain rules. The most common language is Perl, but C, Python, PHP, or any other language that can access environment variables and produce output can be used. You can even write shell scripts that are valid CGI programs. This section is not going to explain the details of how to write them, however—there are plenty of books that cover that already.

CGI programs are just files on your system, like any other HTML or image file. The difference is that when they are requested by a browser, Apache executes them and returns their output instead of the contents of the file. Because you only want this to happen for programs and not for HTML files, the server must be configured to identify certain files as CGI programs. This is normally done in one of two ways—by putting all CGI programs into a certain directory, or by giving them all a file extension like `.cgi`.

The choice is yours, but the latter option is simpler to use as you can freely mix CGI scripts, HTML, and image files in the same directory. To set it up, use the following steps:

1. On the module's main page, click on the icon for the virtual server for which you want to set up CGI programs. Or, click on the **Default Server** icon if you want to use them on all servers.
2. Click on the icon for the directory under which you want CGI programs to be enabled. Typically, each virtual server will have an icon for options for its document root directory, but if not, you can create one by following the steps in the earlier Section 29.6 “Setting Per-Directory Options”. If you only want to allow CGI programs to be run in a particular subdirectory of the website, you can create a new directory icon for that as well.
3. On the directory options page, click on the **Document Options** icon and change the **Directory options** field from **Default** to **Selected below**. Then, set the rows **Execute CGI programs** and **Generate directory indexes** to **Yes**, and click the **Save** button at the bottom of the page. This tells Apache that CGI programs can be executed in the directory.
4. Back on the directory options page, click on the **MIME Types** icon. In the **Content handlers** table, select **cgi-script** from the first blank menu under the **Handler** column, and enter `.cgi` into the field next to it under the **Extensions** column. Then click the **Save** button at the end of the form. This tells Apache to treat all files in the directory ending in `.cgi` as CGI programs.
5. Finally, click the **Apply Changes** link on any page. You should now be able to create a file with a `.cgi` extension in the chosen directory and test it out in a web browser.

An alternative to this approach is to specify a directory in which all files are treated as CGI programs. This has the advantage that they can be given any name you like, instead of being forced to have a `.cgi` extension. You can also set permissions on this directory to restrict who is allowed to create CGI programs, while still allowing others to edit normal HTML pages.

To set up a directory for CGI scripts, use the following steps:

1. On the module's main page, click on the icon for the virtual server for which you want to set up a CGI directory. Click on the **Default Server** icon if you want to set it up for all servers.
2. Click on the **CGI Programs** icon to bring up a page for setting various CGI options.
3. The **CGI directory aliases** table works in a very similar way to the **Document directory aliases** table described in Section 29.8 “Running CGI Programs”. In addition to mapping

a URL path to a directory on your server, it also tells Apache that any files accessed through that path should be treated as CGI programs.

In the first empty row of the table, enter a URL path like `/cgi-bin/` into the **From** field and a directory like `/home/example/cgi-bin/` into the **To** field.

4. Click the **Save** button at the bottom of the page to return to the virtual server options page. Then click the **Apply Changes** link to make the CGI directory active.

You should now be able to create CGI programs in the directory and test them out in a web browser. On some Linux distributions, the default Apache configuration will already have a CGI directory available at the URL path `/cgi-bin/` mapped to a directory like `/home/httpd/cgi-bin/`. If this is good enough for you, there is no need to follow these steps. Instead, you can just put CGI programs in that directory.

All CGI programs normally execute as the UNIX user as whom the web server runs, typically named `httpd` or `apache`. On a system with multiple users who cannot be fully trusted, this is not a good thing—anything that one user’s CGI program can do, everyone else’s can as well. For example, if a user writes a CGI program that edits a particular file, he has to make that file writable by the `httpd` user, meaning that everyone else’s CGI programs can write to it as well.

Fortunately, there is a solution. Apache ships with an optional program called `suexec` that can be used for running CGI programs as another UNIX user rather than as the web server user. Typically the CGI programs under each virtual server will be run as the UNIX user who owns that server’s files. To set this up, you can follow these steps:

1. Make sure that the `suexec` program exists on your system, and that it has setuid-root permissions. Apache typically expects to find it in `/usr/sbin` or `/usr/local/apache/sbin` and most Linux distributions include it as a standard part of their Apache package. Some do not have it setuid by default, however, so you may need to run `chmod 6711 /usr/sbin/suexec` to make it so.
2. On the main page of the module, click on the icon for the virtual server on which you want to have CGI programs run as a different user. This will take you to the options page shown in Figure 29.4.
3. Click on the **User and Group** icon on the Virtual Server Options page.
4. For the **Run as UNIX user** field, select **User name** and enter the name of the user who owns the virtual server into the field next to it.
5. Similarly, for **Run as UNIX group**, select **Group name** and enter the primary group of the user specified in the previous step.
6. Click the **Save** button to return to the options page for the virtual server.
7. To activate `suexec` for the first time, you need to stop and restart Apache. Use the **Stop Apache** link at the top of the page to halt it, and then the **Start Apache** link to start it up again.
8. To check that `suexec` is actually working, check the Apache error log file for a line containing `suEXEC mechanism enabled` that was logged when the web server was restarted.

Because it can execute commands as any user on your system, `suexec` has many security restrictions to prevent misuse by normal users. It will only run CGI programs that are owned by

the user and group specified in Steps 4 and 5, and only if they are not writable by any other user or in a directory that is writable by another user. The IDs of the user and group must be above the minimums that are compiled into the program to prevent programs owned by system users (such as `root` or `bin`) from being run. Finally, the program must reside under a directory that is compiled into `suexec` and nowhere else on the filesystem.

This last restriction can be very annoying if you have a large number of virtual servers and want to enable the execution of CGI programs in their directories. The default allowed directory is typically the standard CGI directory for Apache, such as `/home/httpd/cgi-bin`. To change this, you will need to recompile `suexec` with a different directory, such as `/home`.

Whenever `suexec` fails to run a CGI program, it fails with HTTP status code 500. Because there are many things that can go wrong, you should check the file `suexec_log` in the same directory as the other Apache logfiles to see why it is refusing to execute a particular program. For each failure, a line is written to this file explaining the problem, such as incorrect permissions or a file ownership mismatch.

Writing CGI programs can be difficult, because when they fail, very little information is displayed in the browser. All you see is a message like `500 server error`, with no explanation of the real cause. More detailed error information, however, is written to the Apache error log file. This is usually named `error_log`, and can be found in the same directory as the Apache access log files. See Section 29.10 “Configuring Logging” for more details on how to find and change it.

Anything that a CGI program outputs to `STDERR` will also be written to the error log, which is useful if you want your program to generate debugging information that is not sent to the web browser. Because many programming languages like Perl output error messages on `STDERR` if a script fails to compile or run, all such messages will also be written to the error log file.

The biggest problem with CGI programs is that the web server has to launch a new process every time one is requested. If the CGI is written in Perl or PHP, the process then has to load the interpreter for that language, which can be a large program in itself. The end result is that processing a request for a CGI page takes much longer than a request for a static HTML or image file, and generates much more load on the server system.

For this reason, optional modules have been developed that allow the web server to run Perl and PHP scripts using an interpreter that is part of the Apache process. These modules are called `mod_perl` and `mod_php` and are included in the Apache package in many Linux distributions. Installing and configuring them, however, is not covered in this chapter.

29.9 Setting Up Server-Side Includes

Server-side includes allow you to create simple dynamic web pages without the complexity of writing an entire CGI program in a language like Perl. When active, some of the HTML files served by Apache are checked for special tags starting with `<!--`. The content of each tag is then replaced by dynamically generated text that depends on the tag’s parameters and the resulting page is sent to the web browser.

The most common use of server-side includes is incorporating the contents of one HTML page into another. This can be useful if you have a common header or footer that you want to share among multiple pages without repeating it over and over again. Where a special tag like `<!--include file="something.html" -->` appears in the HTML of the page, it is replaced with the contents of the file `something.html`.

Server-side includes can also be used to access and set environment variables, to conditionally display HTML based on variables, and to run CGI programs or shell commands and have their output included in the page. This section will not cover the tags that are available and their purposes. For more information on tags, read the documentation on the Apache website or a good book on HTML.

Normally, allowing untrusted users to create HTML pages containing server-side include tags is perfectly safe because they cannot be used to perform potentially dangerous operations like editing files on the server. The exception to this is the `<!--#exec -->` tag, which can be used to run an arbitrary shell command and include its output in the web page. Because the command runs as the UNIX user as whom Apache is running (normally `httpd`), a user who is not allowed to create CGI programs may be able to use this kind of tag to read or modify files that he would not normally be able to access. For this reason, Apache can be configured to enable server-side includes with or without the risky `exec` tag.

Because checking an HTML file for server-side include tags is CPU-intensive, they are often only activated for files with the `.shtml` extension. This way you can put static HTML in `.html` files and dynamic content into `.shtml` files so the server does not have to waste time looking for tags in files in which they do not exist. You can also check all `.html` files for server-side includes if you wish.

To turn on includes for a virtual server, follow these steps:

1. On the module's main page, click on the icon of the virtual server on which you want to enable server-side includes. Or, click on the **Default Server** icon to enable them for all virtual hosts.
2. Click on the icon for the directory under which you want server-side includes to be enabled. Typically, each virtual server will have an options icon for its document root directory. If not, you can create one by following the steps in Section 29.6 "Setting Per-Directory Options".
If you only want to enable server-side includes in a subdirectory of the website, you can create a new directory icon for that as well.
3. On the directory options page, click on the **Document Options** icon and change the **Directory options** field from **Default** to **Selected below**. If you want to enable server-side includes without the `exec` tag, change the **Server-side includes** row to **Yes**. If you want to enable the potentially risky `exec` tag as well, change **Server-side includes and execs** row to **Yes**. Either way, when they have been enabled, click the **Save** button at the bottom of the page.
4. Click on the **MIME types** icon on the directory options page.

If you want to enable includes on all HTML files, find the **Content handlers** table and select **server-parsed** from the first empty menu under the **Handler** column. Enter `.html` into the field next to it under the **Extensions** column. This tells Apache that files ending in `.html` should be checked for server-side include tags.

If you want to enable includes for only `.shtml` files, enter `.shtml` instead of `.html` under the **Extensions** column. In the **Extra MIME types** table, enter `text/html` into the first empty field under the **Type** column and `.shtml` into the field under **Extensions** next to it. This tells Apache that `.shtml` files should be checked for server-side include tags and that they actually contain HTML.

-
5. Finally, click the **Save** button at the bottom of the MIME Types page and then click the **Apply Changes** link back on the directory options page.

Once server-side includes are enabled, you can test them by creating an .html or .shtml file in the chosen directory with some special tags in it. Then, open the page in your web browser to see the result. If for some reason server-side includes were not enabled properly, nothing will show up at all because the `<!--` tag indicates an HTML comment. If, however, the tag is replaced by the message `an error occurred while processing this directive`, then includes are active but there is an error in the tag's parameters. More details will be written to the Apache error log file (described in Section 29.8 “Running CGI Programs”) if an error of this kind occurs.

There is another method of indicating to Apache that certain HTML files should have server-side include processing performed on them. The web server can be configured so that any .html file with the UNIX execute permission set is processed for include, by following these steps (you can set this permission with a command like `chmod +x file.html`).

1. Follow Steps 1 through 3 of the preceding instructions to enable server-side includes for some directory.
2. On the directory options page, click on the **CGI Programs** icon.
3. On the page that appears, change the **Process includes on files with execute bit?** field to **Yes**. You can also set it to **Yes and set last-modified date** to have Apache read the modification time for each processed HTML file and use that to set the `Last-Modified` HTTP header.
4. Click the **Save** button at the bottom of the CGI Programs page and then the **Apply Changes** link on any page.

You should now be able to set execute permissions on HTML files in the directory, and Apache will parse them for server-side include tags when they are requested. This allows you to selectively turn on include processing, while avoiding the problem of having to rename a file (and break links) just because it now contains include tags.

29.10 Configuring Logging

By default, every request that Apache finishes processing is written to a log file in a standard format. For each request, the client IP address, website username, date, time, URL path, status code, and number of bytes transferred is logged. In the default Apache configuration, there is only a single log file that is used for all virtual servers. You can, however, reconfigure the web server to use different files for different virtual hosts, and even to log additional information for each request.

Apache also has a log file for recording error messages, which are generated when a browser requests a page that does not exist, when an HTTP connection is terminated, or if some other unexpected condition occurs. As Section 29.8 “Running CGI Programs” explains, this log file also contains error output from CGI programs and failure messages from server-side include directives.

To see which log files are being used by Apache on your system and to change them, follow these steps:

1. On the Apache Webserver module's main page, click on the **Default Server** icon. This will bring you to the default server options page similar to the one shown in Figure 29.4.
2. Click on the **Log Files** icon to bring up the log files configuration form shown in Figure 29.7.

3. The **Error log to** field controls where CGI and web server error messages are written. Typically, the **File** option is selected and the path to a file into which error messages should be written is displayed in the field next to it.

You can select the **System Log** option if you want to have messages sent to `syslog` instead (covered in Chapter 13). All messages will use the `local7` facility.

The other available option is **Program**, which, when selected, will cause Apache to run the command entered into the field next to it and feed error log messages to it as input. This can be useful for performing your own filtering or analysis of errors as they are reported.

4. The **Named log format** table lists predefined formats that can be used for logfiles defined in the next step. Each has a **Nickname** which is used to refer to it, and a **Format** string that specifies the fields written to the log for each request. When a log line is written, each of the % fields in the format string is replaced by some detail of the request, such as the client address, HTTP status code, or virtual server name. See the online Apache documentation for more details on which % fields are available.

Several standard formats such as `common` and `combined` are already defined in the default Apache configuration. To create your own log format, fill in the empty row at the bottom of the table. Each format must have a unique nickname.

5. The **Access log files** table specifies the files that are used for logging actual requests processed by the Apache Web server. Multiple files can be specified and the format of each can be selected independently from one of those explained in the previous step. All requests will be written to all listed logfiles.

Each row of the table defines one logfile. Under the **Format** column you can choose the format for the file or select the **Default** option to use the standard Apache log file format. Under the **Write to** column, you can choose whether the logging is being done to a file or to the input of a program. The path to that file or program must be entered into the field in the **File or program** column.

If you want to add an additional log file, fill in the fields in the empty row at the bottom of the table.

6. If you have made any changes to the logging configuration, click the **Save** button at the bottom of the page and then the **Apply Changes** link.

Apache also allows you to define different log files for each virtual server, so requests to the various virtual hosts on your system do not all get mixed up into one file. By default, all requests are written to a single access log file without any field that identifies the virtual server that processed them. To change this and have a virtual server write to its own separate log file, use the following process:

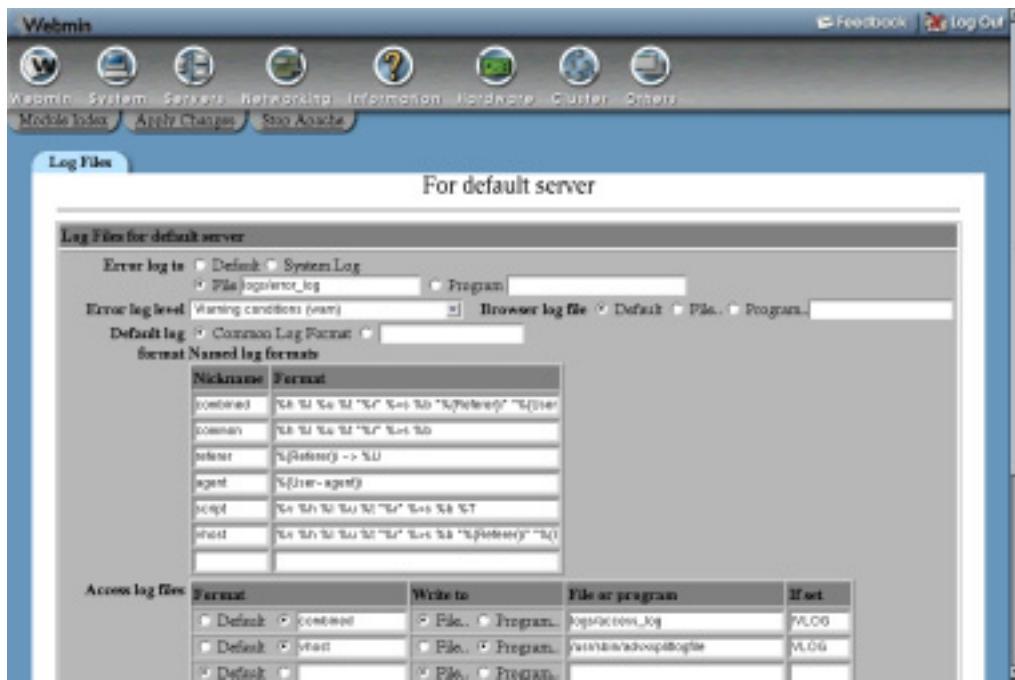


Figure 29.7 Default server log files configuration form.

1. On the module's main page, click on the icon of the virtual server for which you want to configure a new log file.
2. Click on the **Log Files** icon, which will take you to a page similar to the one in Figure 29.7.
3. If you want this virtual server to have its own separate error log file, change the **Error log to** field from **Default** to one of the other options.
4. To add a log format that exists only for this virtual server, fill in the empty row in the **Named log formats** table. It is usually a better idea to define all log formats in the default server, however, so they can be used in any virtual host.
5. Add a row to the **Access log files** table for this virtual server's separate log file. As soon as one is defined, requests to the virtual host will be written to it only instead of the access log list on the Log Files page under the default server.
6. When done, click the **Save** button at the bottom of the page to have your new log-file settings written to the Apache configuration. Then, back on the virtual server options page, hit the **Apply Changes** link at the top to make them active.

If you have multiple virtual servers and want to identify to which one each request was made, another solution is to change the format of the default access log file to include the virtual server hostname in each log line. To set this up, follow these steps:

1. On the module's main page, click on the **Default Server** icon and then on the **Log Files** icon on the default server options page.

2. Find the row for the common format in the **Named log formats** table and change its **Format** field so that it reads

```
%h %l %u %t "%r" %>s %b %{Host}i "%{Referer}i" "%{User-Agent}i"
```

The extra fields will tell Apache to include the virtual server hostname, referrer URL, and browser name for each request on every log line.

3. Find the row for your server's main log file in the **Access log files** table and make sure that the **Format** field is set to **common**, rather than **Default** or some other named format.
4. Click the **Save** button, and then the **Apply Changes** link. All entries written to the logfile from now on will include the additional information.

29.11 Setting Up Custom Error Messages

When a browser attempts to access a page that does not exist, a directory that is password protected, or a CGI program that is malfunctioning, Apache returns one of its built-in error messages. Because these error message pages are not always friendly or nice to look at, you can configure the web server to use your own pages instead. This can be set up to apply to all virtual servers, a single server or just one directory. The steps to follow are:

1. On the module's main page, click on either a virtual server or the **Default Server** icon if you want to define a custom error message that applies to all servers.
2. If you only want the custom message to be displayed for requests to a particular directory, URL path, or filename, click on its icon on the server options page. If no icon for the directory exists yet, you will need to define one by following the steps in Section 29.6 "Setting Per-Directory Options".
3. Click on the **Error Handling** icon in the directory or on the virtual server options page.
4. You can enter error codes and their corresponding custom messages in the **Custom error responses** table. Any existing error messages for the directory or server will be listed, followed by a blank row. To add a new code, start by entering the HTTP error number into the **Error code** field. Some of the more common codes and their causes are:

404 – The requested page does not exist.

403 – Access to the page is denied.

401 – The browser must log in first before accessing the page.

500 – A CGI program failed, or some other internal error occurred.

If you just want to change the message that Apache displays when the error occurs, select **Show message** under the **Response** column and enter the text of your new message into the field under **URL or message**.

On the other hand, if you want the contents of another page to be displayed instead, select **Goto URL** and enter either a URL page (like `/errors/500.html`) or a full URL (like `http://www.error.com/505.html`) into the **URL or message** field. In the latter case, the browser will be redirected to the URL when an error occurs with the chosen code.

5. Click the **Save** button at the bottom of the page. If you want to add another custom error message, click on the **Error Handing** icon again and fill in the new blank row in the table.
6. Click the **Apply Changes** button on any page to make the new custom error message active.

Some web browsers, such as IE in its default configuration, will not display the text of error messages sent by the web server. Instead, only the error code number and a more friendly message generated by the browser is displayed.

29.12 Adding and Editing MIME Types

MIME types are the method used by Apache, mail clients, and many other programs to indicate the type of files and other data. A MIME type consists of two words separated by a slash, such as `text/html`, `image/gif`, or `video/mpeg`. As those examples show, the first word is the general category of type, while the second is the actual type name.

Every response sent by a web server to a browser is accompanied by a type, so that the browser knows how to display it. When a normal file is requested, the web server typically works out the type by looking at the file's extension, such as `.gif` or `.html`. CGI programs must supply their type to the web server before any other content that they generate, which is then forwarded on to the browser. This allows a CGI program to generate HTML, plain text, images, or any other kind of data, regardless of the filename of the CGI script itself.

Browsers never attempt to work out the type of a page by looking at the filename extension in the URL. Instead, they always rely on the MIME type sent by the web server. Apache gets its global list of MIME types and the extensions with which they are associated from a configuration file that applies to all virtual servers. To edit and add to this list of types, follow these steps:

1. On the module's main page, click on the **MIME Types** icon in the Global Configuration section. This will bring you to a page listing all the types about which Apache currently knows, along with the filename extensions. Almost every type that you would ever need to use should already be listed.
2. Click on the **Add a new MIME type** link above or below the list to create a new type.
3. Enter a type name such as `text/foo` in the **MIME type** field of the form that appears. It is acceptable for the same type to be defined twice, as long as each entry has different associated filename extensions.
4. Enter all the filename extensions that you want associated with this type, such as `.foo` and `/fo`, in the **Extensions** text box. Make sure that no other MIME types are using the same extensions.
5. Click the **Save** button below the form. The browser will return to the types list, which will include your new entry.
6. Click the **Apply Changes** link on any page to make the new type active.

You can edit or delete an existing global MIME type by clicking on its name in the list, which will bring up the type editing form. Either change the **MIME type** or **Extensions** fields and click **Save**, or hit the **Delete** button to totally remove it. Either way, you must use the **Apply Changes** link afterward to make the changes active.

MIME types can also be defined on a per-virtual server or per-directory level in the Apache configuration. This can be useful if you want to override a type for some extension in a particular directory or create a type that is only needed by one virtual server. To do this, follow these steps:

1. On the module's main page, click on the icon for the virtual server for which you want to define the MIME type.

2. If you only want the type to be used for requests to a particular directory, URL path, or filename, click on its icon on the server options page. If no icon for the directory exists yet, you will need to define one by following the steps in Section 29.6 “Setting Per-Directory Options”.
3. In the directory or on the virtual server options page, click on the **MIME Types** icon.
4. The **Extra MIME types** table is for entering types that apply only to this virtual server or directory. In the first blank field under the **Type** column, enter a type such as *text/foo*. In the field next to it, under **Extensions**, enter one or more filename extensions such as *.foo*.
5. Click the **Save** button at the bottom of the page. If you want to add more than one type, you will need to click on the **MIME Types** icon again so a new blank field appears in the table.
6. When you are done, use the **Apply Changes** link at the top of any page to make the new type mapping active.

On the MIME types page, there is a useful field labeled **Default MIME type**. If set, any files for which Apache cannot identify the type will be treated as whatever is entered into this field instead. Normally, this is set at the default server level to `text/plain`, but you may want to change it to something else for a particular directory that contains lots of files that have no filename extension.

There is a similar field on the MIME types page for directories, URL paths, and filenames labeled **Treat all files as MIME type**. When it is set, Apache will identify all files in that directory as the specified type, no matter what their extension. This can be used to forcibly set the types of files whose names do not follow the normal convention of ending with a type extension.

29.13 Password Protecting a Directory

The HTTP protocol has a standard method of indicating that a directory or site requires a user-name and password to be supplied before it can be accessed. Apache can be configured to force users to log in before being able to view some or all of the pages on your system. Logins are typically checked against a separate password file, instead of the UNIX user list.

Password protection can be useful for securing a directory that only some people should be allowed to access, or for setting up a website that uses CGI programs to display different content to different users. To protect a directory, follow these steps:

1. On the module’s main page, click on the icon for the virtual server under which you want password protection to be enabled.
2. Click on the icon for the directory, URL location, or filename that you want to protect. If one does not exist yet, follow the steps in Section 29.6 “Setting Per-Directory Options” earlier in this chapter to create it.
3. Click on the Access Control icon, which will bring you to the page shown in Figure 29.8.
4. In the **Authentication realm name** field, deselect **Default** and enter a description for the protected directory, such as *Private files*. This will be displayed to the user in the browser when he tries to log in.
5. Change the **Authentication type** to **Basic**. The **Digest** type is more secure, but is not supported by a lot of browsers.

6. Change the **Restrict access by login** field to **All valid users**. This tells Apache that any of the users in the password file set in Step 7 will be allowed to log in.

You can restrict access to only a subset of users by selecting the **Only these users** option and entering the names of users to allow into the text field next to it. You can also select **Only these groups** and enter the names of groups whose members you want to allow into its field. These options can be useful if the same authentication files are entered on this page for several directories.

7. In the **Text file authentication** box, enter the full path to the file that you want to use to store usernames and passwords into the text field next to **User text file**. This authentication file must contain one line per user, each in the *username:encrypted-password* format. Standard UNIX encryption is used for the passwords, just like in the /etc/shadow file.

The file doesn't necessarily have to exist yet, as it will be created when you follow the instructions in later steps to add users. It should not be under your web server's document root directories though, as this might allow an attacker to download it, crack the passwords, and log in to your website.

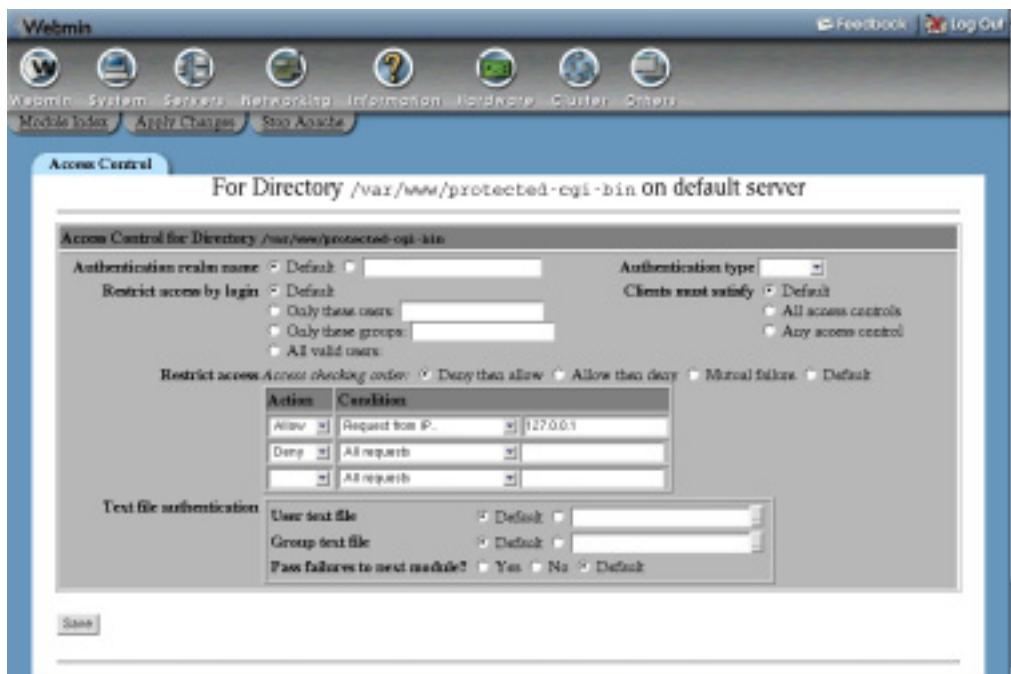


Figure 29.8 The access control form.

8. If you want to categorize users into groups for further restriction, as explained in Step 6, enter the full path to a group file into the **Group text file** field. This file must contain one line per group, in the *groupname: username1 username2 etc* format.

The file does not have to already exist because it will be created when you add groups in later steps. If you just want to set up simple username and password authentication, then this step is unnecessary.

9. Click the **Save** button at the bottom of the page, and you will be returned to the directory options page.
10. If the user and group files already exist or if you are planning to edit them manually, you can skip to Step 21. Otherwise, click on the **Access Control** icon again to redisplay the form.
11. Click on the **Edit users** link next to the **User text file** field. This will bring up a page listing all web server users currently listed in the file, if any.
12. To create a new user, click on the **Add a new user** link above or below the list.
13. On the user creation form, enter a login name into the **Username** field.
14. In the **Password** field, select the **Plain text** option and enter the user's password into the field next to it.
15. Click the **Save** button to have the user added and the list of users redisplayed.
You can edit an existing user by clicking on its name in the list, changing its details, and hitting the **Save** button. To remove a user, click the **Delete** button on the user editing form instead.
16. When you are done creating users, use the **Return to access control** link to go back to the access control form.
17. If you are using a group file as well, click on the **Edit groups** link next to the **Group text file** field to bring up a list of existing groups and their members.
18. To create a new group, click on the **Add a new group** link and fill in the **Group name** and **Members** fields on the creation form that appears, then click **Save**. Members must be entered as a space-separated list of usernames.
19. Existing groups can be edited and deleted by clicking on their names in the list, just as users can.
20. When you are done creating groups, follow the **Return to access control** link to go back to the access control form.
21. Finally, click on the **Apply Changes** link on any page to activate password protection for the directory. You can test it out by trying to visit the protected page and logging in as one of the users that you created.
22. You can add and edit users and groups in the future by editing the text files directly, or by following the relevant steps in this list. There is no need to use the **Apply Changes** link after changing the user or group lists, though, as Apache rereads the files on every request.

The instructions in the preceding list explain how to create text files for storing users and groups, but if your website is going to have a very large number of users, text files are not the best way to store them. Because Apache rereads the user file on every request, the larger it gets, the longer it will take for the web server to look up a user and generate a response. When editing or deleting a user, the entire file must be read in and written out again by the program that is changing it, which can take some time if the file is large. This increases the chance of file corruption if more than one process attempts to manipulate the same user file at the same time.

The solution is to use DBM files for storing users and groups instead. These are binary format database files that are indexed by a key (such as the username), and can be safely edited in-place. Their only down side is that they cannot be viewed or changed by UNIX programs that deal with plain text, like `cat` and `vi`.

The process of setting up authentication from DBM files is almost identical to the steps in the preceding list. The only difference is that the user and group filenames must be entered into the **User DBM file** and **Group DBM file** fields in the **DBM file authentication** box. The **User text file** and **Group text file** fields must be left set to **Default**. Unfortunately, Webmin does not allow you to edit users or groups in DBM files like you can with text files. Instead, you will need to write a Perl script or use a program like `makemap` to create them at the command line.

Apache user and password files are totally separate from the system's UNIX user list. This module, however, can be configured to add, update, or remove a user in a password file when a user with the same name is created, edited, or deleted in the Users and Groups module. This is done using that module's synchronization feature, covered in more detail in Chapter 4.

Synchronization can be useful if you want to grant access to a specific web directory to some of the UNIX users on your system and want their usernames and passwords to remain in sync if they are ever changed. To set up synchronization between an Apache text authentication file and UNIX users managed by the Users and Groups module, follow these steps:

1. On the module's main page, click on the icon for the virtual server under which the protected directory can be found, and then on the icon for the directory.
2. Click on the **Access Control** icon, then on the **Edit users** link next to the **User text file** field.
3. Below the list of users is a form for setting up synchronization for this users file. The checkboxes labeled **Add a user when a UNIX user is added**, **Change the user when a UNIX user is changed**, and **Delete the user when a UNIX user is deleted** are fairly self-explanatory. You would typically select all three, or maybe just the last two if you want to manually add new users to this file.
4. After selecting the options that you want, click the **Save** button. Any changes made in the Users and Groups module from now on will cause this user list to be updated as well.

Each Apache users text file has its own separate synchronization options. Because they are associated with the name of the file, the options will be reset to their defaults if it is renamed. Only changes made in Webmin's Users and Groups or Change Passwords modules will be synchronized with the Apache users file. If a user changes his password with the command-line `passwd` program, his web password will not be changed to match.

If you want to turn off authentication for a directory so that any browser can access it, there is no need to delete the entire **Directory Configuration** icon. Instead, you can just follow these steps:

1. On the module's main page, click on the icon for the virtual server under which the protected directory can be found, and then on the icon for the directory.
2. Click on the **Access Control** icon to go to the page shown in Figure 29.8.
3. Change the **Authentication realm name**, **Authentication type**, **Restrict access by login**, **User text file**, and **Group text file** fields all back to **Default**. If you are using DBM files instead of text, change the **User DBM file** and **Group DBM file** fields to **Default** as well.
4. Click the **Save** button and then the **Apply Changes** link back on the directory options page.

29.14 Restricting Access by Client Address

Apache can also be configured to limit access to a directory, URL location, or filename to certain client systems. The web server knows the IP address of every browser that connects to it and can use that address to determine whether or not the browser is allowed to request certain pages.

In some situations, the client's real IP address will not be available to the web server. If the client is accessing the web through a proxy server or a firewall doing NAT, then the IP address from which the request appears to originate will be that of the proxy or firewall system. There is a way to get the real address, but generally it is not a problem because all clients behind the proxy or firewall are usually treated the same from an access control point of view.

Apache determines whether a client is allowed access or not by checking its IP address and hostname against a list of rules. There are two types of rules—those that allow access and those that deny it. Depending on its configuration, the web server will either check all of the `allow` rules before the `deny` rules, or vice-versa. The first one to match determines whether or not the client is denied and no further rules are checked.

Most people who set up IP access control want to allow access from certain addresses and networks and deny everyone else. For example, you might want to give hosts on your company LAN access to your intranet, but prevent others on the Internet from accessing it. To set up this kind of access control, follow these steps:

1. On the module's main page, click on the icon for the virtual server under which you want IP access control to be enabled.
2. Click on the icon for the directory, URL location, or filename to which you want to restrict access. If one does not yet exist, follow the steps in Section 29.6 “Setting Per-Directory Options” earlier in this chapter to create it.
3. Click on the **Access Control** icon, which will bring you to the page shown in Figure 29.8.
4. Scroll down to the **Restrict access** table and change the **Access checking order** field to **Allow then deny**. This tells Apache that any request that is not specifically allowed by access control rules should be denied, and that all rules that allow access should be checked before rules that deny.

If the alternative **Deny then allow** option is chosen, requests that do not match any rule will be allowed and deny rules will be checked before allow rules.

The **Mutual failure** option has the same effect as **Allow then deny** and should not be used.

5. At first, this table will contain only one empty row for entering your first access control rule. Because you are going to allow only certain clients and block the rest, select **Allow** from the menu in the **Action** column.

The menu and field under the **Condition** column determine what kind of check is done to see if the client is allowed or not. The following are the available condition types:

All requests If chosen, all client requests will have the selected action performed.

Request from host If chosen, only clients whose hostname is the same as or ends with the text entered into the field next to it will have the action performed. Apache gets the hostname by performing a reverse DNS lookup on the client's IP address, which may not always work.

Request from IP If the client's IP address is the exactly same as the one entered into the field next to the menu, the selected action will be performed.

Request from partial IP If chosen, clients whose IP addresses start with the partial IP entered into the field next to the menu will have the selected action performed. For example, you could enter *192.168* to match all clients on that network.

Request from net/netmask If the client's IP address is within the network specified by the entered network address and netmask, the selected action will be performed. An example network specification would be *192.168.1.0/255.255.255.0*.

Request from net/CIDR If the client's IP address is within the network specified by the entered network address and prefix bits, the selected action will be performed. *192.168.1.128/25* is an example of this kind of network specification.

If variable is set If this option is chosen, the selected action will only be performed if the environment variable whose name is entered into the adjacent field is set. Apache provides several ways to set variables based on request headers and browser types and are too complex to cover here.

6. Click the **Save** button at the bottom of the form and, if there are no errors in your selections, you will be returned to the directory options page. To allow more than one client IP address or network, click on the **Access Control** icon again and fill another blank row in the **Restrict access** table. You can build up complex access control rulesets by adding many allow and deny rules.
7. When you are totally done, use the **Apply Changes** link on any page to make the restrictions active.

It is possible to combine both IP address restrictions and username/password access control for the same directory. This can be done in two ways—clients are either checked for any IP restrictions and then forced to enter a password or only prompted for a password if they do not pass the IP restrictions.

The mode that Apache uses is determined by the **Clients must satisfy** field on the access control form. If you set it to **All access controls**, then they must pass both password and IP checks. If **Any access control** is selected, however, a password will only be prompted for if the IP checks fail. This can be useful for granting access to a directory to everyone on your internal network and to people on the Internet who have a valid username and password.

29.15 Encodings, Character Sets, and Languages

As Section 29.12 “Adding and Editing MIME Types” explains, Apache attempts to determine a MIME type for every file that it sends to a browser. In addition to the type, files can also have an encoding that is usually used to indicate how they were compressed. The encoding is determined by the file extension (such as *.gz* for gzipped data) and can be used by the browser to uncompress the file before displaying it.

For example, this would allow you to create a file called *foo.html.gz* that contains compressed HTML data and is identified by the web server as such. For large files, sending them in compressed format can save bandwidth and reduce the time it takes for them to be downloaded. Unfortunately, not all browsers support the common *.gz* and *.z* encoding formats, so this feature

is not always useful. At the time this book was written, Mozilla and Netscape supported compressed encoding, but IE did not.

Encodings can be defined globally, on a per-virtual server basis, or just for a single directory or URL location. They are usually defined globally, however, and can be viewed and edited by following these steps:

1. Click on the **Default Server** icon on the Apache Webserver module's main page.
2. Click on the **MIME Types** icon, and scroll down to the **Content encodings** table. Each row in the table defines two encodings and there is always at least one pair of empty fields for adding a new one. Typically, entries for the `x-compress` and `x-gzip` encodings will already exist as they are included in the default Apache configuration.
3. To add a new encoding, enter its name into the first empty field under the **Content encoding** column. In the field next to it, enter a space-separated list of filename extensions that are used by files encoded in that format.
4. To change the name or extensions for an existing encoding, just edit its fields in the table. For example, you can add extra extensions for an encoding by just entering them into the same field as existing ones.
5. If you want to delete an encoding, just clear its entries in the fields under the **Content encoding** and **Extensions** fields.
6. When you are done editing encodings, click the **Save** button at the bottom of the page and then click the **Apply Changes** link.

Apache takes all filename extensions into account when determining a file's MIME type, encoding, language, and character set and does not care about their order. This means that files named `foo.html.gz` and `foo.gz.html` are both identified as containing gzip-compressed HTML data.

Another piece of information that Apache can supply to browsers requesting a file is the character set used by text in the file. If all your web pages are in English, or a language like Malay that does not use any non-English letters, then you don't need to care about this. If you are creating HTML pages in a different language that uses characters outside the standard ASCII character set, however, then it is useful and often necessary to indicate to browsers what character set each page is in.

Languages like German and French use special characters, like ö, that are represented by bytes above 128. Others like Chinese and Russian have so many characters that each must be represented by two bytes, using special character sets like Big5 and KOI8. For these languages, it is vital that the browser be informed of the character set of each page so that it can decode the text that it contains and use the correct font to display characters.

As with encodings, Apache determines the character set of each file by looking at its filename extension. For example, a file named `foo.html.Big5` would be identified as HTML, in which the text was encoded in the Chinese Big5 format. A file can have both a character set and an encoding, such as `foo.html.Big5.gz`, and the order in which its extensions fall does not matter.

Character sets can be defined globally or for individual virtual servers and directories. To view and edit the global list of character sets, follow these steps:

1. On the Apache Webserver module's main page, click on the **Default Server** icon.
2. Click on the **Languages** icon and scroll down to the **Extra character sets** table. Each row in the table defines two character sets, and there is always at least one pair of empty

fields for adding a new one. In the default Apache configuration, several commonly used character sets are already defined.

3. If you need to add a new character set, enter its standard ISO name into the first empty field under the **Charset** column and the filename extensions associated with it into the adjacent field under **Extensions**. Many common character sets are defined by default, so you may just be able to use one of the existing recognized extensions for your files. Multiple extensions must be separated by spaces.
4. You can change the name or extensions for existing character sets by just editing the fields in the table. It is not usually a good idea to rename the default sets because they use the standard names that are recognized by browsers. Adding extensions is perfectly safe, however.
5. To delete a character set, just clear out the fields containing its name and any associated extensions.
6. When you are done editing, click the **Save** button. If you used up all the blank fields in the **Extra character sets** table and want to add more, click on the **Languages** icon again. Otherwise, use the **Apply Changes** link to make your changes active.

Because most of the commonly used character sets are defined by default in the Apache configuration, it is not usually necessary to add new ones. Instead, you can just find the associated extensions and use them on your filenames.

Apache can also identify the language in which an HTML or text file is written by looking at its filename extensions. At first it may seem that there is no difference between a file's language and its encoding, but that is not always the case. For example, the ISO-8859-2 character set is used for many different European languages, and the Chinese language can be represented by both the Big5 and GB character sets.

Unfortunately, few browsers actually make any use of the language in which a file is written. Some can be configured to request pages in a language chosen by the user, however, and Apache can be set up to use this information to identify the correct file to return. This happens when the **Generate Multiviews** option on the directory options page is turned on for a directory.

When that option is active, a request for a page like `/documents/foo`, which does not actually exist, will cause Apache to scan the directory for `/documents` for all files starting with `foo`, identify their types and languages, and return the one that best matches the client's specified language. This is useful if you want to be able to have multiple versions of the same page in different languages, but have them all accessible via the same URL.

To view and edit the languages and file extensions recognized by Apache, follow these steps:

1. Click on the **Default Server** icon on the Apache Webserver module's main page.
2. Click on the **Languages** icon and find the **Content languages** table. Each row in the table defines two languages, and there is always at least one pair of empty fields for adding a new one. The default Apache configuration contains several commonly used languages.
3. To add a new language, enter its ISO code into the first empty field under the **Language** column and a list of extensions separated by spaces for files in that language under the **Extensions** column.

4. Existing languages can be edited by just changing their codes and extensions in the table, or deleted by clearing out their fields. It is wise not to change the standard codes for existing default languages.
5. When you are done editing languages, click the **Save** button at the bottom of the page. If you ran out of blank fields when adding new ones, click on the **Languages** icon again to return to the table. Otherwise, use the **Apply Changes** link to activate your new settings.

As with encodings and character sets, Apache does not care about the ordering of extensions in a filename when working out its type and language. Therefore, both the `foo.html.de` and `foo.de.html` files would be identified as HTML documents written in German.

29.16 Editing .htaccess Files

As explained in the introduction, Apache options can be set for a directory by creating a file in the directory named `.htaccess`. These are often created by normal users who do not have permission to edit the master web server configuration file and want to change the way Apache behaves when accessing their directories. `.htaccess` files can be used to set almost all of the options that you can configure on a per-directory basis, as explained in other sections of this chapter.

The options in one of these files apply to all the files in its directory and in any subdirectories. They can, however, be overridden by another such file lower down in the directory tree. Per-directory options in the main Apache configuration will be overridden by those in a `.htaccess` file for the same directory, but directory options for a subdirectory will override those in a parent `.htaccess` file!

Webmin can be used to create and edit `.htaccess` files, as well. If some already exist on your system that were created manually, they must be discovered by Webmin before you can use it to edit them. To have Webmin search for existing files on your system, follow these steps:

1. On the module's main page, click on the **Per-Directory Options Files** icon (this is what Webmin calls `.htaccess` files).
2. On the page that appears, there is a button labeled **Find Options Files**, with two options next to it. If **Automatically** is selected, Webmin will look in the document root directory of each virtual server for options files. If **From directory** is chosen, you can enter a directory that will be searched instead.

The latter option is useful if the websites on your system have pages that are outside of the document roots due to the use of aliases or user web directories.

3. Click the button to have the module search the select directories and any under them. The same page will be redisplayed, but with a table of all `.htaccess` files at the top, assuming some were found.

To edit the options set in a file, just click on its path from the **Per-Directory Options Files** list. This will bring up a page similar to the directory options page shown in Figure 29.5. You can click on the icons to edit redirects, username and password access control, IP address restrictions, MIME types, and custom error messages. The instructions in previous sections that apply to directories can be followed here as well. The only difference is that you do not have to use the **Apply Changes** link after making changes, as Apache always rereads the `.htaccess` files that it encounters on every request.

You can also create a new `.htaccess` file by entering the path to the directory in which it should be created into the field next to the **Create Options File** button. When the button is clicked, the file will be created empty and have its ownership set to the user and group configured on the user and group page of the default server. It will be added to Webmin's list of known options files and your browser will be redirected to the options file for the page.

To delete a per-directory options file, click on the **Delete File** link that appears at the top of the page that appears when you click on its name from the list. As soon as it is removed, Apache will cease using any options that it defines for the directory in which it resides.

Section 29.6 “Setting Per-Directory Options” earlier in this chapter explains how to set options that apply only to files of a particular name, no matter what directory they are in. It is also possible for a `.htaccess` file to contain options that apply to only some of the files in the directory that contains it. This can be useful for doing things like denying access to all files matching the pattern `*.c` in the directory `/usr/local/src`, which you cannot do just by using per-directory or per-file options.

To set options like this, follow these steps:

1. On the module's main page, click on the **Per-Directory Options Files** icon. Then, click on the `.htaccess` file in the directory to which you want the options to apply. If it doesn't yet exist, use the **Create Options File** button to create it as explained in the previous instructions.
2. Scroll down to the **Create Per-File Options** form and enter the filename or pattern into the **Path** field. Patterns can only use shell wildcard characters like `*` and `?`, unless you change the **Regexp?** field to **Match regexp**, in which case you can enter a Perl regular expression using characters like `|`, `[`, `]`, and `+`.
3. When you click the **Create** button, the same page will be redisplayed but with an additional icon for the filename or name pattern that you just entered.
4. Click on the new icon, which will bring up another page of icons for different categories of options that can be applied to files whose names match the specified filename or pattern. This page is very similar to the directory options page shown in Figure 29.5, and the pages that it links to are mostly identical.
5. The instructions in other sections of this chapter for creating redirects, custom error messages, or IP access control can be followed on this page as well to set the same options for matching files in the directory. The only difference is that there is no need to click on the **Apply Changes** link to make new settings active.

You can change the filename or pattern for which the options are saved by editing the **Path** field in the **Options apply to** form and then clicking **Save**. You can also remove them altogether so that the options for the directory apply instead by clicking on the **Delete** button in the same form.

On a system that has many virtual websites run by untrusted users, you may want to restrict the directives that those users are allowed to enter into `.htaccess` files. This can also be useful if you have user web directories enabled, which is explained in Section 29.17 “Setting Up User Web Directories”. It is possible for a user to enable CGI scripts for his directory by putting the right directives into an options file, which could pose a security risk on your server.

You can restrict the directives that can be used in `.htaccess` files on a per-directory basis. To do this, follow these steps:

1. On the main page of the Apache Webserver module, click on the icon for the virtual server under which the directory resides.
2. Click on the icon for the directory in which you want to restrict `.htaccess` files or, if one does not exist yet, follow the instructions in Section 29.6 “Setting Per-Directory Options” to create it.
3. Click on the **Document Options** icon.
4. In the **Options file can override** field, select the **Selected below** radio button. Then deselect those categories of directives in the table provided that you don’t want users to be able to include in `.htaccess` files. The available categories are:
 - Authentication options** Deselect this option to prevent the use of directives related to password authentication.
 - MIME types and encodings** Deselect this option to prevent the setting of MIME types, character sets, encodings, and languages for files. This will also stop files with certain extensions being indicated as CGI programs.
 - Indexing and index files** This option controls the use of directives for directory indexing.
 - Hostname access control** Deselect this option to stop the use of IP access control directives.
 - Directory options** This option controls the use of directives that set options for the directory, such as whether indexing is done and if CGI programs are enabled.
5. Click the **Save** button and then the **Apply Changes** link.

Whenever a user tries to use directives that he is not allowed to use, Apache will display an error message when files in the directory containing the `.htaccess` file are requested. It will not simply ignore the disallowed directives.

29.17 Setting Up User Web Directories

On a system with many UNIX users, you may want to allow each user to create his own set of web pages. Instead of creating a subdirectory for each user under some document root directory, you can instead designate a subdirectory in each user’s home directory as a location for web page files. Typically, this subdirectory is called `public_html` and its contents are made available at a URL like `http://www.example.com/~username/`.

The special `~username` path in the URL is converted by Apache to a directory under the home of the user named `username`, no matter what document root directory is being used for the rest of the files on the website. It is also possible for files in the user’s actual home directory to be made available instead, so that `~username` actually maps to the user’s home directory and not a subdirectory. This is a bad idea, however, as it makes all of the user’s files available to anyone with access to the website.

To turn on Apache's user web directories feature so `~username` URL paths can be used, follow these steps:

1. On the module's main page, click on the icon for the virtual server for which you want to activate user directories. To activate them for all virtual servers, click on the **Default Server** icon instead.
2. Click on the **Document Options** icon.
3. In the **User WWW directory** field, deselect the **Default** option and enter `public_html` into the field next to it. Or, if you want a different subdirectory to be used for users' web pages, enter its name instead. To make users' entire home directories available via `~username` URL paths, enter `.` into the field.
On many systems, this option will already be set to `public_html` in the default Apache configuration, meaning that user web directories are already enabled.
4. If the **All users accessible** option is selected, Apache will allow the pages in any user's web directory to be accessed.

To configure the web server to only allow access to the pages belonging to certain users, select the **Only users** option and enter the names (separated by spaces) into the field next to it. This can be useful if there is a small fixed list of UNIX users who should be allowed to publish web pages.

To block only a few users' web pages and allow the rest, select the **All users except** option and enter the names of the blocked users into its field. This is useful for protecting files belonging to important system users such as `root`.

5. Click the **Save** button at the bottom of the page, then use the **Apply Changes** link to activate the new settings. Try creating a `public_html` subdirectory in the home directory of a user, putting some HTML files in it, and seeing if they can be accessed using the `~username/filename.html` URL path.
6. It is also possible to have `~username` URL paths mapping to directories outside users' home directories by entering values starting with `/` into the **User WWW directory** field. For example, if you were to enter `/www` and a browser requested `~jcameron/foo.html`, then the file returned by Apache would be `/www/jcameron/foo.html`. If you entered `/home/*public_html`, then the file returned would be `/home/jcameron/public_html/foo.html`, even if the user `jcameron` did not have his home directory at `/home/jcameron`. As that example shows, any occurrence of a `*` in the user web directory is replaced by the username.

Similarly, you can enter a URL into the directory field, which will be used by Apache to generate a URL to redirect browsers to when a user web directory is requested. For example, if you enter `http://home.example.com/users/` and the URL path `~jcameron/foo.html` is requested by a browser, it will be redirected to `http://home.example.com/users/jcameron/foo.html` instead. This is useful if you want to move user web directory hosting to a separate server, while allowing URLs on your main server to be used to access them.

Even though the above are sufficient to enable user web directories, there are some other things that you might want to do. As the earlier Section 29.16 "Editing .htaccess Files" explains, you may want to limit the kinds of directives that users can put in their `.htaccess` files so that

they cannot execute CGI programs or use server-side includes. You can also change the default directory indexing and document options that apply to user web directories. To accomplish both of these tasks, follow these steps:

1. On the module's main page, click on the icon for the virtual server in which user web directories were enabled, or the default server.
2. Assuming all your users have their home directories under /home and the web subdirectory is named `public_html`, enter `/home/*/public_html` into the **Path** field of the create per-directory, files, or location options form at the bottom of the page.
3. Leave the **Type** field set to **Directory** and the **Regexp?** field to **Exact match**.
4. Click the **Create** button to create a new set of options that will apply to users' web directories, then click on its newly created icon. This will bring up the document options page shown in Figure 29.5.
5. Click on the **Document Options** icon.
6. Change the **Directory options** field to **Selected below** and set to **Yes** those options that you want to apply to user web directories. It is advisable to turn on **Generate directory indexes** and safe to enable **Server-side includes**, but not **Execute CGI programs** or **Server-side includes and execs**.
The **Follow symbolic links** option is relatively safe to turn on as well, but will allow users to make available via the web files that are not in their `public_html` subdirectory by creating links to them.
7. To prevent users from overriding these settings in `.htaccess` files, change the **Options file can override** field to **Selected below** and deselect the **MIME types and encodings** and **Directory options** checkboxes. The others control options that present no security risk and so can be safely left selected.
8. Click the **Save** button and then the **Apply Changes** link to save and activate the restrictions.
9. If you want to turn on server-side includes, set some custom MIME types or IP access controls for user web directories, you can do it by following the instructions in the appropriate sections for this directory. Because server-side includes are quite harmless with the ability to execute external programs disabled, they can be safely enabled for users by setting the right content handler for `.html` or `.shtml` files, as Section 29.9 “Setting Up Server-Side Includes” explains.

29.18 Configuring Apache as a Proxy Server

An HTTP proxy is a server that accepts requests for web pages from browsers, retrieves the requested pages from their servers, and returns them to the browser. They are often used on networks on which clients are not allowed to connect to web servers directly so that restrictions on who can access the web and what sites they can view can be enforced. A proxy can also cache commonly accessed pages, so if many clients visit the same site its pages only have to be downloaded once. This speeds up web access and reduces bandwidth utilization.

Apache is not the best proxy server available for UNIX systems—Squid (covered in Chapter 44) takes that honor. Squid has many more configurable options, is more efficient, and can deal with much larger caches. If you want to set up a proxy on a system that is already running

Apache, however, then it may make sense to use the existing web server as a proxy instead of installing and running a separate server process for Squid.

Apache's proxy support is only available if the `mod_proxy` module has been compiled into the web server or is available to be dynamically loaded. You can see if the module is available by clicking on the **Re-Configure Known Modules** icon on the main page. If `mod_proxy` is checked, then your server can be used as a proxy. If so, you can skip the next paragraph, which deals with loading the proxy module.

On some Linux distributions, the proxy module is included with the Apache package but not loaded by default. If this is the case on your system, you can enable it by following these steps:

1. On the Apache Webserver module's main page, click on the **Edit Config Files** icon. This will bring up a page showing the contents of the primary configuration file, called `httpd.conf`.
2. Look for a line starting with `LoadModule proxy_module`, which is currently commented out with a # at the start. If no such line exists, then the proxy module is probably not installed at all and therefore cannot be used.
3. Delete the # at the start of the line and then click the **Save** button at the bottom of the page.
4. Click the **Stop Apache** link on any page to shut down Apache and then the **Start Apache** link to start it again. This is necessary for the web server to load the enabled proxy module.
5. On the module's main page, click on the **Re-Configure Known Modules** icon and then on the **Save** button at the bottom of its page. This tells Webmin to reanalyze the Apache configuration so that it detects that the `mod_proxy` module is now available.

If Apache was compiled on your system from source, then you will need to recompile it with `mod_proxy` enabled in order to use the proxy features. If you do, Webmin will detect that a new version of the Apache server executable has been installed and will redisplay the form shown in Figure 29.1 when you next visit the module's main page. The proxy module will be automatically selected, so you should be able to just click the **Configure** button to tell Webmin that proxy features are now available.

Once `mod_proxy` has been enabled, you can set your system up as a proxy server by following these steps:

1. On the module's main page, click on the icon for the virtual server that you want to use as a proxy. This must be an IP-based virtual server or the default, as it is impossible to turn on proxying for just a single name-based virtual server. The normal operation of whichever server you choose, however, will not be affected.
2. Click on the **Proxying** icon that should be visible on the virtual server options page. If the icon does not exist, then the proxy module has not been detected by Webmin.
3. Change the **Act as proxy server?** field to **Yes**.
4. By default, Apache will not cache any pages that are requested through it when acting as a proxy server. To change this, create a directory that is writeable by the web server user (usually `httpd`) and enter it into the **Cache directory** field.
5. To limit the amount of data that will be cached, enter a number of kilobytes into the **Cache size** field. If this is left set to **Default**, Apache will only cache 5 Kb of pages.

6. To turn off caching for particular websites, enter a space-separated list of hostnames and domains into the **Domains not to cache** field. This can be useful for avoiding the caching of sites that frequently change.
7. To stop users of the proxy from accessing certain domains, enter a space-separated list of full or partial hostnames into the **Block requests to domains** field. For example, to deny access to all sites in the `foo.com` domain you could just enter `foo.com`.
8. If you have another proxy server on your network and want to pass all requests on to that proxy, enter its URL (like `http://proxy.example.com:8080/`) into the empty field under **Forward to** in the **Requests to pass to another proxy** table and leave the **All** option selected. You can also have just a few requests passed on by selecting the **Matching** option and entering a partial URL or URL type (like `http://www.foo.com/` or `ftp`) into the field next to it.

Like other tables in the Apache module, this one only displays one blank row at a time. If you want to set up several other proxies to which to forward different requests, you will need to re-edit this page after saving and fill in the next blank row. For example, you might want to forward all FTP requests to one proxy, but all other types of requests to another.

9. To exclude some requests from being passed to the other proxies (for example, if they are on your local network), you can fill in the **Don't pass requests to another proxy for** table. In each empty row you can choose from one of the following types:

IP address If this type is chosen, you must enter a complete IP address into the field next to it. Any requests to the web server with this IP will not be passed on to another proxy.

Hostname When this type is chosen, any requests to the web server whose hostname is entered into the adjacent field will not be passed on.

Domain Any requests to websites in the domain entered into the field next to the menu will be retrieved directly and not passed on.

IP network Any requests to websites in the specified IP network (entered as a partial IP address, like `192.168`) will not be passed on to another proxy.

Network/bits Any requests to websites in the IP network (entered in address/prefix, like `192.168.1.0/24` format into the adjacent field) will not be passed on.

To add more than one row, you will need to save the form and edit it again so that a new blank row is displayed.

10. Most of the other options on the form control the layout of the cache directory and the amount of time for which pages are cached. In most cases, the defaults will work fine so you can just leave them set to **Default**.
11. When done, click the **Save** button to update the Apache configuration file with the proxy settings, then the **Apply Changes** link to make them active.

You should now be able to try your settings by configuring a web browser to use your Apache server as a proxy and visiting some web pages. All proxy requests that Apache processes will be written to the access log file for the virtual server in the usual format, but with the full URL recorded instead of just the page.

You may sometimes want to limit who has access to proxy, either by client IP address or by username and password. This can be done by following the instructions in Section 29.14 “Restrict-

ing Access by Client Address” and Section 29.13 “Password Protecting a Directory” and substituting the special directory `proxy:*`. If you set up client address access control, then only hosts with allowed addresses will be able to use your server as a proxy. They will, however, still be able to access normal web pages, as IP address restrictions for the special `proxy:*` directory only apply to proxy requests.

If you set up username and password authentication for your proxy server, then any web browsers that attempt to use it will be forced to log in first. This login is to the proxy server, not to any website that is being accessed through it. If a user visits a password-protected website using the proxy, he will have to log in separately to that site.

It is also possible to set up IP or password restrictions that apply only to some protocols or sites accessed through the proxy, by creating them for special directories like `proxy:http` or `proxy:http://www.example.com/`. Only requests for URLs that start with the text after `proxy:` will be effected by restrictions like these. They can be useful for blocking or limiting access to certain sites or preventing the proxy from being used to request certain protocols like `http` or `ftp`.

29.19 Setting Up SSL

SSL is a protocol for making secure, authenticated connections across an insecure network like the Internet. It encrypts network traffic so an attacker cannot listen in on the network and capture sensitive information such as passwords and credit card numbers. It allows servers to authenticate themselves to clients, so that a web browser can be sure that it is connecting to the website that it thinks it is. It also allows clients to authenticate themselves to servers, which can be used to replace usernames and passwords with digital certificates.

The SSL protocol can be used to encrypt any kind of data that would normally travel over an unencrypted TCP connection. In this chapter, however, we are only concerned with the encryption of web page requests and responses, which is done by encrypting HTTP protocol data with SSL. The result is a new protocol called HTTPS, which is used by all websites that want to operate securely. Almost every browser supports the HTTPS protocol and uses it when retrieving URLs that start with `https://` instead of the normal `http://`. Whereas the normal HTTP protocol uses TCP port 80, the HTTPS protocol uses port 443.

You can configure Apache to use HTTPS on a per-virtual server basis or to use it for all servers. This depends, however, on having the `mod_ssl` Apache module compiled in or available for dynamic loading, which is not always the case. Section 29.18 “Configuring Apache as a Proxy Server” explains how to check for and possibly enable the `mod_proxy` module and you can follow those same instructions for `mod_ssl`, as well. Most modern Linux distributions include SSL support in their Apache package as standard.

At the heart of the SSL protocol are digital certificates, which are used for both authentication and encryption. The server typically sends its certificate to the client to prove its identity so the client knows that its connection to the website has not been redirected by an attacker. Certificates issued by a proper certificate authority, such as Verisign or Thawte, are impossible to forge because they have been signed by the authority’s master certificate. All web browsers include a list of authorities that they can use to validate signatures and thus ensure the authenticity of website certificates.

The down side of this method of certificate validation is that you cannot simply generate your own certificate for your website that will be accepted without complaint by web browsers. It is possible to create a self-signed certificate that Apache will happily use, but any browser connecting to

that web server in SSL mode will display a warning message to the user because the certificate is not signed by a recognized authority. Self-signed certificates are fine for encrypting HTTPS traffic, but if you want browsers to be able to validate your site you will need a *real* certificate signed by a proper authority—and that costs money.

Before you can enable SSL in Apache, you must have a certificate. The easiest way to get one for testing purposes is to generate your own self-signed certificate, which can be done by following the upcoming instructions. To generate a real certificate from a recognized authority, follow the steps at the end of this section, instead. To create a certificate, you will need the `openssl` command, which is included with most modern Linux distributions and freely available for download from www.openssl.org/. If Apache on your system already includes the `mod_ssl` module, then `openssl` is probably already installed or on your distribution CD or website.

To generate your own self-signed certificate, use the following steps:

1. Log in to your system as `root`.
2. Change to the directory in which you want to store your certificate files, such as `/usr/local/apache/conf` or `/etc/httpd`.
3. Run the command `openssl req -newkey rsa:512 -x509 -nodes -out cert.pem -keyout key.pem`.
4. The command will ask the following questions in order to obtain attributes for your new key. To leave any of the requested fields blank, just enter a single period.

Country name The two-letter code for the country in which your web server is located, such as *AU* or *US*.

State or Province Name The name of the state in which your server is located, such as *California*.

Locality Name The city in which your server is located, such as *San Francisco*.

Organization Name The name of your company or organization, such as *Example Corporation*.

Organizational Unit Name The name of your division within the company, such as *Engineering*.

Common Name The hostname of your web server as used in the URL. For example, if browsers usually access the server as `http://www.example.com/`, then you should enter `www.example.com` for this question. Unfortunately, you can only enter a single hostname, so if your web server is accessed by more than one name (such as `www.example.com` and `example.com`), then only one will match the certificate. The hostname, however, can contain the wildcard character `*`, so you can enter `*.example.com` or even just `*`.

Email Address The email address of the administrator for this web server, such as `jcameron@example.com`.

5. When all the questions have been answered, the files `cert.pem` and `key.pem` will be created in the current directory. These are your website's certificate and its private key, respectively.
6. Because the private key **must** be kept secure to ensure the security of SSL connections to your server, change its ownership to the user as whom Apache runs, with a command

like `chown httpd key.pem`. Then, set the permissions with the command `chmod 600 key.pem` so no other user can read it.

Now that a certificate and private key have been created, you are ready to configure your web server to use SSL. The best way to do this is to create a new virtual server that handles all requests to port 443 (the HTTPS port) in SSL mode. This way, any existing virtual servers on your system will not be affected. To do this, follow these steps precisely:

1. On the main page of the Apache Webserver module, click on the **Networking and Addresses** icon.
2. In the blank row at the end of the **Listen on addresses and ports** table, select **All** under the **Address** column and enter **443** under the **Port** column. Then click the **Save** button at the bottom of the page.
3. Back on the main page, scroll down to the create a new virtual server form.
4. Set the **Address** field to **Any** and the **Port** field to **443**.
5. If you want the pages that browsers see when connecting in SSL mode to be the same as those that they see when making a normal HTTP connection, enter the document root directory for your default server into the **Document Root** field. Otherwise, you can enter a different directory so clients will see different pages when making HTTPS requests.
6. In the **Server Name** field, enter the same hostname that you specified for the **Common Name** when creating the SSL certificate.
7. Click the **Create** button to have the new virtual server added to your Apache configuration. An icon for it will be added to the module's main page.
8. Click on the icon for your new server to go to the virtual server options page. An icon labelled **SSL Options** should be visible. If not, either your Apache Web server does not have the `mod_ssl` module or Webmin hasn't detected it yet.
9. Click on the **SSL Options** icon to bring up the page shown in Figure 29.9.
10. Change the **Enable SSL?** field to **Yes**. This tells Apache that the virtual server should treat all connections as HTTPS.
11. In the **Certificate/private key file** field, deselect **Default** and enter the full path to the `cert.pem` file that you created earlier.
12. In the **Private key file** field, enter the full path to the `key.pem` file. If you only have a single file that contains both the certificate and private key, you can leave this field set to **Default** and enter its path into the field above it.
13. Click the **Save** button and then the **Apply Changes** link back on the virtual server options page.
14. Unless an error is reported when applying the configuration, your web server should now be running in SSL mode on port 443. Test it out by using a web browser to go to `https://www.example.com/` or whatever the URL of your site is. Note that there is no need to specify port 443 in the URL as it is used by default for HTTPS—just like port 80 is the default for HTTP.

It is also possible to create IP-based virtual servers that use SSL and handle connections to port 443 on particular IP addresses. It is not, however, possible to create several name-based virtual servers that use SSL because the server sends its certificate to the client before any HTTP protocol data is exchanged. Normally, the `Host: HTTP` header is used by Apache to determine to

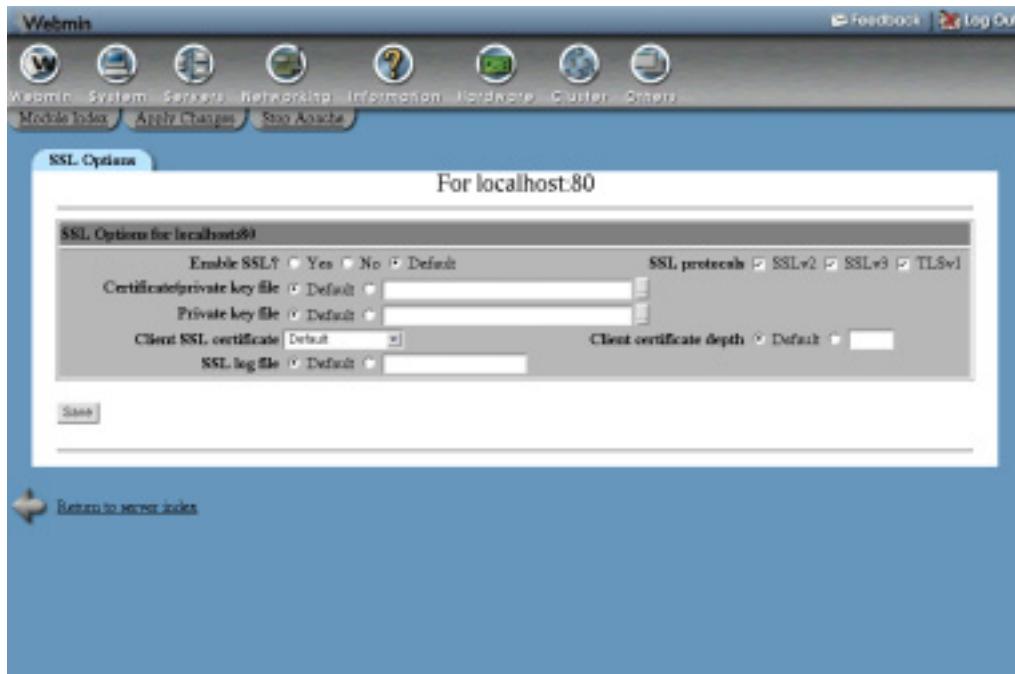


Figure 29.9 The SSL options page.

which name-based virtual server a request is being made, but this header has not been sent by the browser at the time the web server selects the certificate to send to the client. The end result is that having multiple named-based virtual servers on the same IP address in SSL mode will not work properly, if at all.

On some Linux distributions, the included Apache package may already include an example virtual server running on port 443 with SSL enabled. It will probably also come with usable certificate and private key files, although they are likely to be self-signed and to have a different host-name for the common name. In this case, there is no need to follow the steps above to set it up—all you need to do is generate your own SSL certificate files and then visit the SSL options page in the existing virtual server to change the **Certificate/private key file** and **Private key file** fields.

If you want to use Apache to host a real Internet website running in SSL mode, you will need to request a certificate signed by a recognized authority. To do this, you must generate a certificate signing request (CSR) and send it to the authority for verification along with your website's name, company name, and other details to prove that you really do own the website and domain. After they have verified your details, the CA will sign the certificate and send it back to you for use in your web server.

To generating a CSR, follow these steps:

1. Log in to your system as `root`.
2. Change to the directory in which you want to store your certificate files, such as `/usr/local/apache/conf` or `/etc/httpd`.

3. Run the command `openssl genrsa -out key.pem 1024`. This will create just the private key file `key.pem`.
4. Make sure that the file can only be read by the web server user, with commands like `chown httpd key.pem` and `chmod 600 key.pem`.
5. Run the command `openssl req -new -key key.pem -out csr.pem` to generate the CSR.
6. The command will ask the following question to obtain attributes for your new key. To leave any of the requested fields blank, just enter a single period.

Country name The two-letter code for the country in which your web server is located, such as *AU* or *US*.

State or Province Name The name of the state in which your server is located, such as *California*.

Locality Name The city in which your server is located, such as *San Francisco*.

Organization Name

The name of your company or organization, such as *Example Corporation*.

Organizational Unit Name The name of your division within the company, such as *Engineering*.

Common Name The hostname of your web server as used in the URL. For example, if browsers usually access the server as `http://www.example.com/`, then you should enter `www.example.com` for this question. Wildcards cannot generally be used in the hostname of certificates signed by CAs.

Email Address The email address of the administrator for this web server, such as `jcameron@example.com`.

7. When all the questions have been answered, the `csr.pem` file will be created in the current directory. This is your certificate signing request, which should be sent to the certificate authority for signing.
8. After your details have been verified and your money taken, the authority will send you back a signed certificate. It should be a text file that starts with the line

-----BEGIN CERTIFICATE-----

Put it in the same directory as the private key, in a file named `cert.pem`.

If you have overwritten existing self-signed private key and certificate files, it is best to stop and restart Apache to force the new ones to be used. You should now be able to connect to your web server in SSL mode with no warning displayed in the browser.

29.20 Viewing and Editing Directives

The Apache Webserver module can be used to view and edit directives manually, instead of the usual method of editing them through the module's forms and pages. Manual editing is only recommended if you are familiar with the configuration file format, as no checking will be done to make sure that you have entered valid directives or parameters. It is often faster, however, to configure the web server in this way, especially if you are an experienced Apache administrator.

On the options page for every virtual server, directory, URL location, filename, and .htaccess file there is an icon labeled **Show Directives**. When clicked on, it will display all of the directives inside that virtual server or directory, as shown in Figure 29.10. Any directive that the module knows how to edit will be linked to the appropriate form for editing, which will be one of those that can be reached by clicking on another icon on the virtual server or directory's options page. Next to each directive is the name of the file in which it is located and the line number in that file, so that you can use another program like vi or emacs to edit it manually, if you wish.

Below the list are two buttons labeled **Manually edit directives** and **Edit Apache directive**. The first will take you to the editing form described in the next paragraph. The second will bring you to the form for editing the directive selected from the menu next to it, which will be one of those linked from an icon on the previous page. This can be useful if you know the name of the Apache directive that you want to use but do not know where in Webmin it can be edited.

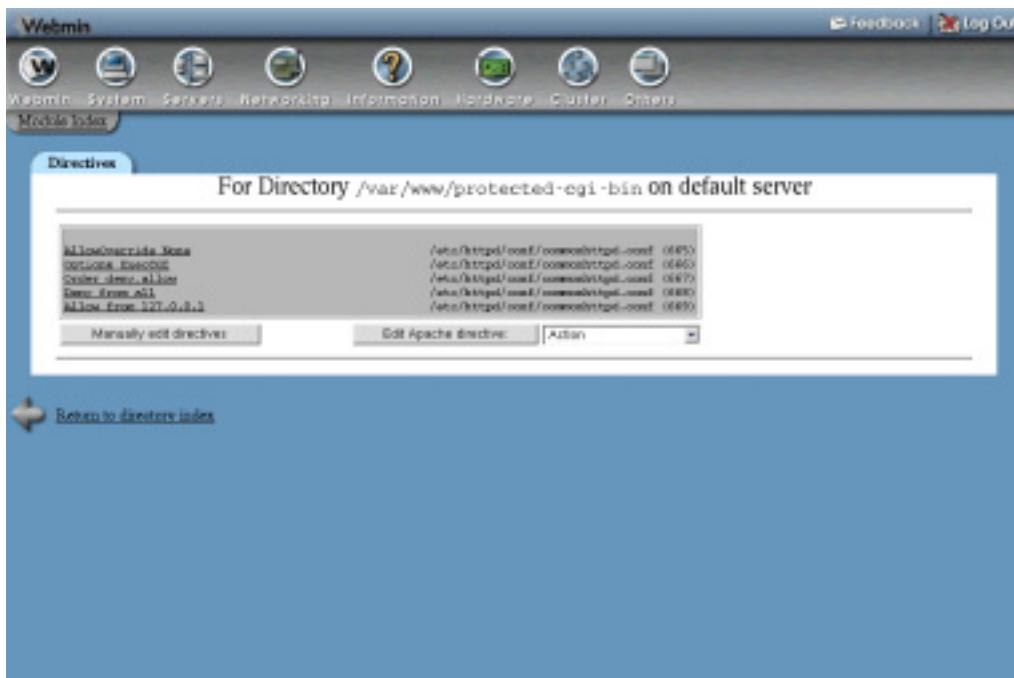


Figure 29.10 Viewing directives for a directory.

To directly edit the text of directives in a virtual server or directory, you can click on the **Edit Directives** icon located next to **Show Directives** on every options page. This will display a text box containing the exact text that appears in the Apache configuration file for that server or directory, including any comments and indentation. When the **Save** button is clicked, any changes that you have made will be written back to the file without any verification. To make them active, you will need to click on the **Apply Changes** link on any of the module's pages.

It is also possible to edit entire an Apache configuration file at once using the **Edit Config Files** icon on the module's main page. When selected, the complete contents of the primary config-

uration file (usually `httpd.conf`) will be displayed in a text box. Above it is a menu for selecting another file to edit and a button labeled **Edit Directives in File** that will switch to the contents of the chosen file. Your Apache Web server may use several different files that Webmin normally hides from you. Only on this page can you see that all files that the module has detected are being used, either by default (such as `httpd.conf`, `srm.conf`, or `access.conf`) or through `Include` directives in the default configuration files.

This page is the only place where you can view and manually edit directives that apply to all virtual servers that are normally editable under the **Default Server** icon in the module. Because these default directives are usually split across multiple files, no **Show Directives** or **Edit Directives** icons appear on the options page for the default server.

If you change any of the directives in the text box, click the **Save** button below it to have the configuration file rewritten. No validation will be done, so be careful with your changes—a mistake with a container directive like `<Directory>` or `</IfModule>` may make it impossible for Webmin to parse some or all of the file. As usual, you will need to click on the **Apply Changes** link back on the module’s main page to make the changes active.

29.21 Module Access Control

As Chapter 52 explains, you can use the Webmin Users module to give a user limited access to some modules. In the case of the Apache Webserver module, a Webmin user or group can be restricted so that he can only edit a subset of the available virtual servers. This can be very useful in a virtual hosting environment in which you want to give people the right to edit the settings for their own servers, but not those belonging to everyone else.

It is also possible to restrict the pages in the module that the user is allowed to edit, as some allow the setting of directives that could be used to subvert security on your system. For example, you would not want a user to be able to change the user and group as whom the CGI programs on his virtual server run.

To set up the Apache module for a user so that he can only edit a few virtual servers, follow these steps:

1. In the Webmin Users module, click on **Apache Web server** next to the name of a user who has been granted access to the module.
2. Change the **Can edit module configuration?** field to **No** so that he cannot change the paths that the module uses for the web server configuration files.
3. For the **Virtual servers this user can edit** field, choose the **Selected** option and select those servers that he should be allowed to manage from the list provided. It is generally a bad idea to allow an untrusted user to edit the default server, as its configuration effects all other virtual servers.
4. Change the **Can edit global options?** field to **No** so that he cannot change settings like the ports and addresses that Apache listens on.
5. Change the **Can create virtual servers?** field to **No** so that he is not allowed to add new virtual hosts.
6. To stop him from changing the user and group as which CGI programs are run, set the **Can change virtual server users?** field to **No**. This only really matters if you have `suexec` installed, as explained in Section 29.8 “Running CGI Programs”.

7. Unless you want him to be able to change the address and port on which the virtual server accepts requests, set the **Can change virtual server addresses?** field to **No**. If they are changed, they could interfere with other virtual servers.
8. If the **Can pipe logs to programs?** field is set to **Yes**, he will be able to configure the virtual server to log to a command that will be run as the user as whom Apache normally runs (usually `httpd`). This may be a security risk on your system, so it is usually a good idea to set this field to **No**.
9. Change the **Can start and stop Apache?** field to **No**. He will be able to apply changes but not shut down the entire web server.
10. The **Limit files to directory** field controls where he can configure the server to write its log files to. Allowing them to be written anywhere may allow him to overwrite files, so it is best to set this to something under his home or document root directory, such as `/home/jcameron/logs`.
11. The **Directive types available** field determines what icons appear in the virtual server options page, and therefore what kinds of directives he is allowed to edit. If you choose **All**, then all of the icons will be visible, along with the **Show Directives** and **Edit Directives** icons for manually editing the configuration files. If you choose **Selected** instead, only those pages chosen from the list provided will be visible and the manual editing icons will not.
It is usually a good idea to deny access to the user and group and log files pages, and always good to prevent inexperienced users from editing the configuration files manually. An error in the `httpd.conf` file might cause the entire web server to stop working the next time it is restarted.
12. Finally, click the **Save** button at the bottom of the page. The restrictions will be applied to the user or group immediately.

You should be aware that these restrictions will not stop a truly malicious user causing problems with your Apache configuration. It is quite possible to use the forms to introduce intentional syntax errors into the configuration files which could interfere with the proper working of the web server. Fortunately, you can always track who has done what using the Webmin Actions Log module, covered in Chapter 54.

29.22 Configuring the Apache Webserver Module

Like other modules, this one has several options that can be changed by clicking on the **Module Config** link on the main page. Those listed under **System configuration** are related to the locations of the Apache configuration files and programs on your system, which, by default, will be set to match the Apache package that comes with your operating system. If your version of UNIX does not come with Apache, then the module will assume that the web server has been compiled and installed from the source distribution and will use the paths to files that the source install does.

Table 29.1 lists both the configurable options that you can safely change (in the first section), and those that are related to file locations that generally do not need to be edited (in the second).

Table 29.1 Module Configuration Options

Display virtual servers as	When this field is set to Icons , all virtual servers on the module's main page will be shown as icons. When changed to List , however, they will be shown in a table instead. This latter option makes sense when you have lots of virtual servers, as it uses up less space on the page.
Order virtual servers by	<p>This field controls the ordering of virtual servers on the main page, but regardless of which option is selected, the default server will always appear first. The available choices and their meanings are:</p> <ul style="list-style-type: none"> Order in config files Servers appear in the same order that they are listed in the Apache configuration file, which will typically be the order in which they were added. Server name Virtual servers are ordered by their primary hostname. IP address Servers are ordered by the IP address on which they listen. This option only really makes sense if you have a large number of IP-based virtual hosts.
Maximum number of servers to display	If the number of virtual servers exceeds the number set in this field, then the module's main page will display a search form instead. On a system with thousands of servers, this keeps the size of the page manageable; however, you can change it if the default maximum is too low or high for your liking.
File to add virtual servers to	<p>By default, all new virtual servers will be added to the primary Apache configuration file, usually <code>httpd.conf</code>. If, however, you put all servers into a separate file on your system, then this field should be set to the full path to that file. Naturally, Apache must be configured to read the specified file, such as by an <code>Include</code> directive in the primary configuration file.</p> <p>This field only controls the default setting for the Add virtual server to file field on the virtual server creation form, covered in Section 29.5 “Creating a New Virtual Host”. You can still choose to add a server to a different file when filling out the form, if you wish.</p>
Test config file before applying changes?	<p>When this field is set to Yes, the module will use the <code>apachectl configtest</code> command to test your Apache configuration before actually applying it when the Apply Changes link is clicked. This is turned on by default so that in the event that your configuration is invalid it will be detected before Apache is signaled to reread it, which would cause the web server to stop.</p> <p>If for some reason the testing always fails on your system, you may need to set this option to No instead. This will have no negative effect as long as your configuration files never have any syntax errors in them.</p>
Test config file after manual changes?	<p>When this option is enabled, the module will test the Apache configuration after any manual changes and revert to the old settings if an error is detected.</p> <p>If you are not too familiar with the configuration file format, this option can protect you from mistakes.</p>

Table 29.1 Module Configuration Options (Continued)

Test config file after other changes?	This option tells the module to test the Apache configuration after any form is submitted, to ensure that incorrect input has not caused the configuration files to somehow become invalid. It is most useful on systems that have untrusted or inexperienced users managing their own Apache virtual hosts.
Show Apache directive names	<p>By default, this field is set to No. If you change it to Yes, however, then the name of the Apache directive that each field sets will be shown next to its label on every page in the module that allows you to edit directives. In addition, the name will be a link to the Apache documentation for that directive at the location set in the Base directory for Apache documentation module configuration field.</p> <p>This option can be useful for finding out more about what each field actually does, and for learning the directives actually used in the Apache configuration. Some fields and tables in the module's forms actually set more than one type of directive. In cases like these, all of the directives will be listed, each with its own link to its documentation.</p>
Base directory for Apache documentation	<p>This field controls where the linked to Apache documentation is located when the Show Apache directive names option is enabled. When Apache website is selected, the links will be to the official documentation pages at http://httpd.apache.org. This will work fine, but may not be the best choice if you have a local copy of the documentation or do not have Internet access.</p> <p>The alternative is to enter a base URL under which the Apache module HTML files can be found. If, for example, you entered http://www.example.com/docs/, then Webmin would generate links like http://www.example.com/docs/mod_cgi.html#scriptlog.</p>
Apache server root directory	<p>This is the directory under which all the Apache configuration files are located. If you installed from source code, then it should be set to <code>/usr/local/apache</code>. Webmin uses this directory to find configuration files like <code>httpd.conf</code> if they are not set explicitly in some of the fields below.</p>
Path to httpd executable	<p>In this field, you must enter the full path to the <code>httpd</code> server program, such as <code>/usr/local/apache/sbin/httpd</code>. Webmin executes this program in order to find out which version of Apache you are running and which modules are compiled into it.</p>
Apache version	<p>Normally this field is set to Work out automatically, which causes Webmin to run <code>httpd</code> with the <code>-v</code> flag to get the Apache version number. There are some special modified versions of the web server, however, that do not report the version in the format that the module expects, which will cause an error message to be displayed when you try to use the module.</p> <p>If this happens on your system, you may need to enter the correct version number into this field. It must be entered in the format that Apache version numbers are normally in, such as <code>1.3.19</code> or <code>2.0.43</code>.</p>

Table 29.1 Module Configuration Options (Continued)

Path to the apachectl command	This field must contain the full path to the command <code>apachectl</code> , such as <code>/usr/local/apache/bin/apachectl</code> . Webmin uses it to start and stop Apache, test the configuration, and apply changes. The older releases of the web server did not include this command, so if you do not have it on your system you can select the None option instead. This will cause Webmin to use TERM and HUP signals to stop the server and apply changes, and run the <code>httpd</code> command directly to start it or test the configuration.
Command to start apache	If this field is set to something other than Automatic , the module will run the specified command when the Start Apache link is clicked. Many operating systems that include an Apache package have boot scripts that can be used to start and stop the web server. On systems like those, this field will be set to something like <code>/etc/init.d/apache start</code> so that the boot script is used. If Automatic is chosen instead, Webmin will use the <code>apachectl</code> command to start Apache, or run the <code>httpd</code> server program directly. If you have compiled and installed Apache from the source code, it is best to select the Automatic option so that the module will not attempt to use a boot script that does not exist or will not work.
Command to stop apache	This field is similar to the previous one, but determines the command used to shut down the web server when the Stop Apache link is clicked on. As with the start command, you should set this to Automatic if you have installed Apache from source code.
Path to httpd.conf	These three fields control where Webmin looks for the <code>httpd.conf</code> , <code>srm.conf</code> , and <code>access.conf</code> configuration files, respectively. When set to Automatic , the module will look in the <code>etc</code> and <code>conf</code> subdirectories under the server root directory, which is the correct location most of the time. If the files are located elsewhere (as they are by default in some Linux distributions), however, then these fields need to be set to the correct full paths to the configuration files.
Path to mime.types	Like the fields above, this one controls where the module looks for the <code>mime.types</code> file that contains the global list of MIME types and extensions. If <code>TypesConfig</code> directive exists in one of the configuration files, then the path that is specified will be used. If a path is entered for this field, then it will be used. Otherwise, if the field is set to Automatic , then Webmin will attempt to locate the <code>mime.types</code> file in the <code>etc</code> and <code>conf</code> subdirectories under the server root.

29.23 Summary

This chapter has introduced the Apache Web server, and explained how Webmin can be used to configure it to perform various common tasks. After reading the chapter you should know what virtual hosts are and how to create them, how to set different options for different directories, how to redirect requests, how to set up authentication, and much more. Because Webmin's Apache module has a vast number of configurable settings, only those related to common operations are covered.

DNS Server Configuration

In this chapter the DNS protocol and the BIND DNS server are explained, as is the Webmin module for creating and managing DNS domains.

30.1 Introduction to the Domain Name System

DNS is a protocol used primarily for converting hostnames like *www.example.com* into IP addresses like *192.168.1.10*, and vice-versa. At the IP level, all hosts on the Internet refer to each other by IP addresses, not by the hostnames that users enter into programs like web browsers and telnet clients. This means that a system needs a way of finding out the IP address associated with a hostname before they can communicate. Although there are several ways this can be done (such as reading the */etc/hosts* file or querying an NIS server), DNS is the most common.

As well as looking up IP addresses for hostnames, the DNS protocol can also be used to find the hostname associated with an IP address. This is most often used for finding the hostname of a client that is connecting to a server, such as a web server or SSH daemon. DNS can also be used to look up the address of a mail server for a domain and additional information about a host such as its location, operating system or owner. However, by far its most common application is converting hostnames to IP addresses.

Most systems use the DNS protocol to send requests to a server, which does most of the work of resolving a hostname into an IP address. A normal system is only a DNS client, and never has to answer requests from servers or other clients. Almost all companies, organizations and ISPs will already have one or more DNS servers on their network that all the other hosts can use. If your company already has a DNS server, then there is no need to read this chapter. Instead, see Chapter 19 for information on how to set up your Linux system as a DNS client.

The domain name system is divided into zones (also called domains), each of which has a name like *example.com* or *foo.com.au*. Zones are arranged in a hierarchy, which means that the

`foo.com.au` zone is part of the `com.au` zone, which in turn is part of the `au` domain. At the very top of the hierarchy is the `.` or root zone, upon which the entire DNS system depends.

For each zone, there is at least one DNS server that is primarily responsible for providing information about it. There may also be several secondary or slave servers that have copies of information from the primary, and act as backups in case the master server for the zone is unavailable. A single DNS server may host multiple zones or sometimes may not host any at all. A server is typically responsible for providing information about the zones that it hosts and for looking up information in other zones when requested to by DNS clients.

For a zone hosted by a server to be available to DNS clients that do not query that server directly, it must be registered in the parent zone. The most common parent domains like `.com`, `.net` and `.com.au` are managed by companies that charge for zones registered under them. This means that you cannot simply set up a DNS server that hosts a domain like `example.com` and expect it to be visible to the rest of the Internet. You must also pay for it to be registered with one of the companies that adds sub-domains to the `.com` domain.

Each zone contains multiple DNS records, each of which has a name, type and values. The most common type of record is the address or A record, which associates a hostname with an IP address. Other types include the NS or name server record which specifies the DNS server for the zone or a sub-domain, and the MX or mail server record type which defines a host that should receive mail for the zone.

Every zone should have at least one secondary server in case the primary is down or un-contactable for some reason. Secondaries can also share the load on the primary server, because other servers looking up records in the domain will randomly choose a server to query instead of always asking the primary first. In fact, there is no way for other systems to know which server is the master and which are the slaves for a particular zone.

Slave servers can request a copy of all the records in a zone at once by doing a zone transfer. This is done a secondary DNS server when a zone is first added to it and periodically when it detects that the zone has changed or the records in it have expired. A master server can also be configured to notify slaves when a zone changes so that they can perform a zone transfer immediately, ensuring that they are always up to date.

Every zone has a serial number, which is simply a counter that must be incremented each time any record in the zone is changed. The serial is used by slave servers to determine if a zone has changed, and thus if a transfer is needed. Most of the time, it does not matter what the serial number is as long as it gets incremented. However, some domain authorities require it to be in a certain date-based format, such as `YYYYMMDDnn`.

Normally a single server hosts either entirely master zones or entirely slaves. However, this does not have to be the case—a DNS server can be both a master for some zones and a slave for others. There is no upper limit on the number of servers a zone can have, although few have more than three. The important `.com` and root domains have 13 servers, as they are critical to the functioning of the Internet and frequently accessed. Generally, the more slaves a domain has the better, as long as they can all be kept synchronized.

When a server receives a request from a client to lookup a record, it first checks to see if the record is in one of the zones that it hosts. If so, it can supply the answer to the client immediately. However, if the record is not in a hosted zone then the server must query other servers to find it. It starts by querying one of the servers responsible for the root zone, which will reply with the address of another DNS server. It then queries that other server, which will either pro-

vide an answer, or the address of yet another DNS server to ask. This process continues until a server that is responsible for the domain is found and an answer retrieved from it. If the record that the client asked for does not actually exist, then one of the servers in the query process will say so, and the search will be terminated.

For example, imagine if a DNS client asked a server for the IP address of www.webmin.com. The steps that would be followed by the server to find the address are:

1. Ask one of the root servers, such as `a.root-servers.net` (`198.41.0.4`) for the address of www.webmin.com. The server would reply with a list of servers for the `.com` domain, one of which is `a.gtld-servers.net` (`192.5.6.30`).
2. Ask the `.com` server for the address of www.webmin.com. The reply would be a list of servers, one of which is `au.webmin.com` (`203.89.239.235`), the master server for the `webmin.com` domain.
3. As the server for `webmin.com` for the address of www.webmin.com. The reply would be `216.136.171.204`, which is the correct IP address.
4. The resulting IP address is returned to the client, along with a TTL (time to live) so that the client knows how long it can cache the address for.

As you can see, a DNS server can find the address of any host on the Internet by following the simple process used in the steps above. The only addresses that it cannot discover are those of the root servers. Instead, they read from a file when the server program starts. Because the addresses of the root servers very rarely change, it is safe for a DNS server to store them in a fixed file.

If the steps above were followed exactly for every DNS request, then the root servers would have to be queried every time a client anywhere in the world wanted to lookup an IP address. Even though there are 13 of them, there is no way that they could deal with this massive amount of network traffic. Fortunately, DNS servers do not really query the root servers for every request. Instead, they cache results so that once the IP address of a server for the `.com` domain is known, there is no need to ask for root servers for it again. Because every response from a server includes a TTL, other servers know how long it can be safely cached for.

The relationships between IP addresses and their hostnames are stored in the DNS in a different way to the relationship between hostnames and addresses. This is done so that it is possible to lookup a hostname from an IP using a similar process to the steps above. However, this means that there may be a mismatch between the relationship between an IP address and hostname, and between the hostname and IP address. For example, www.webmin.com resolves to `216.136.171.204`, but `216.136.171.204` resolves to `usw-pr-vhost.sourceforge.net!` This can be confusing, but is an inevitable result of the way that queries for IP addresses work.

When a client wants to find the hostname for an IP address like `216.136.171.204`, it converts this address to the record `204.171.136.216.in-addr.arpa`. As you can see, this is just the IP address reversed with `in-addr.arpa` appended to the end. The special `in-addr.arpa` zone is hosted by the root DNS servers, and its sub-domains are delegated to other DNS servers in exactly the same way that forward zones are. Typically each of the final class C zones (like `171.136.216.in-addr.arpa`) will be hosted by the DNS server for the company or ISP that owns the matching class C network, so that it can create records that map IP addresses in that network to hostnames. All of these records are of the special PTR or reverse address type.

The biggest problem with this method of reverse zone hosting is that there is no easy way for anything smaller than a class C network (which contains 256 addresses) to be hosted by a single DNS server. So if a server hosts the zone *example.com* which contains just a single record, *www.example.com* with IP address *1.2.3.4*, the same server cannot also control the reverse mapping for the IP address *1.2.3.4*. Instead, this will be under the control of the ISP or hosting company whose network the web server for *www.example.com* is on. Only organizations big enough to own an entire class C network can host the reverse zone for that network on their own DNS server.

Many organizations have an internal network that uses private IP addresses such as those starting with *192.168*. A network like this might not be connected to the Internet at all, or connected only through a firewall doing NAT. Some people even have networks like this at home, with several machines connected to a small LAN. Only one of these machines (the gateway) might have a single real Internet IP address assigned by an ISP.

On a private network like this, it can also make sense to run a DNS server to assign hostnames to the systems on the internal LAN. It is quite possible to host a zone called something like *home* or *internal* that contains records for internal systems, as well as a reverse zone for the *192.168* network so that IP addresses can be looked up as well. The server can also be set up to resolve real Internet hostnames by querying the root servers, just as any normal Internet-connected DNS server would. However, it will never receive queries from outside the LAN for records in the *home* network because, as far as the rest of the Internet is concerned, that zone does not exist.

30.2 The BIND DNS Server Module

BIND (Berkeley Internet Name Domain) is the most common DNS server for UNIX systems. Several versions have been released over the years, the most recent being version 9. The BIND DNS Server module (found under the Servers category) supports the configuration of versions 8 and 9. The older version 4 has a different configuration file format and can be configured using the BIND 4 DNS Server module, documented in Section 30.18 “The BIND 4 DNS Server Module”.

Because BIND is available for almost all UNIX systems and works identically regardless of the operating system, the instructions in this chapter apply not just to Linux but to other versions of UNIX as well. Most versions of UNIX and Linux include BIND 8 or 9 as a standard package, so it is rarely necessary to install it. If the module cannot find the DNS server, an error message will be displayed on the main page. If this happens, check your operating system CD or website for a BIND package or download and compile the source from www.isc.org/.

BIND’s primary configuration file is `/etc/named.conf`, which contains all of the zones that the server hosts, and global configuration settings that apply to all zones. The records in each zone are stored in separate files, usually found in the `/var/named` directory. This Webmin module always updates all of these files directly, instead of by communicating with the running BIND process. This means that if you are running some other program that dynamically updates zones by communicating with BIND (such as a DHCP server), then this module should not be used as it may interfere with these changes. However, very few systems have this kind of dynamic updating activated.

Versions 9 of BIND has some features that version 8 does not. The most important one that is supported by this Webmin module is views. A view is a set of zones that are visible to only

some DNS clients. Normally, all clients see the same zones, but with BIND 9 you can restrict the visibility of some domains to only particular clients, identified by their IP addresses. This can be useful for creating zones that are only visible to systems on an internal network, even if your DNS server is connected to the Internet.

If you have never set up BIND on your system, when you enter the module for the first time the main page will display a form for setting up the DNS server, as shown in Figure 30.1. This form is only shown if Webmin detects that the configuration file named.conf does not exist or if the zone files directory that it specifies is non-existent. If you are certain that your BIND configuration is valid and that the DNS server is already running, do not click the **Create** button, as your named.conf file will be overwritten. Instead, click on the **Module Config** link and check that all the paths are correct for your system.

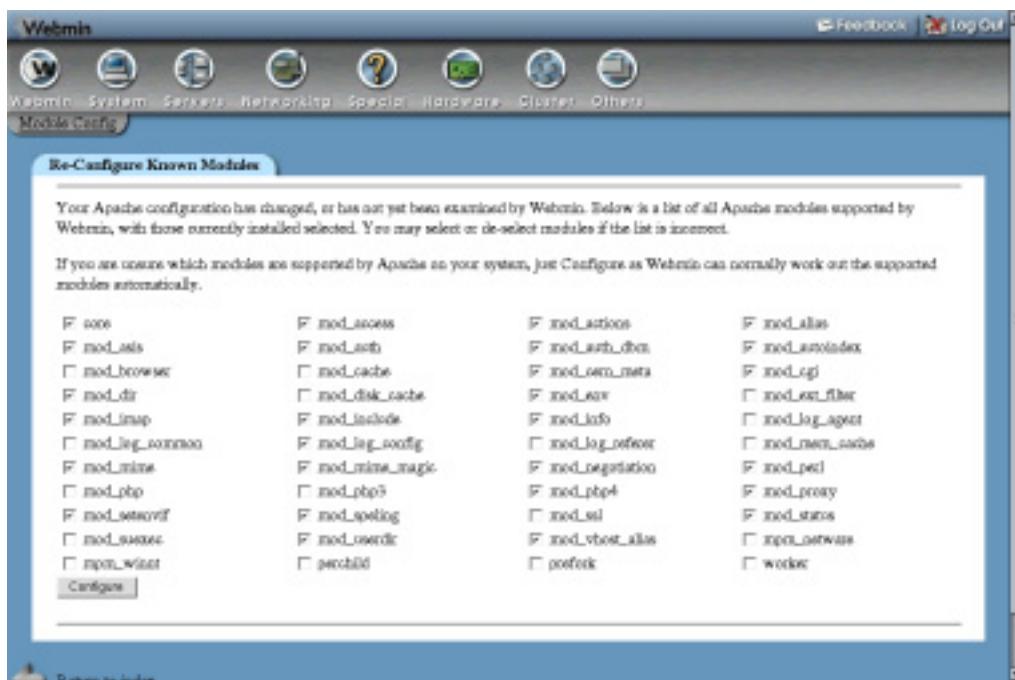


Figure 30.1 The BIND setup form.

If you are setting up BIND for the first time, the setup form gives you three choices:

Setup nameserver for internal non-internet use only If you choose this option, your DNS server will be set up so that it can only resolve records in zones that it hosts. This is only useful on a private network that has no Internet connection at all.

Setup as an internet name server, and download root server information This is the most useful option. It sets your DNS server up to be able to host zones and to lookup records on the Internet. In order to query other Internet domains, a list of the root zone servers is needed, as explained in the introduction. When this option is

selected, Webmin will FTP to `rs.internic.net` and download a file listing the server names and IP addresses for inclusion in the BIND configuration.

Setup as an internet name server, but use Webmin's older root server information This option is identical to the previous one, but does not download the root zone file. Instead, it uses a copy of the file that comes with Webmin which may not be as up to date. However, if for some reason your system cannot connect to the `rs.internic.net` FTP server, this is your best choice.

Depending on which option you choose, a basic `named.conf` file will be created the root zone added to it. The module's main page will then be re-displayed, so that you can add more zones or configure the server further.

When BIND has been set up on your system, the main page will appear as shown in Figure 30.2. At the top is a table of icons for setting global options that apply to your entire DNS server. Below them are icons for each of the zones your server hosts, followed by icons for views if you are running BIND version 9. At the very bottom are buttons for applying the current DNS configuration or starting the BIND server.

If you have just set up BIND for the first time, there will probably be only one zone icon—the root zone. Some Linux distributions that include a BIND package come with a basic configuration file that defines zones like `localdomain` and `127.0.0.0`, which are used for resolving the `localhost` and `127.0.0.1` loopback hostname and IP address.

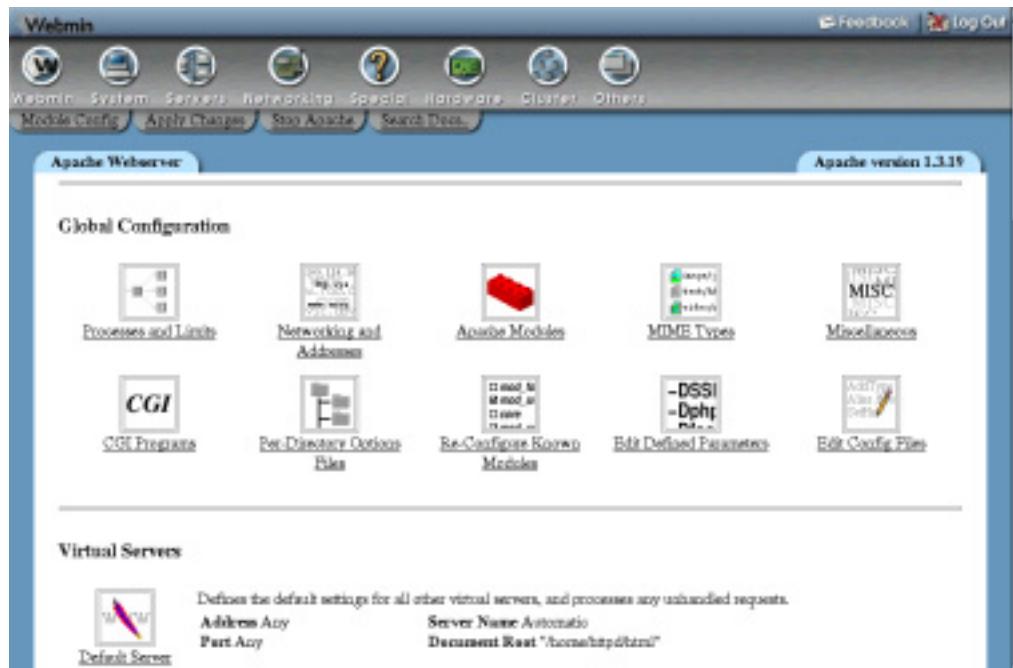


Figure 30.2 The BIND DNS Server module main page.

30.3 Creating a New Master Zone

A master zone is one for which your DNS server is the authoritative source of information. A single zone may be hosted by multiple servers, but only one is the master—all the rest are slaves. If you want to add a new master zone to your server’s configuration, the steps to follow are:

1. Decide on a name for the new zone, such as *example.com* or *internal*. If this is going to be Internet domain that will be visible to other everyone in the world, the domain name must not have been registered by anyone else yet. However, you cannot normally register it yourself until your DNS server has been set up to host it.
2. On the module’s main page, click on the **Create a new master zone** link below the table of existing zones. This will take you to the page shown in Figure 30.3 for entering the details of the new zone.
3. If this is to be a forward zone like *example.com* or *foo.com.au*, leave the **Zone type** field set to **Forward**. However, if it is a reverse zone for looking up hostnames from IP addresses, set the field to **Reverse**.
4. In the **Domain name / Network** field, enter the name of the zone without any trailing dot. For a reverse zone, just enter the network address like *192.168.1*. Webmin will automatically convert this to the *in-addr.arpa* format for you when the domain is created.
5. The **Records file** field controls where the configuration file containing the zone’s records is stored. If you leave it set to **Automatic**, the filename will be determined automatically based on the module’s configuration and the **directory** setting in the *named.conf* file. This is usually the best option, as it will result in the records file being created in the same directory as any existing zones, such as */var/named*.
However, if you de-select the **Automatic** option and enter a filename instead, all records for the zone will be written to that file. If you enter the name of an existing file, it will be overwritten when the domain is created.
6. In the **Master server** field, enter the full domain name of the master DNS server for this zone. This must be the canonical name of your system, such as *server.example.com*, not a short name like *server*. This server (and the values from the next 5 fields) are used to create the new zone’s SOA record.
7. In the **Email address** field, enter the address of the person responsible for this zone. You can use the @ symbol in the address, which Webmin will automatically convert to a dot for inclusion in the SOA record.
8. The **Refresh time** field determines how often secondary servers should check with this master server for updates to the zone. The default is reasonable, but you may want to increase it for zones that rarely change, or decrease it for those that are frequently updated.
9. The **Transfer retry time** field determines how long a secondary server should wait after a failed zone transfer before trying again.
10. The **Expiry time** field controls the maximum amount of time that a secondary DNS server for the zone should cache records for before re-transferring them from the master.
11. The **Default time-to-live** field determines the TTL of records in the zone that do not have one set explicitly.

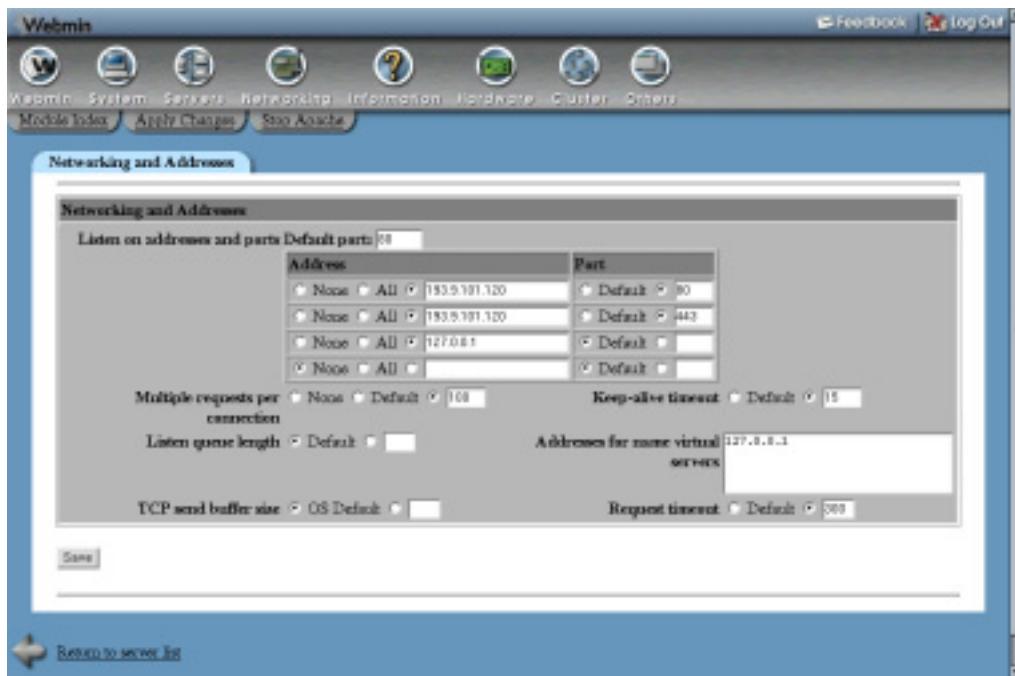


Figure 30.3 The new master zone creation form.

12. Click the **Create** button at the bottom of the page. As long as the form has been filled in correctly and the zone does not already exist on your server, you will be taken to a page for adding new records to the zone.
13. Return to the module's main page which will now include an icon for your new zone and click the **Apply Changes** button at the bottom to activate it.

A newly created zone will contain only one record (unless you have set up a template). To add more, follow the instructions in Section 30.4 “Adding and Editing Records”. Once you have set up the basic records in your domain, you can register it with the authority that manages the parent domain, such as .com or .com.au. Some domain authorities will not allow you to register zones that do not have at least two servers (one master and one slave) and name server records in the zone for those servers.

30.4 Adding and Editing Records

The most useful feature of the BIND DNS Server module is the ability to add, edit and delete records in the master zones hosted by your server. For example, if you wanted to set up a web server in your domain *example.com*, you would need to add an Address record for *www.example.com* with the IP address of the server. To add a new record like this, the steps to follow are:

1. On the module's main page, click on the icon for the zone that you want to add to. This will bring up the page shown in Figure 30.4, at the top of which is a table of icons, one for each record type.
2. Click on the icon for the type of record that you want to add. The most common type is **Address**, which associates an IP address with a hostname. See Section 30.5 "Record Types" for a complete list of all the supported record types.
3. Clicking on the icon will take you to a page listing all existing records of that type. Above the list is a form for entering a new record.
4. In the **Name** field, enter the name of the new record relative to the zone name. For example, if you wanted to add the record *www.example.com*, you should just enter *www*. It is also possible to enter the full record name, as long as it has a dot at the end to indicate that it is not relative to the zone. Do not enter just *www.example.com*, as it will be converted to *www.example.com.example.com*, which is probably not what you want.
5. If this record is going to change more frequently than the rest of the zone, change the **Time-To-Live** field from **Default** to the estimated time between changes. This determines how long DNS clients and other servers will cache the record for.
6. If you are adding an Address record, enter the complete IP address of the host into the **Address** field. See Table 30.1 on page 348 for a description of the fields that appear when adding other types of records and what they mean.
7. The field **Update reverse?** only appears when adding an Address record. It controls the automatic creation of a corresponding record in a reverse zone which associates the hostname with the IP address. Naturally, this can only be done if the IP that you enter is in a network that your system is the primary reverse DNS server for. This keeps the forward and reverse zones synchronized, which can be very useful.

If **Yes** is selected, a reverse address record will be added as long as one does not already exist in the reverse zone for the same IP address. Often many hostnames will have the same IP, such as those used for name-based virtual hosting. In cases like these, you don't want to change the reverse mapping if one already exists.

The **Yes (and replace existing)** option works the same as **Yes**, but if a reverse record for the IP address already exists it will be updated with the new hostname. This can be useful if you know there is an existing record that you want to replace.

If **No** is selected, no reverse address will be created even if it is possible.

8. When you are done filling in the form, click the **Create** button at the bottom. As long as it is filled in correctly, the record will be added to the list below the form. When writing to the zone's records file, Webmin will use the full canonical format for the record name, such as *www.example.com.*, even if you just enter *www*.
9. To activate the new record so that it can be looked up by DNS clients and other servers, you will need to click the **Apply Changes** button on the module's main page. If you are planning to add or edit several records, it is usually better to wait until all the changes are complete before hitting the apply button.

If it is available, you can instead use the **Apply Changes** button at the bottom of the master zone page shown in Figure 30.4. This uses the `rndc` command to tell BIND to reread only the file for this zone, which can be much faster on a system that hosts a large number of domains.

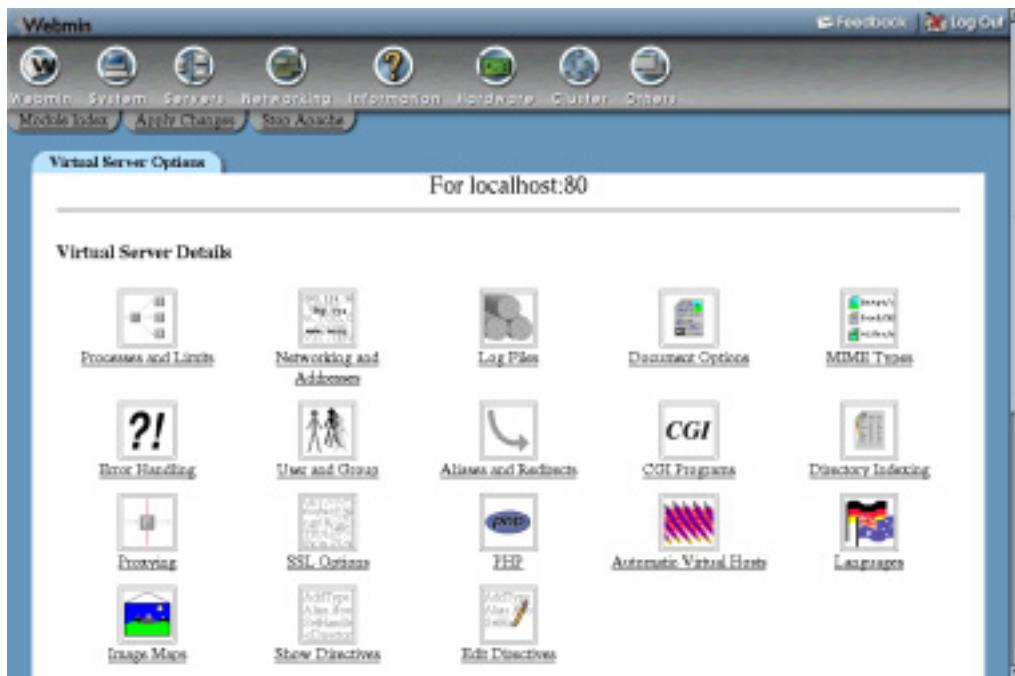


Figure 30.4 The master zone editing page.

Although the instructions above are focused on adding an Address record, the process of adding other record types to a forward zone is almost identical. The **Update reverse?** field does not exist and the **Address** field is replaced with one or more different fields. Section 30.5 “Record Types” explains in detail what fields are available for each type of record known to Webmin.

When adding a Reverse Address record to a reverse zone, the form is quite different. The **Address** field appears before the **Hostname** and the hostname must always be entered in canonical form with a dot at the end, like `www.example.com`. The **Update reverse?** field is replaced with **Update forward?**, which controls the automatic creation of a record in the corresponding forward zone. However, there is no option to overwrite an existing forward record. If one with the same name already exists, it will not be touched even if **Yes** is selected.

Every time a record is added to or updated in a zone using Webmin, its serial number will be automatically incremented. This also applies to reverse zones that are automatically updated when adding an Address record and vice-versa. This means that when you apply the changes, other DNS servers will be able to detect that the zone has changed by comparing the new serial number with the old one that they have cached.

To edit an existing record in a zone, the steps to follow are:

1. On the module’s main page, click on the icon for the zone that you want to edit, which will bring you to the page shown in Figure 30.4.
2. Click on the icon for the type of record that you want to change, which will display a page listing all records of that type in the zone. Alternately, you can click on the **All**

- Record Types** icon which will bring up a list of every single record in the zone regardless of type.
3. Click on the name of the record that you want to edit. Your browser will display a form similar to the one used for adding a record, but with the fields already filled in with the details of the existing address.
 4. To re-name the record, edit the contents of the **Name** field. It will be shown in canonical form with a dot at the end initially, but you can change it to a name relative to the domain if you wish.
 5. Adjust the **Time-To-Live** field in you want this record to have a different TTL or set it to **Default** to make it the same as the rest of the zone.
 6. If this is an Address record, change the IP in the **Address** field. For other record types, the fields are the same as those on the record creation form and have the same meanings.
 7. For Address records, the field **Update reverse?** is displayed. Selecting **Yes** will cause the corresponding record in the reverse zone to be have its name and address changed to match this forward record. If you change the IP so that the reverse address is no longer in the same network, it will be removed from the old reverse zone and added to the new reverse zone (if it is hosted by your server).
 8. For Reverse Address records, the field **Update forward?** is shown instead. If **Yes** is selected, the corresponding Address record in its forward zone will be changed to match any changes that you make on this form.
 9. Click the **Save** button to update the record in the zone file and return to the list of record types.
 10. To activate the changes, click the **Apply Changes** button back on the module's main page.

To delete a record from a zone, click on the **Delete** button on the editing form instead of **Save**. For Address records, if the **Update reverse?** field is set to **Yes**, the corresponding Reverse Address record will be deleted as well. Apart from that, the process of deleting a record is identical no matter what type it is. The same thing happens when deleting a Reverse Address record. The matching Address record is deleted as well, as long as the **Update forward?** field is set to **Yes**.

The list of records in a zone is initially sorted according to the module configuration, which usually means that records will be displayed in the order that they were added. To change this, you can click on a column heading like **Name**, **Address** or **Real Name** to sort them by that column instead. The sorting is only temporary though, and will be lost if you return to the main page and re-open the zone. To change it permanently, see the **Order to display records in** field in Section 30.17 “Configuring the BIND DNS Server Module”.

30.5 Record Types

Webmin does not support all of the record types that BIND knows about, only those that are most commonly used. The list below covers all of the supported types, and explains what they are used for and what fields are available when adding or editing a record of that type in Webmin. Next to each type name is the short code used by BIND itself for identifying the type in the records file.

Address (A) An Address records associates an IP address with a hostname. Any system that you want to be able to connect to via HTTP, telnet or some other

protocol using its hostname must have an address record so that clients can look up its IP. A single hostname can have more than one Address record, which is often done to spread the load for a website across multiple servers. It is also valid to create multiple records of this type with different names but the same IP, such as when setting up name-based Apache virtual servers.

When creating or editing an Address record, the field **Address** is displayed for entering the IP associated with the hostname. A field labelled **Update reverse?** also appears, which controls the automatic creation and modification of a Reverse Address record in the appropriate reverse zone. See Section 30.4 “Adding and Editing Records” for more details.

Name Sever (NS) Records of this type defines a name server that is responsible for a zone. Every zone must have at least one Name Server record for itself and may have additional records that specify the DNS servers responsible for subdomains. If you set up a secondary DNS server for a zone, be sure to add a Name Server record for the zone on the master server. In this case, the name of the record will be the canonical name of the zone, such as *example.com*.

When creating or editing a record of this type, a field labelled **Name Server** will be displayed. This must be filled in with the IP address or hostname of the DNS server that is responsible for the zone. If you enter a hostname, it must have an IP address set by an Address record in some zone on your server.

Name Alias (CNAME) This type of record creates an additional name for an existing Address or Reverse Address record. When a DNS client requests the IP address of a record of this type, it will get the IP of the record that the Name Alias points to instead. This kind of record can be useful if you have a single host that needs to be accessible under several different names, such as a web server doing name-based virtual hosting. Even though this could also be done by creating multiple Address records, creating just a single Address and multiple Name Aliases is more flexible as it allows easier updating if the IP address of the host ever changes.

The form for editing and creating Name Alias records contains a field labelled **Real Name**. This must be filled in with either the canonical name of the record that the alias points to (such as *webserver.example.com*), or with a short name that is relative to the zone that the Name Alias record is in.

Mail Server (MX) Mail Server records tell mail delivery programs like Sendmail and Qmail which system to contact when delivering mail to a domain or host. Without a record of this type, mail for a domain will be delivered to the system whose IP is specified in the Address record for the zone itself. This is not always desirable, as you may want that IP to be the address of a web server, so that web browsers can connect to <http://example.com/> as well as <http://www.example.com/>. A Mail Server record can solve this problem by having only email for *example.com* sent to another hosts, and all other traffic to the web server.

Each Mail Server record has a priority, which tells mail delivery programs which mail server should be tried first. The record with the lowest priority should point to the system that actually receives and stores email for the domain, while those with

higher priorities generally point to systems that will simply relay mail. Delivery programs will try each in turn starting with the lowest, so that if the primary mail server is down, email will still be sent to a relay that can hold it until the primary comes back up.

When adding or editing a Mail Server record, two additional fields are displayed. The first is labelled **Mail Server** and must be filled in with the canonical or relative hostname of a system that can accept mail for the domain or hostname entered in the **Name** field. The second is labelled **Priority** and must be used to specify a numerical priority for this particular mail server. Normally a priority of 5 is used for the primary mail server and 10 for backup relays. If you only have one mail server for your domain, it doesn't really matter what number is entered into this field. It is possible for two servers to have the same priority, in which case one will be chosen randomly to deliver to.

A Mail Server record can use the * wildcard in its name, which indicates to mail programs that a particular mail server is responsible for all hosts in a domain. For example, a record named like *.example.com would match the hostname pc1.example.com and any other hosts in the zone. This can be useful if you want to force mail that would otherwise be delivered directly to workstations in your domain to go through a central mail server instead. Webmin will not let you use wildcards unless the **Allow wildcards** module configuration option is set to Yes though, as explained in Section 30.17 “Configuring the BIND DNS Server Module”.

Host Information (HINFO) Records of this type are used to record information about the hardware and operating system of a particular host. For example, you might create one that says that *server1.example.com* is an x86 PC running Linux. However, they are very rarely used and are in fact considered a security risk, as they give out information to potential attackers that could be used to take over a server.

When creating or editing a Host Information record, the fields **Hardware** and **Operating System** are displayed for entering the architecture and operating system type of a host. The values you enter must not contain any spaces. Typically, they are replaced in the hardware type and operating system strings with _ characters.

Text (TXT) A Text record associates an arbitrary message of some kind with a name. Although they are hardly ever used, they can be useful for attaching comments to hostnames. Be aware though that any such comments will be available to anyone on the Internet that can look up records in your domain, and so should not contain sensitive information.

The field **Message** is displayed when entering or editing a Text record. You can enter any text that you like, including spaces.

Well Known Service (WKS) A record of this type associates a hostname, port and protocol with a name. It can be thought of as a generalized variant of the Mail Server record, which tells clients which host provides a particular service for some domain or hostname. However, almost no programs actually look up WKS records, so in practice they are pretty much useless.

When adding or editing one of these records, the fields **Address**, **Protocol** and **Services** are available. The first is for entering the IP address of a host that provides the services for the host or domain entered into the **Name** field. The second is for selecting the network protocol that the services use, either TCP or UDP. The last is for entering a list of port numbers or names (from the `/etc/services` file) for services that the host provides.

Responsible Person (RP) This type of record is used for specifying the person or group responsible for a particular host. Each of these records has two values associated with it—an email address, and the name of Text record containing the person's name. Responsible Person records are rarely seen and are not used by any mail delivery program or Internet client.

The **Email Address** field shown when editing or adding one of these records is for entering the complete address (like `jcameron@example.com`) of the person responsible for the host whose name is entered into the **Name** field. The **Text Record Name** field is for entering the relative or canonical name of a Text record that contains the person's real name.

Location (LOC) Location records are used to specify the physical location in latitude and longitude of a host. They are hardly ever seen, and thus not used by many programs. However, they can be useful in large organizations that have hosts in many countries.

When adding or editing a Location record, the field **Latitude and Longitude** is displayed for entering the location of the host in the **Name** field. It must be formatted like `42 21 43.528 N 71 05 06.284 W 12.00m 30.00m 10000.00m 10.00m`.

Service Address (SRV) Records of this type are used to associate a domain name, service name and protocol with a particular host. They allow you to specify which server a client should contact for a particular service and hostname, instead of just connecting to the host. In a way, they are like Mail Server records but far more flexible. For example, you can specify that the POP3 server for `example.com` is `mail.example.com`, but the web server is `www.example.com`. At the time of writing, SRV records are mostly used by Windows client systems.

When adding or editing a Service Address record, the fields **Protocol** and **Service name** are displayed near the **Name** text box. For the protocol, you must select either TCP or UDP from the menu. For the service name, you must enter a well-known name from the `/etc/services` file, such as `pop3` or `telnet`. To look up an SRV record, a client combines the service name, protocol and name to get a record name like `_telnet._tcp.example.com`. Webmin does this for you automatically when editing or adding a Service Address record, but you can see the combined name on the page listing records of this type. Webmin also automatically added the `_s` before the service and protocol, but hides them when an SRV record is being displayed or edited. This means that there is no need to enter them manually when creating or editing a record of this type.

The **Priority** field must be used to enter a numeric priority for this server, which has the same meaning as the priority in a Mail Server record. The **Weight** field must

contain a weighing for this particular server or zero if there is only one record with the same name, protocol and service name. A higher weighting tells clients to try this server more often than one with a lower weight.

The **Port** field must contain a port number for clients to connect to on the server, which does not necessarily have to be the standard port for the service. In the **Server** field, you must enter the hostname or IP address of the system that actually provides the service, and that clients actually connect to.

Public Key (KEY) This type of record stores key information for a host, used for IPsec VPNs. Since they are rarely used and Webmin's IPsec module is not covered in this book, the details of this record type are not explained here.

The record types support by Webmin in reverse zones are:

Reverse Address (PTR) A reverse address record associates a hostname with an IP address in a reverse zone. For DNS clients to be able to lookup hostnames from IP addresses in your network, you will need to create one record of this type for each host. However, most of the time this is done automatically by Webmin when adding and editing Address records. If you create your own Reverse Address records, make sure that they are synchronized with the matching Address records.

When adding or editing a record of this type, the fields **Address** and **Hostname** are displayed. The first is for entering a complete IP address, like *192.168.1.10*. This will be automatically converted by Webmin to the `in-addr.arpa` format used internally by the DNS system for reverse addresses. The second field is for entering a hostname in canonical form, such as *pc1.example.com*.—be sure to always put a dot at the end, or else the hostname will be relative to the reverse zone, which is definitely not what you want.

Name Server (NS) Name Server records in a reverse zone have an identical purpose to those in a forward domain—they tell other DNS servers the IP address or hostname of a server responsible for the zone or a sub-domain. This means that one must be added for each primary or secondary DNS server for the zone.

The **Zone Name** field that appears when adding or editing a record of this type is for entering the name of the zone that the server is responsible for, which will typically be the zone that contains the record. However, unlike Reverse Address records this field is not automatically converted to `in-addr.arpa` format. Instead, you must enter it in fully qualified form like *1.168.192.in-addr.arpa*, if defining a name server for the *192.168.1* network. In the **Name Server** field, you must enter an IP address or canonical form hostname for the DNS server, such as *ns1.example.com*.

Name Alias (CNAME) Records of this type behave exactly the same in reverse zones as they do in forward domains. However, you must fill in the **Name** and **Real Name** fields with reverse names in `in-addr.arpa` format, as Webmin will not convert them for you.

Name Alias fields are most useful in reverse zones for doing partial subnet delegation, as covered in Section 30.14 “Setting Up Partial Reverse Delegation”.

30.6 Editing a Master Zone

You can use Webmin to edit many of the settings that apply to an entire master zone, such as the expiry and retry times, and the clients that are allowed to query it. These settings effectively apply to all records in the zone, although some (such as the TTL) can be overridden on a per-record basis.

Webmin uses the term zone parameters to refer to all information stored in the domain's SOA record, including the primary name server, administrator email address and retry and expiry times. All of these are set when the zone is created, but you can edit them at any time by following these steps:

1. On the module's main page, click on the icon for the zone that you want to edit. This will take you to the form shown in Figure 30.4.
2. Click on the **Zone Parameters** icon, which will bring up a form for editing the parameters.
3. The **Master server** field only needs to be edited if the Internet hostname of the DNS server has changed. Enter a fully-qualified hostname, with a dot at the end.
4. To change the address of the person responsible for the zone, edit the **Email address** field. Any @ symbols that it contains will be automatically converted to dots for use in the SOA record, as BIND requires.
5. The **Refresh time**, **Transfer retry time**, **Expiry time** and **Default time-to-live** fields all have the same meanings as explained in Section 30.3 “Creating a New Master Zone”. If records in your zone are going to be changing frequently in future, you may want to reduce some of these times. However, any changes, may not be detected by secondary servers and DNS clients until the old refresh or expiry time has elapsed, even if the new times are much lower. This is because they will wait for the old times to elapse before checking with the master server again to discover the new ones.
6. Click the **Save** button at the bottom of the page when you are done, and then the **Apply Changes** button back on the module's main page. The serial number in the SOA record will be automatically incremented when the form is saved, so that secondaries know that the zone has changed.

The is another set of options that you can edit for a master zone which are stored in the named.conf file in the zone's section. These control which servers and clients are allowed to query records in the zone, do zone transfers and update records over the network. The most useful of these options specifies a list of slave DNS servers for the zone that should be notified when a change occurs, so that they can perform immediate zone transfers and thus remain synchronized.

To edit these master zone options, the process to follow is:

1. On the module's main page, click on the icon for the zone that you want to edit. This will take you to the form shown in Figure 30.4.
2. Click on the **Edit Zone Options** icon, which will bring up a form showing the existing settings.
3. The **Check names?** field determines the level of checking that BIND performs on records in this zone when it reads the records file. The available options are:

Warn If an invalid record is found, an error will be written to the system log file, but processing of other records continues normally.

Fail Invalid records cause the entire zone to be rejected, but other zones will still be processed normally.

Ignore No checking is done at all.

Default The global default from the Zone Defaults page is used. If it is not set, then the default compiled into BIND will be used instead. The default is to fail when invalid records are encountered.

4. To have secondary servers notified when records in the zone change, set the **Notify slaves of changes?** field to **Yes**. BIND works out which slaves will be notified by looking at the Name Server records for the zone, and the list of IP addresses in the **Also notify slaves** field. If your zone has any secondary servers, then you should definitely turn this option on.
5. To allow some systems to update records in the zone dynamically, fill in the **Allow updates from** field with a list of IP addresses, IP networks (like *192.168.1.0/24*) and BIND ACL names. Only those hosts that match will be able to modify records using commands like `nsupdate` and if the list is left empty updates will not be allowed at all. You should be careful allowing the dynamic update of zones in which Webmin is also being used to edit records, as it is very likely that updates made dynamically will be overwritten by changes made in this module or vice-versa.
6. By default, all DNS clients and servers will be able to lookup records in the zone. This may not be what you want for a zone that is used only on an internal network, as it may give away sensitive information to potential attackers. To restrict queries, fill in the **Allow queries from** field with a list of IP addresses, IP networks and BIND ACL names. If the field is left empty, the field with the same name on the global **Zone Defaults** page determines which clients are allowed.
7. To restrict the clients and servers that are allowed to perform zone transfers of all the records in this domain, fill in the **Allow transfers from** field. Often you will only want to allow secondary servers to perform transfers, especially if your zone is very large or contains records that you want to hide from attackers. Enter a list of IP addresses, IP networks and ACL names into the field to limit transfers to only matching clients. If it is left empty, the **Allow transfers from** field on the Zone Defaults page applies instead.
8. To specify additional slave servers to be notified when the zone changes, fill in the **Also notify slaves** field with a list of IP addresses. BIND normally uses the addresses of all secondary servers for the zone from its Name Server records, but this may not always be complete.
9. When you are done, click the **Save** button at the bottom of the page to update the BIND configuration file with your changes. You will need to use the **Apply Changes** button on the module's main page to make them active.

If a master zone is no longer needed, you can use this Webmin module to totally delete it along with all the records that it contains. To do this, the steps to follow are:

1. On the module's main page, click on the icon for the zone that you want to edit.
2. Click on the **Delete Zone** button at the bottom of the page.
3. When deleting a forward zone, the field **Delete reverse records in other zones?** controls whether matching Reverse Address records in hosted reverse zones for all of the address

records in this one should be removed as well. It is generally safe to set this to **Yes**, as only records with the exact same IP address and hostname will be deleted.

4. Similarly, when deleting a reverse zone the field **Delete forward records in other zones?** determines whether matching forward records should be deleted too.
5. Once you have made your selection and are sure you want to go ahead with the deletion, click the **Delete** button. The zone's entry in the named.conf file will be removed and its records file deleted.

You can convert a master zone to a slave zone of the same name without needing to delete and re-create it. This can be useful if the new server is taking over as the master for some domain, or if the master and secondary servers are switching roles. Section 30.8 “Editing a Slave Zone” explains how to carry out the reverse action of converting a slave zone to a master, which may be useful in this situation.

To convert a zone, the steps to follow are:

1. On the module's main page, click on the icon for the zone that you want to edit, then on the **Edit Zone Options** icon.
2. When you click on the **Convert to slave zone button**, the zone's entry in named.conf will be immediately updated to convert it to a slave zone. The browser will then return to the module's main page.
3. Normally, every slave zone has a list of master server IP addresses it can use to perform zone transfers from. In the case of converted zones, this list will be initially empty unless the **Default master server(s) for slave zones** module configuration option is set. Follow the instructions in Section 30.8 “Editing a Slave Zone” to set the master servers addresses correctly.
4. To activate the change, click on the **Apply Changes** button on the module's main page.

30.7 Creating a New Slave Zone

A slave or secondary zone is one for which your DNS server gets the list of records from a master server for the zone. Generally, slave servers are used to reduce the load on the primary server or act as a backup in case it goes down. For important zones (such as a company's Internet domain), you should always have at least one slave server so that your website is still accessible and email can still be delivered even if the primary goes down.

The secondary DNS server for a domain should not usually be located on the same network as the master, so that the failure of that network cannot take them both down. Many ISPs and hosting companies will host secondary zones for their customer's domains for free, on their own DNS servers. If your ISP provides this service and you want to set up a secondary server for an Internet domain, you should take advantage of it. If so, most of this section can be skipped. However, if you want to add a slave server for an internal domain or have a large company network with many connections to the Internet, then the instructions below explain how to set it up:

1. On the main page of the BIND DNS Server module, click on the **Create a new slave zone** link above or below the list of existing zones. This will bring up the form shown in Figure 30.5, for entering the details of the new domain.

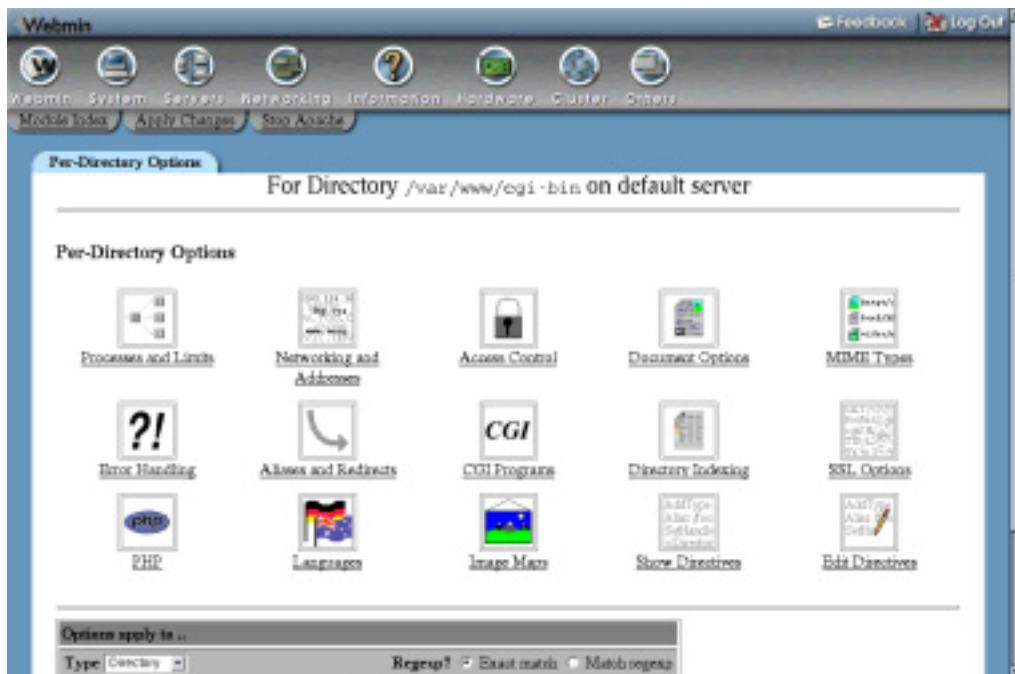


Figure 30.5 The slave zone creation form.

2. For a forward zone like *example.com*, set the **Zone type** field to **Forward** and enter the zone name into the **Domain name / Network** field. For a reverse zone that maps IP addresses to hostnames for a network, choose the **Reverse** option and enter the network address (like *192.168.1*) into the **Domain name / Network** text field.
3. The **Records file** field determines if BIND keeps a cache of the records in this zone in a file and, if so, where that file is located. If the option **None** is chosen, records that the DNS server transfers from the master will be kept in memory only, and lost when the server is re-started. This should only be chosen if there is a good network connect between the master and slave servers, as it will increase the number of zone transfers that your server must perform.
If you choose **Automatic**, Webmin will generate a filename in the zone files directory specified in the `named.conf` file (usually `/var/named`). Whenever your server does a zone transfer, all records will be written to this file in the standard format.
If the final option is selected, you can enter the full path to a file in which records should be stored into the field next to. This can be useful if you want to separate the records files for master and slave zones.
4. In the **Master servers** field, enter the IP addresses of the master DNS server and any other secondary servers for the zone. BIND will try these servers in order when doing a zone transfer, so the master should be first on the list. You must enter at least one address, so that your server knows where to get records from.

5. Click the **Create** button to have the new slave zone added to your server's configuration. Your browser will be re-directed to a page for editing options for the zone.
6. Return to the module's main page and click the **Apply Changes** button on the main page to make the addition active.
7. On the master server, add a new Name Server (NS) record for the zone with the IP address of the secondary server. This can be easily done in Webmin by following the instructions in Section 30.4 "Adding and Editing Records".
8. Configure the master DNS server to notify this slave of any changes to records in the zone. The steps in Section 30.6 "Editing a Master Zone" explain how.
9. If this is an Internet domain, notify the registrar for the parent zone of the new secondary server. Most provide online forms for editing the list of nameservers for a domain, to which you can add the secondary's IP. This is necessary so that other hosts on the Internet know to use the slave server is the master is down.

Another type of zone that is closely related to the slave zone is the stub. They are like slave zones, but only contain Name Server records that have been transferred from a master server, instead of all the records. Stub zones are rarely used, but can be useful for ensuring that the Name Server records in a zone for its sub-domains are the same as those use in the sub-domain itself. The steps for creating one are almost identical to those above, but in Step 1 you must use the **Create a new stub zone** link on the main page instead.

30.8 Editing a Slave Zone

After a slave zone has been created, it is still possible to edit several options that apply to it. Naturally there is no way to add or edit the actual records within the zone, but you can still change the list of master servers, the records file and the clients that allowed to query it. To change these setting, the steps to follow are:

1. On the module's main page, click on the icon for the slave zone that you want to edit. Your browser will display the form shown in Figure 30.6.
2. Scroll down to the **Zone Options** form at the bottom of the page.
3. To edit the list of other master and slave servers for this zone, change the IP addresses in the **Master servers** field. If a new secondary server has been added, it should be added to this list on all other secondaries so that they can do zone transfers from it. If the IP address of the master has changed, the list must be updated with the new address.
4. To change the amount of time that the server will wait before giving up on a zone transfer, de-select **Default** for the **Maximum transfer time** field and enter a number of minutes into the text box next to it.
5. If the **Records file** field is set to **None**, records transferred from the master server for this zone will be kept in memory only. However if a filename is entered, records will be written to that file instead in the standard format. This is the best option, as it minimizes zone transfers and allows you to view the records on the secondary server, as explained below.
6. To have this DNS server notify others when the zone changes, change the **Notify slaves of changes?** field to **Yes**. This is only really useful if there are other secondary servers that perform zone transfers from this one, and may not be able to receive update notifications from the master server.



Figure 30.6 The slave zone editing form.

The DNS servers to notify are determined from the Name Server records for the zone, and the contents of the **Also notify slaves** field.

7. By default, all DNS clients and servers will be able to lookup records in the zone. To change this, fill in the **Allow queries from** field with a list of IP addresses, IP networks and BIND ACL names. If the field is left empty, the field with the same name on the global **Zone Defaults** page determines which clients are allowed.
8. To restrict the clients and servers that are allowed to perform zone transfers of all the records in this domain, fill in the **Allow transfers from** field with a list of IP addresses, IP networks and ACL names. If it is left empty, the **Allow transfers from** field on the Zone Defaults page applies instead.
9. The other fields on the form such as **Check names?** and **Allow updates from?** are not really used for slave zones, and so can be left unchanged.
10. When you are done making changes, click the **Save** button. As long as there were no syntax errors in your input, you will be returned to the module's main page. Click the **Apply Changes** button there to make the modifications active. Note that this will not always force a re-transfer of the zone, even if the master servers have changed. For slave zones that use records files, BIND will only do a transfer when the zone expires or the server receives notification of a change.

When editing a slave zones that uses a records file, it is possible to browse the records in Webmin. At the top of the page that appears when you click on the slave zone's icon is a table of record types, just like the one that appears on the master zone form. Each can be clicked on to

list the names and values of records of that type in the zone, as known to the secondary server. Editing or adding to them is impossible of course, as any changes must be made on the master server which is the authoritative source of records for the domain.

To stop your system acting as a slave server for a zone, you will need to delete it from the BIND configuration. This is generally a safe procedure, as all the records in the zone have been copied from a master server and can be easily replaced. However, you should update the Name Server records in the zone and notify the parent domain's registrar that your system is no longer a secondary for the zone, so that other DNS servers do not waste time querying it.

To delete a slave zone, the steps to follow are:

1. On the module's main page, click on the icon for the slave zone that you want to edit. This will take you to the form shown in Figure 30.6.
2. Click on the **Delete** button in the bottom right-hand corner of the page, which will display a confirmation form.
3. Hit the **Delete** button if you are sure you want to delete the zone.
4. After your browser returns to the module's main page, click on **Apply Changes** to make the deletion active.
5. On the master server, remove the Name Server (NS) record for this secondary server from the zone.
6. If this is an Internet domain, notify the parent zone registrar of the removal of this secondary server. Failure to do so could cause problems if other DNS servers attempt to query this one for records in the domain when it cannot provide answers.

The final thing that you can do to a slave zone is convert it to a master. This is only possible for zones that use a records file, so that Webmin can view and edit that file in future. If you do such a conversion, make sure that the original master server is changed to become a slave or stops hosting the zone altogether—the same domain cannot be served by two masters.

The steps to convert a zone are:

1. Click on its icon on the module's main page.
2. Scroll down to the bottom of the slave zone page and hit the **Convert to master zone** button. This will immediately update the named.conf file to change the zone's type, but will not make any other changes.
3. To make the conversion active, click on the **Apply Changes** button on the module's main page.
4. You can now edit records in the domain just as you would with any normal master zone, by following the instructions in Section 30.4 “Adding and Editing Records”.

30.9 Creating and Editing a Forward Zone

A forward zone is one for which your DNS server simply forwards queries to another server on behalf of whoever is making the request. They can be useful if the zone is actually hosted by another server that cannot be reached by clients of this server. It is possible to set up BIND to forward all requests for any non-hosted zones to another server, as explained in Section 30.12 “Configuring Forwarding and Transfers” below. A forward zone entry does the same thing, but for just a single domain.

To set one up, the steps to follow are:

1. On the module's main page, click on the **Create a new forward zone** link above or below the list of existing domain icons. This will take you to the zone creation form.
2. Set the **Zone type** field to either **Forward** or **Reverse**, as when creating master and slave zones.
3. For a forward zone, enter its full name (without a dot at the end) into the **Domain name / Network** field. For a reverse zone, enter its network (like *192.168.1*) into the field instead. Webmin will automatically convert it to in-addr.arpa format when the zone is added.
4. In the **Master servers** field, enter a list of IP addresses for the DNS servers that can be queried to look up records in the zone. These must all be master, slave or forward hosts for the domain.
If no addresses are entered at all, BIND will always perform normal lookups of records in the zone instead of forwarding requests to another server. This can be used to override the global forwarding settings on the Forwarding and Transfers page for a single zone.
5. Click the **Create** button to have the zone added to BIND's configuration file. Your browser will be taken to a page for editing options in the new domain.
6. Return to the module's main page, and hit the **Apply Changes** button to make it active.

After a forward zone has been created, you can delete it or edit the few settings that it has by following these steps:

1. Click on the icon for the zone on the module's main page, which will bring your browser to a small form for editing its options.
2. To change the list of DNS servers that requests are forwarded to, edit the IP addresses in the **Master servers** field. If none are entered, requests for records in this domain will be looked up directly.
3. If the **Try other servers?** field is set to **Yes**, BIND will try a normal direct lookup for requests in this zone if it cannot contact any of the listed servers.
4. Click the **Save** button to store your changes and then **Apply Changes** back on the main page to activate them.

Or to delete the forward zone, click on **Delete** and then **Delete** again on the confirmation page.

30.10 Creating a Root Zone

As the introduction explains, a root zone is one that contains the information that your DNS server needs to contain the Internet root servers. Without one, it is impossible to resolve records in domains other than those hosted by your server. Fortunately, one will almost always exist already in your BIND configuration, created either by Webmin or included as part of the default setup.

You may need to create a root zone if one does not exist yet because you selected the **internal non-internet use only** option when setting up the module for the first time, but have now connected your system to the Internet. Adding a second root zone can also be useful when views have been configured, as explained in Section 30.15 “Using BIND Views”.

Webmin will only allow you to create a root zone if none yet exists or if a view exists that does not contain one, because there is no point having two such zones. To add one, the steps to follow are:

1. On the module's main page, click on the **Create a new root zone** icon.
2. Fill in the **Store root servers in file** field with a filename to use for the root zone file. If one already exists, then this field will already contain its path. Otherwise, you should enter something like `/var/named/db.cache`.
3. The **Get root servers from** field controls where Webmin copies the root file from. The choices are:

Download from root FTP server This is the best option, as it tells the module to make an FTP connection to `rs.internic.net` and download the latest version of the file. However, this may not work if your system cannot make perform FTP downloads due to a firewall.

Use Webmin's older root server information This option should be used if the first will not work. If selected, the module will use a copy of the root zone file that comes with Webmin, which will work but may not be up to date.

Existing root servers in file If the file entered in Step 2 already exists, then this option should be chosen. If you are adding a root zone to a view and one already exists in another view, it will be selected by default so that the file can be shared between both zones.

4. Click the **Create** button to add the zone and return to the module's main page. Then hit **Apply Changes** to make it active.

Once a root zone has been added, an icon representing it will appear on the main page. You can delete it by clicking on the icon and hitting the **Delete** button; however, this may prevent the lookup of records in non-hosted Internet domains from working as explained above.

30.11 Editing Zone Defaults

If you add lots of zones that contain similar records, then it can be a lot of work to add them manually after creating each one. For example, in a web hosting company all of your domains might contain a `www` Address record for the IP address of your web server and an Mail Server record that directs mail to a central server. Fortunately, Webmin allows you to create a list of records that get added to all new domains, called a zone template.

A template consists of one or more records, each of which has a name, type and value. For Address records, the value can be an option which indicates that it can be entered by the user at zone creation time. This is useful if one of the records (such as `www`) in the new domains does not have a fixed address, and you want to be able to easily set it when the zone is added. Templates can only be used when creating forward zones, as they do not make much sense for reverse zones.

It is also possible to edit the default expiry, refresh, TTL and retry times for new zones. Webmin's initial defaults are reasonable, but may not be appropriate for your network. To change these defaults and set up template records, the steps to follow are:

1. On the module's main page, click on the **Zone Defaults** icon. The form at the top of the page labeled **Defaults for new master zones** contains all the fields that need to be edited.

2. Edit the **Refresh time**, **Transfer retry time**, **Expiry time** and **Default time-to-live** fields if you want to change the defaults times for new zones. Existing master zones will not be effected by any changes you make here though.
3. If all your new domains are managed by the same person, enter his address into the **Default email address** field. This will save you from having to type it in on the master zone creation page every time.
4. In the **Template records** table, two blanks rows appear for entering new records. To add more than two, you will need to save this page and re-edit it. The records in existing rows can be edited by just changing their fields, or deleted by clearing out the record name.
Under the **Record name** column you must enter the name of the record relative to the zone, such as *www* or *ftp*. To create a record for the zone itself (such as a Mail Server record for the domain), just enter a single dot.
Under the **Type** column, select a type for the record from the list. See Section 30.5 “Record Types” for more information on what each is used for.
As its name suggests, the field under the **Value** column is for entering a value for the new record. For the Address type, you can select **From form** in which case you will be able to enter an address when creating a new domain, which will be used by all template records that have this option selected. For Mail Server records, both the priority and server name must be entered separated by a space, such as *5 mail.example.com*. Values for records of all other types should be entered in the same format as is used when adding a record to a zone.
5. If you are familiar with the records file format used by BIND, you can create your own file of records to be included in new zones. If a filename is entered into the **Additional template file** field, its contents will be added to the zone file created by Webmin for new master domains.
6. When you are done adding template records, click the **Save** button at the bottom of the page. The changes will apply to any new master zones created from now on.

Now that you have created a template, you can choose whether or not to use it for each new master zone that you create. On the creation form (explained in Section 30.3 “Creating a New Master Zone”) is a field labeled **Use zone template?**, which is set to **Yes** by default if there are any template records. Next to it is a field named **IP address for template records**, which used for entering the IP for records for which the **From form** option is selected. If you chose to use a template and if there are any records that do not have an IP address specified, then this field must be filled in.

The Zone Defaults page also contains several options that apply to all existing domains, but can all be set or overridden on a per-zone basis as explained in Section 30.6 “Editing a Master Zone”. You can control which clients are allowed to query the server and what kind of checking is done for the records of various domain types. Being able to limit the allowed client hosts is particularly useful, so that you can stop non-internal clients using your DNS server. However, you should make sure that master Internet zones hosted by your server are accessible to everyone, so that other DNS servers on the Internet can look them up.

To change these global options, the steps to follow are:

1. On the module's main page, click on the **Zone Defaults** icon and scroll down to the **Default zone settings** section.
2. To control which hosts are allowed to query your DNS server, change the **Allow queries from** field to **Listed** and enter a list of IP addresses, IP networks (like *192.168.1.0/24*) and ACL names into the text box below. Clients that do not match any entry on the list will be denied, unless they are requesting a record in a zone which has its own separate settings allowing them.
3. To control which hosts are allowed to perform zone transfers from your server, change the **Allow transfers from** field to **Listed** and fill in the text box below with a list of IP addresses, IP networks and ACL names. Only servers that are acting as secondaries for zones that this server hosts really need to be able to do transfers, so it is usually a good idea to enter just their IP addresses. If you are restricting queries, this field must be filled in so that hosts that cannot lookup records are not allowed to perform transfers either.
4. The fields **Check names in master zones?** and **Check names in slave zones?** control the checking of records in all zone files for master and slave zones respectively. The available options for each are:
 - Warn** If an invalid record is found, an error will be written to the system log file, but processing of other records continues normally.
 - Fail** Invalid records cause the entire zone to be rejected, but other zones will still be processed normally.
 - Ignore** No checking is done at all.
 - Default** The default checking level is used, which is **Fail**.
5. To have BIND check responses that it receives from other DNS servers, set the **Check names in responses?** field to **Warn** or **Fail**. The default is simply to pass potentially erroneous responses on to clients.
6. The **Notify slaves of changes?** field determines whether BIND sends a notification to all slaves of master zones hosted by this server when they change. To turn this on, select **Yes**. Otherwise, select **No** or **Default**. Enabling notification is a good idea, as it ensures that secondary servers are kept in sync with the master.
7. When done, click the **Save** button at the bottom of the page to update the BIND configuration file and then the **Apply Changes** button on the module's main page to make the changes active. The new settings will apply to all zones that do not explicitly override them on their own options pages.

30.12 Configuring Forwarding and Transfers

BIND can be configured to forward all requests for zones that it is not the master or slave for to another DNS server. When doing this, it acts like a DNS client itself, accepting requests from real clients and then sending them off to another server or servers for resolution instead of carrying out the normal process of contacting the root zone servers and finding the correct server for the domain. This can be useful if your DNS server is unable to contact the rest of the Internet, but can still communicate with a DNS server that does have full network access. For example, it may be on an internal network behind a firewall that only allows connections to a limited set of destinations.

To set up forwarding, the steps to follow are:

1. On the module's main page, click on the **Forwarding and Transfers** icon.
2. In the form that appears, fill in the **Servers to forward queries to** field with the IP addresses of DNS servers that requests should be sent to. BIND will try them in order until one returns a positive or negative a response.
If the list is empty, the server will revert to the normal method of looking up records by contacting the root servers and so on.
3. If you want your server to attempt to resolve a client's query directly when it cannot contact any of the forwarding servers, set the **Lookup directly if no response from forwarder** field to **Yes**. This is only useful if your server is actually capable of doing lookups.
4. Click the **Save** button at the bottom of the page, and then hit **Apply Changes** back on the main page to make the new setting active. Assuming the forwarding list was filled in, your server will now send all client queries to the listed servers.

The same form also contains fields for configuring BIND's behavior when doing zone transfers. You can control how long it will wait for a transfer to complete, the protocol used for transfers and the number that can be active at the same time. To edit these settings, follow these steps:

1. On the module's main page, click on the **Forwarding and Transfers** icon.
2. By default, BIND will wait 120 minutes (2 hours) for a zone transfer from a master to complete. To change this, enter a different number of minutes into the **Maximum zone transfer time** field. This can also be set or overridden on a per-slave zone basis.
3. BIND versions before 8.1 only support the transfer of a single zone at a time. Because this can be slow when transferring many zones from the same master server, the **Zone transfer format** field can be set to **Many** to use a new format that combines multiple domains into the same transfer. If **One at a time** or **Default** is chosen, then each zone will be transferred separately. This is the best choice unless you are sure that all slave servers are running BIND 8.1 or above.
4. By default, your name server will not carry out more than 2 concurrent zone transfers from the same master server. To increase this limit, change the **Maximum concurrent zone transfers** field to something higher. This can speed up the process of transferring a large number of domains, but at the expense of putting a higher load on the master server.
5. Click the **Save** button when you are done making changes and then click **Apply Changes** on the main page to activate them. The new settings will apply to all subsequent zone transfers.

30.13 Editing Access Control Lists

An access control list (or ACL) is a list of IP addresses, IP networks or other ACLs that are grouped together under a single name. The ACL name can then be used when specifying the list of clients allowed to query, update or perform zone transfers from a zone. This can be used to reduce the amount of duplication in your BIND configuration, and to make it clearer. For example, the ACL *corpnet* might match the IP networks *192.168.1.0/24*, *192.168.2.0/24*, and *1.2.3.0/24*, which are all part of your company's network. When configuring who can query a zone, you could just enter *corpnet* instead of that list of network addresses.

To view and edit ACLs in Webmin, the steps to follow are:

1. On the module's main page, click on the **Access Control Lists** icon. This will take you to a page listing existing ACLs and allowing the addition of one more. If you want to add more than one ACL, you will need to save the form and re-edit it to force the display of a new blank row.
2. To add a new ACL, find the blank row at the bottom of the table and enter a short name consisting of only letters and numbers in the **ACL Name** column. Then in the field under **Matching addresses, networks, and ACLs**, enter a list of IP addresses, IP networks and other ACL names that this new ACL will contain.
IP addresses must be entered in their standard format like *192.168.1.1*, but hostnames are not allowed. IP networks must be entered in network/prefix format like *192.168.1.0/24* or *192.168.1/24*. You can also precede any address, network or ACL name with a ! to negate it. For example, the entry *!192.168.1.0/24* would match all hosts outside the *192.168.1* network.
3. Existing entries in the list can be edited by changing their fields in the table and ACLs can be deleted by clearing out the field containing their names.
4. When you are done adding and editing ACLs, click the **Save** button. To activate the changes, hit **Apply Changes** back on the main page. After an ACL is created it can be used in the query and it can transfer and update restrictions of master and slave zones.

BIND has four built-in ACLs that can be used in all the same places that user-defined ACLs can. They are:

`any` Matches any client address.

`none` Matches nothing.

`localhost` Matches the IP addresses of all network interfaces on your system. Even though it is called `localhost`, it doesn't just match `127.0.0.1`.

`localnets` Matches all clients on all networks that your system is directly connected to. BIND works this out by looking at the IP addresses and netmasks of all network interfaces.

30.14 Setting Up Partial Reverse Delegation

Partial reverse zone delegation is method for transferring the management of a small set of reverse IP addresses to another DNS server. Normally, reverse zones cover an entire class C network containing 256 addresses. However, many organizations have networks much smaller than this, containing maybe 16 or 32 addresses. Normally, this would make it impossible for the organization to manage its own reverse address mappings, as the addresses come from a network that is owned by an ISP or hosting company.

Fortunately, there is a solution. The ISP can set up Name Alias (CNAME) records in the reverse zone for the parent network that point to Reverse Address records in a special zone on the organization's DNS server. The parent zone must also contain a Name Server (NS) record for the special sub-zone that points to the customer's server, so that other DNS clients know where to look when resolving the Name Alias records.

An example may make this clearer. Imagine for example that an ISP had granted addresses in the range *192.168.1.100* to *192.168.1.110* to Example Corporation, which owns the *example.com* domain. The company already runs its own DNS server to host the *example.com* zone, but wants to control reverse address resolution for its IP range as well. The ISP would create Name Alias records in the *192.168.1* zone pointing to the special sub-zone *192.168.1.100-110*, which will contain the actual Reverse Address records named like *192.168.1.100-100.101*. The ISP also needs to create a Name Server record for *192.168.1.100-110* which tells other servers that Example Corporation's DNS server should be used to find records under that zone.

Webmin handles reverse address delegation well and automatically converts special network zones like *192.168.1.100-110* to and from the real zone names used by BIND such as *100-110.1.168.192.in-addr.arpa*. The exact steps to follow on both the server that hosts the actual class C network zone and the server that a subset of it is being delegated to are:

1. Decide on the range of addresses that are being delegated, such as *192.168.1.100* to *192.168.1.110*. Typically, the sub-zone name is based on the range of addresses being delegated, but this does not have to be the case as long as it is under the parent network domain.
2. On the server that hosts the class C network zone, add a Name Server record for *192.168.1.100-110* with the server set to the IP address or name of the sub-zone's DNS server.
3. For each address in the range, add a Name Alias record to the reverse zone named like *101.1.168.192.in-addr.arpa*, with the **Real Name** set like *101.100-110.1.168.192.in-addr.arpa*. As you can see, the alias points to a record inside the zone for the sub-network.
4. When all of the Name Alias records have been created, everything that needs to be done on this server is finished and you can hit **Apply Changes**.
5. On the DNS server for the sub-network, create a new master zone for the reverse network *192.168.1.100-110*. Webmin will automatically convert this to the correct *in-addr.arpa* format for you.
6. Add Reverse Address records to the new zone as normal for IP addresses like *192.168.1.100-110.101*. Adding a record for the IP *192.168.1.101* will not work.
7. When you are done creating reverse records, click the **Apply Changes** button to make them active. You should now be able to look them up using a command like *nslookup* on the server for the parent network zone.

The instructions above can be used to delegate multiple ranges from a single class C network to several different DNS servers. There is no limit on the size of ranges, nor any requirement that they follow normal network block boundaries; however, for routing reasons most IP allocation is done in power-of-two sized blocks (like 4, 8, 16 and so on), which means that any sub-zone ranges will be the same size.

The only problem with reverse address delegation when using Webmin is that Reverse Address are not automatically created and updated when Address records are. This means that you will have to create all such records manually on the sub-zone server, as in the steps above.

One inconvenience in setting up partial reverse delegation is the number of similar Name Alias records that must be created on the parent network zone server. Fortunately, there is a sim-

pler alternative—record generators. A generator is a special BIND configuration entry that creates multiple similar records using an incrementing counter. This module allows you to create and edit generators, by following these steps:

1. On the module's main page, click on the icon for the reverse zone that you want to create records in. This will typically be a class C network domain that is going to have a range of addresses delegated to some other server.
2. Click on the **Record Generators** icon. This takes you to a page containing a table of existing generators, with a blank row for adding a new one.
3. In the empty row, select **CNAME** from the menu under the **Type** column.
4. Under the **Range** column, enter numbers for the start and end of the address range into the first two fields, such as *100* and *110*. The third field is for entering the gap between each step and should be left blank. If you were to enter *2*, then the range would go *100, 102, 104* and so on.
5. In the **Address pattern** field, enter *\$* (a single dollar sign). When the records are created, the *\$* will be replaced with the number of each record, which will in turn resolve to an IP address in the range.
You could also enter *\$.1.168.192.in-addr.arpa.*, which makes things more obvious but is longer to type.
6. In the **Hostname pattern** field, enter *\$.100-110*. Similarly, the *\$* will be replaced with the number of each record, which will resolve to something like *101.100-110.1.168.192.in-addr.arpa.*
7. If you like, enter a comment that describes what this generator is for into the **Comment** field.
8. Click the **Save** button, return to the module's main page and click on **Apply Changes**.

A generator can replace the Name Alias records that the first set of instructions in this section tell you to create, so there is no need for them anymore. Note that the automatically generated replacements will not be visible or editable in the normal way, only through the Record Generators page.

30.15 Using BIND Views

BIND version 9 introduced the concept of views, which are groups of zones that are visible only to certain DNS clients. Views can be used to hide internal zones from the Internet, to present the same zone in two different ways, or to stop non-local clients resolving non-hosted domains through your server. Every view has a unique name, and a list of matching IP addresses and IP networks that determines which clients and servers it is visible to.

When it detects that you are running BIND 9, several additional features are available in the module. You can create views, move zones from one view to another, and choose which view zones are created in. On the main page, each current view is represented by an icon under **Existing Client Views** heading and each zone icon has a label that indicates which view it is in.

If any views exist, then every zone must be in a view. If none are defined will Webmin allow the creation of zones outside views, as this is not supported by BIND. This includes the root zone, which must be available to a client for DNS requests for records in domains not hosted by

this server to succeed. For this reason, it often makes sense to put the root zone in a view that is available to all clients.

To add a new view to your BIND configuration, the steps to follow are:

1. On the module's main page, click on the **Create a new view** link in the **Existing Client Views** section. This will take you to a form for entering its details.
2. Enter a short alphanumeric name for the view (such as *internal* or *everyone*) into the **View name** field. Each view must have a unique name.
3. Leave the **DNS records class** field set to **Default**.
4. If the zones in this view are to be visible to everyone, set the **Apply this view to clients** field to **All clients**. Otherwise, choose **Selected addresses, networks, and ACLs** and enter a list of IP addresses, IP networks and BIND ACL names into the text box below. Only clients that match one of the entries in this list will have access to the view.
5. Click the **Create** button at the bottom of the form. You will be returned to the main page, which will include an icon for your new view.
6. Move any existing zones that you want to be in this view into it. A zone can be moved by clicking on its icon, then on **Edit Zone Options**, and then selecting the new view from the menu next to the **Move to view** button before clicking it.
If this is your first view, all existing zones must be moved into it (or another view) before the new configuration will be accepted by BIND.
7. When you are done moving zones, click the **Apply Changes** button on the main page.

Once a view has been created, you can change the list of addresses and networks that it matches by clicking on its icon on the main page and updating the **Apply this view to clients** field. Then hit the **Save** button followed by **Apply Changes** to make the new client list active.

A view can be deleted by clicking the **Delete** button on the same form. This will bring up a confirmation page that allows you to choose what should happen to the zones that it contains, if any. The available options are:

Delete totally All zones in the view are deleted, along with their records files.

Move out of views Zones in the view are moved out to the top level. This option should only be used when deleting the last view, for the reasons explained above.

Move to view Zones are moved to a different existing view.

When one or more views have been defined on your system, you can choose which one to use when adding new zones. This is done using the **Create in view** field on the master, slave, forward and root zone creation forms, which allows you to select a view from its menu. Naturally, there is no option for creating a zone outside of any views as this is not allowed by BIND.

One common use of views is hiding internal zones from clients outside your internal network. This is a good way of hiding the structure of your network and the hosts on it from potential attackers. To set it up, the steps to follow are:

1. Create a new view called *internal* that matches clients on your internal LAN.
2. Create a second view called *everyone* that matches all clients.

3. Move any zones that are for internal use only into the *internal* view. Zones for Internet domains such as *example.com* must not be put in this view, as that would make them inaccessible to the rest of the world.
4. Move all other zones (including the root zone) to the *everyone* view.

Views can also be used to prevent clients outside your network looking up non-hosted domains on your server, as follows:

1. Create a new view called *internal* that matches clients on your internal LAN.
2. Create a second view called *everyone* that matches all clients.
3. Move the root zone to the *internal* view, which will prevent the server from looking up records for non-local clients that require contact with the root servers.
4. Move all other zones to the *everyone* view.

30.16 Module Access Control

Like others, the BIND DNS Server module allows you to control which of its features are available to a particular Webmin user or group. This can be useful for giving people the rights to manage only records in their own zones and nobody else's. Even though this would normally require `root` access to the records files, with Webmin it can be granted to people without giving them level of power that a `root` login would allow.

Once you have created a user with access to the module as explained in Chapter 52, the steps to limit his access to only certain zones are:

1. Click on the BIND DNS Server next to the name of the user in the Webmin Users module. This will bring up a page of access control options.
2. Change the **Can edit module configuration?** field to **No**, so that the user is not allowed to change the paths that the module uses to `named.conf` and other files.
3. For the **Domains this user can edit** field, choose **Selected zones** and select the ones that you want him to have access to from the list to its right. If you want him to be able to edit almost all zones, it may be better to choose **All except selected** and select only those that he should not be allowed to manage records in. If your DNS server uses views, you can use the **Zones in view** options to allow or deny access to all zones in a view as well.
4. Change the fields **Can create master zones?**, **Can create slave/stub zones?**, **Can create forward zones?** and **Can edit global options?** to **No**.
5. If you want Reverse Address records in zones that the user does not have access to to be updated by changes to Address records in zones that he does, set the **Can update reverse addresses in any domain?** field to **Yes**. This may not be a good idea from a security point of view though, as he would be able to change almost any existing Reverse Address record on your system. For that reason, I suggest that this field be set to **No**.
6. To stop the user creating more than one Address record with the same IP, set the **Can multiple addresses have the same IP?** field to **No**. Even though creating multiple records is harmless, you may want to set this to **No** to prevent the user allocating the same IP twice.
7. Leave the **Read-only access mode?** field set to **No**. If it is changed to **Yes**, the user will only be able to view zones and records using the module, and not change anything. This might

- be useful for creating a different kind of restricted user though—one who can see all settings, but not edit them.
8. Leave the **Can apply changes?** field set to **Yes**, so that he can use the **Apply Changes** button to make his additions and modifications active.
 9. Unless you want the user to be able to edit his records file manually, change the **Can edit records file?** field to **No**. Most untrusted users are not smart enough to perform manual editing.
 10. The **Can edit zone parameters?** field determines if the user can see and use the **Edit Zone Parameters** icon for his domains. Setting this to **Yes** is quite safe, as the user can only harm his own zones by setting the parameters to silly values.
 11. Similarly, the **Can edit zone options?** field determines if the **Edit Zone Options** icon is visible or not. You should set this to **No**, as it is possible for a user to create a syntax error in named.conf by improper use of the zone options form.
 12. Unless you want the user to be able to delete his own domains, change the **Can delete zones?** field to **No**. Users should contact the master administrator instead if they want to delete zones.
 13. The **Can edit record generators?** field can be left set to **Yes**, as it simply allows the creation of multiple records at once. However, some users may get confused by this feature so it might be a good idea to change the field to **No**.
 14. The **Can lookup WHOIS information?** And **Can search for free IP numbers?** fields can also be left on **Yes**, as those features merely display information to the user.
 15. Change the **Can create and edit views?** field to **No**, so that the user cannot manage BIND 9 views. If the user is allowed to create zones, you can use the **Views this user can edit and add zones to** field to limit those that he can create zones in.
 16. **Can create slave zones on remote servers?** should be set to **No**, but this doesn't really matter as the user is not going to be allowed to create master or slave zones anyway.
 17. Finally, click the **Save** button to make the new restrictions for the user active.

30.17 Configuring the BIND DNS Server Module

The BIND module has several options that can be set by clicking in the **Module Config** link on the main page. Those listed under **System configuration** control where Webmin looks for the BIND configuration file, PID file and program on your system, are initially set to match the BIND package that comes with your operating system. Normally they do not need to be changed, unless you have compiled and installed the DNS server software yourself.

Table 30.1 lists both the configurable options that you can safely change (in the first three sections) and those that are related to file locations which generally do not need to be edited (in the fourth). Most of the options only need to be changed by people who have customized their BIND setup or run large name servers. For the average sites, the defaults will work fine and there is no need to adjust the module configuration.

30.18 The BIND 4 DNS Server Module

Even though BIND version 8 has been available for several years now, version 4 is still in use by many people and is included as standard by even the latest release of HP/UX and possibly other operating systems. Fortunately, there is a Webmin module that supports BIND 4, which was

Table 30.1 Module Configuration Options

Chroot directory to run BIND under	For security reasons, some people like to run BIND limited to a single directory with the <code>chroot</code> command. If you are doing this, then Webmin will be confused by your configuration files unless this option is set to the directory that the server is restricted to, such as <code>/home/bind</code> . The module will then treat all configuration and record file paths as relative to this directory, just as BIND would. If you do not know what <code>chroot</code> is or are not using it, leave this option set to Default .
User to start BIND as	When this field is set to Default , the module will start BIND as <code>root</code> . However, you can enter a different username such as <code>named</code> to have it run as that UNIX user instead. This can prevent your system being taken over by an attacker who finds a bug in the DNS server program. Be sure that all zones files are readable by the user. The Owner for zone files option documented below can help with this.
Group to start BIND as	If Default is chosen for this field, the UNIX group that BIND runs as is determined by the primary group of the user set in the User to start BIND as field. If you enter a group name, the DNS server will be run as that group instead. If the previous field is set to Default though, it makes no difference what you select as BIND will always be run as the <code>root</code> user and <code>root</code> 's primary group.
Add new zones to file	Normally, Webmin will add new all new zones to the <code>named.conf</code> file. If this is not the way things are done on your system, you can enter a different filename for this field. However, for the new zones to be recognized by BIND and Webmin, <code>named.conf</code> must have an include directive to read this file.
Display domains as	If Icons is selected, the module's main page will display each zone as an icon. However, if you choose List instead zones will be shown in a table, which takes less space and is easier to read. This makes sense if you are hosting a large number of domains.
Order to display records in	This field controls the default sorting method used when viewing the list of records of some type in a domain. The available options and their meanings are: <ul style="list-style-type: none"> By name Records are sorted by name, or in the case of Reverse Address records by IP address. IP sorting is done the correct way, not simply alphabetically. By value Records are sorted by their value part. For Address records, their means sorting by IP address—all other types are sorted alphabetically. By IP Address and Reverse Address records are sorted by IP, others types by value. As added No sorting is done at all—records are simply shown in the order that they were added to the file.

Table 30.1 Module Configuration Options (Continued)

Maximum number of zones to display	If the number of zones hosted by your server exceeds the number set in this field, they will not be displayed on the module's main page. Instead a simple search form is shown for finding domains whose names contain the entered text.
Update reverse is	This field determines the default setting for the Update reverse? option on the Address record creation and editing forms. Normally it is set to On by default , but if you rarely want Webmin to automatically update reverse records you should change it to Off by default . This option also effects the Update forward? field on the form for creating and editing a Reverse Address, in exactly the same way.
Reverse zone must exist?	Normally, adding an Address record with an IP address in a reverse zone that is not hosted by this server is not a problem, even if Update reverse? is set to Yes . Sometimes though you do want Webmin to generate an error message in this case, so that you know that the entered IP address is incorrect. Setting this field to Yes turns on this behavior.
Support DNS for IPv6 addresses	If this field is set to Yes , the module will allow the creation and editing of records of a new type—the IPv6 Address. Because they are only useful if you are running an IPv6 network, this option is turned off by default. When editing or adding IPv6 Address records, the appropriate reverse address records will be updated and created as well. However, they will be in the special <code>ip6.int.</code> domain instead of <code>in-addr.arpa</code> .
Allow comments for records	When this field is set to Yes , an additional Comment field will be displayed on the form for adding and editing records. This allows you to enter a comment for the record, which will be displayed in the records list. These can be useful for adding additional notes to hostnames which are visible only to you, rather than to everyone on the Internet as would be the case with a comment in a Text record. In the records file, comments are added to the end of record lines using the BIND comment character <code>:</code> . This means that if you have existing comments in your files, they should show up when this option is enabled.
Allow wildcards (not recommended)	Normally, the module does not allow the <code>*</code> wildcard character to be used in record names as it is not well supported by some DNS servers and clients. If you do want to use wildcards (such as for a Mail Server record for all hosts in a domain) then you will have to set this field to Yes .

Table 30.1 Module Configuration Options (Continued)

Allow long hostnames	Normally Webmin prevents record names from exceeding 255 characters. When this field is set to Yes , you are allowed to create names of up to 635 characters long, which are supported by some versions of BIND. The length restriction applies to the complete canonical name of the record, not just to the short name that you might enter on the record creation form.
Allow underscores in record names?	The use of the <code>_</code> character in DNS names is not technically allowed by the protocol specification, but many DNS servers and clients support it. In fact, Windows systems often depend upon such records to operate properly. When this field is set to No the module will prevent you from creating such records, while selecting Yes will allow it.
Convert record names to canonical form?	When this field is set to Yes (as it is by default), Webmin converts the names of any new or updated records to canonical form before adding them to the records file. This means that relative names like <code>www</code> have their domain added, to become like <code>www.example.com.</code> , both when they are written to the records file and displayed in the module. The advantage of this approach is the elimination of records that have no name, and thus are dependent on the name of the previous record. However, this automatic conversion will cause problems if you have two zones that share the same records file. It can also be annoying if you like to edit records manually and prefer to use short names. To turn it off, change this field to No . The only downside is that the module's automatic updating of reverse address records may stop working for records with relative names.
Categorize zones by view?	By default, when using BIND 9 views the module's main page simply displays the name of its parent view under the icon for each zone. If this field is set to Yes , zones will be categorized by views instead so that you can more clearly see which zone belongs to which view.
Serial number style	When Running number is chosen for this field, Webmin will generate a serial number for new zones that starts with the current UNIX time number, and is incremented by one for each change. Selecting Date based instead forces the serial number to be in <code>YYYYMMDDnn</code> format, which uses the current date followed by an incrementing counter for the changes within the day. This section option generates serial numbers in the format that is required by some registrars, such as those in Germany. As far as BIND and the DNS protocol are concerned, there is no difference between the two methods.
Add \$ttl to top of new zone files	If Yes is chosen for this field, the module will add a <code>\$TTL</code> line to the top of all new records files that it creates. Newer versions of BIND log a warning message if this line is not present, but older versions complain if it is there, and some really old releases cannot handle it at all. If BIND on your system doesn't like <code>\$TTL</code> lines, then you will need to set this field to No .

Table 30.1 Module Configuration Options (Continued)

Directory for master zone files	When Default is selected, the module works out which directory to put new master zone files into from the <code>directory</code> line in the <code>named.conf</code> file. If you normally put master and slave files in separate directories, then the master directory should be entered into this field.
Directory for slave/stub zone files	Like the previous field, this one allows you to specify a different directory from the default for new slave zone record files.
Format for the name of forward zone files	This field determines the filename format that Webmin will use for new record filenames. An occurrence of <code>ZONE</code> in the filename will be replaced with the name of the new forward domain. If you do change this field because you like to use a different name format like <code>example.com.db</code> , make sure that the new value contains the string <code>ZONE</code> .
Format for the name of reverse zone files	This field has the same purpose as the previous one, but is used for reverse zone filenames instead of forward.
Owner for zone files (user:group)	This field controls the ownership of newly created record files. It must be entered in <code>user:group</code> format, such as <code>named:daemon</code> . If you are running BIND as some user other than <code>root</code> , this field should be changed so that the zone files created by Webmin are readable and editable by the DNS server user.
Permissions for zone files (in octal)	Like the previous one, this field controls the UNIX permissions on new record files. You must enter a 3-digit octal number of the kind that is used by the <code>chmod</code> command, such as <code>755</code> .
Default master server(s) for slave zones	The IP addresses entered into this field will be listed by default in the Master servers text box when adding a slave zone, and will be added to a zone's configuration when converting it from a master to a slave. This can be useful if you create lots of slave zones that get their records from the same master server.
Default remote slave server	This field determines the default Webmin server to add a slave zone to when adding a master zone. It is only used when using the module's cluster features, which are not covered in this chapter.
Automatically update serial numbers	Normally this field is set to Yes , which causes the module to automatically update a zone's serial number every time a record in it is changed. To turn off this behavior, change the field to No instead—however, this will cause problems with caching by secondaries and other DNS servers unless you have some mechanism to update the serial numbers separate, such as a script that runs once per day.

Table 30.1 Module Configuration Options (Continued)

Domain for reverse IPv6 addresses	This field is only relevant if you are using the module to manage IPv6 address and reverse address records. It determines which root domain is used for reverse addresses—either the old ip6.int , or the new ip6.arpa . If any such zones already exist on your system, you will need to make the right choice here for the module to behave properly.
Full path to the named.conf file	This field determines where the module looks for the primary BIND configuration file, <code>named.conf</code> . You should only need to change it if you have compiled and installed the DNS server software yourself, and chosen to use a different location for the configuration file such as <code>/usr/local/etc/named.conf</code> .
Full path to the named executable	If you have installed the BIND server program in a different location to the default expected by Webmin, then you will need to change this field. This may be the case if the server has been compiled and installed manually.
Full path to whois command	The module uses the <code>whois</code> command to display ownership information about a domain which you click on the Lookup WHOIS Information icon. This field must contain the path to the command on your system, such as <code>/usr/local/bin/whois</code> .
Command to reload a zone	When the Apply Changes button is clicked on a master zone's options page, the command set in this field is used to signal BIND to reread the zone's records file. By default the <code>rndc</code> command is used, which communicates with BIND via a socket file. However, you may want to use <code>rndc</code> instead, which can communicate via a network connection.
Default PID file location	To determine if BIND is running, the module looks for a PID file containing its process ID. Normally the path to this file is specified in <code>named.conf</code> , but if not the code will use the path in this field instead. If you have compiled and installed the server yourself, you may need to change this to something different like <code>/usr/local/var/named.pid</code> .
Command to start BIND	If Webmin detects that BIND is not running, a button will be displayed on the module's main page so that you can start it. If Default is chosen the <code>named</code> executable is run directly, but an alternate command can be used instead. On some operating systems, this field is set by default to a bootup script like <code>/etc/init.d/named start</code> . If you have compiled and installed BIND yourself, you should change it back to Default as the script is unlikely to work properly if it exists at all.

written long before version 8 became available. In fact, it was the very first module to be written, and was the inspiration for the rest of the package.

BIND 4 lacks many of the features of version 8, such as zone options that control who can query and transfer records, generators, forward zones, stub zones, change notification, and many

global options. However, it can still perform the basic tasks of hosting forward and reverse master and slave zones.

Version 4 uses `/etc/named.boot` as its primary configuration file, which has a totally different format to versions 8 and 9. The records files for zones are still in the same format though, which makes it relatively easy to upgrade to the latest version of BIND if necessary. The types of records that are supported are the same, with the exception of the Location and IPv6 Address types.

Because BIND 4 is rarely seen these days, its icon will only appear under the Servers category if the `named.boot` file exists on your system. When you enter it, the main page displays only a table of zone icons, a form for setting new zone defaults and a button for either starting the server or applying changes.

The form for adding a master zone is pretty much the same as in the BIND DNS Server module, except that zone templates are not supported. The slave zone creation form is identical, except for the rarely used **Server port** field. The process of editing records in a master zone is the same, and automatic updating of reverse zones works in the normal way. You can never view records in a slave zone though.

Master zone parameters from the SOA record can be edited using a form below the list of record types in a zone, which is identical to the form on the Zone Parameters page in the BIND 8 module. These are the only parameters that you can edit for a master zone, as BIND version 4 does not support any other associated options. When editing a slave zone, only the list of master servers and the records filename can be changed.

The process of deleting a master or slave zone is the same in the BIND 4 DNS Server module, and the removal of reverse or forward records in other zones is supported. You cannot view or delete the root zone, however, even though it exists in the configuration file, the module never displays an icon for it.

30.19 Summary

This chapter has introduced the Domain Name System and explained the roles of DNS clients and servers. It has also introduced the popular BIND server and described how Webmin can be used to configure it. The chapter has covered the creation of various types of domains, the management of records within master domains, and the use of advanced DNS features such as record generators and partial reverse delegation. After reading it you should know everything necessary to set up your system as a DNS server.

CVS Server Configuration

If you have already have a CVS repository on your system, this chapter explains how to make it available to other hosts by setting up a network-accessible CVS server.

31.1 Introduction to CVS

CVS stands for Concurrent Versions System and is a set of programs that allows multiple developers to work on the same source code without interfering with each other. This chapter assumes that you are already familiar with the workings of CVS and want to know how to set up a server that allows a repository to be accessed over the network. A server allows people to check out code to their personal computers, work on it, and then check it back in again over the network.

There are actually several ways of making a repository network-accessible, such as via SSH or RSH. The method that Webmin supports is the running of CVS in pserver mode from a super server like inetd or xinetd. There is no actual separate CVS server process that runs all the time in the background, like Apache or an NFS daemon. Instead, the CVS program is run by inetd only when a client CVS program connects to it.

To control who can access files, a CVS server uses username and password authentication. The supplied login can be checked against the system UNIX user list or a separate file of usernames and passwords. Users can be given full read-write access to the repository or read-only access to prevent them from checking in files.

31.2 The CVS Server Module

This module can be found in Webmin under the **Servers** category. When you enter it, the main page displays a list of icons as shown in Figure 31.1. Almost all of the actual forms and pages for configuring the server can be reached by clicking on the icons.

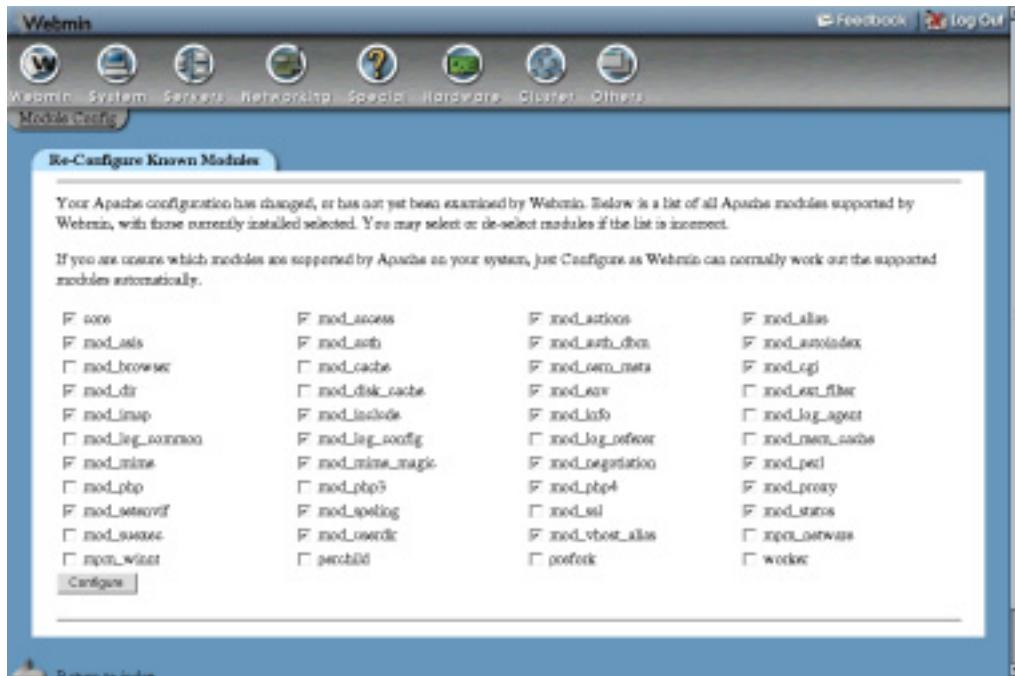


Figure 31.1 The CVS Server module.

If a CVS repository is not found on your system, the main page will display an error message instead. By default, Webmin looks in `/usr/local/cvsroot`, but this is unlikely to be correct on most systems. To change the path, click on **Module Config** and update the **CVS root directory** field.

The module's main page can also display an error message if the cvs command is not found on your system, or if its version number cannot be determined. If the command is not in Webmin's search path, you will need to adjust the **Path to CVS executable** field on the module configuration page.

31.3 Setting Up the CVS Server

If the CVS server has not yet been set up on your system, a button labeled **Setup CVS Server** will be displayed at the bottom of the module's main page. When clicked, an entry will be added to the `inetd` or `xinetd` configuration (covered in Chapter 15) to run the server program `cvs pserver` on the standard TCP port 2401. The current CVS root directory will be supplied as a parameter to this command, so make sure that it is set correctly before clicking the button. Clients should now be able to connect to the repository, as explained in Section 31.4 "Using the CVS Server".

The server is always set up to run as a service called `cvspsvc`, which runs as `root` with no IP access control restrictions. If you are using `xinetd`, the Extended Internet Services module can be used to restrict access to the service. Because most repositories only need to be accessible to CVS

clients on a company network, this is usually a good idea. It is also possible to control access to the service if you are using `inetd` through the use of TCP wrappers, but that must be done manually.

When the server is active, the button will be replaced with one labeled **Deactivate CVS Server** instead. If clicked, the appropriate entry in the `inetd` or `xinetd` configuration will be disabled so that clients can no longer connect. To enable it again, hit the **Activate CVS Server** button that will appear on the main page instead.

If you have both `xinetd` and `inetd` installed, then the former will be used when setting up the server for the first time. When setting up, activating, or deactivating the server, the module will automatically signal `inetd` or `xinetd` to reread its configuration file, so there is no need to click the **Apply Changes** button in the appropriate module.

31.4 Using the CVS Server

Once your server is running, CVS clients can connect to it using a repository path like `:pserver:username@hostname:repository`. Naturally, `username` must be replaced with the name of a user that the CVS server accepts, `hostname` with the hostname of your system, and `repository` with the correct repository directory.

To check out a directory, you could run commands like:

```
CVSROOT=:pserver:jcameron@fudu.webmin.com:/usr/local/cvsroot
export CVSROOT
cvs login
cvs checkout path/to/directory
```

The `cvs login` command should only be run if the server requires a password for the username. It will prompt you for a password, which is permanently stored in the file `~/.cvspass` for use by other CVS commands in the future. Any time the `cvs` command is run to update, check out, commit to, or access the repository in any way, it will make a connection to the server specified in the `CVSROOT` environment variable and log in with the username from the variable and the specified password.

31.5 Adding and Editing Users

The CVS server requires that all clients supply a valid username and password before they can perform any operations on the repository. Users are authenticated against a list of accounts used only by the server and optionally against the standard UNIX user database from the `/etc/passwd` and `/etc/shadow` files. Using only the server's user list for authentication is the best option, as it gives you more control over who is allowed to connect. See the documentation for the **Check users against system password file?** field in Section 31.7 “Configuring the CVS Server” for more information on turning on or off UNIX authentication.

For each CVS user, you can specify the name of a UNIX user whose permissions will be used for all file access. This can be useful if all files in your repository are owned by a single user, in which case all CVS access should be done as that user. This avoids the problem of ensuring that file permissions in the repository are set correctly so that people can edit and read each others' files.

To add a new login to the CVS user list, follow these steps:

1. On the module's main page, click on the **CVS Users** icon. This will take you to a page containing a table of all existing users, as shown in Figure 31.2.
2. Click on the **Add a new CVS user** link above or below the table to go to the user creation form.
3. Enter a unique name for the user into the **Login username** field. A UNIX user of the same name must already exist. If not, the **Access files as UNIX user** field must be set.
4. The **Login password** field is for entering the user's password. The available options are:
 - None required** If selected, the CVS server will accept any password or none at all when a client connects as this user.
 - Copy from UNIX** If chosen, the user's password is initially set to that of the UNIX user with the same name.
 - Set to** If selected, you must enter a password for the user into the field next to it.
5. The **Access files as UNIX user** field controls which UNIX user the CVS server will switch to when this CVS user connects. If **Same as username** is selected, then the UNIX user with the same name will be used. If the other option is chosen, however, you must enter a UNIX username into the adjacent field to access files as instead.
6. Click the **Create** button to have the user added to the server's list. You will be returned to the table of users, which should include your new entry. If you choose for it to access files as a different UNIX user, the name will be shown in brackets next to the CVS login name.

Once a user has been created, you can edit it by clicking on its name in the table on the CVS users page. The editing form is the same as the one used for user creation, except that the **Copy from UNIX** option in the **Login password** field is replaced with **Leave unchanged**, which must be selected if you are not changing the password. When you are done editing a user, click the **Save** button. Or to delete the user, click on **Delete** on the editing form instead.

This page can also be used to set up synchronization between the CVS user list and the UNIX user database. This can be useful for keeping passwords in sync, or for having a CVS user deleted when the corresponding UNIX user is removed. Chapter 4 explains in more detail how user synchronization works in Webmin, but follow these steps to set it up for this module:

1. On the main page, click on the **CVS Users** icon and then scroll down to the form below the list of existing users.
2. To have a new CVS user created for each UNIX user created in Webmin, select the **Add a new CVS user when a UNIX user is added** checkbox. Below it is a field labeled **Access files as UNIX user** that you can use to set the UNIX access user for automatically created CVS users.

This option is most useful when all users access repository files as a single UNIX user. Otherwise, you might as well just allow all UNIX users to log in to your CVS server.
3. To have the password or username of a CVS user changed when the corresponding UNIX user is updated, check the **Update a CVS user when the matching UNIX user is modified** checkbox.



Figure 31.2 The CVS users list.

4. To have a CVS user deleted when the UNIX user of the same name is removed, check the **Delete a CVS user when the matching UNIX user is deleted** box.
5. Click the **Save** button to save the new synchronization settings. They will apply to users created, edited, and deleted in the Users and Groups module from now on.

31.6 Limiting User Access

This section explains how to configure the CVS server to limit which users are allowed to write to the repository. This can be useful if you want to give anyone on the Internet access to your source code. For example, if you are hosting an open-source project. A user with no password but read-only access could be created for anyone in the world to log in as, while developers log in with password-protected accounts that have full read-write privileges.

To restrict write access to your server, follow these steps:

1. On the module's main page, click on the **User Access Control** icon. This will take you to a form listing users who currently have read and write access.
2. To give read-only access to only a few users and read-write to the rest, select the **Listed users are read-only** radio button. Then, enter the names of those users who should not be able to write to the repository into the left-hand text box provided. On the right, the option **All users can write** should be selected.

Alternately, to give read-write access to a few users and read-only to everyone else, select **Only listed users can write**. Then, enter the names of those users who will be

able to write to the CVS repository into the right-hand text box. On the left, **No read-only users** should be selected.

There is not much point entering usernames into both boxes, as any user who is in both lists will be given read-only access.

3. Click the **Save** button to make your changes immediately active and return to the module's main page.

31.7 Configuring the CVS Server

There are a few options related to authentication and logging that can be set for the CVS server. In most cases the defaults will work fine, but you can change them by following these steps:

1. On the main page of the module, click on the **Server Configuration** icon to go to a form for editing server options.
2. To allow any UNIX user to log in to the server, set the **Check users against system password file?** field to **Yes**. If **No** is selected, only those users defined on the CVS users page will be able to access the repository.
3. The **Event types to log in history** field controls what kind of client activity is logged to the CVSROOT/history file. You can either select **All types** to log everything or **Selected types** to log the kinds of events that are checked below.
4. To change the directory in which CVS puts lock files, edit the **Lock files directory** field. This is necessary if you have read-only users who cannot write to the repository, as when **Default** is selected lock files will be created within the repository itself.
5. Click the **Save** button to have the server begin using the new settings.

31.8 Browsing the Repository

One handy feature of the CVS Server module is the ability to view files and directories in the repository. You can do this by clicking on the **Browse Repository** icon on the module's main page, which will take you to a page listing the contents of the repository root directory. A list of modifications to a file can be viewed by clicking on its name, and the latest checked-in version is displayed by clicking on the number under the **Rev** column.

The listing can be sorted by clicking on column headings, such as **Age** or **Author**. Below the listing is a menu labelled **Show only files with tag** for restricting the listing to files tagged with the chosen name. Tags are often used to identify code for a branch or version of a project, or from a particular source.

31.9 Configuring the CVS Server Module

This module has a few settings that can be changed by clicking on the **Module Config** link on the main page. Those that you can safely edit are shown in Table 31.1.

Table 31.1 Module Configuration Options

CVS root directory	This must be set to the directory that contains your CVS repository. The module will check for a CVSROOT subdirectory and display an error on the main page if it is not found. There is currently no way to specify multiple root directories, even though the CVS server can be manually configured to support them.
Repository browser header file	This field can be used to specify a file containing HTML to be displayed at the top of each page in the repository browser. The default file, <code>header.html</code> , contains information about the authors of the browser script. You may want to enter the full path to some other file instead, perhaps containing information about the files on your CVS server.
Character set for repository file	Normally, Webmin chooses a character set for each page based on the currently selected language. This information is sent to the browser, which uses it to work out the encoding of text on the page and the font with which to display it. This may not be appropriate when browsing the repository, however, as your source code probably just uses the <code>us-ascii</code> or <code>iso-8859-1</code> character set. This field can be used to specify an alternate character set to be used on the browser pages if they are not being displayed properly when Webmin is using a different language. Most people can just leave it set to From Webmin language .

31.10 Summary

If your system hosts a CVS repository, the instructions in this chapter can be used to make the files it contains available to other hosts. After reading it you should understand how to create and edit CVS users, how to restrict what certain users can do, and how to browse the files that your repository contains.

DHCP Server Configuration

This chapter tells you what DHCP is and how to use Webmin to set up a DHCP server on your network so that other systems can obtain IP addresses automatically.

32.1 Introduction to the Dynamic Host Configuration Protocol

DHCP is a protocol that allows hosts to request and be assigned an IP address on a local area network. It is used to simplify the process of IP assignment, as a single server can manage the addresses of multiple clients. It is also useful for systems like laptops that are moved between multiple networks, as they do not need to be reconfigured for each LAN to which they are connected.

DHCP is usually used on Ethernet networks, although it can be used on any type of LAN that supports broadcast traffic such as 802.11b and Token Ring. It is not used for address assignment for dial-up connections—the PPP protocol has its own method of telling clients their IP addresses. Because broadcasts are not normally forwarded by routers, a DHCP server can only assign addresses to hosts on a single LAN, unless you have a router that is configured to forward DHCP packets.

A DHCP server can also supply other information to clients in addition to an IP address. The addresses of DNS servers and the network gateway can be sent, along with the DNS domain, NIS server, NIS domain, static routes, and much more. DNS and routing information allows clients to fully integrate themselves into the network to which they are connected without needing any manual configuration.

When a server assigns an IP to a client, it is given a lease on that address for a certain amount of time, during which no other client will be assigned the same address. When the lease expires, the client must contact the server again. Typically, it will be assigned the same IP address as before and the lease will be extended for the same time period. If a client does not

contact the server when its lease is up, the server assumes that the client has been shut down and marks the address as available to be assigned to other hosts.

Most operating systems include support for configuring a network interface to use DHCP to get its IP address. Chapter 16 explains how to set it up for Linux systems, and it is relatively simple to configure Windows and Mac OS clients to use it as well. DHCP has become the standard protocol for address assignment on IP networks, replacing the older BOOTP protocol used by some UNIX operating systems.

32.2 The ISC DHCP Server

The most common DHCP server for UNIX system is the ISC server, of which several versions have been released. The latest is version 3, but version 2 is still in common use. Release 1 uses a very different configuration file format to later versions and is not seen much anymore. The ISC DHCP server supports a wide range of options and can be configured to behave differently for different clients, networks, and address ranges.

The ISC server can be used to assign fixed addresses to hosts or addresses from certain ranges. Every host is identified by its MAC addresses, which on an Ethernet LAN is the address on the host's Ethernet card. A static IP address and other options can be associated with a particular hardware address, which allows you to fix the address that certain systems receive while using dynamic allocation for others.

The server's configuration file contains four different types of entries, which contain options that affect different clients:

Subnet A subnet is an entire IP network, such as *192.168.1.0*. Entries of this type are used to dynamically allocate addresses within certain ranges to clients within the network.

Shared network A shared network is a group of subnets that share the same physical network.

Host This is a single client host identified by its MAC address and assigned a fixed IP address.

Group This is a group of hosts for which the same options can be set.

Entries in the server configuration are arranged in a hierarchy that determines what client options and other settings apply to a particular client. Options in higher-level entries are overridden by those lower in the hierarchy, which allows an administrator to avoid repeating configuration information while still being able to set individual options for specific hosts. Figure 32.1 below shows which kinds of entries can be defined under which other types.

The ISC DHCP server's primary configuration file is called `dhcpd.conf` and can usually be found in the `/etc` directory. Other configuration files can be included by the primary file, but on most systems only `dhcpd.conf` is used. The only other file used by the server is `dhcpd.leases`, which contains all granted leases and is always kept up-to-date. Whenever the server is started, it rereads this file to find out which leases are currently active. This means that there is no danger of lease information being lost if the server is stopped and restarted, which is necessary for it to reread the primary configuration file.

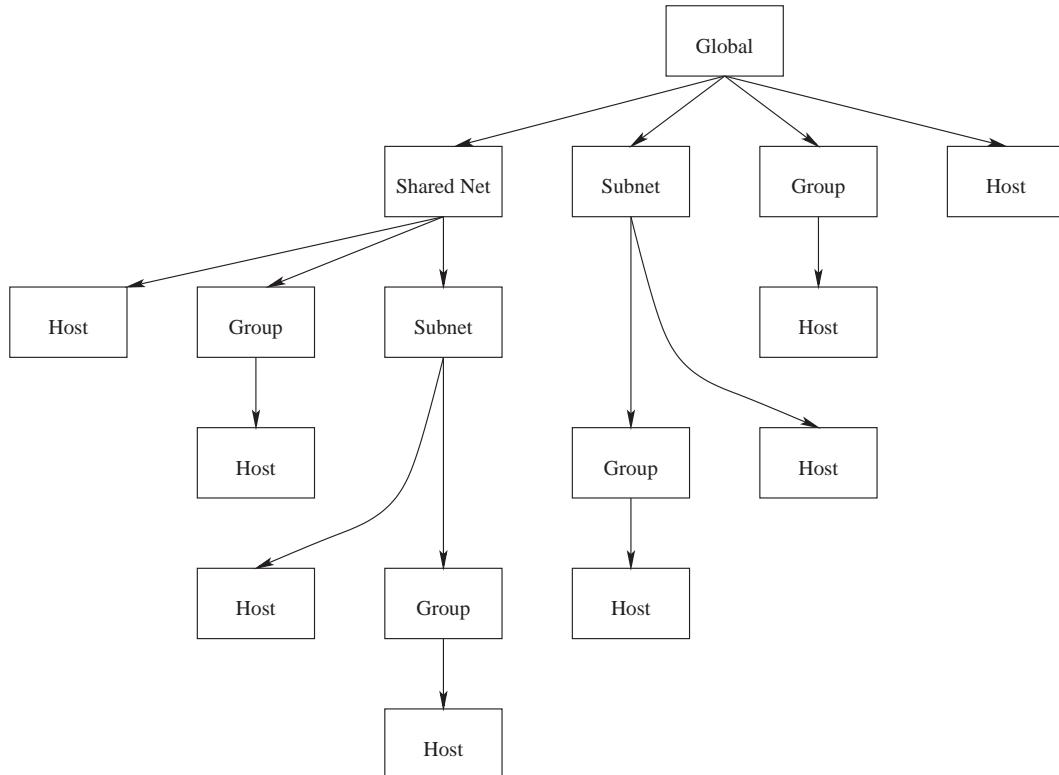


Figure 32.1 The DHCP Server configuration hierarchy.

Webmin's DHCP Server module directly updates the configuration and lease files when you manage subnets, hosts, groups, and leases. To activate the current configuration, it kills the server process and reruns it, as there is no way to signal the server to reread its configuration file.

32.3 The DHCP Server Module

This module can be used to set up your system as a DHCP server so that clients on your LAN can be automatically assigned IP addresses, DNS servers, and other information. If there is already a server on your network, setting up another one is a bad idea as they may interfere with each other. If you just want to configure your system to obtain its own IP address via DHCP, then there is no need to set up a server. Instead, see Section 16.3 “Adding a Network Interface”.

The DHCP Server module can be found in Webmin under the **Servers** category. Clicking on its icon will take you to the main page, which lists all existing subnet, shared network, host, and group configurations. Figure 32.2 shows an example. If this is the first time that you have used the module, however, and the server has not been configured manually, then the page will probably be almost empty.

At the bottom of the page are buttons for editing global settings and displaying current dynamic address leases. Below them is the **Start Server** or **Apply Changes** button, which either

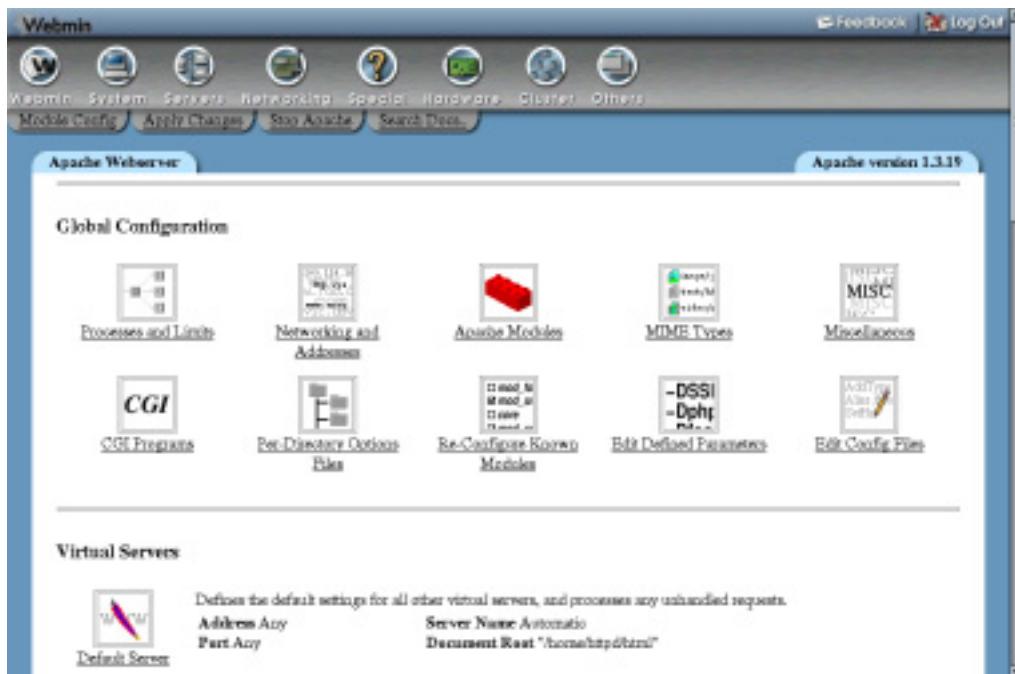


Figure 32.2 The DHCP Server module.

starts the server if it is not running or restarts it to force a reload of the configuration if it is running. You cannot, however, start the server until at least one valid subnet has been defined.

If the ISC DHCP server is not installed on your system, the main page will display an error message notifying you that the `dhcpd` program could not be found. All Linux distributions include a DHCP server package on their CD or website, which you will need to install before you can use the module. Make sure that the package you add is called `dhcpd` or `dhcp-server`, as there is often a separate package for the DHCP client programs.

The same error can also appear if the server is installed, but in a location other than the one that the module expects. This can happen if you have compiled and installed it yourself from the source code, rather than using your distribution's standard package. If so, you will need to adjust some of the paths explained in Section 32.11 "Configuring the DHCP Server Module".

Because this module only supports the configuration of ISC DHCP server versions 2 and 3, the main page will also display an error message if it detects that version 1 of the server is installed. Unfortunately, this older release uses a totally different configuration file format and so cannot be managed by the module. Some operating systems (such as Solaris) include this older version by default, but it can be replaced by the latest one.

The ISC DHCP server is also available for several other UNIX operating systems in addition to Linux. Because it works the same on all of those systems, the behavior of this module is identical as well. The only differences are the default paths that it uses for the server configuration files and programs.

On some operating systems and Linux distributions, the DHCP server package includes a sample configuration file that defines several hosts and subnets. These are not going to be of much use for your network and will probably prevent the server from working at all as they do not match its actual network interfaces. For this reason, it is best to simply delete them before setting up your own configuration.

Once a few entries have been added to the server configuration, the main page displays a table of icons networks under the heading **Subnets and Shared Networks**. Each icon represents either a subnet (shown with its network address under it) or a shared network (shown with its name). By default, subnets are listed first, followed by shared networks, and both lists are in the order that they appear in the configuration file. If you have a complex DHCP configuration, you can change this by clicking on one of the following links next to **Display nets and subnets by:**

Assignment The default sorting mode. Subnets are shown before shared networks and both are listed in the order in which they appear in the configuration file.

File structure Subnets are listed after the shared networks of which they are part, which are sorted by their order in the configuration file.

Name/IP address Subnets are listed, sorted by IP address, followed by shared networks sorted by name.

In the bottom part of the page is a table of icons with the heading **Hosts and Host Groups**. An icon is shown for each host or host group, with the name or number of members displayed beneath it. Because many servers have a large number of hosts, you can control the order that they are displayed in by clicking on one of the following links next to **Display hosts and groups by:**

Assignment Hosts are listed before groups and both are in the same order in which they appear in the configuration file.

File structure Hosts are listed after the groups of which they are part, which are sorted by their order in the configuration file.

Name Hosts are listed, sorted by name, followed by groups in the order in which they appear in the configuration file.

Hardware address Hosts are listed, sorted by MAC addresses, followed by all groups.

IP address Hosts are listed, sorted by their fixed IP address, followed by all groups.

Changes to the sorting modes will be remembered by the module, so that they will be used every time you visit the main page from now on.

32.4 Adding and Editing Subnets

In the simplest DHCP server configuration, all you need is a single subnet entry to hand out IP addresses within particular range to clients on a single LAN. The server allows you to do much more than that, but for many networks this is all that is needed. Unless, of course, you want to assign fixed addresses to some hosts or have multiple IP networks on the same LAN.

To add a new subnet entry, follow these steps:

1. On the module's main page, click on the **Add a new subnet** link in the **Subnets and Shared Networks** section. This will take you to the page shown in Figure 32.3.
2. In the **Network address** field, enter the address of your local LAN, such as *192.168.1.0*. This must be a network to which your system is directly connected.
3. In the **Netmask** field, enter the mask for the local LAN, such as *255.255.255.0*. The best way to find the correct network address and netmask is to use the Network Configuration module to look at the settings for your Ethernet interface.
4. The **Address ranges** section is actually a table for entering multiple ranges, but only one blank row is displayed at a time. In the first field, enter the starting address for the range of IPs that you want assigned to clients, such as *192.168.1.100*. In the second, enter the ending address for the range, such as *192.168.1.150*. Both addresses must be within the network, and the first must be lower than the second.

To add more than one range, you will need to reedit this subnet after saving so that a new blank row appears in the table. The server will always assign addresses from the start of the first range up to the end, then go on to the second and any subsequent ranges. Because each client must have a unique IP, make sure that your ranges are big enough to support all the client hosts that may be connected to the network at any one time.

5. If you want this subnet to be part of a shared network (explained in Section 32.8 “Adding and Editing Shared Networks”), select it from the **Shared network** menu. Otherwise, choose **<None>** to have the subnet created outside of any shared nets.
6. To set the lease length for clients on this network, change the **Default lease time** from **Default** to the number of seconds by adding that number into the field next to it. This will be the length of the lease for hosts that do not explicitly request one.

You should also set the **Maximum lease time** field, so that clients cannot request a lease longer than the specified number of seconds. If not set, there is no upper limit on lease length.

7. Unless the client systems on your LAN will be network-booting from another server, the **Boot filename** and **Boot file server** fields can be left set to **Default**. Only diskless workstations need to do this.
8. The **Server name** field is for entering the network hostname of your DHCP server system. Usually this can be left set to **Default**, in which case the server will work it out automatically.
9. Click the **Create** button at the bottom of the page. A new entry for the subnet will be added to the server's configuration, and you will be returned to the module's main page.
10. Click on the new icon for the subnet, which will take you to an editing form that is almost identical to the creation page.
11. Click on the **Edit Client Options** button to go to a page listing information that will be sent to clients, as shown in Figure 32.4. All of the fields have a **Default** radio button, which, if selected, typically indicates that no information related to that option will be sent to clients.
12. Fill in the **Default routers** field with the IP address of the default gateway on your network, such as *192.168.1.1*. This will be used by clients that have their address assigned by DHCP to communicate with systems outside the network.

13. Fill in the **Subnet mask** field with the netmask for your network, such as 255.255.255.0.
14. Enter the broadcast address for your network into the **Broadcast address** field, such as 192.168.1.255.
15. Fill in the **Domain name** field with the DNS domain name such as *example.com* that clients should append to partial hostnames.
16. In the **DNS servers** field, enter a space-separated list of DNS server IP addresses that clients can use, such as 192.168.1.104 1.2.3.4.
17. If you are running NIS (covered in Chapter 17) and want clients to connect to an NIS server at boot time, fill in the **NIS domain** field with the name of your NIS domain, and fill in the **NIS servers** field with the IP address of your NIS master or slave server. This is only useful if the client hosts are capable of getting their NIS settings from DHCP.
18. If you have Windows clients and are running a Samba or Windows server, fill in the **Net-BIOS name servers** field with the IP address of a system that can do NetBIOS name resolution for clients. Any UNIX system running Samba will be able to perform this role.
19. Click the **Save** button at the bottom of the page to go back to the subnet form.
20. If this is your first subnet, you will need to make sure that the server is configured to use the right network interface for your system. Return to the module's main page and click on the **Edit Network Interface** button at the bottom of the page.
Then, select the interface for the new subnet from the **Listen on interfaces** list and click **Save**. If you have multiple network interfaces and have created subnet configurations for each of them, then all the interfaces must be selected for the server to work properly.
21. If you are running version 3 of the ISC DHCP server (shown on the main page) and this is your first subnet, you may need to set the DDNS update style before the server can be started. Even if you are not using DDNS, some versions insist on an entry existing in the configuration file for it. Click on the **Edit Client Options** button on the main page and scroll down to the **Dynamic DNS update style** field. Select **None** and click **Save** to return to the module index.
22. Back on the main page, click on the **Start Server** or **Apply Changes** button. If something goes wrong, the error message generated by the DHCP server will be displayed. The most common problem is a mismatch between the network interface settings and the network address for the subnet. Another that often shows up is related to the ddns-update-style directive, which Step 21 explains how to set.

Once your first subnet has been created and the server started, you can test it by configuring a client system to use DHCP. When the client boots up, it should contact the server and be assigned an address, DNS, and routing information. You should also be able to see the client on the leases page, covered in Section 32.5 “Viewing and Deleting Leases”.

An existing subnet can be edited by clicking on its icon on the main page, changing fields, and hitting the **Save** button. If you want to edit options for clients in the subnet, you will need to click on **Edit Client Options** as in the instructions above, make your changes, and then click **Save** on that page. After any modifications, the **Apply Changes** button must be used to make them active.

A subnet can be deleted using the **Delete** button on its editing form. Any hosts, groups, or address pools that it contains will be removed as well—so be careful. After deleting, use the network interfaces page to deselect the interface for the subnet. Failure to do so will cause the

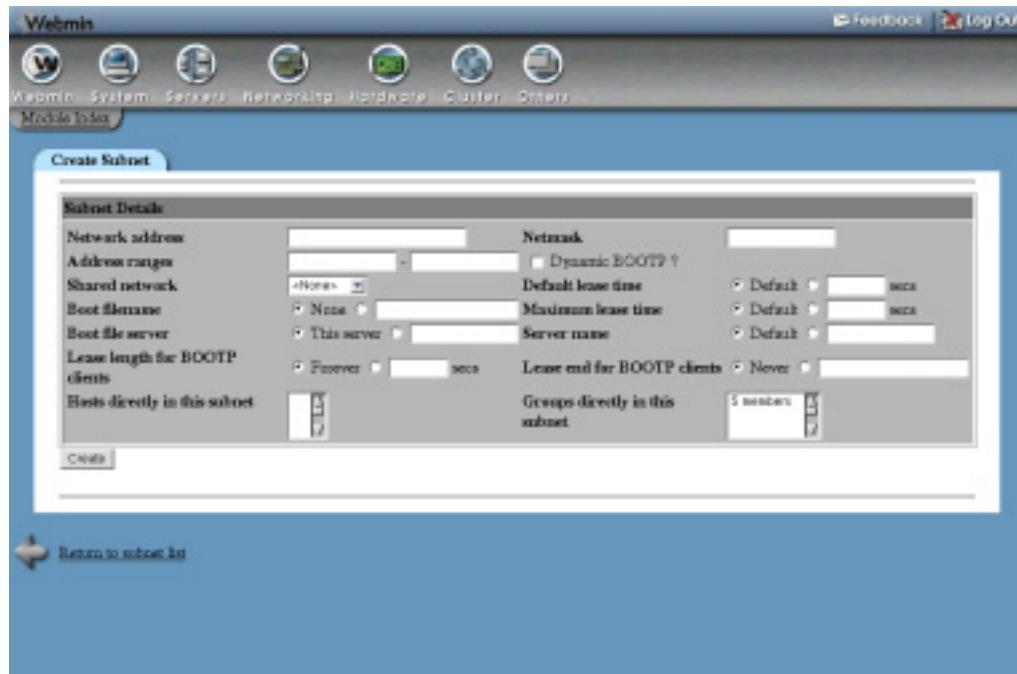


Figure 32.3 The subnet creation form.

DHCP server to display an error message when **Apply Changes** is clicked, which must be done to make the deletion active.

If the subnet contains any hosts or groups, a confirmation page will be displayed when **Delete** is clicked that lists all the groups and hosts that will be deleted as well. Only when the **Yes** button is hit will the subnet (and all it contains) actually be removed.

Another way to create a subnet inside a shared network is to click on the **Add a new subnet** link on the shared network's page. This will bring up the same subnet creation form shown in Figure 32.3, but without the **Shared network** field. Instead, the shared network is shown at the top of the page under the title. The rest of the creation process is identical.

A subnet configuration entry must be created for each IP network on which you want to allocate addresses. Typically, there will be one for each LAN connected to your system via an Ethernet, Token Ring, or 802.11b network card. If two IP networks are actually on the same LAN, then both their subnets must be inside a shared network, as explained in Section 32.8 "Adding and Editing Shared Networks".

You must also make sure that every network interface that is connected to a network on which your DHCP server is assigning addresses is selected on the network interface page. If not, an error will be reported when the server is started or when changes are applied. For most system administrators, this is not a big issue though as they have only a single LAN in their organization.

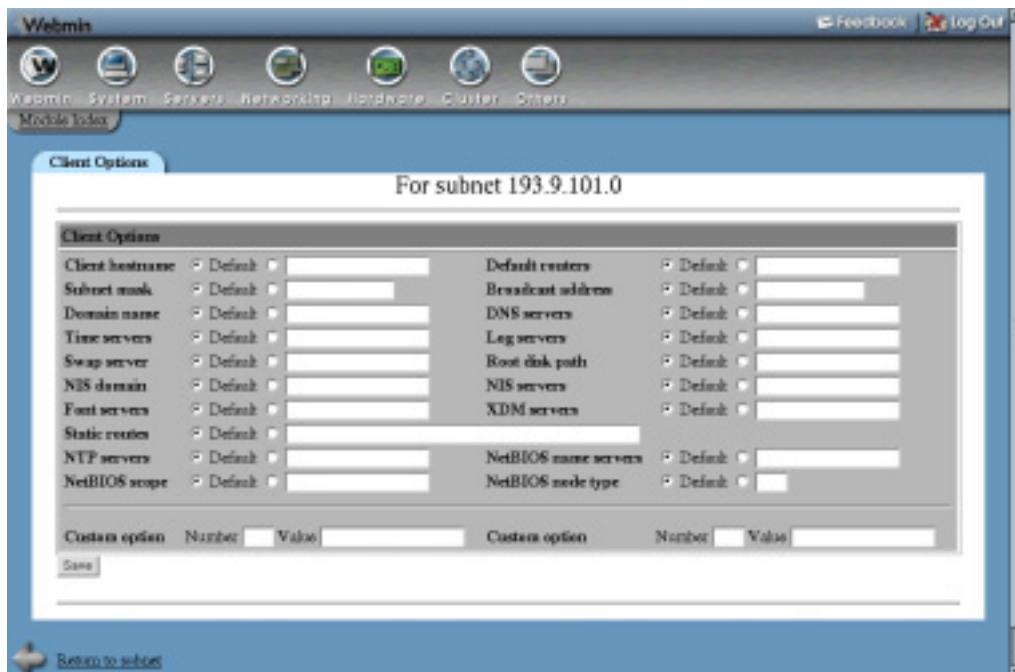


Figure 32.4 The subnet client options page.

32.5 Viewing and Deleting Leases

Every time the DHCP server supplies a dynamic address to a client, it records information about the assignment in its lease file. Fixed addresses assigned to specific hosts (covered in Section 32.8 “Adding and Editing Shared Networks”) do not trigger the creation of a lease, as they are considered permanent. You can use this module to view all current leases or expired leases, and to delete those that exist. Removing a lease tells the server that its IP address is no longer in use and can be assigned to some other client. This should only be done if the client really isn’t using the address any more, for example, if it crashed while holding a long lease.

To view and delete leases, follow these steps:

1. On the module’s main page, click on the **List Active Leases** button. This will display a table listing all currently active leases, with the IP address, client name, and start time shown for each.
2. To show leases that have expired as well, click on the **List all active and expired leases** button at the bottom of the page.
3. To remove a lease, click on its IP address in the list. The DHCP server will be stopped and restarted automatically to make the deletion active.

It is also possible to view the leases to clients in just a single subnet by clicking on the **List Leases** button on the subnet editing form. This can be useful if you have several networks connected to your system with a large number of clients and want to limit the size of the lease display.

32.6 Editing Global Client Options

Section 32.8 “Adding and Editing Shared Networks” explains how to set client options (such as DNS and gateway IP addresses) that are supplied to all clients in a subnet. If you have more than one network or many fixed hosts, however, it can be more convenient to set options that apply to all clients of the server. These options can still be overridden for individual subnets, hosts, and groups if you wish.

To edit global client options, follow these steps:

1. Click on **Edit Client Options** near the bottom of the module’s main page. This will take you to a form similar to the one shown in Figure 32.4.
2. Change any of the fields as explained in Steps 11 through 18 of Section 32.4 “Adding and Editing Subnets”.
3. At the bottom of the form are fields for setting the default and maximum lease times for all clients, along with a few other options. These have the same meanings as similarly named fields on the subnet creation page.
4. Click the **Save** button to update the DHCP server configuration file and return to the module’s main page.
5. Hit the **Apply Changes** button to make your new settings active.

Client options specified for a subnet override those defined globally, and are in turn overridden by options for hosts within the subnet.

32.7 Adding and Editing Fixed Hosts

If you want to fix the IP address that is assigned to a specific host, you will need to add a host entry to the DHCP server configuration. This also allows you to set client options that apply only to that host, such as the DNS server addresses or default router.

The server identifies hosts by their MAC (Medium Access Control) address, which on an Ethernet LAN is the Ethernet address of the client’s network card. Typically this address is fixed, but a few network cards allow it to be changed. On Linux systems, you can find the MAC address by running the `ifconfig eth0` command as `root` and looking for a string of 6 bytes in hex separated by colons, like `00:D0:B7:1D:FB:A1`. On Windows, the `winipcfg` program can provide the information, although it is displayed with dashes instead of colons. Other operating systems have their own ways of finding the Ethernet address.

Once you know the MAC address of the host, it can be added to the DHCP server configuration as follows:

1. On the module’s main page, click on the **Add a new host** link in the **Hosts and Host Groups** section. This will bring up the host creation form shown in Figure 32.5.
2. Enter a name into the **Host name** field. This should match the hostname that the client is configured with, or its fully qualified name on your network. This, however, is not mandatory.
3. Select the type of network (such as Ethernet) that the host is on from the menu in the **Hardware address** field. In the text box next to it, enter the host’s MAC address as a series of 6 hex bytes separated by colons, like `00:D0:B7:1D:FB:A1`.

4. Enter the IP address that should be always assigned to this client into the **Fixed IP address** field.
5. If you want this host to inherit client options from a subnet, select **Subnet** from the menu in the **Host assigned to** field. The list next to it will be filled in with the names of all existing subnets, allowing you to select the one under which the host should be located. The fixed IP address must be within the subnet's network, however, and the client must be connected to its LAN.

Hosts can also be created inside shared networks or host groups by choosing **Shared Network** or **Group** from the menu and selecting the appropriate entry from the list to the right.
6. If this host needs to network boot from a server, enter the name of that server into the **Boot file server** field. You must also fill in the **Boot filename** field with the path to an appropriate boot file (downloadable via TFTP) on the server.

Generally, network booting is used by simple clients like X terminals and diskless workstations. For it to work, you must set up a TFTP server that contains the correct boot files for the client, which is not covered in this chapter.
7. Click the **Create** button at the bottom of the form, and you will be returned to the module's main page, which will now include an icon for the new host.
8. To edit the client options that are assigned to this host, click on its icon to go to its editing page, then on **Edit Client Options**. This is not always necessary if the host is a member of a subnet that already has these options set, or if they have been defined globally as explained in Section 32.6 "Editing Global Client Options".
9. Fill in the form as you would for a subnet, as explained in Section 32.4 "Adding and Editing Subnets".
10. Click the **Save** button to return to the host form.
11. Return to the main page, and hit the **Apply Changes** button. From now on the host will be assigned the IP address and options that you have chosen. It will no longer appear on the lease list, as its IP assignment is permanent.

Once a host has been created, you can change its fixed IP address, MAC address, and other options by clicking on its icon on the module's main page, which will take you to the host editing form. After making modifications, hit **Save** to update the server configuration and then **Apply Changes** to make them active. A host can also be deleted with the **Delete** button on the editing page. From then on, the client system will receive a dynamically allocated address from one of the ranges for its subnet rather than a fixed address.

A host can also be created by clicking on the **Add a new host** link on the subnet, shared network, or group editing page. If done this way, the **Host assigned to** field is no longer displayed on the creation form. Instead, the parent to which it will be added is shown at the top of the page. All the other steps in the process of adding the host are the same.

If you have a large number of hosts and want all of them to use the same client options, then they should be placed in a group or shared network. See the Section 32.9 "Adding and Editing Groups" for more information on group management. The DHCP server configuration allows you to define several levels of groups, which allows for quite complex configurations. If you have more than one fixed address host on your network, they definitely should be under a subnet or group to avoid duplicating settings.

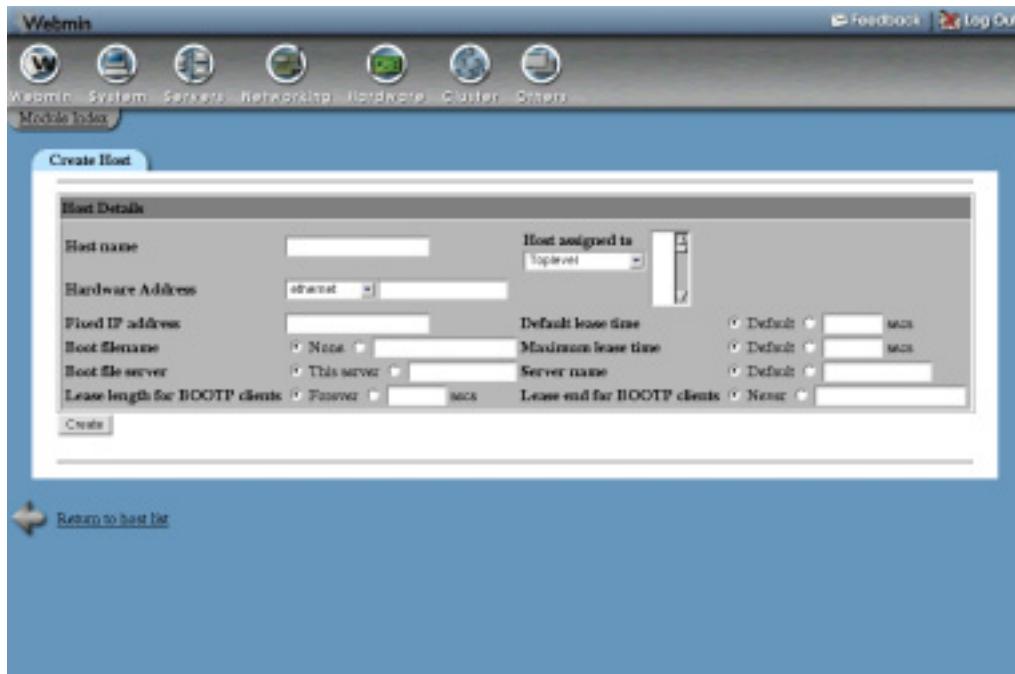


Figure 32.5 Creating a new host.

32.8 Adding and Editing Shared Networks

A shared network is a group of subnets that share the same physical LAN. If you have multiple IP networks on the same physical network, the DHCP server configuration entries for all of them must be placed inside a shared network. Failure to do so may cause the server to behave incorrectly or report an error message when started. On the other hand, you cannot put subnets that do not share the same LAN in the same shared network.

It is also possible for a shared network to contain a single subnet, although this does not really achieve anything. It may be useful for grouping configuration entries, however, as a shared network can contain hosts and groups as well and have client options that apply to all its members.

To create a shared network, follow these steps:

1. On the module's main page, click on the **Add a new shared network** link under **Subnets and Shared Networks**.
2. Enter a short name for the network into the **Network name** field, such as *homelan*. This is used only when displaying the shared network on the main page.
3. To set the lease lengths for all clients of subnets under this shared network, fill in the **Default lease time** and **Maximum lease time** fields. Their meanings are the same as on the subnet creation form, documented in Section 32.4 “Adding and Editing Subnets”.
4. In the **Subnets in this shared network** field, select any existing subnets that you want to move into this shared network. All existing subnets—including those in other shared net-

works—will be listed. You must choose at least one subnet, as a shared network cannot be empty.

5. Click the **Create** button at the bottom of the page. Your new shared network will be added to the server's configuration and an icon for it will appear on the module's main page.
6. If you want to set client options that will apply to all subnets in the shared network, click on its icon and then on **Edit Client Options**. Set any of the fields that you want and then hit **Save** to return to the shared network form.
7. Click the **Apply Changes** button to make it active.

Once a shared network has been created, subnets can be created in or move to it using the **Shared network** field on the subnet form. The same field can also be used to move a subnet out of any shared networks by selecting the <None> option.

Once a shared network has been created, it can be renamed or edited by clicking on its icon, changing fields, and hitting the **Save** button. It can also be removed altogether with the **Delete** button. Trying to delete a shared network that contains subnets, hosts, or groups will bring up a confirmation page asking if you really want to go ahead. If you click **Yes**, all the configuration entries that the shared network contains will be deleted as well. As usual, after making changes or deleting, you must click the **Apply Changes** button on the main page to activate the new settings.

32.9 Adding and Editing Groups

Unlike subnets, hosts, and shared networks, group entries in the DHCP server configuration do not actually effect the server's behavior in any way. Instead, they are just used to define options that will apply to multiple hosts. Even though there are other ways that this can be achieved (such as putting the hosts under a subnet), using a group gives you extra flexibility.

Groups can be defined under subnets and shared networks, but not other groups. Groups do not normally have names. Instead, they are identified in Webmin by the number of hosts that they contain. Newer versions of the DHCP server do support group names and Webmin will display them, but there is no support yet in the module for setting the name of a group.

To create a new host group, follow these steps:

1. On the module's main page, click on the **Add a new host group** link under **Hosts and Host Groups** to go to the group creation form.
2. Select any existing hosts that you want to be members of this group from the **Hosts in this group** list.
3. If you want this group to be under a subnet, choose **Subnet** from the menu in the **Group assigned to** field and select the subnet in the list next to it. All hosts in the group must have fixed IP addresses that fall within the subnet's network.

Similarly, a group can be created inside a shared network by choosing **Shared Network** from the menu and selecting the network name from the list. In both cases, the group will inherit client options and other settings (like the lease length) from its parent subnet or shared net.

4. If hosts in the group need to network boot from a server, enter the name of that server into the **Boot file server** field. You must also fill in the **Boot filename** field with the path to an appropriate boot file (downloadable via TFTP) on the boot server.

5. Click the **Create** button. You will be returned to the module's main page, which will now include an icon for the new group.
6. Click on the group icon to bring up its editing form, and then on **Edit Client Options**. This will take you to the page shown in Figure 32.4 for setting options that are sent to client hosts in this group.
7. Set any of the options such as the DNS or NIS servers by following Steps 10 through 18 of Section 32.4 "Adding and Editing Subnets".
8. Click the **Save** button at the bottom of the page to save the options and return to the group form.
9. Go back to the module's main page and hit **Apply Changes** to make your new group active.

Once a group has been created, new or existing hosts can be moved into it using the **Host assigned to** field on the host form. Any host added to a group will inherit client options and network boot settings from the group, unless overridden by settings for the host itself.

As usual, a group can be edited by clicking on its icon on the module's main page, making changes, and clicking **Save**. A group can also be removed with the **Delete** button on its editing page. However, if it contains any hosts, you will be asked to confirm the deletion before it and the hosts are actually removed.

A group can also be created under a subnet or shared network by clicking on the **Add a new host group** link on the page reached by clicking on one of their icons. The group creation form that is displayed no longer has a **Group assigned to** field. Instead, the name of the subnet or shared network to which it will be added is displayed at the top of the page. Apart from that difference, the preceding instructions can still be followed.

32.10 Module Access Control

As Chapter 52 explains, the Webmin Users module can be used to limit what a user or group can do with a particular module. For this module, you can control exactly which hosts, groups, subnets, and shared networks a user can edit. This can be useful for granting a subadministrator the right to set options for only a few hosts within your server configuration, while preventing him from changing subnets and other hosts.

Once a user has been given access to the module, limit him to editing only certain hosts by following these steps:

1. In the Webmin Users module, click on **DHCP Server** next to the name of the user. This will bring up the module access control form.
2. Change the **Can edit module configuration?** field to **No** so he cannot edit the configuration file path and the commands that the module uses.
3. Leave **Can apply changes?** set to **Yes** so he can activate any changes that he makes.
4. Change **Can edit global options?** to **No** so he cannot change options that apply to all clients.
5. **Can view leases?** can be safely left set to **Yes**, but **Can remove leases?** should be set to **No**.
6. The **Uniq host names?**, **Uniq subnet IP addresses?**, and **Uniq shared-net names?** fields should be changed to **Yes** to prevent the creation of clashing hosts, subnets, and shared networks.

7. The **Use security level** field determines to which configuration entries in the hierarchy the user is allowed access. The available options and their meanings are:
 - Level 0** The user will have access to all entries to which he has been granted.
 - Level 1** The user will have access to granted entries, as long as he can access all their children as well.
 - Level 2** The user will have access to granted entries, as long as he can access all parent and ancestor entries.
 - Level 3** Like levels 2 and 3 combined. Generally, you should leave this option set to level **0** for simplicity's sake.
8. Assuming you are limiting the user to editing only certain hosts, deselect all three options in the **Access groups** and **Access shared nets** fields. This will stop the user from viewing and editing any groups or shared networks.
To stop the user from creating hosts and subnets, deselect **create** in the **Access hosts** and **Access subnets** fields.
9. Change the **Enable per-subnet ACLs?** and **Enable per-host ACLs?** fields to **Yes**. This allows you to select exactly which hosts and subnets the user can access from the **Per-object ACLs** section provided.
If the first of these fields is set to **No**, the **Access subnets** checkboxes determine whether the user can view and edit all subnets. Similarly, if the **Enable per-host ACLs?** field is set to **No** then the **Access hosts** checkboxes control the viewing and editing of all hosts.
10. In the **Per-object ACLs** section, select **read/write** for any hosts and subnets that the user should be able to configure and **not allowed** for the rest. Choosing **read only** will allow him to view the host or subnet without being able to change it.
11. Finally, click the **Save** button at the bottom of the page to make the new restrictions active.

Another common use of the DHCP Server module's access control page is limiting a user to the viewing and cancelling of leases only. This can be done by setting the **Can view leases?** and **Can remove leases?** fields to **Yes** and everything else to **No**. The user should also be denied access to all hosts, subnets, and so on, or possibly given read-only permissions.

32.11 Configuring the DHCP Server Module

Like most others, the DHCP Server module has several options that can be set by clicking on the **Module Config** link on the main page. They are divided into two groups—those under **Configurable options** control the module's user interface, while those under **System configuration** tell the module where to look for the server's configuration files and programs. Because the latter set of options are set automatically by Webmin based on your operating system or Linux distribution, you should not need to change them unless you have compiled and installed the DHCP server yourself.

Table 32.1 covers the user interface and DHCP server path options.

Table 32.1 Module Configuration Options

Sort leases by	This field controls how the list of leases is sorted. The available options are: Order in file Leases are displayed in the order in which they appear in the leases file, which normally means they will appear in the order in which they were granted. IP address Leases are sorted by the IP addresses assigned to the clients. Hostname Leases are sorted by client hostnames, which are not necessarily related to their IP addresses. The hostname is determined by the client itself, and reported to the server when a lease is obtained.
Display subnets and hosts as	When Icons is selected (as it is by default), all subnets, shared networks, hosts, and groups on the main page are shown as icons. Choosing List instead changes the display to a text table, which contains more information and uses less screen space.
Icons in row	When using the icon display mode, this field determines the number of icons that appear in each row of the subnet and host tables.
Display lease times in	When Local time is selected for this field, times in the lease table are shown in the time zone of the server system. Changing it to GMT forces GMT to be used as the time zone instead, which may make more sense when administering a server in a different country.
Show IP addresses for hosts?	When this field is set to Yes , the fixed IP address of each host will be shown under its icon along with the hostname on the module's main page.
Show MAC addresses for hosts?	When Yes is selected, the Ethernet address of each host will be shown under its icon on the main page. It is possible to have both the IP and MAC address displayed.
DHCP server config file	This field must be set to the full path to the DHCP server's configuration file, <code>dhcpd.conf</code> . You should only need to change it if you have compiled and installed the server manually instead of using the version that comes with your operating system.
DHCP server executable	This field must contain the full path to the DHCP server program, usually called <code>dhcpd</code> . When starting or restarting the server, Webmin runs this command with the appropriate command-line arguments for the configuration file, lease file, and interfaces.

Table 32.1 Module Configuration Options (Continued)

Command to start DHCP server	If this field is not set to Run server executable , the module will use this command instead when the Start Server button is clicked. On some Linux distributions, it is set to a bootup script that is normally used to start the server at boot time. If you have compiled and installed the DHCP server yourself, however, you should set it back to Run server executable .
Command to apply configuration	When this field is set to Kill and restart , Webmin will kill the DHCP server process and rerun it when the Apply Changes button is clicked. If you have compiled the server yourself, then you should select this option. Only on Linux distributions that provide a bootup script that can restart the server will it be set to something different by default, such as <code>/etc/init.d/dhcpd restart</code> .
Path to DHCP server PID file	This field must be set to the full path to the DHCP server's PID file, such as <code>/var/run/dhcpd.pid</code> . The module uses the PID file to determine if the server is running and restarts it when Apply Changes is clicked.
DHCP server lease file	This configuration field should be set to the full path of the file in which the DHCP server stores leases. You should always make sure that the specified path is the same as the default compiled in the <code>dhcpd</code> server executable, otherwise the page listing leases will be empty or incomplete.
Interfaces file type	This field determines which file the module updates when you change the network interfaces on which the server listens. Various Linux distributions have their own special files, which are read by the bootup script specified in the Command to start DHCP server field. This option should only be changed if you have compiled and installed the server yourself. If so, set it to Webmin , which tells the module to just keep the list of interfaces in its own configuration file.

32.12 Summary

This chapter has introduced the DHCP protocol and the common ISC DHCP server for UNIX systems. It has explained how you can use Webmin to configure the server to automatically assign dynamic or fixed IP addresses to clients on your network and how to set options that apply to single hosts, groups of hosts, or entire subnets. Even though not all of the module's features are covered, after reading the chapter you should know enough to set up a DHCP server for a typical network.

Downloading Email with Fetchmail

This chapter explains how to configure the Fetchmail program to download email from another server and deliver it to addresses on your system.

33.1 Introduction to Fetchmail

Fetchmail is a relatively simple program that downloads email from another server using the POP3 or IMAP protocol and delivers it to a mailbox on your system. It is most useful if you want to run your own mail server, but cannot have mail delivered directly, for some reason. The solution is to have Fetchmail download email periodically using a protocol like POP3 and then connect to the SMTP server on your system to have it delivered as if it were sent directly.

If your system has a dial-up connection to the Internet that is only occasionally active, it is not usually possible to have mail delivered directly. The same applies if you do not have a fixed IP address. In situations like this, it is still possible to run your own email domain and server by having mail for your domain sent to a mailbox at your ISP and using Fetchmail to periodically transfer it to your system.

Even if you do not have your own Internet domain, Fetchmail can still be used to download email from an email account in your ISP's domain. Many mail clients like `pine`, `elm`, and `Usermin` read the UNIX mail file in `/var/mail` directly, instead of downloading messages via the POP3 or IMAP protocol. To use one of these programs, email must be downloaded to your system and delivered to a local user.

Fetchmail can download email from multiple mailboxes on different servers and deliver it to different addresses on your system. If email to all addresses in a domain has been combined into a single mailbox, Fetchmail can usually separate it for delivery to the correct users on your system. This is possibly Fetchmail's most useful feature, but unfortunately it is not 100 percent reliable.

The Fetchmail program can retrieve mail using the POP2, POP3, and IMAP protocols, one of which will be supported by almost all mail servers. It can also use the ETRN mode of the SMTP protocol to force a mail server to deliver all queued messages that are awaiting delivery to your system. Unfortunately, it does not support the retrieval of mail from proprietary email systems, like Exchange or Lotus Notes, or from web-based email services, like Hotmail, unless they support one of the standard protocols as well.

To perform periodic checks, Fetchmail is usually run as a background daemon process that connects to all mail servers at regular intervals. It can also be run from a Cron job at times and dates of your choosing, or even started manually from the command line or some other script.

Fetchmail is often run by individual users rather than the system administrator, each with their own separate `.fetchmailrc` configuration file in their home directory. Because it does not require `root` privileges to run, each user can safely configure Fetchmail on a multi-user UNIX system to download mail from his own remote mailboxes. This means that each user may have his own separate Fetchmail daemon process running that uses his own configuration.

A single configuration file can also be used and Fetchmail can be run as `root` to download email for all users on your system. This option makes more sense if you are the only user of your Linux box, or if you are downloading email for an entire domain to be redistributed to local users. Typically, `/etc/fetchmailrc` is used as the global configuration file.

In fact, it is possible for Fetchmail to be run on both individual users' configuration files and a global file at once. The Webmin module for configuring it, however, expects you to use one mode or the other.

33.2 The Fetchmail Mail Retrieval Module

Webmin's module for managing Fetchmail can be found under the **Servers** category. When you click on its icon for the first time, the main page will display the Fetchmail configurations of all users on your system. For each user who has a `.fetchmailrc` in his home directory, the user's name and all servers from his file are displayed along with the protocol used to connect to each and the users as whom to login. Figure 33.1 shows an example.

If Webmin cannot find the `fetchmail` program, then the main page will display an error message instead. This may be because it is not installed or because the module is looking in the wrong place. Most Linux distributions come with a package for Fetchmail—check the CD or website and use the Software Packages module (covered in Chapter 12) to install it.

If you want to manage just a single Fetchmail configuration file on your system, now is the time to switch the module to that mode. Unless you want to manage the configurations of all the users on your system, this is the best choice. It allows you to set up a daemon process to periodically check for and download email to local mailboxes, which is what most administrators use Fetchmail for.

To change the module to use a single file, follow these steps:

1. Click on the **Module Config** link in the top left corner of the main page.
2. In the **Fetchmail config file to edit** field, select the second radio button and enter the configuration file path into the field next to it. If you already have a Fetchmail configuration file, then naturally you should enter its path; otherwise, `/etc/fetchmailrc` will do fine.

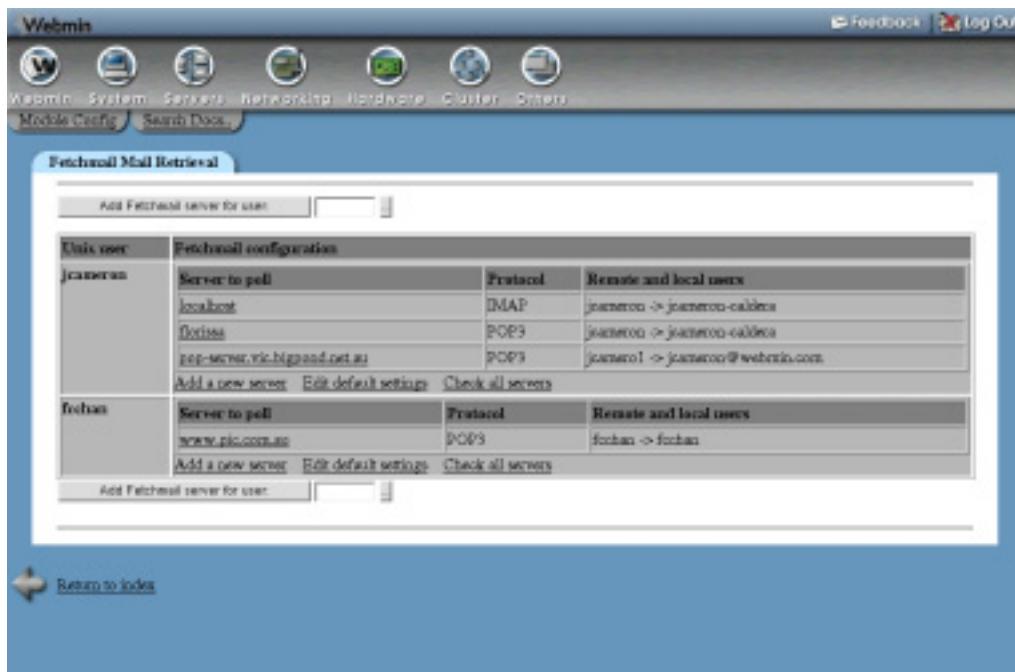


Figure 33.1 The Fetchmail module main page.

3. Click the **Save** button at the bottom of the form to update the module configuration and return to the main page.

When in single configuration file mode, only servers from that file will be displayed on the main page under the file's name. Below them is a form for starting the Fetchmail daemon to regularly check the listed servers and accounts, as will be explained in more detail later in the chapter.

Because the module does not support the starting of the Fetchmail daemon for individual users, if you are using it to manage multiple individual configuration files you will need to create a Cron job or start a daemon for each users' configuration. The easiest method is to use the Scheduled Cron Jobs module (covered in Chapter 10) to create a job for each user that runs the `fetchmail` command on a schedule of your choice. Once every 30 minutes is usually good enough, depending on how much email you get. By default, the `fetchmail` program will use the `~/.fetchmailrc` file in the home directory of the user that runs it.

Another package that can be used by users to manage their own Fetchmail configurations and even start their own daemons is Usermin, which is closely related to Webmin. See Chapter 47 for more details.

Because Fetchmail is available for UNIX (and works mostly the same on all varieties), this module behaves the same as well. The only difference is that the **Check condition** field for turning off checking if a particular network interface is down will not work on operating systems other than Linux and FreeBSD, at least with the current version of Fetchmail. Even though the field always appears, it should not be used on other versions of UNIX.

33.3 Adding a New Mail Server to Check

Before Fetchmail will download email from a mail server for you, an entry for it must be added to its configuration. To do this, follow these steps:

1. On the module's main page, click on the **Add a new server** link below the table of existing servers. If you are managing multiple users' configurations, you must use the link in the section for the user to whose list you want to add the server.
The **Add Fetchmail server for user** button can also be used to add a server to the user entered into the adjacent field. This method must be used if the user does not have any servers defined yet.
2. No matter what link or button you use, the form shown in Figure 33.2 will be displayed for entering the new server's details.
3. In the **Server name** field, enter a unique name for this mail server entry. Typically, this will be its actual hostname, such as *mail.yourisp.com*.
4. If you want this server to be checked on schedule, make sure the **Polling enabled?** field is set to **Yes**. Otherwise, it will only be checked if manually run from Webmin or at the command link.
5. The **Mail server to contact** field is useful if you need to connect to more than one port or protocol on the same host. Because the **Server name** must be unique, you can only create two entries for the same actual mail server by entering different values for the server name (such as *mail.yourisp.com-1* and *mail.yourisp.com-2*) and entering the actual hostname for the server into this field.

This situation is fairly rare, however, so you can usually just leave this field set to **Same as server name**.

6. From the **Protocol** menu, select the mail retrieval protocol to use for this server. The most common are POP3 and IMAP. Your ISP or mail server administrator will be able to tell you which one to use.
7. If the mail server is using a nonstandard port for the chosen protocol, then the **Default** option will not work for the **Server port** field. Instead, you must enter the correct port number, such as *1110*.
8. The **Check condition** field can be used to prevent periodic checking of this server if a network interface is down. This is useful if you have a dial-up connection to the Internet that is only active occasionally and want to avoid useless attempts to connect to the mail server when it is not active.

If **Always check** is chosen, Fetchmail will always try to connect. If, however, **Only if interface is up** is selected, no connection will be made if the network interface entered into the field next to it is down. Your primary PPP interface for dialup is normally named *ppp0*. See the Network Configuration module (covered in Chapter 16) for a list of active interfaces.

For checking to be performed, you must enter an interface name as well as a network and netmask to specify a range of valid local addresses for the interface. This can be useful if you dial up to several different ISPs, but only want Fetchmail to check for mail when connected to a particular one. Most ISPs assign addresses within a certain class C or B network to all customers, such as *203.51.0.0/255.255.0.0*.

To allow Fetchmail to check as long as the interface is up, no matter what IP address it has, just enter *0.0.0.0* into both the network and netmask fields. This covers all possible addresses.

9. In the **Mail server user details** section, enter the login name with which to connect to the mail server into the **Remote user** field.
10. Enter the correct password for the user into the **Remote password** field.
11. The **Local user(s)** field is for entering the email address to which to send retrieved messages. This is typically a local username like *jcameron*, but it can also be an address on another server like *jcameron@example.com*.
It is also possible to enter several usernames, in which case Fetchmail will attempt to work out which of those names each downloaded message is for. This is useful if you have email for several addresses forwarded to the same mailbox on your ISPs mail server and want to split up the retrieved messages for delivery to the correct local mailboxes. If Fetchmail encounters a message whose recipient is not in the list, it will be bounced back to the sender.
The final alternative is to just enter * in the **Local user(s)** field, which tells Fetchmail to deliver each message to the local user on your system whose name is the same as the username part of the message's destination address.
12. If you want Fetchmail to delete messages from the mail server after downloading them, set the **Leave messages on server?** field to **No**. Unless another mail client is being used to access the mailbox, this is the best option as it prevents an additional copy of every message being stored on your ISP's server—which may have a limit on mail file sizes.
Selecting **Yes** causes Fetchmail to keep track of received messages and only download those in the mailbox that are new. In effect, it is synchronizing the remote mailbox to a local one, except that messages deleted on the server will not be deleted locally.
13. If you are keeping messages on the remote server, the **Always fetch all messages?** field should be set to **No**. Otherwise, set it to **Yes** to guarantee that all messages in the mailbox are downloaded.
14. The **Command to run before connecting** field can be used to enter a shell command that will be executed by Fetchmail just before connecting to the mail server. One of the most common uses of this feature is running a command to set up an SSH tunnel to allow access to a server that you cannot connect to directly. This can be quite complex though, and so is not covered here. Typically, this field can be left empty.
15. Similarly, the **Command to run after disconnecting** field is for entering a shell command to be executed after Fetchmail logs off from the remote mail server. It is often used for killing the SSH process started by the **before** command.
16. Finally, click the **Create** button to save the new server. It will be used hereafter whenever Fetchmail is run when it makes a periodic check.

Once you have created a new server entry, it will be listed on the module's main page. To edit it, just click on the server name in the **Server to poll** column, which will bring up the editing form in your browser. Change any of the fields and click **Save** to update the Fetchmail configuration file.

Servers can be deleted by hitting the **Delete** button on the editing form. It is usually better, however, to change the **Polling enabled?** field to **No**, which effectively disables the server.

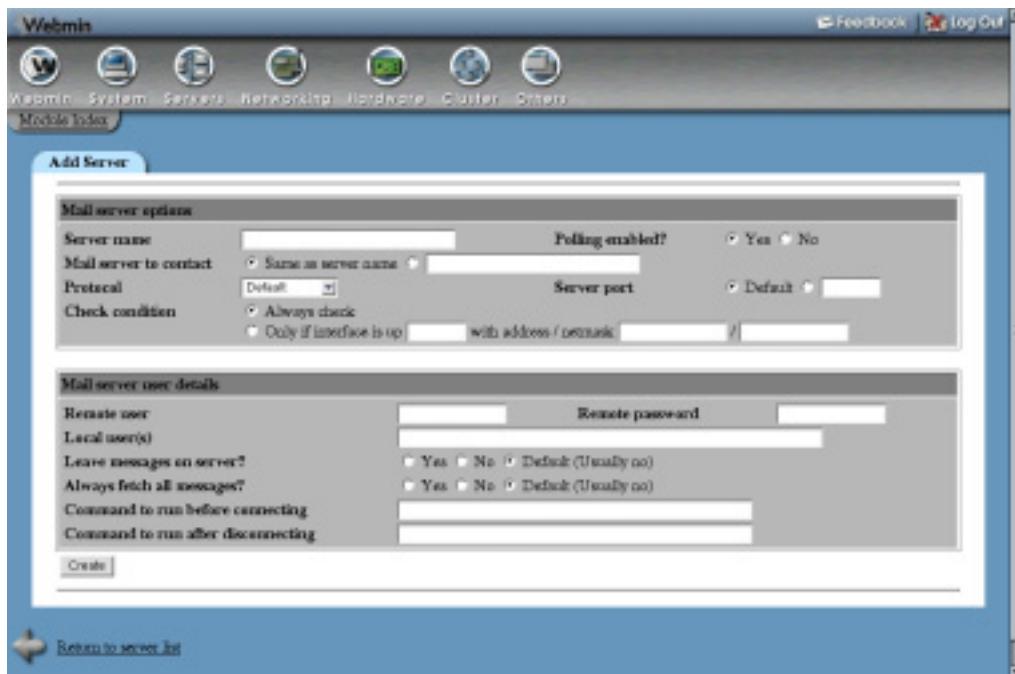


Figure 33.2 The server creation form.

Fetchmail will not connect to it unless you explicitly tell it to check that server, as explained in Section 33.4 “Downloading Email”.

It is possible to have Fetchmail check more than one mailbox on the same server and deliver mail from additional mailboxes to different users. This could be done by creating multiple configuration entries for the same server, but there is a simpler and better method, done using the following steps:

1. On the module’s main page, click on the name of the server to which you want to add an additional mailbox to check.
2. Click on the **Add another user** button. The editing form will be redisplayed, but with an additional empty **Mail server user details** section at the bottom.
3. Fill in the empty **Remote user**, **Remote password**, **Local user(s)**, and other fields in the new section, as explained in the steps in Section 33.2 “The Fetchmail Mail Retrieval Module”.
4. Hit the **Save** button. You will be returned to the module’s main page and the new remote and local usernames will be displayed next to the server.

Even though its ability to extract mail for multiple users from a single mailbox is one of Fetchmail’s most useful features, it is not 100 percent reliable. There is no way that the program can accurately determine what address an email was sent to in all cases. Normally, the **To:** or **Cc:** header will contain the destination address, but for messages received from mailing lists this is not the case—instead, the **To:** header contains the list’s address. There are other mail headers

that Fetchmail attempts to check to find the real destination address of a message, but they are not always available.

When an email message is delivered directly to a server via the SMTP protocol, the source system informs the destination server of the message's real destination address. Unfortunately, the address does not have to be in the actual message at all—instead, it is specified as part of the SMTP conversation between the servers. When the email is delivered to a mailbox, this information is lost and cannot be accurately recovered.

Only when Fetchmail is downloading email from a mailbox and delivering it to a single recipient is it guaranteed to do the right thing. In this case, it never has to check the destination address of each message because they are all being sent to a single local mailbox.

33.4 Downloading Email

Once you have created at least one server entry, you can use this module to have Fetchmail connect to the server and download messages. The module can be used to retrieve email from all servers in a configuration file, or just a single server.

To check them all, follow these steps:

1. On the module's main page, click on **Check all servers** below the table of servers. If you are managing the configurations of multiple users, this link will appear under the table for each user.
2. A page showing the output of the `fetchmail` command will be displayed so you can see the POP3 or IMAP protocol exchange between your system and the remote mail servers as Fetchmail downloads messages. If an error occurs (such as a failure to connect or an incorrect password), you will be able to see it in the output.
3. Downloaded messages will be delivered to the local addresses specified in the server configuration entry. By default, mail will be sent by making an SMTP connection to the mail server on your system. The actual SMTP protocol commands used to deliver the mail will be shown on the generated page so you can see if any errors occur.

Delivery can fail if there is no mail server running on your system or if it does not access email for the specified local address. If this happens, Fetchmail will attempt to send a bounce message back to the sender.

It is also possible to check for mail on a single server, even one that has the **Polling enabled?** field set to **No**. To do this, use the following process:

1. On the module's man page, click on the name of the server to bring up its editing form.
2. Click on the **Check this server** button at the bottom of the page.
3. A page showing output from Fetchmail as it downloads and delivers messages will be shown, as described earlier.

33.5 Running the Fetchmail Daemon

If you are using the module to manage a single Fetchmail configuration file, it is possible to start a background process to regularly check the servers and mailboxes in that file.

To do this, follow these steps:

1. At the bottom of the main page is a button labeled **Start Fetchmail Daemon**. In the description next to it is a text field for entering the number of seconds that the daemon should wait between checks. A short period (such as 60 seconds) means that you will receive email sooner, but at the cost of using more bandwidth and CPU time, due to frequent checking.
2. After entering a checking period, hit the **Start Fetchmail Daemon** button to start the background `fetchmail` process. The page will be redisplayed, but with the button now labeled **Stop Fetchmail Daemon**.

As the name suggests, you can click on the new **Stop** button at any time to kill the running daemon process. When the module detects that it is no longer running, the **Start** button will appear again.

If your system is rebooted, the Fetchmail daemon will, of course, be stopped. For it to be started again automatically at boot time you will need to create a bootup action as explained in Chapter 9. This action must run the `fetchmail -d interval -f configfile` command, with *interval* replaced by the checking period and *configfile* replaced with the full path to the configuration file.

33.6 Editing Global Settings

The Fetchmail module can also be used to edit options that apply to all servers in a configuration file. This can be useful for stopping any server from being contacted if a network interface is down, or defining a default protocol. To edit these global options, follow these steps:

1. On the module's main page, click on the **Edit default settings** link below the table of servers. If the module is being used to manage the Fetchmail configurations of multiple users, this link will appear under the table for each user who has any servers defined. Either way, your browser will display a page for editing global options.
2. To set a default protocol for all servers, select one from the **Protocol** menu. The most common are **POP3** and **IMAP**, with the former being used if the **Default** option is selected. When the **Protocol** field on the server editing or creation form is set to **Default**, then the protocol selected here will be used.
3. To define a default port for Fetchmail to connect to, fill in the **Server port** field. It is usually best to leave this set to **Default**, though, in which case the program will use the appropriate port for the protocol selected for each server. Only when the **Server port** field on the server form is set to **Default** will the value entered here be used; otherwise, it will be overridden with whatever you enter for that server.
4. The **Check condition** field can be used to prevent Fetchmail from connecting to any servers if a particular interface is down or does not have the correct IP address. The instructions in Section 33.3 “Adding a New Mail Server to Check” explain how this field works and what to enter.
Setting the check condition globally makes more sense than setting it repeatedly for individual servers, as the servers that Fetchmail is checking are all likely to be accessible over the same network connection.
5. Click the **Save** button to make the new global settings active.

When you are using the module to manage multiple users' Fetchmail configurations, there is no way to define options that apply to all users—just the global settings for a single user at a time.

33.7 Module Access Control

As Chapter 52 explains, it is possible to restrict what a Webmin user can do with a module to which he has been granted access. For the Fetchmail module, you can limit the UNIX users for whom he can edit Fetchmail configurations. Once a user has been created, you can further restrict access by following these steps:

1. In the Webmin Users module, click on **Fetchmail Mail Retrieval** next to the name of the user. This will bring up the module access control form.
2. Change the **Can edit module configuration field?** to **No** to stop the Webmin user from switching the module to single-file mode or changing the path to the Fetchmail program.
3. The **Can edit fetchmail config for** field determines for which UNIX users this Webmin user can edit Fetchmail servers. The available options and their meanings are:
 - All users** The configuration of any user can be edited. This is the default.
 - Current Webmin user** Only the UNIX user whose username is the same as the Webmin user can be edited. This option can be useful for allowing people to edit their own Fetchmail settings, although the Usermin program is a better alternative.
 - Only users** Only the configurations of users entered into the text field next to this option can be edited.
 - All except users** The Fetchmail settings for all users except those entered into the adjacent text field can be edited.
4. Click the **Save** button to make the new module restrictions active.

This kind of access control is only useful if the module has been configured to allow the editing of individual `.fetchmailrc` files. In single configuration file mode, no restrictions apply.

33.8 Configuring the Fetchmail Mail Retrieval Module

Like other modules, this one has several options that can be edited by clicking on the **Module Config** link on the main page. The options are shown in Table 33.1.

33.9 Summary

After completing this chapter you should understand what Fetchmail does, and how to set it up on your system to download email from other servers to local users. You should be familiar with the various options that can be set for each remote server and mailbox and understand the difference between the module's single-file and all-users modes.

Table 33.1 Module Configuration Options

Fetchmail config file to edit	As explained earlier in the chapter, the module can manage either a single Fetchmail configuration file or files belonging to all UNIX users. When this field is set to All users' .fetchmailrc files , the contents of every user's <code>.fetchmailrc</code> file will be displayed on the main page for editing. Selecting the other option tells the module to edit the single configuration file whose path is entered into the text field next to it.
User to run the fetchmail daemon as	Normally, the Start Fetchmail Daemon button runs the background checking process as <code>root</code> . To have it run as a different UNIX user, enter some other username into this field. Often there is no need for the daemon to be run as <code>root</code> because it requires no special privileges. Running as an unprivileged user like <code>nobody</code> reduces the chance that a security hole in Fetchmail could lead to the takeover of your system by an attacker.
Mail delivery command	When this field is set to Use SMTP (as it is by default), Fetchmail will deliver downloaded email by making an SMTP connection to the mail server on the same system on which it is running. The alternative is to enter a command that can accept an email message for delivery as input, such as <code>/usr/sbin/sendmail -oem -f %F %T</code> . This option can be useful if you do not have a mail server running all the time to accept SMTP connections.
Path to the fetchmail program	This field must contain either the full path to the Fetchmail executable (such as <code>/usr/local/bin/fetchmail</code>), or just <code>fetchmail</code> if it is in Webmin's program search path. You should only need to change it if you have installed the program in some directory that is not in the search path.
Path to the fetchmail daemon PID file	This field must contain the location of the PID file that Fetchmail creates when started in daemon mode. The default is always <code>/var/run/fetchmail.pid</code> , but this may be incorrect if you have compiled it from the source or installed a package that uses a different PID file path.

Managing Majordomo Mailing Lists

This chapter documents the process of setting up mailing lists on your system using Webmin and the freely available Majordomo list management program.

34.1 Introduction to Mailing Lists and Majordomo

Mailing lists provide a way of facilitating group discussions via email or broadcasting messages to multiple email addresses. At its heart, a mailing list is simply an email address that forwards all mail sent to it to a list of member addresses. Typically, messages are modified so that replies go back to the list address instead of to the original sender, making it easy for members to participate in a group discussion via email.

Majordomo is the most popular mailing list management program for Linux and UNIX systems. As well as forwarding mail from the list address to members, it handles subscription and unsubscription, moderation, and message filtering. It can append headers and footers to messages, send out periodic digests containing list mails from the past few days, create archives of list messages, and much more.

Because it is written in Perl, Majordomo will run on almost all versions of UNIX and behaves the same on all supported operating systems. It only requires that you have a mail server installed that can forward messages to a program or to a file of addresses, which Sendmail and Postfix can do. See Chapter 37 for more information on setting up Sendmail on your system.

Users of a mailing list typically subscribe by sending email to the special `Majordomo` address on your system, such as `Majordomo@example.com`. In fact, there are several types of commands that can be sent to this address by simply including them in a message body. The most commonly used ones and their parameters are:

subscribe list address Adds the *address* to the specified *list*. The address can actually be omitted, in which case the **From:** address of the command email will be

used instead. If this list requires confirmation for new subscriptions, the subscriber will receive a confirmation message that he must answer before being added.

unsubscribe list address Removes the *address* from the specified *list*. If the address is not supplied, the sender of the command email will be used instead.

lists Sends back a list of all mailing lists on the server and their descriptions. Some lists may be hidden from certain addresses.

who list Sends back the list of subscribers for the specified *list*. Access to this command is often restricted to subscribers.

info list Sends back the information text for the specified *list*, which is usually a description of what the list is for and who can join it.

intro list Sends back the welcome message for the specified *list*, which subscribers get when they join.

index list Sends back a list of files available for download associated with the specified *list*, which can be retrieved by the **get** command.

get list file Returns the contents of a downloadable file associated with the specified *list*.

For example, to add yourself to a mailing list you could just send email to *Majordomo@example.com* containing the line `subscribe example-list yourname@example.com`. A single message can contain several commands—one on each line. After Majordomo receives a message, it will process the commands and send back a response email containing information about the success or failure of each command and any information that they produce.

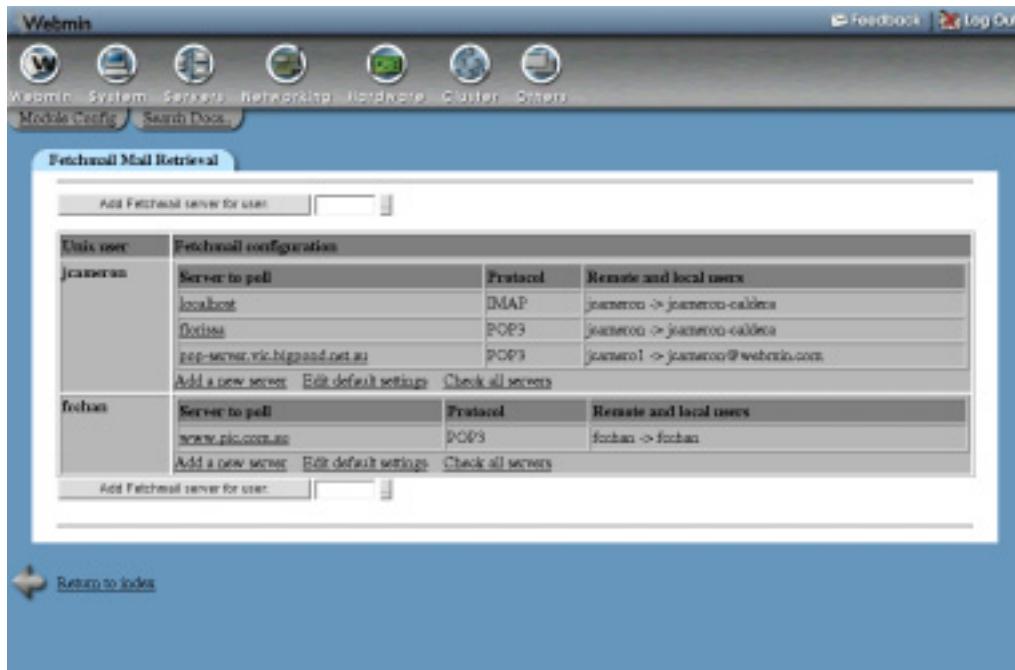
Of course, before any of these commands will work on your system you must set up Majordomo and create at least one mailing list. Section 34.2 “The Majordomo List Manager Module” explains how.

34.2 The Majordomo List Manager Module

This module allows you to set up and manage multiple Majordomo mailing lists on your system. It is designed to interface with the Sendmail Configuration module to set up the mail aliases that Majordomo needs to operate, but can be used with any mail server that uses an `/etc/aliases` file. If you are not using Sendmail, see Section 34.3 “Using Other Mail Servers” for details on how to configure the module to work with your server. If Webmin detects that Sendmail is not installed, an error message will be displayed on the main page notifying you that its configuration file cannot be found.

Like other email-related modules, this one can be found under the **Servers** category in Webmin. When you enter it, the main page displays a table of icons—one for each mailing list on your system. Figure 34.1 shows an example.

For Majordomo to operate, it requires that you define the `Majordomo` and `Majordomo-owner` mail aliases and forward messages to the Majordomo program and the list administrator, respectively. If the module detects that these aliases do not exist (usually because you have never set up Majordomo or used the module before), a field labeled **Owner email address** will be displayed on the main page.



Neither does it deal well with multiple virtual email domains. Some systems have one Majordomo configuration file for each domain, and possibly multiple alias files as well. The only way to use the module in this situation is to clone it once for each domain (as explained in Chapter 51), and configure each clone to use a different `majordomo.cf` file.

34.3 Using Other Mail Servers

By default, this module assumes that you are using the Sendmail mail server. Many people, however, prefer alternatives like Postfix or Qmail due to their superior configuration file formats, reliability, and design. Fortunately, the module can work with any mail server that uses an aliases file in the same format as Sendmail's `/etc/aliases`.

All that you need to do is tell the module where the aliases file is located instead of having it read the Sendmail configuration and find it automatically. To do this, follow these steps:

1. On the module's main page, click on the **Module Config** link. This will bring up the standard module configuration form.
2. For the **Sendmail-style aliases file** field, deselect the **Get from sendmail.cf** option and enter the full path to your mail server's aliases file. This will usually be something like `/etc/aliases` or `/etc/postfix/aliases`.
3. Click the **Save** button to return to the main page. Any error message about the Webmin's inability to find the `sendmail.cf` file will have disappeared.
4. Click on the **Edit Majordomo Options** button at the bottom of the page and fill in the **Sendmail command path** field with the path to a program that works in the same way as the real `sendmail` command. All mail servers come with a program like this to preserve compatibility with programs that expect Sendmail to be installed—usually found at `/usr/sbin/sendmail` or `/usr/lib/sendmail`. Majordomo uses this command to send outgoing email to list members.
5. Click **Save** to return to the main page. The module and Majordomo will now function properly with your mail server.

Not all mail servers have a standard aliases file, so this module cannot be used with them. Qmail requires a patch before it will read an aliases file, as normally it uses `.qmail` files in the `/var/qmail/aliases` directory to define aliases.

34.4 Creating a Mailing List

Once the Majordomo module has been set up correctly, you can use it to create a new mailing list. Every list must have a name that cannot be used by any other list, UNIX user, or email alias. Lists are typically named like *example-list* or *engineering-l*, but really any short name consisting of numbers, letters, and dashes is allowed. The name forms the part of the list's email address before the @, so the resulting address will be something like *example-list@example.com*.

To create a list, follow these steps:

1. On the module's main page, click on the **Add a new mailing list** link above or below the table of existing list icons. This will bring you to the list creation form, shown in Figure 34.2.
2. Enter the chosen name into **List name** field.

3. Fill in the **List maintainer's address** field with the email address of the person who will be responsible for this list. They will receive notifications of subscriptions and unsubscriptions, and bounce messages if delivery to a list member fails.
4. The password entered in the **Maintenance password** field can be used by the maintainer to change the list's configuration by email. Make sure that this password does not fall into the hands of anyone else, as it grants full control over the list to anyone who knows it.
You will generally not need to configure the list via email anyway, as this Webmin module provides a much nicer interface than manually editing the configuration file.
5. In the **Description** field, enter a short description of this list that will be displayed next to its name when a user sends the `lists` command to Majordomo.
6. The text in the **Introductory message** field will be sent to all new list members when they subscribe. You should enter a description of the list, posting guidelines, and any other information that new subscribers might need to know.
7. To have a footer appended to each email sent to the list, fill in the **Forwarded mail footer** field. Often the footer contains the list name and information about how to unsubscribe.
8. If you want this list to be moderated, change the **Moderated list?** field to **Yes**. A moderated list is one that requires all messages sent to it be approved before they are sent to list members.
9. If the moderator is the same as the list administrator, leave the **Moderator's address** field set to **Same as maintainer**; otherwise, enter an email address into the adjacent text field. This address will receive a copy of every email sent to the list for approval.
10. If you want Majordomo to store copies of messages sent to this list in archive files, select one of the options other than **No** from the **Archive mailing list?** menu. An archive is a collection of files under the list's directory that contains email to the list for a year, month, or day.
11. Finally, click the **Create** button to have the module create the list and all the Sendmail aliases that it needs to operate. You will be returned to the main page, which should now contain an icon for your new list.

People can subscribe to the new list and send email to it as soon as it is created. However, you will probably want to customize its settings some more before announcing its existence. The next few sections explain how.

34.5 Managing List Members

Even though people can subscribe and unsubscribe themselves to and from a mailing list, you can also use this Webmin module to manage the membership list. To directly edit the member list, follow these steps:

1. On the module's main page, click on the icon for the mailing list. This will bring you to a page which contains icons for various categories of list options.
2. Click on the **List Members** icon to go to the membership management page shown in Figure 34.3.

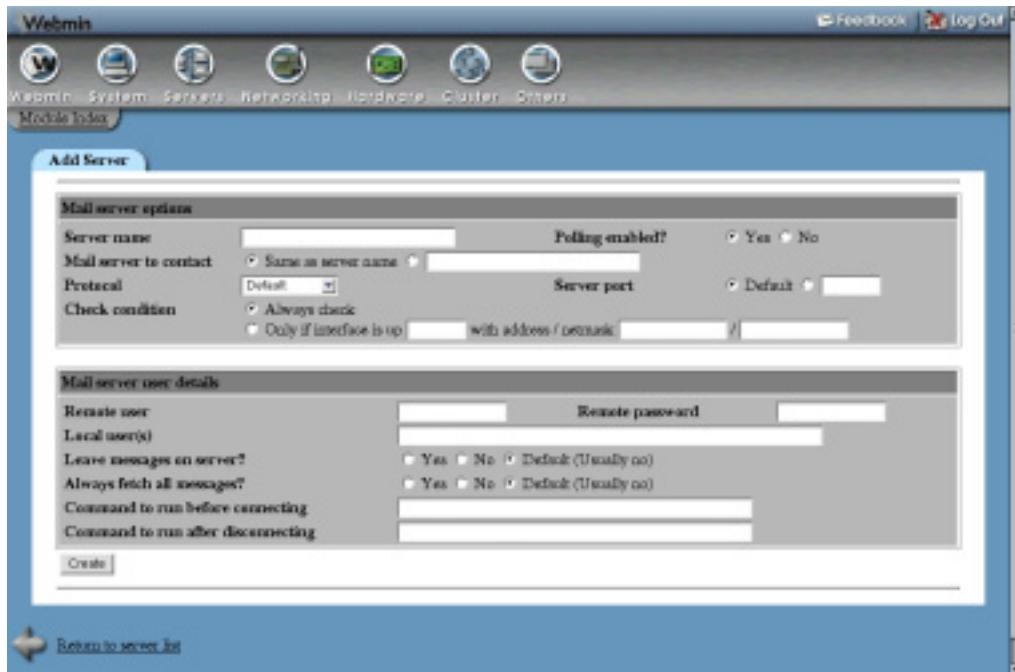


Figure 34.2 Creating a new mailing list.

3. On the page that appears, add to, remove from, or edit the list of members in the text box. Any new subscribers that you add by editing this list will not receive a welcome email or confirmation message.
4. Hit the **Save** button to make the new member list active.

The same page can also be used to subscribe people to the list. Just enter an address into the **Subscribe address to list** field and click the **Add** button. When a member is added this way, he will receive the usual welcome message as though he subscribed normally by sending email to *Majordomo@yourdomain.com*. Internally, the module does the subscription in the proper way by faking an email to Majordomo instead of just directly updating the file containing list members.

An address can be deleted from the list by entering it into the **Remove address from list** field and hitting **Remove**. As far as the subscriber is concerned, there is no difference between this method and just deleting his address from the member text box. Internally, however, the unsubscribe is done by a faked email to Majordomo that tells it to delete the address from the member file, rather than Webmin updating that file itself.

34.6 Editing List Information, Headers, and Footers

After a list has been created, you can still edit the description, welcome message, and footer that were chosen on the creation form.

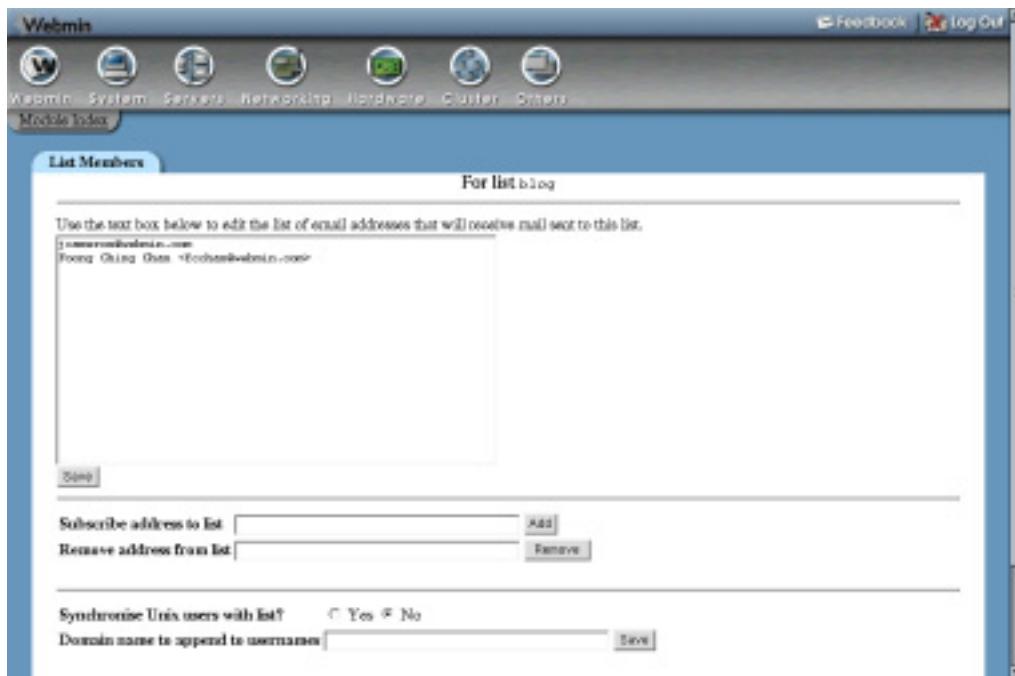


Figure 34.3 Managing mailing list members.

To do so, follow these steps:

1. On the module's main page, click on the icon for the list.
2. Click on the **Messages and Description** icon to bring up a page for editing the description, welcome, and information message.
3. To change the description shown in response to the `lists` command sent to the Majordomo address by users, edit the **List description** field.
4. The **Informative message** field contains text that will be sent back in response to the `info listname` command, and possibly also to new subscribers. Change it to whatever you want.
5. The welcome message is usually the same as the informative message. To change this, select **As entered below** for the **Introductory message** field and fill in the text box with a message to be sent to new subscribers.
6. Click **Save** to return to the list icons page, and then hit the **Headers and Footers** icon.
7. To have a header added to the top of all list messages, fill in the **Resent email header** field. Most lists don't use a header—a footer is less annoying to readers.
8. To edit the footer appended to the bottom of list messages, change the **Resent email footer** field.
9. If you want every list message to have additional mail headers added to it, fill in the **Extra SMTP headers for resent mail** field. These must be in the standard email header format, such as:

```
Subject: blah
X-Mailing-List: example-list@example.com
```

New headers cannot override those in forwarded messages—they can only add to them. If you want to change the subject or reply-to address, see Section 34.8 “Editing Forwarded Email Options”.

10. Click the **Save** button at the bottom of the form. All the new list information, header, and footer settings will be immediately active.

34.7 Editing Subscription Options

By default, Majordomo allows anyone to subscribe themselves to a newly created list and sends a confirmation message to new members to make sure that they really do want to subscribe. These features prevent people from being subscribed against their wishes by someone else, but can be annoying on a small company mailing list with trusted users.

To change these and other subscription options, follow these options:

1. Click on the list’s icon on the module’s main page, and then on **Subscription and Moderation**.
2. The **Subscribe policy** field controls who can be subscribed to the list. The options are:

Anyone can subscribe themselves When selected, an address can only be added to the list (by an email to Majordomo) if the address that it comes from is the same. This security feature can be trivially defeated by faking the source address of a message, however, and can be irritating to people who want to use a special address for subscriptions.

Anyone can subscribe anyone When this option is selected, any address added to the list will be accepted.

Maintainer approval required This option tells Majordomo to forward all subscription requests to the list maintainer for approval. It should be used if you want to control who can join the list, instead of allowing just anyone to sign up.

3. The **Unsubscribe policy** field controls who can be removed from the list. Its options are the same as the **Subscribe policy** field.
4. To have Majordomo send a message to new addresses confirming their subscriptions, set the **Require subscribe confirmation?** field to **Yes**. This confirmation message must be replied to before the address is actually added to the list, indicating that the person really does want to subscribe. I would recommend always turning this feature on unless you are running a very small internal mailing list.
5. If you want the welcome message (entered when the list was created) to be sent to all new members, set the **Sent introductory message?** field to **Yes**.
6. The **Add only raw addresses to list?** field determines whether Majordomo stores complete addresses (like *Jamie Cameron <jcameron@webmin.com>*) or just raw addresses (like *jcameron@webmin.com*) in the membership list. Unless you want to be able to see the real names associated with subscriber addresses, this field should be set to **Yes**.
7. If you want the list maintainer to receive a copy of all subscription and unsubscription requests, set the **Forward subscribe/unsubscribe requests to maintainer?** field to **Yes**. This does not, however, mean that the maintainer must approve the requests.

8. The password specified in the **Maintenance password** field must be included in all messages to approve subscriptions or change the list's configuration by email. It should be given only to the maintainer, as anyone who knows it can edit any of the list's settings.
9. To require that every message to the list be approved by a moderator, change the **Moderated list?** field to **Yes**.
10. The **Moderator's address** field controls where requests to approve messages to the list are sent. You can either set it to **Maintainer** (in which case the address in the **Approval email address** field will be used) or enter something different.
11. The **Moderation password** field is for entering a password that must be included in all messages that approve postings to the list. It should be known only by the moderator (if there is one) and yourself—otherwise, subscribers could approve their own postings!
12. To change the address of the list owner (to which email to *listname-owner* is sent), edit the **Owner's email address** field. The owner will receive bounce messages that are sent back if email to a subscriber cannot be delivered.
13. To change the maintainer's address, edit the **Approval email address** field. All notifications of and requests for subscription and unsubscription will be sent to this person.
14. Finally, click **Save**. The new settings will take effect immediately.

When confirmation is enabled for a list, anyone who sends a `subscribe` command to the Majordomo address will receive an email asking them to send back a special auth command. This additional step guarantees that the person being added to the list actually wants to join because the auth command contains a random number that Majordomo associates with the subscribing address.

When moderation or subscribe/unsubscribe approval is enabled, additional messages will be sent to the moderator's or maintainer's addresses. See Section 34.10 “Moderating and Maintaining a Mailing List” for more information on how to actually deal with these emails.

34.8 Editing Forwarded Email Options

When an email message is sent to the list, Majordomo does not just forward it to subscribers unchanged. Instead, it modifies the headers—and possibly the content—based on the list's configuration. The most common modification is the addition of a **Reply-To:** header so that member replies will go to the list rather than to the original sender. This is generally what you want in a group discussion list.

Other modifications that Majordomo can perform on list messages include adding some text to the subject line and changing the priority. To configure these changes, follow these steps:

1. On the module's main page, click on the list icon and then on **Email Options**.
2. To have **Reply-To:** header added to list email, enter the list's email address (such as *example-list@example.com*) into the **Reply-To: address in resent email** field. Selecting the **None** option will cause reader replies to go the original sender of each message by default.
3. To set the sender address of forwarded messages, fill in the **Sender: address in email** field. This should be *owner-listname*, which is forwarded to the actual owner's address. Any bounce messages that come back from subscribers will be sent to this address.

4. The **Hostname for resent email** should be set to the mail domain of your system, such as *example.com*. Leaving the **Default** option selected tells Majordomo to work out the domain automatically, which it may not do correctly.
5. To have specific text prepended to the subject line of every list message, enter it into the **Subject: prefix for resent email** field. Typically the list name in brackets will be added, such as [*example-list*].
6. To change the priority of list messages, select one from the **Resent email priority** menu. Traditionally, mailing lists use the **Bulk** priority, and some programs depend on this to identify list email.
7. Every email message has several **Received** headers, each of which is added by a mail server through which the message passes. To have Majordomo strip them from forwarded messages, change the **Remove Received: headers from resent email** field to **Yes**. You may want to do this to hide the IP address and other details of people who post to the list.
8. To limit the size of messages that can be sent to the list, fill in the **Maximum allowable message size** field. On a list with many members, it is wise to stop people from sending large messages due to the load that their forwarding will place on your system and network. A 40,000 byte limit is quite enough for the kinds of text emails that most people send to a mailing list.
9. When you are done editing email forwarding options, click **Save**. As long as there are no errors in the form, you will be returned to the list option icons page.

Even though the addition of a **Reply-To:** header is convenient for most users, some people consider it to be a bad idea. Because almost all email clients have a **Reply to all:** button for sending email to the original sender and all recipients, there is no need for the normal **Reply:** button to send mail to the list. In fact, having a **Reply-To:** header makes it difficult for readers to reply just to the original sender instead of to the entire list.

34.9 Editing List Access Control

Majordomo can be configured to restrict who can get information about a list, who can post to it, and the content of messages that they send. Often you will want to restrict posting to list members or to a limited group of people, instead of allowing anyone in the world who knows the list address to send email to its members. These poster restrictions are not totally effective, however, as it is easy for anyone to forge their **From:** address.

To control access to a list, follow these steps:

1. On the module's main page, click on the list's icon to bring up the page of option category icons.
2. Click on **Access Control** to display a form for editing information and posting restrictions.
3. The first six fields on the page can be used to restrict access to the `get`, `index`, `info`, `intro`, `which`, and `who` Majordomo commands (explained in the introduction). You may want to do this to hide details of the list from some or all people. In particular, the list of members should not be available to potential attackers.

The fields—named **Access to get command** and so on—have the following options:

Anyone The command is available to anyone who can send email to the Majordomo address.

List members The command is only available to list members, identified by the **From:** address of messages to Majordomo.

Nobody Nobody can use the command for this list at all.

4. The Majordomo `lists` command usually returns information about all mailing lists on your system. The **Include this list in response to lists command for** field, however, can be used to limit the senders who will see this particular list, which can be useful if you want to hide it from casual queries. The available choices are:

Everyone The list is visible to anyone who can send email to Majordomo.

Addresses matching regexps Only senders whose addresses match one of the Perl regular expressions entered into the adjacent text box will be able to see the list.

Addresses not matching regexps Only senders whose addresses do not match one of the entered regular expressions will be able to see this list.

5. To limit posting to list members only, change the **Who can post to the list?** field to **List members**. This makes a lot of sense as it protects your list from spammers who have somehow obtained its address.

If only a few people should be allowed to post, create a file containing their email addresses and enter the path into the **Addresses in file** field. This option is useful for announcement lists on which most subscribers only receive information and cannot post.

6. To block postings containing certain words or text, use the **Taboo body regexps** field. All banned words or sentences must be entered one per line with a / at the start and end, such as `/smeg/`.
7. It is also possible to block messages whose headers match certain regular expressions by filling in the **Taboo header regexps** field. For example, you could enter `/Subject:.*money.*/` to block all postings with the word *money* in their subject line.
8. When you are done on this page, click the **Save** button. The new restrictions will take effect immediately.

34.10 Moderating and Maintaining a Mailing List

If your mailing list requires maintainer approval to subscribe or unsubscribe, you will have to deal with messages sent by Majordomo when someone wants to join or leave the list. These emails will have subject lines like `SUBSCRIBE example-list foo@bar.com` or `UNSUBSCRIBE example-list foo@bar.com`. To approve a subscription, you must send email to the Majordomo address on your system containing the line `approve password subscribe list address`.

In this command, *password* must be the maintenance password set for the list on the subscription and moderation page, *list* must be the name of the list to which to add the subscriber, and *address* must be his email address. In fact, it is possible to use this command at any time to add someone to a list, even if they have not asked to join.

Similarly, to approve the unsubscribe of a list member you must send the command `approve password subscribe list address` to the Majordomo address.

If you are the moderator for a mailing list, you will receive a message with a subject like `BOUNCE: foo@bar.com: Approval required whenever someone tries to post to the list.` In the body of the message will be the original email that was sent to the list address, which you are supposed to read to make sure it is appropriate for posting.

Unfortunately, the only way to approve a message is to save the entire email to a file and run the `approve` command on it. For example, if the message was saved to `/tmp/email` you would run `approve /tmp/email` to send it to the list. In addition, you must create a file called `.majordomo` in your UNIX home directory that contains the names of lists, their moderator passwords, and Majordomo email addresses. The `.majordomo` file must be formatted like this:

first-listname	first-password	Majordomo@whatever.com
second-listname	second-password	Majordomo@example.com

Future versions of Webmin will hopefully simplify the approval process.

34.11 Deleting a Mailing List

If a list is no longer needed, you can easily delete it and all associated files and aliases using this module. The membership list, configuration files, and any archives will be permanently removed. To delete a list, follow these steps:

1. On the module's main page, click on the list's icon.
2. Hit the **Delete List** button below the table of option category icons. A confirmation page will be displayed, showing all the files and Sendmail aliases that Webmin will delete when removing the list. Because the aliases include any that start or end with the list name, make sure that none of your own unrelated aliases are included.
3. To go ahead, click on the **Delete List** button. Once the deletion is complete, you will be returned to the main page of the module.

34.12 Creating a Digest List

A digest mailing list is one that combines several messages from another list into a single email before sending it to subscribers. Digests are always associated with normal lists and have their own subscribers who only receive postings in digest format. Digests are never moderated, and are not posted to by subscribers—instead, posts go to the original list.

The number of messages that are combined into a single digest email can be determined by their size, or by a time period. Majordomo can be configured to send out a new digest once it reaches a certain size, or after a certain number of days from the time the first message was added. It will never send out an empty email to digest subscribers.

The process of adding a digest list is similar to that for adding a normal list, as explained in Section 34.4 “Creating a Mailing List”. To add a digest list, follow these instructions:

1. On the module's main page, click on the **Add a new digest list** link. This will take you to the list creation form, which is similar to the one shown in Figure 34.2.

2. Enter a name for the list into the **List name** field, such as *example-digest*. Traditionally, a digest list will have the same name as the original list, but with *-digest* added to the name.
3. Select the name of the original list that you want to send out in digest form from the **Make digest of list** menu. Although it is possible to have a digest of a digest, this is not very useful in practice.
4. Enter the address of the person who is responsible for this list into the **List maintainer's address** field.
5. Enter a password for the maintainer to use in the **Maintenance password** field.
6. In the **Description** field, enter a short description of this digest that will be displayed next to its name when a user sends the `lists` command to Majordomo.
7. Fill in the **Introductory message** field with a message that will be sent to all new list members when they subscribe.
8. Fill in the **Digest mail footer** field to have a footer appended to each digest sent out.
9. To have the digest sent on a regular schedule, select **Oldest message is** for the **Send digest when** field and enter the number of days between digests into the adjacent field. This assumes that messages are being regularly posted to the list. Because Majordomo only checks the age of the digest when a message is added, it is possible in practice for the interval to be greater than the specified number of days.
To have a digest email sent out when it reaches a certain size, select the **Messages total** option and enter the minimum number of lines that the digest must reach into the adjacent field. As soon as it exceeds this limit, the digest will be sent to subscribers.
10. Finally, click **Create** to have Webmin create the new digest list and all the Sendmail aliases that it needs. The alias for the original list will be updated to support the digest as well.

Once a digest list has been added, an icon for it will appear on the main page. Just as with a normal list, you can click on it and then on the category icons to edit the membership list and change settings such as the subscription policy, footers, **Reply-To:** address, and so on. Typically the **Reply-To:** address should be set so that replies to a digest email go to the original mailing list—in fact, there is no way to post directly to a digest list created by Webmin. Any mail to *example-digest@example.com* will bounce because the module does not create a mail alias with that name.

34.13 Editing Digest Options

After a digest list has been created, you can still edit options that are specific to digests such as the time period or message size that will trigger an email. To do this, follow these steps:

1. Click on the icon for your digest list on the module's main page, and then click on the **Digest Options** icon.
2. To change the subject line used for digest messages, edit the **Digest title** field. By default, this will be set to whatever was supplied for the **Description** on the creation form.
3. To have the digest sent out on a regular basis, enter a number of days for the **Oldest message age before sending** field. If the **Unlimited** option is selected, Majordomo will not take the age into account when deciding when to send.

4. To have the digest sent when it reaches a certain size, enter the minimum number of lines into the **Max digest size before sending** field. Selecting **Unlimited** tells Majordomo to ignore the size when deciding when to send the digest.
It is actually possible to enter values for both the **Oldest message age** and **Max digest size** fields. If so, it will be sent as soon as either condition is met. You can set both fields to **Unlimited** instead, but this means that the digest will never be sent!
5. Every digest message has volume and issue numbers which are included in the subject line. The current volume is shown in the **Current volume number** field—if you like, you can increase it by one every year and reset the issue number to 1 at the same time.
6. The number set in the **Current issue number** field is automatically incremented by Majordomo every time a digest is sent out. You should only change this when changing the volume number.
7. Click the **Save** button at the bottom of the page when you are done editing digest options. Any new settings will take effect immediately.

34.14 Editing Global Majordomo Options

There are a few options related to the email domain and mail program that effect all mailing lists, as well as the master *Majordomo* email address. Generally, you will not need to adjust them, but if you do, follow these steps:

1. On the module's main page, click on the **Edit Majordomo Options** button to bring up the global options form.
2. The **Mail server hostname** field must contain the default email domain name for your server, such as *example.com*. This can be overridden on a per-list basis by the **Hostname for resent email** field on the email options page, but it is simpler to set it globally here. The value entered here sets the Majordomo variable \$whereami, which can be used in the subsequent fields.
3. To change the address to which Majordomo commands must be sent, edit the **Majordomo master address** field. Changing this is rarely necessary, however, and you must update the actual Majordomo email alias as well for it to work properly.
4. To change the address of the Majordomo administrator, edit the **Majordomo owner's address** field. This is typically set to an alias that forwards mail to the real owner address, which can be changed instead of this field.
5. To have Majordomo use a different program for sending email, edit the **Sendmail command path** field. Whatever you enter must be able to accept the same parameters as the `sendmail` command; however, most replacements for this command supplied with other mail server packages will work.
6. Click the **Save** button to make the new settings active.

34.15 Module Access Control

As Chapter 52 explains, it is possible to give a Webmin user limited access to a module. People who are granted access to this module can be restricted to managing only certain mailing lists and prevented from creating new ones or editing global Majordomo options. This allows you to

give a user the rights to edit his own lists, without giving him `root` access or control over other mailing lists.

Once a user has been given access to the module, follow these steps to restrict his access:

1. In the Webmin Users module, click on **Majordomo List Manager** next to the user's name. This will take you to the module access control form.
2. Change the **Can edit module configuration?** field to **No**, so that he cannot edit the paths to Majordomo commands.
3. In the **Mailing lists this user can manage** field, choose the **Selected** option and select the lists that he should be able to configure from the box below it. Or, choose **All lists** to let him manage all mailing lists.
4. Change the **Can edit global options?** and **Can create new mailing lists?** fields to **No**.
5. If the **Can edit list membership?** field is set to **No**, the user will not be allowed to directly edit the member list or subscribe and unsubscribe people from within Webmin. From a security point of view, this doesn't really achieve much as the user will still be able to subscribe anyone he wants by sending mail to the Majordomo address with the list password.
6. Click **Save** to make the restrictions active.

34.16 Configuring the Majordomo List Manager Module

The configurable options for the Majordomo module are divided into two groups—those that control the module's operation and user interface that can be safely edited, and those related to configuration file and program paths. When you click on the **Module Config** link on the main page, the first groups of options are displayed under **Configurable options**, and the second under **System configuration**.

Fields in the latter group generally do not need to be changed unless you are not using the standard Majordomo package for your operating system, or if you are running a mail server other than Sendmail. The names and meanings of all configuration fields are shown in Table 34.1.

34.17 Summary

This chapter has explained what mailing lists are and introduced the Majordomo package that can be used to run mailing lists on your system. It has explained how to set up a list, how to edit the various options that apply to it, and how to subscribe and unsubscribe people. It has also described how to require moderation for a list and how to approve or reject messages if you are a moderator.

Table 34.1 Module Configuration Options

Use random number for list alias	When you create a list called <i>example</i> , the module will normally add a mail alias called <i>example-list</i> that sends mail directly to the file of list members. Even though this alias is not supposed to be sent to by anyone except Majordomo itself, an unscrupulous person may send mail to it to bypass the usual moderation and content checks. To prevent this, set this configuration field to Yes . From now on, new lists will use a random name for this special alias that cannot be guessed by someone trying to bypass list restrictions. If you don't care or just want to use consistent names for mailing list aliases, leave it set to No instead.
Sort mailing lists by	When this field is set to Name , mailing list icons on the module's main page will be sorted by name. If the default of Order create is chosen, they will be shown in the order in which they were added instead. Sorting by name makes lists much easier to find if you have a large number on your system.
Permissions for majordomo files	This field controls the permissions on all new files created by the module. If you select User writable , they will be set to mode 644. If you choose User and group writable , mode 664 will be used. Sometimes Sendmail fails if an alias include file is writable by anyone except its owner, in which case you will have to choose the User writable option.
Full path to majordomo config file	This field must contain the full path to the <code>majordomo.cf</code> file, such as <code>/usr/local/majordomo/majordomo.cf</code> . This is the primary configuration file that contains the locations of list files and other global settings. You should only need to change it if you have Majordomo installed in a directory different from what Webmin expects.
Directory containing majordomo programs	This field must be set to the directory that contains programs such as <code>majordomo.pl</code> , <code>resend</code> and <code>majordomo_version.pl</code> .
Path to majordomo wrapper	If the setuid-root program <code>wrapper</code> is in the same directory as all the other Majordomo programs, this field can be set to In programs directory . If it is installed elsewhere, however, you must enter the full path to the program, such as <code>/usr/bin/wrapper</code> .

Table 34.1 Module Configuration Options (Continued)

Sendmail-style aliases file	When Get from sendmail.cf is selected, the module will read the Sendmail configuration to find the location of the mail aliases file. If you are running a different mail server such as Postfix or Qmail, however, this will not work. Instead, you must enter the path to the server's aliases file into this field. See Section 34.3 "Using Other Mail Servers" for more details.
Directory containing sendmail safe programs	On some systems, Sendmail uses the SMRSH program to restrict the directory in which programs run from aliases can reside. Because Majordomo is driven entirely by alias programs, the module needs to create a link from this directory to the real program location when creating a mailing list. If you are using SMRSH on your system, this field must be set to the restricted directory, usually <i>/etc/smrsh</i> .

The MySQL Database

In this chapter, the MySQL database and the Webmin module managing it are explained, and the instructions for creating databases, tables, and users are provided.

35.1 Introduction to MySQL

MySQL is a free, easy-to-use database server that supports multiple databases and tables, and allows clients to query them with SQL. It is most useful for programmers writing applications that need to use a simple database to store information. Popular languages like Perl, C, Java, and PHP all have APIs for accessing a MySQL database.

A MySQL server can host multiple databases and each database can contain multiple tables. A table in turn contains fields, each of which has a type and size. Tables contain records, each of which usually contains information about some object, such as a person, product, or purchase. Fields can be thought of as the columns in a table, and the actual records of data as the rows.

SQL (which stands for Structured Query Language) is a language for extracting data from—or updating data in—a database. Almost all databases use SQL and its syntax is generally the same across all the different database packages such as Oracle, PostgreSQL, and MySQL. This chapter does not cover the SQL syntax, however, as it is too complex. There are plenty of other good books devoted entirely to it.

Compared to other databases, MySQL lacks some features. It does not support transactions for most table types, which means that every SQL command is executed immediately and cannot be undone. It cannot execute certain complex SQL commands, particularly those that involve nested queries. Other databases like PostgreSQL and Oracle support transactions and more complex SQL and deal better with extremely large tables. Chapter 36 explains how to use PostgreSQL, but Oracle (being an expensive commercial product) is not covered in this book.

Packages for MySQL come with almost all Linux distributions and it can be compiled on most UNIX variants. Its behavior is identical on all systems, with the exception that some versions of UNIX support larger table sizes due to their filesystems' support for larger files. On a standard ext2 or ext3 Linux filesystem, a table cannot be bigger than 4 GB. The Webmin MySQL module, however, will behave exactly the same on all operating systems.

MySQL is divided into two parts—the server that manages the actual files containing tables and records, and client programs that communicate with a server. The standard `mysql` client program allows users to execute SQL commands and display their results, while the `mysqladmin` program is for performing basic administrative tasks and the `mysqldump` program is for making backups. Other applications that query the database (such as Webmin itself) are also clients.

The data files in which tables are actually stored are located in subdirectories under a directory like `/var/lib/mysql` or `/usr/local/mysql/var`. These files are never read or written to by any programs except the MySQL server and should not even be copied for backup purposes unless the server process has been shut down.

35.2 The MySQL Database Server Module

This module allows you to create databases, tables, and fields, edit records, and manage MySQL users through a simple web interface. Its icon can be found under the **Servers** category, and when you click on it, the module's main page will display a table of icons for existing databases as shown in Figure 35.1 (assuming MySQL is installed and running).

If the MySQL server process is not running, the message **MySQL is not running on your system** will appear on the main page instead. To start it, just hit the **Start MySQL Server** button at the bottom of the page. If you want to make use of the database in future, use the Bootup and Shutdown module (covered in Chapter 9) to have it started at boot time. On Linux systems, the MySQL package will probably include a `mysql` action that you can easily enable.

If the database server is running but Webmin does not know the correct password to log in to it with, the main page will display a **MySQL Login** form instead. You should enter the administration username into the **Login** field (usually `root`), and the corresponding password into the **Password** field. Even though it is possible to enter the username and password for any MySQL user, non-root users cannot perform tasks such as creating databases and tables, so neither will the module be able to.

By default, the module is configured to log in with the username and password that the MySQL package for your distribution uses by default. Only if you have changed it manually or through Webmin will the **MySQL Login** page appear.

If the database server is not installed at all on your system, the main page will display an error message like **The MySQL client program /usr/bin/mysql was not found on your system**. Check your distribution CD or website for all MySQL-related packages and install them using the Software Packages module. Often there are several, named something like `mysql`, `mysql-client`, `mysql-server` and `mysql-devel`. Each Linux distribution seems to use a different set of packages, so make sure you install them all.

On FreeBSD and NetBSD, the module expects the MySQL package for those operating systems to be installed. On other UNIX variants, it assumes that you have compiled and installed MySQL from the source code distribution, available from www.mysql.com/.



Figure 35.1 The MySQL Database Server module main page.

If the module complains that it cannot find the `mysql` program even though you have it installed, you will need to adjust the paths that it uses. This can happen if you installed it from the source instead of using the package that comes with your Linux distribution. See Section 35.14 “Configuring the MySQL Database Server Module” for details.

The MySQL module uses SQL commands to perform actions like creating tables, adding fields, and editing records. Webmin must connect to the database server to execute these commands, which can be done in one of two ways. It can either run the `mysql` command with the correct parameters and parse its output or use the Perl DBI library to connect directly.

The former method is always available, because the `mysql` command is always installed when the database server is. It is not totally reliable, however, as certain kinds of table data produce output that cannot always be parsed. For this reason, you should install the `DBI` and `DBD::mysql` Perl modules. If either is missing, a message will be displayed at the bottom of the main page prompting you to install one or both by clicking on a link. This will take you to a page in the Perl Modules module (covered in Chapter 27) where `DBI` and/or `DBD::mysql` are downloaded and installed for you.

35.3 Creating a New Database

When MySQL is first installed, a database called `mysql` is created that contains authentication and access control related tables. If you want to store your own data, it is best to create your own database to which to add tables instead of messing with the `mysql` database.

To do this, follow these steps:

1. On the module's main page, click on the **Create a new database link** above or below the table of existing database icons. This will take you to a form for entering the new database's details.
2. Enter a name for the new database into the **Database name** field. Names should contain only letters and numbers, and no spaces.
3. It is possible to use the form to create an initial table for the new database. You can, however, just as easily add one after it is created, as Section 35.4 "Creating a New Table" explains.
4. Click the **Create** button at the bottom of the form to create the database. You will be returned to the module's main page, which will now include a new database icon.

35.4 Creating a New Table

Tables can be added to newly created or existing databases at any time. Every table has one or more fields, each of which has a type (such as integer, decimal, or text) and a size. Fields can also be indexed to speed up SQL queries that look up records based on the values in that column.

To add a new table to a database, follow these steps:

1. On the module's main page, click on the database icon. This will bring you to the database editing page shown in Figure 35.2 that contains an icon for each existing table and buttons for performing various actions.
2. Enter the number of fields that you want your new table to have in the **Fields** text box next to the **Create a new table** button, and click the button. This brings up a form for entering the details of the new table and its initial fields.
3. Enter a name for this table into the **Table name** field. It should consist of letters, numbers, and the _ character, and must be unique within this database.
4. To have its fields copied from an existing table, select it from the **Copy fields from table** menu. Any additional fields that you enter below in the **Initial fields** table will be added after the copied one.
5. The **Table type** menu can be used to choose a different storage type for this table. The most commonly used types are:

MyISAM This is a standard table type for MySQL versions 3.23 and above. On operating systems that support large files, tables of this size can be approximately 2,000,000,000 GB in size. Table files are OS independent, keys can be 500 bytes long, and 32 key columns can be used in a single table.

ISAM This is an old standard MySQL table type, now replaced with MyISAM. An ISAM table file can only be 4 GB in size, keys can only be 256 bytes long, and a table can have at most 16 key columns.

Heap The data in Heap tables is stored only in memory. This makes them very fast, but useful only for temporary data as the table's contents will be lost if MySQL is shut down.

If you select the **Default** option, or if the chosen type is not supported by MySQL on your system, the **MyISAM** type will be used.

6. The **Initial fields** section is for entering the details of the actual fields that your new table will contain. Each row that you fill in defines a single field, based on the values that you enter under each of the following headings:

Field name You should choose a unique name for this field, which should consist of letters, numbers, and the `_` character. It is not a good idea to choose a name that is the same as an reserved SQL word, such as `select`, `update`, or `index`.

Data type From this menu, you must select the type for data in this field. The most common are `varchar` (for variable length text strings) and `int` (for integer numbers). See Section 35.6 “Field Types” for a complete list of supported types.

Type width This refers to the size of data that can be stored in this field. This has different meanings depending on the type. For example, the width is the maximum text length for a `varchar` field, but it is the maximum number of decimal digits for an `int` field. Once again, Section 35.6 “Field Types” covers widths in more detail.

If you leave this text box blank, the default width will be used. Many types (such as `blob`, `text`, and `date`) have fixed sizes and so should not have a width entered at all.

Primary key? If this box is checked, this field will be part of the primary key for the table. Key fields are indexed by MySQL so that SQL statements that refer to all primary keys in the table in the `where` clause run faster. However, no two records can have the same values in their primary key field(s).

Traditionally, the first field in a table is the key. Not all types can be used—typically, a primary key field is an `int` or `varchar`. All tables should have a primary key, so that data in them can be edited in Webmin.

Autoincrement? If this option is checked for a numeric field, MySQL will automatically insert a number one higher than the maximum in the table whenever a record is added (unless the record creation statement specifies a value explicitly). This can be useful for the automatic generation of ID numbers and is often enabled for primary key fields.

7. Once you have entered all fields, hit the **Create** button at the bottom of the form. If the table cannot be created for some reason, a SQL error message from MySQL will be displayed. This can occur if a field name is invalid or if a type width does not make sense for a type. If this happens, use your browser’s **Back** button to return to the form and fix the problems.

Once the table is successfully created, you will be returned to the database editing page which will now include a new table icon.

35.5 Adding and Editing Fields

New fields can be added to a table and existing ones changed or deleted. Adding a field poses no risk to existing data, but changing the type or size of one may, and deleting a field will cause the data that it contains to be lost.



Figure 35.2 The database editing page.

To add a new field, follow these steps:

1. On the module's main page, click on the icon for the database that contains the table, and then on the table icon. This will bring up the page shown in Figure 35.3, which lists the names, types, and other details of all existing fields.
2. Select the type for the new field from the menu next to the **Add field of type** button before clicking it. See Section 35.6 “Field Types” for a list of types and their purposes.
3. On the field addition form that appears, enter a unique name for this field into the **Field name** text box. No two fields in the same table can have the same name, and only letters, numbers and `_` can be used.
4. If you are adding a **char** or **varchar** field, you must enter a maximum number of characters into the **Type width** text box.

If adding a **float**, **double**, or **decimal** field, you must enter two numbers into the **Width and decimals** text boxes. The first is the total number of digits that a value can contain and the second is the number of digits to the right of the decimal point. For negative numbers, the minus sign counts as a digit, so a field with **Width and decimals** set to 5 and 2 could store numbers from -99.99 to 999.99.

For **date**, **datetime**, **time**, **blob**, and **text** fields, there is no width input at all, as these types have fixed or unlimited sizes.

For **enum** and **set** fields, you must enter a list of possible values into the **Enumerated values** text box.

For all other field types (such as **int**) the **Type width** can be either set to **Default** to have the field use the default size for the chosen type, or a width can be entered. For **int** fields, this is the maximum number of digits that a value in this field can contain.

5. For integer field types (such as **int** and **smallint**), the **Type options** radio buttons allow you to choose whether values in this field should be left-filled with zeros (the **Fill with zeros** option) or if they should be unsigned (the **Unsigned** option). If **None** is selected, values will be signed and no additional zeros will be added.

For **float**, **double**, and **decimal** fields, the same **Type options** are also displayed but without the **Unsigned** option. Fields of these types are always signed.

For **char** and **varchar** fields, **Type options** has two different choices—**Case sensitive** and **Case insensitive**. If insensitive is selected, SQL queries that match values in this field will ignore case differences.

6. To prevent SQL NULL values being inserted into this field, change the **Allow null?** input to **No**. This can be useful if every record has a value for this field and must be selected if this field is going to be part of the primary key for the table.
7. To have a default value inserted when a record is added to the table and no value is specified for this field, fill in the **Default value** text box. Naturally, the value must be of the correct type for the field.
If your table already contains some rows, their values for this field will be set to whatever you enter here when the new field is added.
8. Change the **Part of primary key?** selection to **Yes** if this field is going to be the key for the table. More than one field can be part of the key, in which case the key is a combination of all of them.
9. Finally, click **Create**. If there are no errors in your inputs, the field will be added to the table and you will be returned to the table editing page shown in Figure 35.4.

Newly created or existing fields can be edited as well, by following the next set of steps. Making changes to the type of field or reducing its size, however, may result in data loss if the old values are not compatible with the new type. For example, converting a **varchar** to an **int** will cause all nonnumeric values to be lost—however, converting an **int** to a **varchar** is generally safe as long as the new size is large enough.

1. On the module’s main page, click on the icon for the database that contains the table, and then on the table icon. This will bring up the page shown in Figure 35.3, which lists the names, types, and other details of all existing fields.
2. Click on the name of the field that you want to modify to go to the field editing form.
3. To rename the field, edit the **Field name** text box.
4. To change the field’s type, select a new one from the **Data type** menu. As explained above, this should be done with care.
5. Depending on the current type, different inputs will be displayed for editing its size. These are the same ones as explained in Step 4 of the preceding field creation instructions.

Increasing the size of a field will not harm any data that it contains, but decreasing it will cause values to be truncated if they are longer than the new size.

Field name	Type	Allow nulls?	Key	Default value	Extra
<code>id</code>	<code>varchar(32)</code>	No	Primary		
<code>location</code>	<code>varbinary(16)/osname</code>	Yes	None		
<code>box</code>	<code>varbinary(64)</code>	Yes	None		
<code>ga</code>	<code>int(4)</code>	Yes	None		
<code>type</code>	<code>int(4)</code>	Yes	None		
<code>sect</code>	<code>datetime</code>	Yes	None		
<code>return</code>	<code>datetime</code>	Yes	None		
<code>sender</code>	<code>varchar(16)</code>	Yes	None		
<code>created</code>	<code>datetime</code>	Yes	None		
<code>createdby</code>	<code>varchar(16)</code>	Yes	None		
<code>modified</code>	<code>datetime</code>	Yes	None		
<code>modifiedby</code>	<code>varchar(16)</code>	Yes	None		

Figure 35.3 The table editing page.

6. The **Type options**, **Allow nulls?**, **Default value** and **Part of primary key?** inputs have the same meanings here as in the field creation steps. Change them if you want to adjust these options for the existing field.
7. When you are done, hit the **Save** button at the bottom of the form. The field will be immediately updated, and any data that it contains will be modified or truncated as appropriate.

An existing field can be removed by clicking the **Delete** button on the field editing form instead of **Save**. Any data that it contains will be immediately deleted forever. Naturally, you cannot delete the last field in a table.

35.6 Field Types

MySQL supports most of the same field types as other SQL databases. Table 35.1 lists all the types about which Webmin knows, and gives a short description of each one.

Newer versions of MySQL may introduce more types, but at the time of writing these are the only ones that can be used in new or modified fields in Webmin. You should still be able to edit the data in fields of unsupported types though.

35.7 Viewing and Editing Table Contents

The MySQL module allows you to view the contents of any table in any database. Tables that have a primary key can have their records modified or deleted and new ones added as well.

Table 35.1 MySQL Field Types and Their Uses

Type	Description
int	Stores a single integer, which can be signed or unsigned. An int field can, at most, store numbers in the range -2^{31} to $+2^{31}$, but this can be further restricted by specifying a maximum number of decimal digits when the field is added.
tinyint	Like an int , but only numbers in the range -128 to $+127$ can be stored.
smallint	Like an int , but only numbers in the approximate range -32768 to $+32767$ can be stored.
mediumint	Like an int , but only numbers in the approximate range -2^{23} to $+2^{23}$ can be stored.
bigint	A bigint is similar to an int , but supports the much larger maximum range of -2^{63} to $+2^{63}$.
float	<p>Stores a non-integer number, using the potentially imprecise floating point format. When a field of this type is added, you must specify the number of bytes used internally to store it, which changes the number of digits of precision that a float field can support. By default, a float occupies 4 bytes.</p> <p>If a number is inserted that contains more digits than the float can represent, it will be rounded to the nearest supported value. For this reason, you should not use a field of this type for storing numbers that must be recorded precisely, such as amounts of money. Instead, a decimal field (which has a specific decimal precision) should be used.</p> <p>You must also specify the number of digits to the right of the decimal point that should be stored and displayed for a float field. If a value with more fractional digits is inserted, it will be rounded off.</p>
double	A double field is like a float , except that it uses 8 bytes for internal storage and thus can be used to store values with more digits.
decimal	<p>A field of this type can store a noninteger number to a specific decimal precision. When adding one, you must enter the total number of digits to store and the number to the right of the decimal point. Any inserted value that uses more digits will be truncated or rounded off.</p> <p>Because decimal fields are stored internally as strings, using them in mathematical SQL expressions will not be as fast as using a float or double. They do, however, have a guaranteed known precision.</p>

Table 35.1 MySQL Field Types and Their Uses (Continued)

Type	Description
date	A date field stores a day, month, and year. The year is always in 4-digit format, and if you attempt to insert a date with a 2-digit year, MySQL will convert it to 4 digits by adding 1900 or 2000, depending on whether the year is before or after 1970. Any valid date in the years 1000 to 9999 can be stored. When displayed or inserted, dates always use the format YYYY-MM-DD, such as <i>1970-01-24</i> .
datetime	Like a date field, this type stores the time as well. For insertion or display, datetime fields must always use the format YYYY-MM-DD HH:MM:SS.
timestamp	Fields of this type are typically used to store the date and time that a record was last modified. Unless a value is explicitly specified, MySQL will update any timestamp field in a record with the current time whenever it is inserted or updated. Internally, timestamp fields use the UNIX time format, which means they are limited to the years 1970 to 2037. When displayed, the format YYYYMMDDH-HMMSS is always used.
time	A time field stores only a time without a date. The HH:MM:SS format is always used for display and updates. Because they can be used to represent an elapsed period of time as well, the hours can range from -838 to +838.
year	This kind of field is used to store a 4-digit year, but only in the range 1901 to 2155.
char	This type stores a string of characters up to a specified length. Internally, any value in a char field is padded with spaces to the right, which are removed when it is displayed. This means that if you add a char field with a size of 100, it will use up 100 bytes for each row. The maximum size of a field of this type is 255.
varchar	Stores a variable-length string of characters, up to the maximum specified when the field is added. The number of bytes used internally is related to the amount of data inserted, instead of being fixed at the specified size. Like a char , the maximum size is 255 characters. varchar is a better field type than char in almost all situations and should be used for all storage of short text strings.

Table 35.1 MySQL Field Types and Their Uses (Continued)

Type	Description
blob	A blob (which stands for Binary Large Object) can store any kind of data up to a maximum 2^{16} bytes. The number of bytes used internally is proportional to the amount of data inserted.
text	This type is similar to a blob , but is used for storing text. The biggest difference between the two is that text fields are sorted and compared case-insensitively, while blob fields are case sensitive.
tinyblob	Like a blob , but can only store 256 bytes of data.
tinytext	Like a text field, but can only store 256 bytes of text.
mediumblob	Like a blob , but can store more data - 2^{24} bytes to be exact.
mediumtext	Like a text field, but can store 2^{24} bytes of text.
longblob	An even larger kind of blob , which can hold up to 2^{32} bytes of data.
longtext	An even larger kind of text field which can hold up to 2^{32} bytes.
enum	An enum field stores a single text value, which must be a member of a list that is specified when adding or modifying the field. Up to 65535 different possible values are allowed.
set	The set type is like enum , but fields of this type can store from 0 to 64 different values from the allowed list, instead of just a single value.

Unfortunately, there is no way to edit the contents of a table without a key, as the module needs some way of identifying specific records. All tables in a database should have one, however.

To view the contents of a table, follow these steps:

1. On the main page, click on the icon for the database that contains the table, and then on the icon for the table itself.
2. On the table editing form, click on the **View Data** button at the bottom. This will bring you to a page displaying the first 20 rows in the table.
3. If the table contains more rows than can be displayed on one page, the start and end of the visible range and the total number of rows will be displayed at the top. Next to it are left and right arrows for moving to the next or previous 20 records.

4. For large tables, a search form is also displayed at the bottom of the page. To use it, select a field name from the first menu and a comparison type from the second, and enter a value to search for in the final text box. When the **Search** button is clicked, only rows for which the chosen field matches will be displayed. To switch back to viewing all records, click the **Reset search** link that now appears above the table.

The **contains** comparison type finds records in which the field contains the entered text, while the **matches** type finds records for which the field value matches an SQL pattern as used in a `like` clause. In such a pattern, `%` matches any string of characters, and `_` matches any single character, just like `*` and `?` do at the shell prompt.

5. When viewing a large table, a button labeled **Jump to** is also displayed at the bottom of the page. If a number is entered into the adjacent field and the button is clicked, the display will move immediately to that row.

If the table has a primary key, this same page can also be used to edit, delete, or add records. Records to edit must first be selected using the checkboxes to the right of each row or the **Select all** and **Invert selection** links. When you click the **Edit selected rows** button, the page will be re-displayed with the values of all chosen records in text boxes. Make whatever changes you like and click the **Save** button at the bottom of the page to update the database. Or, hit **Cancel** if you want to stop editing without saving your modifications.

To delete records, select them using the same checkboxes and selection links, and click the **Delete selected rows** button. The chosen records will be immediately removed from the database with no further confirmation.

To add a new record, hit the **Add row** button below the table. An additional row will appear containing empty text boxes for you to enter new details. Clicking **Save** will add the new record to the table and move the display so that you can see the new row. You can also click **Cancel** if you change your mind about adding a record.

Records are normally edited or added in text fields that appear in the table in the appropriate columns. If you are editing a table that contains a `blob` or `text` field, however, or if the **Use vertical row adding interface** module configuration option is enabled, a different layout is used. Text boxes for fields are listed in a separate box inside or below the table instead, with field name labels to the right. For `text` or `blob` fields, a text box is displayed so you can enter multiple lines of text if necessary.

35.8 Deleting Tables and Databases

When a table is removed from a database, all records and fields that it contains will be lost. You can remove any table, although deleting those in the `mysql` database is a bad idea as they contain important MySQL access control information.

To remove one, follow these steps:

1. On the module's main page, click on the icon for the database from which you want to remove the table, and then on the icon for the table itself.
2. Click on the **Drop Table** button below the list of fields. This will take you to a confirmation page that asks if you are sure and tells you how many records will be deleted.
3. To go ahead, click the **Drop Table** button again. Once it has been removed, you will return to the list of surviving tables in the database.

It is also possible to delete an entire database and all the tables and records in it. Any database can be removed, but deleting the `mysql` database is a very bad idea. As usual, unless you have made a backup there is no way to undo the deletion.

Assuming you really want to delete a database, follow these steps:

1. On the main page, click on the icon for the database that you want to remove.
2. Hit the **Drop Database** button below the list of tables. A confirmation page will be displayed, telling you how many tables and records will be lost if you go ahead.
3. To continue with the deletion, click the **Drop Database** button and you will return to the module's main page when it is done.
4. Alternately, you can choose to remove all the tables and their records by clicking on **Just delete all tables** instead. The database itself will be left empty.

35.9 Executing SQL Commands

The MySQL module also provides a simple interface for running SQL commands on a database and displaying their output. To use it, follow these directions:

1. On the main page, click on the icon for the database in which you want to run commands.
2. Click on the **Execute SQL** button below the list of table icons. This will take you to a page for entering SQL commands, running files of commands, and loading data into the database.
3. Enter any one SQL command into the text box at the top of the page and hit the **Execute** button. If there was a mistake in your SQL syntax or the command cannot be executed, the error message from MySQL will be displayed. Otherwise, a table of results from the SQL (if any) will be shown. Only SELECT statements produce results—UPDATE, INSERT, and other commands that modify records do not.
4. When you are done viewing the results, use the **Return to Execute SQL form** to return to the form.
5. Every command that is executed successfully is added to a history for the database. You can rerun a previous SQL command by leaving the text box empty and selecting it from the menu below, then hitting **Execute**.

To clear out the command history, click the **Clear History** button instead. This can be useful if it is getting cluttered up with old statements that you don't need to reuse.

The same page can be used to run multiple commands from a text file and display their output. Because the process is exactly the same as restoring a backup, it is explained in the restoration part of Section 35.10 “Backing Up and Restoring a Database”.

35.10 Backing Up and Restoring a Database

If one of your databases contains important information, it should be backed up regularly in case of a disk failure or SQL mistake, which could cause data loss. It is also a good idea to create a backup before performing some potentially risky operation, such as changing the type of a field or running a complex SQL statement that modifies lots of records.

To use the module to make a backup, follow these steps:

1. On the main page, click on the icon for the database that you want to back up.
2. Click on the **Backup Database** button below the list of tables. This will take you to a form for entering the backup destination and options.
3. In the **Backup to file** field, enter the full file path to which the backup should be written, such as `/tmp/backup.sql`. If the file already exists, it will be overwritten.
4. To restrict the backup to only some records, deselect the **All rows** option for the **Only backup rows matching where clause** field and enter an SQL WHERE clause into the adjacent field, for example `foo = "bar"`. This only works if the clause is valid for all tables in the database, so in the example all tables would need to have a `foo` field.
5. If the **Add drop table statements to backup?** field is set to **Yes**, the backup will include SQL statements to delete existing tables of the same name when restoring. This means that if you restore it on another system, data in those tables will be replaced with the new data from the backup. If **No** is selected, the restored data will be added to any that already exists.

The best choice really depends on what you are trying to do. For a normal backup, you should select **Yes** so that any corrupt or conflicting data is removed when the backup is restored. If you are transferring records to another system or database, however, **No** should be selected instead so that existing records in the target table are not lost.
6. To start the backup, hit the **Backup Now** button at the bottom of the form, and a page showing its success or failure will be displayed.

MySQL backup files are, in fact, just lists of SQL `CREATE TABLE` and `INSERT` statements that, when run, restore the database to the state it was in when the backup was made. Although this uses more disk space than a more compressed binary format would, it allows you to easily view and modify the file if you wish. It also means that a backup file can be used on a system with a different architecture, as the file contains only ASCII text.

If you have a database that is being used for an important production purpose, it should be backed up regularly, such as once per day. Instead of following the preceding instructions every day, you can use the Scheduled Cron Jobs module (covered in Chapter 10) to create a job that does the backup for you. To find out what command to run, use the preceding instructions to make a backup first and then visit the Webmin Actions Log module (covered in Chapter 54) to see the command that it used.

Once a backup file has been created, it can be restored on the same system or on another server running MySQL. Depending on what the **Add drop table statements to backup?** field was set to at backup time, the contents of any existing tables with the same names as those in the backup may be deleted. Therefore, you should generally only restore if the tables do not exist or contain outdated or invalid data that you want to overwrite.

Because a backup file is just a list of SQL statements, the restoration process just involves running all the commands in the file. This means that you can use the following same steps to execute a file of your own commands as well:

1. On the module's main page, click on the icon for the database into which the backup should be restored.
2. Click on the **Execute SQL** button, and scroll down to the **Select SQL commands file to execute on database** section.

3. If the backup file is on the system running MySQL and Webmin, choose the **From local file** option and enter the full path to the file into the adjacent text field.

If the backup is on the PC that your browser is running on, choose **From uploaded file** and use the **Browse** button to select the backup file.

4. Hit the **Execute** button to restore the backup or execute the SQL commands in the file. A page listing all output from MySQL as the execution proceeds will be displayed. Generally there will be none unless an error occurs or the file contains `SELECT` statements.

35.11 Managing MySQL Users

Your MySQL database server requires all clients to authenticate themselves with a username and password before they can execute SQL commands. It has its own tables of users, passwords, and permissions that are consulted when a client tries to log in, rather than the UNIX user files `/etc/passwd` and `/etc/shadow`. Detailed permissions can be defined for each user to limit the kinds of SQL statements that he can use, the client hosts he can connect from, and the databases, tables, and fields that he can modify.

Typically after MySQL has been first installed, only the `root` user be able to log in. This user will have permissions to access all databases and tables and perform all actions, and so is generally used for administration purposes only. If you want to write an application that uses a database, it is a good idea to create another user for that purpose and set up the application to log in as that user.

The standard MySQL install also creates an **Anonymous** user with no password and access to databases starting with `test`. This special user is used for any login attempt for which no other matching user is found. Anonymous users are explained in more detail later in this chapter.

To add a user, follow these steps:

1. On the module's main page, click on the **User Permissions** icon. This will take you to a page listing existing users, as shown in Figure 35.4.
2. Click on the **Create a new user link** above or below the table to go to the user creation form.
3. In the **Username** field, select the second radio button and enter a name for this user. Even though it is possible to create multiple user entries with the same name (as explained later), this new one should be unique.
4. Assuming you want the user to have a password, change the **Password** field to **Set to** and enter it in the adjacent field. If you choose **None**, no password needs to be given and attempts to log in with a password will be rejected.
5. To allow this user to log in only from a specific host, select the second radio button in the **Hosts** field and enter a hostname into the text box. The hostname must be the same as the one returned by a reverse lookup of the client's IP address, which will almost always be a complete hostname like `pc.example.com` instead of just `server`. You can enter an IP address instead, or a hostname or IP address SQL pattern like `%example.com`. To allow a user to connect only from the same system as the database server is running on, enter `localhost` as the host.

If **Any** is selected, this user will be able to connect from any host. Be careful when creating a user who has a host specified. If he tries to connect from somewhere else and



Figure 35.4 The MySQL user list.

an Anonymous user exists with a matching host, he will be logged in as the Anonymous user instead!

6. Select the entries for the actions that you want the user to be able to perform in the **Permissions** list. For an application user, being able to select, insert, update, and delete records is usually enough. Untrusted users should never be given permissions beyond **Drop tables**, as that would allow him to harm the database, access arbitrary files, or enhance his own permissions.

If a user does not have any permissions at all, he will be unable to connect unless some have been granted for a specific database or host (as explained in Section 35.12 “Managing Database, Host, Table, and Field Permissions”).

7. To create the user, click the **Save** button at the bottom of the page. The new MySQL login will be usable immediately and will have access to all databases and tables with the permissions specified in Step 6. See Section 35.12 “Managing Database, Host, Table, and Field Permissions” for information on how to restrict a user to only certain databases or tables.

When a client tries to log in, MySQL searches for the first matching user and host in the list of users. The server always checks entries with specific hostnames before those that allow any host and Anonymous user entries before those for a specific user. This means that a user may end up with the Anonymous permissions even though he is in the user list with greater privileges. Due to the confusion this can cause, I recommend deleting all Anonymous user entries unless you fully understand their effects.

It is possible and even useful to have multiple entries for the same user in the list, as long as they have different hostnames. For example, if you want to allow the user *fred* to log in from only *server.example.com* and *www.foo.com* clients, you would need to create two entries from *fred* with the **Host** field set differently. They should have the same password and permissions, however, unless you want to require a different password or grant different permissions depending on the host from which the user is connecting.

New and existing users can be edited by clicking on their names in the list, which brings up an editing form almost identical to the one used for creating a user. The only difference is that the **Password** field has a **Don't change** option that is selected if the user has a password and that tells Webmin to leave the password unchanged when the user is saved. After making changes, click the **Save** button at the bottom of the form to update the user in the database. Or, to delete it, hit the **Delete** button. If there are multiple entries for the same user, you will have to update them all individually when changing the password or permissions.

Unless you have already created another administration user with full privileges, the `root` user should not be deleted. Because this Webmin module normally logs in as `root` itself, modifying or removing this user may force you to log in to MySQL again, as explained in the introduction to the module earlier in the chapter. By deleting the `root` user or removing its privileges, it is possible to deny yourself access to the database, which can only be fixed using command-line programs like `mysqladmin`.

Like many other modules, the MySQL Database Server module can be configured to automatically create, update, or delete a MySQL user when the same thing happens to a corresponding UNIX user. This can be useful if you allow some of the UNIX users on your system to access databases and want to keep their passwords and usernames synchronized.

To set up synchronization, follow these steps:

1. On the module's main page, click on the **User Permissions** icon. Scroll down to the form below the list of existing MySQL users.
2. If you want a new MySQL user to be created for each new UNIX user, check the **Add a new MySQL user when a UNIX user is added** box. Then, select the permissions that should be granted to the user from the list to the right. When a MySQL user is automatically added, it will be allowed to log in from any host.
3. If you want MySQL users to be renamed or have their passwords changed when the same thing happens to matching UNIX users, check the **Update a MySQL user when the matching UNIX user is modified** box. If more than one entry exists for the same user, they will all be affected.
4. To have a MySQL user deleted at the same time as the UNIX user of the same name, check the **Delete a MySQL user when the matching UNIX user is deleted** box. If more than one entry exists for the same user, they will all be deleted.
5. Click the **Save** button to make the new synchronization settings active.

35.12 Managing Database, Host, Table, and Field Permissions

Users created by following the instructions in Section 35.11 “Managing MySQL Users” have access to all databases on the server with the same permissions. It is possible, however, to give a user access to only specific databases by following these steps:

1. Make sure the user does not have any permissions on the user permissions page. Any that he has set here will apply to all databases, which is not what you want.
2. On the module's main page, click on the **Database Permissions** icon. This will bring up a list of users and the privileges they have for specific databases.
3. Click on the **Create new database permissions** link above or below the list.
4. In the form that appears, the **Databases** field controls which databases he will have access to. You can either select the **Any** radio button to grant permissions for all databases, select the second radio button to grant access to the database selected from the menu, or choose the final button to grant access to databases whose names match the SQL pattern entered into the adjacent field.

Typically, the second option is the one that you want to select so you can grant access to a single database. If the user should have access to more than one, you will need to add multiple database permissions entries.
5. In the **Username** field, select the second radio button and enter the name of the MySQL user to whom access should be granted.
6. The **Hosts** field allows you to choose from which client host(s) the user will be able to connect to the database. You should normally select **Any**, which gives him access from anywhere unless the user himself is prevented from connecting from some hosts, as explained in Section 35.11 “Managing MySQL Users”.
7. From the **Permissions** list, select the privileges that the user should have for the chosen database. These will be added to any that are set for the user on the user permissions page.
8. Click the **Save** button to add and activate the new permissions. You will be returned to the database permissions list.

You can edit database permissions by clicking on a database name from the list. This will take you to an editing form identical to the creation form in which the database, username, hosts, or permissions can be changed. The **Save** button saves and activates any changes, while the **Delete** button removes the permissions from the database.

When MySQL is first installed, database permissions for the Anonymous user in the `test` and `test_%` databases will be created automatically. Assuming the Anonymous user exists on the user permissions page, these give anyone who can connect to MySQL access to records in those databases. Unless you are making use of anonymous logins, these database permissions can be safely deleted.

MySQL also allows permissions to be granted on databases to all users connecting from certain client hosts. This can be useful if you want to increase the privileges that a particular client system has, such as a web server connecting to your database server.

To add host permissions, follow these steps:

1. On the module's main page, click on the **Host Permissions** icon. This will take you to a page listing existing permissions granted to client hosts, if any. When MySQL is installed, no permissions of this type are initially defined.
2. Click on the **Create new host permissions** to bring up a form for adding a new host permissions entry.

3. If the permissions should apply to all databases, select the **Any** radio button in the **Databases** field.
If they are for only a specific database, select the second radio button and choose a database from the menu next to it.
If you want to grant permissions to databases whose names match a SQL pattern, select the final radio button and enter the pattern into the adjacent text field.
4. In the **Hosts** field, select the second radio button and enter a hostname, IP address, or hostname or IP pattern (like `%.example.com` or `192.168.1.%`) into the field next to it. Selecting the **Any** button isn't particularly useful.
5. From the **Permissions** menu, choose those privileges that will be granted to all users connecting to the chosen database from the specified host. These will be added to any other permissions that are granted on the user permissions or database permissions pages.
6. Click the **Save** button to activate the new client host permissions.

As usual, you can edit existing an host permissions entry by clicking on the database name from the list, editing fields, and clicking **Save**. You can also remove it with the **Delete** button.

MySQL also supports the granting of permissions to specific tables and fields to users connecting from certain hosts. Webmin allows you to set these up by clicking on the **Table Permissions** and **Field Permissions** icons on the main page. Because they are quite complex and rarely used, however, they are not covered in this chapter.

35.13 Module Access Control

Normally, a Webmin user who has access to the MySQL Database Server module can manage all databases and use all of the module's features. As Chapter 52 explains, however, it is possible to restrict what a user can do with a module. In this case, you can grant access to only certain databases, control the directory to which backups can be written, and restrict the creation and deletion of databases. This can be useful if various databases on your server are owned by different people and you want to give each of them a Webmin login to manage only those that belong to them.

To set up this kind of module access control, follow these instructions:

1. Click on MySQL Database Server next to the name of a user or group in the Webmin Users module who has access to the module.
2. On the access control form, change the **Can edit module configuration?** field to **No**. This is necessary to prevent the user changing the programs that the module uses for accessing the database.
3. In the **Databases this user can manage** field, choose the **Selected** option. Then, select the databases he should have access to from the list below.
4. Change the **Can create new databases?** field to **No**. There is no reason that a restricted user of this type should be able to add new databases.
5. Unless you want the user to be able to delete his own databases, change the **Can drop databases?** field to **No**. Leaving it set to **Yes** is harmless, though, as he will only be able to delete those to which you have granted him access.
6. Change the **Can stop and start MySQL server?** field to **No**.

7. If you want this Webmin user to be able to control access by MySQL users to his databases, change the **Can edit permissions?** field to **Only for managed databases**. This will give him access to the database, host, table, and field permissions pages, but limit him to viewing and editing entries for the databases to which he is granted access. To deny access to MySQL permission management altogether, select **No** instead. Choosing **Yes** is a bad idea, as it will allow the user to create MySQL users with access to all databases on the server.
8. If the **Can edit table data?** field is set to **No**, the user will not be able to create tables, edit fields, run SQL commands, or make backups. Instead, he will only be able to use the module's record-viewing and editing feature.
9. When the **Login to MySQL as** field is set to **Username from Module Config**, all database actions performed by this user will be done as the MySQL user set in the module configuration—typically `root`. You may, however, want the Webmin user to log in as a less-privileged MySQL user as an additional security precaution. This way, even if the user finds a way to defeat the module's restrictions, he will still not be able to execute SQL commands as `root`.
To use a different login, select the **Username** option and enter a valid MySQL login and password into the adjacent fields. This alternate user must have the privileges to perform everything that the module needs to do, however, such as creating tables and possibly granting permissions.
10. Normally, Webmin runs the `mysqldump` command to make backups as the `root` UNIX user and allows the backup file to be created anywhere on your system. Because this may allow important files to be overwritten, you should change the **Backup file directory** field to a safe directory in which to create backups, such as `/home/someuser/backup`.
Better still, the **Write backup as UNIX user** field should be changed to a user other than `root`, such as the Webmin user's UNIX login. The `mysqldump` command will be run as this user instead, which prevents it from being used to overwrite files.
11. Finally, to make the new access control restrictions active, click **Save**.

If you want to give a large number of users access to MySQL though a web interface, an alternative to configuring the Webmin module for each user is to install Usermin. It has a MySQL module with an identical interface, and can be easily configured to limit which databases are visible. See Chapter 47 for more information.

35.14 Configuring the MySQL Database Server Module

Like many other modules, this one has several options that you can set by clicking on the **Module Config** link in the top-left corner of the main page. Those fields listed under **Configurable option** relate to the module's user interface and the method it uses to connect to the database, while those under **System configuration** define the paths to the MySQL programs and files.

Unless you have installed the database server in a different directory from the default for your operating system, fields in the second section do not generally need to be changed. This can happen if you installed MySQL from the source code instead of using the package supplied with

your Linux distribution, or if you have two copies of MySQL installed and are configuring a clone of the module (covered in Chapter 51) to manage the second install.

If you have multiple copies of MySQL installed on your system, you should clone this module once for each server. The last three configuration options can then be customized to connect to each of the MySQL installs, which will probably be listening on different ports or using different socket files.

The names and purposes of all the module configuration fields are listed in Table 35.2.

Table 35.2 Module Configuration Options

Administration login	This field contains the username as whom the module logs into MySQL to perform all its actions. It is normally set by the MySQL Login form on the module's main page, which appears if Webmin cannot log in the first time you use the module. It does not usually need to be changed after that. If you change this field, make sure the new user has full access to all databases or some parts of the module will not work. You will probably need to change the Administration password field as well to match the new username.
Administration password	This is the password with which Webmin logs into the database. Like the Administration login field, this is normally set by the MySQL login form.
Number of rows to display per page	When viewing and editing records in a table, the module limits the number of records displayed at a time so the page does not become too large. To increase or decrease the number appearing on each page, change this field.
Show databases and tables as	Normally, databases on the module's main page and tables on the database editing page are shown as icons. If this field is changed to List , they will be shown in table form instead. Because a table takes up less screen space, this mode can be useful if you have a large number of databases or tables in your MySQL server.
Use vertical row editing interface	If this field is set to Yes , an alternate interface will be used for editing and adding records, as explained in Section 35.7 “Viewing and Editing Table Contents”.
Show blob and text fields as	When this field is set to Links to download , blob , or text fields in the table viewing page will not have their actual contents shown. Instead, a link to a separate page for viewing or downloading the contents of each page will appear. This can be useful if your MySQL database contains large binary objects such as GIF images or PDF files, which make no sense to display in an HTML table.

Table 35.2 Module Configuration Options (Continued)

Use DBI to connect if available?	When this field is set to Yes (as it is by default), the module will use the Perl DBI interface to connect to the database server as long as the appropriate Perl modules are installed. Only if DBI is not available will it use the <code>mysql</code> command to connect, as the output from that command cannot always be reliably parsed. Changing this field to No tells the module to always use the <code>mysql</code> command, even if DBI could be used instead. This should only be necessary if there is something wrong with the DBI modules on your system.
Path to mysqlshow command	This field must contain the full path to the <code>mysqlshow</code> command, such as <code>/usr/local/mysql/bin/mysql</code> . The module uses it to get lists of databases and tables for display on the main and database editing pages.
Path to mysqladmin command	This field must contain the full path to the <code>mysqladmin</code> command, which the module uses to check to see if the database server process is running.
Path to mysql command	This field must contain the full path to the <code>mysql</code> command, which is used to execute SQL commands if the Perl DBI modules are not installed.
Path to mysqldump command	This field must contain the full path to the <code>mysqldump</code> command, which the module uses to make backups of databases.
Path to mysqlimport command	This field must contain the full path to the <code>mysqlimport</code> command, which the module uses to import records from text files. This feature is not covered in this chapter, however.
Command to start MySQL server	This field must contain the full path to a command for starting the MySQL database server, such as <code>/usr/local/mysql/bin/safe_mysqld</code> . On many Linux distributions, it is set by default to the script that is used to start the server at boot time, like <code>/etc/init.d/mysql start</code> . If you have installed MySQL yourself, however, you should change it to use the <code>safe_mysqld</code> command instead.
Command to stop MySQL server	When this field is set to Automatic , the module will use the <code>mysqladmin</code> command to shut down the database server when the Stop MySQL Server button on the main page is clicked. Otherwise, the entered command will be run to shut it down. On Linux distributions that contain a shutdown script that can stop the database, this field is set to that script by default, such as <code>/etc/init.d/mysql stop</code> . If you have compiled and installed MySQL manually, however, you should choose the Automatic option.

Table 35.2 Module Configuration Options (Continued)

Path to MySQL shared libraries directory	When MySQL is installed from the source distribution, the <code>mysql</code> , <code>mysqlshow</code> , and other programs make use of shared libraries that are installed as well. For Webmin to be able to run them, it must set the <code>LD_LIBRARY_PATH</code> environment variable to include the directory containing those libraries, such as <code>/usr/local/mysql/lib</code> . Normally, the module's main page will display an error message warning you of a shared library problem if it is not set correctly. Because Webmin assumes that shared libraries are in the <code>lib</code> directory above the directory containing the <code>mysql</code> program, this field does not usually need to be set. Most of the time, it is only necessary if your MySQL commands depend upon other libraries in non-standard directories like <code>/usr/local/lib</code> .
MySQL host to connect to	When this field is set to localhost , the module will connect to the MySQL server on the same system. Entering a hostname tells it to connect to the server on that host instead, which can actually be a completely different system. Although most features of the module will work properly in this case, starting and stopping the database server will not.
MySQL port to connect to	This field is only used when connecting to another host. It defines the TCP port on which to connect to the database server. If Default is selected, the standard MySQL port is used, which is usually the right thing to do.
MySQL socket file	This field is only used when MySQL host to connect to is set to localhost . It specifies the UNIX socket file used for communication with the database server, which MySQL clients use instead of making a TCP connection. If Default is chosen, the default socket file path compiled into the <code>mysql</code> program or DBI module is used.

35.15 Summary

After reading this chapter you should understand what the MySQL database server can do and how it compares to other similar database packages. You will be able to use Webmin to create databases and tables stored in MySQL on your system and to view and edit records of data that those tables contain. You should also understand how to manage MySQL users and how the permissions system can be used to grant access to certain databases, tables, and fields to specific users.

The PostgreSQL Database

This chapter covers the PostgreSQL database server and explains how to use Webmin to manage tables, users, groups, and data.

36.1 Introduction to PostgreSQL

Like MySQL, PostgreSQL is a free database server that supports multiple databases and tables and allows clients to query them with SQL. It is most useful for programmers writing applications that need to use a database to store information. Popular languages like Perl, C, Java, and PHP all have APIs for accessing a PostgreSQL database.

A PostgreSQL server can host multiple databases, and each database can contain multiple tables. A table, in turn, contains fields, each of which has a type and size. Tables contain records, each of which usually contains information about some object—such as a person, product, or purchase. Fields can be thought of as the columns in a table, and the actual records of data as the rows. Some fields can also contain multiple values, like an array.

SQL (which stands for Structured Query Language) is a language for extracting data from or updating data in a database. Almost all databases use SQL, and its syntax is generally the same across all the different database packages, such as Oracle, PostgreSQL, and MySQL. Its syntax, however, is not covered in this chapter.

PostgreSQL has many features that other free databases (like MySQL) lack, such as transactions, array fields, views, and triggers. It is not quite as powerful as expensive commercial databases like Oracle, but it comes close. Because it can use multiple files to store each table, their sizes are not limited by the maximum file size on your system—instead, a single table can contain up to 16 TB of data.

Packages for PostgreSQL come with many Linux distributions, and they can be compiled and installed on almost all varieties of UNIX. Its operation is the same on all operating systems, and therefore so is the Webmin module for managing it.

PostgreSQL consists of a server process that reads and writes the actual database files and a set of client programs that communicate with the server. The most commonly used is the `psql` command, which allows a user to execute SQL queries and view the results. None of the clients access the database files directly—that is left entirely to the server.

All of the PostgreSQL database files are stored under a directory such as `/var/lib/pgsql` or `/usr/local/pgsql`. There are several text configuration files that affect the operation of the server and clients, as well. The most important is `pg_hba.conf`, which lists client hosts that are allowed to connect to the server. This is the only file that Webmin edits directly. All other database configuration is done by connecting to the database server, either directly or through the `psql` command.

36.2 The PostgreSQL Database Server Module

This module allows an administration to manage databases, tables, fields, and records in a PostgreSQL server. In many ways, it is very similar to the MySQL module covered in Chapter 35. When you click on its icon in the Servers category of Webmin, the main page displays a list of existing databases on your system, as shown in Figure 36.1.

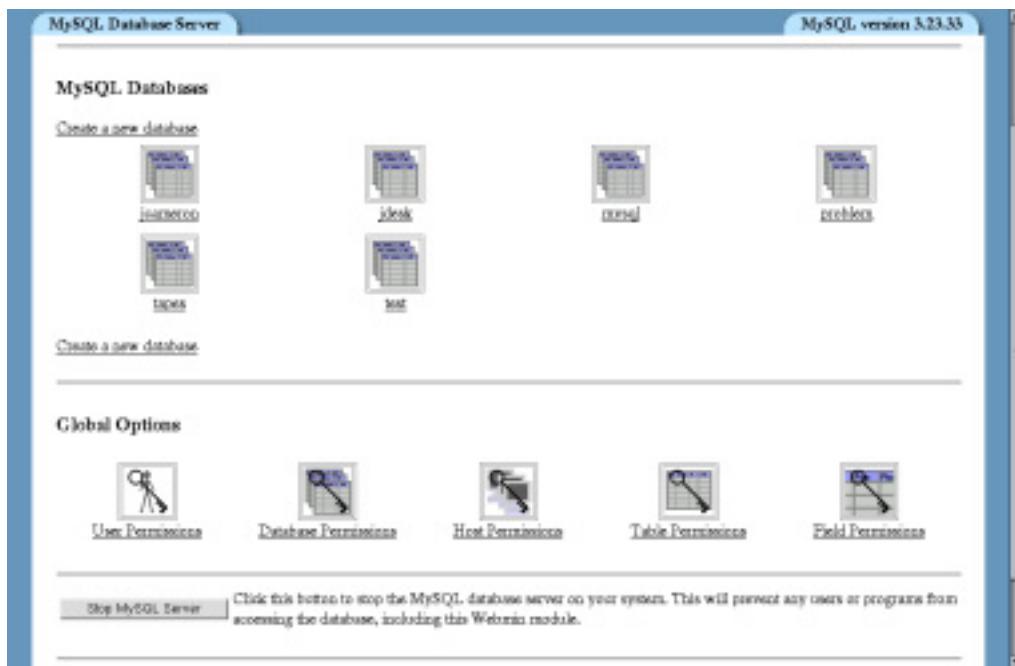


Figure 36.1 The PostgreSQL module main page.

If Webmin detects that PostgreSQL is not installed, has not been initialized, or cannot be connected to, the main page will not appear as shown in Figure 36.1. Instead, some kind of error message will be displayed. The most common ones are covered in the next few paragraphs.

If the message **PostgreSQL is not running on your system** appears, you will need to start the database server before this module can be used to manage it. Just click the **Start PostgreSQL Server** button at the bottom of the page. If you want it to be started at boot time from now on, use the Bootup and Shutdown module (covered in Chapter 9) to create a new action to start it. On most Linux distributions, the PostgreSQL packages include a bootup action script called `postgres`, or `postgresql`, that is not enabled by default.

If PostgreSQL is running through Webmin, Webmin does not know the administration username and password needed to connect to it, and the **PostgreSQL Login** form will be displayed on the main page. You must enter valid login details for your database server, typically for the `postgres` user who has full access to all databases and features. Logging in as another less privileged user may work at first, but you will not be able to use all of the features of the module. Sometimes PostgreSQL is set up to authenticate users by their UNIX username, rather than by a separate login and password (the `ident` authentication mode). If this is the case on your system, you will need to check the **Connect as same Unix user?** box on the form.

If an error message like **The PostgreSQL host configuration file hba.conf was not found** appears, then either the module's configuration is incorrect or your server has not yet been initialized for the first time. Many packaged versions for Linux systems need to be initialized before they can be used, usually by running the `initdb` command. If the module knows how to do this on your system, an **Initialize Database** button will be displayed that you can click on to set up the server for the first time.

The error message **The PostgreSQL client program psql was not found on your system** indicates that PostgreSQL is not installed at all, or that it is in a different directory from the one Webmin expects. On Linux and FreeBSD systems, the module assumes that you have installed the packages for the database included with your distribution, while on other operating systems it assumes that a standard installation from the source code was done into `/usr/local/pgsql`. If you have installed it and the error message still appears, you will need to read Section 36.17 “Configuring the PostgreSQL Database Server Module” for details on how to adjust the paths that the module uses.

If you are running Linux, and PostgreSQL is not installed, use the Software Packages module (covered in Chapter 12) to install all packages starting with `postgres` from the distribution CD or website. Often there will be several, such as `postgresql`, `postgresql-server`, and `postgresql-devel`. For other operating systems, visit www.postgresql.org/ to download the source code distribution, then compile and install it.

The PostgreSQL module uses SQL commands to perform actions like creating tables, adding fields, and editing records. To execute these commands, Webmin must connect to the database server, which can be done in one of two ways. It can either run the `psql` command with the correct parameters and parse its output, or use the Perl DBI library to connect directly.

The former method is always available, because the `psql` command is always installed when the database server is. It is not totally reliable, however, as certain kinds of table data produce output that cannot always be parsed. For this reason, you should install the `DBI` and `DBD::Pg` Perl modules. If either is missing, a message will be displayed at the bottom of the main page prompting you to install one or both by clicking on a link. This will take you to a page in the Perl Modules module (covered in Chapter 27) where `DBI` and/or `DBD::Pg` are downloaded and installed for you.

36.3 Creating a New Database

When PostgreSQL is first installed, a database called `template1` is usually created. Because this is used as the template for any new databases, you should create your own to contain tables in which your application can store data. To do this, follow these steps:

1. On the module's main page, click on the **Create a new database link** above or below the table of existing database icons.
2. Enter a unique name for it into the **Database name** field. This should consist only of letters, numbers, and the `_` character.
3. When the **Database file path** field is set to **Default**, the files that actually contain the database's data will be created in the default directory. On Linux systems, this will usually be something like `/var/lib/pgsql/data`—on other operating systems, it will probably be `/usr/local/pgsql/var`.

To use a different directory, as the UNIX user the database runs as (usually `postgres`), first create it with the `mkdir` command and then run `initlocation` with the directory name as a parameter. Then select the **section radio** button for the **Database file path** field and enter the directory in the adjacent text box.

Unfortunately, unless PostgreSQL has been compiled to support absolute data directory paths, an error will occur when you click the **Create** button. By default, this feature is not enabled.

4. Click the **Create** button. The database will be added, and you will return to the module's main page, which should include its new icon.

If a database called `template1` exists on your server, any tables or other objects that it contains will be copied to the newly created database. This can be useful if you want to add many databases with similar structures.

36.4 Creating a New Table

A table can be added to an existing database at any time. Each table has one or more fields, each of which has a type, a size, and other attributes. To add a table, follow these steps:

1. On the main page, click on the icon for the database to which you want to add the table. This will take you to the database editing page shown in Figure 36.2, on which is an icon for each existing table.
2. Enter the number of fields that you want your new table to have into the **Fields** text box next to the **Create a new table** button, and then click the button. This brings up a form for entering the details of the new table and its initial fields.
3. Choose a name for the table and enter it into the **Table name** field. The name must be unique within the database, and should use only letters, numbers, and the `_` character.
4. Each row of the **Initial fields** table defines a field that will be added to the new table. The kind of field that is added depends on what you put in the row under each of the following columns:

Field name The name for this field, which must be unique within the table and should be made up of only letters, numbers, and `_`. If left blank, no field will be added in this row.

Data type The selection that you make from this menu determines the type of data that can be stored in this field. The most common types are `varchar` for variable-length text strings, `int4` for integer numbers, and `float4` for fractional numbers. See Section 36.7 “Field Types” for more details.

Type width This field can be left blank, in which case the default size for the chosen type will be used. Otherwise, you must enter a number that is the number of characters (for `char` or `varchar` fields) or digits (for numeric fields) that the field can store. Some types, such as `blob` and `date`, do not need or allow a type to be specified at all. Again, see Section 36.7 “Field Types” for more information on the meanings of sizes for each type.

Field options If **Array?** is checked, this field will be an array capable of storing more than one value. If **Allow nulls?** is checked, the database will allow SQL NULL values to be inserted into this field. If **Primary key?** is checked, this field will be part of the primary key for the table. All tables should have a key, which is usually the first field and of `int` or `varchar` type. When **Unique?** is checked, PostgreSQL will prevent more than one record from having the same value for this field. Primary key fields are also automatically unique.

5. When you are done entering fields, hit the **Create** button at the bottom of the page. The table will be added to the database, and you will return to the page listing existing tables.

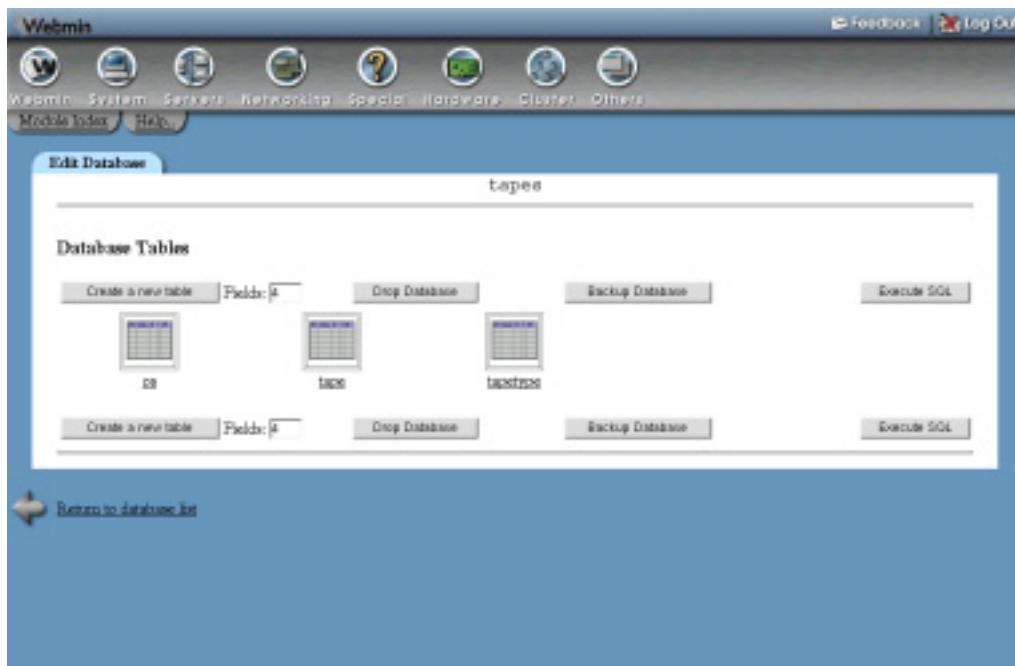


Figure 36.2 The database editing page.

36.5 Adding and Editing Fields

New fields can be added to tables, and the names of existing fields can be changed. There is no way to change the type of field size, though, unless you delete and re-add it. When a field is created, it will always initially contain NULL values in existing rows of the table.

To add a field, follow these steps:

1. On the module's main page, click on the icon for the database that contains the table. Then click on the icon for the table itself. This will bring you to a page listing the names, types, and sizes of existing fields, as shown in Figure 36.3.
2. Select a type for the new field from the menu next to the **Add field of type** button. When clicked, your browser will display the field creation form for entering the rest of the details.
3. Choose a name for this field that consists of only letters, numbers, and the _ character, and enter it into the **Field name** text box. No two fields in the same table can have the same name.
4. For a char or varchar field, enter the maximum number of characters that it can hold into the **Type width** text box. For a numeric field, you must instead enter two numbers separated by a comma, like 10,2. The first is the maximum number of digits that a number in this field can store, and the second is the number of digits to the right of the decimal point.

For other field types, the **Type width** text box does not appear at all.

5. If you want this field to be able to store multiple values of the same type, select **Yes** for the **Array field?** option.
6. Click **Create** to have the field added to the table, as long as there were no errors in your input.

Once a field has been created, you can only change its name, unlike in MySQL where its type or size can be modified. This means, however, that there is no risk to the data that it contains. To rename a field in a table, follow these steps:

1. On the module's main page, click on the icon for the database containing the table, and then on the **table** icon. This will bring you to the list of fields in the table, an example of which is shown in Figure 36.3.
2. Click on the name of the field that you want to change.
3. On the editing form, update the **Field name** text box with a new name. Naturally, this must follow the same naming rules as apply when creating a field.
4. Click the **Save** button to have the change made in the database.

36.6 Deleting a Field

Unlike MySQL, the PostgreSQL database server has no built-in SQL command for deleting a field from a table. It is possible, however, to carry out the removal of a field by creating a new temporary table that lacks the field, deleting the old table, and renaming the temporary table back to the original name. This works, and Webmin can do it all for you automatically; however, some information such as indexes and default field values will be lost in the process. The actual data in the table (apart from that in the deleted field), however, will be safe.

Field name	Type	Allow null?	Key	Default value	Extra
<code>id</code>	<code>varchar(32)</code>	No	Primary		
<code>location</code>	<code>enum('offline','online')</code>	Yes	None		
<code>box</code>	<code>varchar(64)</code>	Yes	None		
<code>ga</code>	<code>int(4)</code>	Yes	None		
<code>type</code>	<code>int(4)</code>	Yes	None		
<code>sect</code>	<code>datetime</code>	Yes	None		
<code>return</code>	<code>datetime</code>	Yes	None		
<code>number</code>	<code>varchar(16)</code>	Yes	None		
<code>created</code>	<code>datetime</code>	Yes	None		
<code>createdby</code>	<code>varchar(16)</code>	Yes	None		
<code>modified</code>	<code>datetime</code>	Yes	None		
<code>modifiedby</code>	<code>varchar(16)</code>	Yes	None		

Figure 36.3 The table editing page.

If your table does not contain any indexes or fields with default values, you can go ahead and remove a field by following these steps:

1. Click on the icon for the database containing the table on the module's main page, and then on the table icon itself.
2. Click on the **Drop Field** button on the table editing form, below the list of existing fields. This brings you to a page listing all the fields in table, each of which has a **radio** button next to it under **Drop This One**.
3. Select the **radio button** field that you want to remove from the table.
4. Check the **Select box to confirm...** checkbox at the bottom of the form.
5. Hit the **Drop Field** button to remove the chosen field. Once it has been deleted, the same page will be redisplayed so that you can remove another if you wish.

36.7 Field Types

PostgreSQL has a large number of field types, all of which are supported by Webmin. Not all of them are particularly useful for the average database, however. Those that are commonly used are listed in Table 36.1.

PostgreSQL has several types for storing geometric objects, such as `point`, `path`, `box`, and `circle`, and types for network information such as `inet`, `cidr`, and `macaddr`. Fields of all these types can be created and edited using this module, even though they are not documented above. No other databases (such as Oracle or MySQL) support these types, however, so it may be wise to avoid them if you want your programs to be database-independent.

Table 36.1 PostgreSQL Field Types

Type	Description
char	A fixed length string of text, padded to the right with spaces, if necessary. Fields of this type must have a width, which determines the number of characters that they can store. This width cannot be greater than 254 characters. To store larger strings, use a <code>text</code> field.
varchar	Similar to the <code>char</code> type, except that text is not padded with spaces. This is the best type to use for the storage of short strings.
int2	A 2-byte integer, which can store numbers in the range -2^{15} to $+2^{15}$ (approximately 32,000).
int4	A 4-byte integer, which can store values from -2^{31} to $+2^{31}$ (approximately 2 billion).
int8	An 8-byte integer, which can store numbers in the range -2^{63} to $+2^{63}$.
float4	A number that can contain decimals, such as <i>12.34</i> . Because this type uses a 4-byte floating-point format internally, numbers with many digits cannot be stored accurately and will be rounded off to the nearest possible number. For this reason, <code>float4</code> and <code>float8</code> fields should not be used for values that must always be accurate, such as monetary values. The numeric type should be used instead.
float8	Like the <code>float4</code> type, but uses 8 bytes for storage and thus can handle values with more significant digits.
numeric	Fields of this type can accurately store decimal numbers up to a maximum number of digits. When adding a <code>numeric</code> field, you must specify the width as two numbers separated by a comma. The first is the total number of significant digits that it can store and the second is the number of digits to the right of the decimal point.
date	Stores a year, month, and day. Dates are always displayed by PostgreSQL in the format <code>YYYY-MM-DD</code> , and should be entered in that format as well.
time	Stores an hour, minute, and second. Times are always displayed in the format <code>HH:MM:SS</code> .
timestamp	Stores a year, month, day, hour, minute, second, and timezone offset. Values in a <code>timestamp</code> field are always shown in the format <code>YYYY-MM-DD HH:MM:SS+ZZ</code> , and should be entered in that format as well.
text	A field of this type can store an unlimited amount of text. No maximum size can be or needs to be specified.

36.8 Viewing and Editing Table Contents

The PostgreSQL module allows you to view and edit the contents of any table in any database, even those that do not have primary keys. Unlike the MySQL module, it can identify specific rows to edit using the special `oid` column, which contains a unique identifier for each record.

To view the contents of a table, follow these steps:

1. On the main page, click on the icon for the database that contains the table, and then on the icon for the table itself.
2. On the table editing form, click on the **View Data** button at the bottom. This will bring you to a page containing the first 20 rows in the table.
3. If the table contains more rows than can be displayed on one page, the start and end of the visible range and the total number of rows will be displayed at the top. Next to it are left and right arrows for moving to the next or previous 20 records.

Unlike the MySQL module, there is no way to search for records or jump to a particular row number on this page.

This same page can also be used to edit, delete, or add records. Records to edit must first be selected using the checkboxes to the right of each row, or the **Select all** and **Invert selection** links. When you click the **Edit selected rows** button, the page will be redisplayed with the values of all chosen records in text boxes. Make whatever changes you like, and click the **Save** button at the bottom of the page to update the database. Or hit **Cancel** if you want to stop editing without saving your modifications.

To delete records, select them using the same checkboxes and selection links, and click the **Delete selected rows** button. The chosen records will be immediately removed from the database with no further confirmation.

To add a new record, hit the **Add row** button below the table. An additional row will appear containing empty text boxes for you to enter new details. Clicking **Save** will add the new record to the table, and move the display so that you can see the new row. You can also click **Cancel** if you change your mind about adding a record.

36.9 Deleting Tables and Databases

This module also contains buttons for deleting a table from a database, or an entire database and everything in it. When a table is removed, all records and fields that it contains will be lost.

To remove a table, follow these steps:

1. On the module's main page, click on the icon for the database from which you want to remove the table, and then on the icon for the table itself.
2. Click on the **Drop Table** button below the list of fields. This will take you to a confirmation page that asks if you are sure and tells you how many records will be deleted.
3. To go ahead, click the **Drop Table** button again. Once it has been removed, you will return to the list of surviving tables in the database.

It is also possible to delete an entire database and all the tables and records in it. Any database can be removed, but deleting `template1` is a bad idea as the module connects to it when retriev-

ing the list of other databases and assumes that it will always exist. As usual, unless you have made a backup, there is no way to undo the deletion.

Assuming you really want to delete a database, follow these steps:

1. On the main page, click on the icon for the database that you want to remove.
2. Hit the **Drop Database** button below the list of tables. A confirmation page will be displayed, telling you how many tables and records will be lost if you go ahead.
3. To continue with the deletion, click the **Drop Database** button and you will return to the module's main page when it is done.

It is possible to remove the `template1` database if you change the **Initial PostgreSQL database** field on the module configuration to some other database that is not going to be removed.

36.10 Executing SQL Commands

The PostgreSQL module also provides a simple interface for running SQL commands on a database and displaying their output. The steps for using it are:

1. On the main page, click on the icon for the database in which you want to run commands.
2. Click on the **Execute SQL** button below the list of table icons. This will take you to a page for entering SQL commands, running files of commands, and loading data into the database.
3. Enter any one SQL command into the text box at the top of the page and hit the **Execute** button. If there was a mistake in your SQL syntax or the command cannot be executed, the error message from PosygreSQL will be displayed. Otherwise, a table of results from SQL (if any) will be shown. Only SELECT statements produce results—UPDATE, INSERT, and other commands that modify records do not.

Unlike the MySQL module, there is no command history or support for running multiple SQL statements from a file.

36.11 Backing Up and Restoring a Database

If one of your databases contains important information, it should be backed up regularly in case a disk failure or SQL mistake causes data loss. It is also a good idea to create a backup before performing some potentially risky operation, such as running a complex SQL statement that modifies lots of records.

Due to changes in the parameters of the `pg_dump` and `pg_restore` commands, the module only allows you to create and restore backups when using PostgreSQL versions 7.2 and above. If you are using an older release, the buttons explained in the following steps will not be visible.

To use the module to make a backup, complete the following steps:

1. On the main page, click on the icon for the database that you want to backup.
2. Click on the **Backup** button below the list of tables. This will take you to a form for entering the backup destination and options.

3. In the **Backup file path** field, enter the full file path that the backup should be written to, such as `/tmp/backup.tar`. The file must not already exist. If it does an error will occur when you hit the **Backup** button.
4. From the Backup file format menu, select the type of file that should be created. The available options are:

Plain SQL text The file will contain a series of SQL commands that recreate the tables in the database and repopulate them with data. This format is convenient in that backup files can be manually edited, but you cannot include large objects (like blobs) in an SQL backup or selectively restore from it.

Tar archive The backup file will be a standard UNIX tar file, containing various files that specify table structures and contents. Large objects are supported, and selective restoring is possible.

Custom archive The file will be in PostgreSQL's custom backup format, which is compressed and supports large objects, data exclusion, and reordering at restore time.

5. To make the backup, hit the **Backup** button at the bottom of the form. If everything goes well, you will be redirected to the table list—otherwise, a page showing the backup command run and its error output will be displayed.

If you have a database that is being used for an important production purpose, it should be backed up regularly, such as once per day. Instead of following the preceding instructions every day, you can use the Scheduled Cron Jobs module (covered in Chapter 10) to create a job that does the backup for you. To find out what command to run, use the instructions above to make a backup first and then visit the Webmin Actions Log module (covered in Chapter 54) to see the command that it used.

Once a backup file has been created, it can be restored on the same system or on another server running PostgreSQL. The steps to restore a backup are:

1. On the module's main page, click on the icon for the database into which the backup should be restored.
2. Hit the **Restore** button below the list of tables to bring up a form for selecting the backup file.
3. In the **Backup file path** field, enter the full path to the file containing PostgreSQL backup data such as `/tmp/backup.tar`. This file can be in any of the formats available on the backup form.
4. Normally, the restore process will attempt to recreate tables before restoring data into them. To avoid this, change the **Only restore data, not tables?** field to **Yes**. This will only work if all the tables in the backup already exist. All data that the tables currently contain will be combined with restored records.
5. Normally, the restore process will fail if a table in the backup already exists in the database. To have existing tables dropped before restoration, change the **Delete tables before restoring?** field to **Yes**. It makes no sense to set both this and the previous field to **Yes**.
6. Click the **Restore** button to reload data and tables from the backup file. An error message showing output from the `pg_restore` command will be displayed if something goes wrong—otherwise, you will be returned to the list of tables in the database.

36.12 Managing PostgreSQL Users

As you would expect, the PostgreSQL database server does not simply allow anyone to connect and start manipulating data. Instead, it verifies clients by requiring them to send a username and password, which it checks against its own internal user list. This list of database users is totally separate from the UNIX user list in the `/etc/passwd` file.

By default, only the user `postgres` will exist, and he will have full access to all databases and tables. If you are writing an application that uses a database, a new user should be created for that application to log in as. If multiple people will be accessing your database using the `psql` command or other client programs, each should have his own login and password.

To add a new user, follow these steps:

1. On the module's main page, click on the **PostgreSQL Users** icon. This will take you to a list of existing users and their abilities, as shown in Figure 36.4.
2. Click on the **Create a new user link** above or below the list, which will bring up the user creation form.
3. Enter a unique name for the user, made up of only letters, numbers, and the `_` character, into the **Username** field.
4. To set a password for this user, select the second radio button in the **Password** field and enter a password into the text box next to it. If **None** is chosen, the user will not be able to log in unless the server has been configured to allow connections without a password (as explained in Section 36.14 “Restricting Client Access”).
5. If you want this user to be able to create his own databases, change the **Can create databases?** field to **No**. Only the master administration user (`postgres`) really needs to be able to do this.
6. To give this user the rights to edit and create PostgreSQL users, change the **Can create users?** field to **Yes**. Again, this should normally be left as **No**.
7. The **Valid until** field controls for how long this user can be used. If **Forever** is selected, it will have no expiry date—but if the second option is chosen and a date in `YYYY-MM-DD` format is entered into the text field, the account will not be usable after that date.
8. Click the **Create** button to have the new account added to PostgreSQL's user list. People or programs will be able to log in as this user immediately.
- Often the database server is set up by default to allow any local user to log in without needing to supply a password at all. To change this, see Section 36.14 “Restricting Client Access”.
9. To configure exactly which tables and views this new user can access, follow the instructions in Section 36.15 “Editing Object Privileges”.
10. Once a user has been created, it can be edited by clicking on its name in the user list shown in Figure 36.4. This takes you to the editing form that is almost identical to the user creation form, except that the user's name cannot be changed. Once you have finished modifying the password, expiry date, and other fields, hit the **Save** button to make the changes active.

A user can be deleted as well by clicking the **Delete** button on its editing page. Be careful not to remove the `postgres` user, as it is normally used by this Webmin module to log in to PostgreSQL.

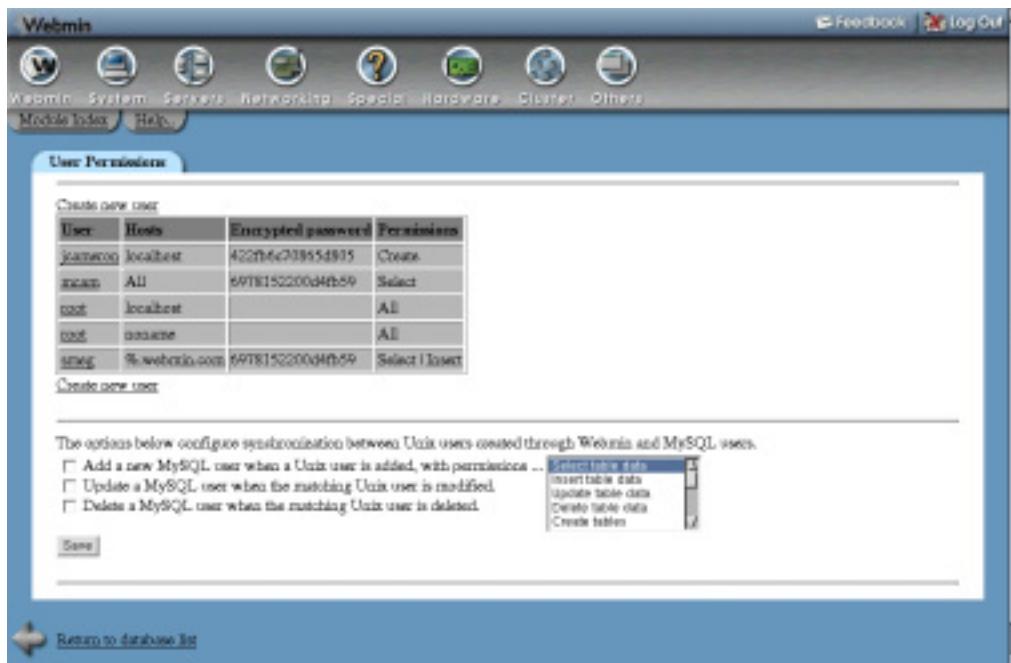


Figure 36.4 Existing PostgreSQL users.

In fact, even editing this user can cause problems if you set an expiry date or take away the user's ability to create databases or other users.

Like many other modules, this module can be configured to automatically create, update, or delete a PostgreSQL user when a UNIX user is added, modified, or removed, respectively. This can be useful if you allow some of the UNIX users on your system to access databases and want to keep their passwords in sync.

To set up synchronization, follow these steps:

1. On the module's main page, click on the **PostgreSQL Users** icon and scroll down to the form below the list of existing accounts.
2. If you want a new PostgreSQL user to be created for each new UNIX user, check the **Add a new PostgreSQL user when a UNIX user is added** box. Automatically created users will not, however, have any specific object permissions.
3. If you want PostgreSQL users to have their passwords changed when the same thing happens to matching UNIX users, check the **Update a PostgreSQL user when the matching UNIX user is modified** box.
4. To have a PostgreSQL user deleted at the same time as the UNIX user of the same name, check the **Delete a PostgreSQL user when the matching UNIX user is deleted** box.
5. Hit the **Save** button to make the new synchronization settings active.

36.13 Managing PostgreSQL Groups

PostgreSQL keeps its own internal list of groups, each of which can contain zero or more users. Groups are most useful when assigning object permissions, as they allow you to grant access to a table or view to many users at once. Apart from that, they perform no role in access control or authentication.

To create a group, complete the following steps:

1. On the module's main page, click on the **PostgreSQL Groups** icon. Your browser will display a table of existing groups and their members, if any. When PostgreSQL is first installed, no groups are defined.
2. Click on the **Create a new group** link to go to the **group creation** form.
3. Enter a name consisting of letters, numbers, and the `_` character in to the **Group name** field. No other group or user can have the same name.
4. Leave the **Group ID** field unchanged as Webmin automatically chooses the ID.
5. Select the users who will be members of this group from the **Members** list. In most browsers, you can **Ctrl-Click** to select more than one username, or **Shift-Click** to select an entire range.
6. Click **Create** to add the group. Object permissions can now be assigned to it, as explained in Section 36.15 “Editing Object Privileges”.

Just like a user, a group that you have created can be edited by clicking on its name in the list on the PostgreSQL Groups page, changing the name or membership list on the editing form, and hitting **Save**. It can also be deleted by clicking the **Delete** button on the same form.

36.14 Restricting Client Access

The default PostgreSQL configuration usually allows any user to connect to the database server from the same system without needing to log in, but prevents all remote access. If you want to allow clients to connect from other systems (for example, if you are setting up a database server that will be accessed from a separate web server), then PostgreSQL needs to be configured to allow this.

To grant access to another host, follow these instructions:

1. On the module's main page, click on the **Allowed Hosts** icon. You will be taken to a page listing hosts from which connections are allowed, the databases clients can access, and the authentication modes used. Typically, only local connections and those from `127.0.0.1` will initially be allowed.
2. Click on the **Create a new allowed host** link above or below the list to bring up the host creation form.
3. In the **Host address** field, select **Single host** and enter the IP or hostname of the remote client system into the adjacent field. You can also select **Network** and enter the network address (like `192.168.1.0`) and netmask (like `255.255.255.0`) into the fields next to it to allow an entire LAN.
4. To give the specified host or network access to all databases on your server, leave the **Database** field set to **All databases**. Otherwise, make a selection from the menu to limit the client to just that one.

If you want to grant a client access to two databases, you will need to add two host entries each with a different choice selected from **Database** menu.

5. In the **Authentication mode** field, select **Plaintext password**. The **No authentication required** option will also allow users on the client system to connect, but without needing to provide a valid password. Clearly, this is not very secure.
6. Hit the **Create** button to add the new allow host entry.

If your system has multiple users, each of whom has data in a PostgreSQL database that belongs to them, you should not allow them to log in to the database server without a password. By default, PostgreSQL allows exactly this, which is not particularly secure! Fortunately, it can be easily fixed. There is a risk that you will lock Webmin itself out of the database, however, as it is often set up by default to log in as the user `postgres` without a password.

Follow these instructions to reconfigure the module to log in with a password and to force local users to do the same thing:

1. On the module's main page, click on the **PostgreSQL Users** icon and then on the `postgres` user to bring up its editing form.
2. Select the second radio button for the Password field and enter a nice, secure password into the adjacent text field. Click **Save**.
3. Go back to the module's main page, and hit the **Module Config** link.
4. In the **Administration password** field, select **Set to** and enter the password you chose into the text field. Click **Save** at the bottom of the form.
5. Click on the **Allowed Hosts** icon, and then on **Local connection** in the **Host address** column. Change the **Authentication mode** field to **Plaintext password**, and click the **Save** button. After your browser returns to the list of allowed hosts, click on **127.0.0.1** and make the same change.
6. Return to the module's main page. If all went well, you will still be able to see and manage databases, and all users will require a password to connect.

When a client connects to the database server, PostgreSQL checks the host entries on the Allowed Hosts page in order. As soon as it finds one that matches the client address and requested database, the authentication mode for that entry is used. You can use this feature to block certain hosts while allowing all others by creating a host entry with the **Host address** field set to the IP you want to block, and the **Authentication mode** set to **Reject connection**. This entry must appear in the list above any broader entry that would allow the same client.

Because new allowed host entries are always added to the end of the list, the page has a feature for moving around. The up and down arrows under the **Move** column in the list can be clicked on to move an entry up or down one place, respectively.

36.15 Editing Object Privileges

Each PostgreSQL object (a table, view, index, or sequence) has an owner, who is the user who created it. By default, only the owner can select data from or update records in an object, which is not too useful if your server has multiple users who will all need access to the same tables. Fortunately, it is possible to grant access to database objects to other users or groups by following these steps:

1. On the module's main page, click on the **Granted Privileges** icon. Assuming you actually have some tables in your databases, this will bring up a page listing all existing objects and their current permissions.
2. Click on the name of the object to which you want to grant access, which will take you to its privileges editing form.

Hit the **Save** button at the bottom of the page to make the new permissions active. The **Grant privileges to** table lists all users and groups to whom access has been granted, followed by a blank row for adding a new user or group. Most of the time, however, it will just contain that one empty row.

In the **User** column, select the name of the user or group to whom to grant privileges from the menu, or choose **Everyone** to grant access to all PostgreSQL users.

In the **Privileges** column, check the boxes for the rights that should be granted to the chosen user or group. The available options are their meanings are:

SELECT When checked, the user will be able to view records in this table or view with an SQL **SELECT** query.

UPDATE When chosen, this option gives users the ability to update existing records in the table.

INSERT This option gives users the right to add new records to the table with an SQL **INSERT** statement.

DELETE When checked, the user will be able to delete existing records from the table.

RULE Allows the user to create rules on the table or view. A rule is a piece of SQL code that is executed to transform data inserted, updated, or deleted in the table.

REFERENCES Allows the user to create a field that references this table as a foreign key.

TRIGGER When checked, the user will be able to create triggers for this table.

Because the table only displays one empty row at a time, you will need to save and re-edit the object permissions if you want to grant access to more than one user. If several users are to be given the same permissions, it is better to put them in a group and grant access to the group, instead.

Unlike MySQL, there is no way to give a user access to an entire database, or just to a field within a table. All privileges are granted at the table level only.

36.16 Module Access Control

As Chapter 52 explains, it is possible to create a Webmin user who has access to only a subset of the features of some modules. In the case of the PostgreSQL Database Server module, you can limit a user to managing tables and fields in specific databases, and prevent him from editing users, groups, or granted permissions. This can be useful if different people own various databases on your server and you want to give each of them a Webmin login to manage only those that belong to them.

Once a user has been given access to the module, you can limit him to only certain databases by following these steps:

1. In the Webmin Users module, click on PostgreSQL Database Server next to the name of a user or group who has access to the module.
2. On the access control form, change the **Can edit module configuration?** field to **No**. This is necessary to prevent the user changing the programs that the module uses for accessing the database.
3. In the **Databases this user can manage** field, choose the **Selected** option. Then select the databases he should have access to from the list.
4. Change the **Can create new databases?** field to **No**. There is no reason that a restricted user of this type should be able to add new databases.
5. Unless you want the user to be able to delete his own databases, change the **Can drop databases?** field to **No**. Leaving it set to **Yes** is harmless though, as he will only be able to delete those to which you have granted him access.
6. Change the **Can stop and start PostgreSQL server?** field to **No**.
7. Change the **Can edit users, groups, hosts and grants?** field to **No**, so that he cannot create a new PostgreSQL user with access to all databases.
8. Set the **Can create backups?** field to **No**, as giving a user the rights to make a backup may allow him to overwrite files on your system.
9. The **Can restore backups?** field can be safely set to **Yes**, as there is no danger in allowing a user to reload data into his databases from a backup file.
10. Finally, to make the new access control restrictions active, click **Save**.

36.17 Configuring the PostgreSQL Database Server Module

Like most other modules, this one has several options that you can set by clicking on the **Module Config** link in the top-left corner of its main page. Those fields listed under **Configurable option** relate to its connection to the database and user interface, while those under **System configuration** define the paths to the PostgreSQL programs and files.

Unless you have installed the database server in a different directory to the default for your operating system, fields in the second section do not generally need to be changed. This can happen if you installed PostgreSQL from the source code instead of using the package supplied with your Linux distribution, or if you have two copies of PostgreSQL installed and are configuring a clone of the module (covered in Chapter 51) to manage the second install.

The names and purposes of all the module configuration fields are listed in Table 36.2.

Table 36.2 Module Configuration Options

Administration login	This field contains the username that the module logs in to PostgreSQL as in order to perform all its actions. It is normally set by the PostgreSQL Login form on the module's main page, which appears if Webmin cannot log in the first time you use the module. It does not usually need to be changed after that. If you change this field, make sure the new user has full access to all tables and features or some parts of the module will not work. You will probably need to change the Administration password field as well to match the new username.
-----------------------------	---

Table 36.2 Module Configuration Options (Continued)

Administration password	The password with which Webmin logs into the database server. Like the Administration login field, the PostgreSQL Login form normally sets this.
UNIX user to connect to database as	When Same as Administration login is selected, the module will connect to the database as the UNIX user with the same name as the database user in the Administration login field. This is necessary if your server has been configured to only perform <code>ident</code> authentication, which is the case on some Redhat and Debian Linux releases. If root is selected instead, all database connections will be done as the <code>root</code> UNIX user. This is the default, and should be used whenever the server is configured to authenticate clients by a username and password.
Number of rows to display per page	When viewing and editing records in a table, the module limits the number of records displayed at a time so that the page does not become too large. To increase or decrease the number appearing on each page, change this field.
Use DBI to connect if available	When this field is set to Yes (as it is by default), the module will use the Perl DBI interface to connect to the database serve, as long as the appropriate Perl modules are installed. Only if DBI is not available will it use the <code>psql</code> command to connect, as the output from that command cannot always be reliably parsed. Changing this field to No tells the module to always use the <code>psql</code> command, even if DBI could be used instead. This should only be necessary if there is something wrong with the DBI modules on your system.
Path to psql command	This field must contain the full path to the <code>psql</code> command, which is used to execute SQL commands if the Perl DBI modules are not installed.
Path to PostgreSQL shared libraries	When PostgreSQL is installed from the source distribution, the <code>psql</code> program makes use of shared libraries that are installed as well. For Webmin to be able to run them, it must set the <code>LD_LIBRARY_PATH</code> environment variable to include the directory containing those libraries, such as <code>/usr/localpgsql/lib</code> . Normally, the module's main page will display an error message warning you of a shared library problem if it is not set correctly. Because Webmin assumes that shared libraries are in the <code>lib</code> directory above the directory containing the <code>psql</code> program, this field does not usually need to be set. Most of the time, it is only necessary if your PostgreSQL commands depend upon other libraries in nonstandard directories like <code>/usr/local/lib</code> .
Initial PostgreSQL database	Because PostgreSQL always requires clients to specify a database when connecting, the field must contain the name of a database that always exists for Webmin itself to use when connecting. The default <code>template1</code> database will work fine, unless you decide to delete it.

Table 36.2 Module Configuration Options (Continued)

Command to start PostgreSQL	When the Start PostgreSQL Server button on the module's main page is clicked, the command in this configuration field is run. It must contain a valid shell command to start the database server as the correct user. On most Linux systems that have a package for the database, it will be set to use a bootup script like <code>/etc/init.d/postgresql start</code> by default. If you have installed PostgreSQL manually, however, you will need to change it to something like <code>su postgres -c "/usr/local/pgsql/bin/postmaster -i -S -D/usr/local/pgsql/var"</code> instead. This command runs the postmaster database server process with the correct parameters as the user <code>postgres</code> .
Command to stop PostgreSQL	This field controls what happens when the Stop PostgreSQL Server button on the main page is clicked. If Kill process is chosen, the module will simply kill the running postmaster server process. If the second option is selected, however, the shell command entered into the adjacent text field will be run. This is default on Linux systems that include a PostgreSQL package that has a shutdown script for stopping the server, such as <code>/etc/init.d/postgresql stop</code> . If you have compiled and installed PostgreSQL yourself, you should choose the Kill process option, as no such shutdown command or script exists.
Command to initialize PostgreSQL	This field specifies the command that the module runs when the Initialize Database button on the main page is clicked. Because this is only shown when the <code>pg_hba.conf</code> file is not found, the command is typically only run once. It must be set to some script or program that can initialize the database for the first time and create the <code>pg_hba.conf</code> file. On Linux distributions that have PostgreSQL packages, it is set by default to run a bootup script like <code>/etc/init.d/postgresql start</code> , which runs the correct <code>initdb</code> initialization command. If the option None is chosen, the Initialize Database button will not be shown even if the database needs initialization. If you have compiled and installed PostgreSQL yourself, this is the safest option, as the install process should set up the database and configuration files for you.
Path to postmaster PID file	This field must contain the full path to the <code>PID</code> file created by the postmaster server process, such as <code>/usr/local/pgsql/var/postmaster.pid</code> .
Path to host access config file	This field must be filled in with the full path to the <code>pg_hba.conf</code> file that contains host access restrictions, as explained in Section 36.14 “Restricting Client Access”. If it is not set correctly, the module will assume that PostgreSQL has not yet been initialized and display an error message on the main page.

Table 36.2 Module Configuration Options (Continued)

PostgreSQL host to connect to	When this field is set to localhost , the module will connect to the PostgreSQL server on the same system to view and manage databases. It is possible to enter a different hostname instead, however, in which case the module will connect to the server on that host. Not all features of the module can be used on a remote server, however. The Allowed Hosts page and the Start and Stop buttons are examples of this.
PostgreSQL port to connect to	This field can be used to have the module connect to a port other than the default when your database server is running on a non-standard port.
Path to pg_dump command	This field must contain the full path to the <code>pg_dump</code> command, used for backing up a database.
Path to pg_restore command	This field must contain the full path to the <code>pg_restore</code> command, used for re-loading the contents of a backup into a database.
Default backup repository directory	This field sets the default value for the backup form's Backup file path field. It can be useful if all your backups are made to the same directory.

36.18 Summary

This chapter has introduced the PostgreSQL database server, and explained its capabilities and limitations. It has shown how Webmin can be used to manage a PostgreSQL server on your system. After reading it, you should know how to create databases, create tables and fields, edit table data, manage users, groups and permissions, and create and restore backups.

Configuring Sendmail

In this chapter, a basic introduction to email is given, followed by a description of the Sendmail mail server and the Webmin module for configuring it.

37.1 Introduction to Internet Email

When you use a mail client program like Outlook or Evolution to send email, the program simply passes the message to a mail server for delivery to the destination. This server (also known as the MTA or Mail Transfer Agent) locates the correct system for the destination address, connects to the MTA on that system, and transmits the message. When the other server receives the email, it checks to make sure it is truly destined for this system and, if so, stores it in a local mail file.

Later, when the destination user checks his email, the file is read by a program such as Elm, Pine, or Usermin, or by a POP3 server. Mail clients like Evolution and Outlook are usually run on a different system from the mail server, and use the POP3 protocol to download messages for local storage. Once the email has been retrieved in this way, the delivery process is complete and the messages can be read by a user.

Mail transfer agents use the SMTP (Simple Mail Transfer Protocol) to send email to each other over the Internet. Clients also use SMTP to send mail to servers for onward delivery. Very few MTAs implement the POP3 protocol for mail retrieval. Instead, they simply write mail to a file that is read by a separate POP3 server program. Chapter 15 explains how to enable a POP3 server on your system, and Chapter 33 explains the protocol in a little more detail.

The mail server that a client contacts to send a message is usually on its local network, at the ISP the client system is connected to, or even the same system. Ideally, this first MTA will contact the destination server directly, but this is not always the case. The destination MTA may be down or unreachable, in which case email will be sent to an alternate server instead. SMTP for-

warding rules or per-user mail forwarding settings may cause an email message to be sent to other servers before it is finally delivered into a mailbox file.

If you want your system to be able to receive email, you will need to run an MTA program. This is only the first step. To run your own mail domain—such as *example.com*—so that mail to *foo@example.com* is delivered to the mailbox *foo* on your system, some network and DNS configuration is required. Typically, your system will need a fixed IP address and a permanent connection to the Internet. This means that running a proper mail server on a system that connects occasionally via dialup is impractical.

For other servers on the Internet to know to send mail to your system, appropriate DNS records must be created. Typically, an MX (Mail Server) record is defined for the domain—like *example.com*—that has the hostname of your system as its value, like *server.example.com*. This tells other MTAs to connect to your system to deliver email for the *example.com* domain. If you are running your own DNS server for the domain, Chapter 30 explains in detail how to create an MX record. Otherwise, you will have to tell whoever is hosting the domain (your ISP, for example) to add the correct record.

In fact, it is possible for any system to receive email addresses to its full Internet hostname, such as *jcameron@linuxbox.example.com*. As long as there is no MX record matching *linuxbox.example.com*, other mail servers will do a normal A (Address) record lookup for the hostname and connect directly to the system.

Sometimes it makes sense to run a mail server even if you have a dynamically assigned IP address or ephemeral Internet connection. The Fetchmail program (covered in Chapter 33) can be used to download email using the POP3 protocol and deliver it to a local mailbox on your system, which requires that an MTA be running. This local mail server may never accept a direct SMTP connection from another system on the Internet, but it can still deliver mail sent by programs on your system to local mailboxes.

Occasionally it is useful (and even necessary) to run a mail server that only deals with outgoing mail sent by local client programs, and not with delivering incoming messages. Instead of configuring mail clients to use a remote MTA (such as the one at your ISP), you can tell them to use your own system's mail server instead. It will accept messages from local clients and try to deliver them to their final destinations, or put them in a queue for later delivery. Some programs that send email can only use a local mail server, to which they connect by running the `sendmail` program.

Almost all ISPs and companies run their own mail servers. If you are happy to use an email address at your organization or ISP's domain, there is no need to run your own. Instead, you can simply configure your Linux mail client, like KMail or Evolution, to download mail from and send mail out via another server. For most people who just have a single email address and don't need to host their own email domain, there is no need to set up a mail server.

37.2 The Sendmail Configuration Module

Sendmail is the most popular MTA in use on the Internet today and has been since it was first developed. It is included as standard with almost all variants of the UNIX operating system, and works the same on all of them. It has many useful features for routing and processing email, such as aliases, domain routing, and user-creatable forward files.

Sendmail has a one-to-one mapping between UNIX users and mailboxes. Each user has his own mail file, typically in the `/var/mail` or `/var/spool/mail` directory. Each time a message is delivered to a user, it is appended to the file with the same name as the user in that directory, such as `/var/mail/jcameron`. Sendmail has no concept of “mail users.” If you want to create a new mailbox, you will need to add a new UNIX user as explained in Chapter 4.

Sendmail can accept email in two different ways, either from an SMTP connection or by another program invoking the `sendmail` command with the right parameters and feeding the email to it as input. Either way, the message is delivered to either a local user’s mailbox or to another system. As you would expect, if the Sendmail server process is not running, then it is impossible for email to be sent to your system via SMTP. Any queued email will also not be delivered.

Sendmail’s primary configuration file is appropriately named `sendmail.cf`, and is found in the `/etc` or `/etc/mail` directory. There are also separate text and DBM format files for local domains, mail forwarding aliases, address mappings, and other features discussed later in this chapter. Almost all of these additional files are actually in the UNIX DBM database format but are built from a corresponding text file that the system administrator (and Webmin) can edit. Sendmail only reads the DBM files though and rechecks them for every message received so that any change to one of the databases files becomes immediately active.

To set up Sendmail using Webmin, you will need to visit the Sendmail Configuration module, which can be found under the Servers category. Assuming you have the server installed, the module’s main page will look like the example shown in Figure 37.1.

If the module cannot find the Sendmail server program or primary configuration file on your system, an error message to that effect will be displayed instead. Check your Linux distribution CD or website for the `sendmail` package, and install it using the Software Packages module (covered in Chapter 12). If other packages whose names start with `sendmail-` are available (such as `sendmail-cf` or `sendmail-docs`), install them as well.

This error can also occur in the unlikely event that you have installed Sendmail or its configuration file in a different location than the one the module expects. By default, it assumes that you will use the packages that come with your operating system, but these are often out-of-date. For this reason, you may have compiled and installed the latest version in a different directory. If so, see Section 37.15 “Configuring the Sendmail Configuration Module” for instructions on how to change the program and configuration file paths.

Sendmail has gone through many different releases over the years, and in that time its primary configuration file (usually found at `/etc/sendmail.cf`) has changed. If you are running a very old version or using an old configuration file, the module’s main page will display an error message indicating that the file format is not supported. Configurations older than version 7 trigger this error, but fortunately they are rarely found on modern UNIX systems.

If no error message appears on the main page, the table of icons shown in Figure 37.1 will be displayed. Each can be clicked on to access one of Sendmail’s features, such as mail aliasing, domain routing, or the mail queue. The rest of this chapter explains how to use the pages and forms under each of the icons. Next to the name of each icon (such as **Address Mapping**) is the internal name of the Sendmail configuration feature in brackets that it controls, such as `virtuser`. These names are mostly useful to experienced administrators who want to know how the icons in the module relate to actual configuration files and directives.

When clicked on, some of the icons may display a message like **Your Sendmail configuration does not have the address mapping (virtuser) feature enabled**. On many operating sys-

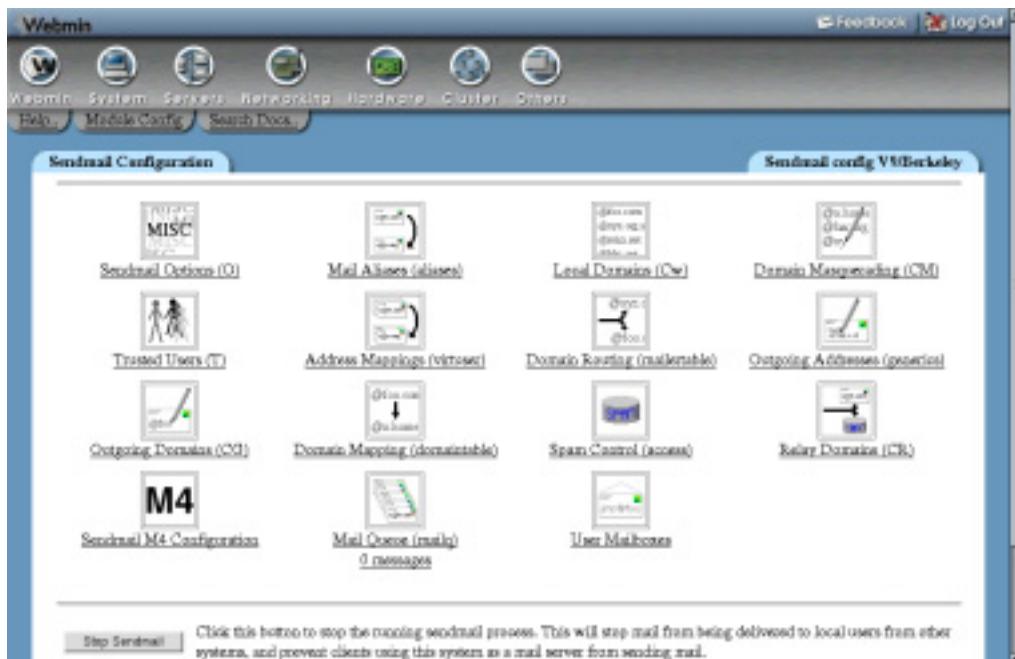


Figure 37.1 The Sendmail Configuration module.

tems, the primary Sendmail configuration file does not have all the available features activated by default. To make the chosen icon's pages available, follow the instructions in Section 37.11 “Adding Sendmail Features with M4”.

If the Sendmail server process is running, a button labeled **Stop Sendmail** will appear at the bottom of the main page. As its name suggests, clicking on this button will shut down the server so that your system no longer accepts SMTP connections and no longer scans the mail queue. When Sendmail is not running, a **Start Sendmail** button will appear instead, which can be used to start the server process.

If you want Sendmail to run all the time, use the Bootup and Shutdown module (covered in Chapter 9) to have its server process started at boot time. Most packages will include an action script that can be enabled, and may even be enabled by default. If yours does not, you will need to create an action that runs the command `/usr/sbin/sendmail -bd -q30m` at boot time.

37.3 Editing Local Domains and Domain Masquerading

When Sendmail receives an email message via an SMTP connection, it needs to work out whether it should be delivered locally or forwarded to another server. This is done by looking at the message's **To:** address—specifically the domain part after the @. The domain is compared a list of local domains, and if a match is found the email is delivered to the mailbox of the user whose name is to the left of the @ in the **To:** address. If no such user exists, a bounce message is generated and sent back to the original sender.

If the domain is not local, Sendmail will look up the mail server for the domain and attempt to connect to it in order to transfer the message. This is what usually happens when a client on the same network connects to send out email. A problem will occur, however, if Sendmail attempts to connect back to itself, which can happen if the DNS says that it is the mail server for a domain that is not on its local domain list. If this happens, a bounce message will be sent back to the sender, containing text like `mail loops back to me`.

By default, this local domains list contains only the full hostname of your system, such as `server.example.com`. If you are setting up a server to receive email from the Internet for a specific domain (like `example.com`), it will need to be added to the list. To do this, follow these steps:

1. On the module's main page, click on the **Local Domains** icon. A page containing a text box in which all current local domains are listed will be displayed.
2. Add as many extra domain or hostnames to the list as you like—one per line. It is quite possible for a server to accept mail for several domains, especially if it is going to be used for virtual hosting. As the introduction explains, mail will only be sent to your system in the first place if an appropriate MX DNS record exists for each domain.
3. Click the **Save** button at the bottom of the page to make them active.

Sendmail will always accept email messages for local delivery in which the **To:** address does not contain a domain and instead contains just a username. These are often sent by programs running on the system itself, such as the Cron daemon or the `mail` command.

The flip side of the local domains list is domain masquerading. This Sendmail feature allows you to set the domain that is added to email sent out from your system when none is specified, such as by the `mail` command. It is also possible to have Sendmail modify the **From:** address domains of messages received via SMTP, such as those send by mail clients.

To set up domain masquerading, follow these steps:

1. Click on the **Domain Masquerading** icon on the module's main page.
2. Fill in the **Masquerade as domain** field with the name of the domain that should be appended to outgoing **From:** addresses that lack one, such as `example.com`. If the field is left blank, Sendmail will not do domain modification.
3. To have Sendmail rewrite the **From:** addresses of messages from other domains, fill in the **Domains to be masqueraded** field. This can be useful if some of the mail clients that send out messages via your server insist on using the wrong domain.
4. Click the **Save** button to make masquerading active.

On most mail servers, you do not need to bother configuring masquerading, as all mail is sent by client programs using SMTP. All mail clients allow the user to specify a complete **From:** address, which should include the correct domain.

37.4 Managing Email Aliases

A mail alias specifies that email received by your server for a particular mailbox should be forwarded to a different destination instead. That destination can be another email address, a file of addresses, a local file, or even the input to a program. They can be useful for setting up pseudo mailboxes that actually send email to a real person, such as `sales@example.com` or

webmaster@example.com. An alias can have the same name as a UNIX user, in which case it will intercept all mail to that user and forward it to a different destination.

On most operating systems, Sendmail has several aliases defined by default for system users like `bin`, `nobody`, and `uucp`, all of which forward mail to `root`. There will also be a `postmaster` alias, which every mail server must have, and which should forward messages to someone responsible for the mail server. Typically, this will be the `root` user, as well.

To create a mail alias of your own using Webmin, follow these steps:

1. On the module's main page, click on the **Mail Aliases** icon. You will be taken to a page listing all existing aliases and their destinations, with a form at the top for adding a new one. Figure 37.2 shows an example.
2. In the **Address** field of the **Create Alias** form, enter the user or mailbox name for this alias (the part of the address to the left of the @). If your server hosts multiple domains, the alias will forward email sent to the entered name at any of those domains. For example, if your server accepts mail for `foo.com` and `bar.com`, then an alias called `sales` will forward email to both `sales@foo.com` and `sales@bar.com`.

If you want to be able to forward the same mailbox name differently at multiple domains, see Section 37.6 “Managing Virtual Address Mappings”.

3. Assuming you actually want this alias to be used by Sendmail, leave the **Enabled?** field set to **Yes**. Changing it to **No** will cause the alias to be ignored. This field can be used when editing an alias to temporarily disable it, rather than totally deleting it.
4. The **Alias to** field determines where email to this alias will be sent. The following options are available from the menu:

<None> Nothing at all will be done with received email. It makes no sense to select this option when creating a new alias.

Email address Email will be forwarded to the user or address that is entered into the adjacent field. Be careful not to set up a forwarding loop by sending email back to the alias' address again! If you are creating an alias that has the same name as a UNIX user and really do want email to be delivered to his mailbox as well as some other destinations, enter the username preceded by a backslash (like `\jcameron`) into this field. The backslash tells Sendmail to bypass alias checking.

Addresses in file Email to the alias will be sent to all the addresses in the text file whose file path is entered into the adjacent text field. Each address must be on a separate line. This option can be useful for creating a simple mailing list, and the Majordomo list manager uses aliases of this type (covered in Chapter 34).

Write to file The full text—including all headers—of email received by the alias will be appended to the file whose path is entered into the text box.

Feed to program The program whose path and parameters are entered into the text box will be run and the full text—including all headers—of email received by the alias will be fed to it as input. This kind of alias is most useful to programmers who want to perform their own custom processing or filtering of email messages. The program is usually run as the UNIX user daemon, not `root`, or the user with the same name as the alias.

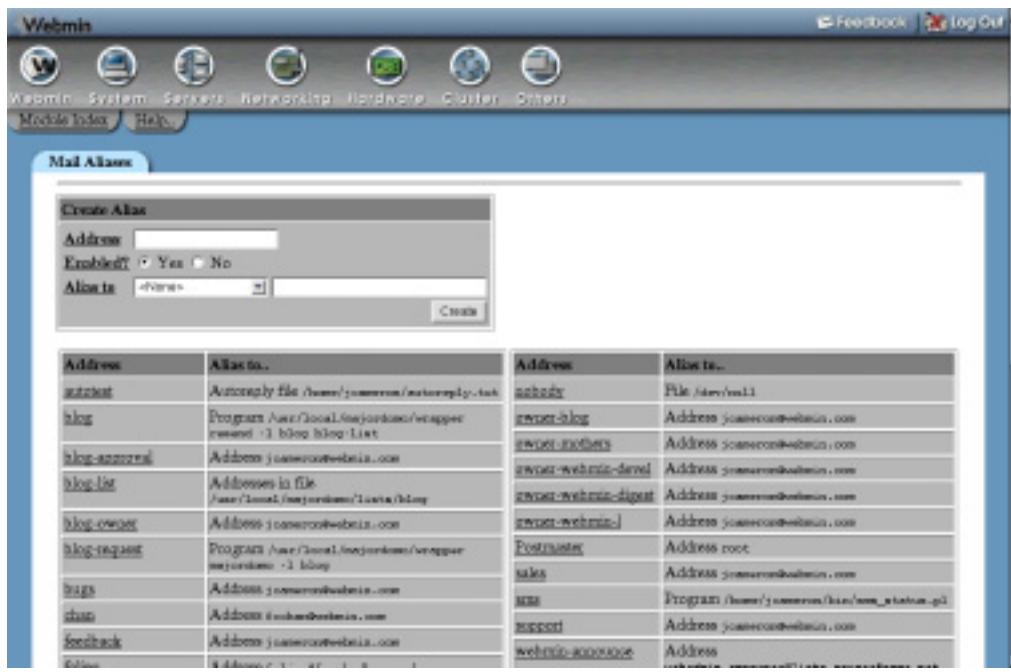


Figure 37.2 The mail aliases list.

Autoreply from file When email is sent to the alias, the contents of the file specified in the adjacent text box will be sent back to the original sender. See Section 37.12 “Creating Autoreply Aliases” for more information on using aliases of this type.

Apply filter file Email sent to the alias will be processed according to the rules in the filter file entered into the text box, which can forward to different destinations depending on the message contents. See Section 37.13 “Creating Filter Aliases” for more details. It is possible for an alias to have multiple destinations. To add more than one, you will need to re-edit this alias after saving it and fill in the row with <None> selected at the bottom of the **Alias to** table.

5. Click **Save** to have the alias added to the list and immediately made active.
6. As is usual in Webmin, you can edit an existing alias by clicking on its name in the list on the Mail Aliases page. This will bring up an editing form that contains all the same fields as the creation form, but has **Save** and **Delete** buttons at the bottom. The first of these will update the alias with any changes that you have made, while the second will permanently delete it.

If a UNIX user has a file named `.forward` in his home directory, email that would normally be delivered to his mail file will be sent to the addresses listed in the `.forward` file. In many ways, these files are equivalent to aliases that can be created by individual users instead of by the system administrator. It is even possible for a `.forward` file to contain entries that tell Sendmail to send email to a list of addresses in another file, feed it to a program as input, or append it to a file.

This module does not support the editing of `.forward` files. Usermin (covered in Chapter 47), however, does allow normal users to edit their own forwarding files using a web-based interface almost identical to the one described in this section.

37.5 Configuring Relaying

In the early days of the Internet, mail servers could safely deliver mail to local domains and forward all other email to another MTA, regardless of its source. Today, allowing your server to forward any email that it receives is an invitation for spammers to use your system as a relay. A well-configured server should only accept email for non-local domains from trusted client hosts, such as those on the company network or home LAN. Email sent to local domains is safe, and can be accepted from anywhere.

For this reason, the Sendmail packages that come with modern Linux distributions are configured by default to prevent the server accepting non-local email from anywhere except the same system. If you are setting up a mail server for a company or for your home LAN, you will need follow these steps to allow other hosts to relay mail as well. If Sendmail on your system is an open relay (one that accepts non-local email from anywhere), people sending out millions of spam email messages can use it to cover their tracks. Even if you are running a small mail server for a tiny company that you think no spammer will ever know about, it is still a very bad idea to leave your system open to relaying.

1. On the module's main page, click on the **Relay Domains** icon to bring up a form for entering relay networks and domains.
2. In the **Domains to which relaying is allowed** field, add the address of the network from which you want to allow clients to relay. It should be entered without any trailing zeros, for example, `192.168.1`. More than one network can be entered, as can specific IP addresses.
You can also enter domain names like `foo.com` to which Sendmail will allow relaying. Any received email message (no matter what its source) that is destined for a specified domain will be delivered to the appropriate server. This can be useful if your system is a mail gateway for other domains that cannot be reached directly by the rest of the Internet, as explained in Section 37.7 “Configuring Domain Routing”.
3. Click the **Save** button to activate the new relay domains list.

One side effect of Sendmail's relaying restrictions is that there is no way to use your system as a server for outgoing email when you are connecting from an untrusted network. In fact, that is the whole point. It can sometimes be annoying—if you dial into many different ISPs and don't want to reconfigure your mail client to use a different outgoing mail server for each one, for example. In an ideal world, it would be possible to use your own mail server for outgoing email no matter where you are connecting from, but this is normally impossible without turning off relay restrictions altogether.

There has been an attempt to solve this problem by adding extensions to the SMTP protocol to support authentication, so that clients who log in with a username and password are allowed to relay. Unfortunately, these extensions are not widely supported by mail clients or Sendmail yet, and there is no support in this Webmin module for configuring them.

Another solution involves trusting clients that make a POP3 connection before SMTP, which most mail client programs do. This requires cooperation between the POP3 server and Sendmail, however, which are usually unrelated programs. At the time of writing, Webmin does not support the configuration of Sendmail and a POP3 server to do this either.

37.6 Managing Virtual Address Mappings

Address mappings are similar to aliases, except that they apply only to email sent to a specific user and domain, rather than to a user at any domain as aliases do. Another difference is that address mappings can only forward email to a single address, rather than to a program, file, or list of addresses. This limitation, however, can be overcome by combining both mappings and aliases.

You can use address mappings to send email to *sales@foo.com* and *sales@bar.com* to different final destinations, even though your server hosts both domains. This is particularly useful if you manage a large number of email domains for different customers, many of whom want to have similar addresses (such as *sales*) in their domains.

Address mappings can also be used to redirect all email to a particular domain to the same users at a different domain, so that mail to *bob@foo.com* and *fred@foo.com* will be sent to *bob@bar.com* and *fred@bar.com*, respectively. Better still, you can have all email to any address at a domain sent to another single address, which is useful for the POP3 mail client Fetchmail that is explained in Chapter 33.

To create a new address mapping, follow these steps:

1. On the module's main page, click on the **Address Mappings** icon. A page listing all existing mappings will be displayed, with a form at the top labeled **Create Mapping** for adding a new one.
2. If you want to create a mapping for email to just a single address, select **Address** in the **Mail for** field, and enter the address into the adjacent text box. Unlike an alias, it must be entered in full like *fred@example.com*.

If you are creating a mapping for all email to a domain, select the **Domain** option and enter the complete domain or hostname into the text field next to it, such as *example.com*.

Either way, the domain in the address or the entered domain must appear in Sendmail's list of local domains, explained in the "Editing Local Domains" section earlier in this chapter.

3. If the destination of the mapping is a single address, select **Address** in the **Send to** field and fill in the text box next to it with either a complete address (like *jcameron@foo.com*) or a mailbox name (like just *jcameron*).

If **Domain** was selected for the **Mail for** field, you are allowed to select the **Domain** option for this field as well. If so, you must enter a domain name (like *foo.com*) into the adjacent text box, to which all email sent to the original domain will be forwarded.

The final **Return error** option in the **Send to** field can be selected if you want a specific error message to be returned to senders. If chosen, an error type must be selected from the menu next to it, and a more detailed error message entered into the text box. This option can be useful for sending back bounce messages that explain why a particular address or entire domain is no longer reachable.

4. Click the **Create** button to add the address mapping to the list of those shown below the form. It will be made immediately active.

As with aliases, you can edit or delete existing mappings by clicking on their addresses in the list. This will bring up a form identical to the one used for creating a mapping, but with **Save** and **Delete** buttons at the bottom.

If a mapping exists for both a domain and an address in that domain, Sendmail will use the second for email to that specific address, and the first for email to any other mailbox in the domain. In effect, more specific address mappings take precedence over those that are more general. Their ordering in the list on the Address Mappings page does not matter at all.

If you want to create a mapping that forwards email to a program, file, or list of addresses, you will need to create an alias as well. The address mapping will send email to the alias, and then the alias will forward it on to the real destination. The alias should assign a name that is related to the address mapping, such as *jcameron-example-com* for mapping for the virtual address *jcameron@webmin.com*.

On a system with many domains and users, it is quite likely that two people will want to have the same mailbox name in different domains, such as *bob@foo.com* and *bob@bar.com*. Because Sendmail ultimately only delivers email to UNIX users' mail files, and two UNIX users cannot have the same name, this can be a problem. The usual solution is to create users named like *bob-foo* and *bob-bar*, and set up appropriate address mappings to forward email to them. The only downside is that the users will need to log in to the POP3 server as *bob-foo* or *bob-bar* instead of just *bob*.

37.7 Configuring Domain Routing

Sendmail's domain routing feature can be used to forward all email to a particular domain to a different server. It is most useful if the DNS is set up to send email for some domain to your system, which should then be forwarded to another MTA that is unreachable by the rest of the Internet. Routing can also be used to override the normal method by which Sendmail works out which host to send email to, which can be handy on networks in which connectivity is incomplete or some DNS information is not available to all hosts.

To add a new domain routing rule, follow these steps:

1. On the main page of the module, click on the **Domain Routing** icon. A page listing existing routings (if any) will be displayed, above which is a form for adding a new one.
2. To have email to just a specific domain or host routed elsewhere, select the **Host** option in the **Mail for** field and enter the domain or hostname into the adjacent field.

Alternately, if you want email for all hosts within a domain to be routed, select **Domain** instead and enter the domain name into its field. A routing of this kind for the domain *example.com* will not affect email sent to an address in the domain (like *jcameron@example.com*), but only email to addresses on hosts under the domain (like *jcameron@foo.example.com*). Normally, this is not what you want.

Any domain or hostname that you enter must be on the list for which Sendmail allows relaying, as explained in Section 37.5 "Configuring Relaying".

3. From the **Delivery** menu, select **SMTP**. This field tells Sendmail which protocol to use when routing email for the domain. Most of the other options are useless, as they relate to UUCP, which is hardly used anymore.

4. In the **Send to** field, enter the hostname of the system to which mail should be forwarded. The **Ignore MX for SMTP delivery** box should be checked as well, so that Sendmail always delivers directly to this host instead of trying another DNS look up to determine the correct destination.
5. Hit the **Create** button to add and activate the new domain routing. You should test it to make sure it is really working, as small mistakes (such as selecting **Domain** instead of **Host**) can prevent a route from working.

As on other similar pages in the module, an existing routing rule can be edited or deleted by clicking on its domain name in the list on the Domain Routing page. There is no way to temporarily disable a rule, however, as there is with aliases.

Sendmail can also be configured to forward all non-local email to a specific server, rather than just email to particular domains. This is useful if your company or ISP has a central mail server to which you want to hand off email, rather than having your system connect to the real destination server. The next section explains how to set this up.

37.8 Editing Global Sendmail Options

The global options control such things as the maximum message size, number of days to retry email, load average limits, and the outgoing mail server. The following steps explain how to edit some of the most useful ones:

1. On the module's main page, click on the **Sendmail Options** icon. A form for editing global options (shown in Figure 37.3) will be displayed.
2. When the **Send outgoing mail via host** field is set to **Deliver directly**, Sendmail will look up the correct mail server for non-local messages and connect to it to deliver email. If you select the second option and enter a hostname into the text box, however, all messages except those for local users will be sent to that host instead. This will typically be a central mail server at your ISP or on your company's network.

If a domain routing rule exists for an address, it will take precedence over any server set using in this field.

3. The **Delivery mode** field controls how Sendmail processes incoming messages. The available options are:

Background or Default In this mode, email is immediately accepted from clients and then delivered to the destination by a separate background process. This is usually what you want.

Queue only or Deferred When one of these modes is chosen, Sendmail will simply add incoming messages to its mail queue. Only when the queue is explicitly flushed will they be sent to their destinations. This can be useful if your system is not always connected to the Internet, or if you want explicit control over when the server sends out messages. When a message is received in either of these modes, Sendmail performs no validation of the source or destination address, which would otherwise cause problems if your system is disconnected from the network.

Interactive This mode is similar to **Background** except that email is delivered by the same process that accepts it from the client. This means that clients must wait until Sendmail has

transferred their messages to the next or destination server, which may take some time. It does, however, cut down the number of processes that Sendmail needs to start.

4. The **SMTP port options** field can be used to set the TCP ports that Sendmail listens on for SMTP connections. The actual value that can be entered can be quite complex, but if you want your system to accept connections from anywhere on the standard SMTP port, you should enter just `Name=MTA`. On some operating systems, this is not the default and Sendmail will only accept connections from `localhost`.
5. The **Max load average for sending** field can be used to set a load average above which Sendmail will not send out messages. The load average is a rough estimate of the number of processes running on your system at a moment in time, as explained in Chapter 11. This option and the next are useful for limiting the amount of CPU time that Sendmail can use up on your system, although the latter is more useful.

If **Default** is selected, the server will continue to process the mail queue and send messages no matter what the load average is.
6. To set a load average above which Sendmail will no longer accept new messages, enter a value in the **Max load average for receiving** field. When this level is reached, the server will close the SMTP port until the load drops back below it again. Typically, whatever you enter should be lower than the limit set for the **Max load average for sending** field, so that the mail queue is still processed even when the load is high.

When **Default** is selected, Sendmail will accept new connections regardless of the load average.
7. The **Time before giving up** field specifies the amount of time that Sendmail will attempt to send an email to an uncontactable remote server for before returning a bounce message to the sender. The value you enter must be a number followed by `d` (for days) or `h` (for hours). It should only be changed if your system is likely to be disconnected from the Internet for longer than the default of 5 days and you don't want messages in the queue to bounce.
8. The similar **Time before sending warning** field specifies the time that Sendmail will hold a message in the queue before sending a warning to the original sender. If your system is a secondary mail server for some domain or is often disconnected from the network, it should be increased to the same time as the previous field.
9. To change the address to which Sendmail sends fatal or double-bounced messages, enter a new address in the **Send error messages to** field. When **Postmaster** is selected, they will be sent to the Postmaster mailbox instead, which is usually an alias for `root`.
10. To limit the amount of disk space that Sendmail will use up for queued messages, fill in the **Min free disk space** field. If the number of free blocks drops below this level, new incoming messages will no longer be accepted. The exact size of a block depends on the type of filesystem in use, but they are typically 1 kB or 512 bytes in size.
11. To stop large messages being sent via your mail server, fill in the **Max message size** field. Any email larger than the number of bytes entered will be rejected when it is received. If you have a slow network connection and untrusted client users, this option can be useful for saving bandwidth.
12. Finally, click **Save and Apply** to save the new global options. Webmin will automatically restart Sendmail for you to activate them.

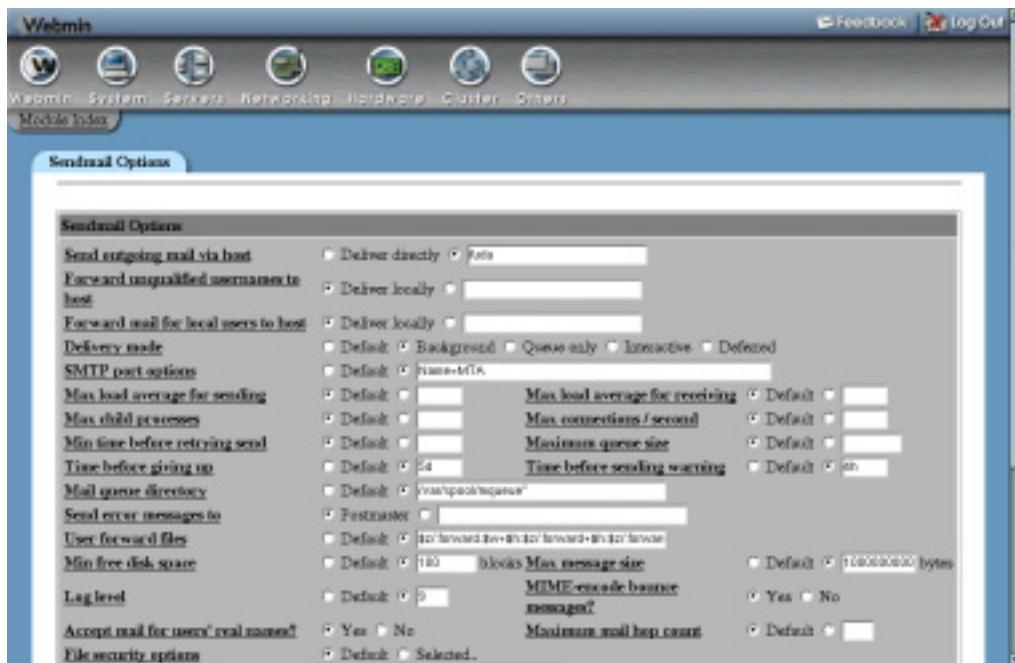


Figure 37.3 The Sendmail options page.

37.9 Viewing the Mail Queue

When Sendmail receives a message, it is placed into the mail queue. If it can be sent to its destination immediately, then it will be removed from the queue almost at once. If, however, some temporary error occurs when sending, then it will remain queued for later processing. The Sendmail server process makes periodic checks of messages in the queue, retrying each one at longer and longer intervals. Finally, after a message has been in the queue for too long (usually 5 days), it will be removed and a bounce email will be sent back to the original sender.

Most messages that are in the queue for a long time are there because the destination mail server is down or unreachable. Another common cause is a temporary error reported by the remote MTA, such as a lack of disk space. Webmin allows you to view messages in the queue and even delete them by following these steps:

1. On the module's main page, click on the **Mail Queue** icon to go to a page listing the details of queued messages. The number of emails in the queue is displayed below the icon so you can see how long it is at a glance.
2. The ID, sender, destination, subject, and size of all queued messages are displayed in a table on the mail queue page. In the final column is the current status, which indicates what Sendmail is trying to do with the message at the moment. **Sending** will appear when Sendmail is trying to connect to the remote server, and **Deferred** will be shown—along with a reason—when a connection has been tried and failed.

If the queue contains more than 20 messages, only the first 20 will be displayed. To page through the rest, use the left and right arrow buttons that appear above the list.

3. To view the actual contents of an email, click on its ID in the queue listing. All headers, the text body, and any attachments will be displayed. To view an attachment, just click on its icon. To remove just this message from the queue, hit the **Delete** button at the bottom of the page.

If the email is locked because it is currently being sent, Webmin will display an error message along with a button labeled **Force deletion anyway**, that you can click to override the lock. This may cause the message to be only partially sent, however, and so is not recommended.

4. To remove multiple messages from the queue, first select them using checkboxes next to their IDs and the **Select all** and **Invert selection** links on the queue list page. Then, click the **Delete selected messages** button to get rid of those that you have chosen. To override any locks on the selected emails, check the **Even if locked** box first.

After you hit the **Delete** button, a page listing the ID and deletion result of each chosen message will be displayed. Deletion can fail if the message is no longer in the queue, or if it is currently locked.

Even though Sendmail will automatically retry messages in the queue, you can force it to attempt delivery of all queued messages immediately using Webmin. This can be useful if you have a dial-up Internet connection and queue up several emails while you are disconnected. The steps for flushing the queue are:

1. On the module's main page, click on the **Mail Queue** icon to bring a list of queued messages.
2. As long as the queue is not empty, a button labeled **Flush Mail Queue** will be visible at the bottom of the page. Click it to begin immediate processing of all waiting messages.
3. A page showing the output from Sendmail as it attempts to deliver queued email will be displayed. If you have a large queue containing messages for down remote servers, it may take a long time to completely appear.

37.10 Reading Users' Email

As the introduction explains, Sendmail stores messages received by users in files in the `/var/mail` or `/var/spool/mail` directory. These are read and emptied by the POP3 server, command-line mail clients like `pine` or `elm`, or web-based mail clients like Usermin. This Webmin module, however, can also act as a simple mail client, allowing you—the system administrator—to read any user's email.

This feature is useful for deleting large messages in user mailboxes that would otherwise take a long time to download over a dialup POP3 connection. It also allows you to read email for system users such as `root` without needing to telnet in or run a separate mail client. More controversially, you can even invade people's privacy on a multi-user system by reading their personal email—assuming they have not downloaded and deleted it via POP3 yet.

The following steps show you how to check the contents of a user's mailbox:

1. On the module's main page, click on the **User Mailboxes** icon. A page listing all of the users on your system and the sizes of their mailboxes will be displayed, unless you have more than 200 users. In that case, a small form for entering a username will appear instead.
2. To view an actual message, click on the sender's name in the **From** column. A page showing the important headers, body text, and attachments will appear. Click on an attachment icon to view it, assuming that the data type is supported by your browser or some external program.

To remove just this email from the user's mailbox, click the **Delete** button at the bottom of the page. This can take quite some time if the mailbox is extremely large (over 10 MB) or contains lots of messages, as Webmin needs to rewrite the entire mail file. Click on the name of a user to bring up a list of messages in his mailbox, an example of which is shown in Figure 37.4. By default, the most recent messages are shown first, even though they are actually at the end of the mail file.

If the mailbox contains more than 20 emails, only the first 20 will be displayed. To page through the rest, use the left and right arrow buttons located above the list.

3. To delete multiple messages, first select them using the checkboxes and **Select all** and **Invert selection** links on the mail list page. Then, click the **Delete** button above or below the list. Once again, this can take awhile for large mailboxes.
4. To search the user's mailbox for messages matching some criteria, use the **Find messages where** form below the list. The following types of search can be selected from the menu:

From: matches, Subject: matches, To: matches, or Cc: matches Finds messages in which the From, Subject, To, or Cc field contains the text entered into the adjacent text box. The comparison is case insensitive, but regular expression characters cannot be used.

Date: matches Finds messages in which the sending date header contains the entered text. This header will not be converted to local format, so whatever you enter must match the date format used by the sender.

Body matches Finds messages whose body contains the entered text. The body includes all attachments in their unencoded form, not just the text that is shown when you read an email.

Size is greater than Finds messages whose total size is greater than the number of bytes entered into the adjacent field.

For each of the above search types, an inverse type is also available, such as **From: doesn't match** or **Size is less than**. After choosing your search type and entering the text to match, hit the **Search** button. A page listing all matching messages will be displayed, from which you can view the contents of emails or select some or all to delete, just like in the normal mail list.

The mail reading section of the module actually allows you to compose new messages and reply to or forward existing ones. In fact, it can be used as a full-featured email client, although it is not the best program for the job. The default **From** address for sent messages is determined from the mailbox user's name and the system hostname, but this can be changed on the module access

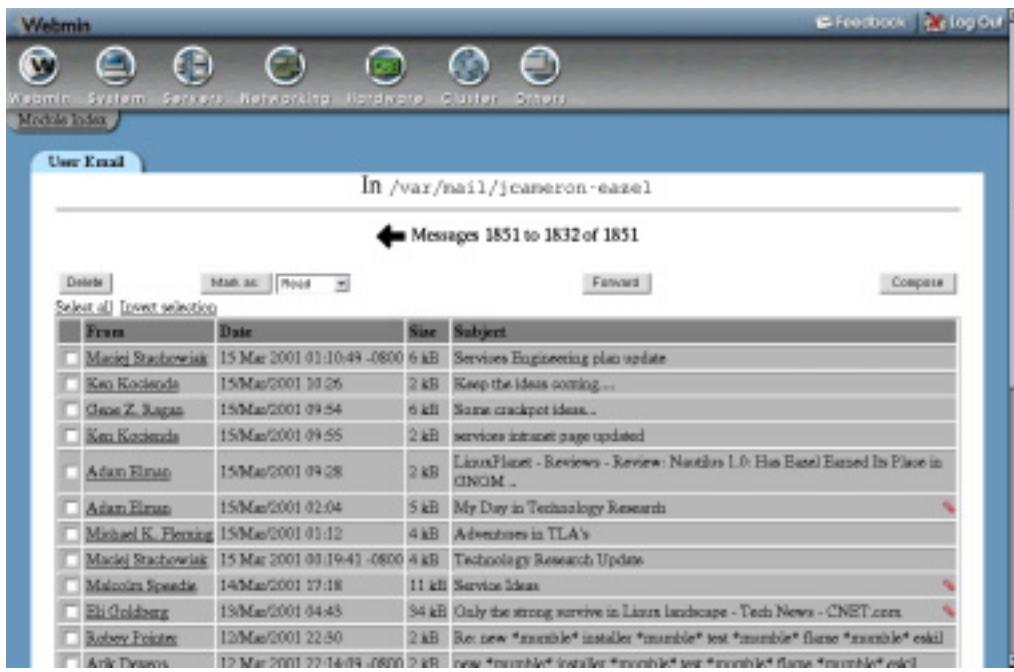


Figure 37.4 The contents of a user's mailbox.

control page on a per Webmin-user basis. It is even possible to create Webmin users who can use this module to read only their own mailbox and send email from only their address. You should really use a program like Usermin, however, if you want to give the same mail-reading web interface to a large number of users on your system.

By default, the module assumes that the mail for each user will be stored in a file with the user's name in the `/var/mail` or `/var/spool/mail` directory. It is possible to configure Sendmail to use a different file path instead, such as the `mbox` file in each user's home directory. If this is the case on your system, you will need to reconfigure the module, as explained in Section 37.15 “Configuring the Sendmail Configuration Module”. Otherwise, all the mailboxes will show up empty because Webmin is looking in the wrong place for them.

37.11 Adding Sendmail Features with M4

The primary Sendmail configuration file `sendmail.cf` is extremely complex, and almost impossible to edit manually. Fortunately, it is usually built from a series of M4 macro files that are much simpler and can be modified using this module or by manual editing. M4, for those who have not heard of it before, is a program that parses text files and expands macros in them. These macros can include other text files, define variables and functions, or exclude text based on certain conditions. M4 is very similar to the preprocessor used by the C programming language which handles `#include` and `#define` statements. Fortunately, this module hides most of the complex details from you.

Often, the default Sendmail configuration that comes with your operating system will not have some features enabled, such as address mapping or domain routing. Webmin can detect this, and will display an error message if you try to use a module page for a Sendmail feature that is not enabled in the primary configuration file. To turn on a missing feature, an entry must be added to the primary M4 configuration file, from which `sendmail.cf` is rebuilt. Sendmail will then be able to use it and this module will be able to configure it.

Before you can manage your Sendmail M4 configuration, this module must know where to find the M4 files. To check if it has the correct paths and to set them if not, follow these steps:

1. On the module's main page, click on the **Sendmail M4 Configuration** icon.
2. If an error message like **The Sendmail M4 configuration file /etc/sendmail.mc was not found** or **The locations of the Sendmail M4 base directory and M4 config files have not been set** is displayed, then the module cannot find the M4 files. If a table of entries from the file is displayed instead, then everything is configured OK and you can skip the rest of these steps.
3. If you are running Linux, check your distribution CD or website to make sure that all the packages whose names start with `sendmail` are installed. Sometimes the M4 files are in a separate package named something like `sendmail-cf`. If you do find and install such a package, go back to Step 1 and check again to see if the module has found the configuration files.
4. If you have compiled and installed Sendmail from the source package, the M4 files will be in the `cf` subdirectory of the source code. Click on the **Module Config** link on the module's main page, and set the **Sendmail M4 base directory** field to `/path/to/source/cf`, and the **Full path to M4 config file** to the appropriate `generic-` in the `/path/to/source/cf/cf` directory. For the latter field, the file you choose depends on your operating system—for example, on Linux you would enter `generic-linux.mc`, while on Solaris you should enter `generic-solaris2.mc`. The `.cf` files in that directory should not be selected, as they are fully built Sendmail primary configurations, not M4 macro files. See Section 37.15 “Configuring the Sendmail Configuration Module” for more information on these fields.
5. If you are using the Sendmail package that came with your version of UNIX and cannot find any M4 files on your system, then this feature of the module cannot be used. This is unfortunately the case on some operating systems.

Once the Sendmail M4 Configuration page displays the contents of your primary M4 file, you can use it to add new features such as address mapping or domain routing. The page can in fact be used to modify any of the M4 macros in the file, but unless you are any experienced Sendmail administrator it is best to stick to these instructions for adding features:

1. From the menu next to the **Add new entry of type** button at the bottom of the page, select **Feature** and then hit the button to go to the feature creation form.
2. Select the one that you want to add from the **Feature** menu. The most commonly used features have names next to them in brackets that correspond to their icons on the module's main page.
3. In the **Parameters** field, enter hash `-o` followed by the path that should be used for the feature's text and DBM files. For example, if adding a `virtusertable` feature you

should enter `hash -o /etc/mail/virtusertable` (assuming your system has an `/etc/mail` directory).

4. Log in to your system as `root` via SSH, telnet, or at the console and create the empty text file with a command like `touch /etc/mail/virtusertable`. Naturally, this is not necessary if it already exists.
5. Click the **Create** button to update the M4 file and return to the previous page.
6. At the very bottom of the M4 file contents list, click the **Rebuild Sendmail Configuration** button. A confirmation page showing the exact changes that will be made to your `sendmail.cf` file will be displayed. Typically they will be limited to directives for the new feature. If the confirmation form indicates that a huge number of lines are going to be changed, it is likely that the M4 file being edited was not originally used to build your current `sendmail.cf` file, and so should not be used in future.
7. To go ahead and use the newly rebuilt Sendmail configuration, click the **Yes, replace it now** button. Your `sendmail.cf` file will be updated and the server process restarted to immediately activate it.
8. Go back to the module's main page, and click on the icon for the feature that you have just enabled. You will not be able to add and edit address mappings or whatever it was for which you added support.

As you can see from looking at the existing M4 file entries, editing or adding to it can still be quite complex. For this reason, the book does not cover all of the possible features or other macro types that you can add. For most people, however, the defaults will work just fine. At most, all you should need to do is add a feature or two as explained above. If you want to learn more about editing the M4 configuration, you should buy a book dedicated to Sendmail administration.

One problem with using the M4 page to rebuild your Sendmail configuration is that any changes that have been made directly to `sendmail.cf` will be overwritten. The module's Sendmail Options page unfortunately does precisely this, so if you follow the instructions above to add a new feature, any changes made on that page will be lost! The only real solution is to edit the entries in the M4 file that correspond to those on the Sendmail Options page. For example, the **Send outgoing mail via host** field is set by the `SMART_HOST` define.

37.12 Creating Autoreply Aliases

The Sendmail Configuration module lets you easily create an alias that triggers an automatic reply to anyone who sends email to it. When you do this, Webmin creates a simple script that is run from the alias and receives the contents of email sent to it as input, just like a command specified using the **Feed to program** alias type.

To set up an autoreply alias, follow these steps:

1. On the module's main page, click on the **Mail Aliases** icon.
2. In the **Create Alias** form, enter a name for the alias, and select **Autoreply from file** from the type menu. The name can be that of an existing UNIX user if you want to set up an automatic reply to any message sent to him—for example, if he is on vacation.
3. In the field next to the type menu, enter the path to a file that will contain the autoreply message, such as `/home/someuser/autoreply.txt`. The file does not have to exist yet.
4. Click the **Create** button to add the alias, then click on its name in the list to edit again.

5. Follow the **Edit** link next to the autoreply filename field. This will bring up a page containing a large text box for entering the contents of the reply message.
6. After entering the text that you want sent back to any sender, click **Save** at the bottom of the page. The autoreply alias is now fully active.

The reply text can contain several special macros that start with \$, such as \$SUBJECT, \$TO, \$FROM, \$DATE, and \$BODY. When the reply is sent, these will be replaced with the original message's subject, destination address, sender address, sending date, or body, respectively. Be careful using the \$BODY macro though, as it will be replaced with the entire unencoded contents of the email to which you are replying.

You can also add to, or override, the headers used in the reply message by starting the autoreply text with one or more lines in standard SMTP header format, followed by a blank line. For example, to set the subject of the automatic reply you could enter *Subject: This is an automatic reply* at the top of the text box, with an empty line after it.

One problem with Webmin's autoreply script is its inability to reliably determine the **From:** address to use when sending the reply. This is normally just taken from the **To:** address of the original message, but this is not possible when replying to a message that was sent to multiple people. Even though the code attempts to find the right address automatically, it can sometimes get the wrong one and send an automatic reply that appears to be from the wrong person. For this reason, you should set the correct address by including a header line like *From: Jamie Cameron <jcameron@example.com>* at the top of your reply text.

If you are setting up an automatic reply alias for a UNIX user, it is usually a good idea to have a copy of all email sent to the user stored in his mailbox as well. To do this, re-edit the alias and select **Email address** from the second type menu. Then enter the user's name, preceded by a backslash, into the text field next to it (like `\jcameron`) and hit **Save**.

37.13 Creating Filter Aliases

Sendmail aliases normally forward email messages to their destinations regardless of their content. It is possible, however, to use this Webmin module to create an alias that forwards to different addresses or files depending on the headers or body of a message sent to it. When you create an alias like this, the module internally creates a script that passes the contents of email to the alias as input, just like a command specified using the **Feed to program** alias type.

A filter consists of a series of rules, each of which has a condition and action. The condition specifies both a header and text to check if it is contained in that header, while the action specifies an address to which to forward or a file to which to append the message. When a message is received by the filter, it is checked against the rules, in order, until one that matches is found and its action is performed. At the end of the list is a default action, which determines where to forward email that does not match any of the conditions.

Compared to other mail filtering or classification programs like Procmail and SpamAssassin, Webmin's filters are limited in their functionality and flexibility. They are easy to create, however, and require no additional software. To create one, the steps to follow are:

1. On the module's main page, click on the **Mail Aliases** icon.
2. In the **Create Alias** form, enter a name for the alias and select **Apply filter file** from the type menu. The alias name can be that of an existing UNIX user if you want to filter his email before delivery.

3. In the field next to the type menu, enter the path to a file that will contain the filter rules, such as `/home/someuser/filter.rules`. Because the file format is fairly complex, you should not enter the name of any existing file unless it was created by following these same steps.
4. Click the **Create** button to add the alias, and then click on its name in the list to edit again.
5. Follow the **Edit** link next to the filter rules file field. This will take you to the form shown in Figure 37.5, which contains a table of empty rows for entering new rules and a text field for the default action address.

6. In the first row, select the header on which you want to filter from the menu next to **If the**. To check the entire undecoded email body, select **body** from the list instead.

To have the action performed if some text is found in the header or body, select **matches** from the second menu. To perform the action only when some text is not found, choose **doesn't match**, instead.

In the field next to this menu, enter the text for which the filter should check the header or body. The comparison will always be case insensitive, and Perl regular expression characters like `[`, `,`, `+`, and `*` can be used.

In the **then forward to** field, enter the email address or mailbox name to which messages matching the condition that you specified will be sent. It is also possible to enter an absolute file path like `/home/someuser/mail/somefolder` to which email will be appended instead. This could be another user's mail file, or a mail folder—however, it must be writable by the UNIX user daemon as whom Sendmail runs.

7. To enter another rule, fill in the second blank row by repeating the instructions in Step 6 again. Only five empty rows are displayed, but you can create more than five rules by saving and re-editing the filter file.
8. In the **Otherwise forward to** field at the bottom of the table, enter an address or filename to which messages that do not match any of the rules will be sent or appended. If this is left blank, unmatched email will be thrown away!
9. Finally, click the **Save** button to activate the new filter rules. To make sure they are really working, you should send a few text messages to the alias and check to see that they are delivered correctly.

To have email forwarded to multiple addresses by a single rule or the default action, just enter them separated by commas into one of the **forward to** fields. You can also enter the name of another alias as the destination, which can then send messages to multiple files, addresses, and/or programs.

When creating a filter alias with the same name as a UNIX user, it is often useful to specify that messages matching some rule be delivered to the user's normal mailbox. Just entering the user's name as the destination, however, would be a big mistake as this would trigger an endless loop through the filter! Instead, you should prepend a backslash to the username—like `\jcameron`. As with aliases, this tells Sendmail to ignore any aliases for this mailbox.

Because it can often be difficult to work out what a filter alias is doing and why it is doing it, the filter script creates a log file in the same directory as the rules file. The log has the same name as the filter file, but with `.log` appended. Every message received by the alias and delivered to some destination by the filter causes a line to be added to the log, containing the date, time, sender, destination, and rule matched. The log file will only be created if the directory or log itself is writable by the daemon user, though.

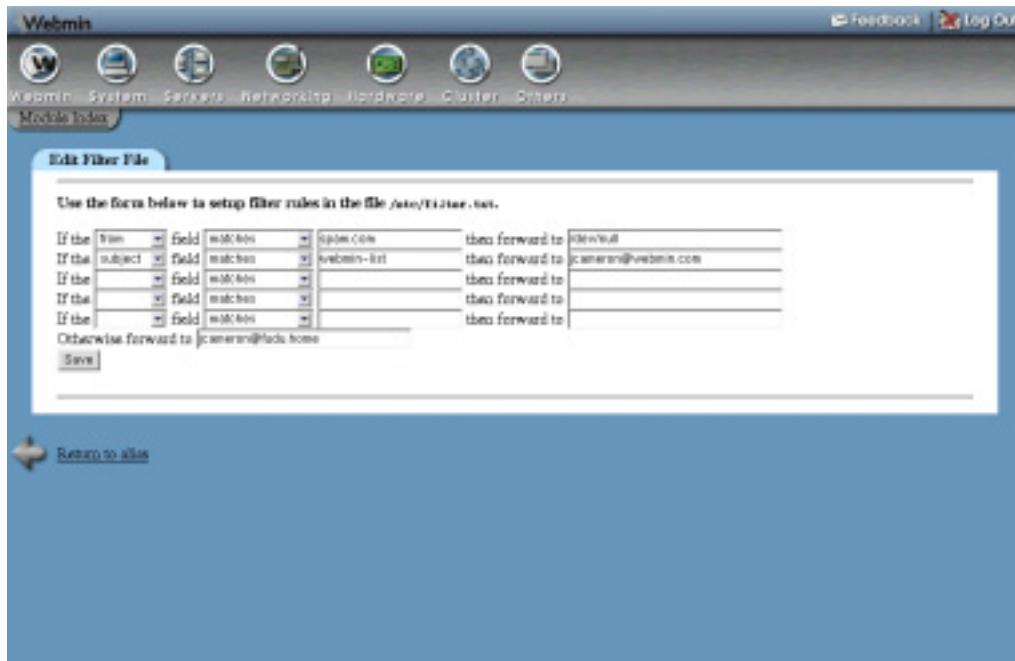


Figure 37.5 The filter creation form.

37.14 Sendmail Module Access Control

The Sendmail Configuration module probably has the most powerful access control features of any module in Webmin. You can use them to limit the aliases and virtual addresses a Webmin user can edit, or restrict him to reading only the mailboxes of certain UNIX users. These features are most useful in a virtual hosting environment, where customers own email domains and the user accounts. On this kind of system, you can create one Webmin user per customer who can only manage the address mappings, aliases, and mailboxes for his own domains, while not being able to use other features of the module or touch other customers' information.

Once you have created a Webmin user who has access to the module (as explained in Chapter 52), follow these steps to restrict what he can do:

1. In the Webmin Users module, click on Sendmail Configuration next to the name of the user whom you want to restrict.
2. Change the **Can edit module configuration?** field to **No**, so that he cannot modify paths to Sendmail programs and files.
3. Set all of the **Yes/No** fields in the second section to **No**, which will prevent the user from seeing most of the module's icons.
4. Select **No** from the **Can manage mail queue?** menu, or **View only** if you just want the module user to be able to see the contents of the queue. Selecting **Yes** would be a bad idea, as it would allow him to delete queued email belonging to other domains.
5. For the **Address mappings this user can edit** field, select the **Matching** option and enter a Perl regular expression for allowable mapping sources into the text field next

to it. For example, to let him create and edit mappings in the domains *foo.com* and *example.com*, you should enter (@*foo.com*/@*bar.com*)\$.

6. It is safe to select all of the checkboxes in the **Address mapping types this user can edit** field.
7. In the **Aliases this user can edit** field, select **Matching** and enter a regular expression that only lets him modify or create aliases starting with the customer's domain names. For example, if the user owns the domains *foo.com* and *example.com* you should enter ^(*foo/example*)- to limit him to aliases like *foo-jcameron* or *example-fred*. This naming convention ensures that users cannot step on each others' aliases. To limit the number of mappings that the user can create, select the second radio button in the **Maximum number of address mappings** field and enter a number into the box next to it. This can be useful for preventing a single customer from more address mappings than he has paid for.
8. In the **Alias types this user can edit** field, deselect the checkboxes for types of aliases that the Webmin user should not be allowed to create. Good candidates to deny access to are **Write to file**, **Feed to program**, **Autoreply**, and **Filter file**, as they use the permissions of the Sendmail daemon user and thus may be a security risk.
9. To limit the number of aliases that the customer can create, select the second radio button in the **Maximum number of aliases** field and enter the maximum into the box next to it.
10. To stop the Webmin user from creating aliases that run programs, append to files, or use address files outside a certain directory, enter it into the **Limit files and program to directory** field. Unfortunately, this can be subverted by the clever use of symbolic links and so is not a very strong security measure.
11. In the **Outgoing addresses this user can edit** field, select **Matching** and enter the same regular expression as in the **Address mappings this user can edit** field. This will limit the user to rewriting addresses for only his own domains. To prevent the editing of outgoing addresses at all, select **None**. In most cases, there is no need for a Sendmail administrator to edit them anyway.
12. In the **Users whose mail can be read** field, select one of the last five options to limit the customer to only those UNIX users who belong to him. If he has been given limited access to the Users and Groups module as well, then you should allow him to read the email of the same users that he can create and edit in that module.
13. Leave the rest of the fields on the form set to their defaults. They are only really useful if you are setting up the module as a web-based mail reading interface. Although this is possible, there are much better alternatives such as Usermin (covered in Chapter 47).
14. Click the **Save** button to make the restrictions for the user active.

Even though it is possible to configure this module to limit a user to certain domains, the module's interface is not particularly friendly compared to products like Plesk or Cpanel. These are web-based virtual server management interfaces that have been designed from the ground up for that purpose, unlike Webmin which was designed to allow the management of everything on a system.

37.15 Configuring the Sendmail Configuration Module

The **Module Config** link on the main page takes you to a form seen in many other modules for editing settings that apply to the operation of the module itself. Those listed under **Configurable**

options relate to its user interface, while those under **System configuration** define the paths to the Sendmail programs and files.

Settings in the latter group do not usually need to be changed. By default they are set to match the Sendmail package supplied with your operating system. If you have compiled and installed the MTA yourself from the source code, however, then it is quite possible that they will be incorrect.

The available module configuration options are shown in Table 37.1.

Table 37.1 Module Configuration Options

Seconds to wait before refreshing mail queue	When Don't refresh is selected, the mail queue display will only be refreshed when you click the reload button in your browser. On the other hand, if the second option is selected and a number entered, the browser will automatically reload the page after that number of seconds has elapsed. This can be useful if you want to continually monitor the mail queue as it changes.
Mail messages to display per page	This field controls the number of email messages that appear per page in the users' mailboxes and when viewing the mail queue. You may want to increase it if you like to use a large browser window.
Width to wrap mail messages at	When reading a message in a mailbox or in the queue, Webmin will automatically wrap the body text so that each line is less than the number of characters specified in this field. You may want to change it depending on the width of your browser window.
Sort tables by	When Order in file is chosen, the lists of aliases, address mappings, routed domains, outgoing addresses, domain mappings, and spam control rules will be displayed in the order in which they appear in their source text files, which is typically the order in which they were added. If Name is chosen, however, they will be sorted by name instead, which can make individual entries much easier to find.
Send mail via connection to	As Section 37.10 “Reading Users’ Email” explains, this module can be used to compose and send email messages. When this field is set to Sendmail executable , Webmin will send these messages by running the Sendmail command with the -t option—usually <code>/usr/lib/sendmail</code> . Assuming your mail server is running properly, this should work fine. To have messages sent via an SMTP connection to some other MTA instead, select the second option for this field and enter another mail server’s hostname into the text box. You can even enter <code>localhost</code> to have email sent via the Sendmail server on this system, but using SMTP instead of running a program.
When reading mail start at	When Latest is selected, user mailboxes will be sorted so that the newest email appears first. If the Oldest option is chosen, the oldest email will appear at the top of the list instead.

Table 37.1 Module Configuration Options (Continued)

Wrapping mode in mail textarea	This option sets the wrapping mode used for bv HTML <code>textarea</code> on the mail composition page. If Off is chosen, no wrapping is done and long lines will cause the text box to scroll to the right. If any of the other options are selected, long lines will be wrapped as you entered them (but not in the actual email!).
Keep track of read/unread emails	When this field is set to Yes , the module will keep track of which messages in user mailboxes have been read and show ticks next to those that have. It will also display additional buttons for marking single emails or multiple messages as read or unread. This feature is really only useful if you are using the module as an actual mail client, which is why the default for this field is No .
Show To: address in mailboxes?	When Yes is selected, the display of messages in a user's mailbox will include the To address as well as the From, Subject, and Date headers. This can be useful if mail to several different addresses ends up in the same mailbox.
Maximum number of records to show in tables	If the number of entries in the list of aliases, address mappings, routed domains, outgoing addresses, domain mappings, or spam control rules exceeds the number in this field, then they will not be shown. Instead, a search form will appear allowing you to find entries by name. This is done to prevent an extremely large alias file, making the Mail Aliases page enormous and slow to display. On most small systems the default limit of 200 will never be hit.
Show buttons at top for	On the pages for listing a user's email and reading a single message, buttons appear at the bottom for deleting, replying, and so on. This field controls when those same buttons appear at the top as well, so that you do not have to scroll all the way down to the top. Both the mailbox list and single message display pages have buttons at the top. Mailboxes only Only the mailbox list has buttons above the table as well as at the bottom. Never Neither page has buttons at the top, only at the bottom.
Headers to show in mail queue	The boxes checked for this field determine which columns are shown on the mail queue listing. The first five correspond to actual email headers, while the Size box enables the display of message sizes and the Status box enables the display of the current status or last error.
Sort mail queue by	This field controls the order in which messages in Sendmail's mail queue are shown. The default is to sort by queue ID, but you can have the module sort by other attributes of each message instead—but at the cost of slowing down the display for a large queue.

Table 37.1 Module Configuration Options (Continued)

Show size of mail queue on main page?	This field determines if the current size of the mail queue is shown on the module's main page below the mail queue icon. You may want to disable this if your system consistently has a large queue that Webmin is slow to count.
Minimum mail file size to index	Normally, Webmin creates an index file of messages in a user's mailbox so that it does not have to reread the mail file on every page. This speeds up the display of the mailbox and single email pages, but can break down if the mailbox changes often. A common symptom of index failure is the display of numerous blank messages instead of the actual contents of a mailbox. This field can be used to turn off indexing altogether, or limit it to only large mail files. The default of 1 MB stops the indexing of small mailboxes that are frequently cleared by POP3, while enabling it for large boxes that are left off the server.
Forward messages with quoting?	When this option is enabled (as it is by default), single messages forwarded using the module's mail reading feature will have > characters prepended to each line. Since some people don't like this form of quoting, this configuration field can be used to turn it off.
Confirm before deleting messages?	This field determines if Webmin will ask you for confirmation before deleting messages in a user's mailbox.
Full path to sendmail.cf	This field must contain the full path to the primary Sendmail configuration file, <code>sendmail.cf</code> . It can almost always be found in the <code>/etc</code> or <code>/etc/mail</code> directories.
Sendmail M4 base directory	If you want to use the module's M4 reconfiguration page (covered in Section 37.11 “Adding Sendmail Features with M4”), this field must be set to the base directory under which all the <code>.m4</code> file subdirectories can be found. These subdirectories are named <code>cf</code> , <code>domain</code> , <code>feature</code> , <code>mailer</code> , and <code>ostype</code> among others. If you have compiled Sendmail from the source code, the M4 configuration files will be included with the source and you should set this field to point to their base directory. It is important to use the files that come with the version of Sendmail you actually have installed, so that the primary configuration file can be generated properly.

Table 37.1 Module Configuration Options (Continued)

Full path to M4 config file	To use the module's M4 reconfiguration page, this field must be set to the primary M4 configuration file that specifies which features to enable. If you have compiled Sendmail from the source code, the primary file will be one of the <code>generic-* .mc</code> files in the <code>cf</code> directory under the M4 base directory. The exact one depends on your operating system—for example, <code>generic-linux.mc</code> should be used if you are running Linux. Of course, if you have already manually created your own primary M4 configuration file and generated a <code>sendmail.cf</code> file from it, then its path must be entered here instead.
Full path to sendmail pid file	This field must contain the full path to the file into which the Sendmail server process writes its process ID, such as <code>/var/run/sendmail.pid</code> . If it is not set correctly, the module will not be able to tell that the server is running.
Command to start sendmail in server mode	This field must contain a command to start the Sendmail server as a background process, which is run when the Start Sendmail button on the module's main page is clicked. If you have compiled the program yourself, you should set it to <code>/usr/lib/sendmail -bd</code> , assuming that the <code>sendmail</code> executable is in the <code>/usr/lib</code> directory. On many operating systems, this field will be set by default to use a bootup script like <code>/etc/init.d/sendmail start</code> , which is included with the Sendmail package for the OS. This is unlikely to work if you have compiled the server yourself instead of using the package.
Command to stop Sendmail	When Kill process is selected for this field, the module will simply kill the Sendmail server process when the Stop Sendmail button on the main page is clicked. If a command is specified (such as <code>/etc/init.d/sendmail stop</code>) it will be run instead. If you have compiled and installed manually, you should select the Kill process option as no bootup script to stop the server is likely to exist.
Makemap command	This field must contain the path to the <code>makemap</code> command used for rebuilding DBM files from text files. It is acceptable to enter just <code>makemap</code> if the program is in one of Webmin's search path directories, such as <code>/usr/bin</code> or <code>/usr/local/bin</code> .
Sendmail command	This configuration field must contain the full path to the actual Sendmail program, such as <code>/usr/sbin/sendmail</code> or <code>/usr/lib/sendmail</code> .
Full path to sendmail aliases file	When Automatic is selected, the module works out the path to the Sendmail automatically aliases text files from the <code>AliasFile</code> entry in the <code>sendmail.cf</code> file. If for some reason this is not correct on your system, however, you can choose the second option and enter a specific alias file path instead.

Table 37.1 Module Configuration Options (Continued)

Source file for virtusers database	The address mappings (also known as <code>virtusers</code>) that Sendmail actually uses are taken from a DBM format file, rather than from the text file that the module reads and edits. The path to the DBM is specified in <code>sendmail.cf</code> , and when this configuration field is set to Same as DBM , the module will assume that the text file has the same path, minus any <code>.db</code> or <code>.dir</code> extension. Even though this is almost always the right thing to do, the source text file from which the DBM is built may be in a totally different location on some systems. If this is the case on your system, then you must select the section option for this field and enter the correct path to the <code>virtusers</code> text file, such as <code>/usr/local/etc/virtusers</code> .
Source file for mailertable database	This field has a similar purpose to the Source file for virtusers database , but applies to the domain routing or <code>mailertable</code> file instead. Once again, the default of Same as DBM is almost always correct.
Source file for generics database	This field has a similar purpose to the Source file for virtusers database , but applies to the outgoing addresses or <code>generics</code> file instead—the usage of which is not covered in this chapter.
Source file for the access database	This field has a similar purpose to the Source file for virtusers database , but applies to the spam control or <code>access</code> file.
Source file for the domains database	This field has a similar purpose to the Source file for virtusers database , but applies to the domain mapping or <code>domaintable</code> file.
User mail file location	For the User Mailboxes feature of the module to work properly, this field must be set to the directory containing user mail files, such as <code>/var/mail</code> . The only time you would need to change it is if you have reconfigured Sendmail to deliver to a different directory—for example, by using Procmail as the local delivery agent. If the option File under home directory is selected, the module will look for a mail file in each user's home directory instead. The exact filename is specified by the Mail file in home directory field below. Again, this should only be chosen if you have actually configured Sendmail to deliver to such files instead of to the <code>/var/mail</code> directory.
Mail file in home directory	When the User mail file location configuration field is set to File under home directory , this field must contain the name of a file in users' home directory to which incoming email is appended. The default is <code>mbox</code> , but <code>Mailbox</code> or <code>Inbox</code> may be used on some systems.

Table 37.1 Module Configuration Options (Continued)

Mail file directory style	When a directory is specified for the User mail file location field, this menu controls where the module looks under it for the actual mail files. The default of mail/username tells the module that mail files are in the directory, and have the same name as the user who owns them. For 99 percent of systems, this is correct and thus this field should not be changed. If you have a large number of users on your system, however, and have configured Sendmail to group user mail files into subdirectories under <code>/var/mail</code> , then the other menu options will be useful. Their meanings should be self-explanatory.
SMRSH directory	SMRSH is a program that Sendmail can be configured to use when executing programs from aliases or <code>.forward</code> files. It limits the programs that can be run to those in a particular directory, usually <code>/etc/smrsh</code> . If you have SMRSH on your system and want to be able to use this module to create autoreply or filter aliases, this field must be set to the SMRSH directory. This tells the module to create a symlink from that directory to the autoreply or filter script that it generates, so that they can actually be run. If None is selected, no such links will be created. Many Linux distributions include a Sendmail package that uses SMRSH. On those systems, this field will be set to <code>/etc/smrsh</code> by default.
Extra mail queue directories	This field can be used to specify extra mail queue directories for the module to check for queued messages that are not defined in the <code>sendmail.cf</code> file. This is rarely necessary unless you have two separate Sendmail configurations—one for incoming mail and one for outgoing. In this case, the module can only read one of the configuration files and so will not be able to automatically work out all the queue directories. Because recent versions of Mandrake Linux include Sendmail packages that work like this, the field will be set correctly by default on that distribution.

37.16 Summary

After completing this chapter, you should be familiar with the concepts behind Internet email, the protocols used to transport it, and the types of programs that use them. You should also understand what tasks Sendmail performs, and how it can be configured to redirect email with aliases, limit the addresses from or domains to which email is forwarded, and support multiple virtual domains. The use of Webmin's Sendmail module for reading users' email and viewing the mail queue should also be clear. Finally, if necessary, you should know how the module can be set up to grant limited access to a Webmin user so that he can manage only a subset of the email domains hosted on your system.

Configuring Qmail

This chapter explains how to configure the Qmail mail server on your system and compares it to the other popular server—Sendmail.

38.1 Introduction to Qmail

Qmail is probably the second most popular UNIX mail server on the Internet, behind Sendmail and in competition with Postfix. Because Sendmail uses a single server process that runs as root, any security hole in that process can allow an attacker to take over an entire system. Qmail was designed to avoid this problem by using multiple server processes and programs, each of which has only the privileges that it needs.

Before you can configure Qmail, you need to understand how Internet email works. Section 37.1 “Introduction to Internet Email” explains pretty much everything you need to know, so read it now if you are unfamiliar with SMTP, MX records, and so on. All the same principals apply to Qmail as well.

The biggest difference between Qmail and Sendmail is the location and format of the user mail file directories. Even though Qmail usually has a one-to-one mapping between UNIX users and email accounts, it does not store user email in a directory like `/var/mail`. Instead, the `Mailbox` file or the `Maildir` directory in each user’s home directory is used, depending on the Qmail configuration. The `Mailbox` file has the same standard format as normal user mail files created by Sendmail, but the `Maildir` directory is quite different. It contains three subdirectories, under which each message is stored in its own separate file. This mail storage system makes delivery of new mail much more reliable, and avoids the need to rewrite a large file when deleting a message.

All of the Qmail programs, configuration files, and queued messages are stored under the directory `/var/qmail`. There is no single master file—instead, numerous small files in the

control and alias subdirectories tell the server what to do. Because they are reread for each incoming message, any changes to these files take effect immediately.

Unlike Sendmail, Qmail does not have a permanently running server process to accept SMTP connections. Instead, it depends on a super server like `tcpserver`, `inetd`, or `xinetd` (covered in Chapter 15) to run a small program when a mail client or other MTA connects to the SMTP port. As soon as the email has been accepted and added to the queue, this program exits. A separate Qmail daemon process periodically scans the mail queue and attempts delivery of messages in it to remote systems or local mailboxes.

Email can also be added to the Qmail queue by feeding it as input to the appropriately named program `qmail-queue`, which is found in the `/var/qmail/bin` directory. Also in that directory is a program named `sendmail`, which takes the same input and parameters as the real Sendmail command but is actually just a wrapper for `qmail-queue`. On most systems that have Qmail installed, symbolic links exist from `/usr/lib/sendmail` and/or `/usr/sbin/sendmail` to this program so that other scripts or programs that expect Sendmail to be installed still work.

38.2 The Qmail Configuration Module

Webmin's module for configuring Qmail can be found under the Servers category. Assuming you have Qmail installed, clicking on its icon will take you to the module's main page, as shown in Figure 38.1. Each of the icons on the main page is a link to a page for one of the module's features, such as aliases or local domains. Under the name of each is the Qmail file or program name related to the feature, so that experienced administrators can see what each icon page is really configuring.

At the bottom of the page is a button labeled either **Start Qmail Processes** or **Stop Qmail Processes**. As their names suggest, they start the queue-processing daemon if it is not running, or stop it if it is running, respectively. Because the Qmail SMTP listener is run from `inetd`, other hosts will always be able to connect to your system. However, any email that they send will not be delivered to local mailboxes or other servers if the queue-processing server is down.

If you do not have Qmail installed, the error message **The QMail base directory /var/qmail does not exist** will appear on the main page instead. Unfortunately, very few operating systems include a Qmail package, so you will almost certainly need to download it from www.qmail.org, compile it, and install it manually. The installation process involves the creation of several UNIX users and an `inetd` or `xinetd` service, both of which can be done using Webmin. It should be possible to compile Qmail on any UNIX system, and its behavior and installation location is the same on all of them. As you would expect, this module behaves identically on all operating systems as well.

If you install Qmail, you should also configure the POP3 server program `qmail-pop3d` that is included in the package. The standard POP3 server that comes with most UNIX variants is written to look for Sendmail-style mail files in `/var/mail`, and will not work with Qmail's `~/Mailbox` files or `~/Maildir` directories. Users will be able to log in, but will not see their email!

At the time of writing this book, only MSC and Debian Linux include Qmail packages as standard. These can easily be installed using the Software Packages module, and will set up all the required users and Internet services for you. The developer of Qmail is reluctant to allow it to be packaged in formats other than the source `.tar.gz` file, which is why it is not as commonly included with Linux distributions as other MTAs.



Figure 38.1 The Qmail Configuration module.

Before you can install Qmail, you will need to shut down or uninstall any other mail server installed on your system, such as Sendmail or Postfix. This is necessary because only one program can be listed on the SMTP port, and you want to be sure that the listener is always Qmail. Uninstallation is the best option because it ensures that all start-up scripts that might restart Sendmail are deleted, and that the `sendmail` command can be linked to Qmail's wrapper program.

38.3 Editing Local Domains

When Qmail receives an email message via SMTP or one of its programs, it needs to work out whether it should be delivered locally or forwarded to another server. This is done by looking at the message's **To:** address—specifically the domain part after the @. The domain is compared to a list of local domains and, if a match is found, the email is delivered to the mailbox of the user whose name is to the left of the @ in the **To:** address (if it exists).

If the domain is not local, Qmail will look up the mail server for the domain and attempt to connect to it in order to transfer the message. This is what usually happens when a client on the same network connects to send out email. A problem will occur, however, if your server attempts to connect back to itself, which can happen if the DNS says that it is the mail server for a domain which is not on its local domain list. If this happens, a bounce message will be returned to the sender, containing text like that domain isn't in my list of allowed `rcpthosts` or too many hops.

To edit Qmail's list of local domains, follow these steps:

1. On the module's main page, click on the **Local Domains** icon. A page containing a text box for entering domain or hostnames will be displayed.
2. Select the **Domains listed below** radio button. If you leave **Only local hostname** selected, Qmail will only accept email to addresses at the system's hostname (such as *mailserver.example.com*).
3. Enter all the domains for which your system should accept mail into the text box.
4. Click the **Save** button to activate them. It is a good idea at this point to send a test email to any new domains to make sure that everything is working properly.

38.4 Managing Email Aliases

A mail alias tells your server that mail for a particular address should be forwarded to a different destination. That destination can be another email address, a local file, a directory, or even the input to a program. They can be useful for setting up pseudo mailboxes that actually send email to a real person, such as *sales@example.com* or *webmaster@example.com*. An alias can have the same name as a UNIX user, in which case it will intercept all mail to that user and forward it to a different destination instead.

To create a mail alias using Webmin, the steps to follow are:

1. Click on the **Mail Aliases** icon on the module's main page. You will be taken to a page listing all existing aliases and their destinations, with a form at the top for adding a new one. Figure 38.2 shows an example.
2. Enter a name for your new alias, such as *sales*, in the **Address** field of the form. Leave the menu set to **<All domains>** for now. Its use is explained in Section 38.6 "Managing Virtual Mappings".

The special alias named **default** will be used for any email that does not match any other alias or user mailbox. It can be useful for forwarding all messages that would otherwise bounce to some address.

3. The **Alias to** field determines where email to this alias will be sent. The following options are available from the menu:

<None> Nothing at all will be done with received email. It makes no sense to select this option when creating a new alias.

Email address Email will be forwarded to the user or address entered into the adjacent field. Be careful not to set up a forwarding loop by sending email back to the alias's address again! If you are creating an alias that has the same name as a UNIX user and really do want email to be delivered to his mailbox, as well as some other destinations, enter the username preceded by a backslash (like *\jcameron*) into this field. The backslash tells Qmail to bypass alias checking.

Mail directory Email to the alias will be added to the Qmail mail directory, whose path is entered into the text box. It must contain subdirectories named *cur*, *tmp*, and *new* to be valid.

Mail file Email received by the alias will be appended to the file whose path is entered into the text box. This should be a standard Sendmail-style mail file.

Feed to program The program whose path and parameters are entered into the text box will be run and the full text, including all headers of email received by the alias, will be fed to it as input. This kind of alias is most useful to programmers who want to perform their own custom processing or filtering of email messages. The program is usually run as the Qmail UNIX user `alias`, not `root` or the user with the same name as the alias.

Autoreply from file When email is sent to the alias, the contents of the file specified in the adjacent text box will be sent back to the original sender. Section 37.12 “Creating Autoreply Aliases” explains how autoreply files work in the Sendmail module, and they have exactly the same functionality in this module as well.

Apply filter file Email sent to the alias will be processed according to the rules in the filter file entered into the text box, which can forward to different destinations depending on the message contents. See Section 37.13 “Creating Filter Aliases” for more details.

It is possible for an alias to have multiple destinations. To add more than one, you will need to re-edit this alias after saving it, and fill in the row with `<None>` selected at the bottom of the **Alias to** table.

4. Click **Save** to have the alias added to Qmail’s configuration and activated.

As is usual in Webmin, you can edit an existing alias by clicking on its name in the list on the Mail Aliases page. This will bring up an editing form that contains all the same fields as the creation form, but has **Save** and **Delete** buttons at the bottom instead. The first of these will update the alias with any changes that you have made, while the second will permanently delete it.

If a UNIX user has a file named `.qmail` in his home directory, email that would normally be delivered to its mail file will be sent to the addresses listed in the `.qmail` file instead. If a file named `.qmail-suffix` exists, email to `username-suffix` at your server will be sent to the addresses in that file. These `.qmail` files have exactly the same format as those in the `/var/qmail/alias` directory that Webmin creates when you follow the instructions above, and thus can be used to deliver to files, directories, or programs, too.

This module does not support the editing of per-user `.qmail` files, though. Usermin (covered in Chapter 47) does allow normal users to edit their own forwarding files, however, using a web-based interface almost identical to the one described in this section.

38.5 Configuring Relaying

Qmail can be configured to restrict the destination domains to which it will relay email. This is typically done to stop spammers from using your system as an open mail relay, which allows them to hide their true addresses. There is no support in Qmail, however, for allowing clients from certain addresses to relay, so setting up relay domain restrictions will make the server useless for sending outgoing email. One solution to this problem is to run two SMTP servers—one for incoming messages that only relays mail for local domains, and another for outgoing email that uses TCP-wrapper or `xinetd` restrictions to limit access to trusted clients.

The solution recommended by the Qmail website is to use the `tcpserver` daemon to run the Qmail SMTP program, and have it set the `RELAYCLIENT` environment variable for certain clients. This tells the latter program to allow relaying no matter what is in the relay domains list, which achieves the desired objective of giving trusted clients full relay privileges. It is complex

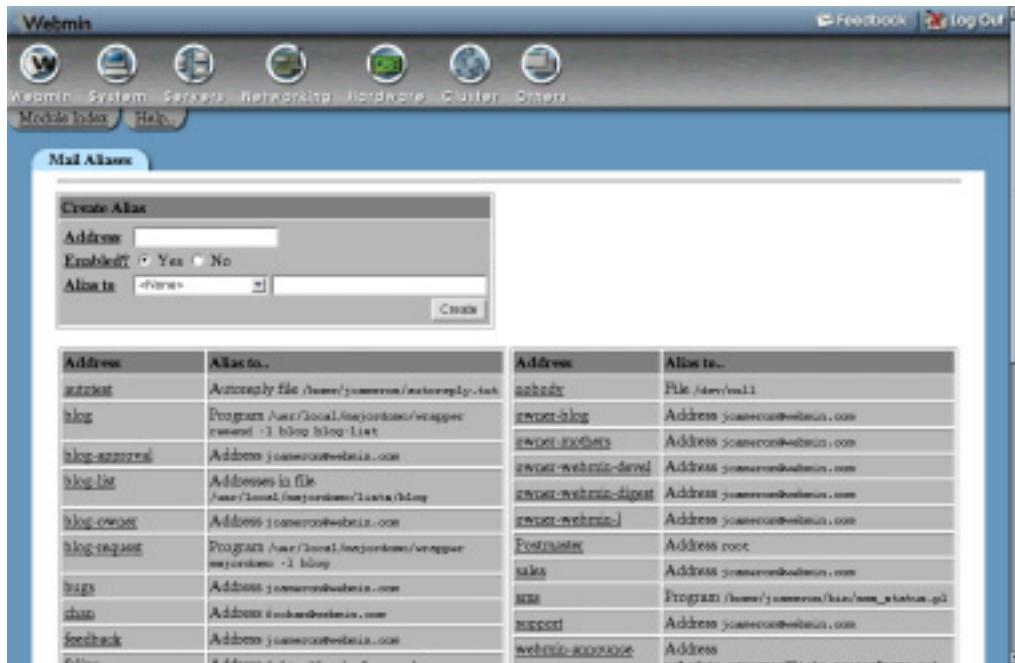


Figure 38.2 The mail aliases list.

to set up, however, and does not work with `inetd` or `xinetd`. At the time of writing of this book, Webmin does not support the configuration of this kind of relaying access control.

By default, Qmail will allow relaying to any domain. The steps to follow to change this are:

1. On the module's main page, click on the **Accepted Domains** icon. A page listing domain and hostnames to which relaying is allowed will be displayed.
 2. Select the **Domains listed below** radio button.
 3. Enter domains to which relaying should be allowed into the first text box on the page. All local domains (discussed in Section 38.3 “Editing Local Domains”) must be included as well, or mail to them will bounce.
 4. You can also enter less frequently used relay domains into the second text box. The only difference between the two is that email to domains in the first box will be processed faster.
 5. Click the **Save** button to make the relaying restrictions active.

To turn off relay domain limitations, select the **Any domain** checkbox on the Accepted Domains page and hit **Save**. Any domains that you have entered will be lost.

38.6 Managing Virtual Mappings

Qmail can be configured to treat email to the same mailbox at different domains differently, so that `sales@example.com` and `sales@foo.com` are not delivered to the same user or alias. This is vital if you are hosting multiple mail domains, as there are certain to be clashing mailbox names.

(like *sales* or *webmaster*) in several of them. Creating an alias that only applies to a certain domain, rather than to all domains, accomplishes this because the aliases created by following the instructions in Section 38.4 “Managing Email Aliases” do.

Before you can add domain-specific aliases, Qmail must first be configured to treat the domain specially. It adds a suffix like *example-* or *foo-* internally to the **To:** address of any email sent to the domain, so that you can create aliases like *example-sales* or *foo-sales*. Fortunately, Webmin does most of the work of adding these prefixes for you where appropriate.

To designate a domain as special for aliases, follow these steps:

1. On the module’s main page, click on the **Virtual Mappings** icon. A page listing all existing domains and their prefixes will be displayed, with a form at the top for adding a new one.
2. In the **Mail for address** field, select **Addresses with domain** and enter the domain name (like *example.com*) into the text box next to it.

If the **Any address not matching another virtual mapping** option is chosen, the suffix entered in the next step will be added to all **To:** addresses that do not match another virtual domain. This is not normally very useful.

If **Address** is selected and a mailbox name and domain is entered into the two text fields next to it, the mapping will apply only to that specific address. The suffix will be prepended to the username part of the address, for example, transforming *fred@example.com* to *example-fred@example.com*. This is less useful than mapping an entire domain, but can be done to give the user *fred* the ability to create personal .qmail files for different domains.

3. In the **Prepend to username** field, it is simplest to leave **Automatically chosen prefix** selected. This tells the module to take the first part of the domain name (like *example* in the case of *example.com*) as the prefix, which almost always works fine.

If you prefer to select your own prefix, choose the **Specified prefix** option and enter it into the adjacent text box. It should consist of only letters, numbers, and the – character. If you enter a UNIX username as the prefix, mail to the domain will be effected by the .qmail- files in his home directory. For example, if the prefix is *bob* and email is received for *fred@bob.com*, then ~*bob*/ .qmail-*fred* will control where it is forwarded to.

If **Nothing** is chosen, no prefix will be added for the domain at all. This can be useful if a parent domain has virtual mapping enabled.

4. Hit the **Create** button to add the new virtual domain mapping to the list.
5. Go back to the main page, and click on the **Local Domains** icon.
6. Remove the domain you have just added from the list. Otherwise, any email to it will be delivered normally as though the virtual mapping did not exist.
7. Click **Save** to update the local domains list.

As is usual in Webmin, you can edit or delete a virtual mapping after it has been created by clicking on the domain name on the list in the Virtual Mappings page. Change any of the fields and click **Save** to activate the new prefix, or hit the **Delete** button to remove it altogether. Be careful changing the prefix or deleting a mapping, as any existing aliases that use that prefix will not be updated and thus will stop working.

Once you have configured Qmail to perform virtual mapping for a domain, you can add aliases that are specific to it. To do this, follow the instructions in Section 38.4 “Managing Email Aliases”, but select the domain that the alias should be in from the menu in Step 2. After it has been added, the alias will appear in the list with its prefix—like *example-sales*—rather than as the actual address that it really matches, like *sales@example.com*.

Be aware that an alias that is not specific to any domain will not apply to email sent to that mailbox name at other domains. This is unlike the behavior of Sendmail aliases, and can be confusing if you have just added a virtual mapping for a domain and are wondering why all your old aliases have stopped working.

38.7 Configuring Domain Routing

Normally, Qmail delivers email for nonlocal domains by looking up the appropriate mail server in the DNS and connecting to it. You can use this module to configure the MTA for connecting to a different server for certain domains instead, or to send all outgoing mail via a single server. This can be useful if your system is a gateway for several internal mail servers that cannot be reached directly from the rest of the Internet, or if you want all outgoing email to be sent via your ISP or company’s server.

To specify an alternate mail server for a domain, follow these steps:

1. On the module’s main page, click on the **Domain Routing** icon to go to a page containing a list of all existing routings (if there are any) with a form at the top for adding a new one.
2. In the **Send via SMTP server** field, select the second radio button and enter a hostname or IP address into the field next to it. If the **Delivery directly** option is chosen, Qmail will perform a DNS lookup for the domain and deliver mail to the resulting server manually, even if it has been configured to send all outgoing mail via another server. In the **Create Domain Route** form, enter the domain name that you want routed via a different server into the **Mail for host or domain** field.
3. The **SMTP port** field should normally be left set to **Default**. If you choose the second option, however, Qmail will connect to the port number entered into its text field instead of the SMTP default of 25. This can be useful if, for some reason, a particular mail server is not using the normal port.
4. Click the **Create** button to save and activate the new domain routing rule.

Once a routing has been added, it will appear below the creation form on the Domain Routing page. You can edit or delete it by clicking on the domain name, changing the details, and hitting the **Save** or **Delete** button, respectively. Once again, any changes will be immediately made active.

To tell Qmail to send all outgoing email via a specified mail server, do the following:

1. Click on the **Domain Routing** icon on the main page.
2. Scroll down to the **Deliver all other outgoing mail via** field and select the second radio button. Then enter the hostname or address of the server into the text box next to it.
3. Click **Save** to activate the new setting.

To have Qmail look up and deliver normally to destination servers again, select **Deliver directly** in Step 2 instead.

38.8 Editing Global Qmail Options

Qmail has several settings that apply to all email messages that it processes, related to the hostname that it uses, SMTP timeouts, and the maximum message size. The steps below explain how to set them and what they mean.

1. On the main page of the module, click on the **QMail Options** icon to bring up a form showing and allowing the editing of global options.
2. The **Local host name** field can be used to tell Qmail your system's hostname. It should be set to the Internet domain or hostname, such as *example.com*.
3. To set the hostname that Qmail will send to remote SMTP servers, select the second option for the **Hostname for SMTP HELO** field and fill in its text box. If **Default** is selected, the hostname from the previous field will be used.
4. To change the amount of time that your server will wait for a remote MTA to accept an SMTP connection, fill in the **SMTP connection timeout** field. If **Default** is selected, a timeout of 60 seconds will be used. It may be useful to lower this to prevent your system wasting too much time trying to contact down servers—60 seconds is usually far too long to wait.
5. To set the number of seconds that Qmail will wait for a response to each SMTP command sent to a remote server, modify the **SMTP outgoing response timeout** field. If **Default** is chosen, a 20-minute timeout is used.
6. To stop your MTA from accepting large emails, select the second button in the **Maximum message size** field and enter the maximum number of bytes that an email can contain in the text box next to it. If **Unlimited** is chosen, mail of any size will be accepted. Setting a limit can be useful on systems with limited disk space or network bandwidth.
7. To set the amount of time that Qmail will wait for new data from a remote mail server connecting to your system, fill in the **SMTP incoming data timeout** field. The default is 20 minutes.
8. When your server accepts a message to an address like *fred@1.2.3.4* where *1.2.3.4* is one of the system's local IP addresses, it will convert that address into the hostname specified in the **Hostname for email to local IP address** field. Even though email is not supposed to be addressed like this, it can sometimes happen and Qmail can deal with it. If **Default** is selected, the host or domain name from the **Local host name** field is used instead.
9. To change the greeting that Qmail will present to SMTP clients when they connect, choose the second radio button in the **SMTP greeting message** field and enter some text into the adjacent text box. This message should start with the system's hostname, and if **Default** is selected, that is all it will contain.
10. Click the **Save** button to update the Qmail configuration files with the new settings.

38.9 Editing Mail User Assignments

Qmail's mail user assignment feature allows you to create "fake" mailboxes that can receive email just like real UNIX users. Each user assignment defines an additional mailbox, and has an associated UNIX username, UID, GID, and home directory in which the mail file and *.qmail* files are located. They are most useful if you want to avoid having to create a UNIX account for

every mailbox on your system, or if you want to direct mail to multiple users into the mailbox of a single real UNIX user.

To create a new mail user, follow these steps:

1. On the module's main page, click on the **Mail User Assignments** icon. A page listing existing assignments will be displayed, with a form at the top for creating a new one (as seen in Figure 38.3).
2. In the **Address username** field, select **Exact username** and enter a name (like *fred* or *joe*) into its text box. You can also choose **Usernames starting with** and enter a prefix into the box next to this option to have the mail user receive email addressed to any mailbox whose name starts with the prefix. This can be useful if you want to have email for an entire domain delivered to a single user, for later retrieval and separation by a program like Fetchmail (covered in Chapter 33). For example, if the domain *foo.com* was mapped to the prefix *foo* on the Virtual Mappings page, you could select this section option and enter *foo-* here.
3. In the **UNIX user** field, enter or select the name of a user who will own the destination mail file or directory.
4. In the **Home directory** field, enter a directory into which delivery will be made. This does not have to be the home directory of the user from the **UNIX user** field, but must be writeable by him.
5. In the **UID** box, enter the ID of the user from the **UNIX user** field.
6. In the **GID** box, enter the primary group ID of the user from the **UNIX user** field.

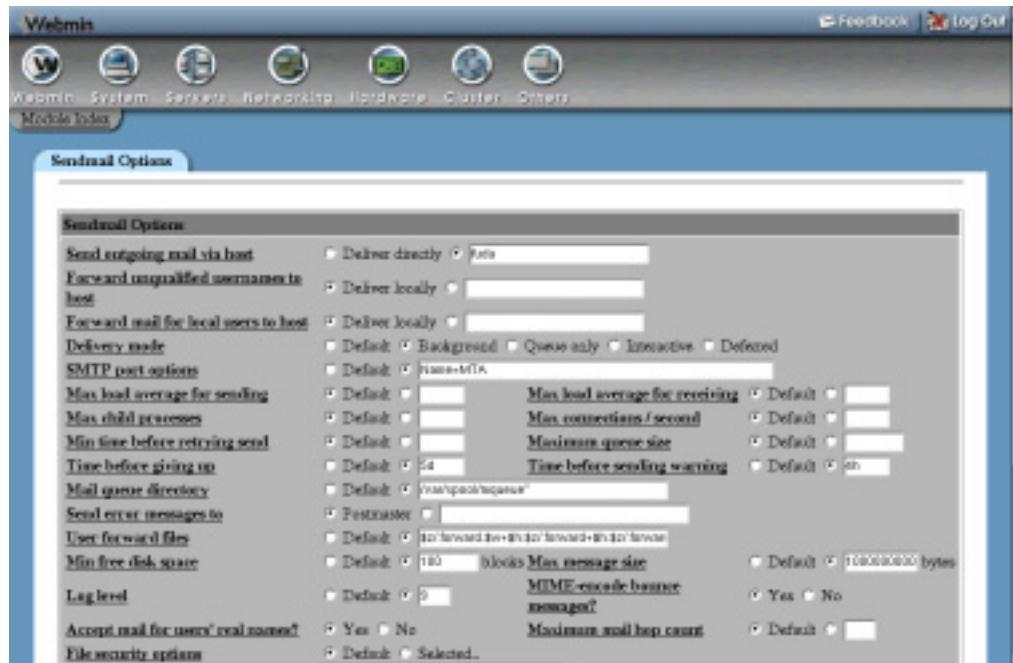


Figure 38.3 The mail user assignments list.

7. Hit the **Create** button to add and activate the new mail user assignment. It will now appear in the list on this page.

As usual, you can edit existing mail users by clicking on their names in the list, making changes on the form that appears, and clicking the **Save** button. You can also delete a user with the **Delete** button located next to **Save**. Again, any such changes take effect immediately.

One problem with mail users created by following the preceding steps is that the standard Qmail POP3 server setup does not recognize them. There are instructions and programs on the www.qmail.org, however, for setting up a POP3 server to support “fake” users and virtual domains, where they are most useful.

38.10 Viewing the Mail Queue

When Qmail receives a message, it is placed into the mail queue. If it can be sent to its destination immediately, then it will be removed from the queue almost at once. If some temporary error occurs when sending, however, then it will remain queued for later processing. The Qmail server process makes periodic checks of messages in the queue, retrying each one at longer and longer intervals until it eventually gives up.

Most messages that are in the queue for a long time are there because the destination mail server is down or unreachable. Another common cause is a temporary error reported by the remote MTA, such as a lack of disk space. Webmin allows you to view messages in the queue and even delete them by following these steps:

1. On the module’s main page, click on the **Mail Queue** icon to go to a page listing the details of queued messages. The number of emails in the queue is displayed below the icon, so that you can see how long it is at a glance.
2. On the mail queue page, the ID, sent date, sender, and destination of all queued messages are displayed in a table. If the queue contains more than 20 messages, only the first 20 will be displayed. To page through the rest, use the left and right arrow buttons that appear above the list.
3. To view the actual contents of an email, click on its ID in the queue listing. All headers, the text body, and any attachments will be displayed. To view an attachment, just click on its icon. To remove just this message from the queue, hit the **Delete** button at the bottom of the page.
4. To remove multiple messages from the queue, first select them using the checkboxes next to their IDs and the **Select all** and **Invert selection** links on the queue list page. Then, click the **Delete selected messages** button to get rid of those that you have chosen.

Unlike in the Sendmail module, there is no button on the queue page to force an immediate delivery attempt for all queued messages.

38.11 Reading Users’ Email

As the introduction explains, Qmail can be configured to store email in `Mailbox` files or `Maildir` directories in user home directories, or even under `/var/mail` like Sendmail does. Webmin allows you to read users’ email, but before it can do that you must properly configure the module so that it knows where to look. See Section 38.12 “Configuring the Qmail Configu-

ration Module" for details on which fields need to be changed. By default, the `~/Maildir` directories will be used because it is the most common Qmail configuration.

To read mail in users' mailboxes, follow these steps:

1. On the module's main page, click on the **User Mailboxes** icon. A page listing all of the users on your system and the sizes of their mailboxes will be displayed, unless you have more than 200 users. In that case, a small form for entering a username will appear instead.
2. Click on the name of a user to bring up a list of messages in his mailbox. By default, the most recent messages are shown first, even though they are actually at the end of the actual mail file.

If the mailbox contains more than 20 emails, only the first 20 will be displayed. To page through the rest, use the left and right arrow buttons above the list.

3. To view an actual message, click on the sender's name in the **From:** column. A page showing the important headers, body text, and attachments will appear. Click on an attachment icon to view it, assuming that your browser or some external program supports the data type. To remove just this email from the user's mailbox, click the **Delete** button at the bottom of the page.
4. To delete multiple messages, first select them using the checkboxes and **Select all** and **Invert selection** links on the mail list page, then click the **Delete** button.
5. To search the user's mailbox for messages matching some criteria, use the **Find messages where** form below the list. The following types of search can be selected from the menu:

From: matches, Subject: matches, To: matches, or Cc: matches Finds messages in which the From, Subject, To, or Cc field contains the text entered into the adjacent text box. The comparison is case-insensitive, but regular expression characters cannot be used.

Date: matches Finds messages in which the sending date header contains the entered text. This header will not be converted to local format, so whatever you enter must match the date format used by the sender.

Body matches Finds messages whose body contains the entered text. The body includes all attachments in their unencoded form, not just the text that is shown when you read an email.

Size is greater than Finds messages whose total size are greater than the number of bytes entered into the adjacent field.

For each of the above search types, an inverse type is also available, such as **From: doesn't match** or **Size is less than**. After choosing your search type and entering text to match, hit the **Search** button. A page listing all matching messages will be displayed, from which you can view the contents of emails or select some or all to delete—just like in the normal mail list.

The mail reading interface even allows you to compose, forward, and reply to messages in a user's mailbox. It was not designed, however, to be a general purpose web mail client. Instead, you should use a program like Usermin that has a nicer interface and supports Qmail mail directories just as well.

38.12 Configuring the Qmail Configuration Module

Like most other modules, this one has several settings that apply to the operation of the module itself rather than to Qmail. They are divided into two groups—those that affect the user interface, and those that specify the paths to Qmail configuration files and programs. When you click on the **Module Config** link on the main page, the first group of settings is listed under **Configurable options** while the second appears under **System configuration**.

Table 38.1 lists all of the available fields and explains what they do.

Table 38.1 Module Configuration Options

Sort tables by	When Order in file is chosen, the lists of aliases, virtual mappings, domain routes, and mail users will be displayed in the order in which they appear in their source text files, which is typically the order in which they were added. If Name is chosen, however, they will be sorted by name instead, which can make individual entries much easier to find.
Maximum number of records to show in tables	If the number of entries in the list of aliases, virtual mappings, domain routes, or mail users exceeds the number in this field, they will not be shown. Instead, a search form will appear allowing you to find entries by name. This is done to prevent an extremely large alias file making the Mail Aliases page enormous and slow to display. On most small systems, the default limit of 200 will never be hit.
Mail messages to display per page	This field controls the number of email messages that appear per page in the users' mailboxes and when viewing the mail queue. You may want to increase it if you like to use a large browser window.
Send mail via connection to	As the Section 38.11 “Reading Users’ Email” explains, this module can be used to compose and send email messages. When this field is set to <code>qmail-inject</code> executable, Webmin will send these messages by running the <code>/var/qmail/bin/qmail-inject</code> command and feeding it the email as input. Assuming your mail server is running properly, this should work fine. To have messages sent via an SMTP connection to some other MTA instead, select the second option for this field and enter another mail server’s hostname into the text box. You can even enter <code>localhost</code> to have email sent via the Qmail server on this system, but using SMTP instead of running a program.
Mailbox format	If you have configured Qmail to store users’ email in individual files under a directory (usually called <code>Maildir</code>), this field must be set to Directory . If a standard Sendmail-style file is used per user, then Single file must be chosen instead.

Table 38.1 Module Configuration Options (Continued)

Minimum mail file size to index	When displaying the contents of a single file mailbox, Webmin creates an index file of messages that it contains so that it does not have to reread the mail file on every page. This speeds up the display of the mailbox and single email pages, but can break down if the mailbox changes often. A common symptom of index failure is the display of numerous blank messages instead of the actual contents of a mailbox. This field can be used to turn off indexing altogether, or limit it to only large mail files. The default of 1 MB stops the indexing of small mailboxes that are frequently cleared by POP3, while enabling it for large boxes that are left on the server.
Forward messages with quoting?	When this option is enabled (as it is by default), single messages forwarded using the module's mail reading feature will have > characters prepended to each line. Since some people don't like this form of quoting, this configuration field can be used to turn it off.
Sort mail queue by	This field controls the order in which messages in Qmail's mail queue are shown. The default is to sort by queue ID, but you can have the module sort by other attributes of each message instead—but at the cost of slowing down the display for a large queue.
Show size of mail queue on main page?	This field determines if the current size of the mail queue is shown on the module's main page below the mail queue icon. You may want to disable this if your system consistently has a large queue that Webmin is slow to count.
Confirm before deleting messages?	This field determines if Webmin will ask you for confirmation before deleting messages in a user's mailbox.
Qmail base directory	This field must be set to the base directory under which all the Qmail files and programs are located. This is pretty much always /var/qmail.
Sendmail mail file location	If the Mailbox format field is set to Single file , this field tells the module where to look for user mail files. If File under home directory is chosen, the module will use the filename from the next configuration field—usually Mailbox . If the second option is selected, however, you must enter a path (like /var/mail) into the box next to it. The module will then assume that Qmail delivers to files in this directory with the same names as their owners, just like Sendmail does.
Sendmail mail file in home directory	When using the Single file mailbox format, this file tells the module the name of the mail file in users' home directories. It should almost always be set to Mailbox .

Table 38.1 Module Configuration Options (Continued)

Qmail maildir in home directory	When using the Directory mailbox format, this file tells the module the name of the Qmail mail subdirectory in users' home directories. It should almost always be set to <i>Maildir</i> .
Mail file directory style	If the Mailbox format field is set to Single file and the second option chosen in the Sendmail mail file location field , this menu tells the module how user mail files are laid out under the directory (usually <i>/var/mail</i>). The default of mail/username tells the module that mail files are in the directory and have the same name as the user who owns them. For 99 percent of systems, this is correct, and thus this field need not be changed. If you have a large number of users on your system, however, and have configured Qmail to group user mail files into subdirectories under <i>/var/mail</i> , then the other menu options will be useful. Their meanings should be self-explanatory.
Command to start Qmail	When the Start Qmail Processes button on the module's main page is clicked, this field is consulted to see which command to run. If Just run rc script is selected, the <i>/var/qmail/rc</i> script is used, which starts the <i>qmail-send</i> queue-processing daemon. If instead you select the second radio button, whatever command you enter into the adjacent text field will be run—it must be something that actually starts <i>qmail-send</i> though.
Command to stop Qmail	When the Stop Qmail Processes button on the module's main page is clicked, this field is consulted to see how to stop the <i>qmail-send</i> process. If Just kill qmail-send is selected, the module will kill all processes with that name. If you choose the second option and enter a command into its text field, however, that command will be run instead.

The two most common Qmail configurations are delivery to the **Mailbox** file or **Maildir** directory in users' home directories. By default, the module is set up to read mail from *~/Maildir*, but if you have set up Qmail to use the *~/Mailbox* file instead you must change the **Mailbox format** field to **Single file**.

38.13 Summary

This chapter has explained how Qmail differs from Sendmail and how it can be configured using Webmin. After reading it, you should be familiar with the creation and management of mail aliases, virtual domains, mail routing rules, and user assignments. You should also know how to read users' email and manage the mail queue.

Analyzing Log Files

This chapter explains how to create reports from your web or proxy server log files using the Webalizer package.

39.1 The Webalizer Logfile Analysis Module

Webalizer is a freely available program for analyzing and generating reports from Apache, Squid, and WU-FTPd log files. If you are running a website and want to see which pages are visited the most, at what times the most traffic comes, or which countries it comes from, Webalizer is the tool to use. If you manage a Squid proxy server and want to see which sites clients most commonly access and when the proxy is most heavily used, it can generate reports showing that information as well.

Unlike many of the other servers that Webmin can configure, Webalizer is relatively simple. When the `webalizer` command is run, it reads in a log file and generates HTML pages and images based on the records in that log. It can also read statistics gathered in previous runs from a history file, so that the report can include data that is no longer in the log file. The same history file is then updated with information from the latest report for use in subsequent processing. This allows the system administrator to safely delete the original log file once it has been summarized.

By default, Webalizer uses the global configuration file `/etc/webalizer.conf`, which specifies the kinds of tables and graphs to generate and titles to use. On a system that hosts multiple virtual servers, several configuration files usually exist so that different reporting options can be set for different sites. Unfortunately, there is no way to combine both options from both the global and per-log configuration files—only one can be used when generating a report.

Because log files are always having new requests appended to them, Webalizer is usually run on schedule by a program like Cron. It does not have its own server process or daemon, so it depends upon a scheduler to invoke it every day or two to reprocess each log file and regenerate each report.

Due to its relative simplicity, Webalizer behaves identically on all varieties of UNIX. This means that the functionality and layout of the Webmin module is identical as well, although the Scheduled Cron Jobs module must be installed and working for the scheduled reporting feature to work.

Webmin's Webalizer module icon can be found in the Servers category. When you first click on it, a page listing all the log files that Apache or Squid have been configured to use on your system will be displayed, as shown in Figure 39.1. By analyzing the configurations of those servers, the module can generally work out where all of the logs on your system that can be analyzed are located. However, you can easily add extra log files to the module for reporting as well.

If the module detects that Webalizer is not actually installed on your system, the main page will display an error message. If this happens, you will need to install it either from your Linux distribution CD or the program's website at www.webalizer.org. Many versions of Linux include a Webalizer package as standard, which you can install using the Software Packages module (covered in Chapter 12).

If you plan to use the module to analyze multiple log files, it is important to make sure that the global Webalizer configuration is set up correctly to support this. The version that comes with some Linux distributions (like Red Hat) incorrectly uses absolute paths for the history and caches files that store information about previous processing runs. To fix this, follow these steps before setting the options for any log files:

1. On the module's main page, click on the **Edit Global Options** button at the bottom. This will take you to a form for editing options that apply to all log files.
2. In the **Webalizer history** file field, make sure that the second radio button is selected and `webalizer.hist` appears in the text box. If some absolute path like `/var/stats/webalizer.hist` is displayed, change it.
3. Similarly, make sure that the **Webalizer incremental file** field is set to `webalizer.current` and not some full path.
4. The **Webalizer DNS cache file** can be left set to an absolute path, if you like, so that DNS information is shared between different reports.
5. Click the **Save** button at the bottom of the page to record the new settings.

39.2 Editing Report Options

Before you can generate a report from a log file, you must set certain options, such as the output directory, the UNIX user to run the report as, and report layout settings. Assuming the log has been automatically identified by the module and is displayed on the main page, you can use the following steps to set these options:

1. On the module's main page, click on the name of the web server log file for which you want to generate a report. A page listing the current settings for that file will be displayed, as shown in Figure 39.2.
2. The **All log files in report** field shows exactly which files will be used in any report created by Webmin and Webalizer. Because many systems are configured to move, truncate, compress, and eventually delete the Apache and Squid log files on a regular basis (often using a program like `logrotate`), the module will include all files in the same directory that start with the same name as the primary log file. For example, if you are reporting on

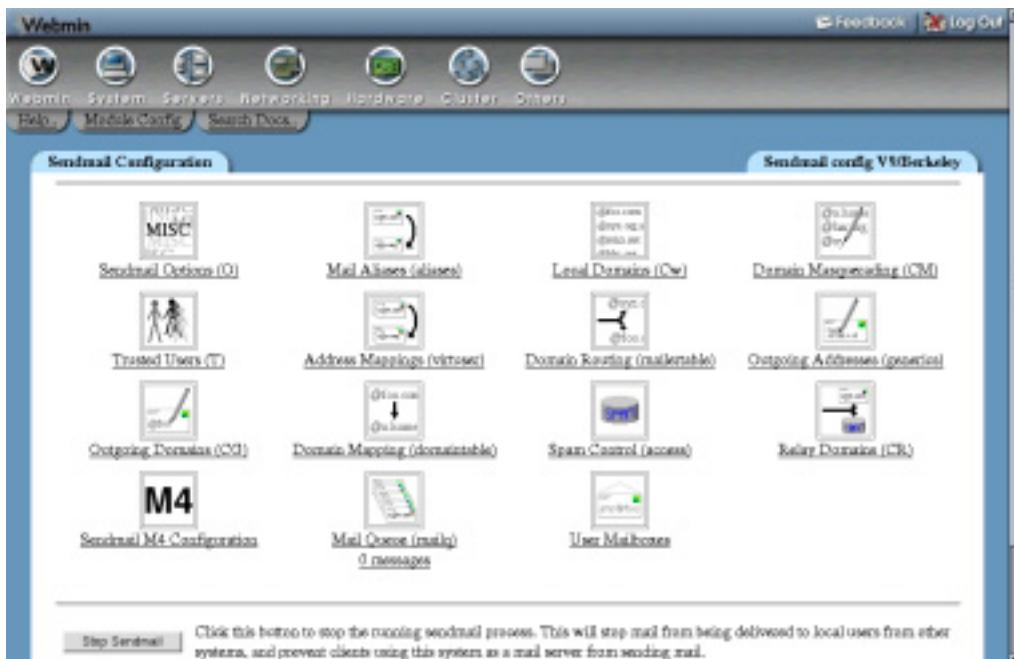


Figure 39.1 The Webalizer module main page.

/var/log/httpd/access_log, the files access_log.0.gz, access_log.1.gz, and so on in the /var/log/httpd will be displayed in this field as well.

3. In the **Write report to directory** field, enter the directory in which the HTML pages for the report should be created. This must already exist and should generally be under the website's document root—for example, /home/example.com/stats. It must be owned or writeable by the user specified in the next field. Make sure that the directory is not used for anything else, as Webalizer will create an index.html file and other HTML pages that may overwrite anything that it already contains.
4. Enter the name of the UNIX user that the generated report files should be owned by into the **Run webalizer as user** field. This should be the user who owns the website's HTML files, so that he can edit or move them if necessary. You can also just enter root if the reports are only for your own use.

Because of the way the module runs Webalizer, the user you specify does not have to have read access to the log file. He must, however, be able to write to the report directory!

5. Leave the **Always re-process log files?** field set to **No** so that Webalizer can make use of cached information from previous report runs. Setting it to **Yes** will cause all caches and previous statistics to be thrown away before each run, so that the entire log file is reprocessed. This means that any data that is no longer in the log files will not be included in the report.

Selecting **Yes** is most useful if you want to bypass Webalizer's caching of old statistics, which may be incorrect if the log file has completely changed since the last run.

6. In the **Report options** field, select **Custom options** to have the module copy the global Webalizer configuration file for this log, so you can later define options that apply only to this report. If you have only one website on your system or don't care about customizing reports for different virtual servers, you can select the **Use global options** radio button instead. If so, Steps 9 through 19 can be ignored.

The final option for this field, **Other config file**, allows you to specify an existing Webalizer configuration file to be used when generating the report. This can be useful if you have used the program before on this log file and have already customized settings for it.

7. Leave **Scheduled report generation** set to **Disabled** for now. Section 39.4 "Reporting on Schedule" explains how to enable it.
8. Click the **Save** button at the bottom of the page. As long as there were no errors in your input, you will be returned to the module's main page.
9. If **Custom options** was chosen in Step 6, click on the log filename again and then on the **Edit Options** button at the bottom of the page. This will bring up the **Options** form shown in Figure 39.3.
10. In the **Website hostname** field, select the second radio button and enter your website's name from the URL into the text field, such as *www.example.com*.
11. To customize the kinds of files that Webalizer considers to be pages, edit the extensions in the **File types to report on** field. Other types (such as images or audio files) are not counted for most reporting purposes.
12. If your site uses other directory index HTML files than those starting with `index.` (such as `home.html`) enter their filenames into the **Directory index pages** field. Normally, this field can be left empty.
13. Webalizer generally converts times in log files into your system's local time zone. To force the use of GMT instead, change the **Report times in GMT?** field to **Yes**. Unless people in different time zones are viewing the report, you should leave it set to **No**.
14. If the log file might contain records that are dated after the records that they appear before, set the **Handle out-of-order log entries?** field to **Yes**. This will slow down report generation slightly, but if **No** is chosen and the log does contain out of order records, Webalizer will not process it completely. Some web servers like Netscape's are guilty of generating log files like this.
15. The **Webalizer history file**, **Webalizer incremental file**, and **Webalizer DNS cache** fields can generally be left unchanged, as long as they are set to relative paths. The introduction to this chapter explains in more detail why this is necessary.
16. In the **Graphs and tables to display** section, deselect those that you don't want included in the report.
17. In the **Table rows and visibility** section, you can change the size of each table that appears or remove it altogether by selecting **None**.
18. To turn on the creation of extra pages in the report listing all clients that access your site, URLs accessed, and so on, select the appropriate checkboxes in the **Generate pages listing all section**. Otherwise, only tables showing the top 20 will be include in the report.
19. Finally, click the **Save** button at the bottom of the page. Reports generated from now on will use these options.

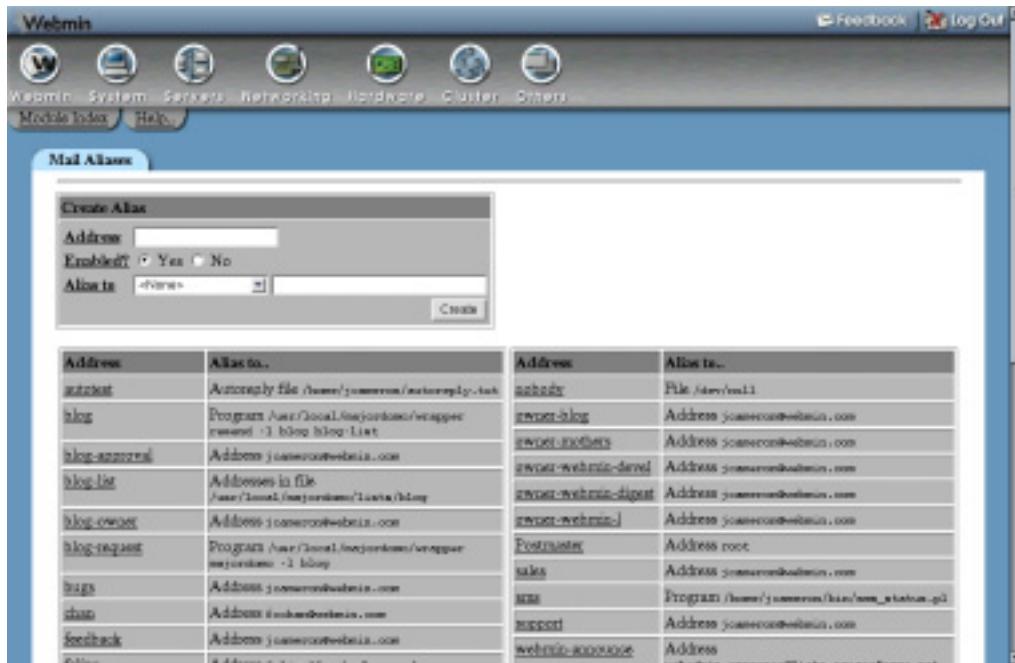


Figure 39.2 The log file options page.

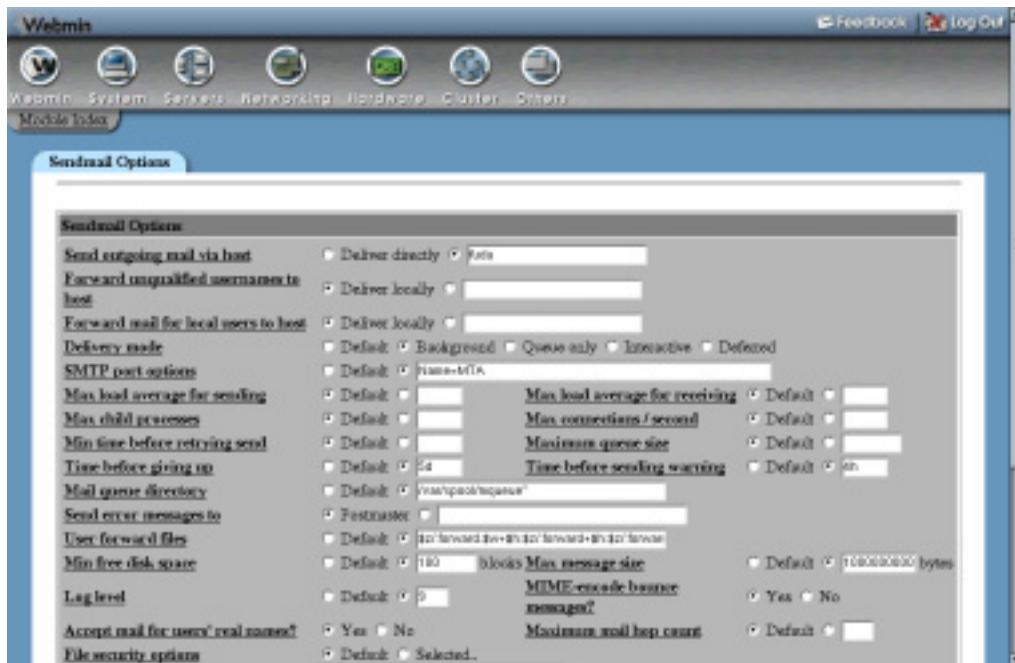


Figure 39.3 The report options page.

Although the preceding instructions are written with Apache log files in mind, they apply to Squid logs as well. The only difference is that Squid has no document `root` directory, so you will have to create a new directory for the report. This could be under the `root` directory of your web server so that the report can be viewed by anyone. If so, the name of the UNIX user who owns the web server's HTML files should be entered in the **Run webalizer as user** field.

39.3 Generating and Viewing a Report

Once you have set the options for a report, actually generating it is simple. Just follow these steps:

1. On the main page, click on the name of the log file for which the report is being generated.
2. Hit the **Generate Report** button at the bottom of the form. A page showing the output from Webalizer as it is run on each of the log files will be displayed so that you can see any errors that occur. This can take a long time (perhaps hours) the first time a large log file is processed, as a reverse look up must be done for every client IP address in the file. Fortunately, the actual CPU and network load that is generated is minimal.
3. If all goes well, the report's HTML pages will be created in the destination directory. To view it, click on the **View completed report** link below the output.
4. The report's first page shows a graph of hits received by the website by month, with links below to pages containing details for each individual month. Each of the month pages show tables and graphs of hits by day, hour, client, page, and country for the site, and may also show hits by user, browser, and referrer as well—if that information is included in your log files.
5. The same report can be viewed directly from the module's main page by clicking on the **View** link in the **Report** column for the log file or by hitting the **View Report** button on the log file options form.

39.4 Reporting on Schedule

Instead of manually generating a report from a log file, you can use the Webalizer Logfile Analysis module to set up a Cron job that runs Webalizer on a regular basis. A report should generally be refreshed every one or two days, depending on the size of the log file. Because some large logs take a long time to process, refreshing too frequently (such as once per hour) could cause multiple Webalizer processes to be run on the same log file at the same time, which will corrupt the resulting report.

It is generally a good idea to generate a report for the log file from within Webmin at least once before setting up scheduled reporting, so that you can see if it is really working or not. Once you have done that, follow these steps to set up scheduled reporting:

1. On the module's main page, click on the log file's name. This will bring you to the options form shown in Figure 39.2.
2. Change the **Scheduled report** generation field to **Enabled, at times chosen below**.
3. Select the times and days on which the log file should be reprocessed from the **Minutes**, **Hours**, **Days**, **Months**, and **Weekdays** lists. For each one, you can either choose **All** to have the report generated every minute, hour, or whatever. Or you can choose **Selected** to have Webalizer run only at the times or dates selected from the list. To select multiple

entries, hold down the **Control** or **Shift** keys while clicking. You can also **Control-click** to deselect entries that have already been chosen.

By default, the log will be processed at midnight every day. If you have multiple reports that are being generated on schedule, try to stagger them so that they are not all run at the same time. For example, in your second report, select **1** as the hour instead of **0** and so on.

4. Click the **Save** button to have Webmin create a Cron job for the report. You will be able to see it in the Scheduled Cron Jobs module (covered in Chapter 10), but you should only edit the dates and times here.

To turn off regular report generation for a log file, select **Disabled** for the **Scheduled report generation** field. The Cron job will be deleted, but the times and dates at which it was set to run will be remembered so that you can easily enable it again.

39.5 Adding Another Log File

Even though the module attempts to automatically identify all the log files on your system by reading the Apache and Squid configuration files, there may be some that it misses. This can happen if the Apache Web server or Squid Proxy Server modules (covered in Chapters 29 and 44, respectively) have not been set up properly, if you have more than one copy of Apache installed on your system, or if the web server has been configured to log to a filter program rather than to a normal file.

If you want to generate a report from an FTP server log file, you will definitely need to add the file to the module, as it does not detect WU-FTPd logs automatically. You can also add logs from other web servers such as Zeus, TUX, Netscape, or NSCA, assuming they use the standard CLF format that Apache does. It is even possible to create a report on the logs created by Webmin and Usermin, found at `/var/webmin/miniserv.log` and `/var/usermin/miniserv.log`, respectively.

The steps to manually add a log file for reporting are:

1. On the module's main page, click on the **Add a new log file for analysis** link above or below the table of existing logs.
2. In the **Base logfile path** field, enter the full path to the log file, such as `/usr/local/apache/var/foo.com.log`. If any other log files exist in the same directory whose names start with `foo.com.log`, they will be included in the report as well.
3. From the **Log file type** menu, select either **Apache** for CLF format files generated by a web server, **Squid** for logs from the Squid proxy server, or **FTP** for transfer logs from WU-FTPd.
4. The rest of the form can be completed in exactly the same way as you would for an existing log file. Just follow Steps 3 through 19 in Section 39.2 "Editing Report Options".

One difference between manually added log files and those automatically detected by the module is the presence of a **Delete** button at the bottom of the log file options page. Clicking it will delete the log from the list on the main page, but will leave any reports and the log file itself untouched.

39.6 Editing Global Options

Webalizer has a master configuration file named `/etc/webalizer.conf` that is used by the module if the **Report options** field is set to **Use global options**. It is also copied when you select **Custom options** to provide the initial settings for the per-log file configuration. Changing the global options afterwards, however, will have no effect on any logs that are already using their own configuration file.

If you only have one log file on your system that needs analysis, it makes more sense to use only the global `webalizer.conf` file instead of having one created just for the report on that log. And if you plan to set up reporting on multiple log files, you should edit the global Webalizer configuration first to provide a template from which the per-log configurations are copied. To edit it, follow these steps:

1. On the module's main page, click on the **Edit Global Options** icon. Your browser will display an **Options** form similar to the one in Figure 39.3.
2. Follow Steps 11 through 19 in Section 39.2 “Editing Report Options” to configure the appearance of all reports. The fields on this form have exactly the same meanings as those on the per-report options page.
3. Click the **Save** button to update the configuration file with your changes.

If you are generating more than one report, it makes much more sense to set options for each individually. That way, you can set a different web server hostname for each so that the title and links to pages on each report are correct.

39.7 Module Access Control

As Chapter 52 explains, you can create a Webmin user or group that has access to only a limited subset of the features of most modules. In the case of the Webalizer module, you can grant a user the rights to edit options for and generate reports from only some of the logs on your system. This can be useful if your system hosts multiple Apache virtual servers, each owned by a different person. As long as each server has its own separate log file, you can give a Webmin user the rights to manage both a virtual server and its log report.

Once a user has been given access to the module, you can use the following steps to limit him to only some of the log files on your system:

1. In the Webmin Users module, click on **Webalizer Logfile Analysis** next to the name of the user. This will bring up the standard module access control form.
2. Change the **Can edit module configuration?** field to **No** so he cannot modify the paths to Webalizer or its global configuration file.
3. Leave **Can only view existing reports?** set to **No** so the user can edit the options for reports on log files that he owns.
4. Set **Can edit global webalizer options?** to **No** to prevent the user editing options that may apply to other people's logs.
5. In the **Run Webalizer as user** field, select the last radio button and enter the name of the UNIX user as whom this Webmin user normally logs in. This will stop him from setting up reports that are generated as `root`, which could be a serious security risk, as it would allow system files and those belonging to other people to be overwritten.

6. In the **Only allow viewing and editing of reports for logs under** field, enter either the full path to a log file (like `/var/log/httpd/example.com.log`) or a directory that has log files under it (such as `/home/example.com/logs`). The module will hide any automatically discovered logs outside that directory so that the user cannot set up reports for other people's websites.
7. Hit the **Save** button to activate the new restrictions.

Once a user has been restricted in this way, he will be able to use the module to set up reporting for only those log files in the directory set in Step 6. Reports will only be generated as the UNIX user specified in Step 5, which stops the Webmin user from overwriting files that he would not normally be able to at a shell prompt. This makes the module quite safe for untrusted people to use, although a malicious user could set up a reporting Cron job that runs extremely frequently and uses up an excessive amount of CPU time.

You can set the paths that the module uses for the Webalizer program and its global configuration file by using the module configuration form, reachable through the standard **Module Config** link on the main page. When clicked on, it displays a form containing the fields shown in Table 39.1.

Table 39.1 Module Configuration Options

Path to webalizer command	This field must contain the full path to the <code>webalizer</code> executable, or just <code>webalizer</code> if it is in one of Webmin's program path directories, as is usually the case.
Path to webalizer configuration file	This field must contain the full path to the global Webalizer configuration file, which is usually <code>/etc/webalizer.conf</code> . Additional files created for specific log file reports are always stored in the <code>/etc/webmin/webalizer</code> directory.
Sample webalizer configuration file	If the global configuration file does not exist, the module will copy the file from the path specified in this field instead. This is used on some Linux distributions that include a sample file in some other directory, but not <code>/etc/webalizer.conf</code> . Once the module is running, the same file is never used so changing this field will have no effect.
Automatically include logfiles from	The checkbox in this field controls from which servers the module automatically retrieves log files. By default, both Apache and Squid are selected, but if you are only generating reports on one (or neither) of these servers, you may want to deselect them so that the main page is not cluttered up with log files that you don't care about.

39.8 Summary

This chapter has explained how to use Webalizer to create reports containing graphs and tables from the web, FTP, and proxy server log files on your system. It has explained how to set up scheduled report generation, how to view reports, and how to restrict a Webmin user to only being able to configure reports for selected log files.

The ProFTPD Server

This chapter explains the FTP protocol, how to set up the ProFTPD server and how to configure it for various purposes.

40.1 Introduction to FTP and ProFTPD

FTP stands for File Transfer Protocol, and along with telnet and SMTP is one of the oldest protocols still in common use on the Internet. FTP is designed to allow client programs to read, write, and delete files on a remote server, regardless of the operating system that the server is running. It is essentially a file sharing protocol, but unlike the more common NFS and SMB protocols, it is better suited to use over a slow or high-latency network.

Typically, FTP is used to transfer files from one system to another. Sometimes those files are Linux distribution CD images or RPM packages, downloaded by various client hosts on the Internet from a large server system that hosts them for everyone to access. Other times, the files are pages for a website, uploaded by an FTP client run by the site's owner to a system that runs both the web server and an FTP server.

Even though the FTP protocol has been mostly replaced by HTTP as a method of downloading files, it still has many advantages. The biggest is the clients' ability to upload files to the server, assuming that it has been configured to allow them. Another is a semi-standard directory listing format, which clients can use to fetch a list of files in a directory from the server.

When an FTP client connects to a server, it must first authenticate itself before any file transfers can take place. Often, clients will log in as the special anonymous user, which requires no password and is usually configured to be only able to download files. On UNIX systems, most FTP servers allow any local user to log in with the same username and password that he would use for telnet or SSH, and give his client access to the same files with the same permissions.

Another unique feature of the FTP protocol is its support for translating files between the data format used on the client and that used on the server. The most common use of this is the

conversion of text files between the UNIX, Windows, and MacOS formats, each of which uses different characters to represent the end of a line. This feature can be disabled for the transfer of binary files such as images, executables, and ISOs, as it corrupts non-text data.

Many different FTP client programs exist, from the basic UNIX `f tp` command to browsers like Internet Explorer and Mozilla. Every modern operating system has at least one, and almost all include a client of some kind as standard. FTP servers are also plentiful, but this chapter focuses on only one—ProFTPD, which in my opinion is the most flexible server available for UNIX operating systems.

Even though all varieties of UNIX ship with an FTP server as standard, the supplied server is usually either very basic and lacking in features, or the more powerful WU-FTPd. The latter is the most common FTP server in use today, and although it has many configurable options, it is not as capable as ProFTPD when it comes to virtual hosting, directory restrictions, and locking users into their home directories. It is covered in Chapter 41, however, so if you already have WU-FTPd installed and don't want to bother switching, read that chapter instead.

ProFTPD generally uses a single configuration file, found at `/etc/proftpd.conf`. This file is made up of directives, each of which usually occupies a single line and has a name and value. Each directive sets a single configurable option, such as the name of a hidden file or the path to a welcome message. There are also special container directives for grouping other directives that apply only to a single virtual server or directory, and which span multiple lines.

40.2 The ProFTPD Server Module

The ProFTPD Server module icon can be found in Webmin under the **Servers** tab on the main menu. When you click on it, the module's main page will appear as shown in Figure 40.1, assuming that you actually have the server installed. If the main page displays an error message like **The ProFTPD server /usr/sbin/proftpd could not be found on your system** instead, then the server is probably not installed and thus the module cannot be used. Most Linux distributions include a ProFTPD package on their CD or website, so use the Software Packages module (covered in Chapter 12) to install it. If no package exists, download the source code from www.proftpd.org, compile, and install it. If you already have some other FTP server installed, it should be removed first so that the two programs do not clash.

Another error that the main page might display is **The program /usr/sbin/ftp does not appear to be the ProFTPD server error**. This will occur if Webmin detects that some other FTP server is installed instead—if so, you will need to remove it and install ProFTPD.

ProFTPD can be run in two different modes—either as a standalone daemon process that listens for FTP connections, or from a super server like `inetd` or `xinetd`. The former accepts connections faster, but at the cost of more memory being used up by a process that is running all the time. The latter is better for systems that do not expect to receive a lot of FTP traffic, as the ProFTPD program is run only when it is needed.

Because the standalone mode is easier to set up and because memory is plentiful on most systems, this chapter assumes that you will be running it in that mode. To start the ProFTPD server process, follow these steps:

1. In the Internet Services and Protocols module (covered in Chapter 15), make sure that any existing service named `ftp` has **Program disabled** or **No program assigned** selected. This ensures that no FTP service will be run by `inetd`. If you disable a service,



Figure 40.1 The ProFTPD Server module.

make sure to hit the **Apply Changes** button on that module's main page to activate your changes.

2. In the Extended Internet Services module, make sure that any services with `ftp` in their names (such as `wu-ftp`, `proftpd`, or `vsftpd`) have their **Service enabled?** field set to **No**. Again, you will need to hit the module's **Apply Changes** button to activate any changes.
3. Back in the ProFTPD Server module, click on the **Networking Options** icon.
4. Select **Standalone daemon** from the **Server type** menu.
5. Click the **Save** button at the bottom of the page.
6. Back on the module's main page, a button labeled **Start Server** should appear at the bottom. Click it to start the ProFTPD daemon.
7. If you want the daemon to be restarted at boot time, use the Bootup and Shutdown module to create an action called `proftpd` that runs the command `/usr/sbin/proftpd` at boot time. The actual path may be `/usr/local/sbin/proftpd` or `/usr/sbin/in.proftpd`, depending on which Linux distribution you are running or if you compiled and installed the program yourself instead of using a package. Also, some ProFTPD packages may include a bootup script like this already, which you may just have to enable.

Once ProFTPD has been started, you can test it by using the command line UNIX FTP client to connect to your own system. Just run `ftp localhost`, and make sure that you can log in as some user other than `root`. You can verify that the server really is ProFTPD by checking the version displayed by the `ftp` command just before it prompts for a username, unless it has been configured by default not to display version information.

40.3 Running ProFTPD from inetd or xinetd

Setting up ProFTPD to run from a super server isn't too hard either, and may be a good idea if your system is low on memory or hardly ever receives FTP connections. Before you can do this, you must kill any existing `proftpd` server process (easily done with the Running Processes module), and disable or delete any action that starts it at boot time.

If your system uses the superior `xinetd`, follow these instructions to set up the FTP service. Because many packages include an `/etc/xinetd.d` configuration file for the server, some of the fields explained below may already be filled in correctly.

1. Go to Webmin's Networking category and click on the **Extended Internet Services** icon. If it does not exist, `xinetd` is not installed and you will need to set up the server using `inetd` instead.
2. On the module's main page, check for an existing service named `ftp` or `proftpd`. If one exists, click on it—otherwise, follow the **Create a new internet service** link above or below the table.
3. In the **Service name** field, enter `ftp` (unless it has already been filled in).
4. Make sure the **Yes** option is selected in the **Service enabled?** field.
5. Leave the **Bind to address** field set to **All**, and the **Port number** to **Standard** or **21**.
6. Select **Stream** from the **Socket type** menu, and **Default** or **TCP** from the **Protocol** list.
7. In the **Service handled by** field, select the **Server program** option and enter the path to the `proftpd` executable (such as `/usr/sbin/proftpd`) into the adjacent text box. The path depends on whether you installed the program from a package or compiled it from the source code.
8. In the **Run as user** field, enter `root`.
9. Select **No** for the **Wait until complete?** field.
10. Leave all the other fields set to their defaults, and hit the **Save** or **Create** button at the bottom of the form.
11. Back on the module's main page, click the **Apply Changes** button below the list of services.

To set up an `inetd` service for ProFTPD using the Internet Services and Protocols module instead, follow these steps:

1. Go to Webmin's Networking category and click on the **Internet Services and Protocols** icon. If it does not exist, your system is probably using `xinetd` instead. See the steps in the previous paragraph for instructions on how to configure it.
2. On the module's main page, click on `ftp` in the **Internet Services** table. If it is not visible, enter `ftp` into the **Edit service** field and hit the button. Either way, the same page for editing the FTP protocol service will be displayed.
3. In the **Server Program** section, select **Program enabled**.
4. In the **Program field**, select the **Command** option and enter the full path to the ProFTPD server executable into the field next to it, such as `/usr/sbin/proftpd`. In the **Args** field, enter just `proftpd`. The path depends on whether you installed the program from a package or compiled it from the source code.
5. Set the **Wait mode** to **Don't wait**, and enter `root` in the **Execute as User** field. All others can be left unchanged.
6. Click the **Save** button, and then click **Apply Changes** back on the module's main page.

Once ProFTPD has been set up to run from `inetd` or `xinetd`, you can test it by using the command-line UNIX FTP client to connect to your own system. Just run `ftp localhost`, and make sure that you can log in as some user other than `root`. If your test connection fails with an error like **Service not available**, the most likely cause is that ProFTPD is configured to run as a standalone server. This can be easily fixed by following these steps:

1. Go to the ProFTPD Server module and click on the **Networking Options** icon on the main page.
2. From the **Server type** menu in the form that appears, select **Run from Inetd**.
3. Hit the **Save** button at the bottom of the page.

The instructions in the rest of this chapter will work fine no matter in which mode ProFTPD is running. The only difference is that the **Apply Changes** button will not appear on the main page, as there is no need to restart a server process for any configuration changes to take effect. Instead, changes will apply to the next FTP session that is started.

40.4 Using the ProFTPD Server Module

ProFTPD uses a very similar configuration file format to Apache, and so the user interface for this module is the same in many ways as the Apache Configuration module. At the highest level in the configuration are global settings that affect the entire server. Below them are virtual servers, and then anonymous FTP options, per-directory options, and options that apply only to certain FTP commands. The diagram in Figure 40.2 shows the hierarchy.

The options that apply to each connection or FTP command are determined by the virtual server to which it is connected, the type of login, the directory in which the requested file sits, and the specific FTP command used. Options set by objects lower in the hierarchy override those at upper levels, so you can prevent uploading to a server but allow it for a directory. Similarly, options for a more specific directory (like `/usr/local/upload`) override those for its parents (such as `/usr/local`).

A special case is the default server, which defines settings for clients that do not connect to any specific virtual server. Unlike Apache, options set in the default server do not affect virtual servers. Instead, if you want to specify a setting that affects all of them, it must be in the special global section of the ProFTPD configuration. This applies to directory and FTP command-specific options as well.

The module has a page for editing options for each object in the tree, which contains icons linking to objects further down. For example, on the virtual server options page are icons for the various categories of options that apply to that server (such as logging, users, and groups), along with icons for any directories or FTP commands that have their own options within the virtual server. There is also an icon for options specific to anonymous FTP connections.

On each page in the hierarchy, are forms for adding objects (such as a directory or group of FTP commands) under it, and a **Configure** icon for changing or deleting the current object. Every page also contains an **Edit Directives** icon allowing you to view and manually change the ProFTPD directives for the directory, virtual server, or whatever it is that the page represents. The exception is the default server page, which has no such icons because it cannot be changed or deleted and because its directives cannot be separated from the rest of the configuration file.

At first glance, some of the forms in the module may appear daunting as they display fields for almost all of the available ProFTPD options in some category related to an object. Many of

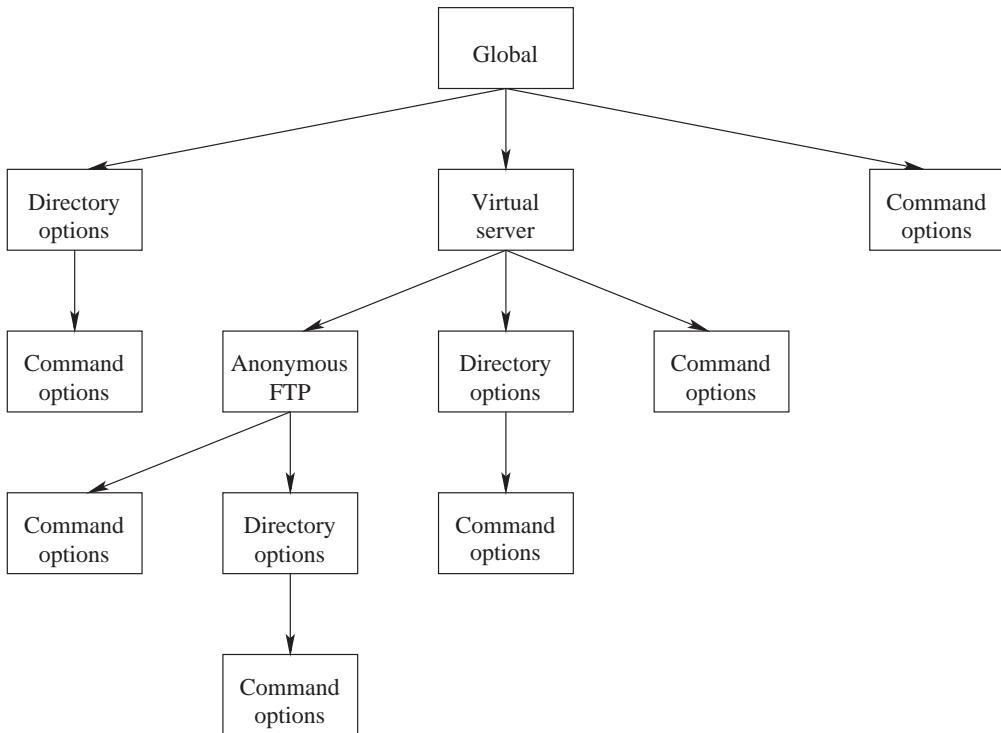


Figure 40.2 The ProFTPD configuration hierarchy.

these options, however, are extremely specialized and can be ignored most of the time. The steps in the various sections of this chapter explain which ones you need to modify to achieve a specific result. The others can be left alone, as their defaults are usually adequate.

Because each new version of ProFTPD that is released supports new directives, the ProFTPD Server module can detect the version that you are running and adjust its user interface to display only those fields that are valid for your version. This means that the forms may not look exactly the same on all systems, and that some parts of the instructions in this chapter may not be valid for your FTP server if you are running an older release.

40.5 Creating Virtual Servers

ProFTPD's most useful feature is probably its support for virtual FTP servers. This allows you to define a totally different set of options that apply to clients connecting to a particular IP address. In most ways, they are similar to Apache's IP-based virtual servers, with which most website administrators should be familiar.

Virtual servers are only really useful if your system has multiple IP addresses. Typically, this is done by adding additional virtual IP addresses to your Internet-connected network interface, as explained in Chapter 16, which covers Network Configuration. As usual, any extra IP addresses must be properly routed to your system. If you are connected to an ISP and assigned only a single static address, you cannot just add additional virtual interfaces and expect them to

work. Unlike Apache, ProFTPD does not support name-based virtual servers because there is no provision in the FTP protocol for them. Clients never tell the server the hostname to which they are connecting, so the FTP server can only use the IP address on which a connection was received to determine which virtual server the client wants.

When your system receives an FTP connection, ProFTPD will compare the connected address with those of all configured virtual servers. The first one to match defines the options that apply to the connection. If no match is found, the default server is used instead.

To add a new virtual FTP server to your system, follow these steps:

1. In the Network Configuration module, add a new virtual IP address to the external network interface on your system. Make sure that it will be activated at boot time and is active now.
2. Back in the ProFTPD Server module, scroll down to the **Create virtual server** form at the bottom of the main page.
3. In the **Address** field, enter the IP address that you just assigned. It should not be used by any other virtual server already defined.
4. Leave the **Port** field set to **Default**.
5. In the **Server name** field, select the second radio button and enter a name for this server that will be displayed to connecting clients. For example, you could enter *Example Corporation's FTP server*. If **Default** is selected, clients will see a message like **ProFTPD 1.2.2rc2 Server** instead.
6. Hit the **Create** button to add the server. Once it has been created, you will be taken to the new server's options page.
7. Return to the module's main page and click the **Apply Changes** button to make it active.

Once a virtual server has been created, you can set options that apply to it by clicking on its icon on the main page, then clicking on one of the category icons. Some of these are explained in more detail later in this chapter. It is also possible to change the attributes of a virtual server by clicking on the **Configure Virtual Server** icon, editing the fields on the form (which have the same meanings as those on the creation form) and clicking **Save**. Or you can remove it altogether by hitting the **Delete virtual server** button on the configuration form.

40.6 Setting Up Anonymous FTP

In its default configuration, ProFTPD will generally allow all UNIX users to log in with their normal passwords and access all files on the system with the same permissions that they would have if logged in via telnet or SSH. Some packages also have anonymous FTP enabled for the default server as well, so that anyone can connect as the anonymous user and view files in a specific directory. To set up anonymous FTP for a new virtual server, configure what clients can do, and determine which directories they can access, follow these steps:

1. On the module's main page, click on the icon for the default or virtual server for which you want to configure anonymous FTP.
2. On the virtual server options page, click on the **Anonymous FTP** icon. If this is the first time that it has been setup for this server, a small form will appear for entering anonymous FTP settings.

3. In the **Limit to directory** field, enter the directory to which anonymous clients should be restricted, such as `/home/example.com/anonftp`.
4. In the **Access files as user** option, select the second radio button and enter the name of an unprivileged UNIX user such as `ftp` or `nobody`. Clients will not only be restricted to the chosen directory, but will also be only able to access files with the permissions of that UNIX user. Naturally, you should make sure that it can actually read and list the directory and files that it contains. This user must not be in ProFTPD's denied list, or have an invalid shell. See Section 40.8 "Limiting Who Can Log In" for more information on editing this list and allowing users with any shell.
5. If you are happy for clients to use the group permissions of the user set in the previous field, leave the **Access files as group** field set to **Default**. Otherwise, select the second radio button and enter a group name into its field.
6. Hit the **Create** button to set up the initial anonymous FTP configuration. Assuming it is successful, the browser will be redirected to the anonymous FTP options page on which are icons for the various categories of configurable options that relate to anonymous FTP connections.
7. Click the **Save** button to return to the anonymous FTP options page. Click on **Authentication** and in the Username aliases table enter *anonymous* under **Login username**, and the name of the user that you chose in Step 4 under **Real username**. This tells ProFTPD that clients logging in as anonymous should be given the permissions of that user.
8. In the **FTP commands** field, enter `WRITE` and hit the **Create** button to start the process of defining options that apply to FTP commands that modify data on the server. You will be taken to the per command-options page.
9. Click on the **Access Control** icon, and select **Deny all clients** in the **Access control** policy field. This tells ProFTPD to block attempts by anonymous clients to upload, delete, or rename files.
10. Click the **Save** button.
11. Return to the module's main page, and hit **Apply Changes**. To make sure that everything is working, try logging into the virtual server as the anonymous user and downloading some files.

If you are using your system to host multiple web and FTP sites for different customers, each can be given his own virtual anonymous server to make files available to people via FTP. Browsers assume that `ftp://` URLs require an anonymous login and most don't deal well with FTP servers that require authentication.

40.7 Restricting Users to Their Home Directories

By default, clients that log in to ProFTPD as a valid UNIX user (not anonymous) can browse your system's entire filesystem, just as they could if the user logged in via SSH or telnet. This is not always desirable on a system that has multiple untrusted users whom you want to prevent from seeing each others files. Even though UNIX permissions can be used to stop users from listing each others' directories, they cause problems if you are also running a webserver and need its `httpd` user to have access to everyone's files.

Fortunately, ProFTPD makes it easy to restrict users to their home directories or to some other directory. Because this only applies to FTP connections, it is pretty useless if those same

users can telnet or SSH in. It is easy, however, to allow a user to connect only via FTP by giving him a shell like `/bin/false`. On a virtual hosting server, users only really need to upload files for their websites and do not need UNIX shell access at all. Just make sure that `/bin/false` or whatever nonfunctional shell that you choose is included in the `/etc/shells` file so that ProFTPD does not deny the users access.

To restrict the directories that FTP clients can access, follow these steps:

1. If you want the restriction to apply to only a single virtual server, click on its icon on the module's main page and then on the **Files and Directories** icon on the virtual server options page. This is not advisable, however, as it may allow users to avoid the restriction by connecting to another virtual server. Instead, you should just hit the **Files and Directories** icon in the **Global Configuration** section on the main page. Any restrictions defined on it will apply to all servers. Either way, the page for configuring how the server lists directories and which ones are available (shown in Figure 40.3) will appear.
2. The **Limit users to directories** field is actually a table that allows you to enter one directory limitation at a time. It will always have one blank row, and if this is the first such restriction you have created that is all it will contain.

In the **Directory** column, select **Home directory** if that is to what you want users to be restricted. You can also select the third radio button and enter a path like `/home` or `/var/www` to confine users to that directory. It is also possible to enter a path relative to the users' home directories, such as `~/public_html`.

In the **UNIX groups** column, either select **Everyone** to have the restriction apply to all users, or select the second radio button and enter a group name to have it apply only to the members of that group. Multiple groups can be entered by separating their names with commas, like `users,staff`.

3. Click the **Save** button to return to the virtual server options page. If you want to add another restriction (such as for a different group and directory), click on **Files and Directories** again and fill in the new blank row in the table.
4. When done, return to the module's main page and hit the **Apply Changes** button to make the restrictions active.

From now on when restricted users connect, they will be unable to see files outside the specified directory or even work out to which directory they have been limited. Unlike some other FTP servers that support this kind of restriction, there is no need to copy any files or libraries like `/bin/ls` into the directory, as ProFTPD does not depend on any external programs.

40.8 Limiting Who Can Log In

ProFTPD does not allow every UNIX user to log in, even if they have valid usernames and passwords. The separate `/etc/ftpusers` file lists users who are not allowed to authenticate, which typically include system accounts such as `bin`, `daemon`, and `uucp`. In addition, there is a separate configuration option that controls whether the `root` user is allowed to log in or not. By default it is not, because passwords sent by the FTP protocol are not encrypted, so allowing `root` to authenticate could be a major security risk.

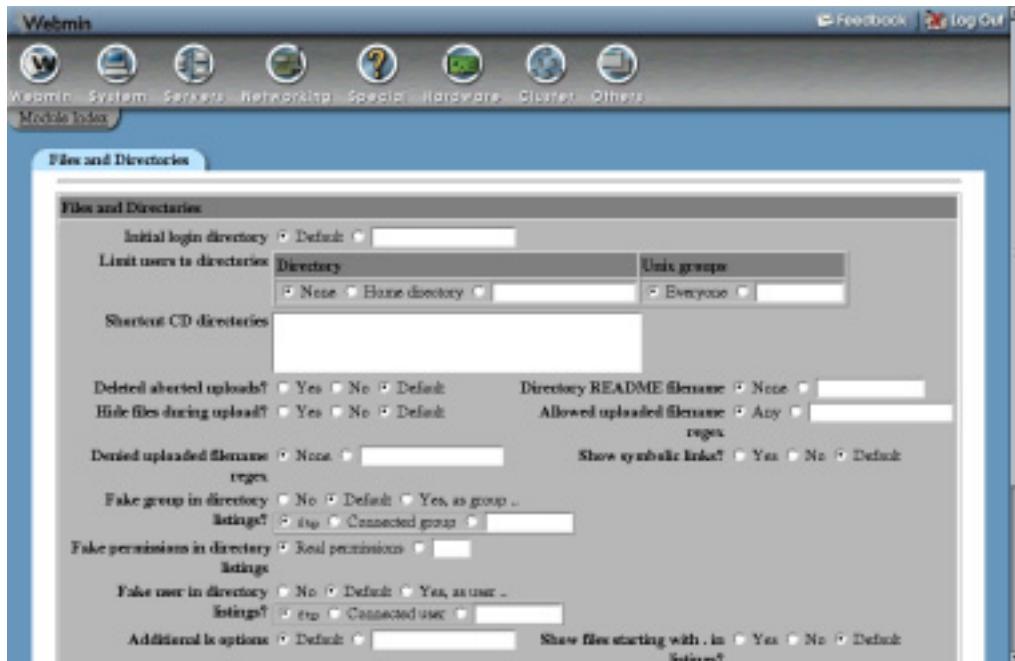


Figure 40.3 The files and directories form.

ProFTPD also prevents users without a valid shell from logging in by default. A valid shell is one listed in the `/etc/shells` file. This feature can be useful for preventing a large group of users from logging in, such as those who should only be able to connect to a POP3 server to download their email. It can be turned off, however, if necessary.

To edit the list of denied users and other login restrictions, follow these steps:

1. On the module's main page, click on the **Denied FTP Users** icon. In the form that appears is a text box listing all blocked UNIX users. Edit it to add or remove any that should or should not be allowed to log in, and hit the **Save** button.
2. To allow the `root` user to connect, click on the **Authentication** icon and change the **Allow login by root?** field to **Yes**.
3. To allow users with unlisted shells to log in, change the **Only allow login by users with valid shell?** field to **Yes** as well.
4. Hit the **Save** button to return to the main page, then click **Apply Changes** to make the new restrictions active.

The options for allowing the `root` user and users with invalid shells to log in can also be set on a per-virtual server basis as well, under the **Authentication** icon on the virtual server options page. It is not generally useful from a security point of view, however, to allow clients of just a single server to log in, as users can choose any server to connect to.

40.9 Setting Directory Listing Options

Normally, when an FTP client requests a directory listing, ProFTPD will return a complete, accurate list in the format produced by the `ls -l` command. Sometimes this gives away too much information about your system, such as the names of users and groups or symbolic link destinations. It can often be useful to hide certain files that are not relevant to clients but must be kept in an FTP-accessible directory for other reasons. This kind of information hiding is best applied to anonymous FTP users, as they should not be able to discover anything about your system that they do not need to know.

To change the format of directory listings, follow these steps:

1. On the module's main page, click on the icon for the default or virtual server for which you want to change directory listings to bring up its options page.
2. Assuming that you only want to change the listed information for anonymous clients, click on the **Anonymous FTP** icon to go to the anonymous FTP options page. Otherwise normal UNIX users will be affected as well.
3. Click on the **Files and Directories** icon to bring up a form similar to the one in Figure 40.3 for setting the various listing options.
4. To hide files with certain group owners, enter one or more group names, separated by spaces, into the **Hide files owned by groups** field. Be aware that files hidden in this way can still be downloaded, renamed, or deleted unless UNIX permissions or the server's configuration prevents it.
5. Similarly, to hide files with certain user ownership, fill in the **Hide files owned by users** field with a list of UNIX usernames.
6. To hide files that the anonymous FTP user would not be able to read, change the **Hide files that cannot be accessed?** field to **Yes**.
7. To have ProFTPD convert symbolic links in listings to their target file permissions and size, change the **Show symbolic links?** field to **Yes**. Normally both the link and target name are shown, and the displayed permissions and ownership are those of the link. Even with this feature enabled, however, the link target must still be within the anonymous FTP directory.
8. Normally, directory listings include the real user and group owners of files. To change this, set the **Fake group in directory listings?** field to **Yes, as group**. Then, from the box provided, select either **ftp** to force the group owner to be always shown as **ftp**, or the third radio button to have it shown as whatever group you entered into the adjacent text box. The **Connected group** option only really makes sense for non-anonymous clients, as it makes files appear to be owned by the primary group of the connected user.
9. You can also change the UNIX user owner of files with the **Fake user in directory listings?** field. If **Connected user** is chosen, files will appear to be owned by the user currently logged into the FTP server.
10. By default, ProFTPD will show real UNIX file permissions in listings. To force the display of fakes instead, select the second option in the **Fake permissions in directory listings** field and enter an octal number like **0644** of the kind used by the `chmod` command. This has no effect on the actual permissions that apply if a client tries to download or upload a file of course.

11. To hide dot files like `.login` and `.profile` in listings (as the `ls` command usually does), set the **Show files starting with `.` in listings?** field to **Yes**.
12. Finally, hit the **Save** button at the bottom of the page to update the ProFTPD configuration file.
13. Return to the module's main page and press the **Apply Changes** button to make the settings active.

As well as hiding certain files (as explained in Steps 4 and 5), you can also prevent clients from reading or writing those files altogether. This can be done using the **Make hidden files inaccessible?** field, explained in Section 40.12 “Restricting Access to FTP Commands”.

40.10 Message and Readme Files

ProFTPD can be configured to display messages to clients when they log in or enter certain directories. This can be useful for notifying users of possible mirror sites, the locations of various common files on the server, and the details of the contents of a directory.

To set the messages that are displayed to clients, follow these steps:

1. If you want the messages to be used by all virtual servers, click on the **Authentication** icon on the module's main page. To set messages for a specific virtual server, click on its icon and then on **Authentication** on the server options page. Either way, the same form will be displayed.
It is also possible to set most of the message file options below for only anonymous clients by clicking on the **Anonymous FTP** icon on the virtual server page and then on **Authentication**. Naturally, you cannot set the pre-login message because the server does not know if a client is anonymous or not at that stage.
2. In the **Pre-login message file** field, enter the full path to a file whose contents should be sent to clients as soon as they connect. If you don't want any message file to be used at all, select **None** instead.
3. In the **Post-login message** file field, enter the path to a text file whose contents will be sent to clients after they have been properly authenticated. If the client is limited to a directory (because it logged in anonymously or has a home directory restriction in force), the file must be within and relative to that directory. If the filename is relative (like `welcome.txt`), it will be searched for in the directory in which the client is initially placed.
4. To set a message sent to clients when they request to disconnect, fill in the **Logout message file** field. Again, this must be relative to and under any directory in which the client is restricted.
5. If you have a restriction on the maximum number of simultaneous logins in force, you can set the message sent to clients blocked by it by filling in the **Too many connections message file** field. You should enter a full path, which can be anywhere on your system. See Section 40.14 “Limiting Concurrent Logins” for more details.
6. Hit the **Save** button at the bottom of the page to go back to the global, virtual server or anonymous FTP options page.
7. Click on the **Files and Directories** icon on the same page.
8. In the **Directory README filename** field, enter a relative name like `readme.txt` that will be searched for in each directory that a client enters. If this is the first time the client

has entered the directory in this session (or if the file has changed since the last time), its contents will be sent to the FTP client.

9. To have the server send a message to clients suggesting that a particular file should be read, fill in the **Notify user of readme files matching** field. If files in the directory matching the specified regular expression (like *README.**) exist, a short message containing their names and modification times will be sent.
10. Click the **Save** button on this form, then return to the module's main page. Finally, click the **Apply Changes** button to activate the new message file settings.

The files sent to the client by the options covered above can contain certain special cookies that start with a %, which are replaced by ProFTPD with text determined at the time of sending. According to the ProFTPD documentation, the currently supported cookies are listed in Table 40.1.

Not all may make sense in all situations though. For example, %U will not be set in the pre-login message file.

Table 40.1 Cookies Usable in Files

%T	Current date and time
%F	Available space on file system
%C	Current working directory
%R	Remote host name
%L	Local host name
%u	Remote username reported by ident protocol
%U	Username originally used in login
%M	Max number of connections
%N	Current number of connections
%E	Server administrator's email address
%x	The name of the user's class
%y	Current number of connections from the user's class
%z	Maximum number of connections from the user's class

40.11 Setting Per-Directory Options

The ProFTPD module allows you to set options that apply only to a specific directory, rather than globally or to an entire virtual server. This allows you to do things like hide a directory from clients, allow uploads by anonymous clients in just one location, or set the user and group ownership of files added to a directory.

To create a new set of per-directory options, follow these steps:

1. If you want the options to apply to all virtual servers, enter the directory into the **Directory path** field in the **Add per-directory options for** form on the module's main page and hit the **Create** button. Alternately, you can limit them to a particular virtual server by clicking on its icon and using the same form on the virtual server options page. You can also define options that only apply to anonymous clients by hitting the **Anonymous FTP** icon for a virtual server and using its directory options creation form.
In all cases, the directory should be entered as an absolute path like `/usr/local`. It is also possible to specify a path relative to the connecting user's home directory, like `~/public_html`. You can even enter a path in a particular user's home directory, like `~jcameron/www`. Normally, the options will apply to the directory and all its contents and subdirectories. To have them apply to only the contents and not the directory itself, add `/*` to the end of the path that you enter, like `/usr/local/*`.
2. After hitting **Create**, you will be taken to a page of option category icons for the directory, as shown in Figure 40.4. As usual, clicking on these icons will take you to forms for configuring various settings that apply only to requests for, and listings of, that directory.
3. To totally deny access to clients, click **Access Control** and change the **Access control policy** field to **Deny all clients**, then click **Save**.
4. Normally, files uploaded by clients will end up owned by the UNIX user as whom the client logged in. To change this, click on the **User and Group** icon and enter a username for the **Owner of uploaded files** field. The uploaded files' group will be the primary group of the specified user, unless you fill in the **Group owner of uploaded files** field as well. Again, click **Save** after making any changes to return to the per-directory options page.
5. To limit only the uploading or downloading of files in this directory, you will need to create a set of per-command options under it. Section 40.12 “Restricting Access to FTP Commands” explains how.
6. To activate your changes for this directory, return to the module's main page and hit the **Apply Changes** button.
7. Once a set of options for a directory has been created, an icon for the directory will appear on the main, virtual server options, or anonymous FTP page—depending on where you created it. You can click on this icon to bring up the Per-directory Options page again and use the icons and forms to make any changes that you wish. It is also possible to change the path to which the options apply by clicking on the **Configure Directory** icon and updating the **Directory path** field on the form that appears. Then, hit the **Save** button followed by the **Apply Changes** button back on the main page. All the options set for the old directory will now apply to the new.

You can also remove a directory options object from the ProFTPD configuration entirely by clicking on **Configure Directory** and then hitting the **Delete directory config** button. All settings and per-command options for the directory will be immediately and permanently deleted from the FTP server's configuration.

If you define options for both a directory and one of its children (such as `/usr/local` and `/usr/local/bin`), ProFTPD will always give precedence to the most specific directory when deciding which options to apply to a particular client request. This means that a setting

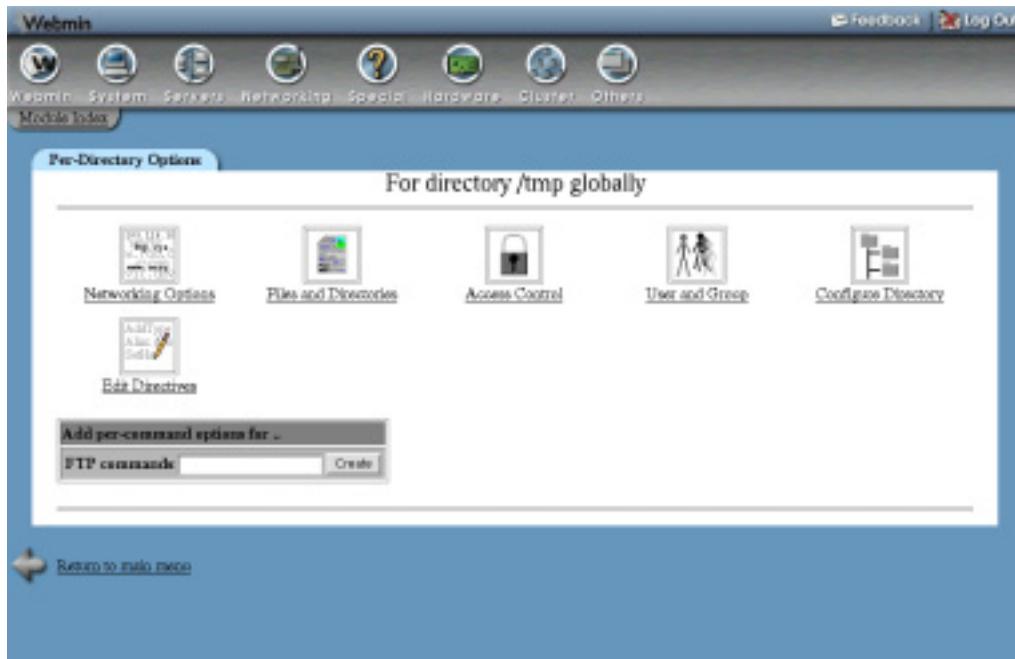


Figure 40.4 The per-directory options page.

made for */usr/local* will apply to a download of */usr/local/bin/foo*, unless it is overridden by a setting for */usr/local/bin*.

40.12 Restricting Access to FTP Commands

When a client wants to download or upload a file, list a directory, or perform any other operation, it sends a command to the server. ProFTPD can be configured to restrict which commands a client can use for a particular virtual server or directory, or when logged in anonymously. However, before you can do this, you need to have a basic understanding of which FTP commands exist and what they do. Table 40.2 lists the ones that are relevant for access control purposes.

ProFTPD allows you to define options that only apply to particular client commands or groups of commands. Typically, this is used to deny access to certain operations, such as uploading by anonymous FTP users. It is also possible to allow or deny only certain UNIX users, or only clients connecting from certain addresses.

To create a new set of per-command options, follow these steps:

1. First decide if the options should apply to commands only in a particular directory, only to clients of a virtual server, only to anonymous clients, or to all users of your FTP server. On the per-directory, virtual server, anonymous FTP, and main pages is a form titled **Add per-command options for**. In the **FTP commands** field, enter one or more commands from the list above, separated by spaces. When you hit the **Create** button, your browser will be taken to the page shown in Figure 40.5.

Table 40.2 FTP Commands and Their Purposes

CWD	Change to a different current working directory (Like the UNIX cd command)
MKD	Create a new empty directory (Like the UNIX mkdir command)
RNFR	Rename an existing file or directory (Like the mv command)
DELE	Delete a file (Like the rm command)
RMD	Delete a directory, which must be empty (Like the rmdir command)
RETR	Download a file from the server to client
STOR	Upload a file from the client to server
SITE_CHMOD	Change the UNIX permissions on a file
READ	This is not a command, but a shorthand for all FTP commands that deal with file reading
WRITE	Again, this is not a command but a shorthand for all commands for writing or modifying files
DIRS	This is shorthand for all directory listing and movement commands
LOGIN	This one is not really an FTP command at all, but is used to represent client connections. See Section 40.15 “Restricting Clients by IP Address” for information on how to use it
ALL	Represents all FTP commands

2. Click on the **Access Control** icon to bring up a form for restricting who can use these commands.
3. To completely deny access to everyone, change the **Access control policy** field to **Deny all clients**. To allow access to everyone, select **Allow all clients** instead. This is most useful if you are editing options for commands within a directory and there is a set of options for the same commands at a higher level (such as for the virtual server or anonymous FTP) that denies access. For example, anonymous clients cannot typically use the WRITE commands, but you may want to allow it for a particular directory.
4. To only allow certain UNIX users or members of a certain group access to the commands, fill in the **Only allow users** and **Only allow group** fields. Multiple user or group names must be entered separated by spaces.
5. Certain users and groups can be denied while allowing everyone else access to the FTP commands, by filling in the **Deny users** and **Deny groups** fields.
6. The **Restrict access** table can be used to block clients from certain IP addresses by entering a series of rules. The three radio buttons at the top control the order in which entries in the table are evaluated. If **Deny then allow** is selected, any client that matches a Deny

row or that does not match an Allow row will be blocked. Conversely, if **Allow then deny** is chosen only clients that match a Deny row and do not match an Allow row will be prevented from using the commands. This mode is also the default.

The table will always have one empty row for adding a new rule, and because this is a new set of per-command options, that is all it will contain. In the empty row, select either **Allow** or **Deny** from the **Action** menu. Then from the **Condition** menu, choose one of the following to determine which clients match and thus are allowed or denied:

All All clients match, no matter where they are from.

None No clients match the rule.

IP address Only clients from the IP address entered in the adjacent text field match.

Network Only clients from the entered IP network match. The network address must be a partial IP with a trailing dot, like *192.168.1*.

Hostname Only clients whose IP address reverse resolves to the entered name match. You can specify an entire domain by putting a dot at the front, like *.example.com*.

If you want to add more than one rule, you will need to re-enter this page after saving so that a new blank row appears. To delete a rule, select the blank option from the **Action** menu.

- When you are done choosing who can use the FTP commands, hit the **Save** button. Then, return to the module's main page and click **Apply Changes** to make the restrictions active.

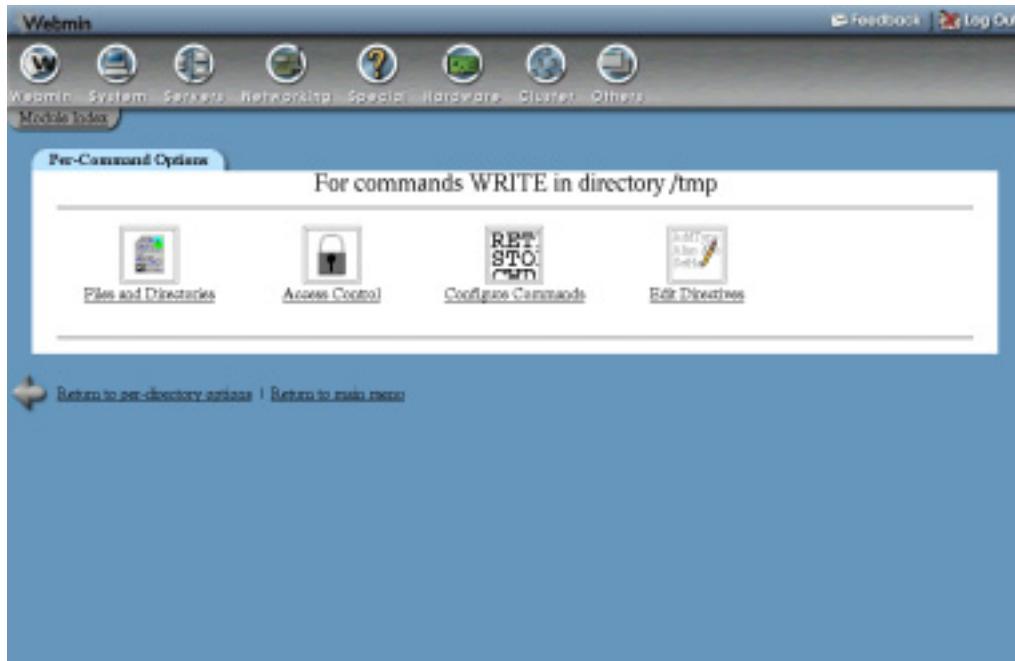


Figure 40.5 The per-command options page.

8. Once a set of options for a command or commands has been created, an icon for them will appear on the main page, virtual server options page, anonymous FTP page, or directory options page, depending on where you created it. You can click on this icon to bring up the same per-command options page again and use the icons and forms to make any changes that you wish. It is also possible to change the commands that the options apply to by clicking on the **Configure Commands** icon and selecting different entries from the **FTP commands** list on the form that appears. Then hit the **Save** button, then the **Apply Changes** button back on the module's main page. Alternately, you can click on **Delete commands config** to remove the options for these commands from the configuration entirely.

40.13 Configuring Logging

By default, ProFTPD logs all transfers to the file `/var/log/xferlog` in the standard FTP logging format (unless a different path has been selected at compile time). You can configure the server to log transfers to and from each virtual server differently, however, and anonymous FTP traffic as well. This is most useful in a virtual hosting environment in which your system hosts FTP sites for many different customers.

It is also possible to define additional log files that use different formats, and optionally include only a subset of FTP commands. This can be useful if you only care about uploads and don't want your log files clogged up with useless information.

To configure where and how logs are written globally or for an individual virtual server, follow these steps:

1. If you want to change the location of the global log file that is used for all transfers (unless overridden by a virtual server), click on the **Logging** icon on the main page.
If you want to configure a specific virtual server to use a different log file, click on its icon and then on the **Logging** icon on the virtual server options page.
To change the logging settings for anonymous clients only, click on a virtual server icon, then on **Anonymous FTP** and finally on the **Logging** icon on the anonymous FTP options page.
2. On the resulting logging options form, the **FTP transfers logfile** field controls where logs are written to. To specify a file, select the last option and enter a full path like `/var/log/example.com.xfers`, into the adjacent text field. To turn off logging altogether, select **Logging disabled**. To use the global default, select the **Default** option. If you are editing the global logging settings, ProFTPD will use the compiled-in default log file `/var/log/xferlog`.
3. The **Custom logfiles** table can be used to define additional logs for specific commands and with arbitrary formats. As usual, it will always have one empty row for adding a new custom log file. To add one, fill in the fields under these headings:

LogFile The full path to the log file, such as `/home/example.com/ftplog`.

For FTP commands If **All** is selected, all FTP commands will be logged. If you choose the second option, however, only those command classes in the adjacent text box will be included. Recognized classes are **NONE** (no commands), **ALL** (all commands), **INFO** (information requests), **DIRS** (directory navigation), **READ** (file download),

WRITE (file upload and directory creation), SITE (non-standard commands like CHMOD), and MISC (other miscellaneous commands). Multiple classes must be separated by commas, like *READ, WRITE*. You cannot use the names documented in Section 40.12 “Restricting Access to FTP Commands”.

Log format If **Default** is selected, the standard FTP log format will be used. But if the second option is chosen, you must enter a recognized log format name into the text box. The next paragraph explains how to set up named log formats.

Because only one empty row appears in the table, you can only add one custom log at a time. To add more, click on the **Logging** icon again after saving and fill in the new blank row. To delete a custom log, just clear out its field in the **LogFile** column.

4. Click the **Save** button to save the new settings, and then click **Apply Changes** on the main page to activate them.

If you want to use your own custom formats for log files, they must first be defined globally. To create a format, complete the following steps:

1. On the module’s main page, click on the **Logging** icon to bring up the global log file options page.
2. The **Custom log formats** table is for defining your own formats. In the first blank field under **Format name**, enter a short name for your new format such as *filesonly*. In the field next to it, under **Format string**, enter text containing the log codes recognized by ProFTPD, like *Downloaded %f at %t*. The special codes in the string starting with % are replaced by the server with information about the command, as explained in Table 40.3. As usual, you can add more than one custom format by re-entering the page after saving so that a new blank row appears. A format can be deleted by just clearing out its **Format name** field.
3. Click the **Save** button to return to the main page, and then click **Apply Changes**. The new format can now be used in custom log files.

According to the ProFTPD documentation, the recognized log format codes are those shown in Table 40.3.

Table 40.3 Logging Format Codes

%a	Remote client IP address
%A	Anonymous password, or UNKNOWN for non-anonymous clients
%b	Bytes sent for request
%{NAME}e	Contents of environment variable NAME. Note that the server does not set any environment variables itself.
%f	The absolute filename stored or retrieved

Table 40.3 Logging Format Codes (Continued)

%F	The filename stored or retrieved, as the client sees it
%h	Remote client DNS name
%l	Remote username (from <code>ident</code>), or UNKNOWN if none is available
%L	Local server IP address
%m	FTP command name received from client, such as RETR
%p	Local server port number
%P	Local server process ID
%r	Full command line received from client
%s	Numeric FTP response code
%t	Current local time
%{FORMAT}t	Current local time formatted by the standard UNIX <code>strftime</code> function using FORMAT
%T	Time taken to transmit or receive the file, in seconds
%u	Local authenticated username as which the client logged in
%v	Name of the virtual server that the client is connected to, from the ProFTPD configuration
%V	DNS name the virtual server that the client is connected to

40.14 Limiting Concurrent Logins

If your system is configured to allow anonymous FTP logins and you expect to receive a lot of traffic, it makes sense to limit the number of connections that can be open to the FTP server at any one time. This puts a ceiling on the network and CPU load that FTP transfers can generate, which is important if the system is being used for some other purpose (such as running a web server).

This limit can be set globally, on a per-virtual server basis, or just for anonymous clients. This means that you can set a limit that applies to all servers, and then increase or decrease it for a particular virtual host. You can also set a lower limit for anonymous clients versus those that have valid logins.

ProFTPD can also be configured to limit the number of concurrent connections that a single client host can have. This is useful if you want to stop people from downloading more than one file at a time from your server, and thus taking more than their fair share of bandwidth.

To set a connection limit for your server, follow these steps:

1. If you want to set a global limit, click on the **Networking Options** icon on the module's main page.

To set a limit for a single virtual server, click on its icon and then on **Networking Options**.

To define a limit that applies only to anonymous clients, click on the icon for a virtual server, then click on **Anonymous FTP**, and finally click the **Networking Options** icon on the anonymous FTP options page.

2. On the form that appears, find the **Maximum concurrent logins** field. To set a limit, select the third radio button and enter a number in the text box next to it. Alternately, you can select **Unlimited** to turn off any restriction that applies to this virtual server that has been set globally.

3. To define an error message sent to clients that try to connect when the limit has been reached, enter it into the **Login error message** box in the **Maximum concurrent logins** field. If the message contains the special code %m, it will be replaced with the maximum allowed number.

4. To set the per-client host limit, fill in the **Maximum concurrent logins per host** field in the same way. It also has a **Login error message** box that can be used to set a message that is sent to FTP clients that exceed the limit.

5. If you are editing the global networking options, you can also set a limit on the total number of ProFTPD subprocesses that can be active at any one time. This is useful for protecting your system from denial-of-service using hundreds of useless connections. Just select the second option for the **Maximum concurrent sessions** field and enter a number into its adjacent text box. If **Default** is selected, no limit will be enforced.

If you are running the server from a super server like `inetd` or `xinetd`, this limit will have no effect. Fortunately, both those servers have configuration options that can be used to achieve the same result.

6. When you are done editing client restrictions, click the **Save** button at the bottom of the form to update the ProFTPD configuration, and then the **Apply Changes** button back on the main page.

40.15 Restricting Clients by IP Address

By default, ProFTPD will allow clients to connect from any IP address. Like everything else, however, this is configurable so that you can restrict access to systems on your own network—either globally or for particular virtual servers. This comes in handy if you are setting up an FTP server that is for internal use only, even though the system it is running on is accessible from the Internet.

To restrict clients by address, follow these steps:

1. To create a global restriction that will apply to all virtual servers, enter *LOGIN* in the **FTP commands** field of the per-command options form on the module's main page, then click **Create**. If you only want to limit who can connect to a particular virtual server, click on its icon before entering *LOGIN* into the same form on the virtual server options page.
2. Regardless of at what level the restriction is being defined, you will be taken to the per-command options page shown in Figure 40.5. Click on the **Access Control** icon to go to the aptly named **Access control** form.
3. The **Restrict access** table can be used to block clients from certain IP addresses by entering a series of rules. The three radio buttons at the top control the order in which entries in the table are evaluated. If **Deny then allow** is selected, any client that matches a Deny row or which does not match an Allow row will be blocked. Conversely, if **Allow then deny** is chosen, only clients that match a Deny row and do not match an Allow will be prevented from logging in. This mode is also the default.

The table will always have one empty row for adding a new rule, and because this is a new set of per-commands options, that is all it will initially contain. In the empty row, select either **Allow** or **Deny** from the **Action** menu. Then, from the **Condition** menu, choose one of the following to determine which clients match and thus are allowed or denied.

All All clients match, no matter where they are from.

None No clients match the rule.

IP address Only clients from the IP address entered in the adjacent text field match.

Network Only clients from the entered IP network match. The network address must be a partial IP with a trailing dot, like *192.168.1*.

Hostname Only clients whose IP address reverse-resolves to the entered name match. You can specify an entire domain by putting a dot at the front, like *.example.com*.

If you want to add more than one rule, you will need to re-enter this page after saving so that a new blank row appears. To delete a rule, select the blank option from the **Action** menu.

4. When you are finished entering client restrictions, click the **Save** button at the bottom of the form. Then return to the main page and click **Save and Apply** to activate them.

You will generally want to give access only to clients on a single network. To do this, select the **Deny then allow** option, choose **Allow** from the **Action menu**, choose **Network** from the **Condition** menu, and enter the network address with a trailing dot (like *10.254.1.*) into the condition text box.

40.16 Limiting Uploads

If clients are allowed to upload files to your server, they will be able to choose any name that they wish for uploaded files. Sometimes this is not desirable; however, you may want to allow the storing of only image files whose names end with *.gif* or *.jpg*, or prevent the uploading of

Windows executables with filenames ending in .exe or .com. Fortunately, ProFTPD has configuration options that allow you to set this up.

There are also several other settings that apply to uploads that can control whether clients are allowed to overwrite files and whether partially transferred files are visible. All can be set globally for a single virtual server or for anonymous clients only. To set these options, complete the following steps:

1. If you want the settings to be global, click on the **Files and Directories** icon on the module's main page. To have them apply to just a single virtual server, click on its icon and then on **Files and Directories**. Or, to effect just clients that log in anonymously, click on a virtual server icon, then on **Anonymous FTP**, and finally on the **Files and Directories** icon on the virtual server options page.

No matter which configuration object you chose, the files and directories form that appears will be almost identical.

2. To hide files that are in the process of being uploaded, change the **Hide files during upload?** field to **Yes**. This tells ProFTPD to use a temporary file whose name starts with .in. for transferred data, which is only renamed to the real filename when the upload is complete. This prevents incomplete uploads, and stops files from being downloaded or accessed while they are still being sent.
3. To have ProFTPD delete uploaded files that are not fully transferred, select **Yes** for the **Delete aborted uploads?** field. Again, this prevents corrupt, partially uploaded files from being created on your system.
4. To allow users to only create files whose relative names match a certain pattern, fill in the **Allowed uploaded filename regex** field with a Perl regular expression. For example, to allow only GIF files, you might enter ^.*\.gif\$.

Because clients are normally allowed to rename files, this option alone is not enough to stop the creation of invalid filenames. You will also need to block access to the RNFR command, as explained in Section 40.12 “Restricting Access to FTP Commands”.

5. You can also block the use of certain filenames by filling in the **Denied uploaded filename regex** field with a regular expression like ^.*\.exe\$. If both this and the previous field are set, only files that match the allow expression, but not this deny expression, will be permitted.

Another common use of this option is blocking the upload of .ftpaccess or .htaccess files, which set per-directory ProFTPD and Apache options.

6. Hit the **Save** button at the bottom of the page.
7. If you want to stop clients overwriting files with new uploads, click on the **Access Control** icon and change the **Allow overwriting of files?** field to **No**. This can be useful on a server that allows anonymous users to upload to a particular directory, perhaps for incoming files of some kind. Remember to click **Save** if you make this change.
8. Return to the module's main page and hit the **Apply Changes** button to activate your new filename restrictions.

40.17 Manually Editing Directives

If you prefer to manually edit your ProFTPD configuration file in some cases or just want to see which directives an action in Webmin has set, you can do so using this module. Except for the default server, every object's options page (virtual server, per-directory, and per-command) has an icon labeled **Edit Directives**. When clicked, it will take you to a form containing a large text box showing the lines from the configuration file in the section related to the object. You can edit them to your heart's content, then click the **Save** button to update the actual file. Be aware though that no validation of your input is done. Also, you will need to use the **Apply Changes** button on the module's main page to activate any changes, as usual.

To view and edit the entire ProFTPD configuration, use the **Edit Config Files** icon on the module's main page. This will bring up a similar form, but showing and allowing the editing of a complete configuration file at once. Because ProFTPD can read multiple configuration files (through the use of `Include` directives), at the top of the form is a button labeled **Edit Directives in File** with a menu of filenames next to it. To switch the view to a different file, just select the one you want and hit the button. Normally, though, only a single `proftpd.conf` file will be used.

40.18 Configuring the ProFTPD Server Module

To change the paths that the module uses for the ProFTPD configuration files and programs, you will need to click on the standard **Module Config** link in the top-left corner of the main page. Unlike other modules, there are no options related to the user interface, so it is rare that you will need to change anything once the module is working. By default, all the configuration fields are set to match the ProFTPD package available for your operating system—or if there is no such package, the fields match the paths used by an installation from the source distribution instead.

Table 40.4 Module Configuration Options

Path to ProFTPD config file	This field must contain the full path to the ProFTPD primary configuration file, which is usually <code>/etc/proftpd.conf</code> .
Path to ProFTPD executable	This field must contain the path to the ProFTPD server program, such as <code>/usr/sbin/proftpd</code> .
Path to ProFTPD PID file	This field must contain the path to its PID file, such as <code>/var/run/proftpd.pid</code> so the module can determine whether or not the FTP server is running.
Path to ftpusers file	This field must contain the path to the file that ProFTPD reads to find out which users are not allowed to log in. Normally, that is <code>/etc/ftpusers</code> , which is used by several FTP servers as well.
Command to start ProFTPD	If Automatic is selected when you click the Start Server button on the main page, the module will simply run the ProFTPD executable program. Alternatively, you can enter a command such as <code>/etc/init.d/proftpd start</code> , which will be used instead to do the same thing.

40.19 Summary

This chapter has introduced the FTP protocol and explained the purposes of FTP clients and servers. It has also covered the features of ProFTPD, and covered the steps that you need to take to install it. After reading the chapter, you should know how the FTP server can be configured to allow anonymous logins, to restrict clients from certain addresses, to limit users to their home directories, and to log in custom formats. You should also understand ProFTPD's powerful virtual FTP hosting support, which can be used to define different options depending on to which of your system's IP addresses a client connects.

The WU-FTPD Server

In this chapter, the WU-FTPD server for the FTP protocol is explained and the steps for configuring it for many common tasks are provided.

41.1 Introduction to WU-FTPD

WU-FTPD (the WU stands for Washington University) is probably the most popular UNIX FTP server on the Internet and is included by default with most UNIX and Linux operating systems. Its large number of configurable options make it superior to the “classic,” or BSD FTP, server that is still used by some flavors of UNIX, but it is not as flexible or cleanly implemented as ProFTPD, which is covered in Chapter 40. That chapter also has a brief general introduction to the FTP protocol, which you should read before going further if you are unfamiliar with concepts such as FTP clients and servers.

In its normal default configuration, WU-FTPD will allow any UNIX user (except for system users) to log in with their standard usernames and passwords and upload, download, and manipulate files on the server system with the same permissions that they would have if connected via telnet or SSH. It can also be set up to support anonymous logins, so that anyone can connect without needing a valid UNIX account, although anonymous clients are typically restricted to a certain directory and prevented from uploading files.

WU-FTPD’s primary configuration file is called `/etc/ftpaccess`, but it also makes use of several other files such as `/etc/ftpsusers` and `/etc/ftphosts`. The `ftpaccess` file contains a series of directives—one per line—each of which has a name and several values. Each directive sets a single option, such as the path to a message file or a directory alias.

Like ProFTPD, WU-FTPD can be run either as a permanent standalone daemon process or from a super server like `inetd` or `xinetd`. Typically the latter option is used, as this removes the need for an additional server process to be running at all times waiting for an FTP connec-

tion. As far as clients and the configuration file are concerned, there is no difference between either mode other than performance.

41.2 The WU-FTPD Server Module

To configure the FTP server from within Webmin, click on the **WU-FTPD Server** icon under the Servers category. If it is properly installed and working, the module's main page (as shown in Figure 41.1) will be displayed. Each of the icons will take you to a form for setting a class of configurable options, such as those related to logging or messages and banners.

If the module cannot find the WU-FTPD server executable, an error message like **The FTP server /usr/sbin/in.ftpd could not be found on your system** or **The FTP server configuration file /etc/ftpaccess does not exist** will be displayed. This usually means that the program is not installed. Check your Linux distribution CD or website for the *wu-ftpd* package and install it using the Software Packages module, covered in Chapter 12. It can also mean that the module is looking in the wrong location for the server program or configuration file. If you are certain that WU-FTPD is installed, see Section 41.14 “Configuring the WU-FTPD Server Module” for information on adjusting the paths in which Webmin looks for them.

Sometimes, even though WU-FTPD is installed, it will be disabled by default. If you run `ftp localhost` at the command prompt and get back the error message `Connection refused, no FTP server is running`. This may be because no `inetd` or `xinetd` service has been created for FTP yet, or because one exists but it is disabled. Follow the instructions below to set up or activate a super server service for WU-FTPD.



Figure 41.1 The WU-FTPD Server module.

If WU-FTPD really does need to be installed, you must first remove any other FTP server (such as ProFTPD or PureFTP) that is currently installed on your system. Make sure that you kill all of its processes as well, so that nothing is left listening on the FTP port. You can verify that the other server has been fully shut down by running `ftp localhost` at the command prompt. The error message `Connection refused` should be displayed, indicating that nothing is listening on port 21.

After installing WU-FTPD, you will need to configure `inetd` or `xinetd` to listen on the FTP port and run the server program. Before you can do this, find out where the program has actually been installed on your system. Typically it will be at `/usr/sbin/in.ftpd`, but this may differ depending on your operating system.

If your system uses the superior `xinetd` super server, use the following instructions to set up the FTP service. Because many packages include a `/etc/xinetd.d` configuration file for the server, some of the fields explained in the following list may be filled in correctly already.

1. Go to Webmin's Networking category and click on the **Extended Internet Services** icon. If it does not exist, `xinetd` is not installed and you will need to set up the server using `inetd` instead.
2. Check for an existing service named `ftp` or `wu-ftpd` on the module's main page. If one exists, click on it—otherwise, follow the **Create a new internet service** link above or below the table.
3. In the **Service name** field, enter `ftp` (unless it has already been filled in).
4. Make sure the **Yes** option is selected in the **Service enabled?** field.
5. Leave the **Bind to address** field set to **All**, and the **Port number** to **Standard** or **21**.
6. Select **Stream** from the **Socket type** menu and **Default** or **TCP** from the **Protocol** list.
7. In the **Service handled by** field, select the **Server program** option and enter the path to the WU-FTPD executable with the arguments `-l -a` (such as `/usr/sbin/in.ftpd -l -a`) into the adjacent text box.
8. In the **Run as user** field, enter `root`.
9. Select **No** for the **Wait until complete?** field.
10. Leave all the other fields set to their defaults, and hit the **Save** or **Create** button at the bottom of the form.
11. Back on the module's main page, click the **Apply Changes** button located below the list of services.

To set up an `inetd` service for WU-FTPD using the Internet Services and Protocols module, you can also follow these steps:

1. Go to Webmin's Networking category and click on the **Internet Services and Protocols** icon. If it does not exist, your system is probably using `xinetd`.
2. On the module's main page, click on `ftp` in the **Internet Services** table. If it is not visible, enter `ftp` in the field next to the **Edit service** button and hit the button. Either way, the same page for editing the FTP protocol service will be displayed.
3. In the **Server Program** section, select **Program enabled**.
4. In the **Program field**, select the **Command** option and enter the full path to the WU-FTPD server executable into the field next to it, such as `/usr/sbin/in.ftpd`. In the **Args** field, enter `in.ftpd -l -a`.

5. Set the **Wait mode** to **Don't wait** and enter *root* in the **Execute as User** field. All others can be left unchanged.
6. Click the **Save** button and then, back on the module's main page, hit **Apply Changes**.

Once WU-FTPD has been set up to run from *inetd* or *xinetd*, you can test it by using the command-line UNIX FTP client to connect to your own system. Just run `ftp localhost`, and make sure that you can log in as some user other than *root*. If not, check your log files (using the System Logs module covered in Chapter 13) for any message from *inetd* or *xinetd* that may explain what went wrong.

41.3 Limiting Who Can Log In

In its normal configuration, WU-FTPD will allow any UNIX user to log in with the exception of system accounts such as *root*, *bin*, and *daemon*. The *root* user is almost always denied by default because the FTP protocol does not encrypt passwords as they are sent over the network, which means that a remote login as *root* could expose its password to attackers.

To change the users and groups who can log in to your system, follow these steps:

1. On the module's main page, click on the **Users and Classes** icon to bring up the form shown in Figure 41.2.
2. Add to the **Unix users to deny** field any accounts that you want to prevent from using your FTP server, or remove those that you want to allow. This will update the `/etc/ftpaccess` file, which is used by other FTP servers such as ProFTPD (in case you decide to switch).
3. To deny users whose UIDs lie within a certain range, fill in the **Unix users and UIDs to deny** field. You can enter a UID range like `%3000-4000`, which will block all users with UIDs between 3000 or 4000. Or, you can enter ranges like `%-100` or `%5000-`, which will deny users with UIDs less than 100 or greater than 5000, respectively. Multiple ranges can be entered, separated by spaces. Normal usernames can be used in this field as well, although this has the same effect as putting them in the **Unix users to deny** field.
4. To deny users whose primary group IDs are within certain ranges, fill in the **Unix groups and GIDs to deny** field. Again, you can enter ID ranges like `%100-200` or `%-10`, as well as group names like *users*. Only primary group membership counts. If a user is a secondary member of one of the listed groups, he will not be blocked.
5. To exclude some users or groups from the deny lists defined in the previous two steps, fill in the **UNIX users and UIDs not to deny** and **UNIX groups and GIDs not to deny** fields. The first field will accept UID ranges or usernames, and the second accepts group ID ranges or group names. These fields are useful if you want to allow just a couple of users while blocking everyone else with a UID range that covers all accounts.
6. Hit the **Save** button at the bottom of the page to save and activate the new user restrictions.

WU-FTPD will also normally prevent users from logging in whose shell is not listed in the `/etc/shells` file. This is normally done to allow the creation of accounts that can log in to a POP3 server, but cannot connect via telnet, SSH, or FTP. Unfortunately, there is no WU-FTPD configuration option that can be changed to turn off this shell check. It is either hard-coded into the program or enforced by the `ftp` PAM service that is really used to authenticate users.

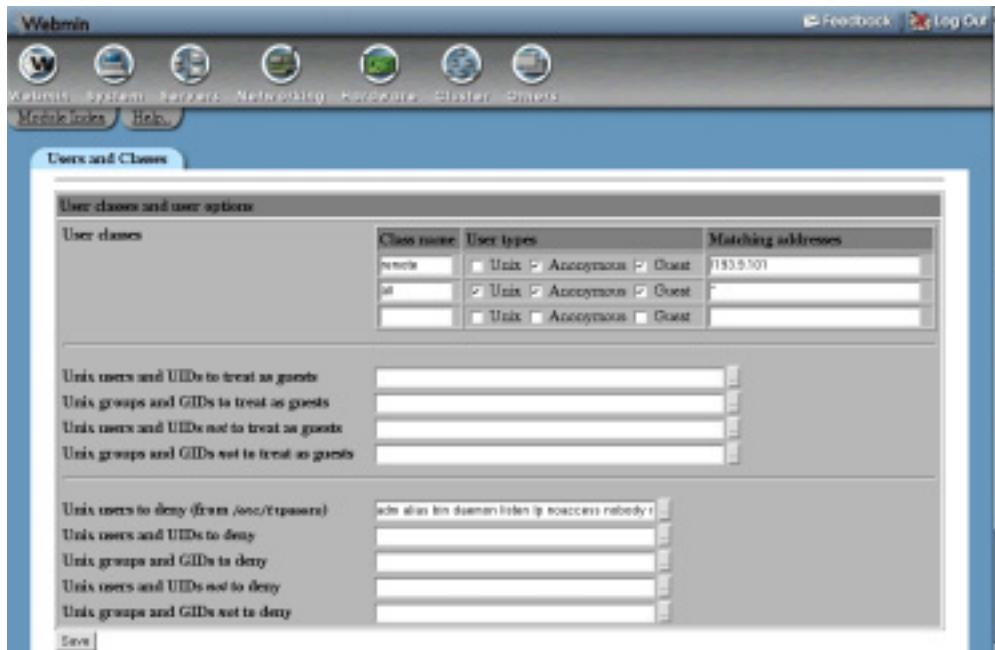


Figure 41.2 The users and classes page.

If WU-FTPD on your system uses PAM (as it does on most Linux distributions), follow these steps to turn off `/etc/shells` checking:

1. Go to the PAM Authentication module, which can be found under the **System** category on the Webmin main menu.
2. Click on the **ftp** or **wu-ftp** service on the main page.
3. On the editing form that appears, click on **pam_shells.so** in the **PAM module** column in the **Authentication steps** section.
4. From the **Failure level** menu, select **Optional** so the success or failure of the `shells` file check is ignored for authentication purposes.
5. Click the **Save** button. Users with an invalid shell will no longer be able to log in to your FTP server.

On other operating systems, the steps above are useless, as the PAM Authentication module is only available on Linux.

41.4 Setting Up Anonymous FTP

Configuring WU-FTPD to accept anonymous logins is slightly more complex than you would expect, due to its use of the `ls` command to generate directory listings. So that this command (and others used by server) cannot escape the directory to which anonymous clients are restricted, the server uses the UNIX `chroot` system call to restrict itself and all programs that it

runs to that directory. This means that the root directory must contain all the programs, files, and shared libraries that WU-FTPD and the `ls` command need to run.

By default, the home directory of the special UNIX user `ftp` is used as the anonymous FTP root, but different roots can be assigned to different client classes. Whatever directory is chosen, however, must contain a `bin` subdirectory with the `ls`, `gzip`, `tar`, `recompress`, `cpio`, and `zcat` commands. It must also have a `lib` subdirectory containing any shared libraries needed by those commands, an `etc` subdirectory with `passwd` and `group` files, and a `pub` directory in which downloadable files are stored.

As you can imagine, copying all these files into place and making sure that they work is quite tricky. Fortunately, many Linux distributions that include a `wu-ftpd` package also have a package named `anonftp` that places all the needed files in the home directory of the `ftp` user. In most cases, all you need to do is install this package and WU-FTPD will allow clients to log in anonymously.

Regardless of the permissions on the root directory, WU-FTPD will always prevent anonymous clients uploading, renaming, or deleting files. All they will be able to do is download the files that you place in the `pub` subdirectory for public distribution.

Anonymous logins can be further configured by following these steps:

1. Click on the **Anonymous FTP** icon on the module's main page.
2. The **Anonymous FTP root directories** table allows you to specify different root directories to be used for different classes of client. Any existing directories (apart from the default of `~ftp`) are listed in the table for editing, and there will always be one empty row for adding a new one. As soon as an entry is added, it will replace the default, so be sure to explicitly add it if you want it to continue working. If you want to add more than one directory, you will need to save and reopen this page so that a new blank row appears. Each row has two fields:

Directory In this field you must enter full path to a valid anonymous FTP directory (one that contains `etc`, `bin`, `lib`, and `pub` subdirectories and all the needed programs).

For class From this menu, you must choose a client class that the directory should be used for, assuming that clients in that class log in anonymously. If **Any** is selected, it will be used for clients not in any other class in this table. See Section 41.5 “Managing User Classes” for details on how to define your own classes.

3. When a user logs in to your FTP server anonymously, they must still supply a password even though it is not used for authentication. Typically this password is the user's email address, which can be used to get a rough idea of what domain clients are coming from. For privacy reasons, however, many modern FTP clients and browsers do not send a real email address anymore, logging in instead with a fake one like `mozilla@example.com`. You can configure WU-FTPD to check the format of anonymous login passwords to make sure that they look like email addresses using the **Anonymous FTP password check** field on this page. If **Default** is selected, no checking will be done. If the second option is chosen, however, the level of checking depends on the choice that you make from its menu:

Allow anything Any password is allowed, even a blank one (this is the same as the default mode).

Must contain @ The password must contain the @ symbol.

Must be an RFC882 email address The password must look like a valid email address, with letters and numbers before and after the @. The second menu determines whether the FTP server just warns clients that violate the check (if **Warn only** is chosen), or blocks them altogether (if **Deny login** is selected).

4. To block certain anonymous passwords altogether (even if they are valid), fill in the **Anonymous FTP passwords to deny** field with a list of complete or partial email addresses. This can be useful for blocking FTP clients that are configured by default to use a fake address. I recommend against using this feature, however, as it will block a lot of people, especially those using web browsers.
5. Hit the **Save** button at the bottom of the page to activate the new anonymous FTP settings.

41.5 Managing User Classes

The FTP server categorizes clients into classes based on their source addresses and types of login. Classification can be used in several different places in the WU-FTPD configuration to define settings that apply only to certain clients. It can also be used to block non-anonymous logins (or even all logins) from outside your network. This can be useful if you only want to allow certain trusted hosts to upload data to your server and let anyone on the Internet log in anonymously to download.

Each class has a name, a list of login types, and a list of client addresses, hostnames, or networks. Only clients that match both the login types and the addresses are considered to be in the class. If more than one class is matched, the first one is used. Clients that do not fall into any class are not allowed to use the FTP server.

The following three types of logins are recognized by WU-FTPD:

UNIX Normal UNIX users can log in via telnet or SSH and access all files on the system with their regular permissions.

Guest UNIX users who have been designated as guests. These are limited to a directory (usually their home) in the same way that anonymous users are. See Section 41.7 “Setting Up Guest Users” for more details.

Anonymous Users who log in anonymously are limited to a certain directory.

To define and edit classes using the module, follow these instructions:

1. Click on the **Users and Classes** icon in the top-left corner of the module’s main page. The form shown in Figure 41.2 will appear in your browser.
2. At the top of the page is a table labeled **User classes**. Each row defines a class and there will always be at least one listed already (typically the `all` class, which matches all clients). The table always has a single empty row at the bottom for you to add a new class. If you want to add more than one, you will need to create them one at a time. You can edit existing classes by changing their fields, or delete a class by clearing out its name field. Make sure you don’t delete them all, though, as this will prevent all users from logging in. The fields for each class are:

Class name A short name for this class that should consist of only letters and numbers, such as `homenet` or `trusted`. More than one row can have the same class name, and a

client that matches the user type and addresses in any row will be considered a member of the class.

User types The types of login that this class matches, as explained above. You must select at least one of the three checkboxes.

Matching addresses This field is where you get to enter the client addresses that the class matches. You can enter single IPs (like *192.168.1.1*), hostnames (like *www.foo.com*), wildcard IPs and hostnames (like *10.254.1.** or **.example.com*), or even paths to files containing more such addresses and hostnames. Multiple entries must be separated by spaces. Negated entries like *!*foo.com* are even allowed, which would match all clients whose hostnames are outside the *foo.com* domain. Be careful when using hostnames, as WU-FTPD must look up clients' hostnames from their IP addresses, the result of which can be faked by an attacker.

3. When you are done defining classes, hit the **Save** button at the bottom of the form. You can now use them in other pages in the module.

41.6 Denying Access to Files

Sometimes it is useful to restrict the types of files that users can download, especially for untrusted anonymous clients. You can block access to a filename in any directory (like *secret.txt*), an absolute path (like */etc/passwd*) or even a directory and all its contents (like */var/log*). The shell wildcards characters * and ? can be used in file and path names as well, which provides extra flexibility. This can be useful if you want to protect files containing secret information, or limit clients to downloading from a certain directory (like */home*). There is no way, however, to prevent the listing of directories using this feature.

To set up filename download restrictions, follow these steps:

1. On the main page, click on the **Limits and Access Control** icon to bring up the form shown in Figure 41.3.
2. Each row in the **Deny access to files** table defines a single filename restriction. As with other tables in this module, at the bottom of the table is a single blank row for adding a new filename or path and, if this is the first time you have used this feature, a single row is all the table will contain. Otherwise, existing restrictions will be listed, allowing you to edit or delete them.

The fields in each row and their meanings are:

Files to deny A list of relative or absolute filenames or patterns to which to deny access, separated by spaces. The wildcard characters * and ? can be used for both, allowing you to enter files like *secret.** or */home/*public_html*.

Relative to chroot? If **Yes** is selected, any absolute path entered in the first field is considered to be relative to the anonymous FTP root directory. If **No** is chosen, paths are taken to be relative to the real root directory.

Deny for classes In this field you must select the checkboxes for classes to which the restriction applies. See Section 41.5 “Managing User Classes” for more information on how to add classes of your own. This can be useful if you want to block access by anonymous clients to some file, but allow real UNIX users.

3. The **Allow access to files even if denied** table has exactly the same structure as the denial table, but is for entering filenames and paths to which access should be allowed even if they are denied by an entry in the table above. This can be used to deny access to everything (by entering * in a **Files to deny** row) and then granting back download rights on only files matching a pattern (like *.html for example).
4. Click the **Save** button at the bottom of the page to save and activate any new file restrictions. If you want to add more than one entry in either table, click on the **Limits and Access Control** icon again to redisplay the form and fill in the new empty rows that appear.

Messages, banners and README files			
Message files	Path	When to display	Classes to display for
	info/welcome	<input checked="" type="radio"/> At login <input type="radio"/> Entering any dir <input type="radio"/> Entering dir	
	info/message	<input type="radio"/> At login <input checked="" type="radio"/> Entering any dir <input type="radio"/> Entering dir	
	info/messages	<input type="radio"/> At login <input type="radio"/> Entering any dir <input type="radio"/> Entering dir	

README files			
README files	Path	When to display last modified date	Classes to display for
	README*	<input checked="" type="radio"/> At login <input type="radio"/> Entering any dir <input type="radio"/> Entering dir	
	README1	<input type="radio"/> At login <input checked="" type="radio"/> Entering any dir <input type="radio"/> Entering dir	
	README2	<input type="radio"/> At login <input type="radio"/> Entering any dir <input type="radio"/> Entering dir	

Greeting level: Hostname and version Hostname Neither

Pre-login banner: None Pre-login

Hostname for messages: System hostname Custom

Owner's email address: Default owner@ourcountybucket.com

Figure 41.3 The limits and access control form.

There is also a similar feature for restricting the names of files that clients can upload. This can be useful for blocking the creation of hidden files or directories whose names start with a dot, or hard to comprehend names containing spaces and control characters. These are often used by sneaky people to hide files on your anonymous FTP server if you allow uploading. Because trusted people may have good reasons for creating such files, you can define restrictions that apply only to anonymous or guest users.

To add and edit upload filename limits, follow these steps:

1. On the main page of the module, click on the **Permissions** icon.
2. In the form that appears, the **Denied upload filenames** table at the bottom lists filenames that are allowed and denied for different types of users. Existing restrictions can be edited by just changing their fields in the table, and a new one created by filling in the

final blank row (which will be all the table contains if you haven't used this form before). The columns in the table and the meanings of their fields are:

Allowed characters A single Perl regular expression that all uploaded files must match. For example, if you entered `^[a-z]+$`, only filenames made up of lower-case letters would be allowed.

File regexps to deny A space-separate list of regular expressions that are not allowed in filenames. A good example is `^\.`, which blocks any name starting with a dot, which hides the file.

User types The types of users to which this restriction applies. Often you will want to place stricter limits on anonymous clients than real or guest users.

Error message file The full path to a file that will be sent to any client which tries to upload a file whose name does not match the allowed expression or does match one of the denied expressions.

3. As usual, click the **Save** button at the bottom of the page to activate any new restrictions when you are done.

If more than one restriction is defined for the same type of user, they will all be checked to determine if an uploaded filename is allowed.

41.7 Setting Up Guest Users

A guest FTP user is a real UNIX user who is limited by WU-FTPD to a certain directory, just as anonymous clients are restricted. They still have full privileges within that directory, though, including the rights to upload files, rename, and `chmod` files. Limiting a user to guest access can be useful if you want to prevent him from seeing parts of your filesystem outside his home directory or some parent directory, like `/home`.

Every user who is designated as a guest by the FTP server configuration can have a different root directory, or some can be the same. The chosen root directories, however, must be set up in the same way as the anonymous FTP root is—with `bin`, `lib`, and `etc` subdirectories containing all the programs and files needed by WU-FTPD. You can just copy those directories across from the anonymous root, though, so the set up process is not that hard.

To set up a user as a guest, his home directory must be specially modified. To do this using Webmin, follow these steps:

1. Go to the Users and Groups module (covered in Chapter 4) and click on the name of the user that you want to restrict.
2. Change his home directory to `guestroot/.homendir`, in which `guestroot` is the root directory that you have prepared, and `homendir` is a subdirectory under it. If you are using `/home` as the root, `/home/./jcameron` could be the directory for the user `jcameron`. This special `./` entry in the path tells WU-FTPD where the root is, but should not confuse other programs.
3. Click the **Save** button at the bottom of the page. Webmin will move his home directory to the new location, if necessary.

Of course, you can specify such a home when creating a new user as well. This is only the first step, though. To configure WU-FTPD to treat certain users as guests, you will need to follow these steps:

1. In the WU-FTPD Server module, click on the **Users and Classes** icon to bring up the form shown in Figure 41.2.
2. In the **Unix users and UIDs to treat as guests**, enter a space-separated list of usernames, UIDs, or UID ranges (like %1000-2000 or %5000-) of users to be designated as guests. You can also enter a list of group names, IDs, and ID ranges in the **Unix groups and GIDs to treat as guests** field to have all their primary members treated as guests, as well.
3. To stop some users from being converted to guests even if they are in the lists or ranges set in Step 2, fill in the **Unix users and UIDs not to treat as guests** and **Unix groups and GIDs not to treat as guests** fields. This can be useful if you want to make everyone a guest except a few trusted users.
4. Click the **Save** button at the bottom of the page to activate the new guest designations.

If a user has been configured as a guest but does not have `/ . /` in his home directory, he will not be restricted to any root directory.

41.8 Editing Directory Aliases

To simplify life for users that frequently need to access directories with long paths, WU-FTPD allows you to define directory aliases that can be used when changing to a different directory. This means that someone using a command-line FTP client can enter `cd stuff:` instead of `cd /usr/local/etc/stuff`, assuming that a suitable alias has been created.

To set up aliases, follow these steps:

1. On the module's main page, click on the **Aliases and Paths** icon.
2. The **CD directory aliases** field is actually a table that lists existing aliases (if any) and always has one blank row for adding a new one. In the first empty field under the **Alias name** column, enter the name for a new alias like `stuff:` or `junk:`. There is no requirement that an alias end with a colon—however, this is a good idea as it reduces the risk of an alias name being the same as an actual relative directory.
In the corresponding field under **Alias to directory**, enter a full directory path—like `/usr/local/junk`—that the alias is equal to. If you want to add more than one alias, you will need to save and reopen this page so that a new empty row appears. To delete one, just clear out both of its fields in the table.
3. The **CD directory search path** text box lets you enter a list of directories that will be searched if a client tries to change to a relative subdirectory that is not in the current directory. For example, if `/usr/local` was included and a client tried to switch to the `bin` directory, he would be placed in `/usr/local/bin` (assuming no real `bin` subdirectory exists).
4. Hit the **Save** button at the bottom of the page to activate the new aliases.

Aliases are not particularly useful in graphical FTP clients that present the user with a directory listing on which to click. Because users do not (and often cannot) change to an explicit pseudo-

directory like `stuff:` and aliases are not included in listings, they are hard to use. Command-like FTP clients like `ncftp` and the classic UNIX `ftp` program are more suited to using them.

41.9 Message and Readme Files

WU-FTPD can be configured to send the contents of various message files to clients when they log in or enter certain directories. This can be useful to display information about your FTP server (such as who runs it and what files are hosted) or details of the contents of a particular directory to FTP users. Each file is only sent once to a client in a single session, to avoid annoying the user with repeated messages.

The server can also be set up to notify clients that certain files exist and inform them of their last modification dates. This is typically used for `README` files containing slightly less important information about a directory or the server, which users may want to read. Again, clients are only notified once per session for each such file.

To define message and banner files, follow these steps:

1. Click on the **Messages and Banners** icon on the module's main page, which will take you to the form shown in Figure 41.4.
2. The **Message files** section is a table for specifying files whose contents will be sent to clients. As usual with tables in Webmin, it lists existing files and their contexts, and has one blank row at the bottom for adding a new one. The meanings of the fields are:

Path The path to the file whose contents should be sent to the client. This can be either an absolute path like `/etc/login.message` or a relative filename like `message.txt`. In the latter case, it is looked for in each directory that the client enters. If you enter a full path and want anonymous clients to be able to see it, it must be under the anonymous FTP root directory.

When to display If **At login** is selected, the file will be sent to clients after logging in. If **Entering any dir** is chosen, the file will be searched for and sent when changing to any directory. When **Entering dir** is chosen, the file will only be sent when the directory whose path you specify in the adjacent text box is entered. Again, this must be relative to the root directory for anonymous clients.

Classes to display for If this field is left blank, the message is sent to all clients. If one or more classes of client is entered (separated by spaces), however, it will only be used for clients that fall into those classes. This can be useful for defining message files just for anonymous users, especially when using absolute paths.

3. To define files that clients will be notified that they exist, you will need to fill in the **README files** table. Again, this lists all existing files and has a blank row for adding a single new one. The meanings of the fields in this table's columns are:

Path The path to the file whose existence and modification time should be sent to the client. This can be either an absolute path like `/etc/README` or a filename relative to the directory being entered like `README.txt`. You can even use shell wildcard characters like `*` and `?` in the filename to match multiple files—for example, `README*`.

When to display last modified date If **At login** is selected, the information will be sent to clients after logging in. If **Entering any dir** is chosen, the file will be searched for and its modification date sent when changing to any directory. When **Entering dir** is

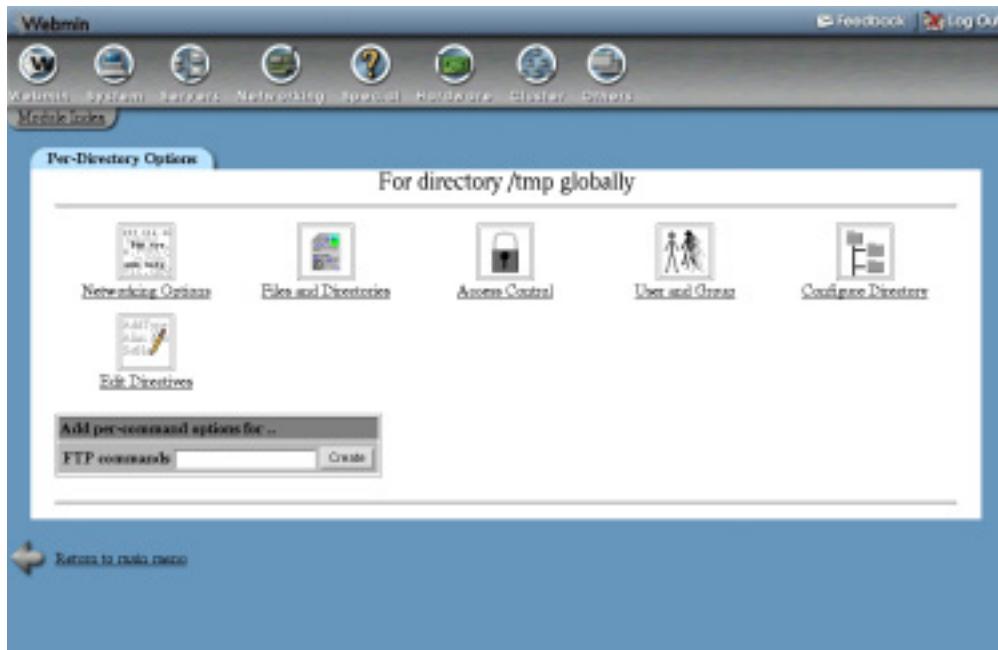


Figure 41.4 The messages and banners form.

chosen, the modification date will only be sent when the directory whose path you specify in the adjacent text box is entered.

Classes to display for If this field is left blank, the modification is sent to all clients. If one or more classes of client is entered (separated by spaces), however, it will only be used for clients that fall into those classes.

4. To change the amount of information that WU-FTPD sends to clients when they connect, adjust the **Greeting level** field. If **Hostname and version** is selected, both the system's hostname and the FTP server version will be sent. If just **Hostname** is chosen, only the hostname will be displayed. If **Neither** is selected, no information will be sent. The latter two options are the most secure, as an attacker may be able to use your FTP server's version to find a bug in it that could be exploited to take over your system.
5. If you want to have the server send a message to clients as soon as they connect, put it in a file and select the **From file** option for the **Pre-login banner file** field. Then, enter the full path to the message file into the text box next to it.
6. To change the hostname that WU-FTPD sends in the greeting and other messages, select the second option in the **Hostname for messages** field and enter an alternative name in the text box. This can be useful if your system's real hostname does not match the name the FTP clients use (*server5.example.com* instead of *ftp.example.com*, for example).
7. When you are done with this form, click the **Save** button to activate your changes. They will apply to all new FTP clients that connect from now on.

On many operating systems, the WU-FTPD configuration will include one or two messages and README file definitions by default. The `.message` file is typically searched for in every directory and sent to clients, as is the modification time of any file whose name matches the `README*` pattern.

Any message files that you define can contain special codes starting with % that are replaced when the file is sent by dynamically generated text. For example, %U is replaced with the client's FTP login name, so a file containing the line *Welcome %U to the example.com FTP server* would be sent to the client as something like *Welcome jcameron to the example.com FTP server*. According to the WU-FTPD manual page, the available codes are those shown in Table 41.1.

41.10 Configuring Logging

In a typical default configuration, WU-FTPD will log all uploads and downloads to the `/var/log/xferlog` file. You can, however, choose the types of users for which logging will be done (UNIX, anonymous, or guest), have the log written to `syslog` instead and select to record commands that are security violations for some types of users. Logging to the system log gives you more flexibility, as you can choose which file messages are written to, although they will be mixed in with other daemon facility messages. See Chapter 13 and the System Logs module for more information on how `syslog` works and to which files it ultimately writes.

Enabling the logging of all commands allows you to track exactly what clients are doing, but it can consume a large amount of disk space. The logging of security violations (attempts to violate WU-FTPD's file restrictions, covered in Section 41.6 "Denying Access to Files") can be useful for detecting hackers and is unlikely to use up much space, as such violations are not usually very frequent.

To edit FTP logging-related options in Webmin, follow these steps:

1. On the module's main page, click on the **Logging** icon to bring up the small logging options form.
2. To have all FTP commands executed by clients (including trivial ones such as `CWD` and `LIST`) recorded in the system log, select types of users for which they should be logged from the **Log all commands for** field.
3. To change the types of users for whom transfer logging is done, select them from the **Log transfers for** field. The **In directions** subfield lets you choose whether uploads (**Inbound**), downloads (**Outbound**), or **Both** are recorded. On an anonymous FTP server, it may make sense to only record uploads due to the large number of downloads.
4. To have transfers written to `syslog`, select **System log** in the **Log transfers to** field. Or, to tell WU-FTPD to write to the `/var/log/xferlog` file instead, select **XFER log file**. If **Both** is chosen, transfers will be logged to both destinations. If you want to use a program like Webalizer (covered in Chapter 39) to analyze your FTP server's logs, they must be written to `xferlog` as the lines that end up being written out by `syslog` have additional information added and thus cannot be parsed.

Anything written to the system log will use the daemon facility, unless WU-FTPD has been compiled to use a different one, such as `local7`. This can in fact be quite useful, as it allows you to separate out the FTP messages and have them written to a different file while still enjoying all the benefits of `syslog`.

Table 41.1 Codes Usable in Message Files

%T	Local time on the server, formatted like Thu Nov 15 17:12:42 1990
%F	Free space on the filesystem of the current directory, in kilobytes
%C	The current working directory
%E	The maintainer's email address
%R	Remote host name
%L	Local host name
%u	Remote username as determined via <code>ident</code> authentication
%U	Username given at log in time
%M	Maximum allowed number of users in this class
%N	Current number of users in this class
%B	Absolute limit on disk blocks allocated
%b	Preferred limit on disk blocks
%Q	Current block count
%I	Maximum number of allocated inodes (+1)
%i	Preferred inode limit
%q	Current number of allocated inodes
%H	Time limit for excessive disk use
%h	Time limit for excessive file use

5. To enable the logging of attempted filename security violations, select the types of users for which this should be enabled from the **Log security violations for** field. These will always be written to `syslog`.
6. Click the **Save** button at the bottom of the page to save and activate the new logging settings.

41.11 Limiting Concurrent Logins

If your system is configured to allow anonymous FTP logins and you expect to receive a lot of traffic, it makes sense to limit the number of connections that can be open to the FTP server at any one time. This puts a ceiling on the network and CPU load that FTP transfers can generate, which is important if the system is being used for some other purpose (such as running a web server).

WU-FTPD allows you to define limits on a per-class basis, so that anonymous clients can be restricted while real UNIX users are not. It also lets you specify the times during which restrictions apply, so that a higher limit can be granted when the server is not as heavily used for other purposes (such as at night).

To set up concurrent login limits, follow these instructions:

1. On the module's main page, click on the **Limits and Access Control** icon. The form shown in Figure 41.4 will appear in your browser.
2. The **Concurrent user limits** table is where limits on the number of connections can be entered. Each row defines a limit that applies to a certain class at certain times. As usual with tables in this module, there will be one empty row at the bottom for adding a new limit (and if this is the first one, the table will only contain that one row). Existing limits can be edited by changing their fields in the table or deleted by selecting the **Empty** option from the class menu.
3. The fields in each row should be filled in as follows:

Apply to class You must select the name of the class to which this limit will apply from the menu. Multiple limits can be defined for the same class at different times.

Maximum users To set a limit for the chosen class, select the second radio button and enter the maximum number of concurrent connections into the adjacent text box. If the **Unlimited** button is selected, no limit will apply to the class at the specified times. For example, you could add a row that turns off restrictions at night above another row that sets them for the entire day.

At times If **Any time** is selected, the limit will apply all the time. If you choose the second option and enter a UUCP-style time specification into the text box, however, only connections made during that period will be restricted. For example, *Any0900-1700* means 9am to 5pm every day; *Mo,Tu,We* means Mondays, Tuesdays, and Wednesdays; *Wk* means weekdays; and *Wk1700-0900,Sa,Su* means times outside office hours.

WU-FTPD always checks the table in descending order for an entry that matches a connecting client's class and the current time, and stops when it finds one. This means that entries that specify times (such as *Any0900-1700*) should be placed above those that have **Any time** selected, so that the specific entry is actually used when appropriate.

Error message file This is the full path to a file containing a message that will be sent to clients whose connections exceed the limit. This should explain why they are being rejected and suggest other times or FTP servers to try.

4. Hit the **Save** button at the bottom of the page to activate the connection limits. To add more than one, you will need to revisit the form so that a new blank row appears in the table.

41.12 Restricting Clients by IP Address

Even though it is possible to block clients from certain addresses by ensuring that they do not fall into any class, there is a feature in the module dedicated specifically to blocking clients based on their IP addresses or hostnames. This can be used to lock out specific hosts that are abusing your FTP server, or to restrict access to clients from only your own company or home network.

To define banned client systems, follow these steps:

1. Click on the **Limits and Access Control** icon on the module's main page to open the form shown in Figure 41.4.
2. Each row in the **Deny access from** table specifies an IP address, hostname, or pattern from which block logins. As with other tables in this module, there will always be an additional empty row for adding a new restricted address.

In the **Deny from address** field, you can also enter the full path to a file containing banned addresses, using negated patterns like `/192.168.1.*`, or even the special address `!nameserved`, which matches all clients that do not have a valid reverse DNS address. Only one can be entered, however—to block additional addresses, you will need to add more rows.

In the **Error message file** field, you must enter the full path to a file containing a message that will be sent to blocked clients. This should explain to connecting users that they have been blocked, and perhaps give a reason why.

If you want to add more than one row, you will need to save this form and reopen it so that a new empty row appears at the bottom of the table. Existing restrictions can be edited by just changing their fields, or deleted by clearing out the address.

3. When you are done, hit the **Save** button at the bottom of the form to activate the new address restrictions.

41.13 Restricting Access to FTP Commands

WU-FTPD can be configured to restrict the FTP commands that certain types and classes of users can use. This is useful for stopping anonymous clients from modifying files, as on most FTP servers they are only allowed to download, not upload, rename, or delete. In fact, this is exactly how WU-FTPD is set up in its usual default configuration.

There are five commands to which you can restrict access, all related to server-side data modification. They are:

chmod Change the UNIX permissions of a file on the server (`chmod` in the UNIX FTP client).

delete Delete a file or directory on the server (`del` or `rmdir` in the UNIX FTP client).

rename Change the name of a file or directory (`rename` in the FTP client).

overwrite Upload a file with the same name as one that already exists.

umask Change the default UNIX permissions for newly created files (`umask` in the UNIX FTP client).

It is not possible to stop clients from using directory listing or download commands. It is also not possible to use this feature to prevent the upload of files that do not already exist. This can be achieved, however, by setting directory permissions appropriately or blocking all uploaded filenames as explained in Section 41.6 “Denying Access to Files”.

To define which clients can use particular commands, follow these steps:

1. Click on the **Permissions** icon on the module’s main page.
2. On the form that appears, the **Command restrictions** table lists existing commands and the user types and client classes that are or are not allowed to use them. As usual, you can add a new command using the blank row at the bottom, edit existing entries, or delete the restrictions on a command altogether by selecting the blank option from the **Command** menu.
3. The FTP server processes this table in descending order when a client tries to do something, and uses the selection in the **Allow?** column for the first entry that matches to decide whether or not it is allowed. This means that the order matters, and if two entries match, the first one in the table will decide what happens.
4. The fields for each row and their meanings are:

Command You must select the command being restricted from this menu or the blank option to delete the row.

Allow? This field determines whether or not an attempt to use the command by a client that matches the chosen user types and classes is allowed.

For user types The restriction will only apply to the types of user selected in this column. See Section 41.5 “Managing User Classes” earlier in the chapter for details on what each means.

For classes Only the client classes selected in this column will be effected by the restriction.

5. When you are done editing or adding to the table, hit the **Save** button to activate your changes.

If a client command does not match any entry in the table, it will be allowed by the FTP server (unless blocked by some other filename restriction set elsewhere).

41.14 Configuring the WU-FTPD Server Module

To change the paths that the module uses for the WU-FTPD configuration files and programs, you will need to click on the standard **Module Config** link in the top-left corner of the main page. Unlike other modules, there are no options related to the user interface so you will probably not need to adjust anything on the configuration form if the module is working for you. By

default, all the configuration fields are set to match the WU-FTPD package included with your operating system or Linux distribution. The configurable fields are shown in Table 41.2.

Table 41.2 Module Configuration Options

Full path to wuftpd	This field must contain the full path to the WU-FTPD server program, such as <code>/usr/sbin/in.ftpd</code> . If it is incorrect, the module will think that the server is not installed.
Full path to ftpaccess file	This field must contain the full path to the primary <code>ftpaccess</code> configuration file, usually found in <code>/etc</code> . Again, an error will be displayed on the main page if it cannot be found.
Full path to ftpconversions file	This field should contain the full path to the <code>ftpconversions</code> file.
Full path to ftpgroups file	This field should contain the path to the <code>ftpgroups</code> file.
Full path to ftphosts file	This field should contain the path to the <code>ftphosts</code> file.
Full path to ftpusers file	This field must contain the full path to the <code>ftpusers</code> file, which lists users who are not allowed to log in to the FTP server.
FTP server PID file	If you are running WU-FTPD in standalone mode, this field must contain the full path to its process ID file. If it is incorrect, the module will assume that the server is not running when it really is.

Even though there are fields for configuration files other than `ftpaccess`, at the time this book was written, the module did not actually edit those files yet.

41.15 Summary

After reading this chapter, you should understand what the WU-FTPD server does and how it compares with other FTP servers, like ProFTPD. You should also know how to set it up, and how to perform configuration tasks such as restricting access to directories and files, specifying message files, and creating aliases. You should also understand how anonymous and guest access work, and which files are needed for these forms to enable access in WU-FTPD.

SSH Server Configuration

This chapter explains the SSH protocol and how to use Webmin to configure both an SSH server and clients on your system.

42.1 Introduction to SSH

SSH is a protocol for securely logging into and transferring files to and from a UNIX system over a network. All SSH traffic is encrypted so that anyone listening in on the network cannot capture passwords, which is a vast improvement over the insecure telnet and FTP protocols. In a way, SSH can be thought of as a secure replacement for those protocols, although in fact it can be used for much more.

An SSH server is a daemon process that runs on a UNIX system waiting for connections. An SSH client is a program run by a user (or from a script) that connects to a server to start a remote login session or transfer some files. Both the client and server authenticate themselves to each other, so that each can be sure of the other's identity. Client authentication is done either with a username and password or a username and private key, while server authentication always uses a key.

There are many different SSH clients and servers available, but the two most common client/server packages on UNIX systems are the freely available (open source) OpenSSH and the original, commercial SSH. In addition, there are two different versions of the protocol that are not compatible—versions 1 and 2. Fortunately, the latest releases of both packages support both versions.

The directory `/etc/ssh` contains all the configuration files used by both SSH servers, and even though the filenames are the same, their formats are slightly different. The primary file is called `sshd_config` and consists of a series of directives—one per line. As is usual with UNIX server configuration files, each sets some option such as the list of denied users or the IP address on which to listen. The same directory also contains the file `ssh_config`, which sets options that apply to the SSH client programs (such as `ssh` and `scp`) running on your system. The Webmin module covered in this chapter can edit both files directly.

42.2 The SSH Server Module

This chapter deals with the configuration of both the SSH and OpenSSH servers, and assumes that you have a basic user's knowledge of the client programs. The Webmin module that can be used to carry out this configuration is named SSH Server and can be found under the Servers category. Clicking on its icon will take you to the main page shown in Figure 42.1, assuming that the SSH package is installed.

If an error message like **The SSH server config file /etc/ssh/sshd_config was not found on your system** appears instead, there is probably no SSH server installed on your system. Most modern Linux distributions come with an OpenSSH package, so check your distribution CD or website for any packages whose names start with `openssh` or `ssh`. Often there will be several, such as `openssh`, `openssh-client`, and `openssh-server`, all of which should be installed. You may also need to install the OpenSSL library as well, which should also be available in package form. Use the Software Packages module (covered in Chapter 12) to check for and install everything that is needed.

If no SSH package exists for your operating system, you will need to download, compile, and install the OpenSSH or SSH source code. As you might expect, OpenSSH can be found at www.openssh.org/, while the original SSH can be downloaded from www.ssh.com/. Installing should be easy on any UNIX operating system, assuming you have a compiler available. The only dependency is an SSL library like OpenSSL, which can be downloaded from www.openssl.org/.

After installation, you should make sure that the SSH server will be started at boot time. Use the Bootup and Shutdown module (from Chapter 9) to create an action that runs the `sshd` command when started. If there is already an action named `sshd` or `ssh-server`, all you will need to do is make sure that it is enabled.

No matter how you install the SSH server, it should allow clients to log in and transfer files immediately using the default configuration once the server process is started. In fact, on a typical system, very little configuration is needed at all as the defaults are suitable for the average server.

The two different SSH implementations and their many versions all have slightly different configuration file formats, to which the module needs to adapt itself. This means that the forms and fields that make up its user interface are not always the same, depending on the version and type of server that you have installed. The instructions and screenshots in this chapter have been written with OpenSSH version 2.5 in mind, but any differences or extra features that other SSH versions have will be mentioned as well.

The main page will always display six icons, under each of which is a form containing fields for setting options related to some category—such as authentication or networking. At the top, the implementation and version of the installed SSH is displayed so you can see which of the instructions in this chapter apply to your system. At the bottom is a button labeled **Apply Changes**, which, when clicked, signals the SSH server to reread its configuration file. No changes made in the module will take effect until you hit this button.

42.3 Restricting Access to the SSH Server

By default, any UNIX user will be allowed to log in remotely to the SSH server on your system, or use it to upload and download files. On a mail server system or one that hosts websites, however, this may not be appropriate. You might want to allow most users to only log in to your

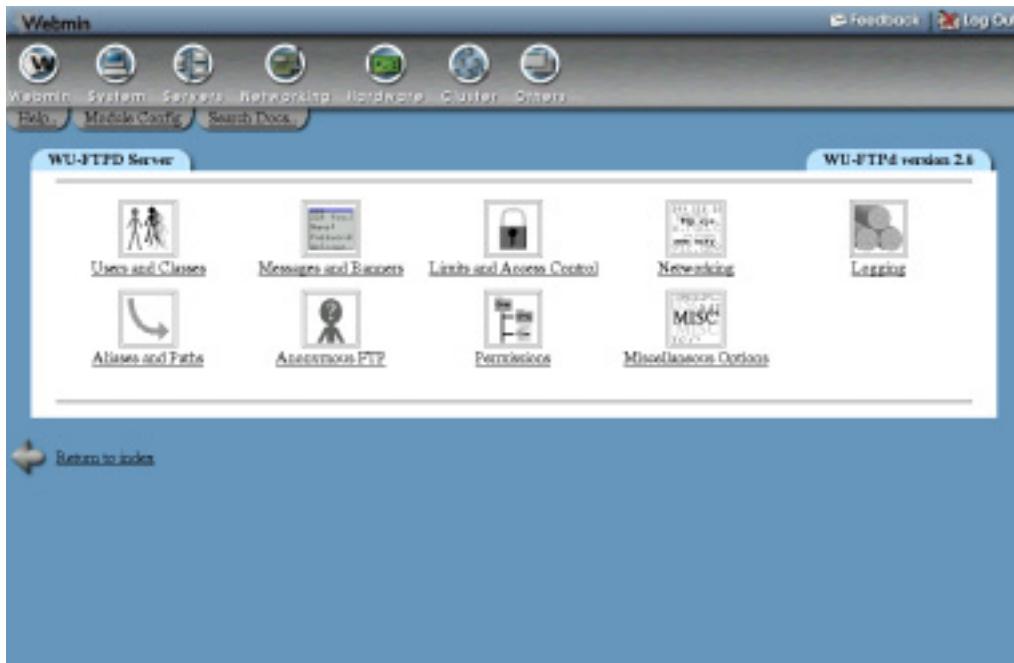


Figure 42.1 The SSH Server module.

POP3, FTP, or Usermin servers instead. Although it is possible to achieve this by giving them a shell like `/bin/false`, this could cause other problems with user Cron or *At* jobs.

Fortunately, the SSH server can be configured to restrict who can log in. Just follow these steps:

1. On the module's main page, click on the **Access Control** icon to bring up a form listing allowed and denied users.
2. To block everyone except a few users, enter a list of usernames separated by spaces into the **Only allow users** field. The * and ? wildcard characters can be used, so you could enter *foo** to allow any user whose name starts with *foo*. You can also deny SSH access for everyone except the members of some groups by filling in the **Only allow members of groups** field. Users who are secondary members of any matching groups will be allowed as well. If both fields are filled in, users specified in either will be allowed.
3. You can also block only a few specific users or groups using the **Deny users** and **Deny members of groups** fields. Again, wildcards may be used and, if both fields are filled in, users from either will be denied.
4. If you are running SSH, the fields **Only allow client hosts** and **Deny client hosts** will appear on the form as well. If filled in, the former tells the SSH server to reject any connections except those from the IP addresses or hostnames entered, while the latter tells it to deny only the hosts and addresses listed in the adjacent field. Both fields accept the * and ? wildcard characters.

If your system has OpenSSH installed, you can use the TCP-wrappers configuration files `/etc/hosts.allow` and `/etc/hosts.deny` to block untrusted clients. Unfortunately, there is not yet a standard Webmin module for editing these files.

5. Click the **Save** button at the bottom of the page to update the SSH server configuration file and return to the main page.
6. Hit the **Apply Changes** button to activate the new restrictions.

42.4 Network Configuration

The SSH server has several options that allow you to configure the IP address on which it listens, the port it uses, and various protocol-related settings. To edit them, follow these steps:

1. Click on the **Networking** icon on the module's main page to bring up the form shown in Figure 42.2.
2. By default, the server will accept connections made to any of your system's IP addresses. To change this (perhaps because you want it to be only accessible from an internal LAN), select the second radio button in the **Listen on address** field and enter an IP address into the text box.

If you are running OpenSSH, version 3 or above, this field will instead contain a table in which you can enter multiple addresses and ports. Above it are two radio buttons: **All addresses**, which if selected tells the server to accept connections to the default port on any IP address, and **Entered below**, which indicates that the addresses and ports in the table should be used. As is usual with tables in Webmin, this one will always have a single blank row at the bottom for adding a new address and port. If none have been defined yet, it will only contain one blank row. The meanings of the fields in the table's two columns are:

Address In this field you must enter a single IP address or hostname on which the server is listed.

Port If **Default** is selected in this column, the standard port set in Step 3 will be used. If the second option is selected, the SSH server will listen on the port entered into the text box in the column.

3. To change the port on which the SSH server listens for connections, edit the **Listen on port** field. If you do change it, clients will need to specify the new port when connecting. If your system uses OpenSSH version 3 or above, this field only sets the default port, which can be overridden in the **Listen on address** table.
4. In the **Accept protocols** field, check the boxes for the SSH protocol versions that your server should accept. It is generally wise to allow both, so older or newer clients can connect without difficulty. This field only appears if you are running OpenSSH; however, SSH accepts only version 1 or 2 depending on the SSH version you have installed.
5. If you are running SSH, the **Idle timeout** field can be used to disconnect clients that have neither sent or received any data for a certain amount of time. Select the second radio button, enter a period of time into the text box, and select the units for that period from the menu. If **Default** is selected, clients will never be cut off no matter how long they are idle. On a busy system, this feature can be useful for stopping people from leaving idle

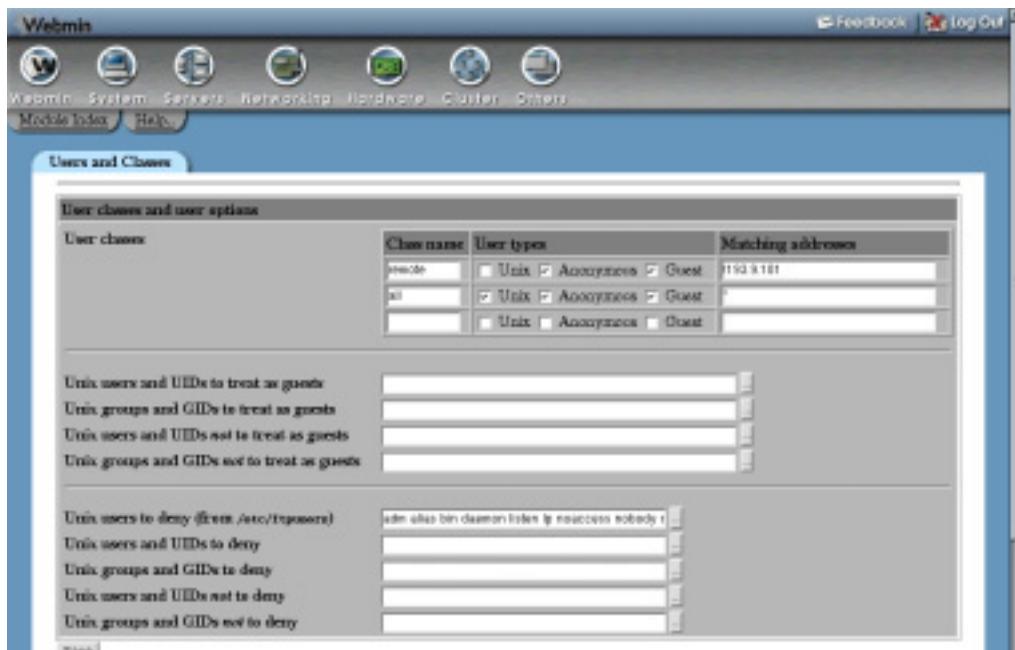


Figure 42.2 The networking options form.

SSH sessions open for days at a time, each of which has an associated memory-consuming `sshd` and shell process.

6. To set the SSH server to disconnect clients that shut down or crashed without properly logging out, select **Yes** in the **Disconnect if client has crashed?** field. The server will periodically send messages to the client to make sure it is still really running, and close the connection if there is no reply. The only time you would want to choose **No** is if this extra traffic causes problems on your network, such as the automatic activation of an ISDN or dial-up connection when it is not really necessary.
7. To configure the amount of time that the server will wait for a client to authenticate after it has connected, change the **Time to wait for login** field. If **Forever** is chosen, the server will never disconnect a client no matter how long it takes, which could allow an attacker to overload your system by making lots of SSH connections that do nothing.
8. One of the SSH protocol's more interesting features is its support for port forwarding, which allows clients to access ports on the server's network that they could not ordinarily. Even though this is very useful for users, you might consider it a security risk as it allows anyone who can make an SSH effectively bypass IP address restrictions on internal servers. To turn off this feature, change the **Allow TCP forwarding?** field to **No**.
This field only appears if you are running SSH version 2 or above or OpenSSH.
9. A related field is **Allow connection to forwarded ports?**, which determines if hosts other than the server itself are allowed to connect to ports forwarded back to the client. You may want to set this to **No** to protect client users from attackers who are misusing

possibly insecure forwarded connections back to the client's network. It only appears, however, if your system runs OpenSSH version 2 or above.

10. To have the server look up the hostnames for client addresses and the address for those hostnames, and then block those that do not match, select **Yes** in the **Reverse-validate client IP addresses?** field. This is useful if you have hostname-based access controls in place and want to detect attackers using falsified DNS records. This field is only visible if you are running OpenSSH version 2.3 or above.
11. To save and activate your changes, hit the **Save** button at the bottom of the page and then **Apply Changes** back on the module's main page. They will take effect for any new client connections.

42.5 Authentication Configuration

All SSH implementations have options related to how clients authenticate and the messages displayed to them after login. Specifically, you can permit or deny authentication by username and password or username and certificate, stop the `root` user from logging in, and control whether or not `rlogin`-style `.rhosts` files are trusted. The exact options differ quite a lot between SSH versions, however, so what is possible with OpenSSH may not be if you are running the commercial SSH server.

To edit authentication settings, follow these steps:

1. Click on the **Authentication** icon on the module's main page to bring up a form like the one shown in Figure 42.3.

The screenshot shows the 'Messages and Banners' section of the Webmin Authentication configuration. It contains two tables: 'Messages, banners and README files' and 'README files'.

Messages, banners and README files

Message files	Path	When to display	Classes to display for
info/nologin	/etc/nologin	<input checked="" type="radio"/> At login <input type="radio"/> Entering any dir <input type="radio"/> Entering dir	
message	/etc/issue	<input type="radio"/> At login <input checked="" type="radio"/> Entering any dir <input type="radio"/> Entering dir	
	/etc/issue.net	<input type="radio"/> At login <input checked="" type="radio"/> Entering any dir <input type="radio"/> Entering dir	

README files

README files	Path	When to display last modified date	Classes to display for
README	/etc/README	<input checked="" type="radio"/> At login <input type="radio"/> Entering any dir <input type="radio"/> Entering dir	
README	/etc/README	<input type="radio"/> At login <input checked="" type="radio"/> Entering any dir <input type="radio"/> Entering dir	
	/etc/README	<input type="radio"/> At login <input type="radio"/> Entering any dir <input type="radio"/> Entering dir	

Below the tables are several configuration options:

- Greeting level: Radio buttons for "Hostname and version", "Hostname", and "Neither".
- Pre-login banner: Radio buttons for "None" and "From file" (with a dropdown menu).
- Hostname for messages: Radio buttons for "System hostname" and "Custom" (with a text input field).
- Owner's email address: Radio buttons for "Default" and "Custom" (with a text input field containing "cameron@carlybracket.com").

Figure 42.3 The authentication options form.

2. Select **Yes** for the **Notify user of new mail?** field to have users informed of any new mail in their mail files when they log in. This only works if you are using the standard mail file location on your system, though, and not if delivery is done to `Mailbox` or `Maildir` in users' home directories.
3. To prevent users logging in with a password, change the **Allow authentication by password?** field to **No**. This means that only certificate authenticates will be accepted, which is not too useful for users who have never logged in before and thus cannot create a private key. It is only useful if your system uses NFS-mounted home directories, or if some other mechanism exists for users to set their public keys. This field is not available if you are running SSH version 3 or above.
4. To allow or deny logins with an empty password (assuming this is actually correct for a user), change the **Permit logins with empty passwords?** field. You may want to block this until users have set their passwords by some other method.
5. Even though a `root` login via SSH is much more secure than one via telnet (which is unencrypted), you may still want to prevent it. To do this, select **No** from the **Allow login by root?** menu. You can also choose **Only with RSA auth** to force `root` logins to use a certificate for authentication, or **Only for commands** to only permit the execution of a single command instead of allowing a full interactive login. That final option is only available, however, if your system runs OpenSSH version 2 or above.
6. To stop users from using certificates to authenticate (and thus forcing the use of passwords instead), select **No** from the **Allow RSA authentication?** field. You might want to do this to force people to enter a password every time, instead of relying on a possibly unencrypted private key to do the authentication for them.
7. To stop the server from strictly checking permissions on users' files in their `~/.ssh` directory, select **No** in the **Check permissions on key files?** field. Even though turning off these checks is a bad idea from a security point of view, the checks can be annoying for users who have set the wrong permissions and cannot figure out why they cannot be authenticated with a certificate.
8. To have the server display the contents of the message-of-the-day file to users after logging in, select **Yes** for the **Display /etc/motd at login?** field. This file usually contains information about your system or notices to users.
9. If you want to have a message sent to clients before they log in, select the second option in the **Pre-login message file field** and enter the full path to a file containing the text you want sent into the adjacent text box. This text often contains a warning about unauthorized use of the system.

This field is only available if you are running OpenSSH 2.3 or SSH version 2 or above.
10. The rest of the options on the page relate to `rlogin`-style authentication using `rhosts` and `/etc/hosts.equiv` files. Because they trust the client host to have already authenticated the connecting user, they are rather insecure with the ease with which a source IP address can be faked. For this reason, enabling this kind of authentication is not recommended.
11. To save and activate your new authentication settings, hit the **Save** button at the bottom of the form, followed by **Apply Changes** on the main page.

42.6 Editing Client Host Options

Although the SSH Server module is primarily for configuring an SSH server, it also lets you set options that apply to all client connections made from your system using the `ssh` and `scp` commands. Options can be set for connections to all hosts, or just to a specific one. You can set the port to which to connect, the protocol to use, and local and remote ports to forward.

The settings made in this module apply to all users on your system, but can be overridden by individual users who edit their `~/.ssh/config` files. This can be done manually or using Usermin, which has an SSH Client module with an identical interface to the one documented here for editing global client settings. Many of the settings do not make much sense to set for all users, even though it is possible to do so using Webmin. For this reason, the instructions in this section only cover fields that are useful on a global level.

To define settings for connections to a specific host, follow these steps:

1. On the module's main page, click on the **Client Host Options** icon. A page containing one icon for each of the hosts for which options have been set will be displayed. Unless you have used this page before, only the special **All hosts** icon will appear, which can be clicked on to edit options for connections to any host.
2. Click on the **Add options for client host** link at the bottom of the page to bring up a form for specifying a host and the options that apply to it. All of the fields on this form have a **Default** option, which, if selected, indicates that the setting for all hosts should be used instead. This allows you to define options globally, and then override them on a per-host basis.
3. In the **Options for host** field, enter the name of the host (as used in the `ssh` command line) to which the options will apply. Wildcards can be used. For example, you could enter `*.webmin.com` to match any host in the `webmin.com` domain. Remember that the name must match that used by users in the `ssh` or `scp` command, so if you enter `foo` and a user runs `ssh foo.example.com`, the options will not apply even though both names would resolve to the same IP address. For this reason, you may want to enter the host-name as `foo*` to catch both possibilities.
4. To have SSH clients connect to a different hostname, fill in the **Real hostname to connect to** field. This could be useful if combined with the **Port to connect to** field to secretly redirect user connections to a specific host to a port on another address which is actually a tunnel to the actual destination.
5. To force clients to use a different port by default, fill in the **Port to connect to** field. This is useful if the SSH server on a particular host runs on a different port from the usual 22, and you want to avoid the need to explicitly specify the port in every `ssh` and `scp` command.
6. The SSH client normally treats the `~` (tilde) character as an escape that indicates that the next character entered by the user is actually a command for the `ssh` program itself. For example, `~.` closes the connection, and `~^Z` suspends the program. The **Escape character** field can be used to specify some different character by selecting the third radio button and entering a single character into the adjacent text box. You can also turn off escape support altogether by selecting **None**. This latter option is useful if you are using the `ssh` command to transfer binary data that may contain a tilde.
7. By default, the SSH client and server will compress and uncompress data sent between them, which can speed up large transfers of text or other compressible data. Sometimes,

however, this can actually slow things down or be a useless waste of CPU time, for example, if you are using `scp` to copy lots of GIF files or always connecting to the host over a fast network. To turn off compression, change the **Compress SSH traffic?** field to **No**.

If compression is enabled, the **Compression level** menu controls the trade off between CPU utilization and the amount of bandwidth used. If **1** is selected, very little compression is done, whereas if **9** is chosen, a lot more CPU time will be expended on reducing the actual amount of data transferred.

These fields and those in the next two steps are not available if your system is running SSH version 3 or above.

8. By default, SSH clients will use the privileged source port 22 when connecting, which indicates to the server that it is a trusted program and thus can be relied on to provide correct information about the user running it. This is necessary for `rlogin`-style authentication to work, but unfortunately many networks have their firewalls configured to block connections with privileges source ports, which completely blocks SSH. To have the clients use a normal port instead, select **No** for the **Use privileged source ports?** field. Unless you are using host-based authentication, this will cause no harm.
9. To set the SSH protocol versions that clients will try when connecting to this server, choose **Selected** in the **Try SSH protocols** field and check the ones to try. The default is to try them both.
10. Hit the **Create** button at the bottom of the page to save the new per-host settings. They will be used by all new client connections made from your system from now on.

After a set of host options is created, an icon for the host will appear on the Client Host Options page. You can click on this icon to bring up its editing form, make changes to the same fields, and hit the **Save** button. To remove the host and have connections to it revert to the default options, hit **Delete** on the same form. It is also possible to change the defaults that apply to all connects by clicking on the special **All hosts** icon and making changes on the form that appears. Of course, some fields do not really make sense in this context, such as **Real hostname to connect to** and **Port to connect to**, and so should not be used.

42.7 Setting Up SSH for New Users

Before a UNIX user can use certificate authentication to log in to an SSH server, he must generate a private key with the `ssh-keygen` command. This module can be configured to work with the Users and Groups module to run this command for all newly created users. If your network uses NFS-mounted home directories, this will allow new users to log in to other hosts without needing to supply a password, with no further setup needed.

To configure the setup of SSH for new users, follow these steps:

1. On the module's main page, click on the **User SSH Key Setup** icon.
2. Check the **Setup SSH key for new UNIX users** checkbox, so that `ssh-keygen` will be run for new accounts.
3. To have the new user's public key added to the list of keys that are authorized to use his account, check the **Copy new identify.pub to authorized_keys** box. If it is not selected, they will need to do this manually before authentication with their new certificate will be accepted.

4. To set a passphrase for new users' private keys, check the **Use password as key passphrase** box. If it is left unchecked, no passphrase will be set (which is more user-friendly, but less secure).
5. Click the **Save** button to have Webmin start using your new settings.

42.8 Configuring the SSH Server Module

Like all modules, this one has several options that can set to control where it looks for the SSH programs and configuration files. By default, they are set to match the SSH package that comes with your operating system, and so generally will not need to be changed unless you have compiled and installed the server from source instead. None are related to the module's user interface.

The editable settings are shown in Table 42.1.

Table 42.1 Module Configuration Options

Full path to sshd program	This field must be set to the full path to the SSH server program, such as <code>/usr/sbin/sshd</code> or <code>/usr/local/sbin/sshd</code> .
Full path to sshd config file	This field must contain the location of the SSH server configuration file, usually found at <code>/etc/ssh/sshd_config</code> .
Full path to ssh client config file	This field must contain the location of the global SSH client configuration file, usually found at <code>/etc/ssh/ssh_config</code> .
Full path to sshd PID file	This field must contain the full path to its process ID file so the module can determine whether or not the SSH server is running. If it is incorrect, the Start Server button will appear on the main page instead of Apply Changes .
Command to start sshd	If Just run server is selected, the module will simply run the <code>sshd</code> program specified in the first field when the Start Server button is clicked. Many packages include a bootup script that should be used instead, which you can tell the module to use by selecting the second option and entering a command like <code>/etc/init.d/sshd start</code> into the adjacent text field. If you have compiled SSH yourself from the source, however, then the first option should be used as no such script is likely to exist.
Full path to ssh-keygen program	This field must contain the full path to the <code>ssh-keygen</code> program, which the module uses to setup SSH for new UNIX users.

42.9 Summary

After reading this chapter, you should be familiar with the uses and benefits of the SSH protocol and programs. You should also understand how Webmin's SSH Server module can be used to configure such server options as the allowed users and hosts, protocol and networking settings, and options related to logging in and authentication. You should also know how to set options that apply to client SSH connections made by users on your system, and how to have SSH set up automatically for new users.

Windows File Sharing with Samba

This chapter explains the protocol by which Windows systems share files, and explains how to set up the Samba program to make files on your UNIX server available to Windows clients.

43.1 Introduction to SMB and Samba

SMB (Server Message Block) is the protocol used by Windows systems to share files and printers across a network, just like the NFS and LPR protocols are used by UNIX systems. Any time you use the Network Neighborhood or Map Network Drive features of Window, the SMB protocol is being used. Because it is the standard method of file sharing on Windows systems, it has become the most commonly used method of sharing files on local networks.

Even though SMB is thought of as a Windows protocol, it was originally developed by DEC and has been implemented by many different companies and in many products. These days it is often referred to as CIFS (the Common Internet File System), even though the protocol itself has not changed. In fact, many ancient clients will still be able to access modern SMB servers like Samba.

An SMB server is a system that has files or printers to which it wants to allow other hosts access. A client is a system that wants to read or write files on a server, or print to a server's printer. A single system can be both a client and server, and all releases of Windows from version 95 onwards include software for these purposes. On a typical organization's network, however, there is a single large server system and many smaller clients that access files on it.

Every host that uses the SMB protocol has a hostname, which is typically the same as its DNS name. A server host can have multiple shares, each of which has a unique name and corresponds to a directory or local printer on the server system. Shares are referred to using the `\hostname\sharename` notation, such as `\corpserver\documents`. On Windows clients, file shares are normally mapped to drive letters like `J:` so they can be more easily referred to. All

Windows applications can read and write files on a server in exactly the same way that they would for local files.

Shared printers accessed by a client are not assigned a drive letter, but may be connected to a fake printer port such as *lpt2*::. Clients can send jobs to the printer, view those that are currently waiting to be printed, and cancel jobs submitted by the same user. Unlike the UNIX LPR protocol, clients using a remote printer must have the appropriate driver installed, and must send data to the server in the format that the printer actually accepts.

Fortunately, it is possible for Linux and UNIX systems to participate in SMB file and printer sharing as well, or this would be a very short chapter. The Disk and Network Filesystems module (covered in Chapter 5) allows your Linux system to mount shares from SMB servers so the files they contain can be accessed like any others. And the Printer Administration module (from Chapter 22) can be used to set up printers on your system that send jobs to Windows printer shares.

Those two chapters explain how your system can act as an SMB client, while this one covers setting up a server so that Windows (and Linux) clients can access its files and print to its printers. The software that makes this possible is called Samba, a completely free reimplementation of the SMB protocol for UNIX systems. Samba has been available and under development for many years, ever since the SMB protocol first started to be used on DOS systems. It allows a UNIX system to do as good a job of serving Windows clients as a real Windows server would. In fact, some would say that it is even better.

Samba uses two daemon processes, named `smbd` and `nmbd`. The first handles actual file or printer share requests from clients, while the second responds to SMB name lookup requests. Both daemons use the `smb.conf` configuration file, which is usually found in the `/etc` directory. Any change made to this file (either manually or by using Webmin) will be immediately detected by both daemons, and will take effect at once. Unlike most other UNIX server processes, they do not need to be signaled to reread the configuration file if it changes.

Unfortunately, there are some complexities that arise when sharing files between UNIX and Windows systems. The SMB protocol has no support for concepts such as file ownership or permissions, at least not in the form in which they exist on UNIX systems. NTFS filesystem access control lists (used on Windows NT, 2000, and XP) are supported instead, which are incompatible with normal UNIX permissions. Samba does have some support for them, but setting it up is complex and covered in a separate chapter.

The SMB protocol supports authentication, so that clients can be forced to provide a valid username and password to the server before they can access a share. The Samba server uses the standard UNIX user database to validate clients, although actual UNIX passwords cannot be used (for reasons explained later). When a client logs in to a Samba server, it accesses files with the permissions of the UNIX user as whom it authenticated—just as an FTP client would. This means that all the normal file permission and ownership rules apply.

Samba can be compiled on every version of UNIX supported by Webmin, and has the same features on all of them. This means that the module's user interface is the same as well, although differences in the default configuration may cause some features to be initially inaccessible.

43.2 The Samba Windows File Sharing Module

This Webmin module allows you to specify directories and printers to be shared to Windows clients using the SMB protocol. It can be found in the Servers category, and when its icon is clicked on the main page (as shown in Figure 43.1) it will be displayed. All existing shares are listed, along with their paths and the users to whom they are available. Below them are icons for setting various global options that apply to all shares, links for managing Samba users, and a button for starting or restarting the server processes.

Over the years, Samba has gained a vast array of configurable options. This module does not allow you to configure all of them though, only the ones that are useful for a small server on a simple network. For example, settings related to login scripts, NT domains, and SSL cannot be edited. If you add them to your `smb.conf` file manually, however, the module will not touch them.

Like all other modules that configure some server, this one can only be used if the Samba server is actually installed. If the module cannot find it, an error message like **The Samba server executable /usr/sbin/smbd was not found** will appear on the main page instead. If you do have Samba installed but in a different location than what the module expects, see Section 43.18 “Configuring the Samba Windows File Sharing Module” for instructions on how to reconfigure it to use the correct paths. Otherwise, you will need to install it.

Most Linux distributions, and several other operating systems, include a Samba package or packages, which can be easily installed using the Software Packages module (covered in Chapter 12). If not, you will need to download the source code from www.samba.org and compile and install it manually. The module expects you to use the package, if one is available, or the source code otherwise, so if you did not and an error message is still being displayed on the main page, the module’s configuration will need to be adjusted to use the right paths.

No matter how Samba is installed, its default configuration file will include at least two shares (the special `homes` and `printers`), as well as several global settings. This means that even if you have never used this module before or configured Samba manually, the list on the main page will not be empty. Of course, if you have been adding shares by directly editing the configuration file, they will be displayed as well.

If Webmin detects that Samba is already running, a button labeled **Restart Samba Servers** will be displayed at the bottom of the page. Predictably, clicking it will kill all running server processes and restart them, forcing the currently configuration to be reloading. This is usually unnecessary, though, as Samba will reread the configuration files as soon as it detects that they have been changed.

If the module finds that both of the Samba server processes are not running, it will display the **Start Servers** buttons, which, when clicked, will run both `smbd` and `nmbd`. No PID file is checked to determine if they are running or not. Instead, the module searches for running processes with those names.

43.3 Managing Samba Users

As mentioned in the introduction, the SMB protocol uses a password encryption format that is incompatible with the standard UNIX format. This was not originally a problem, as old versions of Windows (95 and earlier) sent unencrypted passwords to SMB servers. This allowed Samba to encrypt and verify them against the UNIX password list, just like the FTP or telnet servers do. Unfortunately, recent Windows releases will only send passwords in the new NTLM encrypted

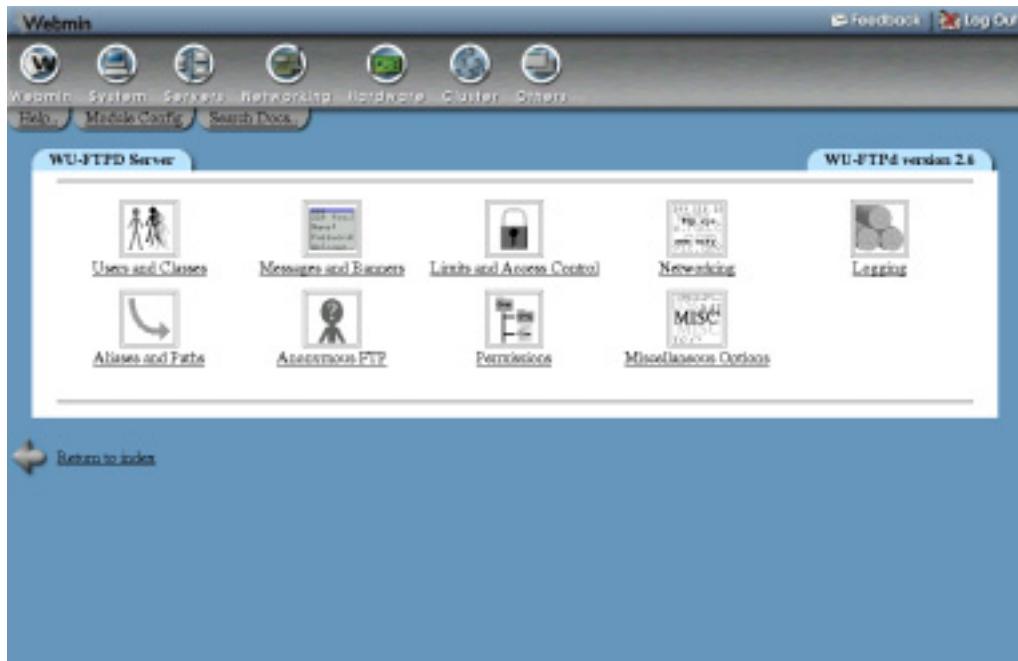


Figure 43.1 The Samba Windows File Sharing module.

format unless a particular obscure registry key is changed. Samba must now maintain a separate list of passwords to validate modern clients.

Unless your server is only going to be accessed by old Windows hosts or Linux systems, you will need to enable this separate encrypted password list. To do this, complete the following steps:

1. Click on the **Authentication** icon on the module's main page.
2. On the form that appears, change the **Use encrypted passwords?** field to **Yes**.
3. Click **Save** at the bottom of the form to return to the main page and activate the new setting. If it did not appear before, the **Encrypted Passwords** section containing three links should be visible now.

Now that Samba's separate password list is enabled, you will need to add some of your existing UNIX users to it. This can be done easily using Webmin by following these steps:

1. On the main page of the Samba module, click on the **Convert UNIX users to Samba users** link in the **Encrypted Passwords** section to bring up the conversion form.
2. The **Don't convert or remove these users** field lists users that will be excluded from conversion, and, by default, contains all system accounts. You may want to add others; however, there is no harm in converting accounts that will never be used.
3. If you have used this form before, the **Update existing Samba users from their UNIX details** option can be checked to have existing Samba users updated to match the corresponding UNIX users.

4. Similarly, the **Delete Samba users who do not exist under UNIX** can be checked to see if it is set to delete Samba users who no longer have a corresponding UNIX user.
5. The **For newly created users, set the password to** field determines the password that will be assigned, as there is no way to convert the users' existing passwords. The best choice is **Account locked**, which prevents the converted users from being used until a password is later set. You can also choose **No password** to leave new accounts password-less (a bad idea in terms of security), or **Use this password** to specify a password for all converted users.
6. Click on the **Convert Users** button to begin the process. A page listing each user converted, skipped, or updated will be displayed.

After conversion, you will probably need to set passwords for the new Samba users. This must be done one-by-one, by following these instructions for each user:

1. On the module's main page, click on the **Edit Samba users and passwords** link to bring up a list of existing users.
2. Click on the name of the user whose password you want to set.
3. In the **Password** field, select the **New password** option and fill in the text box next to it. You can also choose **No access** to block all Samba logins by this user, **No password** to allow logins without a password, or **Current password** to leave the password unchanged.
4. None of the other fields on the form should be changed—just hit the **Save** button to return to the user list.
5. You should now be able to log in to your Samba server as this user with the chosen password and access files in some share. Assuming that the special homes share exists, every user will have access to share with the same name as his username.

Because converting and setting the password for each new user is a tiresome waste of effort, you can configure the module to automatically create a Samba user for each UNIX user created in Webmin. It is also possible to have a Samba user renamed, deleted, or his password changed when the corresponding UNIX user is changed in the Users and Groups module. To set up this synchronization, follow these steps:

1. Click on the **Configure automatic UNIX and Samba user synchronization** link in the **Encrypted Passwords** section of the Samba module's main page.
2. Check the **Add a Samba user when a UNIX user is added** on the synchronization form to have a Samba user created with the right UID and password for each new UNIX user.
3. Check the **Change the Samba user when a UNIX user is changed** box to make sure the corresponding Samba user is renamed or his password changed when a UNIX user is modified.
4. Check the **Delete the Samba user when a UNIX user is deleted** to have Webmin remove the matching Samba user when a UNIX user is removed.
5. Click the **Apply** button to save your settings. Any actions performed in the Users and Groups module (when the **in other modules** options are used) will effect the Samba user list as well.

Unfortunately, this synchronization only applies to the Users and Groups, Change Passwords, and Cluster Users and Groups modules in Webmin. If you add a user at the command (like with `adduser`) or change a password with the `passwd` command, no Samba user will be added or updated.

43.4 Adding a New File Share

In its usual default configuration, Samba will allow any UNIX user to log in and access files in his home directory. The special `homes` share provides this feature, which in many cases is all that you need for users to store their own files on the server. It is often useful, however, to share a directory that everyone has access to, so that documents of interest to the entire organization can be made available. A share like this can be set up to allow guest access (meaning that no login is required to access it), or to require a valid login to the server.

To create a file share, follow these steps:

1. First, decide on the directory that you wish to share and create it if it does not already exist. It must be given the appropriate UNIX permissions so that users can read and/or write to it.
2. On the module's main page, click on the **Create a new file share** link above or below the table. This will take you to the **Share Creation** form shown in Figure 43.2.
3. In the **Share name** field, make sure the first button is selected and enter a unique alphanumeric name for your share into the text box—like *documents*. If you enter the name of a UNIX user, his automatic home directory share will be overridden.
4. In the **Directory to share** field, enter (or select with the little button) the full path to the directory chosen in Step 1.
5. To disable this share so that it cannot be used, change the **Available?** field to **No**. This can be useful if you want to take it offline until all the options have been set properly.
6. To hide this share from the list that appears when the server is browsed, change the **Browsable?** field to **No**. It will still be directly accessible using a `\server\sharename` path.
7. In the **Share comment** field, enter a short description for this file share, like *Corporate documents*.
8. Click the **Create** button to add it to the Samba configuration. Your browser will be returned to the module's main page, on which the new share will be listed.
9. Click on the new share name to bring up its editing page.
10. Click on the **Security and Access Control** icon to display the share's security form.
11. If the files in this share should be read-only, set the **Writable?** field to **No**—otherwise, make sure **Yes** is selected.
12. The **Guest access?** field determines if clients are allowed to access this share without logging in to the server. The available options are:
 - None** Only authenticated users will be granted access.
 - Yes** Anyone will be allowed to access the share, but unauthenticated clients will be treated as guests. Clients that log in will enjoy their normal file access rights.
 - Guest only** All clients, authenticated or not, will be treated as guests.

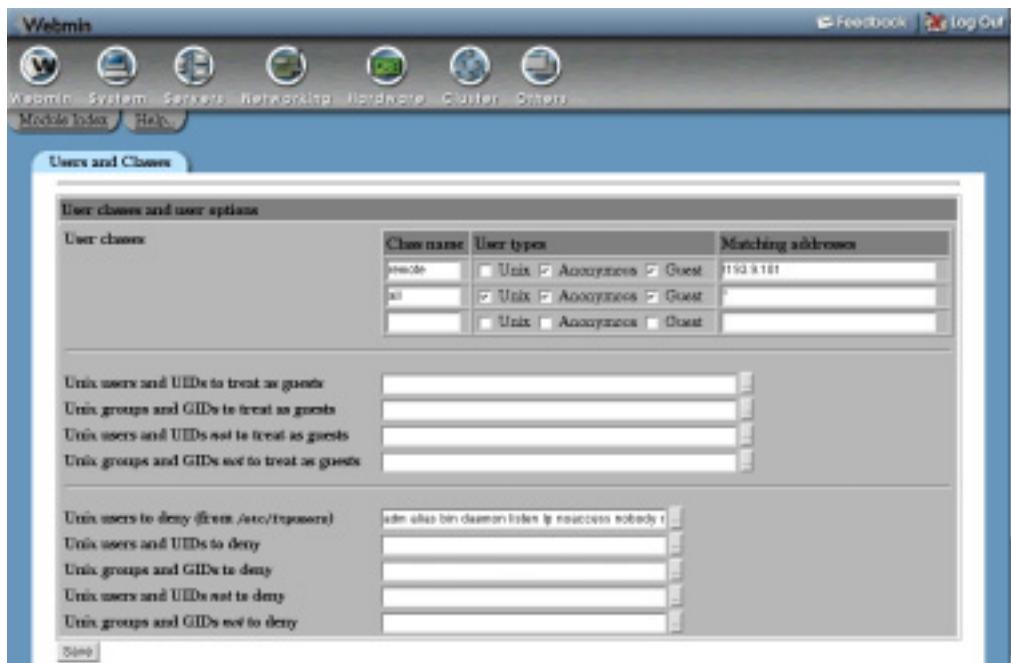


Figure 43.2 The file share creation form.

13. To set the UNIX user as whom guests read and write files, change the **Guest UNIX user** field. This should normally be an account with read-only access.
14. Click the **Save** button at the bottom of the form. The share is now ready for clients to use and should show up when your server is browsed.

A share can be edited after creation by clicking on its name in the list on the module's main page to bring up its editing form, changing details like the path or description, and hitting the **Save** button. Or it can be deleted entirely by clicking **Delete** on the same form. You can also edit additional parameters by clicking on the icons at the bottom of the editing page. Later sections in this chapter explain what they do in more detail.

The homes share can be edited as well, although it does not usually have a path (or if it does, it will contain the special %U code which is replaced by the connecting user's home directory).

43.5 Adding a New Printer Share

The default Samba configuration usually contains the special `printers` share, which indicates that all printers on your system are available to SMB clients. A specific printer, however, can be explicitly shared. This may be better than having them all shared automatically, as it allows you to set different options for each printer or exclude some from sharing altogether.

Before printing and the browsing of printers in Samba will work properly, it must be configured to use the right print system for your UNIX box. See Section 43.15 “Configuring Printers” for details of how to set this. If it is set incorrectly, the server will use the wrong commands for

listing printers and submitting jobs, which may cause the automatically generated list of printers to be empty, or print requests to fail.

To make a printer available to SMB clients, follow these steps:

1. On the module's main page, click on the **Create a new printer share** link above or below the table. This will take you to the Print Share Creation form shown in Figure 43.3.
2. In the **Share name** field, make sure the first button is selected and enter a unique alphanumeric name for your share into the text box, like *hplaser*. This should be the same as the name of the UNIX printer you select in the next step to avoid confusion. If an automatically created printer share with the same name already exists, this new one will override it.
3. From the **UNIX printer** menu, select the printer to make available to SMB clients. This list is taken from the Printer Administration module (covered in Chapter 22). If **Default** is chosen, the print system's default printer will be used.
4. To disable this printer so that it cannot be used, change the **Available?** field to **No**.
5. To hide this printer from the list that appears when the server is browsed, change the **Browsable?** field to **No**. It will still be directly accessible using a `\servername\printename` path though.
6. In the **Spool directory** field, you can enter the name of a directory in which temporary files for printing are stored. Leave it empty to use Samba's default, which will usually work fine.
7. In the **Share comment** field, enter a short description for this print, such as *Office HP Laserjet 5*.
8. Click the **Create** button to add the share to the Samba configuration. Your browser will be returned to the module's main page, which will now include the new printer in the table.
9. Click on the new share name to bring up its editing page.
10. Click on the **Security and Access Control** icon to display the share's security form.
11. The **Guest access?** field determines if clients are allowed to print to this printer without logging in to the server. The available options are:
 - None** Only authenticated users will be granted access.
 - Yes** Anyone will be allowed to access the share, but unauthenticated clients will be treated as guests. Clients who have logged in will have print jobs submitted under their login names.
 - Guest only** All clients, authenticated or not, will be treated as guests.
12. To set the UNIX user as whom guests submit print jobs, change the **Guest UNIX user** field. This doesn't matter much, unless your printer system is configured to block certain users.
13. Click the **Save** button at the bottom of the form to return to the printer's editing page.
14. Click on the **Printer Options** icon.
15. If this printer is to be used by Windows clients and does not have a UNIX driver installed, enter its complete make and model into the **Printer driver** field. This must match exactly the name to which Windows refers to so clients know which driver to install. If **None** is selected, users adding this printer to their Windows systems will be asked to choose the printer model from a list, instead.

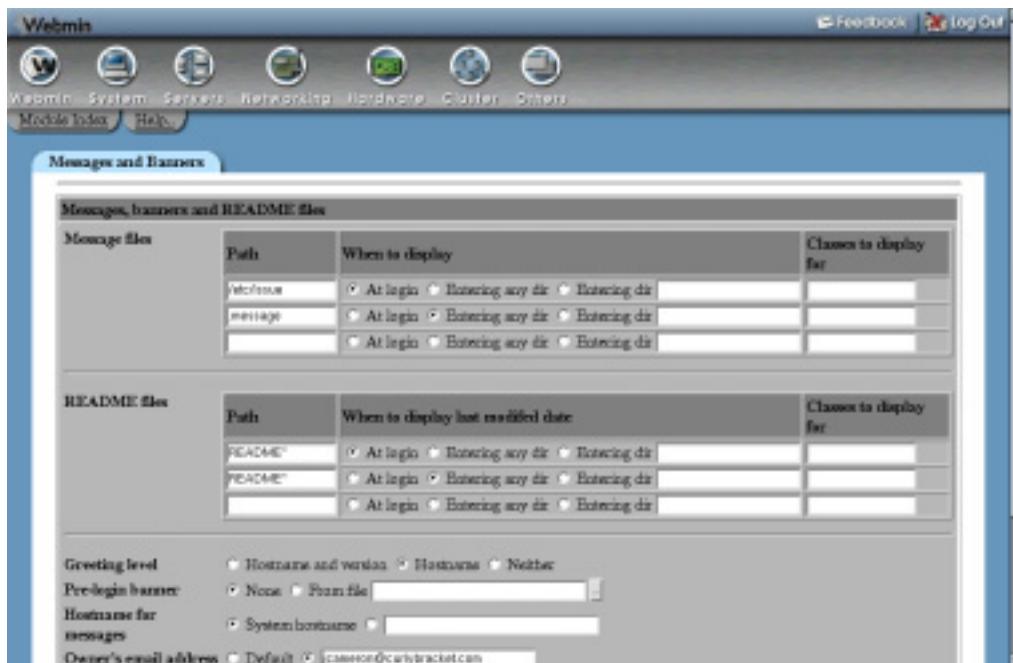


Figure 43.3 The printer share creation form.

If the UNIX printer selection in Step 3 is already set up with a driver, then clients must submit jobs in Postscript format instead of the native data format that the printer users (because the driver will do the conversion). In this case, you can enter the name of a printer that uses Postscript natively, such as *Apple LaserJet II*.

16. Finally, hit the **Save** button on this form. The printer share is now ready for use by Windows clients.

Just as with file shares, printers can be edited and deleting by clicking on their names in the table on the module's main page. The special **printers** share can be modified as well; however, many options do not make sense to set for it, such as the **UNIX printer** or **Printer driver**.

43.6 Viewing and Disconnecting Clients

Every client that is accessing a file or printer share on your system has a connection to the Samba server, and those connections can be viewed using this module. Clients may also lock files that they have open for editing, which prevents others from opening them. One of the server's tasks is the maintenance of these locks, which are associated with sessions and viewable. If a client crashes without properly disconnecting, any locks that it holds will remain until the TCP connection times out, which can take a while. For this reason, the module allows you to kill client sessions and thus release their locks.

To view and delete client sessions, follow these steps:

1. On the module's main page, click on the **View all connections** link above or below the table of shares to bring up a list of all connections to the server.

You can also click on a printer or file share and then on the **View Connections** button on its editing page to display a list of only connections to that particular share.

2. Either way, the page that appears will list the shares currently in use and show the name of the connecting user, the host from whom he connected, the time of connection, and any locked files for each share. In the left-most column is the ID of the Samba server subprocess that is handling this connection. Generally, multiple connections from the same client system to different shares will be handled by one process.
3. To kill a process and thus break all the connections that it is handling, click on its process ID in the first column. Any locks held by the client will be released, freeing the files for use by others.

You should only kill the connections of clients that have really crashed, as killing the session for an active client may cause any files that it has open to be corrupted. It is generally safe, however, to kill a connection to a Windows client with no files open, as it will be immediately and transparently re-established by the client when a file on the share is next opened.

43.7 Editing Share Security Options

Once a printer or file share has been created, you can edit various security-related options that control who has access to it and to which hosts they can connect. This can be useful if some share contains files to which only certain people should have access, or if your Samba server is for use by clients only on your internal network.

To edit share security options, follow these steps:

1. Click on the name of the share in the table to bring up its editing form on the module's main page, then click on the **Security and Access Control** icon.
2. As explained in Section 43.4 “Adding a New File Share”, the **Writable?** and **Guest access?** fields determine whether or not the share can be written to and if authentication is needed. The **Guest UNIX user** field sets the user as whom files are read and written by guest clients. Change them again here if you wish.
3. To only allow certain hosts access to this share, select the second radio button in the **Hosts to allow** field and enter a list of hostnames and IP addresses into the adjacent text box. Partial IPs like *192.168.1.* or network addresses like *192.168.1.0/255.255.255.0* can be used to allow access to an entire network. If your system is an NIS client, you can enter a netgroup name preceded by a @ (like @*servers*) to allow all of the group's members.
If **All** is selected, all hosts will be granted access, unless you fill in the next field. No matter what you enter, connections from the local host (*127.0.0.1*) are always allowed unless it is specifically listed in the **Hosts to deny** field.
4. To block only specific hosts from accessing this share, fill in the **Hosts to deny** field with a similar list of hostnames, IP addresses, networks, or netgroups. If both fields are filled in, **Hosts to allow** takes precedence. If **None** is selected, all hosts will be permitted.

5. To allow only certain users to access this share, fill in the **Valid users** field with a space-separated list of usernames. You can also fill in the **Valid groups** field with a list of groups whose primary and secondary members will be granted access. Only if both lists are empty will all users be allowed.
6. To deny specific users and members of groups, fill in the **Invalid users** and **Invalid groups** fields. If a user appears in both the valid and invalid lists, he will be denied access.
7. To restrict some users to read-only access for this share, enter a list of usernames into the **Read only users** field. You can also enter a list of UNIX groups in the **Read only groups** to restrict their primary members. Everyone else will have full read/write access, assuming that the share is actually writeable and that the **Read/write** fields have not been filled in.
8. To give only certain users permission to write to the share and restrict everyone else to read-only access, enter a list of usernames into the **Read/write users** field. As usual, the **Read/write groups** field can be used to enter a list of groups whose primary members will be allowed to write as well. Naturally, normal UNIX file permissions that may prevent writing to files or directories still apply to all users. If a user appears in both the **Read only** and **Read/write** lists, he will be allowed to write.
The fields in this and the previous step have no effect on printer shares. Instead, all allowed users will be able to print.
9. When you are done editing file security options, click the **Save** button at the bottom of the page to activate the new settings.

In addition to setting security options for a single share, you can set defaults for all shares that will apply unless overridden in individual shares. To do this, click on the **File Share Defaults** icon on the module’s main page instead of the name of a share, and then on **Security and Access Control**. Some settings—like the lists of hosts to allow or deny—really should be set globally, as you probably want to limit access to your entire server to just a trusted network. See Section 43.12 “Editing Share Defaults” for more information on how defaults work.

43.8 Editing File Permission Settings

File shares have several settings related to the UNIX permissions, and ownership of files within them, that can be set globally or on a per-share basis. Because Windows clients and the SMB protocol have no concept of permissions, it is useful to have a way to set the defaults for new files and directories on a per-share basis. To do this, follow these steps:

1. On the module’s main page, click on the name of the share for which you want to set permissions, then click on the **File Permissions** icon on its editing page.
2. In the **New UNIX file mode** field, enter the octal permissions (as used by the chmod command) that should be assigned to newly created files. For example, mode *600* would allow reading and writing by the owner but completely deny access to anyone else.
3. In the **New UNIX directory mode** field, enter the octal permissions for newly created directories. For example, *755* would allow listing and reading by everyone, but only allow the owner to create files in the directory.
4. To make some directories always appear empty to SMB clients, enter a comma-separated list of full paths into the **Directories not to list** field. For example, you might

enter `/proc`, `/dev` to hide the contents of those two directories, which are generally useless to Windows clients.

5. To force all clients to access files as a specific UNIX user (instead of the user as whom they logged in), fill in the **Force UNIX user** field. This can be very useful for a share in which different people edit each other's documents, as it avoids the UNIX permission problems that can occur if files are actually owned by their creators.
By default, the group as whom files are accessed will be the primary group of the specified user. To change this, fill in the **Force UNIX group** field, as well.
6. Because Windows SMB clients have no support for UNIX symbolic links, Samba will always read or write the linked-to file when a client tries to read or write a link. Unfortunately, this presents a potential security risk, as a symlink could be created that points to a normally inaccessible file outside the shared directory. To prevent this, change the **Allow symlinks outside of share?** field to **No**.
7. On UNIX filesystems, files that are read-only to a user can still be deleted if the directory is writeable. This is not the case on normal Windows filesystems, though, which is why Samba prevents it from happening. To change this and let UNIX filesystem semantics apply, change the **Can delete readonly files?** field to **Yes**.
8. Click the **Save** button at the bottom of the page to activate the new file security options.

As Section 43.12 “Editing Share Defaults” explains in more detail, you can edit file permission settings for all shares by clicking on the **File Share Defaults** icon on the main page, followed by **File Permissions**. These will apply unless overridden for a share by the preceding instructions.

43.9 Editing File Naming Options

Samba has several options that control how UNIX filenames are converted to names suitable for Windows systems. These days, most of them are no longer needed, as Windows versions 95 and above have been able to support long filenames properly. Only Windows 3.1 and DOS were stuck with the old 8.3 filename format, and they are hardly used anymore.

To edit the naming options for a share that are relevant to modern clients, follow these steps:

1. Click on the name of the share on the module’s main page, then on the **File Naming** icon.
2. When the **Case sensitive?** field is set to **No**, the server will ignore case when opening files requested by clients. This is the way Windows filesystems work and so this is the default behavior for Samba, as well. It does, however, consume more CPU time and IO bandwidth due to the need to scan directories, as all UNIX filesystems are case sensitive. For this reason, you may want to select **Yes** if all your clients are Linux systems that expect normal the UNIX case rule to apply.
3. Normally, Samba will create files with the exact case specified by clients. To change this and force the use of upper or lower case instead, change the **Preserve case?** field to **No** and select one of the options in the **Default case?** field. This can be useful if Windows clients are creating lots of upper-case files when you prefer to follow the normal UNIX lower-case standard.
4. On Windows filesystems, each file has a hidden attribute that determines if it is normally visible to programs or not. No such attribute exists on UNIX systems. Instead, files whose names start with a dot are hidden by `ls` and other commands. For this reason,

Samba sets the hidden attribute on dot files when the **Hide dot files?** field is set to **Yes**, as it is by default.

The alternative is to use the world execution bit of the UNIX file permissions as the hidden flag, as execute permissions are not otherwise used by Samba. To enable this behavior, change the **Save DOS hidden flag?** field to **Yes**. Because this will mess up permissions for UNIX programs accessing files in the share, it should only be used if the shared directory is only being accessed by SMB clients.

5. Windows files have two more attributes: the archive flag that indicates that a file has been backed up and the system flag that marks a normally untouchable system file. Samba can be configured to store these attributes in the user-execute and group-execute bits of files if the **Save DOS archive flag?** and **Save DOS system flag?** fields are set to **Yes**, respectively. If your Windows clients have no need for this information, or if you find that permissions on UNIX executables and scripts are being messed up, set them both to **No** instead.
6. To activate the new file naming settings, hit the **Save** button at the bottom of the page.

Again, these options can be set for all shares by clicking on the **File Share Defaults** icon on the main page, followed by **File Naming**.

43.10 Editing Other File Share Options

There are a few more file share options related to locking and automatically running commands that you can set using this module as well. Those are used for locking control of the behavior of Samba when a Windows client tries to lock a file to gain exclusive access, so that it can cache data in the file without having to contact the server for every read or write. By default, locking is fully enabled and implemented in exactly the same way as it is on Windows servers, so there is generally no need to change these settings.

Samba can also be configured to run shell commands when a client connects or disconnects, either as `root` or as the connecting UNIX user. This can be useful if you want to move newly added files to some other directory or perform some kind of processing on them.

To edit the module's other file sharing options, follow these steps:

1. Click on the name of the share to edit on the main page, and then click on the **Miscellaneous Options** icon on the share editing page that appears.
2. If this share is exclusively for read-only use (for example, if you are sharing some kind of read-only media like a CD), then the **Fake oplocks?** field can be safely changed to **Yes** to boost performance. This tells Samba to simply grant all lock requests by clients and not to bother actually keeping track of who has locked what, which can boost performance.
None of the other locking fields should be touched unless you really know what you are doing, as the defaults will work fine and any other settings may lead to data corruption if multiple clients try to access the same files.
3. To limit the number of clients that can be connected to this share at any one time, select the second radio button in the **Max connections** field and enter a number into the adjacent text box. This can be useful if you want to limit the load on your system. If **Unlimited** is selected, no maximum will be placed on the number of concurrent connections.

4. The fields **Command to run on connect** and **Command to run on disconnect** allow you to enter shell commands that will be run by Samba as the authenticated user at connection and disconnection time. They will always be run in the share directory, and special % codes like %U for the connecting user or %S for the server name can be used in the command.
5. Similarly, the **Command to run on connect as root** and **Command to run on disconnect as root** fields can be used to enter shell commands that will always be run as the UNIX root user. They will, however, be run in root's home directory instead.
6. Hit the **Save** button to activate the new locking and command settings.

One thing to remember about locking and Samba is that locks taken out by SMB clients will not generally effect or be detectable by UNIX programs or NFS clients. This means that data corruption can still happen if UNIX and Windows programs open the same file, or if the same NFS exported directory is shared by two different Samba servers.

43.11 Editing Printer Share Options

Once a printer share has been created, there are several options that you can set for it. Most of them relate to the commands that Samba will run to print a new job, list the queue, or cancel a job. By default, appropriate commands for the print system in use (explained in Section 43.15 "Configuring Printers") will be used. However, there are times that you will want to specify additional parameters or even use a completely different command.

To edit printer options for a share, follow these instructions:

1. On the module's main page, click on the name in the table of the printer share that you want to edit. On the form that appears, hit the **Printer Options** icon at the bottom of the page.
2. To prevent clients from using up all the disk space in the printer's spool directory with large jobs, change the **Minimum free space** field. You must enter a number of kilobytes that will always be left free on the filesystem.
3. To change the command that Samba will run to print a submitted file, edit the **Print command** field. The special codes %f (for the temporary file to print) and %p (for the printer name) can and should be used in the command, so you can enter something like *lpr -P%p %f ; rm %f*. Your command must always delete the temporary file (as the example does) because the server will not do this for you. All the usual shell meta-characters like :, &, and > can be used, which allows you to enter quite complex series of commands. Whatever command you enter will always be run as the UNIX user connected to the printer share.
4. To edit the command that Samba uses to list jobs waiting to be printed on some printer, select the second radio button in the **Display queue command** field and fill in its text box. Whatever you enter must produce output in the format generated by the standard BSD *lpq* command so that Samba can parse. If the special code %p appears in the command, it will be replaced with the name of the printer.
5. Similarly, you can change the commands that Samba runs to delete, pause, and unpause a print job by editing the **Delete job command**, **Pause job command**, and **Unresume job command** fields, respectively. All can and should use the codes %p for the printer name, and %j for the job ID. For most print systems, there are no defaults for the pause and un-

pause commands, as those features are not supported. Generally, you will not need to change these fields.

6. As Section 43.5 “Adding a New Printer Share” explains, the **Printer driver** field can be used to enter the model of the attached printer (as recognized by Windows) so that clients can automatically select the right driver.
7. When you are done with this page, hit the **Save** button to update the Samba configuration file and thus activate the new settings.

You can also edit these settings for all shares by clicking on the **Printer Share Defaults** icon on the module’s main page and then on **Printer Options**. In fact, all of the command options make much more sense to edit globally as the same commands are likely to be needed for all printers.

43.12 Editing Share Defaults

As the previous few sections in this chapter have mentioned, the Samba configuration allows you to define defaults that apply to all shares, unless specifically overridden. This can be done by clicking on either the **File Share Defaults** or **Printer Share Defaults** icon on the main page, editing the contents of the form that appears, and hitting **Save**. Most of the options in this form, however, are not particularly useful to set globally, except maybe **Available?** and **Browsable?**.

More usefully, you can click on one of the icons on the defaults page and change settings that will apply to all file or printer shares. In the case of the **Security and Access Control** icon (which appears on both pages), global defaults that you set will apply to both file and printer shares, as Samba does not differentiate between them.

Any option that is set globally will appear as the default on per-share forms. For example, if you fill in the **Delete job command** field under **Printer Options** on the Printer Share Defaults page and then went to the same page for a specific printer, the same value would appear. Even though the command does not actually appear in the configuration file for the printer, Webmin still displays it because it will be used by default. Of course, if you enter a different command for the share, it will override the global setting and thus be used and displayed. This behavior may be a little confusing, as it is not the way that other Webmin modules usually work.

43.13 Configuring Networking

This module can be used to set various Samba options that control how the entire server appears to and behaves for Windows clients. You can change the workgroup (under which the system is listed in the network neighborhood display), the server’s name and any aliases, and the description that appears next to the name. Options related to the file sharing protocol and authentication method used can also be edited to support old clients.

It is even possible to set up your system as a WINS server or client—a protocol that some Windows clients use to find IP addresses for SMB server names—if DNS is not available. The biggest difference between WINS and DNS is that clients can register their own names and IP addresses with a WINS server, rather than having it done by an administrator. It is most useful on small file-sharing networks that do not have a DNS server.

To edit these windows networking options, follow these steps:

1. Click on the **Windows Networking** icon in the **Global Configuration** section on the module’s main page to bring up the form shown in Figure 43.4.
2. To set a workgroup for your server, select the second radio button in the **Workgroup** field and enter a short name into the text box next to it. If your network already has a few SMB servers that are members of a workgroup, this server should be made a member too.
3. If your network already has a WINS protocol server, select **Use server** in the **WINS mode** field and enter its IP address. If not, you should choose **Be WINS server** so that Windows clients can use your system to lookup IP addresses for SMB server names. More recent versions of Windows (and Linux clients) do not need to use WINS, as they can look up server names in the DNS—assuming your network has a DNS server that has entries for all your hosts.
4. To set a description for your system, fill in the **Server description** field with something like *Corporate file server*.
5. Normally, Samba will use the first part of your system’s DNS name as the SMB server name. To change this, enter something else in the **Server name** field. Clients will be able to refer to this server by whatever name you specify.
6. To define alternate names that clients can use to refer to your server, fill in the **Server aliases** field with a space-separated list of names.
7. If you want your system to be the master browser for a network (the server that maintains lists of other SMB servers and clients on the network, as seen in Window’s network neighborhood), change the **Master browser?** field to **Yes**. If you are running multiple Samba servers on the same subnet, this option should be set for only one.
If there are other Windows or Samba servers on the network that want to be master browsers, the one with the highest operating system level will win the “election” that decides who gets the job. You can increase your system’s chance of winning by increasing the **Master browser priority** field—the default of 20 will win against Windows 95 systems, but you would need to enter 65 to beat Windows NT servers.
8. To have your Samba server contact another SMB server to validate passwords instead of checking its own user list, select **Password server** from the **Security** menu and enter the other server’s hostname or IP address in the **Password server** field. Otherwise, leave the field set to **Default** or **User level**. **Share level** security is rarely used anymore with modern clients, and **Domain** security is too advanced to cover in this chapter.
9. Normally, an SMB server broadcasts information about itself to other servers on the network so that it can be included in browse lists. If your network spans multiple subnets, however, then broadcasts from one system may not reach others. To get around this problem, the **Remote announce to** table can be used to specify the addresses of browser master servers to which this server’s IP address and workgroup should be sent.
To configure remote announcements on this page, first select the **From list** option above the table. Then, in the **IP address** field of each row, enter the hostname or IP address of a server to announce to and the name of the workgroup that your server should appear under in the **As workgroup** field. If the second field is left empty, the server’s real workgroup (set in Step 2) will be used. To enter more than two remote servers, you will need to save and reopen this page so that more empty rows appear in the table.
10. Finally, click the **Save** button to activate the new network settings.

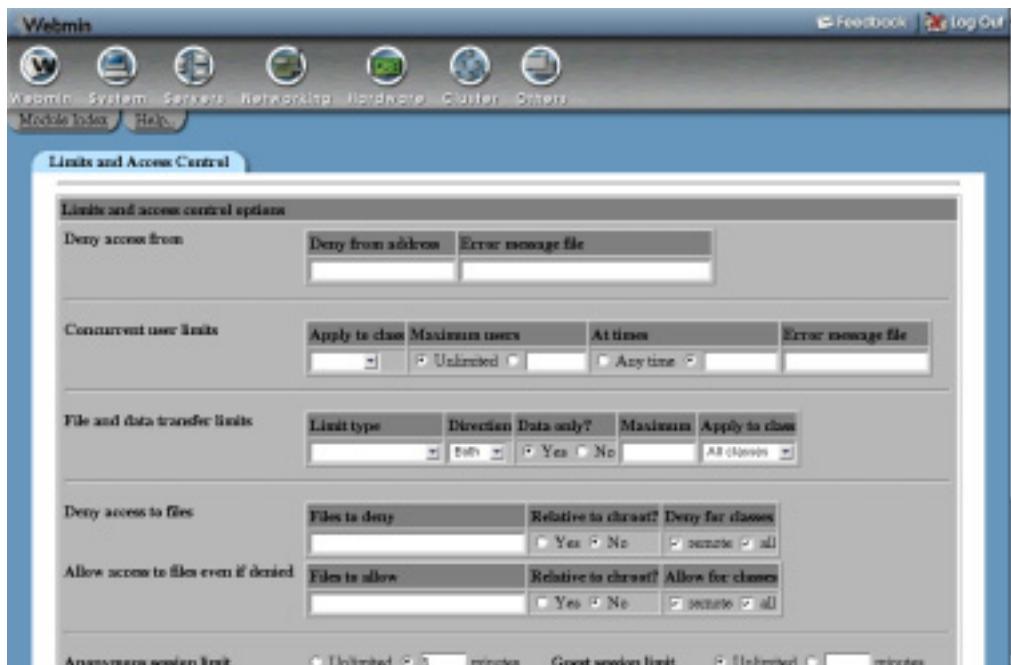


Figure 43.4 The Windows networking options form.

Samba also has numerous global options related to networking that control such things as the IP address to listen on, whether to send keep-alive packets, and how long clients can be idle before they are disconnected. These can be used to tune your server's performance, or limit access to only clients on a local network. To edit them, follow these steps:

1. Click on the **UNIX Networking** icon on the module's main page.
2. To have Samba disconnect clients that have been inactive too long and do not have any files open, select the second radio button in the **Idle time before disconnect** field and enter a number of minutes into the adjacent text box. If **Never** is selected, clients will never be cut off automatically. Because Samba starts one server subprocess per client, this feature is useful for cutting down the amount of memory that they use up. Windows clients will automatically reconnect if disconnected, so there is no down side to using it.
3. To have Samba send packets to detect if clients have crashed without properly disconnecting, select the **Send every** option in the **Keepalive packets** field and enter the number of seconds (such as 60) that a packet should be sent. Because clients can hold locks on files, a dead client may end up locking a file that other people need access to, even though the file is clearly not being used.

The same thing can be achieved by selecting the **SO_KEEPALIVE** checkbox in the **Socket options** field. This tells the operating system kernel to do basically the same thing, and should be the preferred method. The only problem is that you cannot specify the keep-alive packet interval.

4. To restrict Samba to only listening for connections on just one of your system's IP addresses, fill in the **Listen on address** field. On a machine with one interface connected to an internal network and one connected to the Internet, this feature can be used to prevent outsiders connecting to your Samba server.
5. Hit the **Save** button at the bottom of the page to activate the new network settings.

As you will see when you look at the actual form, there are many more fields on it than those documented above. The rest have extremely specialized uses, however, and thus do not need to be touched by the average administrator.

43.14 Configuring Authentication

The SMB protocol allows users to change their passwords for a server from a client system. For a Samba server, this causes the encrypted passwords file to be updated, assuming one is in use (as is usually the case). You can also configure the server to change the user's UNIX password as well, which makes sense if they are being kept synchronized.

Another authentication-related feature supported by Webmin is username mapping. This allows you to map fake client login names to real UNIX usernames, and can be useful if users prefer to use their full names to log in (like *Jamie Cameron* instead of *jcameron*) or if you have a client that is regularly moved between two different networks, each of which has different SMB accounts.

To set these global authentication options using this module, follow these steps:

1. On the module's main page, click on the **Authentication** icon.
2. As explained in Section 43.3 “Managing Samba Users”, the **Use encrypted passwords?** field determines if Samba uses its own separate password file or the standard UNIX user database. Because all recent versions of Windows use a password encryption format that is incompatible with the UNIX format, this field should generally be set to **Yes**.
3. To allow logins by users who have no password set, select **Yes** for the **Allow null passwords?** field.
4. The **Password program** field sets the program that Samba will use to change a user's UNIX password if synchronization is enabled. If **Default** is selected, `/bin/passwd` will be used, which is correct for most UNIX systems. You can enter a different command by selecting the second radio button and filling in the text box with something like `/usr/bin/ypasswd %u`. The `%u` code is replaced with the name of the user whose password is being changed, and is required because the command is run as `root`.
5. To have Samba change a user's UNIX password when his SMB password is changed over the network, set the **Change UNIX password as well?** field to **Yes**. Synchronization in the other direction is unaffected, though (see Section 43.3 “Managing Samba Users” for more details on how that works).
6. To define “fake” SMB accounts, select **Listed below** in the **Username mapping** field. In the table below it, each row specifies a mapping—the first field must contain a valid UNIX username, and the second an SMB login name of your choice. Clients logging in with one of these made-up account names must of course provide the correct password for the associated UNIX user.
7. Hit the **Save** button at the bottom of the page to activate your new authentication settings.

43.15 Configuring Printers

If you are sharing printers from your server, you will probably need to adjust the global printing options. These determine the print system commands that Samba will use to submit, list, and delete jobs, the file it gets the list of printers from, and other related settings. To edit them, the steps to follow are:

1. Click on the **Windows To UNIX Printing** icon on the module's main page to bring up the printer options form.
2. From the **UNIX print style** menu, select the type of print system in use on your box. Unfortunately, practically every different flavor of UNIX has its own set of programs and configuration files for handling printers and print drivers, each of which must be treated differently by Samba. The options that you may want to select from are:

BSD The traditional UNIX print software, found on FreeBSD, NetBSD, and older Linux distributions.

SYSV The print system used on Solaris, UNIXWare, and a few other versions of UNIX.

HPUX The print system shipped with HP/UX.

AIX The print software that comes with AIX—IBM's version of UNIX.

CUPS The superior CUPS print system, which is included with many new Linux distributions.

LPRNG An improved version of the old BSD print system used on all Linux systems that do not run CUPS.

Most packages of Samba will have this option set correctly in the default configuration file. Chapter 22 “Printer Administration” explains in more detail what the differences between the various print systems are, and how to select the right one for your operating system.

3. Normally, Samba will find all the printers on your system and make them visible to clients when the special `printers` share exists. To disable this, change the **Show all printers?** field to **No** instead. The printers will still be accessible using an explicit `\servername\printernname` path.
4. When the **Printcap file** field is set to **Default**, Samba will get the list of printers available on your system from the standard `/etc/printcap` file. This is fine if you want them to all show up, but sometimes you want to hide printers from users. To do this, create a fake printers file that looks like:

```
printer1|Description for printer 1:  
printer2|Description of second printer:
```

Set this field to the path for this file. Only the printers listed in it will be available automatically when a `printers` share exists.

5. Samba caches the output from whatever command is used to list waiting print jobs (such as `lpq`) in order to reduce the frequency with which it is run. By default, this cache time is 10 seconds, but you can increase or decrease it using the **Printer status cache time** field. If your `lpq` command is very slow, you may want to increase it.

6. Hit the **Save** button to activate your new printing settings.

43.16 Accessing SWAT from Webmin

SWAT (which stands for Samba Web Administration Tool) is a program similar to Webmin's Samba module, but included as standard with Samba itself. It provides a web-based administration interface to the configuration file that allows you to create and edit shares and global settings. SWAT is usually run from a super server like `inetd` or `xinetd`, which makes it appear as a web server—usually on port 901. Most Samba packages for Linux include the program and a `xinetd` service for it but have it disabled by default.

You can also access SWAT through this Webmin module, instead of setting it up to run from `inetd` or `xinetd`. Unfortunately, this will not work if Samba has been configured to only allow connections from some IP addresses (using the **Hosts to allow** field) on the global **Security and Access Control** page. Because SWAT also uses Samba's configuration files, any IP restrictions that apply globally will apply to it as well. When it is run from Webmin, SWAT is unable to determine the connecting IP address and fails if any IP restrictions are in force.

Assuming that this is not the case on your system, you can use it by following these steps:

1. On the module's main page, click on the **SWAT** icon. If this is the first time you have done so, the **SWAT Login** form will appear.
2. If you do get the login form, enter `root` in the **Username** field and the `root` user's password in the **Password** field. Because it can be used to totally reconfigure your Samba server, SWAT requires users to authenticate first. Webmin will remember the login and password that you enter so that the login form will be bypassed in the future.
3. After you have logged in, the SWAT main menu will appear. Along the top are icons for various parts of the Samba configuration, which, when clicked on, bring up forms for editing settings. Some things that can be configured in Webmin cannot be in SWAT, and vice versa. So, you may find it superior to the Webmin interface for some tasks.
4. To return to the module's main page, click on the **Return to share list** link at the bottom of any page. To have Webmin forget your SWAT password so that you can log in again, hit the **Logout of SWAT** link in the bottom-right corner.

Be aware that when you log in to SWAT through this module, the username and password that you enter are stored unencrypted in the file `/etc/webmin/samba/swat`, which is readable only by `root`. If this bothers you for security reasons, either do not use this feature of the module or remember to click the **Logout of SWAT** link after you are done using it so that the window of exposure is limited.

43.17 Module Access Control

As Chapter 52 explains, once a Webmin user has been granted access to a module he can be further restricted to only a subset of its functions. For the Samba module, you can allow a user to edit only certain types of settings in certain shares while denying him the ability to create new shares or edit global options. This can be useful if you want to let someone edit the settings that apply to the sharing of only his own directory, while protecting the rest of the Samba server's configuration.

I would advise against granting even limited access to this module to untrusted users, however, as it has many features that could be used by a malicious user to gain `root` access to your

system. For example, someone could allow guest access to a share with `root` permissions, allowing the remote modification of any file. Or they could set the command that is run as `root` at client connection time to something that changes the `root` password.

Instead, these access control features are should only be used to limit the changes that an inexperienced—but still trusted—user can make. To restrict such a user to only editing a few shares, follow these steps:

1. In the Webmin Users module, create a user with access to the module, or modify an existing user to give him access.
2. Click on Samba Windows File Sharing next to the name of the user to bring up the module access control form.
3. Change the **Can edit module configuration?** field to **No**.
4. Set all the fields from **Can apply changes?** down to **Can maintain auto UNIX to SAMBA users sync?** to **No** as well, as they control access to global settings that the user should not touch.
5. To hide shares that he cannot access from the user, change the **Hide inaccessible objects?** field to **Yes**. Leaving it set to **No** lets him see other shares. If he tries to click on any of them, however, an error message will appear.
6. In the **Access file shares** field, deselect **create** but leave **read** and **write** selected. Do the same for the **Access print shares** field. This does not mean that he can edit all shares. Later fields control exactly which ones he can configure.
7. Change the **Enable per-file_share acls?** and **Enable per-print-share acls?** fields to **Yes**, so that the options set in the next step are used.
8. In the **Per-share ACLs** table, select **n/a** under **Access share** and **Connections** for all the shares that he should not be allowed to configure. You should definitely do this for the **global** share as well, as it sets the defaults for all others.
For the shares that you do want the user to manage, select **read write** in the **Access share** column. To allow the user to kill clients connected to this share, select **kill** in the **Connections** column. Or, to let him only see connected clients, choose **view** instead. The former option is not a good idea in terms of security, however, as it allows the user to terminate any process on your system.
The radio buttons in the **security**, **permissions**, **file naming**, and **miscellaneous or printer** columns control the sub-icons on the share editing form to which the user has access. For each sub-icon, you can choose either **edit** to allow editing, **view** to only let him look at the settings, or **n/a** to deny access altogether.
9. Hit the **Save** button at the bottom of the page to activate the new access control settings.

43.18 Configuring the Samba Windows File Sharing Module

The module assumes that you have installed the Samba package available for your operating system or Linux distribution, or have compiled Samba from source code if no such package is available. If this is not the case (for example, if you have compiled the latest version instead of using a package), the paths that it uses for the Samba programs and configuration files will be wrong. This will cause the module's main page to incorrectly display an error message about Samba not being installed.

Fortunately, these paths can be easily changed by clicking on the standard **Module Config** link in the top-left corner of the main page. If you follow this link on the form that appears, there are fields that control the module's user interface (under **Configurable options**) as well as the fields for configuration file and program paths (under **System configuration**). The first group of settings can be safely changed at any time, but those that set paths do not generally need to be adjusted as the defaults are usually correct.

Table 43.1 lists the available configuration fields and their meanings.

Table 43.1 Module Configuration Options

List of UNIX users not to add to the Samba password list	On the form for converting UNIX to Samba users (covered in Section 43.3 “Managing Samba Users”), there is a field listing users not to convert. Its default values are taken from this configuration field, so if you find yourself always adding certain users to exclude them from conversion, you may want to add them here.
Sort users and groups by name	When Yes is selected for this field, the list of Samba users will be sorted by login name. If No is chosen, they will be listed in the order that they were converted or added, instead.
Location of the Samba configuration file	This field must contain the full path to the Samba configuration file, <code>smb.conf</code> , such as <code>/etc/smb.conf</code> or <code>/usr/local/samba/lib/smb.conf</code> .
Location of the Samba password file	For the module to be able to edit and create Samba users, this field must contain the full path to the text file in which they are stored. It is called <code>smbpasswd</code> , and is usually found in <code>/etc</code> or <code>/usr/local/samba/private</code> . Do not enter the path to the <code>smbpasswd</code> program!
Full path to smbstatus	In this field you must enter the path to the <code>smbstatus</code> program, which displays the list of connected users.
Full path to smbpasswd	This field must contain the full path to the <code>smbpasswd</code> program, used for setting users' encrypted passwords.
Full path to smbd	This must contain the location of the <code>smbd</code> server program, such as <code>/usr/sbin/smbd</code> .
Full path to nmbd	Similarly, this field must contain the location of the <code>nmbd</code> server program, usually found in the same directory as <code>smbd</code> .

Table 43.1 Module Configuration Options (Continued)

Full path to swat	If you want to be able to use SWAT from within this Webmin module, this field must contain the full path to the <code>swat</code> executable. If it is incorrect or None is chosen, the rest of the module will still work, but no SWAT icon will appear on the main page.
Full path to smbgroupedit	If you are running Samba version 3 which supports Samba groups as well as users, this field should be set to the path to the group editing command. This will enable the module's group management and synchronization features, which is not covered in this chapter as Samba 3 is still under development.
Command to start Samba servers	This field determines what command is run when the Start Samba Servers button on the module's main page is clicked. It is also used when a user clicks the Restart Samba Servers to restart Samba after the <code>stop</code> command set in the next field is run. If None is selected the module will just run <code>smbd</code> and <code>nmbd</code> , which works fine and should be used if you have compiled and installed Samba from the source code. Some Linux distribution packages include a bootup script to start the servers. On those systems, this field is set to something like <code>/etc/init.d/samba start</code> by default. Of course, this will not work if you have not actually installed the packaged version.
Command to stop Samba servers	Like the previous field, this field determines what command is used to stop Samba when the Restart Samba Servers button is selected. You can either select None to have the module simply kill all of the <code>smbd</code> and <code>nmbd</code> server processes, or enter a command like <code>/etc/init.d/samba stop</code> to have it executed, instead. The first option should be used if you have compiled and installed Samba from the source code, as no such script is likely to exist.

43.19 Summary

Samba is one of most useful servers ever developed, as it allows any UNIX system to act as a file or print server for Windows clients. After reading this chapter, you should understand how to set up Samba so that modern client systems can log into it, and how to create new user accounts. You should also understand how to specify which directories and printers are made available, and how to set options such as IP access control or guest access that apply to some or all shares.

Configuring the Squid Proxy Server

This chapter explains what an HTTP or FTP proxy server is, and explains how you can use Webmin to configure the popular Squid proxy program.

44.1 Introduction to Proxying and Squid

An HTTP proxy server is basically a program that accepts requests from clients for URLs, fetches them, and returns the results to the client. Proxies are used on networks where clients do not have direct access to the Internet but still need to be able to view web pages, and for caching commonly requested pages so that if more than one client wants to view the same page it only has to be downloaded once.

Many companies and organizations have their firewalls set up to block all incoming and outgoing traffic by systems on internal LANs. This may be done for security reasons or to limit what employees can access on the Internet. Because being able to view web pages is extremely useful, a proxy is often set up so that websites can be accessed through it.

Large organizations and ISPs with many client PCs accessing the web at the same time may also want to run a proxy server to reduce the load on their networks. Because one of the main tasks of a proxy is caching pages requested by clients, any page asked for more than once will be returned from the cache instead of being fetched from the originating server. For this reason, clients systems are often recommended or forced to use a caching proxy to access the web.

A proxy is only useful if client browsers are configured to use it instead of connecting to websites directly. Fortunately, every browser in existence, and almost all programs that download files via HTTP for various purposes, can be configured to use a proxy. This configuration tells them to make a special HTTP connection to the proxy server instead, specifying the complete URL to download.

Proxies are not just for HTTP—they can also support FTP and Gopher protocol requests from clients, which they service by making a FTP or Gopher connection to the actual requested server. Even encrypted SSL connections can be handled by a proxy, even though it cannot decrypt the request. Instead, the proxy simply forwards all data from the client to the destination server and back again.

Squid is the most popular proxy server for UNIX systems. It is freely available for download from www.squid-cache.org and is included as a standard package with all Linux distributions and many other operating systems. Squid supports proxying, caching, and HTTP acceleration and has a large number of configuration options to control the behavior of these features.

Squid reads its configuration from the text file `squid.conf`, usually found in or under the `/etc` directory. This file consists of a series of directives, one per line, each of which has a name and value. Each directive sets some option, such as the TCP port on which to be listed or a directory in which to store cached files. Webmin's Squid module edits this file directly, ignoring any comments or directives that it does not understand.

Many versions of Squid have been released over the years, each of which has supported different configuration directives or assigned different meanings to the same directives. This means that a `squid.conf` file from version 2.0 may not be compatible with Squid 2.5, and one from Squid 2.5 certainly will not work with version 2.0. Fortunately, Webmin knows which directives each release supports and only allows editing of those that are known to the running version of Squid.

Cached web pages are stored in files in a multi-level directory structure for increased file-system performance. Squid can be configured to use multiple separate cache directories so that you can spread files over different disks to improve performance. Every time a cacheable page is requested, it is stored in a file so that when a subsequent request for the same page arrives the file can be read and the data served from it. Because some web pages change over time (or are even dynamically generated), Squid keeps track of the last-modified and expiration dates of web pages so that it can clear data from the cache when it is out of date.

The actual program that handles client requests is a permanently running server process called `squid`. It may also start several other subprocesses for tasks such as DNS lookups or client authentication, but all the actual HTTP protocol processing is done in the single master process. Unlike other similar servers, such as Apache or Sendmail, Squid does not start or use subprocesses to handle client requests.

Squid can be compiled on all the flavors of UNIX that Webmin supports, and works almost identically on all of them. This means that the Webmin module's user interface is the same across operating systems as well, with the exception of the default paths that it uses for the Squid programs and configuration files.

44.2 The Squid Proxy Server Module

If you want to set or up configure Squid from within Webmin, you will need to use the Squid Proxy Server module, found under the Servers category. When its icon is selected, the page shown in Figure 44.1 will appear, assuming that Squid is installed and configured correctly. As you can see, the main page consists only of a table of icons, each of which can be clicked on to bring up a form for editing settings in some category.

If you have not configured or started Squid on your system before, the cache directory has probably not been set up yet. The module will detect this and display a message like **Your Squid**

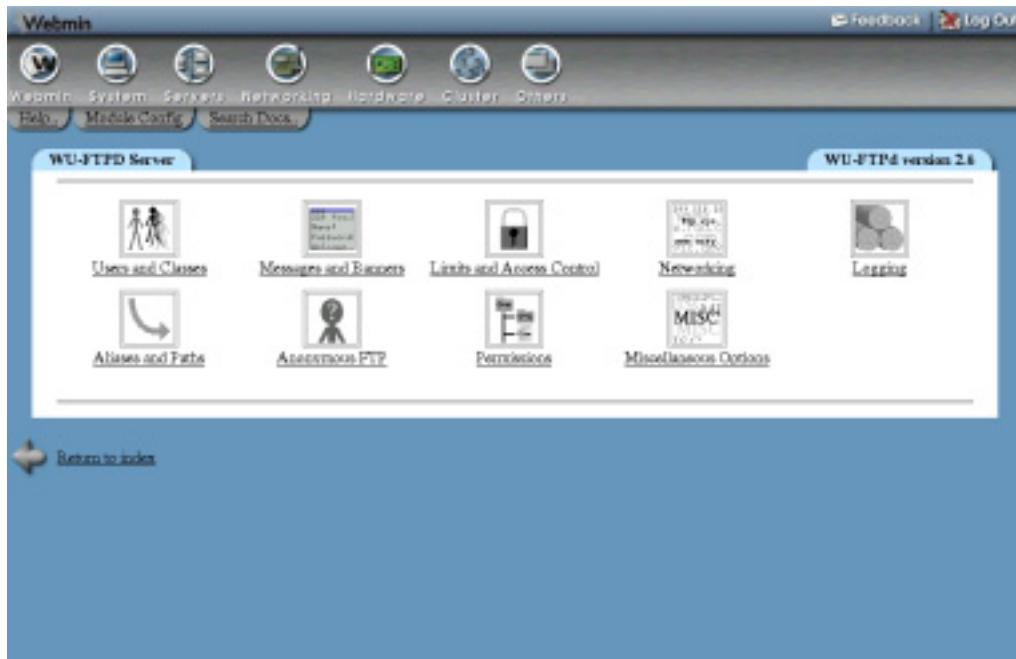


Figure 44.1 The Squid module main page.

cache directory /var/spool/squid has not been initialized above the table of icons. To initialize the cache, follow these steps:

1. If you are unhappy with the displayed cache directory, now is the time to change it. Follow the instructions in Section 44.4 “Adding Cache Directories” to define your own directories before continuing.
2. In the **as Unix user** field, enter the name of the user who will own the cache files and as whom the daemon process will run. Typically, this will be a special `squid` user created for this specific purpose (and the field will default to `squid` if such a user exists), but any user will do. I recommend, however, using the Users and Groups module (covered in Chapter 4) to create a user called `squid` whose home directory is the cache directory, if needed.
3. Hit the **Initialize Cache** button. The Squid configuration will be updated to use your chosen username, and the command `squid -z` will be run to set up the cache directories. All output that it produces will be displayed so that you can see how the initialization is progressing.
4. When the process is complete, return to the module’s main page and the error message should have disappeared.

If Squid is not installed at all on your system (or installed in a different location from the one Webmin expects), an error message like **The Squid config file /etc/squid.conf does not exist** will appear on the main page instead of the table of icons. If you do have it installed, read Section 44.17 “Configuring the Squid Proxy Server Module” for instructions on how to change the paths the

module uses. If it is not installed, you should use the Software Packages module (covered in Chapter 12) to install the `squid` package from your Linux distribution CD or the source website.

If no such package exists for your operating system, you will need to download, compile, and install the latest version of Squid from www.squid-cache.org. As long as you have a compiler installed on your system, this is a relatively simple process with no dependencies.

Once the server is installed, if you want to make use of Squid in the long term, you should arrange to have it started at boot time using the Bootup and Shutdown module (covered in Chapter 9). All Linux packages include a bootup action script for Squid, although it may be disabled by default, thus requiring you to enable it in that module. Otherwise, you will need to create an action that runs a command like `/usr/local/squid/bin/squid -sY`, assuming that you have Squid installed in `/usr/local/squid`.

Once Squid has been installed and initialized, you can start using this module. When Squid is running, every page has two links at the top: **Apply Changes**, which forces the current configuration to be reread, and **Stop Squid**, which shuts down the proxy server. If the server is not running, those links are replaced with **Start Squid**, instead, which (as the name suggests) attempts to start it. If it is not yet running, you will probably want to start it at this time.

Because each version of Squid has introduced new configuration directives, this module's user interface will appear differently depending on the version of Squid that it detects on your system. All of the instructions in this chapter are written for Squid 2.4, as it is currently the most widely deployed version. If you are running an older or newer release, different fields may appear on the forms or may have more or fewer options. For example, each new version has introduced different ACL types and authentication has been handled in three different ways through the history of the program. The basic concepts, however, have always been the same.

When you are using this module, make sure your browser is configured not to use the Squid proxy to access your Webmin server. Otherwise, you run the risk of cutting off your own access to the module if you make a configuration mistake or shut down the server process. All browsers that can use a proxy have a field for listing hosts to which to directly connect, into which you can enter the hostname of your Webmin server.

44.3 Changing the Proxy Ports and Addresses

By default, Squid listens for proxy requests on TCP port 3128 on all of your system's IP addresses. Because this is not the usual port on which proxies are run (8000 and 8080 seem to be the most common), you may want to change it. If your system has more than one network interface, you might also want to edit the listening address so that only clients on your internal network can connect.

To specify the ports that Squid uses, follow these steps:

1. On the module's main page, click on the **Ports and Networking** icon to bring up the form shown in Figure 44.2.
2. In the **Proxy addresses and ports** table, select the **Listed below** option. In the table provided, each row defines a listening port and an optional address to which to bind. Any existing ports and addresses will be listed, followed by a single blank row for adding a new one. In the first empty field in the **Port** column, enter a port number like `8000` or `8080`. In the **Hostname/IP address** column, either select **All** to accept connections on

any of your system's interfaces or select the second option to enter an IP address in the adjacent text box.

Using this table, Squid can be configured to listen on as many ports as you like. Because only one blank row appears at a time, however, you will need to save and reopen the form to add more than one new port.

3. ICP is a protocol used by Squid to communicate with other proxies in a cluster. Fill in the **ICP port** field to listen on a port other than the default of 3130 for ICP. This is not generally necessary, however, as only other proxies ever use this protocol.
4. Squid will normally accept ICP connections on any IP address. To change this, select the second radio button in the **Incoming UDP address** field and enter one of your system's interface IPs into its text field. This can be useful if all of the other proxies that your server might want to communicate with are on a single internal LAN.
5. Hit the **Save** button at the bottom of the page to update the configuration file with your new settings, then hit the **Apply Changes** link back on the main page to activate them.



Figure 44.2 The ports and networking form.

44.4 Adding Cache Directories

In its usual default configuration, Squid uses a single directory for storing cached pages. At most, 100 MB of data will be stored in this directory, which is not likely to be enough if serving a large number of active clients. If your system has more than one hard drive, it makes sense to spread the cache across multiple disks to improve performance. This can be done by specifying multiple directories, each with its own maximum size.

On a system that is dedicated to running a proxy server, the maximum amount to cache in each directory should be about 90 percent of the available space. It is unwise to configure or allow Squid to use up all free disk space, as many filesystems suffer reduced performance when nearly full. Furthermore, disk space may be used by log files and user data as well. If Squid fills up your entire hard drive, problems may occur because other programs are unable to create temporary files or write to logs.

To add a new cache directory and specify the maximum size for the existing one, follow these steps:

1. Click on the **Cache Options** icon on the module's main page to bring up the form shown in Figure 44.3.
2. In the **Cache directories** field, select the **Listed** option. If **Default** was chosen before, Squid will have been using the single compiled-in default cache directory displayed in brackets. If you want to continue using this directory, it must be explicitly entered into the table. The default size is 100 MB, and it uses 16 1st level and 256 2nd level directories.

Each row in the table specifies a single cache directory. Any existing directories (apart from the default) will be listed so that you can edit them, followed by a single blank row. Each row has fields under the following columns:

Directory The full path to the top-level cache directory, such as */var/spool/squid* or */disk2/cache*. This directory must already exist and be owned by the user as whom Squid runs (usually called *squid*). The module will not create it for you.

Type The storage type used in the directory. You should always select **UFS** here.

Size (MB) The maximum amount of data that it will contain, in megabytes. Once this limit is reached, the oldest unrequested files will be replaced with new ones.

1st level dirs The number of subdirectories that will be created under the cache directory. The default of 16 is usually fine, but you may want to increase this for very large caches.

2nd level dirs The number of subdirectories that will be created under each first-level directory. You should just enter 256, unless your cache is going to be very large.

Options Leave this field blank—it is only used for other directory types.

If you are wondering why Squid needs to create two levels of subdirectories under each cache directory, the reason is the poor performance of many filesystems when a directory contains a large number of files. Because every single cached HTML page or image is stored in a separate file, the number of files on a busy proxy system can be huge. Spreading them across multiple directories solves this problem.

3. After adding a directory, hit the **Save** button at the bottom of the page. If you want to add more than one, you will need to click on the **Cache Options** icon again to redisplay the table with a new empty row.
4. When you are done defining directories, return to the module's main page. If a new one has been added, an error message like **Your Squid cache directories have not been initialized** will be displayed. Hit the **Initialize Cache** button to have Squid create all the subdirectories in any new cache directories. The server will be shut down during the process and re-started when it is complete.

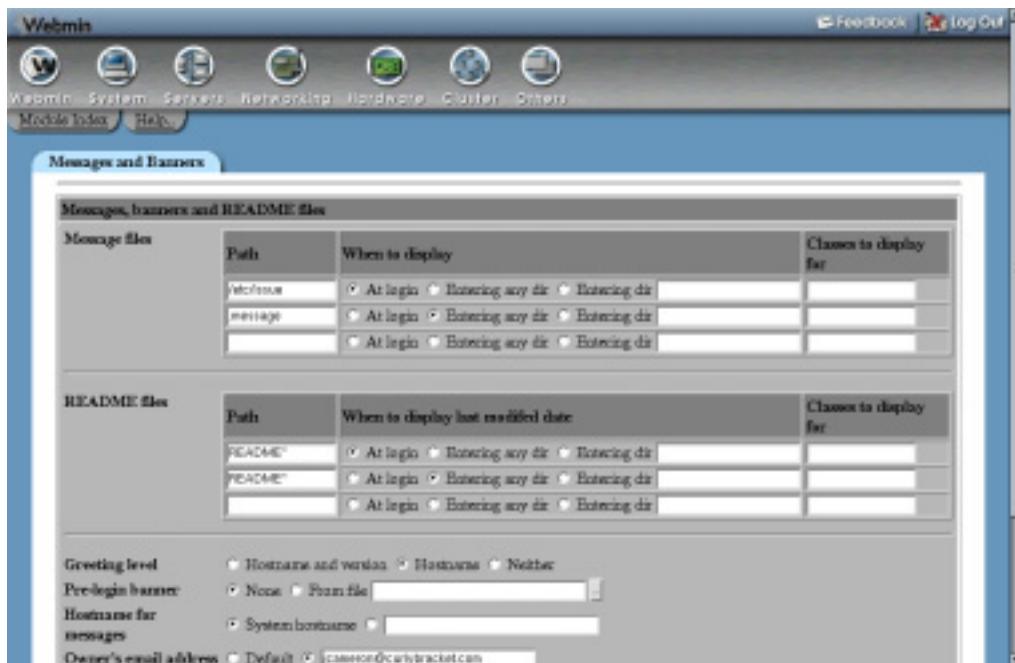


Figure 44.3 The cache options form.

5. After initialization is complete, click on the **Apply Changes** link on any page to start using your new directories.

44.5 Editing Caching and Proxy Options

Squid has numerous settings that limit the size of cached objects, the size of client requests, and the types of pages to cache. They can be used to stop the server from storing enormous files (such as downloaded ISO images), to limit the size of files that clients can upload or download, and to prevent the cache of pages that change frequently (such as those generated by CGI scripts). The defaults will generally work fine, with the possible exception of the maximum upload size, which is only 1 MB.

To edit caching options, follow these steps:

1. Click on the **Cache Options** icon on the main page to display the Cache Options form (from Figure 44.3) again.
2. To set the maximum size of uploaded files, select the second option in the **Maximum request body size** field, enter a number into the text box, and select some units from the menu. 10 or 100 MB should be more than enough for anyone.
3. To stop clients from downloading large files, fill in the **Maximum reply body size** field in the same way. This can be used to prevent the abuse of your network by clients downloading huge movies or ISO files, but can often be subverted by downloading a large file in pieces.

4. If you want to set an upper limit on the time that a page can be stored in the cache, fill in the **Maximum cache time** field instead of leaving it set to **Default**. Otherwise, data will be cached for up to a year or until the expiration date set by the originating server.
5. As well as caching downloaded files, Squid will remember error messages from servers and return them to clients that request the same page. You can change the amount of time that errors are cached by entering a number and selecting **Units** in the **Failed request cache time** field. If **Default** is chosen, errors will be cached for 5 minutes. Even this can be annoyingly long, however, if you have just fixed an error on a website.
6. Squid will cache the responses to hostname lookups to reduce the amount of DNS activity, regardless of the TTLs that the DNS servers supply. If **Default** is selected in the **DNS lookup cache time** field, responses will be remembered for 6 hours. If this seems too long for you, select the second radio button and enter your own cache time instead.
7. The **Don't cache URLs for ACLs** field can be used to completely prevent caching for certain URLs, web servers, or clients. Any request that matches one of the ACLs checked in this field will never be cached, and thus will always be fetched directly. You can use this feature to block the caching of dynamically generated pages by creating a **URL Path Regexp** ACL for .cgi or cgi-bin and selecting it here. See Section 44.6 “Introduction to Access Control Lists” for more details on how ACLs work and can be defined.
8. Hit the **Save** button at the bottom of the page to return to the main menu. Because some additional caching options are on the memory and disk usage form, click on the **Memory Usage** icon to display it
9. To limit the amount of memory that Squid will use, fill in the **Memory usage limit** field. Note that this limit only affects the maximum memory used for storing in-transit and frequently accessed files and negative responses. Because Squid uses memory for other purposes, it will certainly consume more than whatever you enter here. If **Default** is selected, a limit of 8 MB will be enforced, which is probably too low for a busy server.
10. To prevent the caching of huge files, fill in the **Maximum cached object size** field. The default is only 4 MB, so if you have plenty of disk space it should definitely be increased.
11. Hit the **Save** button at the bottom of the form and then the **Apply Changes** link on the main page to activate all of your new settings.

44.6 Introduction to Access Control Lists

ACLs (access control lists) are possibly Squid’s most powerful feature. An ACL is simply a test that is applied to a client request to see if it matches or not. Then, based on the ACLs that each request matches, you can choose to block it, prevent caching, force it into a delay pool, or hand it off to another proxy server. Many different types of ACL exist—for example, one type checks a client’s IP address, another matches the URL being requested, and others check the destination port, web server hostname, authenticated user, and so on.

The most common use of ACLs is to block connections from clients outside your network. If you run a proxy server that is connected to and accessible from the Internet, hosts outside your local network should not be allowed to use it. Malicious people often use other proxies to launder connections used for hacking, sending spam, or accessing websites into which they shouldn’t be allowed.

Because the special CONNECT proxy request can be used to connect to any port, an ACL is often used to block its use for any ports other than 443 (the SSL default). This stops users from using your proxy to connect to servers other than web servers, such as AIM, ICQ, or MSN. Similarly, an ACL can be set up to block normal HTTP requests to ports like 22, 23, and 25, which are normally used for SSH, telnet, and SMTP.

Just defining an ACL in the Squid configuration does not actually do anything—it must be applied in some way to have any effect. This section explains how to use ACLs to control which requests to your server are allowed or denied. Other sections explain how they relate to caching and accessing other servers.

When it receives a request, Squid first determines which ACLs it matches. It then compares this list of matches against a list of proxy restrictions, each of which contains one or more ACLs and an action to perform (either **Allow** or **Deny**). As soon as a restriction is found that matches the ACLs for the request, its action determines whether the request is allowed or denied. If no restrictions match, the opposite of the last action in the list is applied. For this reason, the final action in most Squid configurations is **Allow all** or **Deny all**.

ICP requests from other proxies are also checked to see which ACLs they match and are compared to a similar but different list of ICP restrictions to see whether or not they will be allowed. See Section 44.11 “Connecting to Other Proxies” for a more complex explanation of what ICP is and when it is used.

The typical default Squid configuration includes several ACLs and proxy restrictions. For security reasons, all requests from anywhere are denied by default. This means that you will need to change the restrictions list before anyone can use your proxy. Read on to find out how.

To view the lists of defined ACLs, proxy restrictions, and ICP restrictions, click on the **Access Control** icon on the module’s main page. As Figure 44.4 shows, a table of ACLs showing their names, types, and matches is displayed on the left. To the right are tables of proxy and ICP restrictions showing their actions and the ACLs that they match. The restriction tables have up and down arrows next to each entry to move them in the list, because their order matters.

Before clients can use your proxy, you will need to configure it to allow access from some addresses. To do this, follow these steps:

1. On the access control page, select **Client Address** from the menu below the list of existing ACLs. When you click the **Create new ACL** button, a form for entering matching addresses will appear.
2. In the **ACL name** field, enter a short name such as *yournetwork*.
3. In the empty field under **From IP**, enter the starting IP address in the range to allow, such as *192.168.1.0*.
4. In the field under **To IP**, enter the ending address in the range, such as *192.168.1.100*. Only clients that fall within this range will match the ACL.
5. You can also specify an IP network by entering the network address in the **From IP** field and the netmask (like *255.255.255.0*) into the **Netmask** field. To enter more than one, you will need to save and re-edit this ACL so new blank fields will appear.
6. Hit the **Save** button to add the ACL and return to the access control page on which your new ACL will be listed.
7. Click on **Add proxy restriction** below the **Proxy restrictions** table.
8. On the form that appears, select **Allow** from the **Action** field.



Figure 44.4 The access control lists page.

9. In the **Match ACLs** list, select your new *yournetwork* ACL.
10. Hit the **Save** button on this form to go back to the access control page again. The new restriction will be displayed at the bottom of the table, most likely below the **Deny all** entry.
11. Hit the up arrow next to your new restriction to move it above **Deny all**. This tells Squid to allow connections from your network and deny everyone else.
12. Finally, click the **Apply Changes** link at the top of the page. The proxy will now be usable by clients on your internal network, but no one else!

These instructions assume that you are starting with the default Squid configuration. If the proxy has already been configured to allow access from anywhere (by changing the **Deny all** restriction to **Allow all**), you should change it back again to block clients from outside your network. To learn more about the types of ACLs available and how to use them, read the next two sections.

44.7 Creating and Editing ACLs

Before you can block or allow requests from some address, to some server, or for some page, you will need to create an appropriate ACL. To do this, follow these basic steps:

1. Select the type of ACL to create from the menu below the **Access control lists** table and hit the **Create new ACL** button.
2. On the form that appears, enter a name for your new ACL in the **ACL name** field. If more than one has the same name, it will be treated as matched if any ACL with that

name matches. The name should consist of only letters and numbers, with no spaces or special characters.

3. Fill in the rest of the form as explained in Table 44.1.
4. In the **Failure URL** field, enter a complete URL to which clients who are denied by this ACL will be redirected. This allows you to define custom error pages to be displayed instead of the default Squid responses.
5. Hit the **Save** button at the bottom of the form.

Once an ACL has been created, you can edit it by clicking on its name in the list, changing the fields, and hitting **Save**. You can also delete it (if it is not in use by some proxy or ICP restriction) with the **Delete** button. As usual, the **Apply Changes** link must be used to activate any changes that you make.

Squid has an amazing number of ACL types, although not all are available in all versions of the server. Table 44.1 lists those that you can create for Squid 2.4 and explains what they do and what the fields on the creation form for an ACL of each type mean.

Table 44.1 Squid ACL Types

Type	Purpose	Fields
Browser Regexp	Checks the User-Agent HTTP header sent in the request to identify the type of browser the client system is running. Useful for blocking certain browsers or otherwise treating them differently.	Browser regexp For entering a Perl-style regular expression against which the browser identification string is matched. For example, IE 5.5 sends the identifier Mozilla/4.0 (compatible; MSIE 5.5; Windows 98; H010818).
Client Address	Checks the IP address of the client making the request against a list of network addresses or IP ranges. Often used to allow only clients from your own LAN.	As explained in Section 44.6 “Introduction to Access Control Lists”, when you create or edit an ACL of this type, a table with three columns is displayed. Each row in the table defines a matching address range or network, specified either by a starting and ending IP in the From IP and To IP fields, or a network address and netmask in the From IP and Netmask fields. By saving and re-editing a client address ACL, multiple rows may be added to the table.

Table 44.1 Squid ACL Types (Continued)

Client Hostname	Does a reverse DNS lookup of the client's IP address and compares it to a hostname or domain. Also useful for allowing clients within your network.	Domains A list of host or domain names in which to match clients, such as <code>pc1.foo.com</code> or <code>.example.com</code> .
Client Regexp	Like the Client Hostname ACL type, but tests the reverse address against a series of regular expressions instead.	Regular expressions A list of Perl-style expressions against which to check the hostname, such as <code>^.*foo.com\$</code> . If the Ignore case? box is checked, comparisons are done case-insensitively (which is what you always want, as DNS lookups are caseless).
Date and Time	This ACL type matches depending on the current time and day of the week, rather than any attribute of the client or request. Useful for blocking access to your proxy or certain sites during work hours.	Days of the week If All is chosen, the ACL will match on any day. If Selected is chosen, it will only match on those days selected from the list below. Hours of the day If All is chosen, the ACL will match at any time. Otherwise, you must enter starting and ending times in 24-hour HH:MM format in the adjacent fields. Only requests made within that range will match.
Dest AS Number	AS numbers are used to identify large networks on the Internet. ACLs of this type will match if the destination web server address is within a network with a certain AS number. Not commonly used.	AS numbers A space-separated list of network AS numbers to check to see if the destination address is in it.
Ethernet Address	Matches requests from clients with certain MAC addresses. This can be handy for allowing access from certain systems on a network with dynamic IP addresses, but is useless if there is a router between the clients and the proxy server. This ACL type, however, is only available if Squid is compiled with the <code>--enable-arp-acl</code> option.	Client Ethernet addresses A list of addresses in the usual colon-separated format to match, such as <code>00:D0:B7:1D:FB:A1</code> .

Table 44.1 Squid ACL Types (Continued)

External Auth	When an ACL of this type is in use, clients are forced to log in to the proxy and their usernames are checked against a list. See Section 44.9 “Setting Up Proxy Authentication” for more details.	External auth users If All users is selected, authentication will be required and any valid user will match the ACL. If Only those listed is chosen instead, only users entered in the text box provided will match. This can be used to give some people access to certain sites while denying everyone else.
External Auth Rexexp	This is like the external auth ACL type, but supports the use of regular expressions against which you can match authenticated usernames.	External auth users In this text box, you must enter a list of Perl-style regular expressions against which you can match usernames. If the Ignore case? box is checked, comparisons are done case-insensitively.
Maximum Connections	An ACL of this type will match when a single client has more than a specified number of concurrent connections to the proxy server. Useful for cutting down on the load that a client can generate.	Maximum concurrent requests The number of simultaneous requests above which the ACL matches.
Proxy IP Address	Matches the IP address on the proxy server to which the client is connected. Handy if your system has multiple network interfaces and you want to treat them differently—for example, by denying connections from the Internet interface.	IP address The IP or network address to compare to the local address, such as <i>192.168.1.100</i> . Netmask The netmask to apply to the IP address when matching. If you just want to specify a single IP address, enter <i>255.255.255.255</i> here.
Proxy Port	Matches the TCP port on the proxy server to which the client is connected. An ACL of this type might be useful if your proxy listens on multiple ports, one of which is used for transparent proxying.	Proxy server port A space-separated list of port numbers to which you can compare the local port.

Table 44.1 Squid ACL Types (Continued)

RFC931 User	The <code>ident</code> or RFC931 protocol can be used to identify the remote UNIX user connecting to your proxy, assuming that the client system is running UNIX and has the <code>ident</code> daemon enabled. An ACL of this type can be used to allow certain remote users, but is only useful if you control or trust the client systems.	RFC931 users In this text box, you can enter a list of remote user-names to allow, such as <i>jcameron</i> and <i>fchan</i> .
Request MIME Type	This type of ACL matches the MIME type used in the client request. The most common ones are <code>application/x-www-form-urlencoded</code> for normal POST requests and <code>multipart/form-data</code> for file uploads.	Request MIME type The type of request that will cause this ACL to match.
Request Method	Every HTTP request includes a method, which is typically one of the following: GET Used for normal page requests or form submissions. POST Used only for some form submissions. CONNECT Used to open a direct connection to some remote port, typically for SSL. An ACL of this type is often used to block CONNECT requests to non-SSL ports.	Request methods The checkboxes selected in this field specify the methods for which the ACL will match.
SNMP Community	This type of ACL is useful only for limiting access to Squid's SNMP agent. You should never need to create one for controlling normal proxy requests.	SNMP community string The community string that, if used, causes the ACL to match.
Source AS Number	Like the Dest AS Number ACL type, but matches based on the client's network number instead.	AS numbers A space-separated list of network AS numbers in which to check for the client address.

Table 44.1 Squid ACL Types (Continued)

URL Path Regexp	Matches depending on the path in the requested URL. The path is everything after the hostname and port, such as /images/foo.gif. Useful for detecting requests to pages dynamically generated by CGI programs.	Regular expressions A list of Perl-style regular expressions against which you can compare the URL path. The ACL is considered to match if any of the expressions do. The comparisons are case-sensitive unless the Ignore case? box is checked.
URL Port	This type of ACL matches depending on the port specified in the requested ACL. Useful for blocking access to non-HTTP ports such as 23 and 25. If no port is specified in the URL, Squid will assume the default for the protocol (80 for HTTP and 21 for FTP).	TCP ports A space-separated list of ports with which to compare the requested port.
URL Protocol	Matches depending on the protocol specified in the URL—for example, http or ftp. An ACL of this type could be used to block FTP access for some or all clients, or to non-standard ports.	URL protocols The boxes checked for this field indicate which protocols the ACL will match. The special <code>cache_object</code> protocol is used only by Squid's cache manager program.
URL Regexp	An ACL of this type matches if the entire requested URL matches any one of a series of regular expressions. Handy for blocking access to certain pages or sites.	Regular expressions A list of Perl-style expressions against which to check the URL. If the Ignore case? box is checked, comparisons are done case-insensitively—otherwise, they are case sensitive.
Web Server Address	Like the Client Address ACL type, but matches depending on the IP address of the server that the request is for. It can be used to block entire networks or specific systems hosting content that you would prefer your users not to access.	When editing or creating an ACL of this type, a table for entering a series of network addresses and netmasks is displayed. Like all tables in Webmin, it lists all existing networks followed by a blank row for adding a new one. In the first empty field under IP Address , you should enter a network or single IP like <code>1.2.3.0</code> or <code>192.168.1.55</code> . In the adjacent fields under Netmask , you must enter mask like <code>255.255.255.0</code> or <code>255.255.255.255</code> if specifying just a single address.

Table 44.1 Squid ACL Types (Continued)

Web Server Hostname	This kind of ACL compares the hostname in the requested URL to a list of host or domain names. Because it does not reverse-lookup IP addresses, it is not too useful for blocking access to sites.	Domains A list of hostnames (like <code>www.foo.com</code>) or domain names (like <code>.foo.com</code>) with which to compare the URL hostname.
Web Server Regexp	Like the Web Server Hostname ACL type, but compares the requested hostname with a series of regular expressions instead.	Regular expressions A list of Perl-style expressions against which you can match the hostname from the requested URL. You should check the Ignore case? box, as hostnames are always case-insensitive.

Many types of ACL are inappropriate for certain situations. For example, if a client sends a CONNECT request, the URL path is unavailable and thus a **URL Path Regexp** ACL will not work. In cases like this, the ACL is automatically assumed to not match.

44.8 Creating and Editing Proxy Restrictions

Once you have created some ACLs, they can be put into use by creating, editing, and moving around proxy restrictions. Squid will compare every request to all defined restrictions in order, stopping when it finds one that matches. The action set for that restriction then determines if the request is allowed or denied. This processing system, combined with the power of ACLs, allows you to set up some incredibly complex access control rules. For example, you could deny all access to sites with *quake* in the URL between 9 a.m. and 5 p.m. Monday to Friday, except for certain client addresses.

To create a proxy restriction, follow these steps:

1. Click on the **Access Control** icon on the module's main page to bring up the page shown in Figure 44.4.
2. Click on **Add proxy restriction** below the list of existing restrictions to go to the creation form.
3. From the **Action** field, select either **Allow** or **Deny** depending on whether or not you want matching requests to be processed.
4. The **Match ACLs** list can be used to select several ACLs that, if all are matched, will trigger the action. The **Don't match ACLs** field can also be used to select ACLs that must not match for the action to be triggered. It is perfectly valid to make selections from both lists to indicate that the action should be triggered only if all ACLs on the left match and if those on the right do not.

In its default configuration, Squid has an ACL called **all** that matches all requests. It can be useful for creating restrictions that allow or deny everyone, one of which usually exists by default.

5. Click the **Save** button to create the new restriction and return to the access control page.

6. Use the arrows next to it in the **Proxy restrictions** table to move it to the correct location. If your list ends with a **Deny all** entry, you will need to move it off the bottom for it to have any effect. If the list has an entry that allows all clients from your network and you have just added a restriction to deny access to some sites, you will need to move it above that **Allow** entry as well for it to be used.
7. When you are done creating and positioning restrictions, hit the **Apply Changes** link at the top of the page to make them active.

After a proxy restriction has been created, you can edit it by clicking on the link in the **Action** column for its row in the table. This will bring up an editing form identical to the one used for creating the restriction, but with **Save** and **Delete** buttons at the bottom. The former will save any changes that you make to the action or matching ACLs, while the latter will remove the restriction altogether. Again, the **Apply Changes** link must be used after updating or deleting a restriction to make the change active. If you delete all the proxy restrictions for some reason, Squid will allow all requests from all clients, which is probably not a good idea.

Also on the access control page is a table for editing and creating restrictions that apply to ICP requests. As Section 44.11 “Connecting to Other Proxies” explains, ICP is a protocol used by Squid proxies in a cluster or hierarchy to determine what URLs other servers have cached. You can add to and edit entries in the **ICP restrictions** table in exactly the same way as you would for proxy restrictions. If you are running a cluster of proxies, it may make sense to block ICP requests from sources other than your own network. If not, the default setup that allows all ICP packets is fine.

44.9 Setting Up Proxy Authentication

Even though it is possible to configure Squid to allow access only from certain IP addresses, you may want to force clients to authenticate themselves to the proxy as well. This might make sense if you want to give only certain people access to the web and cannot use IP address validation due to the use of dynamically assigned addresses on your network. It is also handy for keeping track of who has requested what through the proxy as usernames are recorded in the Squid logs.

All browsers and programs that can make use of a proxy also support proxy authentication. Browsers will pop up a login window for entering a username and password that are to be sent to the proxy the first time it requests them, and automatically send the same information for all subsequent requests. Other programs (such as wget or rpm) require the username and password to be specified on the command line.

Each login and password received by Squid is passed to an external authentication program that either approves or denies it. Typically this program checks against a separate users file, but it is possible to write your own programs that use all sorts of methods of validating users. For example, they might be looked up in a database, an LDAP server, or the UNIX user list. Webmin comes with a simple program that reads users from a text file in the same format as is used by Apache and this module allows you to edit users in such a file.

To turn on authentication for your Squid proxy, follow these steps:

1. On the module’s main page, click on the **Access Control** icon to bring up the form shown in Figure 44.4.

2. Select **External Auth** from the menu below the ACL table and hit the **Create new ACL** button.
3. In the form that appears, enter *auth* for the **ACL name** and select **All users** in the **External auth users** field. Then, hit the **Save** button.
4. Click on **Add proxy restriction** below proxy restrictions table.
5. Select **Deny** in the **Action** field and choose your new **auth** ACL from the **Don't match ACLs** list. This will block any proxy requests that are not authenticated, thus forcing clients to log in.
Selecting **Allow** and then choosing **auth** from the **Match ACLs** field can be used for a slightly different purpose. This creates a proxy restriction that allows access to all authenticated clients, which can be positioned to force clients outside your network to log in while not requiring it for those inside the network.
6. Click the **Save** button to return to the access control page again.
7. Use the up arrow next to the new restriction to move it above any entry in the table that allows all access from your own network. If it is below this entry, clients from the network will be able to use the proxy without needing to log in at all. Of course, this may be what you want in some cases.
8. Click on the **Authentication Programs** icon back on the main page.
9. From the **Authentication program** field, select **Webmin default**. This tells the module to use the simple text-file authenticator that comes with the module so that you don't have to write your own. Of course, you can specify your own custom program by selecting the last radio button and entering the full path to a script with some parameters in the adjacent text box. This program must continually read lines containing a username and password (separated by a space) as input, and for each output either the line **OK** or **ERR** for success or failure, respectively. Squid will run several instances of the program as permanent daemon processes when it is started.
10. The login window that appears in browsers includes a description of the proxy server that the user is logging into. By default, this is **Squid proxy-caching web server**, but you can enter your own (such as **Example Corporation Proxy**) by filling in the **Proxy authentication realm** field.
11. Normally, Squid will cache valid logins for one hour to avoid calling on the authentication program for every single request. This means that password changes may take up to an hour to take effect, which can be confusing. To lower this limit, at the cost of increased system load and slightly slower request processing, edit the **Time to cache passwords for** field.
12. Hit the **Save** button and then click on **Apply Changes** on the main page.

Now that authentication is enabled, any attempts to use your proxy from a web browser will cause a login window to appear. Because no valid users have been defined yet, no logins will be accepted, which is not particularly useful! To create some users for authentication, follow these steps:

1. Click on the **Proxy Authentication** icon on the module's main page to bring up a table listing proxy users. At first, this will be empty.
2. Click on the **Add a new proxy user** link above or below the table to display the user creation form.

3. Enter a login name into the **Username** field and a password for the user in the **Password** field.
4. To temporarily disable this user without deleting him, change the **Enabled?** field to **No**.
5. Hit the **Create** button to add the user and then click the **Apply Changes** link. This last step is necessary after creating a user for the changes to take effect, as Webmin's Squid authentication program only reads the user file when first started.

A user can be edited by clicking on its name in the proxy users list, changing the username, password, or enabled status, and hitting the **Save** button. You can also completely remove a user with the **Delete** button on its editing form. Again, **Apply Changes** must be clicked to make any modifications or deletions active. Squid will also cache valid passwords (as explained above) to reduce the load on the authentication program, so a password change may take some time to take effect.

The module's user management feature will only work if you choose **Webmin default** in the **Authentication program** field or if your own custom program takes the full path to an Apache-style users file as a parameter. If your program validates users against some other database or server, or if the module cannot figure out which file contains users from the command, the **Proxy Authentication** icon will not appear.

Sometimes you may want to allow normal UNIX users to log in to your program with the same passwords that they use for telnet and FTP. Even though it is possible to write a program that does proxy authentication against the UNIX user database, there is another solution—configuring the module to add, delete, and update proxy users whenever a UNIX user is created, removed, or renamed. This is most useful for keeping usernames and passwords in sync without needing to grant access to every single UNIX user. Once you have normal authentication set up as explained above, synchronization can be turned on by following these steps:

1. On the module's main page, click on the **Module Config** link in the top-left corner.
2. As their names suggest, the **Create proxy users when creating system users**, **Update proxy users when updating system users**, and **Delete proxy users when deleting system users** fields control the automatic creation, modification, and deletion of proxy users when the same thing happens to a UNIX user. For each one, you can either select **Yes** or **No**. You should probably turn on synchronization for updates and deletions, but leave it off for creations so that you can explicitly control who gets access to the proxy.
3. Hit the **Save** button at the bottom of the form to activate the new settings. From now on, actions performed in Webmin's Users and Groups module will also affect the Squid user list in the ways you have chosen. Adding a user at the command line with `useradd` or changing a password with the `passwd` command, however, will not.

See Chapter 4 for more details on how synchronization with other modules works and how to turn it on.

44.10 Configuring Logging

Squid writes to three separate log files—one for recording client access requests, one for cache events, and one for debugging information. The most useful is the access log file, which can be analyzed by a program like Webalizer (covered in Chapter 39) to generate reports on clients, requested URLs, and individual users. Logging is enabled by default to paths compiled into

Squid, and thus is dependant upon your operating system—but you can change the destinations for log files and some details of the access log format.

To configure how and where logs are written, follow these instructions:

1. Click on the **Logging** icon on the module's main page, which predictably takes you to the logging form.
2. To change the location of the client access log file, edit the contents of the **Access log file** field. If **Default** is selected, the path compiled into Squid will be used (which may be `/usr/local/squid/log/access.log` or `/var/log/squid/access.log`).
3. To change the location of the cache storage log, edit the **Storage log file** field. The default is always the `store.log` located in the same directory as the `access.log` file.
4. To change the path to which the debug log is written, edit the **Debug log file** field. Again, the default is `cache.log` located in the same directory as `access.log`.
5. Squid normally uses its own custom format for the access log. To force the use of the format used by Apache instead, change the **Use HTTPD log format?** field to **Yes**. This format may be necessary for processing by some applications, but it does not record all of the information that the default does.
6. To have Squid write resolved client hostnames to the access log instead of just IP addresses, select **Yes** in the **Log full hostnames?** field. This avoids the need to resolve them later when generating reports, but will slow down the server due to the time that reverse DNS lookups can take.
7. The `ident` or RFC931 protocol can be used to find the name of the UNIX user who is making a connection to your proxy from some remote host. Unfortunately, it is often disabled and not supported on other operating systems, so is of limited use. You can, however, configure Squid to include RFC931 user information in its access log file by selecting some of the ACLs in the **Perform RFC931 ident lookups for ACLs** field. You should ideally create a special **Client Address** ACL that matches only UNIX hosts with the `ident` daemon on your network and select only it.
If you do enable remote user lookups, the **RFC931 ident timeout** field can be used to set a maximum amount of time that Squid will wait for a response from a client. If **Default** is selected, the server will wait 10 seconds (at most) for a response before giving up (but will still allow the request).
8. Click the **Save** button at the bottom of the page to record the changes made on this form and then click the **Apply Changes** link to activate them.

Many Linux packages of Squid include a configuration file for the `logrotate` program to have the log files rotated, compressed, and eventually deleted when they become too old. If you change the paths to the log files using the instructions above, rotation will no longer be done and the logs will consume an unlimited amount of disk space. On a busy system, this could lead to a shortage of space on the logging filesystem that would be avoided if rotation were in effect.

44.11 Connecting to Other Proxies

Instead of retrieving requested web pages directly, Squid can be configured to connect to another proxy server instead and forward some or all requests to it. This feature is useful if your organization has one proxy for each department and a master cache for the entire network, and you

want to have all department proxies query the master for requests that they cannot serve from their own caches. It may also be necessary if your ISP runs a proxy server and you want to set up Squid for your home network as well, yet still make use of the ISP's cache.

By making use of ACLs to categorize requests, you can set up Squid to forward only some requests to another proxy while handling the rest normally. For example, your proxy could always handle requests for web pages on your local LAN, but still forward everything else to a master proxy cache system.

To set up your server to make use of another proxy for requests except those to a certain network or domain, follow these steps:

1. On the module's main page, click on the **Access Control** icon.
 2. Create a **Web Server Hostname** or **Web Server Address** ACL that matches the web servers that your proxy should fetch directly. Call the ACL *direct*, for example.
 3. Go back to the main page and click on the **Other Caches** icon to bring up a page containing a list of other known proxy servers (if any) and a form for setting options that control when they are used.
 4. Click on **Add another cache** to go to the cache host creation form.
 5. In the **Hostname** field, enter the fully qualified hostname of the master cache server, such as *bigproxy.example.com*. Do not just enter *bigproxy*, as Squid sometimes has trouble resolving non-canonical DNS names.
 6. From the **Type** menu, select **parent**, which tells Squid that this other proxy is at a higher level (and thus has more cached pages) than yours.
 7. In the **Proxy port** field, enter a port number that the other proxy is listening on, such as *8080*.
 8. In the **ICP port** field, enter the port that the proxy uses for ICP requests, which will typically be *3130*. If you don't know or the master proxy does not support ICP, enter *3130* anyway.
 9. Hit the **Save** button at the bottom of the page to return to the list of other caches.
 10. In the form at the bottom of that page is a section entitled **ACLs to fetch directly**, which is actually an ACL table similar to the **Proxy restrictions** table explained in Section 44.8 “Creating and Editing Proxy Restrictions”. Instead of allowing or denying requests, however, it determines which ones are fetched directly and which are forwarded to another cache.
- Use the **Add ACLs to fetch directly** link to first add an entry to allow your **direct** ACL, and then add one to deny the **all** ACL. This tells Squid to directly fetch pages from local web servers, but pass all other requests on to the chosen proxy.
11. Finally, click on **Apply Changes** at the top of the page to have Squid start using the other proxy server.

If you just want to have your proxy forward all requests to another proxy server, regardless of their destination, Step 10 in the previous instructions can be skipped completely. This works because Squid will use the other configured proxy by default if no ACLs have been set up to force direct fetching for certain requests.

On a large network with many clients, one single system running Squid may not be able to keep up with the volume of client requests. For example, a big company with hundreds of

employees all running web browsers, or an ISP that has set up a proxy for customers, could put an enormous load on a single Squid server. One solution would be to upgrade to a more powerful machine. Another would be to install Squid on multiple systems and spread the proxying load between them.

This is typically done by creating one DNS address record for each proxy system, all with the same name (such as *proxy.example.com*) but with different IP addresses. Then, when a client looks up the IP address for *proxy.example.com*, it will get back all the addresses and pick one effectively at random to which to connect. Another alternative is to install a layer four switch that can redirect traffic to the same IP address to different destinations, such as multiple proxy servers. This is more expensive (layer four switches don't come cheap), but more reliable because a server that is down can be detected and not used. If you are unfamiliar with the term, a layer four switch is one that can reroute network traffic depending on its protocol, port, and destination. In the case of HTTP requests, it can transparently redirect them to another server while leaving other types of data to be routed normally.

There is one problem with using multiple servers, however—each maintains its own cache, so if two different clients request the same web page from two different proxies it will be downloaded twice! This negates most of the benefit of running a caching proxy server.

Fortunately, Squid has features that solve this problem. It can be configured to contact other caches in the same cluster for each request, and ask them if they already have the page cached. If so, it is retrieved from the other proxy instead of from the originating web server. Because all the proxies in an organization are typically connected via a fast network, this is far more efficient. The protocol used for this inter-cache communication is called ICP and is only used by Squid.

On the module's main page, click on the **Other Caches** icon. To set up two or more proxies to talk to each other with ICP, follow these steps on each system:

1. Click on **Add another cache** to bring up the cache host creation form.
2. In the **Hostname** field enter the full hostname of one of the other caches.
3. From the **Type** menu, select **sibling**, indicating that the other cache is at the same level as this one.
4. In the **Proxy port** field, enter the HTTP port on which the other proxy listens.
5. In the **ICP port** field, enter the port number that the other proxy uses for ICP (usually **3130**).
6. Hit the **Save** button to add the other proxy and return to the other caches list.
7. Repeat Steps 2 through 7 for each of the other hosts in the cluster.
8. Finally, click on **Apply Changes** at the top of the page.

The end result should be that each proxy in the cluster has entries for all the other proxies, so that it knows to contact them for requests not in its own cache. You can, however, set up ACLs to avoid the use of ICP and force the direct fetching of certain requests, just as you can when forwarding requests to a master cache.

44.12 Clearing the Cache

Sometimes it may be necessary to remove all of the files in your Squid cache, perhaps to free up disk space or force the reloading of pages from their originating web servers.

This can be done easily using Webmin by following these steps:

1. On the module's main page, click on the **Clear and Rebuild Cache** icon. A confirmation page asking if you are really sure will be displayed in your browser.
2. To continue, hit the **Clear and Rebuild Cache** button. Because the server will be stopped during the clearing process, it should not be done when the proxy is in use.
3. A page showing Webmin's progress will be displayed as it shuts down Squid, deletes all cached files, reinitializes the directories, and finally restarts Squid. This may take quite some time if you have a large cache or are using a filesystem that is slow to delete files (such as UFS on Solaris).

44.13 Setting Up a Transparent Proxy

A transparent proxy is one that clients connect to without being aware of it, due to the use of firewall rules that redirect connections on port 80 to the proxy system. The advantage of this setup is that you do not have to manually configure all web clients to use the proxy. Instead, they will be connected to it without their knowledge. It also means that users cannot get around the cache and thus avoid its access control rules by not configuring it in their browsers.

Transparent proxying has some down sides to it, however. It is not possible to automatically capture FTP or HTTPS connections, or those to web sites on ports other than 80. It is also incompatible with proxy authentication, as clients cannot tell the difference between the proxy's request to log in and that of a website. Even though authentication may appear to work, it really does not.

Most networks have a router that connects an internal LAN to the Internet. For transparent proxying to work, this router must be configured to redirect outgoing packets on port 80 to the Squid proxy host and port instead. On a small network, the proxy can even be run on the same router host. The IPtables firewall that comes with Linux can perform both kinds of redirection using special DNAT (Destination Network Address Translation) rules in the nat table.

Because most of the work is actually done by the firewall rules that redirect outgoing packets, the instructions for setting up everything are in Chapter 19 “Firewall Configuration” in Section 19.8 “Setting Up a Transparent Proxy”. They are, however, written for Linux users who have IPtables installed. If your router is running a different operating system (or is a dedicated router, such as one made by Cisco), the steps for creating firewall rules obviously will not apply. Those rules for the Squid Proxy Server module, however, are the same no matter what kind of firewall you are running.

44.14 Viewing Cache Manager Statistics

The Squid software comes with a simple CGI program called `cachemgr.cgi` that can connect to the proxy and request statistics about memory utilization, cache hits and misses, and DNS lookup caching. Even though it is normally installed to be run from a web server like Apache, you can access it from within this Webmin module by following these simple steps:

1. On the main page, click on the **Cache Manager Statistics** icon to bring up the program's login form.
2. Leave the **Cache Host** field set to **localhost**, unless you want to connect to another proxy. Most have ACLs set up by default to deny cache manager access from anywhere except **localhost**, though.

3. In the **Cache Port** field, enter the TCP port number that your proxy is listening on, such as 8080.
4. The **Manager name** and **Password** fields can be left empty unless Squid has been configured to require authentication for retrieving statistics, which is not usually the case.
5. Hit the **Continue** button to log in, and a page listing all the various types of statistics available will appear. Click on any of the links to display the detailed information.
6. When you are done viewing cache statistics, click on the **Return to squid index** link at the bottom of the page to go back to the module's main menu.

Because Squid accepts any requests using the special `cache_object` protocol from `localhost` without authentication by default, anyone who can log in to your system via telnet or SSH could run their own version of `cachemgr.cgi` to view these statistics. Even though the information available is not particularly sensitive, you may want to set up Squid to require a username and password be supplied to access it.

This can be done by setting up external authentication and then editing the default **Allow manager localhost** proxy restriction so that the new **auth** ACL is selected in the **Match ACLs** column as well. Or better still, you can create another **External Auth** ACL that has only a few users who are allowed to view statistics listed and assign that to the proxy restriction. This is even more secure because it avoids the problem of every telnet or SSH user who also has a normal proxy login being able to access statistics.

44.15 Analyzing the Squid Logs

Calamaris is a simple Perl program that can generate a report from your Squid log files. If you have it installed, the **Calamaris Log Analysis** icon will appear on the module's main page. If not, you will need to download and install it separately, as it is not included with Squid. Some Linux distributions have a separate package for it, which can be easily installed using the Software Packages module. If not, the program can be downloaded from calamaris.cord.de/, compiled, and installed.

Clicking on the icon triggers the generation of a report from all of your Squid access logs. By default, only the last 50000 entries are processed to avoid putting undue load on the system. However, this can be adjusted on the Module Config page (as explained in Section 44.17 “Configuring the Squid Proxy Server Module”). When the report is complete, it will be displayed in your browser as a single HTML page. At the top are links to tables lower down on the page that contain summaries such as requests by host, by destination domain, and by cache hit status.

Even if you have log rotation enabled on your system to periodically rename and compress the Squid logs, the module will still include the compressed data in the report. It looks for all files in the log directory whose names start with `access.log` (such as `access.log.02.gz`) and decompresses them if necessary before feeding them to Calamaris. The newest files are always processed first, however, so that any log lines limit in force cuts off older entries rather than newer ones.

The Webalizer Logfile Analysis module (covered in Chapter 39) can also be used to generate more impressive reports, containing graphs and pie charts from the squid logs. The module can even recreate a report on schedule (such as daily) and have it written to a directory for later viewing.

44.16 Module Access Control

It can be very useful to give someone the rights to configure Squid without letting them harm or change anything else on the system. This can be done in Webmin by creating a Webmin user with access to the module and then restricting what he can do with it. Chapter 52 explains the general idea behind this kind of access control in more detail, while this section here covers restricting access to the Squid module in particular.

Some care is needed when restricting a user like this, however, as some features of the module could be used to modify files or execute commands with `root` privileges. For example, it is not a good idea to let an untrusted user change the cache directories, as setting `/` or `/etc` as a cache could damage files on the system. Features like ACL and proxy user editing are quite safe, though, and are probably the most useful to allow a subadministrator to use.

To create a user who can only configure Squid, follow these steps:

1. In the Webmin Users module, create a user or group with access to this module.
2. Click on **Squid Proxy Server** next to the user's name in the list on the main page to bring up the access control form.
3. Change the **Can edit module configuration?** field to **No** so the user cannot edit the paths to commands or the Squid configuration file.
4. In the **Allowed configuration pages** list, select those module icons that should be visible to the user. **Logging**, **Cache Options**, and **Helper Programs** should not be chosen, as those pages contain potentially dangerous options.
5. Because Squid can read ACLs values from separate files and this module allows users to edit the contents of these ACL files, you should restrict the directory in which they can be created. To do this, enter a directory belonging only to the Webmin user in the **Root directory for ACL files** field, such as `/home/joe`. Leaving it set to `/` is a bad idea, as this may allow the user to edit any file on your system as `root`.
6. To prevent the user from shutting down Squid, change the **Can start and stop Squid?** field to **No**. He will still be able to apply changes, however, and reconfigure the server so it is unusable.
7. Hit the **Save** button to activate the restrictions.

44.17 Configuring the Squid Proxy Server Module

Like most modules, this one has several settings that you can edit to configure the user interface and the paths that it uses for Squid programs and configuration files. They can all be accessed by clicking on the **Module Config** link on the main page. The user interface fields are listed under **Configurable options** on the form that appears, while those related to program paths are under **System configuration**.

Because the module's default paths match those used by the Squid package for your Linux distribution or operating system (if there is one), fields in the second group do not generally need to be edited. If you are not using the supplied Squid package because you have compiled and installed the program from the source code, however, these paths will need to be changed.

A complete list of the module configuration fields and their meanings appears in Table 44.2.

Table 44.2 Module Configuration Options

Arguments to calamaris command	This field sets the command-line arguments that will be passed to the <code>calamaris</code> program when it is run to generate a report. The default of <code>-aw</code> tells the program to include all available reports and to output in HTML format, but if this is not suitable you can change the behavior of the program by editing this field. Search for <code>calamaris</code> in the System Documentation module to get a complete list of available arguments.
Maximum log lines to pass to calamaris	Because a heavily used proxy may have millions of lines in its log files, the module only passes the last 50000 to Calamaris for analysis by default. This field can be used to edit that number or to force the processing of the entire log file when Unlimited is selected.
Encryption method for proxy passwords	When the module is used to create and edit proxy users, it assumes that the file they are in contains one user per line, in the following format: <code>username:encrypted-password</code> . This field determines the method of password encryption used. The default is crypt , which is the standard format used on UNIX systems. You can, however, select md5base64 instead to switch to MD5 encryption, which is used in the <code>/etc/shadow</code> file on most new Linux distributions. Be aware, though, that this will only work if the authentication program you have configured understands the MD5 format as well—the standard one that comes with this module does not. The only down side of the crypt format is that it is unable to differentiate passwords longer than 8 characters, and theoretically easier to crack.
Sort proxy users	If Yes is selected, the list of users on the proxy authentication page will be sorted by username. If No is chosen, they will be listed in the order in which they were added.
Create proxy users when creating system users	These fields are all related to UNIX-Squid user synchronization and are explained in Section 44.9 “Setting Up Proxy Authentication”.
Update proxy users when updating system users	These fields are all related to UNIX-Squid user synchronization and are explained in Section 44.9 “Setting Up Proxy Authentication”.
Full path to squid config file	This field must contain the full path to the Squid configuration file <code>squid.conf</code> , such as <code>/usr/local/squid/etc/squid.conf</code> .

Table 44.2 Module Configuration Options (Continued)

Delete proxy users when deleting system users	These fields are all related to UNIX-Squid user synchronization and are explained in Section 44.9 “Setting Up Proxy Authentication”.
Command to start squid	This field determines what happens when you click on the Start Squid link on any page. If Automatic is chosen, the command from the Squid executable configuration field is run with the appropriate arguments. You can, however, select the second radio button to specify some other script to be run. On Linux distributions that come with a Squid package, a command like <code>/etc/init.d/squid start</code> is used by default. This will not work, however, if you have compiled and installed the server from source instead of installing the package.
Command to stop squid	Like the field above, this one determines the command to be run when Stop Squid is clicked. If Automatic is chosen, the program from the Squid executable field will be run with the <code>shutdown</code> parameter, which signals that running server process to exit.
Squid executable	This field must contain the full path to the Squid server program, appropriately named <code>squid</code> .
Full path to PID file	For the module to determine if Squid is running or not, this field must contain the full path to its process ID file (if none is set in the configuration file).
Full path to squid cache directory	Because the usual default Squid configuration file does not specify the cache directory, this field must contain the cache path that is compiled into the program on your system. If it is incorrect, the module’s main page will keep on insisting that the cache needs to be initialized.
Squid cachemgr.cgi executable	If you want to use the module’s statistics viewing feature, this field must contain the full path to the <code>cachemgr.cgi</code> program that comes with Squid.
Full path to squid log directory	For the module’s log analysis feature to work, this field must contain the full path to the directory contain Squid’s <code>access.log</code> file, such as <code>/usr/local/squid/log</code> .
Path to calamaris log analysis program	For the Calamaris icon to appear on the main page, this field must contain the full path to the <code>calamaris</code> program. The rest of the module’s features will work fine even if it is not installed.

44.18 Summary

This chapter has explained what HTTP and FTP proxies do and has introduced the popular Squid proxy server. After reading it, you should know how to set up Squid and how to configure it to be listed on different ports, restrict access to certain pages, and require authentication by clients. You should also know how to configure the server to work with several others in a cluster to spread the load between multiple systems.

Filtering Email with Procmail

This chapter explains how to use the Procmail program and Webmin to filter and deliver email coming into your system.

45.1 Introduction to Procmail

Procmail is a powerful program for filtering and redirecting email that would normally be sent to users' mailboxes. It can be used at both the system level to filter message for all users on your system, on a per-user basis, or both. Unlike normal Sendmail aliases, Procmail can be used to deliver messages differently depending on their headers and content. This makes it an excellent tool for blocking unwanted email, such as Spam.

When installed on a system, Procmail effectively replaces the normal `mail.local` email delivery command that Sendmail and other MTAs run to append a message to a user's mail file. Even though it is most commonly used with Sendmail, other MTAs such as Qmail and Postfix can be configured to use Procmail for delivery as well. As far as the program is concerned, the actual mail server in use does not matter as long as email is passed to it properly.

Procmail's primary configuration file is `/etc/procmailrc`, which is usually managed by the system administrator. Individual users can also create their own `.procmailrc` files with the same format in their home directories. The system-wide file is always read and processed first, so any rules that it contains for redirecting messages based on their content cannot be overridden by individual users.

A Procmail configuration file is divided into actions, each of which has a series of conditions and a delivery mode. The conditions determine which messages the action matches, while the delivery mode controls what happens to those that match. Procmail will process actions in order until it finds one that matches, deliver the message as specified, and then stop processing.

The configuration file can also include variable assignments that may be used by later actions or even other variables. It can also contain special conditional sections, which are lists of

actions to be run only if some conditions are matches. In a way, these are like `if-then` statements in a programming language.

Procmail behaves pretty much the same on all UNIX-like operating systems. The only difference is the default delivery location—all Linux distributions use `/var/spool/mail` as the user mail file directory, while other UNIX variants such as Solaris use `/var/mail`. This difference, however, has no effect on the program’s configuration file format or the user interface of the Procmail Mail Filter module.

Procmail is most useful when configured by individual users to perform tasks such as sorting email from different people into different mailboxes, writing to two different mail files, or dropping email from specific addresses. The Procmail Webmin module and this chapter, however, only deal with system-wide configuration. If you want a tool that lets individual users configure the program through a web interface, Usermin (covered in Chapter 47) is the program to use. It has a module with an identical interface that manages `.procmailrc` files instead of `/etc/procmailrc`.

The global Procmail configuration can be used to have mail delivered to a different directory or in a different format to that normally used by your mail server. For example, instead of users’ mail being appended to the files in `/var/spool/mail`, it could be written to the `mbox` file in their home directories instead. Better still, Procmail can be set up to write to a Qmail-style mail directory, usually called `Maildir` and located in users’ home directories.

Because it deals only with email delivered locally on your system, Procmail cannot be used for mail filtering if you use a client program such as Mozilla or Evolution to download email from your ISP’s or company’s server. If you do not run your own mail server but still want to make use of Procmail’s features, you will need to set up Fetchmail (covered in Chapter 33) to download messages and pass them to the MTA on your system.

45.2 The Procmail Mail Filter Module

The Webmin module for managing the system-wide configuration file is called Procmail Mail Filter, and can be found under the Servers category. Clicking on its icon will take you to a main page like the one shown in Figure 45.1. All existing actions are listed and below them are links for adding new actions of various types.

Unlike other modules, this one will not complain if Procmail is not installed on your system. You should use the Software Packages module (covered in Chapter 12) to check for and install the package that comes with your Linux distribution or operating system. If no package exists, you will need to download the source from www.procmail.org, compile, and install it.

Just installing Procmail, however, is not enough for it to be actually used on your system. By default, mail servers like Sendmail, Qmail, and Postfix use their own standard mail delivery programs and not Procmail. Individual users can change this by creating a `.forward` or `.qmail` file containing the line `| /usr/bin/procmail`, which passes all incoming email to the Procmail program. It is better, however, to globally reconfigure your MTA to use Procmail so that individual users do not have to set it up. Section 45.3 “Setting Up Sendmail” explains how to configure Sendmail. Other mail servers will need to be set up differently.

45.3 Setting Up Sendmail

As long as you have the M4 files from which your primary Sendmail configuration file was built, setting up Sendmail to use Procmail is easy. Unfortunately, configuring the mail server by

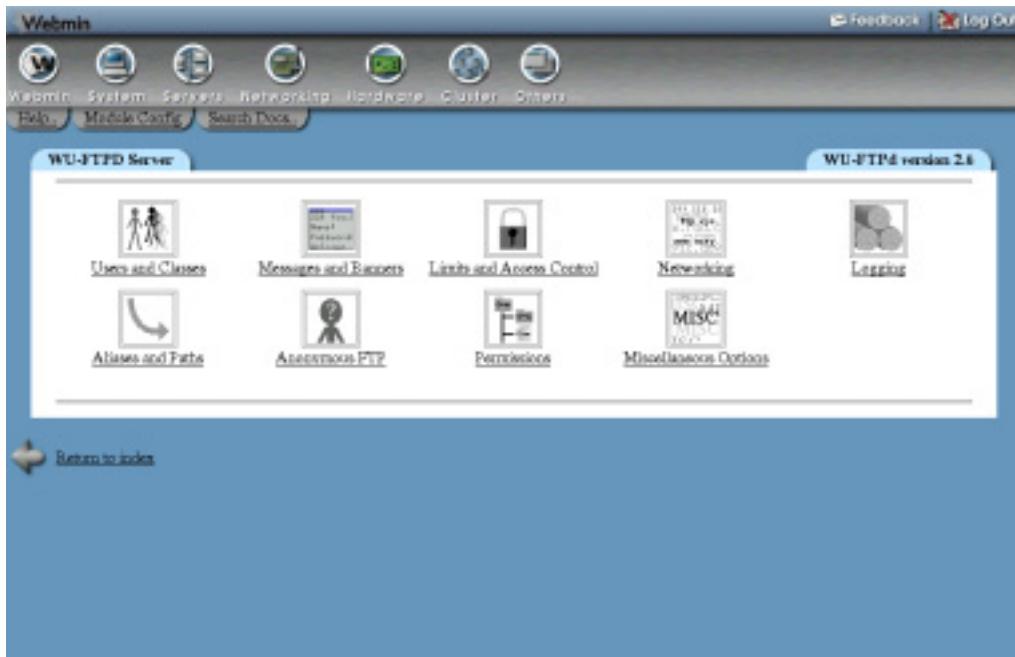


Figure 45.1 The Procmail module main page.

editing `sendmail.cf` directly is not so easy, and is not covered in this chapter. All modern Linux distributions, however, include the M4 files that you will need, in either the `sendmail` package or a separate one such as `sendmail-cf`.

To configure the Sendmail MTA to use Procmail, follow these steps:

1. Go to the Sendmail Configuration module, which can be found in Webmin under the Servers category.
2. Click on the **Sendmail M4 Configuration** icon on its main page. A list of existing M4 directives should appear. If not, the M4 files needed to reconfigure Sendmail are probably not installed on your system.
3. Check to see if the line `FEATURE(local_procmail)` already exists. If it does, delivery using Procmail is already enabled and there is no need to follow the rest of these steps.
4. From the menu next to the **Add new entry of type** button, select **Feature**, and then hit the button to display the feature creation form.
5. From the **Feature** menu select **local_procmail**. Leave the **Parameters** field empty.
6. Hit the **Create** button to have the new feature added to the M4 file. Your browser will be returned to the list of existing directives, at the bottom of which will be the new `FEATURE(local_procmail)` line.
7. Click the up arrow next to the new line as many times as is needed to move it above the `MAILER(local)` line. This is necessary because the file is processed in order, and the new directive changes the behavior of the `MAILER` line.

User classes	Class name	User types	Matching address
remote	<input type="checkbox"/> Unix <input checked="" type="checkbox"/> Anonymous <input checked="" type="checkbox"/> Guest	192.9.1.11	
all	<input checked="" type="checkbox"/> Unix <input checked="" type="checkbox"/> Anonymous <input checked="" type="checkbox"/> Guest		
	<input type="checkbox"/> Unix <input type="checkbox"/> Anonymous <input type="checkbox"/> Guest		

Unix users and UIDs to treat as guests:

- Unix users and GIDs to treat as guests
- Unix users and UIDs not to treat as guests
- Unix groups and GIDs not to treat as guests

Unix users to deny (from /etc/hosts.equiv):

- admin alias bin daemon listed lp mailerroot nobody

Unix users and UIDs to deny:

- Unix groups and GIDs to deny
- Unix users and UIDs not to deny
- Unix groups and GIDs not to deny

[Save](#)

Figure 45.2 The Procmail action creation form.

- When the new FEATURE line is in place, hit the **Rebuild Sendmail Configuration** button at the bottom of the page. A confirmation page will be displayed showing the changes that will be made to the primary Sendmail configuration file. As long as you have not been modifying sendmail.cf directly, they will be related only to the new Procmail support.
- Click on **Yes, replace it now** to have the new configuration saved and activated. From now on, all mail delivered by Sendmail to local users will be processed by Procmail. To check to see if everything worked, try sending a few test messages and make sure that they are delivered as normal.

For more information about how the Sendmail module's M4 features work and where to find the configuration files, read Section 37.11 "Adding Sendmail Features with M4".

45.4 Creating and Editing Actions

As the introduction to this chapter explains, the Procmail configuration file consists of a series of actions. When email arrives, each is checked in order until one matches and its delivery mode is carried out. If no actions match (or none exist), the email is delivered to the default destination which is usually the user's mail file under /var/spool/mail.

To create a new action, follow these steps:

- Click on the **Add a new filter action** link below the list of existing actions on the module's main page. The form shown in Figure 45.2 will be displayed for entering its destination and conditions.

2. Select the type of destination for messages that match this action from the **Delivery mode** menu. The available options are:

Append to file Email will be appended in standard mailbox format to the file entered in the adjacent text field, such as */var/spool/mail/fred*. To throw a message away, enter */dev/null* as the file.

Write to maildir Matching email will be added to the Qmail-style mail directory whose path is entered in the text field. If this directory does not yet exist, Procmail will create it (and the needed subdirectories) for the user.

Write to MH folder Email will be added to the specified MH-style mail directory. This mail format also uses one file per message, but places them in all a single directory and gives message files incrementing numeric filenames, like 1, 2, 3 and so on.

Forward to address Email will be sent to the address or user entered in the adjacent text field, such as *foo@example.com*.

Feed to program Email messages that match will be fed as input to the program whose path and arguments are entered into the text box next to the menu.

If a nonabsolute mail filename or directory (like *Mailbox* or *Maildir*) is entered, Procmail will assume that it is relative to the home directory of the user to whom the email is being delivered.

3. To have Procmail check the bodies of received messages rather than just the headers, check the **Apply conditions to body** box. This is necessary if any of the conditions you enter later need to match text in the email itself.
4. Normally Procmail will ignore the case of headers when checking conditions. To change this, check the **Case-sensitive matching** box.
5. If you want Procmail to continue on through the configuration file even if this action matches, check the **Continue processing even if conditions match** box. This can be used for delivering email to several different files or folders by turning on this option for all delivery rules except the last.
6. Procmail will normally ignore the exit status of the program to which email is fed. To have it fail (and thus bounce the message) if the program fails, turn on the **Wait for action program to finish, and check result** option.
7. If the delivery program that you entered reads in and then outputs email with some modifications, check the **Action program is a filter** box. The **Continue processing even if conditions match** option should also be enabled so that processing continues with the modified version of the message. This feature can be useful if you have written a program that checks and marks messages by adding or changing a header, which can then be checked by later actions.
8. The **Action conditions** section of the form is for entering the conditions that determine which messages will be delivered by this action. If none are specified, messages that reach the action will always be delivered, and if more than one is entered they must all match for delivery to take place.

This section is actually a table that starts out with two blank rows. The menu in each row determines the type of condition and how the text in the box next to it is interpreted.

The available options are:

Matches regular expression For this condition to match, the message headers (and possibly the body too) must match the Perl-style regular expression entered in the text box. Remember that this expression is applied to all the headers as though they were a single block of text, so you should precede any header name with a ^ to indicate the start of a line. For example, to catch messages whose subject contains the word *foo* you could enter ^Subject:.*foo.*.

Doesn't match regular expression This condition type works just like the previous one, except that it matches messages that do not match the regular expression.

Evaluate output of command The shell command entered in the text box will be run, its output read by Procmail, and then interpreted again as an action line from the configuration file. This type of condition is extremely powerful as it allows you to create dynamically generated conditions—however, for everyday mail filtering you probably don't need to use it.

Check exit status of command This type of condition matches if the shell command entered has an exit status of zero, indicating success. It can be used to deliver mail to different destinations depending on the system's hostname, the time of day, or the existence of some file.

Mail is smaller than The condition will match if the total size of the message is smaller than the number of bytes entered in the adjacent text box.

Mail is bigger than As its name suggests, this type of condition is the opposite of the previous one.

9. When you are done entering conditions, hit the **Save** button. The new action will be added to the list on the main page, and will start being used on incoming email. To add more than two conditions you will need to re-edit the action so that two more empty rows appear in the **Action conditions** section.

An existing action can be edited by clicking on its entry in the **Action to take** column on the module's main page, which brings up an editing form similar to the one in Figure 45.2. From here you can make changes and then hit **Save** to activate them, or just hit **Delete** to remove the action altogether.

Because the ordering of actions matters, the module allows you to change their positions in the Procmail configuration with the up and down arrows next to each on the main page. Variable assignments, conditional blocks, and include files can also be moved in the same way.

By following the instructions above, you could easily create an action that delivers all email to the Qmail-style `Maildir` directory in user's home directories. Even though this mail format is preferable due to its superior reliability compared to the traditional files in `/var/spool/mail`, it is not much use unless mail clients or the POP3 server on your system know how to read it. The POP3 server that comes with most operating systems expects to find email under `/var/spool/mail` and so will have to be replaced or reconfigured to support any new mail format or location. Other mail clients that read user mail files directly (such as Pine, Elm, and Usermin) can be configured to use whatever new location you choose.

45.5 Creating and Editing Variable Assignments

Procmail actions can make use of shell-style variables in their conditions and delivery destinations. For example, you could create an action that delivers to the file */mail/\$LOGNAME*, in which *\$LOGNAME* is the username of the user to whom email is being delivered. Several variables (like *.LOGNAME* and *DEFAULT*) are set automatically by Procmail, while others can be set in the configuration file for later use. You can even override the automatic variables to change the behavior of the program, such as the default delivery destination or shell to use for executing commands.

To create a new variable assignment, follow these steps:

1. On the module's main page, click on the **Add a new variable setting** link below the list of existing actions. The variable creation form will be displayed.
2. In the **Variable name** field, enter the name of the variable to set, such as *DEFAULT*. All automatic variables have uppercase names, and those that you create yourself should as well. No spaces or nonalphanumeric characters are allowed.
3. In the **Value** field, enter the value to assign to this variable, such as *Maildir/*. The value can include references to other variables.
4. Hit the **Create** button to add the variable to the list on the main page.
5. Use the up arrow next to the new variable in the list to move it to the correct location, which will typically be at the top of the file. Variable assignments only effect actions and assignments after them, so one added at the bottom may not have any effect.

As with actions, a variable can be edited or deleted by clicking on its name in the list. Variables can also be moved about with the up and down arrows next to them. Because they only effect actions and other assignments below them in the file, you will certainly want to move any new variable up to near the top of the list. One added and left at the bottom will not have any effect (except on the default delivery destination).

Procmail defines and allows you to change several special variables. The names and meanings of the most interesting ones are listed in Table 45.1.

Table 45.1 Variables Set by Procmail for Use in Actions

Variable name	Purpose
DEFAULT	The destination to which email that does not match any action will be delivered. Changing this variable will modify the default destination, which is the best way to have email delivered to a mail directory or file in users' home directories. To specify a directory put a / at the end of the path, such as <i>Maildir/</i> . As with destinations in an action, a nonabsolute path is assumed to be relative to the recipient's home directory.
HOME	The home directory of the user to whom mail is being delivered.
SHELL	The UNIX shell of the recipient. Sometimes it is necessary to set this for commands to be properly executed.
LOGNAME	The username of the UNIX user to whom mail is being sent.

Table 45.1 Variables Set by Procmail for Use in Actions (Continued)

Variable name	Purpose
ORGMAIL	The original default delivery destination, such as <code>/var/spool/mail/\$LOGNAME</code> . Changing this variable, however, does not effect Procmail's behavior.
SENDMAIL	The command that Procmail will run to forward email to another address, if specified by an action. You may want to change this if Sendmail is not installed on your system, as it defaults to <code>/usr/sbin/sendmail</code> .
SENDMAILFLAGS	The command-line parameters passed to the <code>\$SENDMAIL</code> command.
HOST	Your system's hostname, as reported by the <code>hostname</code> command.

45.6 Conditional Blocks and Include Files

A conditional block is a group of actions and variable assignments in the Procmail configuration file that is only processed if some conditions match. They can be used to create quite complex sets of actions, almost like a programming language. The Procmail Mail Filter module allows you to create and edit conditional blocks, but displays their contents as just configuration file text rather than parsing the actions that they contain. This means that you have to be familiar with the Procmail file format to use them.

To create an conditional block, follow these steps:

1. Click on **Add a new conditional block** below the list of actions on the module's main page.
2. In the **Procmail code to execute** text box, enter the configuration file lines for the actions or variable assignments to be processed if the conditions match. As soon as any action in the block matches, processing of the entire configuration file will stop. If none match, however, processing will continue as usual with the next action after the block.
See the `procmailrc` manual page in the System Documentation module for details of the format.
3. Fill in the **Action conditions** section just as you would for a normal action, as explain in Section 45.4 “Creating and Editing Actions”.
4. Hit the **Create** button to create the new block.

As with actions, you can edit or delete a conditional block by clicking on it in the list on the module's main page. The entire block can also be moved around with the up and down arrows.

An include file is a special directive that tells Procmail to read and process a separate configuration file in the same format as `/etc/procmailrc`. Some spam-filtering programs are actually just Procmail files that can be included in your primary configuration. To create an include directive, follow these steps:

1. Click on the **Add a new include file** link on the module's main page.
2. In the **Included file** field on the form that appears, enter the full path to the other configuration file. You can also enter a relative path, in which case Procmail will search for that file in the home directory of the UNIX user to whom that mail is being delivered. When

handling an include, Procmail will stop processing altogether as soon as it finds a matching action in the file. If none are found, it will continue processing the actions that come after the include in the primary configuration file.

3. Hit the **Create** button to finish the process.

Normally, includes are listed on the module's main page just like actions and variable assignments, and can be edited, deleted, or moved about. If the **Show contents of include files?** setting is enabled on the **Module Config** page, however, the module will display the actual actions inside the include file for you to edit or delete. They can even be moved up and down, although only within the file. Enabling this option is not a good idea if you have a massive include file (such as one for spam filtering), as it will make the module's main page unusably large. If the include file path contains a reference to a variable, its contents will never be shown by the module as Webmin has no reliable way to work out the value of Procmail variables in advance.

45.7 Filtering Spam with SpamAssassin

SpamAssassin is a powerful program for detecting unwanted spam messages based on their headers and content. It uses a complex set of built-in rules to determine if an email is spam or not, and can also consult other databases of known spam message texts and mail servers used for sending spam. The `spamassassin` program itself, however, does not perform any real filtering. Instead, it just takes email as input, adds special headers indicating if the message is spam or not, and then writes it out again. This makes it ideal for use in a Procmail action.

Assuming that you have SpamAssassin installed on your system, you can set it up to perform filtering for all users by following these steps:

1. Create a new action that feeds mail to the program `/usr/bin/spamassassin` (or wherever it is located on your system). Make sure the **Wait for action program to finish, and check result** and **Action program is a filter** boxes are checked. No conditions should be entered, unless you want to turn off spam checking for certain messages.
2. Add a second action with the single condition **Matches regular expression** `^X-Spam-Status: Yes`. This special header is set by SpamAssassin on messages that exceed its spam threshold. The delivery mode can be to append to the file `/dev/null` to throw away all spam, or to something like `$HOME/spam` to place it in a different mail file for users to skim through and delete.

Because SpamAssassin occasionally falsely identifies email as spam when it is not, just throwing away messages by sending them to `/dev/null` is a bad idea. It is far better to deliver messages to a separate file or directory that users can read if they wish, just in case.

By default, email identified as spam has its headers and body modified by SpamAssassin to deactivate any attachments and include a report about why it was categorized. This can be changed by editing the global configuration file `/etc/mail/spamassassin/local.cf`—the exact format of which is not covered in this chapter.

45.8 Configuring the Procmail Mail Filter Module

This module has a few options that control its user interface and the path to the Procmail configuration file. As usual, they can be edited by clicking on the **Module Config** link in the top left corner of the main page. Table 45.2 lists the editable settings.

Table 45.2 Module Configuration Options

Show contents of include files? (If no variables are used in filenames)	As Section 45.6 “Conditional Blocks and Include Files” explains, this field determines whether the actions of included files are displayed for editing on the main page. It is set to No by default because most include files are extremely large and complex.
Path to system procmailrc file	This field must contain the full path to the global Procmail configuration file, which is usually <code>/etc/procmailrc</code> .

45.9 Summary

This chapter has explained how Procmail can be used to filter all email received by the mail server on your system by performing various actions based on its content. After reading it, you should know how to create filter actions, and understand how they interact in the Procmail configuration. You should also know how to set the SpamAssassin package to block or redirect email that it classifies as spam.

Creating SSL Tunnels

In this chapter the STunnel program and the Webmin module for setting it up are documented.

46.1 Introduction to SSL and STunnel

SSL is a protocol for encrypting data in a TCP connection as it travels over the network. It was originally developed to protect the traffic between web browsers and servers, but can be used to encrypt any kind of data stream that would normally be sent via the TCP protocol.

The SSL protocol allows clients and servers to authenticate themselves to each other, so that a client can be sure it is really connecting to the host it thinks it is. This is done using certificates that are issued by a certificate authority recognized by the client (so that they can be verified) and associated with a particular hostname. Without certificates, an attacker could redirect an SSL connection to his own server and capture sensitive information from a client that thinks it is talking to the real server.

Any data that travels across the Internet unencrypted can be captured and read by an attacker with access to one of the networks through which it passes. Even data traveling between a client and server system on a LAN can be easily listened in on. When you connect to a telnet, FTP, or POP3 server, your password is sent over the network and can be captured by an attacker.

SSL can be used to protect data in these kinds of situations, but only if both the client and server support it. Most web browsers and mail clients can make SSL-encrypted HTTP, POP3, and IMAP connections, but not all web and POP3 servers can accept them. POP3 in particular is hard to protect, because the standard server that comes with most UNIX systems does not support SSL at all. Fortunately, though, there is a solution—STunnel.

STunnel is a simple program that converts an unencrypted connection into an SSL-encrypted one. It is typically set up to be run from a super server like `inetd` or `xinetd`, and then run some other program like the POP3 server that does not support SSL. This design allows it to protect any server that is normally run from `inetd`, such as telnet, NNTP, and IMAP servers.

Not all servers can be usefully protected with encryption, however, because no client exists to use them in SSL mode. For example, I have never heard of a telnet or FTP client that can use SSL because the common SSH package already allows encrypted remote logins and file transfers.

46.2 The SSL Tunnels Module

This Webmin module makes it easy to set up super server services that run STunnel to start a specific server program. Even though this can be done manually using the Internet Services module (covered in Chapter 15), this one is specifically designed for setting up and configuring STunnel. It automatically detects whether you have `inetd` and/or `xinetd` installed, reads their configurations to check for existing SSL tunnels, and adds to them when you create a new tunnel. If both are installed, new SSL tunnels are added to the `xinetd` configuration, as it is the superior of the two.

The module can be found in Webmin under the Networking category on the main menu. When you click on its icon, a page like the one shown in Figure 46.1 will be displayed, listing all existing tunnels. At the bottom of the page is a button labeled **Apply Changes**, which when clicked, restarts `inetd` or `xinetd`, thus making the current configuration active.

If the program cannot be found on your server, an error message like **The STunnel command /usr/bin/stunnel was not found on your system** will be displayed instead. This can indicate that it is not installed or that the module is looking in the wrong directory for the `stunnel` command. In the latter case, you can adjust the module's configuration, as explained in Section 46.4 “Configuring the SSL Tunnels Module”.

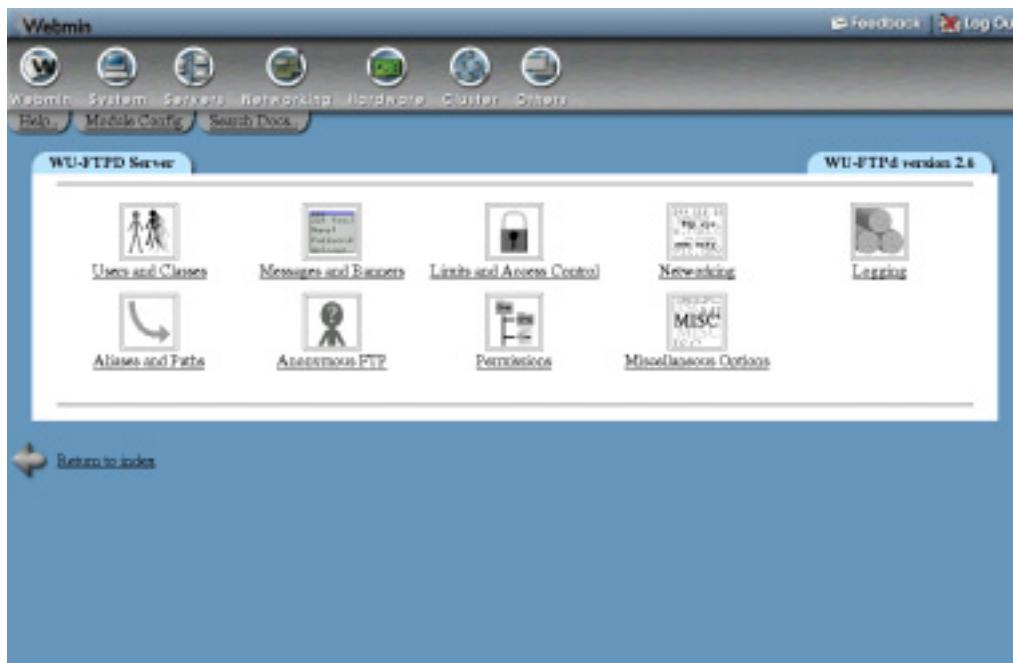


Figure 46.1 The SSL Tunnels module.

If the program really isn't installed, check your operating system CD or website to see if a package for STunnel exists. If so, you can install it using the Software Packages module that is covered in Chapter 12. Otherwise, you will need to download the source code from www.stunnel.org, compile it, and install it.

46.3 Creating and Editing SSL Tunnels

If you want to protect a specific service with SSL encryption, you will need to create a new SSL tunnel. Two different types of tunnels can be created—one that runs a server process like `inetd` does, or one that connects to another host and port in non-SSL mode. The latter is simpler if you already have the server running in non-encrypted mode, but will be slightly slower due to the need to make an extra network connection.

Before you can create a tunnel, you must decide on a port number for it to use. For some protocols, there is a standard port number. For example, 995 is often used for encrypted POP3 and 993 is used for encrypted IMAP. Of course, the port number you choose must not be in use by any other `inetd` service or server on your system.

The following steps will instruct you in how to create a tunnel:

1. On the module's main page, click on the **Add a new SSL tunnel** link above or below the table of existing tunnels. The creation form shown in Figure 46.2 will be displayed in your browser.
2. In the **Service name** field, enter a unique name for this tunnel's `inetd` service, such as *ssl-pop3*.

The screenshot shows the Webmin interface for managing users and classes. The top navigation bar includes links for Feedback, Log Out, and various system modules. The main window title is 'Users and Classes'. Below the title, there is a table titled 'User classes and user options' with columns for 'Class name', 'User type', and 'Matching address'. The table contains three rows: 'root', 'all', and an empty row. Underneath the table, there are several input fields for Unix users and groups, some of which are disabled. At the bottom of the form, there is a 'Save' button.

User classes	Class name	User type	Matching address
	root	<input type="checkbox"/> Unix <input checked="" type="checkbox"/> Anonymous <input checked="" type="checkbox"/> Guest	192.9.1.81
	all	<input checked="" type="checkbox"/> Unix <input type="checkbox"/> Anonymous <input type="checkbox"/> Guest	
		<input type="checkbox"/> Unix <input type="checkbox"/> Anonymous <input type="checkbox"/> Guest	

Unix users and UIDs to treat as guests:

Unix groups and GIDs to treat as guests:

Unix users and UIDs not to treat as guests:

Unix groups and GIDs not to treat as guests:

Unix users to deny (from /etc/hosts.deny): (Value: adm alias bin daemon listen lp nobody)

Unix users and UIDs to deny:

Unix groups and GIDs to deny:

Unix users and UIDs not to deny:

Unix groups and GIDs not to deny:

Figure 46.2 The SSL tunnel creation form.

3. In the **TCP port** field, enter the port number that the tunnel should accept connections on, such as 993.
4. Unless you want the tunnel to be temporarily disabled, set the **Active?** field to **Yes**.
5. If this tunnel should run a program like a POP3 server, select the **Run inetd style program** option. In the **Path to program** field, enter the full path to the server, such as `/usr/sbin/ipop3d`. In the **with arguments** field, enter the program name followed by any command line arguments, such as `ipop3d -a`. As with services created in the Internet Services and Protocols module, you must include the program name as the first argument.
6. If this tunnel should connect to some existing server, you can also select the **Connect to remote host** option. Enter the host to which you wish to connect (such as `localhost`) and the port number to use (such as 110) in the **Remote hostname** and **Remote port** fields, respectively.
7. The **SSL certificate and key file** field determines which SSL certificate will be presented to clients for this connection. If you have generated your own self-signed or real certificate with the `openssl` command, select the **Use cert in file** option and enter the full path to the file in the adjacent text box. Otherwise, you can choose **Use Webmin's cert** to use the same certificate that Webmin uses in SSL mode or **Compiled-in default** to use the certificate that comes with the STunnel software. If you do generate your own certificate, make sure that the file contains both the private key and cert in PEM format.
8. When connecting to a remote host, STunnel can be configured to behave in a way opposite from normal. Instead of accepting an SSL connection and decrypting it, you can instead choose to have it accept a normal connection and encrypt it for connecting to a different SSL-capable server. This mode can be enabled by selecting **Accept normal and connect with SSL** in the **Tunnel mode** field. It can be useful if neither your client or server programs support SSL, but you still want data between them to be encrypted. STunnel could be set up on the client system in this mode, configured to connect to another STunnel service on the server system that uses the **Accept SSL and connect normally** mode.
9. Hit the **Create** button at the bottom of the page to add the new service.
10. After you return to the module's main page, click on **Apply Changes** to make the new tunnel active.

All the details of an existing tunnel can be edited by clicking on its name in the list on the module's main page. This will bring up an editing form similar to the one in Figure 46.2, but with all the fields already filled in. You can either make changes and hit the **Save** button to record them or click **Delete** to completely remove the tunnel. Either way, the **Apply Changes** button on the main page must be clicked to make the changes active.

46.4 Configuring the SSL Tunnels Module

Like other modules, this one has a couple of options that control where it looks for the STunnel program and certificate file. You can edit them by clicking on the **Module Config** link in the top-left corner of the main page, which will take you to a form with the fields shown in Table 46.1.

Table 46.1 Module Configuration Options

Path to stunnel executable	This field must contain the full path to the <code>stunnel</code> program. If it is incorrect, the error mentioned in Section 46.2 “The SSL Tunnels Module” will appear on the module’s main page.
Path to default stunnel PEM file	If a path is entered for this field, it will be used as the default certificate file for the SSL certificate and key file field when creating new tunnels. If None is selected, the Use Webmin’s cert option will be selected by default.

46.5 Summary

This chapter has introduced the SSL protocol and explained what it is useful for. It has also shown how the STunnel program can be configured by Webmin to add SSL encryption to existing protocols, such as POP3 or IMAP.

Usermin Configuration

This chapter explains what Usermin is, why you might want to use it, and how it can be configured from within Webmin. It also provides a brief explanation of the available Usermin modules.

47.1 Introduction to Usermin

Usermin is a web interface similar to Webmin, but designed for normal UNIX users to carry out tasks that they should normally do at the shell prompt. It was written by the same author as Webmin, shares a lot of the same code, and has a similar underlying design and user interface. Whereas Webmin allows you do to things that would normally be done by logging in as `root`, Usermin lets you do things that can be done by logging in as a normal user.

Usermin is a very useful program to install if you want to give users on your system the ability to read and send email, change passwords, or edit files through an easy-to-use web interface. It groups all of these functions together and allows the administrator to choose which users get access to specific features.

Usermin can be downloaded from www.usermin.com in both RPM format for most Linux distributions or `tar.gz` format for other operating systems. It supports all of the same operating systems as Webmin and is installed in exactly the same way. If your version of Linux uses the RPM package format, the Software Packages module (covered in Chapter 12) can be used to install Usermin. Otherwise, you will need to extract the Usermin `tar.gz` file and run the `setup.sh` script, just like you would for Webmin.

Because Usermin uses port 20000 instead of 10000, you will need to go to the URL `http://yourservername:20000/` in your browser to access it after installation. Enter the username and password of any UNIX user on your system on the login page that appears and hit the **Login** button. A main menu very similar to the one in Webmin will be displayed, but with different cat-

egories and modules. In its default configuration, Usermin should be quite usable for tasks such as reading email, changing your password, or logging in via SSH.

Usermin can be navigated in the same way that Webmin can and its modules have very similar designs. In fact, some of the modules are exactly the same as those in Webmin, such as Running Processes and SSH/Telnet Login. The only difference is that they run with the privileges of the logged-in UNIX user rather than `root`.

This chapter focuses primarily on how to configure the program from within Webmin, using the Usermin Configuration module. Usermin does not have any facility to configure itself. You must either use this module, or edit the configuration files in `/etc/usermin` directly. Needless to say, the former option is much easier. The instructions in this chapter were written for Usermin 1.030. Some older versions lack certain features, such as the ability to restrict access to modules for specific users and groups.

Also in Section 47.17 “About the Usermin Modules” is a complete list of the standard modules and a short description of the capabilities of each one. It is not a user’s guide for Usermin however. The program is fairly easy to use and does not really need an instruction manual.

47.2 The Usermin Configuration Module

This module should be used if you want to reconfigure Usermin in any way, such as changing the default theme, the port on which it listens or the client addresses that are allowed to connect. It can be found under the Webmin category on the main menu and the main page that will appear when you click on the icon is shown in Figure 47.1. As you can see, the page is actually a table of icons, each of which can be clicked on to display a form for editing a class of options. At the bottom are buttons for starting or stopping the Usermin server process and possibly for setting it to start at boot time.

As is usual with Webmin modules, if Usermin is not installed, an error message like **The directory `/etc/usermin` either does not exist on your system** will appear on the main page instead. Even though Usermin is similar to Webmin, it must be installed separately by following the instructions in the introduction to this chapter. This error can also occur in the unlikely event that you have chosen a different configuration directory from the default of `/etc/usermin`. If so, read Section 47.18 “Configuring the Usermin Configuration Module” to find out how to change the module to look in the right location.

47.3 Starting and Stopping Usermin

Usermin has its own permanently running web server process, which can be started or stopped using this module. At the bottom of the main page is either a button labeled **Stop Usermin** or **Start Usermin**, depending on whether it is currently running or not. The server can also be stopped and started at the command line by running `/etc/usermin/stop` or `/etc/usermin/start` as `root`.

On operating systems like Linux, Solaris, and HP/UX, that use standard SYSV-style bootup action scripts, the main page also has a button labeled **Start at boot time**. If you select the **Yes** radio button next to it and hit the button, a bootup script will be created or enabled to start the Usermin server when your system boots. Selecting **No** will cause any existing script to be disabled so that it does not start. The action will be visible in the Bootup and Shutdown module (covered in Chapter 9), and you can enable or disable it there as well.

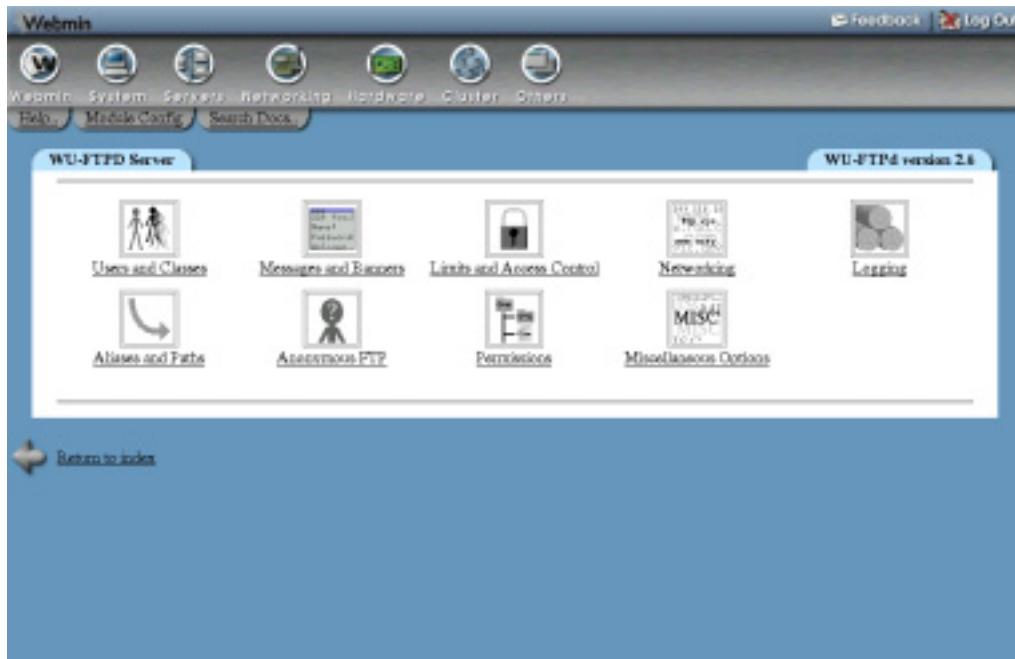


Figure 47.1 The Usermin Configuration module.

On operating systems like FreeBSD and MacOS X that use a different method of running commands at boot time, this button will not appear at all. You can still use the Bootup and Shutdown module, however, to have the command `/etc/usermin/start` run at boot time to achieve the same result.

47.4 Restricting Access to Usermin

By default, Usermin will accept connections from any IP address. Even though it is password-protected, you should limit access to only legitimate client systems, if possible, so that an attacker from outside your network cannot even attempt to log in. To do this, follow these steps:

1. Click on **IP Access Control** on the module's main page to bring up the **Access Control** form.
2. Select **Only allow from listed addresses** and enter a list of hostnames, IP addresses, and networks into the adjacent text box. Networks should be entered with a netmask like `192.168.1.0/255.255.255.0`. You can allow access from an entire DNS domain by entering something like `*.example.com`, but be aware that that is not totally secure as an attacker can fake reverse DNS results.
3. Usermin will normally resolve any hostnames that you enter only once when it first starts up. To change this, check the **Resolve hostnames on every request** box and it will convert hostnames to IP addresses for comparison on every request. This can be useful if the system on which you are running a browser is frequently changing IP addresses but is

able to update a DNS record to match. This can happen on a network using DHCP or if you are connected to an ISP that dynamically assigns addresses.

4. To have Usermin check the TCP-wrappers configuration files `/etc/hosts.allow` and `/etc/hosts.deny` when deciding whether to allow a client, turn on the **Also check TCP-wrappers hosts.allow and hosts.deny files** option. The service name to use when editing those files is `usermin`.
5. Hit the **Save** button to activate the new client address restrictions.

47.5 Changing the Port and Address

Usermin usually listens for connections on port 20000 on all of your system's IP addresses. You may need to change the port, perhaps because a firewall on your network only allows connections to web servers on the standard ports of 80 and 443. Changing the listening IP address can also be useful if your system has multiple network interfaces and you only want to allow connections on the interface connected to the internal LAN.

To change the port or address, do the following:

1. Click on the **Port and Address** icon on the module's main page.
2. To listen on only a specific interface address, select the second option in the **Listen on IP address** field and enter an IP address into the text box next to it. This must be the address of one of your host's real or virtual interfaces.
3. To change the port, enter a number into the **Listen on port** field.
4. Hit the **Save** button to use the new settings. Anyone currently using Usermin will need to log in again at the new URL, as the old one will stop working.

47.6 Configuring the Usermin User Interface

Usermin has several settings that control what appears in its user interface, what module users are directed to when they log in, and if the sending of feedback is allowed. To edit them, follow these steps:

1. On the module's main page, click on the **User Interface** icon to bring up the interface options form.
2. In some themes (covered in Section 47.12 “Changing and Installing Themes”), the title at the top of every page is rendered as an image. Because this can make the page slow to download, you can force the use of plain HTML text titles instead by changing the **Display titles as text?** field to **Yes**.
3. By default, every page in Usermin shows your system's hostname and operating system, which you might regard as a security risk. To turn this off, select **Nowhere** from the **Display login and hostname** menu, and change the **Show version, hostname and OS on main menu?** field to **No**. The first menu can also be used to change the location of system information. It appears by default in the browser status line.
4. After a user logs into Usermin, they will normally see the main menu listing the various modules and categories. To have users redirected to a specific module instead, select it from the **After login, always go to module** menu. This can be handy if most of your users use a particular feature, such as the Read Mail module.

5. Like Webmin, Usermin has a button in the top-right corner of every page for sending feedback. It is disabled by default, but you can turn it on by changing the **Allow sending of feedback** field to **Yes, to address** and entering an appropriate address in the adjacent text box. This gives your users an easy way to send you questions or problem reports.
6. By default, feedback is sent by running `/usr/lib/sendmail` and passing email to it for delivery. If Sendmail is not installed on your system or you want feedback to be sent via another mail server, select **SMTP server** in the **Send feedback via** field and enter a hostname into the text box. This tells Usermin to make an SMTP connection to that host for sending email instead.
7. Click the **Save** button at the bottom of the page to activate the new settings.

47.7 Installing Usermin Modules

Like Webmin, Usermin has a modular design. This means that each module (such as Read Mail or File Manager) is a separate piece of code and can usually be installed or removed without affecting the rest of Usermin. Being able to install new modules is the most useful feature, as several have been developed by people other than the Usermin developer. The best place to find extra modules is the website webmin.thirdpartymodules.com, which is a searchable database of almost all Webmin and Usermin modules. You can also write your own modules, as Chapter 55 “Webmin Module Development” explains. Do not bother trying to install modules for Webmin, as they will be rejected and will not work anyway.

A new module can be installed by following these steps:

1. On the Usermin Configuration module’s main page, click on the **Usermin Modules** icon. This will bring to you to a page with forms for installing, cloning, and deleting modules.
2. If you have already downloaded the module’s `.wbm` file to the system on which Usermin is running, select **From local file** and enter the full path to the file into the text field next to it.
If the module file is on the PC on which your web browser is running, select **From upload file** and use the **Browse** button to find the file on your computer.
If the module is on a website somewhere, select **From ftp or http URL** and enter the full URL into the text box next to this option.
3. Hit the **Install module from file** button to download (if necessary) and install the new module. If everything goes as planned, a page listing the installed modules and the sizes of their directories will be displayed.

Unless you have hidden modules from certain users (as explained in Section 47.15 “Restricting Access to Modules”), this new one will be immediately visible to and usable by all Usermin users.

Any of the modules currently installed—including those that come with Usermin by default—can be deleted on the same page as well. Deleting the default modules is not a good idea, however, as they will be automatically reinstalled the next time you upgrade. It is better to hide the ones that you don’t want people to use, as explained in Section 47.15 “Restricting Access to Modules”. Not all modules can be deleted either, as some are depended upon by other modules.

The Running Processes module in particular has many dependants, so removing it will cause modules like Change Password, Custom Commands, and GnuPG Encryption to stop working.

To remove one or more modules, follow these steps:

1. Click on the **Usermin Modules** icon on the main page.
2. Scroll down to the last form on the page and select all the modules that you want to remove from the **Delete Modules** list.
3. When you hit the **Delete selected modules** button, a confirmation page will be displayed showing exactly what will be removed. If there are some dependency problems that prevent one or more from being deleted, an error message explaining the problem will be shown instead.
4. Click on **Delete** to go ahead with the module's removal.

47.8 Changing the Default Language

Like Webmin, parts of Usermin have been translated into different languages. You can change the default language for users by following the next set of steps, or they can individually specify their own preferred languages using the Change Language module. None of the translations are complete, however, so many messages and labels will still appear in English.

1. Click on the **Language** icon on the module's main page.
2. In the form that appears, select your users' preferred language from the **Display in language** menu.
3. Some browsers (such as Opera) can request that the server display pages in a language chosen by the user. To have Usermin honor such requests, if possible, change the **Use language specified by browser?** field to **Yes**. If a language is sent, it will override both the global and individual users' settings.
4. Hit the **Save** button to have Usermin switch to the new language.

Assuming they have access to the Change Languages module, users can override whatever global selection you make in this module. This can be handy if most but not all users speak English.

47.9 Upgrading Usermin

Even though Usermin can be upgraded by installing the latest RPM or `tar.gz` package from the command line, this module can do the job for you with even less effort. The program can be upgraded from a package that you have downloaded from a URL somewhere or directly from the www.usermin.com site. In all cases, the upgrade must be made using the same type of package from which Usermin is currently installed. This means that if you originally installed the RPM package, you must upgrade with an RPM as well.

The easiest way to upgrade is to have Webmin check for and download the latest version directly from the Usermin site. This ensures that the right kind of package will be used and that nothing will be done if you are already running the latest stable release.

To upgrade using any of the above methods, follow these steps:

1. Click on the **Upgrade Usermin** icon on the module's main page. This will take you to a page with forms for upgrading, installing updated modules, and setting up the automatic install of updates.

2. The **Upgrade Usermin** form is very similar to the form for installing modules, as explained in Section 47.7 “Installing Usermin Modules”. Select either **From local file** if the new package is already on your server system, **From uploaded file** if it is on the PC on which your web browser is running, or **From ftp or http URL** to have the package downloaded from some URL. The easiest option is to choose **Latest version from www.usermin.com** to have the appropriate package automatically downloaded.
3. If Usermin on your system was installed from the `.tar.gz` file, the **Delete old version's directory after upgrade?** box can be checked to have the old version removed after the new one is installed. Unless you want to be able to revert to the old release, this option should be enabled to save on disk space. It does not appear at all for RPM installs, as the RPM package always installs in the same directory.
4. Hit the **Upgrade Usermin** button to begin the upgrade. A page showing the download progress (if necessary) and output from the new version’s `setup.sh` script will be displayed.

The upgrade process will preserve all global and user settings, and should not even be noticeable by users currently accessing your Usermin server. If you originally installed the program from the `.tar.gz` package, the new version will be installed in the directory next to the old one. For example, if Usermin 1.020 was in `/usr/local/usermin-1.020` and you upgrade to version 1.030, it will be installed in `/usr/local/usermin-1.030`, and the old directory deleted if the **Delete old version's directory after upgrade?** option was enabled.

Also on the upgrade page are forms for installing updated modules for Usermin from www.usermin.com, and for having such updates installed automatically. Occasionally, a bug is found in the latest version of the program, and an updated module that fixes the problem is made available at www.usermin.com/uupdates.html for administrators to download and install. Instead of checking manually, you can use this Usermin module to find and install needed updates.

Because this feature is identical to one for installing updates to Webmin, it is not covered in this chapter. Instead, see Section 51.13 “Upgrading Webmin” for details on how to use it. The instructions in that chapter that apply to the Webmin Configuration module can also be used in this module as well.

47.10 Configuring Authentication

Usermin has several options that control the authentication method it uses for validating UNIX users, how multiple failed login attempts are handled, and how UNIX user passwords are checked. The default authentication method uses cookies, but if your users’ browsers cannot handle them, you may want to switch to basic HTTP authentication instead. The only problem with this method is that there is no way to properly log out, as there is no support for logging out in the HTTP protocol. However, it sometimes must be used. For example, browsers on MacOS X cannot load applets (such as the ones in the File Manager and SSH/Telnet Login Usermin modules) from web servers using cookie authentication.

To configure authentication for Usermin, follow these steps:

1. Click on the **Authentication** icon on the module’s main page to bring up the authentication form.
2. When **Enable password timeouts** is selected, Usermin will detect multiple failed login attempts from the same IP address and lock that host out for a configurable amount of

time. This feature should always be turned on, as it stops attackers from using millions of login attempts to guess passwords on your system. The **Block hosts with more than** field specifies the number of login attempts allowed from a single host before blocking is triggered, while the **failed logins for** field sets the number of seconds for which a host is blocked. The defaults are reasonable, but you can increase the timeout if you are feeling paranoid.

3. When **Log blocked hosts, logins and authentication failures to syslog** is selected, Usermin will send messages to the system logs (covered in Chapter 13) when a user logs in, logs out, or enters an incorrect password. All messages are sent with the authpriv facility. You should leave this option turned on so suspiciously large numbers of login failures can be detected.
4. When **Enable session authentication** is selected, Usermin will use its own login form to ask users for a username and password, and set a cookie after the login is complete to identify authenticated clients. To switch to normal HTTP authentication, select **Disable session authentication**.
5. When using session authentication, Usermin can be configured to automatically log users out if they have been inactive for more than a certain period of time. To enable this, check the **Auto-logout after** box and enter the number of minutes into the text field next to it. This feature and the next three are not available when using HTTP authentication.
6. When **Offer to remember login permanently?** is checked (as it is by default), the login form will include a check box for permanently remembering the login. When selected, the cookie sent to the user's browser will be marked to indicate that it should be saved even if the browser is shut down and rerun later. This is convenient because it means that the user will not have to log in to Usermin again, but you may consider it a security risk. If so, unchecking this box will remove the **remember** option from the login form.
7. By default, the login page includes the hostname from the URL in the message above the username and password fields. To hide it, deselect the **Show hostname on login screen?** box.
8. Some people like to have a welcome message shown on the login page the first time a user accesses it, perhaps giving information about the server or for what it is supposed to be used. To enable this on your system, first create an HTML page containing the message that you want to appear. Then, select **Show pre-login file** and enter the full path to the HTML file in the text field. After a user reads it, he must reload or revisit the page (perhaps by following a link in the page itself) to force the real login form to appear.
9. To have Usermin automatically authenticate connections from `localhost` by determining which UNIX user is making the connection, select **Allow login without password for matching users from localhost**. If you run a browser on the same system on which Usermin runs, this feature allows you to access the URL `http://localhost:20000/` and be logged in without needing to enter a username and password. It is convenient, but potentially insecure if an attacker can trick a program (such as Squid) into connecting to that URL. This would grant access to Usermin as the user as whom the program runs. For this reason, **Always require username and password** is selected by default.
10. Usermin can check users' passwords in three different ways: using PAM, by reading the password file directly, or by consulting some other program.

PAM is the best method and can be enabled by selecting **Use PAM for authentication** on this form. It will only work, however, if your operating supports PAM (only Linux and Solaris do), if the `Authen::PAM` Perl module is installed, and if the `/etc/pam.d/usermin` service file is set up correctly on Linux. This file is included in the RPM package of Usermin.

The most reliable method of authentication is directly reading the `/etc/passwd` or `/etc/shadow` file containing usernames and passwords. You can enable this by selecting **Authentication using password file**. The other fields next to it are set by default to match your operating system and do not generally need to be changed. The only problem with this authentication method is that even expired accounts will be able to log in, as Usermin does not check those password file fields.

The final authentication method uses an external program to validate passwords. This program must behave exactly like Squid's external authentication program, covered in Section 44.9 “Setting Up Proxy Authentication”. To enable it, select **Use external squid-style authentication program** and enter the full path to the program, followed by any parameters, into the adjacent text field. This option can be useful for looking up passwords in an LDAP or MySQL database. It cannot, however, be used to create fake users who do not have real UNIX accounts.

11. Finally, hit **Save** at the bottom of the form to activate the new authentication settings. They will not, however, apply to already logged in users.

47.11 Editing Categories and Moving Modules

Every Usermin module has a category that controls where it appears on the module's main menu. You can create your own categories and move modules from their default locations into your own or existing categories, which can be useful if you don't like the default arrangement or want to put everything into one huge category.

To create new categories, or rename existing ones, follow these steps:

1. Click on **Edit Categories** on the module's main page to display the category editing page.
2. To add a category, scroll down to the bottom of the form. In the first empty field under **ID**, enter a unique internal name for your new category, such as *stuff*. Then, in the field next to it, under **Displayed description**, enter the name that will appear in Usermin, such as *My Stuff*.

Existing categories that you have added can be edited by changing the fields in this section as well. You should not change the entries in the **ID** column, however, as they are used internally to associate modules with categories. The ID is never visible to users anyway—only the displayed description is.

3. To change the name of one of the default categories displayed at the top of the form, select the second radio button next to it and entering a new description into the text box to the right. If **Default** is chosen, the standard name based on the user's language will be used.
4. Hit the **Save Categories** button at the bottom of the form to activate the new categories. You can now move modules into any categories that you have created.

To change the categories in which modules appear in, use the following process:

1. Click on the **Reassign Modules** icon on the main page.
2. The page that appears lists every installed Usermin module and the category in which it currently resides. For each module that you want to move, select a new category from the menu next to its name.
3. Click on the **Change Categories** button at the bottom of the page to move the modules.

47.12 Changing and Installing Themes

A theme is an extension of Usermin (much like a module) that controls how its interface appears to users. The currently active theme determines if and how the categories at the top of each page are displayed, what page background is used, what icons each module has, how titles appear, and how each page ends. By changing themes, you can significantly change the look of Usermin without affecting its functionality. Several themes are included by default and you can install more that have been written by other developers.

Like the language, you can set the theme for all users in this module and users can choose their own themes and override the default with the Change Theme module in Usermin. To change the theme for everyone, follow these steps:

1. Click on the **Usermin Themes** icon on the module's main page. This will take you to a page for changing themes, installing a new theme, and deleting existing ones.
2. Select the theme to use from the **Current theme** menu. Those included as standard with Usermin are:

Old Usermin Theme The very simple theme that the first versions of Webmin and Usermin used before theming was added. If you find the default too slow, this may be a better alternative as it uses fewer images.

MSC.Linux Theme The current default Usermin theme.

MSC.Linux Mini Theme A modified version of the default theme, designed for use on small-screen devices such as PDAs.

3. Hit the **Change** button to activate the chosen theme.

New themes developed by other people can also be added to Usermin, although none actually exist as I write this book. Themes for Webmin cannot be used, due to differences in the design of the two packages. You can write your own, however, as Chapter 58 “Creating Webmin Themes” explains.

To install a theme, follow these steps:

1. Click on the **Usermin Themes** icon on the module's main page.
2. Select the theme's file using the second form. Just as when installing a module, you can choose to install a theme from a file on the system running Webmin, the PC your browser is on, or an HTTP or FTP URL.
3. Hit the **Install Theme** button to download (if necessary) and install it.

The final thing that you can do on this page is delete one of the installed themes. The **Old Usermin Theme** cannot be deleted as it is built into the program, and the other two standard themes should not be deleted, as they will be added again if you upgrade to the next version.

To delete a theme that you have installed, follow these instructions:

1. Click on the **Usermin Themes** icon on the module's main page.
2. Select the one to remove from the **Theme to delete** menu at the bottom of the page. If that menu does not appear, it means that all installed themes are in use either by an individual user or are the default for all users.
3. Hit the **Delete** button to bring up a confirmation page asking if really want to go ahead.
4. Click on **Delete** to remove the theme.

47.13 Turning on SSL

Like Webmin, Usermin can operating in SSL mode if the OpenSSL library and Net::SSLeay Perl modules are installed. Chapter 3 “Securing Your Webmin Server” explains how to install them and why SSL should be used, so read it first before continuing with this section. Usermin will also automatically use SSL mode by default if it detects that the needed libraries are available at install time, and will generate its own unique SSL certificate and key for your system, if possible.

If you install the required libraries after Usermin, you can switch to SSL mode by following these steps:

1. Click on the **SSL Encryption** icon on the module's main page. If Net::SSLeay is missing, an error message will be displayed telling you that SSL mode cannot be used. Otherwise, a form for turning it on and off and for generating a new SSL key will appear.
2. Change the **Enable SSL if available?** field to **Yes**.
3. If you have your own SSL key for this host already, enter its full path into the **Private key file** field. If this file just contains the key and not the certificate, you will need to fill in the **Certificate file** field as well. To just use Usermin's own certificate, leave these fields unchanged.
4. Hit the **Save** button to switch to SSL mode. All users that try to connect to the old *http://* URL will from now on be told to use the new *https://* URL, instead.

This same page can also be used to generate a new SSL key for use by Usermin. You should definitely do this if OpenSSL was not installed when Usermin was, as it will fall back to using the key that comes with the program if a new one cannot be generated at install time. This is highly insecure, as the key is available to everyone and can be used to decrypt network traffic, thus totally negating the main benefit of SSL! You might also want to create a new key if the details of the default one (such as the company name and country) are not correct.

Follow these instructions to generate and starting using your own key and certificate:

1. Click on the **SSL Encryption** icon on the module's main page and scroll down to the bottom of the form.
2. If your system is always accessed using the same hostname in the URL, enter it into the **Server name in URL** field, such as *www.example.com*. This will cause the generated certificate to be associated only with that hostname. Otherwise, select **Any hostname** to allow the certificate to be used with any URL hostname. This is more convenient, but slightly less secure.

3. In the **Email address** field enter the address of the person responsible for this Usermin server, such as *joe@example.com*.
4. If appropriate, fill in the **Department** field with the name of the department or group within your organization that this server belongs to, such as *Network Engineering*.
5. In the **Organization** field, enter the name of the company or organization that owns this server, such as *Foo Corporation*.
6. In the **State** field, enter the name of the state in which your server resides, such as *California*.
7. In the **Country code** field, enter the two-letter code for the country in which the server resides, such as *US*.
8. Leave the **Write key to file** field unchanged, unless you want the key file to be written elsewhere.
9. To have Usermin configured to use the newly generated key, leave the **Use new key immediately** field set to **Yes**. If you select **No**, you will need to switch to this key later by following the instructions earlier in this section.
10. Hit the **Create Now** button to generate the key and certificate and store them in the specified file in PEM format.

All of the fields in this form are optional, with the exception of **Server name in URL**. If the key is just for use on your own home server, there is no need to enter a department or organization name. You must make sure, however, that any key you generate here has different details than the one created for Webmin itself. Browsers like Mozilla and Netscape currently have problems if they encounter two different keys with the same server name, department, organization, and so on.

47.14 Configuring Usermin Modules

Almost all Usermin modules have several configurable settings that effect their user interfaces and behavior. There are actually two types of settings—those that apply to all users and are set by the administrator in this module, and those that apply to only a specific user and can be set by users themselves from within Usermin. This latter set of options is called *preferences* and can be set by users by clicking on the **Preferences** link that appears in the top-left corner of the main page in some Usermin modules (the same place that the **Module Config** link appears in Webmin). You can also use the Usermin Configuration module to set the default preferences for users who have not yet set them.

To set the configuration or default preferences for a module, follow these steps:

1. Click on the **Usermin Module Configuration** icon on the module's main page to go to a list of all installed modules.
2. Click on the name of the module that you want to configure. This will bring up a page containing one or two forms—**Configurable options** is for editing the global settings for the module, while **Default user preferences** is for editing preferences. Figure 47.2 shows an example. Because some modules do not have preferences and some do not have configurable options, one or the other of the forms may not be displayed.

The actual fields in both forms depend on the module chosen. For example, the Read Mail module has settings that control where it looks for user email and in what format it expects the mail file or directory to be. The defaults in the **Configurable options** form



Figure 47.2 The Usermin module configuration page.

are set to match your operating system when Usermin is installed and do not usually need to be changed.

3. To edit the module's configuration, make whatever changes you want to the fields in the first form and hit the **Save** button below it to activate them.
4. To change the default preferences, change the fields in the second form and hit its **Save** button to activate them. They will only apply to users who have not set their own preferences for the module, however. This form always includes a **Users can edit preferences?** field that, if set to **No**, stops users from editing the module's preferences in future.

The configurable settings in most modules are fairly obvious and need no further explanation. The Read Mail module's form, however, has a large number of fields that control where it looks for email, how it sends mail, what **From:** address users are assigned, and where user folders are stored. Even though it usually defaults to looking for mailboxes in the `/var/spool/mail` directory, it can be configured to use the Qmail-style `Mailbox` file or `Maildir` directory in users' home directories, which is necessary if you are running a mail server other than Sendmail.

The configuration fields for the Read Mail module and their meanings are listed in Table 47.1.

47.15 Restricting Access to Modules

Usermin will usually allow all users who can log in to access all of the installed modules. However, this may not be appropriate for your system. You may want some users to just be able to read email and change their passwords, while giving others access to everything. Some of the

Table 47.1 Read Mail Module Configuration Options

Mail storage format for Inbox	This field tells the module what format user mail files are in on your system. The available options are: Sendmail style single file Each user has a single file containing all his email messages, located either in a directory like <code>/var/spool/mail</code> or in his home directory. Qmail style directory Each users' mail is stored in a directory in the format used by Qmail—usually called <code>Maildir</code> and located in the users' home directories. Each individual message is in a different file. MH style directory User email is also stored in a directory, but in the slightly different format used by the MH mail program. Remote POP3 server Mail is on another server that supports the POP3 protocol. When this mode is selected, users will be prompted to enter a username and password for the POP3 server the first time they log in to the module. Remote IMAP server Like the previous option, but uses the IMAP protocol instead. When a user first uses this module, they will be prompted to enter an IMAP mailbox name to manage as well.
Sendmail mail file location	This field and the next one are only used if Sendmail style single file is selected as the mail storage format. You can either select File under home directory to have the module read from a file (set in the next field) in the logged-in user's home directory or enter the full path to a directory containing a mail file of the same name as the user, such as <code>/var/spool/mail</code> .
Sendmail mail file in home directory	When the previous field is set to File under home directory , this one specifies the name of that file. Often set to <code>Mailbox</code> , as that is what Qmail uses by default.
Qmail or MH directory location	These two configuration fields are only used if the mail storage format is set to Qmail style directory or MH style directory . You can either select Subdirectory under home directory to have the module read from a directory (set in the next field) within the user's home directory, or enter the full path to a directory that contains a subdirectory with the same name as the user, such as <code>/maildirs</code> .
Qmail or MH directory in home directory	When the previous field is set to Subdirectory under home directory , this one specifies the name of that subdirectory. Set to <code>Maildir</code> by default because that is what Qmail often uses.
Mail subdirectory style	When user mail files or directories are located somewhere other than their home directories, this field must be set to match the layout used. On large systems with thousands of users, it often makes sense to group mail files into subdirectories named after the first letter or two of users' names. The options in this menu match various commonly used directory structures. For most systems, however, the default of mail/username is correct.

Table 47.1 Read Mail Module Configuration Options (Continued)

POP3 or IMAP server name	When the mail storage format is set as the Remote POP3 server or Remote IMAP server , this field specifies the hostname of the server to which to connect. You can select localhost to fetch mail with POP3 or IMAP from the same system, although it is more efficient to configure the module to read mail files directly. This may not be possible, however, if you are using the mail storage format—like <code>mbx</code> —that your IMAP and mail servers support but that Usermin does not.
Send mail via connection to	When Sendmail executable is selected, the module will use the <code>sendmail</code> program to send outgoing email. Even though most mail servers (like Qmail and Postfix) include a fake <code>sendmail</code> program that will work, you can select the second radio button and enter the hostname of another SMTP server to connect to instead. This option is also useful if your system is not running a mail server at all.
Sendmail command	When the previous field is set to Sendmail executable , this one must be filled in with the full path to the actual <code>sendmail</code> command it will use.
Default hostname for From: addresses	When users of this module send email, this field determines what host or domain name is used after the @ in the From: address. If From real hostname is selected, your system's name as reported by the <code>hostname</code> command will be used. If From URL is chosen, the hostname in the URL used to access Usermin (minus any <code>www</code> at the start) will be used. If you select the last option, the domain name entered in the adjacent text box will be used. The last option is generally the best, as it allows you to specify exactly what domain name will be used for outgoing email, such as <code>example.com</code> instead of your server's real hostname like <code>foo.example.com</code> .
Allow editing of From: address	When Yes is selected, users will be able to edit the From: address before sending email. Assuming that you have the module set up to automatically use the correct domain name in the address, No should be selected to prevent confusion and stop users from forging email. Of course, there are a hundred other ways that the From: address can be forged in email.
From: address mapping file	This field can be used to specify a file mapping Usermin login names to From: addresses. It can be very useful if your server hosts multiple email domains and you want different users to use different domain names in their email. If you do enter a file, it must contain one line per user in the format: <code>username address@domain</code> When sending mail, the module will look for the user's address in the first column and use the matching From: address in the second column.

Table 47.1 Read Mail Module Configuration Options (Continued)

Allow access to server-side files?	By default, the Read Mail module allows users to select a file on the server system to attach to outgoing email. If this presents a security risk on your system, change this field to Neither . Users will still be able to upload files from the PC on which they are running a browser, however.
Maximum total attachments size	This field can be used to prevent users from sending excessively large emails by entering a number of bytes that the total unencoded size of all attachments in a message may not exceed.
Minimum mail file size to index	When using the Sendmail style single file mail storage format, the module creates indexes of user mail files to speed up their display. Indexing can fail, however, if a file is being frequently changed by another program such as a POP3 server. This field can be used to turn off indexing for mail files smaller than a specified number of bytes. This will have no effect on the module's functionality—it will just make it a little slower.
Use DBM files for indexes?	When No is selected for this field (as it is by default), mail indexes generated by the module will be in a simple text format. This works fine, but is slow for very large mail files. Selecting Always will force the generation of binary DBM format indexes, which are much faster and include the Subject: and From: lines of messages, making searching faster, as well. If Only for new indexes is selected, the DBM format will only be used if a text index does not yet exist.
Global address book file	The Read Mail module lets users create their own personal address book for use when sending email. This field can be used to specify a file of addresses that will be available to all users as well, but not editable by them. The file must contain one email address and real name per line, separated by a single tab. If the filename contains the special code <code>\$group</code> , it will be replaced with the name of the user's primary or secondary group (depending on which resulting file actually exists). This allows you to create different global address books for different users on your system.
Allowed folder types	By default, users can add external files or remote POP3 and IMAP servers as folders. If this bothers you for security reasons, this field can be used to deny access to certain folder types. Any existing user folders of the denied types will be no longer accessible to their owners.

modules are quite powerful, such as the File Manager and Command Shell, and so should be restricted to people who have FTP or SSH access to your system.

Naturally it is possible to set this up in Usermin, or this section would not have been written. This Webmin module lets you select the Usermin modules that are available to a specific user or

members of a group. This is done by creating a list of rules, each of which applies to some user or group or to all users and that either adds or subtracts a list of modules from that user. This allows for quite complex module restriction configurations to be created. For example, you could give the group *users* access to three modules, and then the user *fred* (who is a member of *users*) access to one more without having to list the other three for him explicitly.

To create a module restriction rule, follow these steps:

1. Click on the **Module Restrictions** icon on the main page to bring up a list of existing restrictions, an example of which is shown in Figure 47.3.
2. Click on **Add a new user or group restriction** above or below the list to go to the restriction creation form.
3. The **Applies to** field determines which users this restriction affects. You can select **Unix user** and enter a single username in the field next to it, **Members of group** and enter a group name, or **All users**. The latter option is useful for defining the modules that everyone can use, except for users to whom you grant access to more modules in later restriction rules.
4. In the **Modules** section is a list of all Usermin modules installed on your system. If **Only selected** is chosen, then only those modules that you check will be granted access to the user or group. If **Add selected** is chosen, then the checked modules will be given to the users in addition to any that they have been granted by previous rules. If **Remove selected** is chosen, the modules that you check will be taken away from the user or group if they have been granted by a previous rule. In most cases, the default of **Only selected** is all you will need for creating restriction rules.
5. Hit the **Create** button to add and activate the restriction. It will immediately apply to all matching users—even those currently logged in.

Once a restriction has been created, you can edit it by clicking on the user or group name in the list on the Module Restrictions page. This will take you to an editing form similar to the one in Figure 47.4. Change the user or group, or list of modules, and hit **Save** to activate the new selections. Or, use the **Delete** button to remove the restriction from the list altogether. Because the ordering of restrictions matters, you can move them around in the list with the up and down arrows that appear in the right-most column on the restrictions page. Again, any changes to the list will take effect immediately.

Normally, if no restrictions exist, all users will have access to all modules. This can be changed by clicking on the **Available Modules** icon on the main page and deselecting those to which nobody should have access. Modules taken away in this way cannot be granted back to specific users on the module restrictions page. Because module restrictions are far more flexible than using the Available Modules page to control which ones are visible, there is no real need to use it.

47.16 Limiting Who Can Log In

By default, Usermin lets any UNIX user on your system log in—even `root`. If this is not what you want, it can be configured to allow or deny access by only certain users or the members of certain groups. This can be useful if many users on your system exist only to receive and download email or upload files with FTP, or if you want to deny `root` access. It is also possible to prevent users from logging in if they do not have a shell in a certain file, just as most FTP servers do.

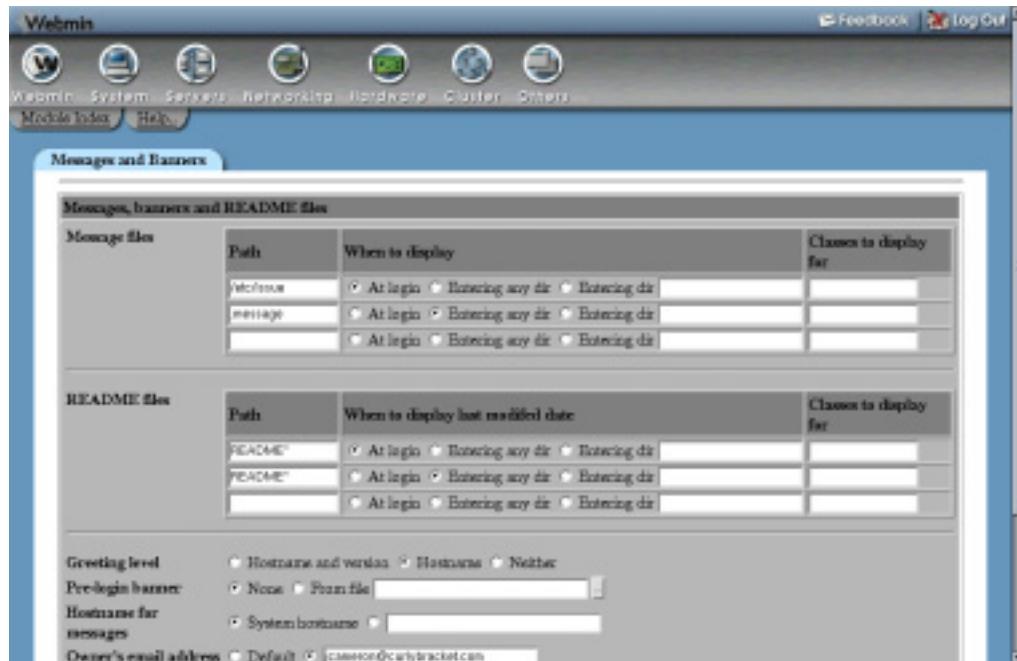


Figure 47.3 The module restrictions page.



Figure 47.4 Editing a user restriction.

To control who can log in to Usermin, follow these steps:

1. Click on the **Allowed Users and Groups** icon on the module's main page.
2. To give only certain users access, select **Only allow listed users** and fill in the text box with a list of user and group names. Groups must be prefixed with an @ (such as @users), and match whether the user attempting to log in is a primary or secondary member.
Alternately, you can allow everyone except certain users by selecting **Deny listed users** and entering the user and group names you want to block.
3. The WU-FTPD and ProFTPD servers deny access to any user whose shell is not listed in the /etc/shells file so you can create users who cannot make FTP logins. Usermin can be configured to do the same thing by checking the **Deny users whose shells are not in file** and entering /etc/shells into the adjacent text field.
4. Hit the **Save** button to activate the new restrictions. They will not, however, affect users who are already logged in.

47.17 About the Usermin Modules

Table 47.2 lists the modules that are included with Usermin 0.990 and provides a brief explanation of what each one does and how safe each is for untrusted users to have access. Not all modules are available on all operating systems due to the differences between the various varieties of UNIX.

Table 47.2 Standard Usermin Modules

Module name	Purpose and risks
Apache Options Files	Allows users to edit Apache .htaccess files in any directory, assuming that UNIX permissions permit it. Its interface is very similar to the feature in Webmin's Apache Configuration module for editing these same files. Because any file (not just .htaccess files) can be edited, you should make sure that file permissions on your system are adequate before giving this module to users.
Change Language	Lets users change the language in which the Usermin interface is displayed. Quite harmless.
Change Password	Allows users to change their UNIX passwords, and possibly their Samba passwords as well. Because it uses the passwd PAM service or the passwd program, normal password length rules apply. Harmless as long as your system has been configured to enforce reasonable password restrictions.
Change Theme	Lets users change the theme that is used to render the Usermin interface when they log in. This module poses no risks at all.

Table 47.2 Standard Usermin Modules (Continued)

Change User Details	This module lets users change their UNIX account details such as the real name, home phone, and office location. Users can also change their login shell, although this feature can be disabled. Relatively safe, as you can restrict the allowed shells to those from the <code>/etc/shells</code> file or some other.
Command Shell	This module is identical to Webmin's Command Shell module, except that commands are run as the logged-in Usermin user. Because any command can be run, you should not give this module to users whom you do not trust with SSH or telnet access.
Custom Commands	Lets users run commands defined in Webmin's Custom Commands module, but not create or edit their own. As long as your command scripts are well written, this module is safe. Remember that commands can be run as other UNIX users, not just the logged-in Usermin user.
Disk Quotas	This module lets users see their current block and file quotas and the number of each used. Totally harmless, as it only displays information.
Fetchmail Mail Retrieval	Using this module, users can edit their personal <code>.fetchmailrc</code> file to set up email downloads from other servers. It can also be used to start the Fetchmail daemon process to check for mail at regular intervals. Because Fetchmail can be configured to run an arbitrary commands before connecting to a mail server, this module should not be granted to anyone that you would not trust with SSH or telnet access.
File Manager	Lets users explore and edit the files and directories on your system, subject to normal UNIX permissions. This module has a similar interface to the Webmin File Manager module, but without the ability to configure file sharing. It can be configured to limit users to their home directories, which makes it reasonably safe.
GnuPG Encryption	This module can be used by people to set up GnuPG, manage keys, encrypt, decrypt, sign, and verify files. It is quite safe, although some of its features can be used to read files on your system—subject to normal UNIX permissions.
HTTP Tunnel	This module allows the system administrator to create an icon in Usermin that actually connects users to another web server. However, instead of just linking to some URL, all requests are tunnelled through Usermin, and so do not appear to come from the user's browser at all. From a security point of view, it is totally harmless.

Table 47.2 Standard Usermin Modules (Continued)

Login Scripts	Allows users to edit their <code>.profile</code> , <code>.login</code> and <code>.cshrc</code> script files that are run when logging in via telnet, SSH, or at the console. If a user cannot log in, these files will never be run and thus the module is quite safe (but useless). If a user can log in, however, then he can run arbitrary commands anyway.
Mail Forwarding	This module lets users edit their <code>.forward</code> or <code>.qmail</code> files to set up mail forwarding for their addresses. It can be configured to prevent the use of programs or writing mail to arbitrary files, which makes it quite safe.
Mount Filesystems	On Linux systems, this module can be used to mount and unmount filesystems that have been set up in Webmin's Disk and Network Filesystems module to be mountable by users. Safe as long as you don't allow the mounting of potentially dangerous filesystems, such as <code>ext2</code> -formatted floppy disks containing setuid-root executables.
MySQL Database	This module uses an interface very similar to the Webmin module of the same name to let users log in to a MySQL server and manage tables and records. It can be configured to restrict which databases a user can see and edit, and of course normal MySQL permissions apply. This makes it quite safe.
Plan File	The <code>.plan</code> file in a user's home directory is displayed when someone uses the <code>finger</code> command to look up information on the user. This module is completely safe, but useless if the <code>finger</code> service is not enabled in <code>inetd</code> on your system.
Procmail Mail Filter	This module lets a user edit his <code>.procmailrc</code> file to configure how mail sent to his address is delivered. Its interface is identical to that of the Webmin module with the same name. Because Procmail can be set up to deliver email to an arbitrary program, this module should not be given to users who are not trusted with SSH or telnet login access. It is useless and harmless, however, if Procmail is not installed or set up on your system. See Chapter 45 for more details.
Read Mail	This is probably Usermin's most powerful module, as it is a complete mail reading and sending interface. It supports multiple folders or different types, an address book, sent mail and drafts files, and GnuPG encryption and verification of outgoing and incoming email. As long as users are not allowed to attach files on the server, it is perfectly safe.
Running Processes	Allows users to view and kill processes that they own on your system, as well as starting new ones. It is like the Webmin module of the same name, but runs with the privileges of the logged-in user instead of <code>root</code> . Users who are not allowed to have shell access should not be granted access to this module either.

Table 47.2 Standard Usermin Modules (Continued)

SSH Configuration	Lets users configure their personal <code>.ssh/config</code> file and manage SSH keys. Because it only allows the editing of specific files, it is totally safe to give to users, but useless if they cannot log in with SSH.
SSH/Telnet Login	This module just displays a Java applet for connecting to your system via SSH or telnet. Because they must still log in as they would with any telnet client program, it is totally harmless.
Scheduled Commands	This module can be used to create and remove <i>At</i> jobs, which are shell commands to be run once at a certain date and time. It is similar to the Webmin module with the same name, but only allows the creation of jobs that run as the logged-in user. Because an <i>At</i> job can execute any command, it should not be given to users who are not trusted with shell access.
Scheduled Cron Jobs	Users can use this module to create, edit, delete, and run their own Cron jobs. Again, it should not be granted to users who do not have shell access, as it can be used to run any command.
System Documentation	This module is identical to Webmin's System Documentation module. Because it only allows the viewing of manual pages and other documentation files, it is quite safe—as long as no files exist in documentation directories that you don't want people to read!
Upload and Download	This module allows users to upload multiple files to their home directories on the Usermin server and to download multiple URLs to the Usermin system. It should not be granted to any user that you do not trust to write to files.

You might be wondering what is so harmful about letting users run commands on your server. The reason is that many more security holes exist for UNIX systems that can give a normal user `root` privileges than those that allow some other system on the network to gain `root` access. Any user who can run a command can potentially exploit one of these holes, so it is better to avoid this where possible. Users who can run commands can also use up large amounts of memory, CPU time, or network traffic by starting resource-wasting processes, which can make your system nearly unusable.

47.18 Configuring the Usermin Configuration Module

This Webmin module has only a single setting that can be changed by clicking on the **Module Config** link on its main page. It is shown in Table 47.3.

Table 47.3 Module Configuration Option

Usermin configuration directory	This field must contain the full path to the configuration directory chosen when Usermin is installed. Because this is almost always /etc/usermin, this field does not generally need to be changed.
--	--

47.19 Summary

This chapter has introduced Usermin—a web interface similar to Webmin but designed for use by normal UNIX users. It has explained how Usermin should be installed and how it can be configured using the Webmin module. After reading the chapter, you should understand how to change Usermin’s various networking and authentication settings, how to install and remove modules, how to configure those modules, and how to control which UNIX users can log in and what they can do.

Cluster Software Management

This chapter introduces Webmin's clustering system, and explains how to use the module for installing software packages on multiple systems concurrently.

48.1 Introduction to Webmin Clustering

Webmin has several modules that make it easy to perform tasks on several machines at once. There are known as a cluster. A large organization might have tens or hundreds of servers that need some software package installed, UNIX user created, or Webmin module added. The cluster modules make this easy. Each corresponds to one of the single-machine modules, but allows the same tasks to be performed on more than one system at a time.

For a system to be part of a cluster, it must have Webmin installed, even if you never actually login to it directly. One of the cluster modules on a single host contacts all of the others using Webmin's RPC (Remote Procedure Call) protocol and instructs them to carry out certain tasks. This master host might be part of the cluster and instruct itself to perform the same tasks, or it may be totally independent.

On the master system, the Webmin Servers Index module (covered in Chapter 53) must first be used to register all of the other managed servers. For each managed server, the `root` or `admin` username and password must be specified so that the master knows how to log in. Once this is done, each of the cluster modules can be set up to manage some or all of the registered systems.

Because Webmin's RPC mechanism allows any file to be accessed or any command run on a server, only the users `root` and `admin` are allowed to receive RPC calls on a managed system by default. This means that entering some other user in the Webmin Servers Index module for a managed server will not work, unless that user has been specifically configured to be able to accept RPC logins. Section 52.5 ‘Editing Module Access Control’ explains how to set this up.

The RPC protocol that the master system uses to control managed hosts is unique to Webmin, and is not based on any other similar protocol, such as Sun's RPC, SOAP, or RMI. It has two different modes: the old mode in which only HTTP requests are used to send commands, and a newer mode in which a permanent TCP connection is used. The latter method is faster and more reliable, but may fail if a firewall is blocking traffic between the master and managed hosts. It uses ports 10001 and above, by default, whereas the old protocol just uses the port on which Webmin accepts normal connections (usually 10000). Chapter 53 explains how to select a mode for a server in more detail, while Chapter 56 covers the internal workings of the protocol.

48.2 The Cluster Software Packages Module

This module allows you to install, view, and delete packages on multiple systems at once. If you need to roll out some program to a large number of systems, this module can be used to perform the installation with a single action. The alternative is to install manually on each host or to use NFS to share program files from a single server to multiple clients.

Before reading on, you should have a complete understanding of how the regular Software Packages module works, what packages are, and what they can do. Chapter 12 covers all of these in detail, so read it now. The user interfaces of the two modules are very similar, and the instructions in this chapter assume that you are familiar with the regular module.

One limitation of the Cluster Software Packages module is that the master system and all managed systems must use the same package system, such as RPM, DPKG, or the Solaris package format. This makes sense when you think about it because there is no way that a single package file of a specific type can be installed on multiple systems if some of them do not support that packaging format. If the hosts on your network use different package formats, you will need to set this module up once for each format in use, on different hosts.

Only on operating systems that have a supported package system will the module appear. At the time of writing, only RPM, Debian's package format, the Gentoo package format, and the Solaris and SCO OpenServer package systems are supported. Even though the Software Packages module supports a few more formats, they are not currently usable with this package.

The module itself can be found along with the other cluster-related modules in Webmin's Cluster category. Clicking on its icon will bring up the main page, an example of which is shown in Figure 48.1. At the top is a list of icons representing managed servers registered in the module, and below them are forms for searching for and installing packages. The latter forms will only appear if some systems have been registered, however, which will not be the case the first time you use the module.

To speed up searching, the module keeps a list of all the packages installed on the systems that it manages. This means that any packages installed or removed directly on one of those systems without using this module will not be detected until the lists are refreshed. This may cause the module to incorrectly report that a package exists when it really does not or vice versa. To avoid this problem, always use the Cluster Software Packages module to add or delete packages from managed hosts. Or, use the **Refresh package lists** button (explained later) to update the lists after making direct changes.

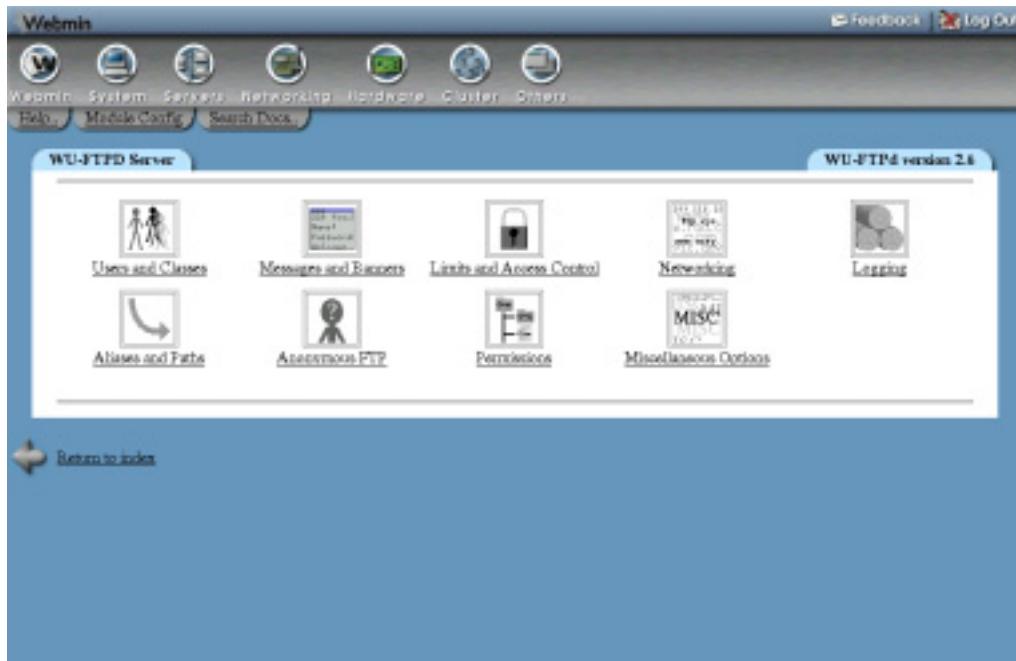


Figure 48.1 The Cluster Software Packages module.

48.3 Registering a Server

Before this module can be used to manage another system's software, that system must be added to its list of servers. To do this, follow these steps:

1. Use the Webmin Servers Index module to add the remote system, and make sure you provide a username and password. This does not have to be done, however, if you want to manage the master server.
2. In this module, select the system from the menu next to the **Add server** button and then click it. The menu will usually include the special entry **this server**, which is the master system. It will never, however, include any servers that have already been added.
You can also select an entire group of servers from the menu next to **Add servers in group**. Groups can be defined in the Webmin Servers Index module as well.
3. A page showing all of the hosts added and the number of packages on each will be displayed. If a host cannot be contacted or the RPC login fails, an error message explaining what went wrong for that host will appear.
4. Return to the module's main page, on which a new icon for each host should now be listed.

The most common cause of problems when adding a server is an incorrect username or password entered for that host in the Webmin Servers Index module. You must provide the **root** or **admin** login, not that of some other user. Adding can also fail if a firewall is blocking connections between the two hosts, or if the master Webmin server is configured to use an HTTP proxy that is disallowing the RPC HTTP request.

48.4 Installing a Package

Packages can be installed on multiple hosts using the Cluster Software Packages module in a similar way to how they are installed on a single host in the Software Packages module. You should read Section 12.3 “Installing a New Package” first, which explains the differences between the various package systems when it comes to installation.

To install a package on multiple hosts, follow these steps:

1. On the main page, scroll down to the **Install a New Package** form.
2. If the package file is already on the master system, select **From local file** and enter its full path into the adjacent text field. If some of the managed systems use NFS to share files with the master and if the package file exists in the same directory, this option is the most efficient as it avoids the need to transfer the file to each managed host using RPC. Instead, the remote Webmin server will just read it directly from the NFS-mounted filesystem.
3. If the package file is on the PC on which your browser is running, choose **From upload files** and click the **Browse** button to select it.
4. If the file is on some web or FTP site, select **From ftp or http URL** and enter the full URL into the text field. Normally, the master server will download the file and then transfer it with RPC to each managed host. If the **Each server should re-download package** box is checked, each host will perform the download instead, which is more efficient if the URL refers to a web server on your local network.
5. Click the **Install** button to go to a page showing the progress of the package file’s download (if necessary), the package name, and a form for choosing installation options. These options depend on the package system in use and are documented in more detail in Chapter 12.
6. By default, the package will be installed on all managed systems. However, you can limit it to just one or the members of a group by making a selection from the **Server(s) to install on** menu. This can be useful if the package is only appropriate for certain systems. You can also select **hosts that don’t have it** to tell the module to skip installation attempts on systems that already have any version of this package. This will prevent upgrades from being attempted as well.
7. Click on **Install** again to go ahead. This will bring up a page showing the results from each managed host. It is quite possible for installation to succeed on one system but fail on another due to dependency problems or because the package is already installed. Installations will be done simultaneously on all managed systems so that you don’t have to wait for them to complete one by one.

48.5 Searching for Packages

This module can be used to quickly search for packages across all managed hosts, as it keeps its own local host of installed packages on each system. Follow these instructions to search for and display the details of packages:

1. On the module’s main page, enter a search term (such as *mozilla*) into the field next to the **Search for package** button. When the button is clicked, a page listing all matching packages will appear, or will contain an error message if none were found. If exactly one package matches, you will be taken directly to its editing page.

2. If multiple packages match, click on the name of the one that you want to view in the list. This will bring up an editing page showing its complete details and icons for each of the hosts on which it is installed. The details are fetched from the first system that has it installed, or the master server, if possible.
3. To see the files that the package contains, select a host from the menu next to the **List files on** button. Clicking on it will open a page showing the details of files in the package from that host, just like the similar list in the Software Packages module.
4. To view the details of one of the hosts on which the package is installed, click on its icon on the package editing form. This will take you to the page covered in Section 48.7 “Exploring and Removing a Server”.

It is quite possible for many different versions of the same package to be installed on different systems in your network. This can make the package details form a little confusing, as it might show the details of version 1.0 of some package when most of your systems are really running version 2.0. The lists of files in a package can also vary significantly between versions and between different packages of the same program from various Linux distribution vendors.

48.6 Deleting a Package

If it is no longer needed, an installed package can be removed from one or all hosts using the Cluster Software Packages module. You can delete a package by following these steps:

1. Find the package that you want to remove by following the instructions in Section 48.5 “Searching for Packages”.
2. To delete from just one host, select it from the menu next to the **Uninstall from** button. To remove from all, leave **<all hosts>** selected. Only hosts that the module knows the package is on will be included in the menu.
3. Click the button to bring up a confirmation page showing the number of files and bytes that will be removed. Depending on the package system, this page may contain fields for setting uninstallation options, such as whether dependency checking is done or not.
4. Hit the **Delete** button to go ahead with the removal. The deletion will be done simultaneously on all chosen systems to speed up the process. A page showing the results from each system from which it is being deleted will be displayed, indicating whether or not it succeeded or why it failed. Those most common causes of failure is a dependency on this package by another package. If **<all hosts>** was selected, the module will only attempt to remove it from systems on which it thinks the package is installed.

48.7 Exploring and Removing a Server

Using the Cluster Software Packages, you can view the details of a managed system and the packages that it specifically has installed, which can be useful if your systems have different package sets. If you no longer want to control software on the system, it can be deleted from this module, as well.

To view the details of and packages on a managed server, use the following steps:

1. Click on the server's icon on the module's main page. This will bring up a page showing the operating system the host is running and a tree of package categories. Just like in the

Software Packages module, you can click on category names in this tree to open them up and view the subcategories and packages that they contain.

2. To view the details of a particular package, click on its icon in the tree. Each icon links to the package editing form explained in Section 48.5 “Searching for Packages”, from which you can delete it from one or all hosts. The displayed details, however, will not necessarily come from this managed system.
3. To remove this system from the module’s control, click on the **Remove from managed list** above the package tree. This will only delete the master system’s copy of the installed package lists, so the removal will happen without asking for confirmation.

48.8 Refreshing the Package List

If packages are installed or removed from a managed system by some method other than this module, its lists of packages will no longer be correct. This is fine as long as the lists are refreshed afterwards. To refresh a package list, follow these steps:

1. Click on the **Refresh package lists** button on the main page.
2. A page showing the results from each managed system will be displayed, listing any new packages found or old ones that no longer exist, or an error message will show if a system cannot be contacted for some reason. As with installs and deletions, the refresh will be done in parallel to speed up the process if you have a large number of managed servers.

48.9 Configuring the Cluster Software Packages Module

This module has only a single user interface-related setting, shown in Table 48.1, that can be changed by clicking on the **Module Config** link on its main page.

Table 48.1 Module Configuration Options

Sort hosts by	Controls the order in which host icons are listed on the main page and package editing form. The options are: Hostname Systems are sorted by their hostname. Order added Systems are listed in the order that they were added to this module. Description Systems are sorted by description, which will be the same as the hostname if no description is set.
----------------------	---

48.10 Summary

After reading this chapter, you should be able to manage the software packages installed on multiple UNIX systems of the same type from a single interface. It has explained how to add hosts to manage and how to remove those hosts from the managed list. It has also covered the installation of packages, searching for a display of the details of packages, and the removal of packages—all on multiple systems at once.

Cluster User Management

In this chapter the Webmin module for managing users and groups across multiple systems is explained.

49.1 The Cluster Users and Groups Module

Before reading this chapter, you should be familiar with Webmin's cluster management capabilities, explained in the introduction to Chapter 48. All of the cluster-related modules (this one, Cluster Software Packages, and Cluster Webmin Configuration) make use of the Webmin Servers Index module and RPC to control other systems. You should also read Chapter 4, which covers the Users and Groups module, as many of the forms and pages in the Cluster Users and Groups module are similar to that one.

This module allows you to manage UNIX users and groups on multiple systems from a single interface. If you have a large number of hosts on your network and want people to be able to log in to all of them, some mechanism is needed for creating UNIX accounts on each system. Using this module is far easier than manually creating an account on each system.

There is a widely available and more commonly used method of managing users, groups, and other services across multiple machines—it's NIS (covered in Chapter 17). NIS client systems query a master server for information as well as reading their /etc/passwd and /etc/group files, which makes the accounts available on all clients. NIS works well and is easily configured from within Webmin, but has some negatives. If the master server goes or the network goes down, client systems will be unable to look up user information, causing logins and many programs to hang. And, because clients must frequently query the server, it does not work as well over a slow network.

This Webmin module, on the other hand, updates the files on each client system so that users and groups remain synchronized. The client operating system does not need to do anything special to make use of centrally managed users—to the client OS, they appear just like other

users. This means, however, that a loss of synchronization can occur if a user is modified directly on a client system instead of through the master server.

Another useful feature of this module that NIS lacks is its ability to create home directories on managed servers. This can be useful if your systems do not share common home directories via NFS, which can be impractical on a wide-area network. The module can also set up users in the Samba password file or in a proxy authentication file on managed servers—just like the normal Users and Groups module can locally. This is very handy if your organization has multiple Samba servers, each with its own password list (although Samba can be configured to query a central server for passwords instead).

The Cluster Users and Groups module requires that all managed systems have the same user file formats. Unfortunately, some UNIX variants use just an `/etc/passwd` file, some use an `/etc/shadow` file as well, and some BSD systems use the `/etc/master.passwd` file for storing users. Each of these different formats stores different information about users, which the module cannot handle. The result is that a cluster cannot contain both Linux and FreeBSD systems or both Solaris and AIX boxes, as they use different formats. A network of Linux and Solaris hosts, however, can be managed centrally because both operating systems use the `/etc/passwd` and `/etc/shadow` files, which is the most common format.

Like the Cluster Software Packages module (covered in Chapter 48), this one stores lists of users and groups on each managed host on the master system. This speeds up searching and editing, but introduces the possibility that the master's information may get out of sync with the real lists of users and groups on managed hosts. This can happen if a user is added, deleted, or modified directly on one of the hosts instead of through this module. Fortunately, it is easy to resynchronize if this happens (using the **Refresh user and group lists** button). Refreshing also happens automatically every time a user is added, deleted, or updated on a managed server.

The Cluster Users and Groups module can be found in Webmin under the Cluster category. When you click on its icon, a page like the one shown in Figure 49.1 will be displayed. At the top are icons for all of the managed servers, and below them fields and buttons for finding and adding users and groups. These fields will only appear, however, if at least one server has been registered.

49.2 Registering a Server

Before this module can be used to manage users and groups, the system to be managed must be added to its list of servers. To do this, follow these steps:

1. Use the Webmin Servers Index module to add the remote system and make sure you provide a username and password. This does not have to be done if you want to manage the master server itself.
2. In this module, select the system from the menu next to the **Add server** button and then click it. The menu will usually include the special entry **this server**, which is the master system. It will never include any servers that have already been added.

You can also select an entire group of servers from the menu next to **Add servers in group**. Groups can be defined in the Webmin Servers Index module, as well.

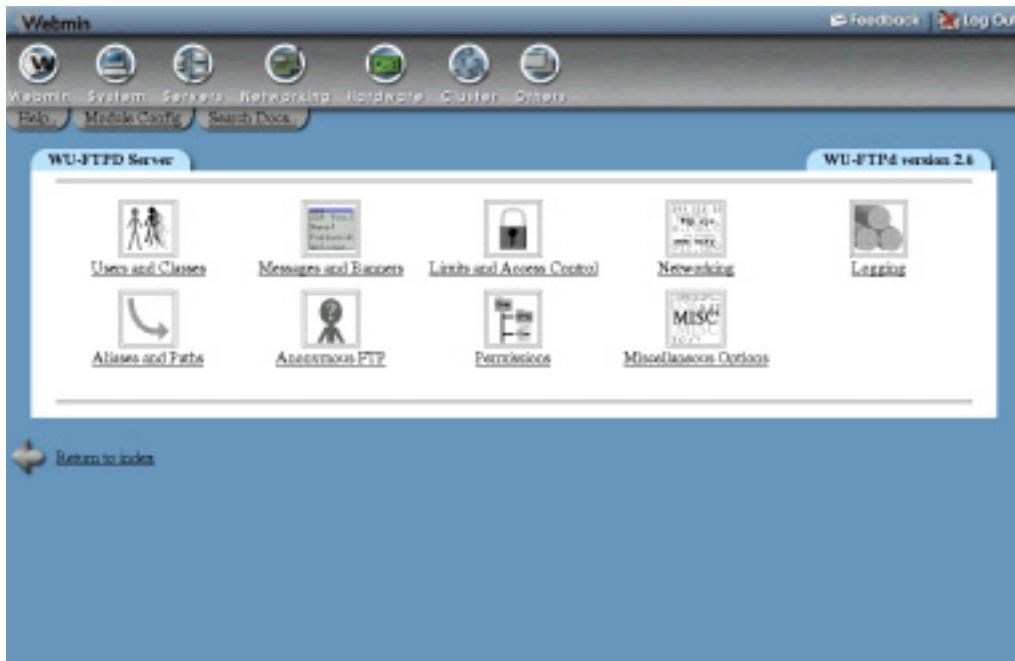


Figure 49.1 The Cluster Users and Groups module.

3. A page showing all of the hosts added, and the number of users and groups on each, will be displayed. If a host cannot be contacted, or if the RPC login fails, an error message explaining what went wrong for that host will appear.
4. Return to the module's main page, on which a new icon for each host should now be listed.

49.3 Creating a New User

The form for adding a UNIX user to multiple systems is almost identical to the one in the Users and Groups module for adding a user locally. If you are familiar with that module, using this one should be a breeze. Just follow these steps:

1. On the module's main page, click on the **Add User** button in the bottom half of the page.
2. Fill in the user creation form that appears just as you would when creating a local user. The **User ID** will be set by default to an ID that is not in use on any system, and so should not need to be changed.
3. The only field to be careful of is **Primary group**, as the group ID for the entered group name will be looked up on the master system. The same group with the same ID should exist on all of the managed hosts as well.
4. The **Secondary group** list includes groups from all systems. If you select one that only exists on some hosts, the user will only be added to that group for the hosts on which it exists.
5. Near the end is a field labeled **Do above file operations on** which determines if home directory creation and file copying is done on just one host in the cluster or all of them. If

your managed systems share home directories via NFS, you should select **One server** so that it is only created once. Otherwise, choose **All servers** so that the user's directory is created on each of them.

6. If **Create user in other modules** is set to **Yes**, the user will be added to the Samba password file, Squid user list, and so on, for each system.
7. Hit the **Create** button to go to a page showing what was done on each managed host, as long as there were no errors in the form. If for reason a host cannot be contacted or logged into, an error message will appear for that host, but all the rest will be updated.

This process cannot be used to add a user that already exists on some of the hosts, as an error message to that effect will be displayed when you hit **Create**. Instead, you should use the module's synchronization feature, covered in Section 49.10 “Synchronizing Users and Groups”, which can copy user details from one host to others.

49.4 Editing an Existing User

Editing a user on multiple servers is slightly more complex than adding one, as you can control exactly which of the user's attributes will be changed. This is necessary because the user may not have the same details on each of the managed systems, and you may want to set some attribute (such as the real name) while leaving another that differs on various systems (such as the shell) intact.

For this reason, the user editing form shown in Figure 49.2 is similar to the form in the Users and Groups module, but has an additional **Don't change** option for each field. The current value of that attribute from the host shown at the top of the page is displayed so that you have some idea of what it is set to—at least on one system.

With this in mind, you can edit a user by following these steps:

1. Scroll down to the **Find users whose** form on the main page, which is used to search for users to edit.
2. If you know the name of the user, just select **Username** from the first menu, **equals** from the second, and enter the name into the third text box. Hitting **Find** will bring up the user's editing form, assuming that he exists.

If not, the form can be used to find users matching some criteria. Select the attribute to search on from the first menu, the type of search to perform from the second, and enter some text or Perl-style regular expression into the text field. Hitting **Find** will take you to a page listing all users that match and clicking on one will bring up its editing form.

3. Once you make it to the editing page, choose the **Set to** option for each of the fields that you want to change and enter or select a new value. One exception is the **Username** field. Changing this will cause the user to be renamed on all managed servers.
4. Because the user may be a member of different secondary groups on different systems, the **Secondary groups** field for choosing which he belongs to is more complex than in the Users and Groups module. To leave his secondary membership unchanged, select **Don't change**. To add him to one or more groups on systems that have them, select **Add to groups** and fill in the text field next to it. To remove him from groups on systems of which he is a member, select **Remove from groups** and enter the names of those groups into its text field.



Figure 49.2 The user editing form.

5. In the **Upon save** section, the **Do above file operations on** field determines if any home directory renaming or file UID changes are done on just one host in the cluster or all of them. You should select **One server** if your systems use NFS to share home directories, or **All servers** if they do not.
6. If **Modify user in other modules** is set to **Yes**, the user will be updated in the Samba password file, Squid user list, and so on for each managed system. This is not necessary and wastes time if the user was never added to other modules when created in the first place.
7. To go ahead with the modifications to the user that you have selected, click the **Save** button at the bottom of the form. A page listing all hosts on which the user exists and the actions taken will be displayed. Any errors encountered when connecting to a particular host will be shown as well, along with any problems encountered when updating the user (such as it no longer existing on the host). A failure updating one host, however, will have no effect on the others.

At the bottom of the user editing form is a list of icons, one for each of the systems on which the user exists. You can click on one of them to bring up the server page covered in Section 49.11 “Listing and Removing a Server”.

49.5 Deleting a User

Deleting a user from multiple systems is comparatively easy. As with the Users and Groups module, you must be careful when removing a user, as his home directory and everything that it contains will be deleted as well.

The steps for deleting a user are:

1. First bring up the editing form for the user that you want to get rid of, as explained in Section 49.7 “Editing an Existing Group”.
2. Click on the **Delete** button in the bottom-right corner of the page to go to a confirmation page.
3. If NFS is being used to share home directories between all your systems, select **One server** for the **Delete home directory if exists on** field. Otherwise choose **All servers** to force the removal of the directory on each system.
4. To have the user removed from the Samba password file, his mail file deleted, his Cron jobs removed, and so on for each server, select **Yes** in the **Delete user in other modules** field.
5. If you want to preserve the user’s home directory, click on **Delete User**. To remove it along with the account, click on **Delete User and Home Directory**, instead. Either way, a page listing each server on which the user exists the tasks performed on each, and any errors encountered will be displayed.

49.6 Creating a New Group

The process for adding a group to all of your managed servers is identical to that for adding a group locally in the Users and Groups module. Just follow these simple steps:

1. On the module’s main page, click on the **Add Group** button.
2. Fill in the creation form that appears just as you would in the Users and Groups module. The only difference is that the **Members** field can contain users from any of the managed systems. The **Group ID** will be set automatically to an ID not in use on any system.
3. Hit the **Create** button to add the group. A page showing the module’s progress as it updates each managed server and any problems encountered will be displayed. Failure adding the group to some system will not affect the rest.

Once the group has been created, you can create or edit users to make them primary members of it. This should only be done for groups that are identical on all managed systems, such as those created by following the preceding steps.

49.7 Editing an Existing Group

Editing a group is similar to editing a user, in that you can choose which of the group’s attributes to change. The group editing form is similar to the one in the Users and Groups module, but has extra **Don’t change** and **Set to** options for each field. Figure 49.3 shows an example.

The steps to follow to change the details of a group are:

1. If you know the exact group name, just enter in into the text field in the **Find groups whose** form on the module’s main page and hit the **Find** button. If not, a group can be found by selecting an attribute on which to search and a match type in that same form, just as you can when searching for users. Section 49.7 “Editing an Existing Group” explains more.

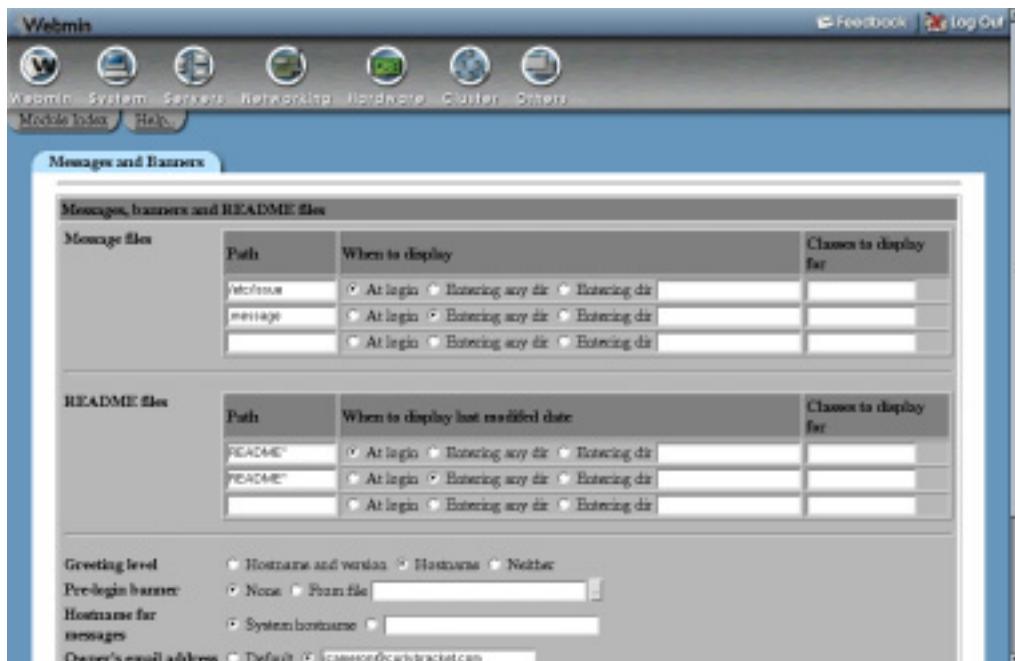


Figure 49.3 The group editing form.

2. On the editing page for the group, select **Set to** for any fields that you want to change. Next to the **Don't change** option for each will be the current value, taken from the system shown at the top of the page.
3. The **Members** field is different to the one on the group editing form in the Users and Groups module because a group's members may differ on different systems in your cluster. You can either select **Don't change** to leave membership alone, **Add users** to add the users entered in the adjacent text field (if they exist on each system), or **Remove users** to remove the specified users (if they are members on each system).
4. As when editing a user, the **Do above file operations on** field determines if any necessary group ID changes on files are done just on one managed system or all of them. If your hosts all have separate home directories that are not shared with NFS, or if **All files** was chosen for the **Change group ID on files?** field, you should choose **All servers**. Otherwise stick with **One server**. Of course, the choice is irrelevant if the group ID is not being changed.
5. Hit the **Save** button to begin the process of updating the group. A page showing the tasks performed on each system on which the group exists will be displayed. If an error of some kind occurs, it will be shown under the affected system's name, but will not prevent the group from being updated on other hosts.

At the bottom of the group editing page is a list of icons for the systems on which this group exists, just like on the user editing page. Clicking on one will take you to the host form covered in Section 49.11 “Listing and Removing a Server”.

49.8 Deleting a Group

Removing a group is much safer than removing a user, as no files are deleted. The module will even stop you from deleting a group if it has any primary members on any systems, just like the normal Users and Groups module does for your local system.

To remove a group, follow these steps:

1. Use the **Find groups whose** form on the module's main page to get to the group editing page, as explained in Section 49.7 “Editing an Existing Group”.
2. Click on the **Delete** button below the form. As long as no primary members exist, a confirmation page will be displayed asking if you really want to delete the group.
3. Hit the **Delete Group** button to go ahead. As usual, the progress of the deletion and any errors encountered on each host on which the group exists will be displayed.

49.9 Refreshing User and Group Lists

If users or groups are added or changed in any way on one of the managed servers without using this module, its cached lists will no longer be accurate. This may cause the module to attempt the modification or deletion of users that no longer exist or to create a user on a system on which it already exists. Fortunately, the caches can be resynchronized as follows:

1. Click on the **Refresh user and group lists** button at the bottom of the module's main page.
2. A page listing all of the systems managed by the module will be displayed, along with the number of users and groups added or deleted from each that are not in the local cache. If, for some reason, a host cannot be contacted, an error message will be displayed, but this will not effect the refreshing of other systems.

49.10 Synchronizing Users and Groups

Synchronization is possibly the module's most powerful feature, but also one of the trickiest to use. It can be used to create users or groups that exist on only one system on all of the other systems in your cluster. This is handy if certain users were created outside of this module on only one host and you want to now make them available on all hosts. It is also useful if a new host is added to the cluster and you want to give it all of the users and groups that the other systems have.

Synchronization, however, can have unexpected and possibly harmful effects if you use it incorrectly. For example, simply synchronizing all users on all hosts would be a bad idea, as it could trigger the creation of system users like `uucp` and `squid` on hosts that do not have them. For this reason, you should make use of the **Only show what would be done?** field to see what the module will do with your synchronization selections before applying them for real.

The synchronization feature will only create new users and groups, not update the details of those that already exist. Neither will it delete users or groups. Instead, it assumes that a mismatch between the users that exist on one system and those that exist on another indicates that users need to be created. The module's other features for editing and deleting users, however, can be used to update users on some systems to match another or delete users that only exist on some systems.

To create users that only exist on some of your systems, follow these steps:

1. Click on the **Synchronize** button in the lower-right corner of the module's main page. This will take you to the form shown in Figure 49.4.
2. The **Servers to synchronize** field determines which systems are checked as part of the process. You can either choose **All servers** to synchronize every managed system, or choose **Selected** and select some of the systems in the list. In the latter case, specified users that exist on any system may be added to those chosen.
3. The **Users to create** section lets you specify which users to synchronize. The available options are:
 - All missing users** This mode should never be used unless all your systems are running the exact same operating system as it will synchronize all users, including system users like squid and uucp.
 - No users** This option tells the module not to synchronize any users and thus does nothing.
 - Only users** When this option is chosen, only the users whose names are entered in the adjacent text field will be considered for synchronization. If you know exactly which users need creation, this is the option to use.
 - All except users** This option should be used with care (like **All missing users**), because it synchronizes all users except those listed in the adjacent text field.
 - Users with UID in range** This option tells the module to only synchronize users whose UIDs are within the range entered in the adjacent text fields.
 - Users with primary group** When this option is chosen, the module will only consider users for synchronization whose primary group matches the group name entered in the field next to it.
4. Leave **Groups to create** set to **No groups**.
5. Change the **Only show what would be done?** field to Yes, so that you can do a test run first.
6. If your systems share home directories with NFS, the **Create home directories?** and **Copy files to home directories?** fields can be set to **No** because the users' directories should already exist. If each system has its own filesystems, however, you should choose **Yes** instead to force the creation of a new empty directory for each added user.
7. To have the new users added to the Samba password file, Squid user list, and so on for each system on which they are created, change the **Create user in other modules?** field to **Yes**. Unfortunately, because users' unencrypted passwords are not available when synchronizing, Samba users will not be created properly.
8. Hit the **Create Users and Groups** button. A page listing all of the selected systems and the actions that need to be performed on each (if any) will be displayed. Check to make sure that only what you expect will be done. If a host already has all of the specified users, the message **Users and groups are in sync** will be displayed.
9. Use your browser's back button to return to the synchronization form and change the **Only show what would be done?** field to **No**.
10. Click on **Create Users and Groups** again to create the users for real. A page listing the selected systems and the actions that are actually being performed will be displayed, along with any errors that occur. As usual, a failure on one host will not affect the rest.



Figure 49.4 The synchronization form.

Missing groups can be created in almost exactly the same way. The only difference is that you should leave the **Users to create** field set to **No users**, but specify the groups to synchronize in the **Groups to create** section.

49.11 Listing and Removing a Server

This section explains how to view information about, and the users and groups on, a managed server or remove it from the list of systems controlled by the module. To do so, follow these steps:

1. On the module's main page, or a user or group editing form, click on the icon for the system that you want to view. This will take you to a page showing its operating system and listing the names of all known users and groups on the server.
2. To view the details of or edit a user, click on its name in the list. This will bring up the usual user editing form, but the current attributes displayed next to the **Don't change** options are taken from this server. You can also view and edit a group by clicking on its name on the server's page.
3. To remove the system from this module's control, click on the **Remove from managed list** button. No confirmation will be requested and you will be immediately returned to the module's main page. No data is lost though, as you can re-add the system at any time.

49.12 Configuring the Cluster Users and Groups Module

Like the Cluster Software Packages module, this one has only a single user setting that can be changed by clicking on the **Module Config** link on its main page. The option is shown in Table 49.1.

Table 49.1 Module Configuration Options

Sort hosts by	Controls the order in which host icons are listed on the main page and package editing form. The options are: Hostname Systems are sorted by their hostname. Order added Systems are listed in the order in which they were added to this module. Description Systems are sorted by description, which will be the same as the hostname if no description is set.
----------------------	---

Some of the module configuration settings in the Users and Groups module on the master server also affect the behavior of this module. The ones that apply are **Maximum user and group name length**, **Show office and phone details?**, **Automatic home directory base**, **Automatic home directory style**, **Generate password for new users?**, **Conceal plain-text password?**, **Default primary group for new users** and everything in the **Password restrictions** section. The **Permissions on new home directories** and **Copy files into new home directories from** settings in the Users and Groups module on managed systems are also used. See Section 4.12 “Configuring the Users and Groups Module” for a complete explanation of what they all do.

49.13 Summary

This chapter has explained how UNIX users and groups on several similar systems can be managed from a single interface on a master server. After reading it, you should know how to define, add, and remove systems. You should also know how to create, edit, and delete both users and groups on multiple systems at once and how to synchronize users and groups so that those missing on some hosts in the cluster are automatically created.

Cluster Webmin Configuration

This chapter tells you how to manage users, groups, modules, and themes on multiple Webmin servers from a single system.

50.1 The Cluster Webmin Configuration Module

Before reading this chapter, you should be familiar with Webmin’s cluster management capabilities, explained in the introduction to Chapter 48. All the modules in the Cluster category make use of the Webmin Servers Index module and RPC to control other systems. You should read Chapters 51 and 52, which cover the Webmin Configuration and Webmin Users modules, respectively, as this one can be used to perform many of the same tasks across multiple systems.

The Cluster Webmin Configuration module really has two purposes: the management of Webmin users and groups across on multiple systems, and the installation and removal of modules and themes. If your network has multiple Webmin servers, this module can be very useful for keeping their user lists and user access control settings synchronized. It also provides an easy way to roll out a new module to a large number of servers at once.

Like the other cluster modules covered in Chapters 48 and 49, this one keeps lists of modules, themes, users, and groups from each managed server on the master system. This speeds up searching, but creates the potential for inconsistencies between how the master thinks the other systems are configured and how they really are. For example, if you install a module on or upgrade a managed host, the master system will not know about it until it is manually refreshed, as explained in Section 50.10 “Refreshing User and Module Lists”.

When you click on the module’s icon under the Cluster category on Webmin’s main menu, the main page shown in Figure 50.1 will be displayed. At the top is a table of icons, one for each of the managed servers. Under each icon is the version of Webmin that is currently running, determined when it was added to the module or last refreshed. Assuming that you have some



Figure 50.1 The Cluster Webmin Configuration module.

servers listed, below them are forms for editing and adding users and groups, followed by more forms for installing and finding modules and themes.

50.2 Registering a Server

Before this module can be used to manage another host running Webmin, it must be added to its list of servers. To do this, follow these steps:

1. Use the Webmin Servers Index module to add the remote system, and make sure you provide a username and password. This does not have to be done if you want to manage the master server itself though.
2. In this module, select the system from the menu next to the **Add server** button and then click on it. The menu will usually include the special entry **this server**, which is the master system itself. It will never include any servers that have already been added.
You can also select an entire group of servers from the menu next to **Add servers in group**. Groups can be defined in the Webmin Servers Index module, as well.
3. A page showing all of the hosts added and the numbers of modules, themes, Webmin users, and groups on each will be displayed. If a host cannot be contacted or the RPC login fails, an error message explaining what went wrong for that host will appear.
4. Return to the module's main page, on which a new icon for each host should now be listed.

50.3 Creating a New Webmin User

If you are familiar with using the Webmin Users module to create a new local user, creating one on multiple systems with this module should be easy. The form it uses has a slightly different layout, but all of the fields it contains have the same meanings. The rarely used **Categorize modules?** field does not exist, however, nor does the **SSL certificate name** field which does not make sense to set across multiple servers.

To create a user on all managed systems, follow these steps:

1. Click on the **Add User** button the module's main page to bring up the creation form.
2. Fill in most of the fields just as you would in the Webmin Users module. The fields to be careful of are explained in the next several steps.
3. The **Member of group** menu includes groups from all managed systems and, thus, some may not exist on some servers. If the user is added to a system that does not have the chosen group, it will be as though **<None>** was selected for that system.
4. Similarly, the **Personal theme** menu includes themes that may not exist on some systems. If the user is added to a system that does not have the chosen theme, it will be as though **Server default** was selected.
5. The **Modules** section lists all available modules from all servers. You can either select modules individually by control-clicking or shift-clicking on the lists or by using the **Select all**, **Select none**, and **Invert selection** links below them. As with the theme, it is possible to select modules that only exist or are supported on some managed systems.
6. When you are done filling in the form, hit the **Create** button at the bottom. This will bring up a page showing the success or failure of the module's attempt to add the user to each managed server. Once the process is complete, people will be able to log in with the new account on any of your systems.

50.4 Editing or Deleting a Webmin User

Like in the Cluster Users and Groups module, when editing a user you can choose exactly which of its attributes to change. This is useful because the user may have been created independently on multiple systems without the benefit of this module, and thus may not have the same settings on all of them. For example, you can change a user's language on all systems without touching his personal theme, which may be different depending on how fast each server is.

The following steps allow you to edit a Webmin user:

1. On the main page, select the user's name from the menu next to the **Edit user** button. Hitting the button will then take you to an editing form like the one shown in Figure 50.2.
2. In each of the fields that you want to edit, select **Set to** and enter a new value in the text box or menu next to it. The **Leave unchanged** option has the attribute's current value displayed next to it, taken from the server shown in the form's header.
The only exception is the **Username** field, which is just a text box that you can edit if you want to rename this user on systems on which it exists.
3. The **Modules** section works slightly differently, as it allows you to add or remove selected modules from the user on all systems. This is useful if he has different modules



Figure 50.2 The user editing page.

available on different hosts and you want to grant access to another one without disturbing those already assigned. The options in this section are available on different hosts and you want to grant access to another one without disturbing those already assigned. The available options in this section are:

Leave unchanged The user's assigned modules will not be touched on any managed hosts.

Only selected modules The selected modules will be granted to the user, overriding any that he currently has on all systems. Be careful with this option, because the list will not have currently assigned modules selected by default.

Add selected modules Modules selected from the list will be added to those that the user already has on all systems.

Remove selected modules Selected modules will be taken away from those assigned on all systems if the user actually has access to them.

As on the user creation form, you can either choose modules from the list by clicking on them, or use the links below the list to select a large number at once.

4. Hit the **Save** button to start the process of updating the user. A page listing all hosts that he exists on will be displayed, along with the success or failure of the attempt to update on each. Generally a user modification should only fail if one of the managed servers is down, or if the user has been deleted

Deleting a Webmin user is even simpler, although you should be careful not to remove the `root` or `admin` user on a managed system that the master server logs in as. Unlike the Webmin Users

module, this one will not stop you from doing things that can mess up your Webmin server, such as deleting the user you are currently logged in as. So, be careful!

To remove a user, follow these steps:

1. Use the **Edit user** button on the main page to bring up the user's editing form.
2. Hit the **Delete** button down near the bottom-right corner. The user will be immediately removed from all systems on which he exists with no confirmation and a page showing the results from each will be displayed.

50.5 Creating a New Webmin Group

Creating a group on multiple servers in this module is just like creating one locally in the Webmin Users module, except that the module selection part of the form is slightly different. Chapter 52 explains in detail what groups are, how they work, and what they are useful for. The same principals apply when creating and using groups on multiple systems.

To add a group, follow these instructions:

1. Click on the **Add Group** button on the module's main page to bring up the group creation form.
2. Enter a name not used by any other user or group on any system into the **Group name** field.
3. If this group should inherit modules and access control settings from some other group, select it from the **Member of group** menu. All groups from all systems are listed, so it is possible that during the creation process the group will be added to a system on which its parent does not exist. If this happens, it will be as though <None> was selected.
4. From the **Modules** lists, select the modules that will be eventually assigned to members of this group, either by clicking on them or using the links below.
5. Hit the **Create** button to begin creating the group. A page showing whether it succeeded or failed on each managed system will be displayed. A failure to create on one (because it is down or the RPC login is incorrect) will not affect the rest.
6. Once the group has been added, you can assign users or other groups to it using this module. It is best to only use groups created like this that exist and are the same on all managed systems so user details remain in sync across all servers.

50.6 Editing or Deleting a Webmin Group

As with users, when editing a group you can choose exactly which of its attributes to change in case the group differs between your managed systems. To edit a group, follow these steps:

1. Select the group from the menu next to the **Edit group** button on the main page, then click the button to bring up the group's editing form.
2. To change the group's name, edit the **Group name** field.
3. The **Members on server** field cannot be edited but shows who belongs to this group on the system shown in the form's title. Membership may be different on other systems if you have created users outside of this module.
4. To leave the parent group alone, select **Leave unchanged** for the **Parent group** field. Otherwise, select **Set to** and choose a group from the menu next to it, or <None> if you don't want it to have any parent. This menu includes all groups from all systems, and so

it is possible to choose one that does not exist on some managed hosts. If so, it will be as though <None> was selected.

5. As when editing a user, the options and lists in the **Modules for members** field can be used to add, remove, or set the modules for this group. See Section 50.4 “Editing or Deleting a Webmin User” for more detail.
6. Hit the **Save** button at the bottom of the page to update the group on all servers on which it exists. A page listing all of the servers and the results of the update on each will be displayed.

Deleting a group is just like deleting a user. Instead of using the **Save** button on the group’s editing form, click on **Delete**. The module will not let you delete a group that has any member users or groups on any servers.

50.7 Editing the User or Group ACL for a Module

As Chapter 52 explains, Webmin users and groups can be further restricted in what they can do with a particular module. This allows you to create a user who can edit only a single Apache virtual host or DNS domain, for example, but not use the rest of the features of the Apache Web server or BIND DNS Server module. The actual access control options available are different depending on the module that you want to restrict, and are covered in detail in the chapter for that module.

The Cluster Webmin Servers module can also be used to configure access control for a particular user and module, but on multiple hosts at once instead of just one. Before doing this, you should be familiar with the process of restricting access on a single system with the Webmin Users module, as a very similar form is used.

For module access control to work across multiple systems, each must have a very similar or identical configuration for the server that the restricted module manages. For example, it makes no sense to give someone access to a particular BIND zone if it does not exist on all servers. Unfortunately, some modules (such as Custom Commands) use command IDs that are unique to a particular server, so trying to give a user access to a particular command on multiple systems will not work even if that command button has been created independently on each system.

To edit access control settings for a user or group in a particular module, follow these steps:

1. On this module’s main page, select the user and module from the menus next to the **Edit ACL for** button. The top button is for users, the bottom for groups. When you hit the button, an access control form that differs depending on the module chosen will be displayed.
2. Follow the instructions in the appropriate chapter of the book to fill in the form. Many forms include lists of configuration objects (such as virtual servers, DNS domains, or Samba shares) to select, which will always be taken from the master server even if the user or module does not exist. This can cause problems if, for example, a DNS zone exists only on another host and it is not appearing in the menu of zones to which to allow access because the list is being taken from the master. Unfortunately, there is no way to avoid this at present.
3. To update the configuration for this module and user on all managed systems, click on the **Save on all hosts** button. You can also change the settings just for the host shown in

the title with the **Save only on this server** button. Either way, the change will be immediately applied to the user or members of the group.

Sometimes it is necessary to edit the access control settings on just a single system instead of all of them. You can do this by following these steps:

1. Open the user's or group's editing page using the **Edit user** or **Edit group** button on the main page.
2. At the bottom of the form is a button labeled **Edit ACL for** with a menu next to it listing all of the modules to which this user has access and hosts on which he exists. Select the entry for the combination of module and host for whom you want to edit access control settings and hit the button.
3. Fill in the access control form that appears as you usually would. Unfortunately, any lists of Apache virtual servers, custom commands, or DNS zones on the form will be taken from the master system, not the chosen host.
4. Hit the **Save only on this server** button to update just the settings on the chosen system.

50.8 Installing a Module or Theme

Probably this module's most useful feature is its ability to install a Webmin module or theme on multiple systems at once. Before you read on, make sure you have read the sections in Chapter 51 that explain what themes and modules are, and how they can be installed on a single system. The process of installing on multiple hosts is very similar, and the form used is almost identical.

The following steps can be used to install a single .wbm or .wbt file containing one or more modules and themes. Unlike in the Webmin Configuration module, there are no separate pages for each.

1. On the module's main page, find the right-hand form in the **Modules and Themes** section.
2. If the file exists on the master server, select **From local file** and enter its full path into the adjacent text field.
3. If the file is on the PC on which your web browser runs, select **From uploaded file** and use the **Browse** button to open a file dialog that allows you to choose the file. If this file is shared via NFS with some or all managed servers at the same location, the module will not bother to transfer it to each host.
4. If the file is on a web or FTP site somewhere, select **From ftp or http URL** and enter the complete URL into its text box. Normally, only the master server will download the file and then use RPC to transfer it to each managed system, but if **Each server should redownload module** is selected, then the managed hosts will redownload it themselves. This may be faster if the URL refers to a web server on your local network.
5. Normally, Webmin will stop a module from being installed if any other modules on which it depends are not available, or if it is written for a later Webmin release. To prevent this, check the **Ignore module dependencies when installing** box. This may, however, allow the installation of a module that will not work. It will not allow you to add modules that do not support the server's operating system.

6. To control to whom access to this new module will be granted, select the **Grant access only to users and groups** option and enter a list of Webmin user and group names into the adjacent text box. You can also give it to every user on all systems by choosing **Grant access to all Webmin users**.
7. Click on **Install Now** to go ahead with the installation. A page showing the progress of the module's download will be displayed (if necessary), followed by a list of error or success messages from each managed host. The installation will be done concurrently on all systems to speed up the process. The failure of one will not affect any of the others.

50.9 Viewing and Deleting a Module or Theme

You can bring up a page showing the details of an installed module or theme by selecting it from the menu next to the **Edit module** or **Edit theme** button on the main page and then clicking the button. The page that appears shows the module's name and directory, supported operating systems, and the modules that this one makes use of and is made use of by. The **Edit ACL for** button can be used to change the access control settings for the module for a particular user and system, as explained in Section 50.7 "Editing the User or Group ACL for a Module".

To actually remove the module or theme, follow these steps:

1. Open the page showing its details, as explained in Step 1 of Section 50.8 "Installing a Module or Theme".
2. The menu next to the **Uninstall module from** button determines which managed hosts it will be removed from. You can either select **All servers** to delete from every host or a specific system.
3. Click the button to display a confirmation page showing the module or theme to delete and the size of the files that will be removed. If some other module on a particular system depends upon this one, however, an error message will be shown explaining why you cannot remove it.
4. To have access to the module taken away from all users and all access control settings returned to their defaults, check the **Remove from users and reset access control settings?** box. This can be useful if you plan to reinstall the module in the future and don't want it to be available to the same people that could use it before.
5. Hit the **Delete** button to go ahead with the module or theme's removal. As with installation, the process will be done concurrently on all hosts and a page showing the results from each will be displayed.

50.10 Refreshing User and Module Lists

If modules, themes, users, or groups have been changed in any way on managed hosts without using this module, its cached information about the configuration of other systems will no longer be correct. This will not cause any serious problems, as the module can detect synchronization problems when updating or removing a user. For example, if the module thinks that a user exists on some managed host when he really does not, it will simply fail to update the user on that particular host.

It is best to refresh the cached lists when necessary, which you can do by following these steps:

1. On the module's main page, click on the **Refresh servers** button.
2. A page listing each of the managed servers and showing the changes in the user, group, module, and theme lists for each will be displayed. If, for some reason, a system cannot be contacted, an error message explaining what went wrong will be shown next to that system's name.

50.11 Listing and Removing a Server

This section explains how to view information about a managed server and the users, groups, modules, and themes on it, or remove it from the list of systems controlled by the module. To view information about a managed server, follow these steps:

1. Click on the icon for the server on the main page or on a module or user details form.
2. On the page that appears, the details of the server itself are shown at the top, followed by lists of modules, and themes and then lists of users and groups.
3. The entries in all of these lists can be clicked on to either bring up a module, a theme details page, a user, or a group editing form. In all cases, the information about the chosen object is taken from this server.
4. To remove the host from this module's control, click on the **Remove from managed list** button. The deletion will happen without confirmation, and you will be returned to the module's main page.

50.12 Configuring the Cluster Webmin Configuration Module

Like the other cluster modules, this one has only a single configurable setting (see Table 50.1) that can be changed by clicking on the **Module Config** link on its main page.

Table 50.1 Module Configuration Option

Sort hosts by	Controls the order in which host icons are listed on the main page and package editing form. The options are: Hostname Systems are sorted by their hostname. Order added Systems are listed in the order in which they were added to this module. Description Systems are sorted by description, which will be the same as the hostname if no description is set.
----------------------	---

50.13 Summary

This chapter has explained how to manage the settings of multiple Webmin systems from a single administration interface. It has covered the creation and management of both users and groups on several servers simultaneously and explained how detailed access control settings can be set for those users and groups. It has also explained how Webmin modules and themes can be installed on or removed from many servers at the same time.

Webmin Configuration

This chapter explains how Webmin can be used to configure itself, install new modules, or upgrade to a new version.

51.1 The Webmin Configuration Module

This module exists to allow Webmin itself to be configured, unlike most other modules that are designed to configure some other server or service. It lets you do things like change the port that Webmin uses, limit the client addresses that can connect, change the theme and language that the user interface uses, and install new modules. This chapter explains how to use the module to carry out all these tasks.

When you click on the module's icon in the Webmin category, the menu of icons shown in Figure 51.1 will be displayed. Each of the icons can be clicked on to display a configuration page or form, on which some of the Webmin settings can be changed.

51.2 Restricting Access to Webmin

Webmin will accept connections from any IP address by default. Even though it is password-protected, you should limit access to only legitimate client systems, if possible, so that an attacker from outside your network cannot even attempt to log in. To do this, follow these steps:

1. Click on **IP Access Control** on the module's main page to bring up the access control form.
2. Select **Only allow from listed addresses** and enter a list of hostnames, IP addresses, and networks into the adjacent text box. Networks should be entered with a netmask like `192.168.1.0/255.255.255.0`. You can allow access from an entire DNS domain by entering something like `*.example.com`, but be aware that that is not totally secure because an attacker can fake reverse DNS results.

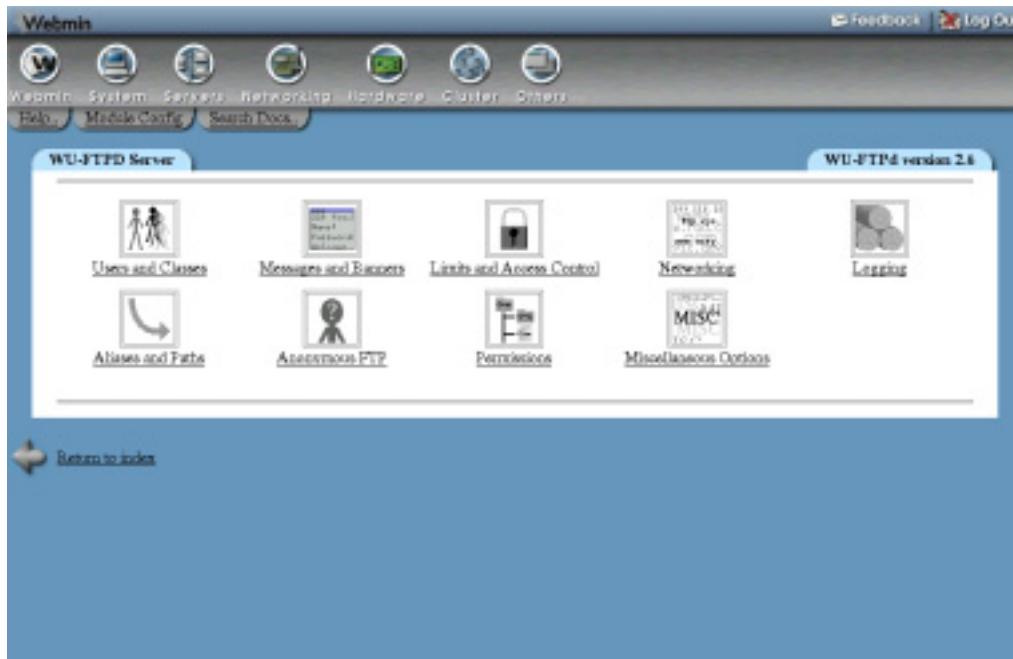


Figure 51.1 The Webmin Configuration module.

3. Normally Webmin will resolve any hostnames that you enter only once when it first starts up. To change this, check the **Resolve hostnames on every request** box and it will convert hostnames to IP addresses for comparison for every request. This can be useful if the system you are running a browser on is frequently changing IP addresses but is able to update a DNS record to match. This can happen on a network using DHCP or if you are connected to an ISP that dynamically assigns addresses.
4. To have Webmin check the TCP-wrappers configuration files `/etc/hosts.allow` and `/etc/hosts.deny` as well when deciding whether to allow a client, turn on the **Also check TCP-wrappers hosts.allow and hosts.deny files** option. The service name to use when editing those files is `webmin`.
5. Hit the **Save** button to activate the new client address restrictions.

51.3 Changing the Port and Address

Webmin usually listens for connections on port 10000 on all of your system's IP addresses. You may need to change the port, however, perhaps because a firewall on your network only allows connections to web servers on the standard ports of 80 and 443. Because servers run by any user can use port 10000, it may be possible for a malicious user on your system to wait for Webmin to be shut down and then start his own fake Webmin server on that port, which could capture the `admin` or `root` password. For this reason, you may want to use a port below 1024 (that only programs run as `root` can listen on) instead. Changing the listening IP address can also be used

ful if your system has multiple network interfaces and you want to only allow connections on the interface connected to the internal LAN.

To change the port or address, use the following steps:

1. Click on the **Port and Address** icon on the module's main page.
2. To listen on only a specific interface address, select the second option in the **Listen on IP address** field and enter an IP into the text box next to it. This must be the address of one of your host's real or virtual interfaces.
3. To change the port, enter a number into the **Listen on port** field.
4. Hit the **Save** button to use the new settings. Your browser will be redirected to the new port and address, and you may need to log in again.

51.4 Setting Up Logging

Like most web servers, Webmin can be configured to create a log file in the standard CLF format that records every request it receives. It also creates a log of actions performed by users, such as the creation of a DNS zone or the deletion of a UNIX group. This action log can even include the details of every file changed and every command run by each action so you can see what Webmin is doing under the hood.

Basic logging is enabled by default, but you can configure it further by following these steps:

1. Click on the **Logging** icon on the main page.
2. If **Disable logging** is selected, then Webmin will write no logs at all. You should, however, choose **Enable logging** to activate it.
3. If the **Log resolved hostnames** box is checked, the log file will contain actual client hostnames instead of IP addresses. This can cause problems if reverse DNS lookups take a long time on your network, as one will need to be done for each request.
4. To prevent the log files from becoming too large, Webmin can be configured to truncate them periodically. To enable this feature, select the **Clear logfiles every** box and enter the number of hours into the adjacent text field.
5. To limit action logging to only specific users, select the **Only log actions by** option and choose some users from the list next to it. This can be handy if most of your users can only perform tasks that you don't care much about and you want to log only actions taken by the more powerful administrators.
6. To limit action logging to only specific modules, select the **Only log actions in** option and choose one or more modules from its list.
7. To enable the logging of file changes and commands run for each action, check the **Log changes made to files by each action** box. This will take up more disk space, but provides some very useful and interesting information.
8. Hit the **Save** button to activate the changes.

The Webmin Actions Log module (covered in Chapter 54) explains how to search for and view actions once you have enabled their recording here. This can be useful for finding out who did what on your system if you have multiple administrators with access to the server.

51.5 Using Proxy Servers

Many Webmin modules are capable of downloading files from other FTP, HTTP, and HTTPS servers. For example, the Software Packages module lets you enter a URL from which you can fetch and install a new package. Webmin will normally connect directly to the host specified in the URL, but it can be configured to use a proxy server instead. This may be necessary if your network does not allow direct access to web and FTP sites, but instead forces clients to connect through a proxy.

Webmin's RPC mechanism (covered in Chapter 53) also makes use of HTTP requests to other Webmin servers. Any proxy configuration will also apply to RPC calls, although it will not direct TCP connections used by the RPC protocol when in fast mode or when transferring large files. Because any other Webmin servers are likely to be on the same network, you will probably want to disable the user of a proxy for those hosts.

To specify HTTP and FTP proxy servers and the hosts for which they will be used, follow these steps:

1. On the Webmin Configuration module's main page, click on the **Proxy Servers** icon.
2. If you want a proxy to be used for HTTP requests, select the second radio button in the **HTTP proxy** field and enter a full URL like *http://proxy.example.com:8080/* into the text box next to it. If **None** is chosen, no proxy will be used. This specified server will also be used for HTTPS connections by making CONNECT proxy requests, so make sure that it supports and allows them.
3. You can also enter a proxy to use for FTP downloads in the **FTP proxy** field. Usually this will be the same as the HTTP proxy.
4. To disable the use of a proxy for certain hosts, fill in the **No proxy for** field with a space-separated list of hostnames, domain names, and full or partial IP addresses. For example, you might enter *.example.com 192.168.1.* to have Webmin connect directly to hosts in that domain and network.
5. If your proxy requires clients to authenticate themselves, fill in the **Username for proxy** and **Password for proxy** fields.
6. Hit the **Save** button to have Webmin start using the new settings.

51.6 Configuring the Webmin User Interface

Webmin has several settings that control the color scheme of the user interface (when using the **Old Webmin Theme**), what server host information is displayed on each page, and if the sending of feedback is allowed. You can change them by following these steps:

1. On the module's main page, click on the **User Interface** icon to bring up the interface options form.
2. The first five fields let you choose the colors to be used for various parts of the interface when using the old-style theme. You can either select **Default** for each, or enter three hexadecimal numbers for the red, green, and blue components of a color. For example, *FF 88 00* would be a shade of orange. These options have no effect on the new default theme, however.

3. In some themes, the title at the top of every page is rendered as an image. Because this can make the page slow to download, you can force the use of plain HTML text titles instead by changing the **Display titles as text?** field to **Yes**.
4. Every page in Webmin shows your system's hostname and operating system in the browser status bar by default. To have it shown elsewhere or turn it off altogether, select one of the following options from the **Display login and hostname** menu:
 - At bottom of browser** The information is shown in the browser's status area, which is usually at the bottom of the window.
 - In browser title** The information is added to the title of each page, which usually appears in the browser window title.
 - Above page heading** The information is shown above the title of each page.
 - Nowhere** The hostname and operating system information are not shown anywhere. If you are worried about untrusted users learning too much about your system, this is the option to choose.
5. When using the default theme, every page in Webmin has a link in the top-right corner for sending feedback to the developer. You may want to configure it so that feedback is sent to the master system administrator instead, so that other users can contact you. To do this, enter your email address in the **Send feedback to** field and change the **Allow sending of feedback?** field to **Only to address above**. The sending of feedback can be completely prevented by selecting **No** in the latter field instead.
6. Click on the **Save** button to activate the new user interface settings.

51.7 Installing and Deleting Webmin Modules

As the first chapter of this book explains, Webmin is essentially a collection of modules, each of which performs some task such as configuring Apache or managing UNIX users. A module can be added or removed without effecting the operation of others, assuming that they do not depend upon it. Even though the main Webmin distribution includes 76 modules at the time I wrote this book, over 100 more written by other people are available for download from the website webmin.thirdpartymodules.com. This is a searchable database of modules and themes that perform tasks the core modules do not, such as managing the FreeBSD firewall, displaying system information, and connecting to a VNC server.

Once you have found a module that you like, it can be installed by following these steps:

1. On the main page, click on the **Webmin Modules** icon. This will bring to you to a page with forms for installing, cloning, and deleting modules.
2. If you have already downloaded the module's .wbm file to the system on which Webmin is running, select **From local file** and enter the full path to the file into the text field next to it.
3. If the module file is on the PC on which your web browser is running, select **From upload file** and use the **Browse** button to find the file on your computer.
4. If the module is on a website somewhere instead, select **From ftp or http URL** and enter the full URL into the text box next to this option.
5. Webmin will normally stop a module from being installed if any other modules that it depends on are not available or if it is written for a later Webmin release. To prevent this,

check the **Ignore module dependencies when installing** box. This may, however, allow the installation of a module that will not work. It will not allow you to add modules that do not support the server's operating system.

6. To control to whom this new module will be granted, select the **Grant access only to users and groups** option and enter a list of Webmin user and group names in the adjacent text box. By default, only the user that you are currently logged in as is listed. You can also give it to every user and group by choosing **Grant access to all Webmin users**.
7. Hit the **Install module from file** button to download (if necessary) and install the new module. If everything goes OK, a page listing the installed modules and the sizes of their directories will be displayed.

Webmin comes with a script called `install-module.pl` that can be found in the installation root directory. If you have installed the RPM version, this will be `/usr/libexec/webmin`. Otherwise it will be wherever the `.tar.gz` file was extracted. This script can be used to install a module from the command line, by passing the `.wbt` file to it as a parameter. It will be granted only to the `root` or `admin` user, if one exists, or the first account listed in the Webmin Users module otherwise.

Any of the modules currently installed, including those that come with Webmin by default, can be deleted on the same page as well. Deleting the default modules is not a good idea, however, as they will be automatically reinstalled the next time you upgrade. Instead, it is better to take away access that you do not want to use with the Webmin Users module, as explained in Chapter 52. Not all modules can be deleted, as some are depended upon by other modules (such as Running Processes).

To remove one or more modules, follow these steps:

1. Click on the **Webmin Modules** icon on the main page.
2. Scroll down to the last form on the page and select all the modules that you want to remove from the **Delete Modules** list. Clones (explained later) can be deleted as well if they are no longer needed. Deleting a module that has clones will automatically remove them as well.
3. When you hit the **Delete selected modules** button, a confirmation page will be displayed showing exactly what will be removed. Or, if there are some dependency problems that prevent one or more from being deleted, an error message explaining the problem will be shown instead.
4. To have access to the module taken away from all users and all access control settings returned to their defaults, check the **Remove from users and reset access control settings?** box. This can be useful if you plan to reinstall the module in the future and do not want it to be available to the same people who used it before.
5. Click on **Delete** to go ahead with the module's removal. A page showing exactly which modules were deleted will be displayed, along with the number of bytes deleted for each.

51.8 Cloning a Webmin Module

In some situations, you may wish that you had the same Webmin module installed twice so each could be configured individually. This can be useful if, for example, you have two versions of Apache installed—perhaps one for testing and one for production. The standard Apache module

can only be set up to manage one at a time, so it might appear that the only way to configure both servers would be to install Webmin twice.

There is a solution, however—module cloning. A clone is a copy of an existing module that shares all of the same code but can be configured separately, assigned to different users, and have its user access control set up independently. To create a clone, follow these steps:

1. Click on the **Webmin Modules** icon on the main page.
2. Go to the second form, entitled **Clone Module** and select the original module from the **Module to clone** menu.
3. In the **Cloned module name** field, enter a new name to be displayed under the clone's icon, such as *Testing Apache Server*.
4. If you want this module to appear in a different category to the original, select it from the **Assign to category** menu.
5. Hit the **Clone Module** button. The copy will be created and granted to the user as whom you are currently logged in, and your browser will be returned to the Webmin Configuration main page.
6. You can now go to the new module, which, by default, will be configured identically to the original. The **Module Config** link can then be used to set it up to use different configuration files and program paths if necessary.

If you clone a module like Custom Commands or System and Server Status, any existing commands or monitors will be copied to the clone as well. You can delete them, if you wish, without affecting the settings in the original module.

There are quite a few clever tricks that can be performed with cloning, such as making a new copy of the System and Server Status module that runs on a different schedule, making a copy of the Users and Groups module for managing NIS users, or having multiples of the Fetchmail module for different configuration files.

51.9 Changing Your Operating System

Webmin behaves differently depending on the operating system or Linux distribution that you have installed, as well as the particular version that you are running. The correct OS is always automatically detected at installation time or provided by the installer, but it is quite possible that your system may be upgraded during the lifetime of the system. If this happens, Webmin will not automatically detect the upgrade. You must tell it by following these steps:

1. Click on the **Operating System and Environment** icon on the module's main page.
2. Select your UNIX vendor and version from the **New operating system** list.
3. Hit the **Save** button to have Webmin start using it.

The operating system and version detected at installation time determines the default values for module configurations, as each flavor of UNIX uses different locations and formats for the various config files that Webmin manages. Changing your OS by following the steps above, however, will not adjust any of these configuration settings. Instead, it will just determine which ones are used for modules installed in the future. Usually this is not a problem, as most OS upgrades will not change the locations of files and programs. Some modules, however, may need to be manually configured after an upgrade. For example, you may need to change the print sys-

tem used by the Printer Administration module if the old OS version used LPRng and the new version uses CUPS.

51.10 Editing the Program Path and Environment Variables

When you run a command like `ls` from the UNIX shell, the `PATH` environment variable determines the directories that your shell will search to find the actual executable, such as `/bin/ls`. Webmin also uses the `PATH` variable to locate commands that it runs when a full path is not specified, such as `webalizer` or `mysql`. By default, this list of directories is set to include all of the common locations for programs on your operating system, but may be incorrect if you have installed executables in some nonstandard directory such as `/usr/local/samba/bin`.

The `LD_LIBRARY_PATH` environment variable also determines where programs look for shared libraries that they need to load when run. Again, Webmin sets this variable by default to include all of the common library directories on your operating system, but it may miss some if you have compiled and installed programs manually. A symptom of this is programs run by Webmin failing with an error message like `libmysqlclient.so.6: open failed`. A library like this might be found in `/usr/local/mysql/lib`, which is not in the default search path.

You can edit these paths and define your own environment variables that will be passed to all programs run by Webmin by following these steps:

1. Click on the **Operating System and Environment** icon.
2. Add any additional program directories to the **Program search path** field. Each directory must be separated by a `:` (colon), just as they are in the `PATH` environment variable. Existing directories should not be removed or changed, however, as they may stop parts of Webmin from working.
3. Add any extra shared library directories to the **Library search path** field, again separated by colons.
4. Sometimes it is useful to have Webmin pass other environment variables to programs that it runs. For example, if you had several custom commands that connected to Oracle, you might want `ORACLE_HOME` to be set appropriately before they are run. The **Additional environment variables** table allows you to define some—just enter a name into the first empty field under **Variable name** and a value into the field next to it under **Value**. As with most tables in Webmin, this one only displays one empty row at a time, so if you want to add more than one variable, you will need to save and reopen this page.
5. When you are done setting paths and variables, hit the **Save** button to activate them.

Any program run by Webmin also has access to several variables set by the web server itself and passed to the CGI programs that make up Webmin. For example, `REMOTE_USER` contains the name of the logged-in user and `REMOTE_HOST` contains the client IP address. All HTTP headers are stored in uppercase variables starting with `HTTP_`. A program can find information about the user's browser in the `HTTP_USER_AGENT` variable, for example.

51.11 Changing Webmin's Language

Many Webmin modules have been translated into different languages, such as German and Japanese. You can change the default language for all users by following the steps below, or for just a

single user by using the Webmin Users module. Not all of the translations are complete, however, so some messages and labels will still appear in English.

1. Click on the **Language** icon on the module's main page.
2. In the form that appears, select your users' preferred language from the **Display in language** menu.
3. Some browsers (such as Opera) can request that the server display pages in a language chosen by the user. To have Webmin honor such requests, if possible, change the **Use language specified by browser?** field to **Yes**. If a language is sent, it will override both the global and individual users' settings.
4. Hit the **Save** button to have Webmin switch to the new language immediately.

Many languages (such as Chinese, Japanese, and Russian) use symbols not found in the standard European alphabet. To display them, a special font often needs to be installed on the system running the browser that you use to access Webmin. Some Linux distributions include these fonts by default, but others do not and installing them can be rather complex.

51.12 Editing Main Menu Settings

As well as general user interface settings that apply to all pages, there are some that control the layout of only the main menu on which module icons are displayed. They can be used to turn categorization off, control the display of your system's hostname and OS, and have users sent directly to a module by default, among other things. These steps explain how to change the main menu settings.

1. Click on the **Index Page Options** icon on the Webmin Configuration module's main page.
2. By default, modules icons are listed four to a row. If you prefer to use a wide browser window, this may be too few to make good use of the available space. Edit the **Number of columns** field to change the number of icons in each row.
3. When the **Categorize modules?** option is set to **Yes**, icons are displayed under categories to reduce the number that appear on any one page in the main menu. When using the default theme, selecting **No** instead will put them all on one big page and remove the list of categories from the top of all pages.
4. When a user logs in to Webmin, he will see the modules by default in the **Webmin** category. If you usually use modules in some other category, select it from the **Default category** menu.
5. When the **Show version, hostname and OS?** field is set to **Yes**, as it is by default, the main menu displays your system's Webmin version, hostname, and operating system. If you don't want this information to be made available to users for security reasons, select **No** instead.
6. If a Webmin user has access to only one module, it makes no sense for him to see the main menu at all as it will contain only one icon. To have such users directed immediately to their only module after logging in, change the **Go direct to module if user only has one?** field to **Yes**.
7. Click on the **Save** button to activate these new main menu settings.

Some nonstandard themes may not implement all of these features, especially those that have their own main menus.

51.13 Upgrading Webmin

Webmin has the ability to upgrade itself when a new version comes out—either from a file that you have already downloaded or from a package that it fetches from www.webmin.com for you. Even though it is quite possible to upgrade from the command line by installing the latest RPM or `tar.gz` package, doing it from within this module is even easier and less prone to error.

Webmin can only be upgraded using the same type of package from which it was originally installed. This means that if you used the `tar.gz` format originally, an upgrade can only be done from another `tar.gz` file. An RPM install can also only be upgraded from a newer RPM package. Of course, when Webmin downloads the newest version for you, it will always choose the right package format.

Recent releases have the ability to check the GnuPG digital signature on the RPM and `tar.gz` packages to ensure that they are authentic. This can only be done if you have the `gpg` command installed on your system, and when using the `tar.gz` package but only when upgrading directly from the Webmin site. Signature checking protects you from installing a fake version of Webmin that is actually a Trojan horse or some other type of malicious program.

To upgrade Webmin, follow these steps:

1. Click on the **Upgrade Webmin** icon on the module's main page. This will take you to a page with forms for upgrading, installing updated modules, and setting up the automatic install of updates.
2. The **Upgrade Webmin** form is very similar to the form for installing modules, as explained in Section 51.7 “Installing and Deleting Webmin Modules”. Select either **From local file** if the new package is already on your server system, **From uploaded file** if it is on the PC on which your web browser is running, or **From ftp or http URL** to have the package downloaded from a URL. The easiest option is to choose **Latest version from www.webmin.com** to have the appropriate package downloaded automatically.
3. If the Webmin version on your system was installed from the `tar.gz` file, the **Delete old version's directory after upgrade?** box can be checked to have the old version removed after the new one is installed. Unless you want to be able to revert to the old release, this option should be enabled to save on disk space. It does not appear at all for RPM installs, as the RPM package always installs in the same directory.
4. To have the GnuPG signature on the package verified, if possible, turn on the **Check GnuPG signature on package?** option. It is enabled by default if the `gpg` program is installed on your system.
5. Hit the **Upgrade Webmin** button to begin the upgrade. A page showing the download progress (if necessary) and output from the new version's `setup.sh` script will be displayed. If you are already running a version later than the one selected to install—or on www.webmin.com—an error message will be displayed instead.

The upgrade process will preserve all users and module configuration settings and should not even be noticeable by people currently accessing your Webmin server. If you originally installed the program from the `tar.gz` package, the new version will be installed in the directory *next to*

the old one. For example, if Webmin 1.090 was in `/usr/local/webmin-1.090` and you upgraded to Version 1.100, it would be installed in `/usr/local/webmin-1.100` and the old directory would be deleted if the **Delete old version's directory after upgrade?** option was checked.

Any modules that the new version includes but the old one does not will be granted to the first user listed in the Webmin Users module, which will typically be `root` or `admin`. You should check after the upgrade is complete to ensure that they have not been given to an untrusted user instead, as most modules can be used to subvert security on your system by default.

51.14 Installing Updates to Webmin

Updated versions of Webmin modules in the latest release are often made available to fix bugs or security problems. Installing these updated modules is always a good idea, as they may fix problems that you have been having or patch security holes that could allow untrusted users to gain `root` access on your system. Updates are always designed to solve problems rather than add new features, which may potentially have problems of their own.

Of course, if you are not having any trouble, you can just wait until the next full release and install it instead. Each version will always include any updates that were made available for previous versions of Webmin. Updates are only created to solve problems in the latest version, so if you are running an older version, do not expect any more to be released for it.

The www.webmin.com/updates.html page lists the downloadable updates for each version of Webmin. You can retrieve any that you need from there to be installed using this module, as explained in Section 51.7 “Installing and Deleting Webmin Modules”. There is an easier method, however. Webmin can be told to check for, download, and install any updates that it does not already have. This can either be done explicitly using this module or set up to happen on schedule.

To check for and install updates, follow these steps:

1. Click on the **Update Webmin** icon on the Webmin Configuration module’s main page.
2. Scroll down to the second form, entitled **Upgrade modules now**.
3. Select the **Update from www.webmin.com** option. The alternate **Update from another source** mode is only useful if running your own repository of new modules, which is not covered in this book.
4. If you just want to see what updates are needed without actually installing them, check the **Only show which modules would be updated** box. Otherwise, uncheck it so that updates are actually done.
5. If you have deleted some of the standard Webmin modules and don’t want them to be reinstalled by the update process, deselect the **Install modules that are not currently installed** option.
6. Hit the **Update Modules** button. A page listing all updates for your operating system will be displayed, along with the problems that they fix. As long as the box in Step 4 was not checked, the progress of each needed module’s download and the results of its installation will be shown as well.

If a new version of Webmin is available, a message will appear at the end of the page informing you of that. Because module updates are only released for the latest version, it is advisable to upgrade the entire program as soon as possible.

Every Webmin module has a version number that the update process uses to keep track of which ones it has already downloaded and installed. A message like **Module cron is already up to date** shown next to a potential update indicates that it has already been installed. A message like **Update to module cron is not related to this OS** means that the module does not support your operating system, or that the problem the update fixes does not occur on your OS.

Instead of manually following the preceding steps every now and then, you can configure Webmin to check for, report on, and install new modules on a schedule. When needed updates are found, an email can be sent to you listing the modules that should be or have been installed, and the problems that they fix. The email will also include notification of the availability of a new Webmin release, if there is one.

To enable automatic updates, follow these steps:

1. Click on the **Update Webmin** icon and scroll down to the final form on the page entitled **Update modules on schedule**.
2. Check the **Scheduled updating currently enabled** box.
3. Unless you run your own repository, select **Update from www.webmin.com**.
4. The **Update modules at** field specifies the time of day that the scheduled update check is run. Typically you should enter something like 3 to have updates done at 3 a.m., assuming that your system is turned on at that time.
The **every** field next to it sets the number of days between checks. For example, if you enter 3 then updating will be done only every third day. 1 or 2 days is usually a reasonable period.
5. If the **Only show which modules would be updated** option is enabled, a report only showing those modules that need updating will be sent out on schedule. This can be useful if you want to be reminded of new modules, but want to install them yourself to control which updates are used.
6. The **Install modules that are not currently installed** option has exactly the same meaning as in the **Update modules now** form and generally does not need to be enabled.
7. If **Only report updates** is checked, an email report will not be sent if no needed updates are available and no new version of Webmin has been released. This is usually what you want, as it cuts down on the number of unnecessary email messages.
8. In the **Email update report to** field, enter the address to which the update report should be sent. If it is left empty, automatic checking will still be done, just not reported. Email is always sent by calling the `sendmail` program, the path to which is taken from the Sendmail Configuration module's configuration.
9. Hit the **Save and Apply** button to enable scheduled updating. A Cron job (covered in Chapter 10) will be created that you can see in the Schedule Cron Jobs module, but should not touch.

Automatic updating can be turned off at any time by deselecting the **Scheduled updating currently enabled** box on this form and clicking on **Save and Apply**.

51.15 Configuring Authentication

Webmin has several options that control how multiple failed login attempts are handled, how users log in, and how UNIX passwords are checked. The default authentication method uses cookies, but if your browser cannot handle them you may want to switch to basic HTTP authentication, instead. The only problem with this method is that there is no way to properly log out because there is no support for logging out in the HTTP protocol. It must sometimes be used, however. For example, browsers on MacOS X cannot load applets (such as the ones in the File Manager and SSH/Telnet Login modules) from web servers using cookie authentication.

To configure authentication for Webmin, follow these steps:

1. Click on the **Authentication** icon on the module's main page to bring up the authentication form.
2. When **Enable password timeouts** is selected, Webmin will detect multiple failed login attempts from the same IP address and lock that host out for a configurable amount of time. This feature should always be turned on, as it stops attackers using millions of login attempts to guess passwords on your system. The **Block hosts with more than** field specifies the number of login attempts allowed from a single host before blocking is triggered, while the **failed logins for** field sets the number of seconds for which a host is blocked. The defaults are reasonable, but you can increase the timeout if you are feeling paranoid.
3. When **Log blocked hosts, logins and authentication failures to syslog** is selected, Webmin will send messages to the system logs (covered in Chapter 13) when a user logs in, logs out, or enters an incorrect password. All messages are sent with the authpriv facility. You should leave this option turned on so suspiciously large numbers of login failures can be detected.
4. When **Enable session authentication** is selected, Webmin will use its own login form to ask users for a username and password, and set a cookie after the login is complete to identify authenticated clients. To switch to normal HTTP authentication, select **Disable session authentication** instead.
5. When using session authentication, Webmin can be configured to automatically log users out if they have been inactive for longer than a certain period of time. To enable this, check the **Auto-logout after** box and enter a number of minutes into the text field next to it. This feature and the next three are not available when using HTTP authentication.
6. When **Offer to remember login permanently?** is checked (as it is by default), the login form will include a check box for permanently remembering the login. When selected, the cookie sent to the user's browser will be marked to indicate that it should be saved even if the browser is shut down and rerun later. This is convenient because it means that the user will not have to log in to Webmin again, but you may consider it a security risk. If so, unchecking this box will remove the **remember** option from the login form.
7. The login page includes the hostname from the URL in the message above the username and password fields by default. To hide it, deselect the **Show hostname on login screen?** box.
8. Some people like to have a welcome message shown on the login page the first time a user accesses it, perhaps giving information about the server or telling unauthorized people to go away. To enable this on your system, first create an HTML page containing the message that you want to appear. Then, select **Show pre-login file** and enter the full path

to the HTML file in the text field. After a user reads it, he must reload or revisit the page (perhaps by following a link in the page itself) to force the real login form to appear.

9. Webmin can automatically authenticate connections from `localhost` by determining which UNIX user is making the connection, and checking to see if a Webmin user of the same name exists. To enable this, select **Allow login without password for matching users from localhost**. If you run a browser as `root` on the same system on which Webmin runs and have a Webmin user called `root`, this feature allows you to access `http://localhost:10000/` and be logged in without needing to enter a username and password. It is convenient, but potentially insecure if an attacker can trick a program (such as Squid) into connecting to that URL, which would grant access to Webmin as the user as whom the program runs. For this reason, **Always require username and password** is selected by default.
10. When the **UNIX authentication** option is selected for a user in the Webmin users module, his password can be checked by using PAM or by reading the UNIX password file directly. If the **Use PAM for UNIX authentication, if available** option is selected and the `Authen::PAM` Perl module is installed, Webmin will attempt to use PAM to validate the user. On Linux, however, this will only work if the `/etc/pam.d/webmin` service file is set up correctly. This file is included in the RPM package of Webmin.
If your operating system does not support PAM, if the Perl module is not installed, or if the **Never use PAM for UNIX authentication** option is selected, Webmin will fall back to directly reading the password file. This is more reliable, but will not prevent the use of passwords that are marked as expired. The **read users and passwords from file** fields specify the file from which to get passwords and the columns to use for the username and password, but should rarely need to be changed as they are set by default to match your operating system.
Because Webmin will use PAM where it can, or read the appropriate password file if PAM is not available, the fields covered in this step should not need to be changed.
11. The **External squid-style authentication program** field can be used to enter the full path and parameters to a program that validates passwords. If it is filled in, the **External authentication program** option will appear in the **Password** menu for a user in the Webmin Users module, indicating that the user's password should be checked using this command. The program must behave exactly like a Squid's external authenticator, covered in Section 44.9 "Setting Up Proxy Authentication".
12. Finally, hit **Save** at the bottom of the form to activate the new authentication settings for subsequent logins.

51.16 Editing Categories and Moving Modules

Every Webmin module has a category that controls where it appears on the module's main menu. You can create your own categories and move modules from their default locations into your own or existing categories, which can be useful if you don't like the default arrangement or want to put everything into one huge category.

To create new categories, or rename existing ones, follow these steps:

1. Click on **Edit Categories** on the module's main page to display the category editing page.
2. To add a category, scroll down to the bottom of form. In the first empty field under **ID**, enter a unique internal name for your new category, such as *stuff*. Then, in the field next to it under **Displayed description**, enter the name that will appear in Webmin, such as *Thirdparty*.
Existing categories that you have added can be edited by changing the fields in this section, as well. You should not, however, change the entries in the **ID** column, as they are used internally to associate modules with categories. The ID is never visible to users anyway—only the displayed description is.
3. To change the name of one of the default categories displayed at the top of the form, select the second radio button next to it and enter a new description into the text box to the right. If **Default** is chosen, the standard name determined by the current language will be used.
4. Hit the **Save Categories** button at the bottom of the form to activate the new categories. You can now move modules into any category that you have created.

To change the categories in which modules appear, do the following:

1. Click on the **Reassign Modules** icon on the main page.
2. The page that appears lists every installed Webmin module and the category in which it currently resides. For each module that you want to move, select a new category from the menu next to its name.
3. Click on the **Change Categories** button at the bottom of the page to move the modules.

51.17 Changing and Installing Themes

A theme is an extension to Webmin (much like a module) that controls how its interface appears. The currently active theme determines if and how the categories at the top of each page are displayed, what page background is used, what icons each module has, how titles appear, and how each page ends. By changing themes, you can significantly change the look of Webmin without affecting its functionality. Several themes are included by default and you can install more that have been written by other developers.

Like the language, you can set the theme for all logins in this module, or override it for a specific user in the Webmin Users module. To change the theme for everyone, follow these steps:

1. Click on the **Webmin Themes** icon on the module's main page. This will take you to a page for changing themes, installing a new theme, and deleting existing ones.
2. Select the theme to use from the **Current theme** menu. Those included as standard with Webmin are:

Old Webmin Theme The very simple theme that the first versions of Webmin used before theming was added. If you find the default too slow, this may be a better alternative as it uses fewer images.

Caldera Theme An improved layout that uses frames to place categories and module icons at the top, and actual forms and pages at the bottom.

MSC.Linux Theme The current default Webmin theme.

MSC.Linux Mini Theme A modified version of the default theme, designed for use on small-screen devices such as PDAs.

3. Hit the **Change** button to activate the chosen theme.

New themes developed by other people can also be added to Webmin, and several are available for download from webmin.thirdpartymodules.com. It is also possible to write your own themes, as Chapter 58 explains.

To install a theme, follow these steps:

1. Click on the **Webmin Themes** icon on the module's main page.
2. Select the theme's file using the second form. Just as when installing a module, you can choose to install a theme from a file on the system running Webmin, the PC your browser is on, or an HTTP or FTP URL.
3. Hit the **Install Theme** button to have it downloaded (if necessary) and installed.

The final thing that you can do on this page is delete one of the installed themes. The **Old Webmin Theme** cannot be deleted as it is built into the program, and the other standard themes should not be deleted as they will be added again if you upgrade to the next version.

To delete a theme that you have installed, follow these instructions:

1. Click on the **Webmin Themes** icon on the module's main page.
2. Select the one to remove from the **Theme to delete** menu at the bottom of the page. If that menu does not appear, it means that all installed themes are in use either by an individual user or for everyone.
3. Hit the **Delete** button to bring up a confirmation page asking if you really want to go ahead.
4. Click on **Delete** again to remove the theme.

51.18 Referrer Checking

One danger when using a web-based administration interface like Webmin is that a link from another website may point to a program on your Webmin server. For example, a malicious site could include HTML code like:

```
<a href="http://localhost:10000/proc/run.cgi?cmd=rm+*>click me</a>
```

Clicking on this harmless-looking link would cause Webmin's Running Processes module to run a command that deletes files on your system! Assuming that you have already logged into Webmin, no password would be required. Worse still, a similar URL could be used in an `` tag for an image, which is fetched automatically by your browser as soon as you open a page that seems innocuous.

Fortunately, there is a solution. Most browsers send the full URL of the page from which a link came in their HTTP requests. By default, Webmin compares the hostname in this URL with

the one used to access the current page and displays a warning if they do not match. This blocks links from other websites to your Webmin server, except for those that do not specify a program, such as `http://localhost:10000/cron/`, and are therefore harmless.

Sometimes, however, you will want to allow such links, such as from an internal Intranet web server that you maintain and trust. For this reason, Webmin can be configured to allow links where the referrer is from a list of trusted hosts. The following steps explain how.

1. Click on the **Trusted Referrer** icon on the module's main page.
2. To turn off referrer checking entirely (which is not a good idea), change the **Referrer checking enabled?** field to **No**.
3. To allow links from certain hosts, fill in the **Trusted websites** field with a list of hostnames, such `intranet.example.com`.
4. In some cases, the browser will not provide any referrer information at all, possibly because it does not support that HTTP feature. When the **Trust links from unknown referrers** box is checked, Webmin will allow requests in this case. If you are paranoid and know that your browser always does supply referrer information, turn this option off.
5. Hit the **Save** button to activate the settings.

Webmin does not simply deny links from untrusted sites. Instead, it displays a warning and gives the user a chance to continue with the link. This warning form contains a checkbox labeled **Don't show this warning in future**, which if selected effectively, changes the **Referrer checking enabled?** field to **No**.

51.19 Allowing Unauthenticated Access to Modules

It is possible to set up certain Webmin modules so that they can only be used to view information or to execute harmless commands. For example, the System and Server Status module's access control features can be set to give a user read-only access, letting them see which monitors are up and which are down. The Custom Commands module can also be configured for a particular user to let him only run commands that display information.

The anonymous access feature of this module lets you grant access to certain modules to clients without them needing to log in at all. Such clients will be treated as a specified Webmin user and thus will only have the rights that you grant to that user. They will, however, never need to supply its username and password when accessing allowed modules on your system. This can be useful for making certain information (such as the server status display) available to everyone on your network without needing to give them all a username and password. It should be used with extreme care, however, as granting anonymous access as a powerful user could compromise your entire system.

To set up unauthenticated access to some modules, follow these steps:

1. First, use the Webmin Users module to create a user called *anonymous*, for example, who has the modules and access control settings that you want to give to unauthenticated clients. Its password can be set to **No login allowed**, as this user will never log in conventionally. The user should be given the **Old Webmin Theme** to minimize the number of image directories to which you will need to allow access later.
2. Then, click on the **Anonymous Module Access** icon in the Webmin Configuration module.

3. The form that appears contains a table with two columns and, initially, two empty rows. Each row specifies a **URL path** on your server to which to allow unauthenticated access, and a **Webmin user** as which requests to that path should be treated.

In the first row, enter */images* for the path and *anonymous* for the user so the directory containing Webmin's title images can be accessed by unauthenticated clients.

In the second row, enter the path for the module that you want to allow (such as */status*), and *anonymous* as the user again. Never enter a path of */*, as it will allow unauthorized access to your entire Webmin server! The path to a specific CGI program (such as */custom/run.cgi*) may make sense in some cases.

4. Hit the **Save** button to turn on anonymous access. If you want to allow more than two URL paths, click on the icon again so that the table is redisplayed with two more empty rows.

If a user who has already logged visits a module that has been allowed anonymous access using the preceding steps, Webmin will still identify him correctly as the logged-in user.

51.20 Turning on SSL

Chapter 3 “Securing Your Webmin Server” explains in detail what SSL is, why it should be used, and which libraries are needed before Webmin can use it. The **SSL Encryption** page in this module can be used to turn SSL mode on or off or to generate another new key.

51.21 Setting Up a Certificate Authority

As Chapter 52 explains, the Webmin Users module can be used to request a client-side SSL certificate for a user. Before this is possible, however, you must set up your system as a certificate authority (CA). An authority is basically just an SSL certificate that can be used to sign other newly issued client certificates and to verify that those supplied by clients come from this CA.

Because client SSL authentication can only be used in SSL mode, Webmin must be running in that mode and thus have the `openssl` command installed before you can proceed. Once these requirements have been satisfied, follow these steps to set up a CA.

1. Click on the **Certificate Authority** icon on the module’s main page to bring up a form for entering the new CA’s details.
2. In the **Authority name** field, enter the name of the person issuing certificates, such as *Network administrator*.
3. In the **Email address** field, enter the address of the administrator for this server, such as *bob@foo.com*.
4. In the **Department** field, enter the subdivision of your organization in which the server resides, such as *Accounting*. This can be left blank if it makes no sense, such as for a home server.
5. In the **Organization** field, enter the name of your company or organization, such as *Foo Corporation*. Again, this may not make sense in all cases and so can be left blank.
6. Fill in the **State** field with the name of the state in which your server resides, such as *California*.
7. Fill in the **Country** code field with the two-letter, uppercase code for your country, such as *AU*.

8. Click on the **Setup certificate authority** button to generate the CA certificate and configure Webmin to use it. If you have done this before, any existing certificate will be overwritten.
9. Webmin users can now generate personal certificates using the Webmin Users module. Unfortunately, the web server will request that all clients supply a certificate as soon as one user has one, which can cause annoying dialog boxes to appear for people who are still set up for username and password authentication in some older browsers.

If you already have a CA in PEM format, with both the private key and certificate in one file, Webmin can be configured to use it instead of generating its own. Just go to the **certificate authority** page and paste it into the **Edit CA certificate** field, then hit **Save**.

To stop using a CA for validating clients altogether, hit the **Shutdown certificate authority** button on the same page. All users will be forced to revert to username and password authentication instead.

51.22 Summary

This chapter has explained how to use Webmin to configure itself, rather than using some other server or service. After reading it, you should understand how to perform such tasks as changing the port on which the web server listens, install and remove modules, switch themes, move modules between categories, and much more.

Webmin Access Control

This chapter tells you how to create new Webmin users with access to only some modules, and how to restrict exactly what users can do in each module.

52.1 Introduction to Webmin Users, Groups, and Permissions

A standard, out-of-the-box Webmin installation has only one user (called `root` or `admin`) who can use every feature of every module. On a home or office system used by just one person, that is all you need. Even if your system has multiple users, there may be only one who needs to perform system administration tasks.

There are, however, many situations in which the administrator may want to give some people access to a subset of Webmin's features. For example, you may have a person in your organization whose job it is to create and edit DNS zones and records. On a normal UNIX system, this person would have to be given `root` access so that he can edit the zone files and restart the DNS server when necessary. Unfortunately, once someone is able to log in as `root`, he has full control of the system and can do whatever he wants.

Webmin solves this kind of problem by allowing you to create additional users who can log in, but only access a few modules. You can further restrict what the user can do within each module so he cannot abuse its features to perform actions that he is not supposed to. Because Webmin still runs with full `root` privileges even when used by a restricted user, it still has access to all the configuration files and commands that it needs.

Some examples of the kind of access control restrictions that you can set up are:

- Creating a user with the right to edit directives in only a few Apache virtual servers that he owns. Global settings or directives in other virtual hosts cannot be edited.
- Allowing a user access to only one MySQL database, but not to other databases or user permissions. Similar access control can be set up for PostgreSQL.

- Giving a user the right to edit and create UNIX users with UIDs within a certain range and with home directories under a restricted directory. Important system users such as `root` or `bin` cannot be edited or even viewed.
- Giving a user access to the Squid access control list, but not to other functions. The user could be allowed to apply his configuration changes, but not to start or stop the proxy server.
- Creating custom commands and then giving a user the rights to run only some of them, but not create or edit any.
- Allowing a user to view and cancel print jobs in the Printer Administration module, but not edit or create actual printers.

Many of these rights would be impossible to grant using command-line tools without giving `root` access to the entire system. Even programs like `sudo` are limited when it comes to allowing a user to edit only part of a file, or run a command with only certain arguments.

You must be very careful when granting access to untrusted Webmin users, however, as even a small mistake in the access control configuration may allow the user to edit arbitrary files on your system or run commands as `root`. All it takes is a small hole for an attacker to sneak through and take total control of your system. Webmin's access control capabilities give you the power to lock down users, but only if used properly.

Even though it is possible to create a user with access to only his own email, home directory, and password, Webmin is not always the best way to provide this kind of single-user web interface. A superior program is Usermin, which was developed by the same author and shares much of the Webmin code and user interface. It is designed to give each UNIX user access to only those things that he would be able to access at the command line, such as his email, home directory files, and GnuPG configuration. Usermin runs most of its code with the permissions of the logged-in user, so there is far less chance of a user doing things that he is not supposed to, or even gaining `root` access. See Chapter 47 “Usermin Configuration” for more details on how you can manage Usermin from within Webmin.

52.2 The Webmin Users Module

If you want to create, edit, or grant permissions to a Webmin user or group, it must be done in this module. When you enter it from the Webmin category, the main page displays all users and groups on your system and the modules to which they have access, as shown in Figure 52.1. If a user is a member of a group, his membership and only those modules that did not come from the group will be shown.

On a normal Webmin system, only the `root` or `admin` user as whom you log in will appear, with access to all modules that are supported on your operating system.

52.3 Creating a New Webmin User

If you want to create a new user who can log in to Webmin, possibly with limited privileges, it must be created in this module. To do this, use the following steps:

1. On the module's main page, click on the **Create a new Webmin user** link above or below the list of existing users. This will bring up the creation form shown in Figure 52.2.

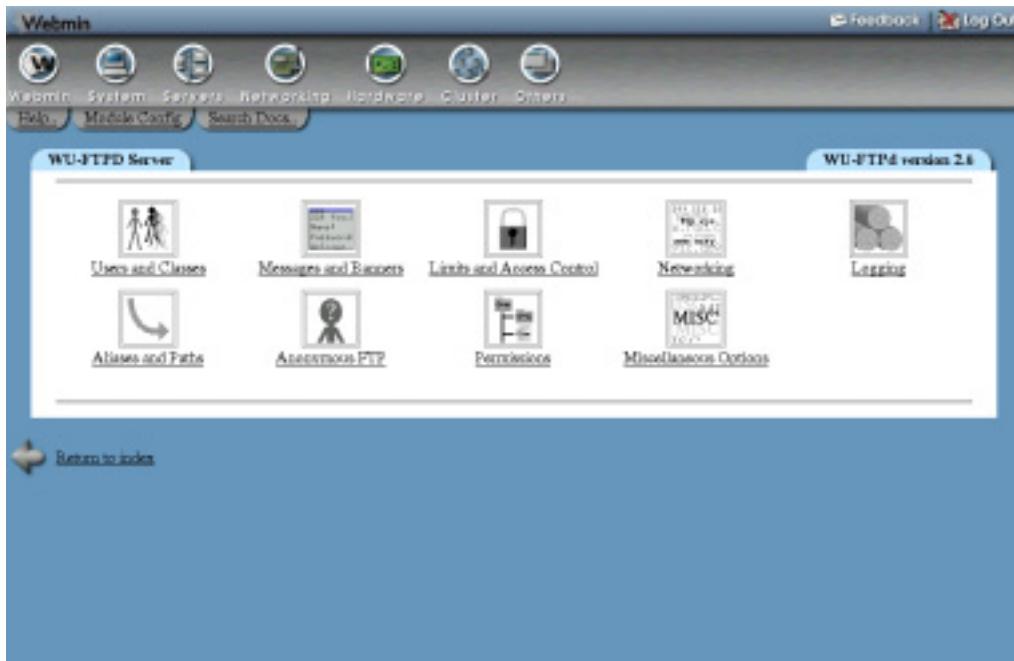


Figure 52.1 The Webmin Users module.

2. Enter a log in name into the **Username** field. The name cannot already be in use by any other user or group.
3. To make the user part of a group, select it from the **Member of group** field. Any modules that the group has will be granted to the user in addition to modules that you select on this page, and any access control restrictions that apply to the group in those modules will apply to the user as well.

See Section 52.6 “Creating and Editing Webmin Groups” for more information on how to add new groups to the list.

4. To give the user a normal password, select **Set to** from the menu in the **Password** field and enter it into the adjacent field.

If the new user has the same name as a UNIX user, you can select **UNIX authentication** to have Webmin use PAM, or read the /etc/shadow file to validate the user.

To prevent the user from logging in at all, select **No password accepted**. This might be a good idea when creating a user who will have limited privileges, so that he cannot log in until you have finished restricting his access.

5. To have Webmin use a different language for the user than the global default, select one from the **Language** field menu.
6. In most themes, module icons on Webmin’s main page are displayed under categories. If this new user is going to be granted access to only a few modules, this is not really necessary and you can change the **Categorize modules?** field to **No**.

7. To have the Webmin user interface displayed using a different theme for the user, set it in the **Personal theme** field.
8. To limit the addresses from which the new user can log in to Webmin, change the **IP access control** field to **Only allow listed addresses**. Then, fill in the text box next to it with hostnames, IP addresses, network/netmask pairs, or wildcard hostnames (like `*.foo.com`). Note that these restrictions are checked only after any global IP access control sets in the Webmin Configuration module have been passed.
9. Select all the modules to which you want the user to have access in the **Modules** section.
10. When done, click the **Save** button to have the new user created. You will return to the module's main page, and he will be able to log in immediately.

To further restrict what the new user can do in each module to which you have granted him access, see Section 52.5 “Editing Module Access Control”.

You can speed up the process of creating a new user who has the same attributes and access permissions as an existing user by using the module’s cloning feature. To clone a user, follow these steps:

1. Click on the username of the existing user that you want to clone on the module’s main page.
2. Click on the **Clone** button at the bottom of the editing form. This will take you to the creation form shown in Figure 52.2, but with most fields already filled in with the attributes of the original user.
3. Fill in the **Username** field and set the **Password**, as they do not get copied from the cloned user. You can also adjust the values in any of the other fields.
4. When done, click the **Create** button. The new user will receive a copy of all module access control settings from the original user, but they will not be updated if the original user is changed in future.

If you want to create many users with access to the same modules and the same access control settings, it is better to create a group and assign the users to it. That way you can change the settings for all members at once by just editing the group.

52.4 Editing a Webmin User

You can change the username, password, language, or any other attribute of a Webmin user (including the one you are logged in as) using this module. To edit a user, follow these steps:

1. Click on his username on the module’s main page. This will bring you to an editing form, similar to the one shown in Figure 52.2.
2. By default, the password will be left unchanged. To edit it, select **Set to** from the **Password** field menu and enter a new password into the field next to it.
3. Change any of the other fields on the form, as explained in Section 52.3 “Creating a New Webmin User”. You can even move the user to another group, which will cause him to lose access to all modules in the original group and need to gain access to those in the new group.



Figure 52.2 Creating a new Webmin user.

If you are editing yourself, Webmin will not allow you to take away access to the Webmin Users module. This is to protect you from locking yourself out of the module and not being able to grant yourself access back again.

- When you are done, click the **Save** button to apply the changes immediately. If the user-name or password was changed, the user is currently logged in, and Webmin is not in session authentication mode, the user will have to log in again.

You can delete a user by clicking the **Delete** button at the bottom of the editing form, which will also take effect immediately. Webmin will not allow you to delete yourself.

52.5 Editing Module Access Control

Many Webmin modules allow you to further restrict the actions that each user can perform using them. The actual access control options are different for each module and are documented in detail in the “Module Access Control” sections of the chapters that cover them. This section only describes the common process that you need to follow to configure what a user (or group) can do with a particular module. To control module access, follow these steps:

- On the Webmin Users main page, find the user or group that you want to restrict and click on the name of the module next to his name for which you want to edit the restrictions. This will bring up the access control editing form, an example of which is shown in Figure 52.3. That screenshot is from the Users and Groups module, so if you select a different module, the available options will not be the same.

The screenshot shows the Webmin interface with the title bar "Webmin". Below it is a toolbar with icons for Feedback, Log Out, and other system functions. The main window has a header "Messages and Banners". Under this, there are two tables: "Messages, banners and README files" and "README files". Both tables have columns for "Path", "When to display", and "Classes to display for". The "When to display" column contains radio buttons for "At login", "Entering any dir", and "Exiting dir". The "Classes to display for" column is empty. At the bottom of the form, there are several configuration options: "Greeting level" (radio buttons for "Hostname and version", "Hostname", and "Neither"), "Pre-login banner" (radio buttons for "None" and "From file" with a dropdown menu), "Hostname for messages" (radio buttons for "System hostname" and "Custom" with a dropdown menu), and "Owner's email address" (radio buttons for "Default" and "caroline@carlytrackett.com").

Figure 52.3 The module access control form for Users and Groups.

2. To stop the user from changing the module's configuration, set the **Can edit module configuration?** field to **No**. This should always be done because in most modules the configuration settings can be changed to allow the user to gain `root` access or otherwise escape the access control restrictions that you have set up.
3. Change other options on the form to restrict the user in whatever way you wish. Each module covered in this book has a section in its chapter that explains exactly what the fields mean and gives examples of how to set up common types of access control.
4. Click the **Save** button to make your changes immediately active and return to the module's main page.

Not all modules allow you to limit what a user can do, as it would not make any sense. For example, the Software Packages module does not allow access control restrictions to be configured. Its primary purpose is the installation of new packages, and any user with the rights to install a package could build and install his own that gives him `root` access. In modules like these, only the **Can edit module configuration?** option appears on the access control form. For modules that have no options other than this, there is no "Module Access Control" section in the chapter of the book that covers them.

At the start of the list of modules, next to every user, is an entry called **Global ACL**. If you click on this, it will take you to an access control form that allows the editing of restrictions that apply in all modules. The fields and their meanings are:

Root directory for file chooser There are many fields in Webmin for entering a file or directory name and next to most of them is a button that pops up a simple file chooser window. Users will not be able to use this file chooser to list directories outside whatever path you enter into this field. By default, it is set to / so the entire filesystem can be browsed.

This option only controls which directories can be browsed using the file chooser. A user can still enter ANY path into a filename field manually, unless the module has its own access control restrictions.

Users visible in user chooser In most Webmin modules, when a username field is displayed there is a button next to it that pops up a window for selecting either single or multiple users. This option allows you to control which users appear in that pop-up window, so a particular Webmin user cannot see all of the UNIX users on your system.

This access control option does nothing to stop the user from manually entering any username that he chooses, it just limits the list that appears in the pop-up window.

Groups visible in group chooser This option works in exactly the same way as the one above, but applies to the pop-up group selection window instead.

Can send feedback email? When using the Webmin theme that is enabled by default, a **Feedback** button appears on every page in the upper-right corner. Changing this option to **No** will remove the button while changing it to **Yes, but not with config files** will prevent the user from sending feedback with the **Include module configuration in email** option selected.

Because all feedback goes to the author of Webmin by default, disabling it makes sense for users other than the master administrator.

Can accept RPC calls? Webmin has its own RPC (remote procedure call) mechanism that is used by the cluster modules, System and Server Status modules, and other modules. Any client program that makes an RPC call to a Webmin server must first log in as a normal user using a web browser client. An RPC client, however, can access all of the features of Webmin, edit arbitrary files, and execute commands as root, regardless of any access control settings. For this reason, for users without full access to Webmin, this option should be set to **No**.

The default is **Only for root or admin**, which means that only if the user is called root or admin can it be used to log in for RPC. Because the root and admin users typically have full access to Webmin anyway, this is not a security problem. If you create a new user with one of these two names, however, and grant him only limited Webmin access, make sure this option is set to **No**.

For almost all Webmin users, even those that are granted only limited access to some modules, the default Global ACL options will work fine and do not need to be changed.

52.6 Creating and Editing Webmin Groups

If you want to create a large number of users who will all have access to the same modules with the same access control options, the best solution is to create a Webmin group. Like users,

groups have access to a subset of the available Webmin modules and have access control permissions in those modules. If you change the available modules or permissions for a group, those of all member users will change as well.

A group can itself be a member of another group, from which it will inherit all allowed modules and access control settings. If the parent group is changed in any way, those changes will flow through to all member groups and their member users. There is no limit to the number of levels of group nesting that you can create.

To create a new group, follow these steps:

1. On the Webmin Users module main page, click on the **Create a new Webmin group** link near the bottom of the page under the **Webmin Groups** section. This will take you to the group creation form shown in Figure 52.4.
2. Fill in the **Group name** field with a unique name that is not used by any other existing user or group.
3. To make this new group a member of an existing one, select it from the **Member of group** menu.
4. Select all the modules to which you want members of this group to have access from the **Members' modules** list. Those from any parent group will be automatically included.
5. Click the **Save** button to have the new group created, and your browser will return to the module's main page.
6. Configure access control settings for members of the group by clicking on module names next to the group name on the main page, as described in Section 52.5 "Editing Module Access Control".
7. You can now create new Webmin users or edit existing ones to become members of the new group.

Once a group has been created, it can be edited by clicking on its name from the table under Webmin Groups on the module's main page. This will take you to the group editing form on which you can change any of its attributes before applying them with the **Save** button. Or you can delete the group altogether with the **Delete** button, as long as it does not have any member users or groups.

52.7 Requesting a Client SSL Key

Users normally authenticate themselves to Webmin with a username and password. If they are running in SSL mode and using a modern browser like IE or Netscape, however, it is possible to set up Webmin to authenticate them via a client-side SSL key, instead. Usually an SSL web server sends its certificate to the client for authentication purposes, but the protocol also allows clients to send their certificates to the server.

The advantages of this method are that there is no longer a need to remember a username and password and that the old method of authentication can be disabled so that only clients with the SSL key can connect. Attackers, therefore, cannot break in by guessing your password or by looking over your shoulder as you type. Some browsers even support the storage of SSL keys on removable smart cards, which is even more secure.



Figure 52.4 The Webmin group creation form.

Before a client key can be issued, Webmin must be switched to SSL mode and a certificate authority key generated. Both these subjects are covered in Chapter 51. Once this is done, the steps to request a key are:

1. Log in to Webmin as the user for whom you want to create a key, using the browser in which the key should be stored. Browsers keep a list of client-side keys, usually protected by some password that must be entered only once when a key is first needed. It is usually possible to export keys to another browser of the same type.
2. Go to the Webmin Users module and click on the **Request an SSL Certificate** icon at the bottom of the page.
3. The form that appears will be different depending on whether you are running Internet Explorer or Netscape. The following instructions apply to Netscape and Mozilla, as they are the most common browsers on UNIX systems.
4. Enter a name into the **Your name** field, such as *Joe Bloggs*.
5. Enter your email address into the **Email address** field, such as *joe@example.com*.
6. If your Webmin system is on a company or organization network, fill in the **Department** and **Organization** fields. Otherwise, they can be left blank.
7. Enter the state your system is in into the **State** field, such as *California*.
8. Enter a two-letter country code like *US* into the **Country code** field.
9. From the **Key size** menu, select the number of bits in the SSL key that will be created. The higher the number, the more secure, but the longer it will take to be authenticated. 1024 bits should be secure enough for anyone.

10. Click on the **Issue Certificate** button. Your browser should pop up a window showing the key-generating progress, which is done on the client system. When it is complete and has been sent back to Webmin, a success page will be displayed.
11. Click on the **pick up your certificate** link to store the newly generated and signed key in your browser. You may be asked by the browser for a password to secure your certificates.
12. To test that everything worked, log out of Webmin and quit your browser. Then, rerun it and attempt to connect. The login page should be bypassed and the main menu displayed. The text **SSL certified** should appear next to your username in the browser's status bar.
13. Once SSL client authentication is working, you may no longer want clients to be able to log in as this Webmin user with a username and password. To enforce this, go to the Webmin Users module, click on your username, select **No password accepted** from the **Password** menu, and hit **Save**.

52.8 Viewing and Disconnecting Login Sessions

When Webmin is in session-authentication mode (as it is by default), it keeps track of all currently logged-in users. You can view this information and cancel sessions that seem to be invalid by following these steps:

1. Click on the **View Login Sessions** icon at the bottom of the Webmin Users module main page.
2. On the page that appears, the ID, login name, and connection time of each active session will be listed, with the newest shown first. It is quite possible for several sessions to exist for the same user, as many people do not bother to properly log out of Webmin. Old sessions, however, will be automatically removed after one week.
3. To view the actions performed in a particular session, click on the **View logs** link in the last column. This will take you to a list of actions in the Webmin Actions Log module, as covered in Chapter 54.
4. To cancel a session, click on its ID. This will immediately log the user out but will not kill any CGI programs that Webmin is currently running.

52.9 Module Access Control

Interestingly, the Webmin Users module has its own set of access control options that can be used to determine which other users a particular Webmin user can edit. This is typically used to give a subadministrator user the right to create and edit only a subset of Webmin users and to grant them access to only a few modules. To set up this kind of access, follow these steps:

1. In the Webmin Users module, click on Webmin Users next to the name of the subadministrator you want to restrict.
2. Change **Can edit module configuration?** to **No**.
3. Set the **Users who can be edited** option to **Selected users** and choose those accounts that you want the subadministrator to be able to edit.

4. Change the **Can grant access to** field to **Selected modules** and choose from the list below the modules that the administrator is allowed to grant to new or edited users. There is not much point in choosing modules that the subadministrator cannot already access.
5. Change **Can rename users?**, **Can edit module access control?**, **Can request certificate?**, **Can configure user synchronization?**, **Can configure UNIX authentication?**, **Can view and cancel login sessions?**, and **Can edit groups?** to **No**. All the other yes/no fields can be set to **Yes**.
6. Change the **Newly created users get** field to **Same module access control as creator**. Because the subadministrator is not allowed to edit the access control settings of modules that he grants to other users, they will always get the same settings that he does.
7. To force all new and edited users to be a member of a single group, change the **Can assign users to groups** field to **Selected** and choose the group from the list provided. To prevent the subadministrator from choosing any group, select the **<None>** option. It may make sense for you to allow the creation of users who must be members of a group that has been set up with the appropriate restricted modules and permissions. If so, you should not select any modules at all from the list in Step 4 so that only those from the group are available to created users.
8. Click the **Save** button to return to the module's main page.
9. If you are not forcing all new users to be a member of a particular group, make sure that the access control settings for the subadministrator in other modules have been set correctly. Any new users that he creates will inherit them.

The Webmin Users access control settings can also be configured to allow a user to change some of his own settings, but not edit other users or grant himself additional privileges. To set this up, follow these steps:

1. Click on Webmin Users next to the name of the user or group to whom you want to grant the rights to edit herself. Naturally, the user must have already been granted access to the module.
2. Change **Can edit module configuration?** to **No**.
3. Set the **Users who can be edited** option to **This user**.
4. Set the **Can grant access to** field to **Selected modules**, but do not select any from the list provided. This will prevent the user from giving himself any additional module access.
5. Change **Can request certificate?**, **Can change language?**, **Can change categorization?**, and **Can change personal theme?** to **Yes**, and all of the other yes/no fields to **No**.
6. Change **Can edit groups?** to **No** and set **Can assign users to groups?** to **Selected**, but do not select any from the list.
7. Finally, click **Save**. The Webmin user will now be able to use the module to change only her own password, language, theme, and categorization mode, and request a client-side SSL certificate.

52.10 Configuring the Webmin Users Module

The Webmin Users module has several options that can be configured by clicking on the **Module Config** link on the main page. The editable fields and their meanings are shown in Table 52.1.

Table 52.1 Module Configuration Options

Display user modules in	By default, this option is set to Table , which tells Webmin to display all modules granted to each user or group next to their names on the main page. On a system with many users, however, this can make the page large and hard to navigate, which is why the Pull-down menu choice exists. If selected, a menu of modules is shown next to each user, along with a button for editing the access control settings of the chosen module.
Sort users and groups by	When Order in file is chosen, users and groups on the main page are shown in the order in which they were added to Webmin. If Name is selected, however, they will be sorted alphabetically by user or group name, instead.

52.11 Summary

This chapter has shown you how Webmin's more advanced access control features can be set up, such as the creation of multiple user accounts and the restriction of users' access in certain modules. It has also explained how to define groups so that settings can be easily applied to many users at once, how to disconnect user sessions, and how to request an SSL key to avoid the need to enter a username and password to log in to Webmin. Many users will not need to use this module, though, as the default Webmin configuration includes a `root` or `admin` account that has full access to all modules.

Webmin Servers

In this chapter, the Webmin module for listing other servers on your network is explained, and its relationship to the Cluster modules and RPC is documented.

53.1 The Webmin Servers Index Module

This module really serves two purposes—one simple and one quite complex. You can use it to create a master index of other systems running Webmin on your network, each of which is shown as an icon that you can click on to link to the server. Each icon can either be a normal link or a “tunnel” that logs you into another server automatically with all traffic sent via the first system.

The module can also be used to define systems that can be controlled by a master Webmin server using the System and Server Status module and the modules in the Cluster category. Each of these other systems must also have Webmin installed and a special RPC (Remote Procedure Call) protocol is used by the master to communicate with and control the slaves. How this all works is explained in detail in this chapter.

When you click on the module’s icon in the Webmin category, a page like the one shown in Figure 53.1 will be displayed. Most of the page is taken up with a table of icons, one for each of the other servers that you have added. Of course, if this is the first time the module has been used, no server icons will appear initially. At the bottom of the page are buttons for automatically finding other Webmin servers on your local network.

Even though it was designed for creating an index of Webmin web servers, there is no reason that you cannot create icons for other types of web servers. The module’s RPC features, however, will naturally only work when communicating with a host running Webmin.

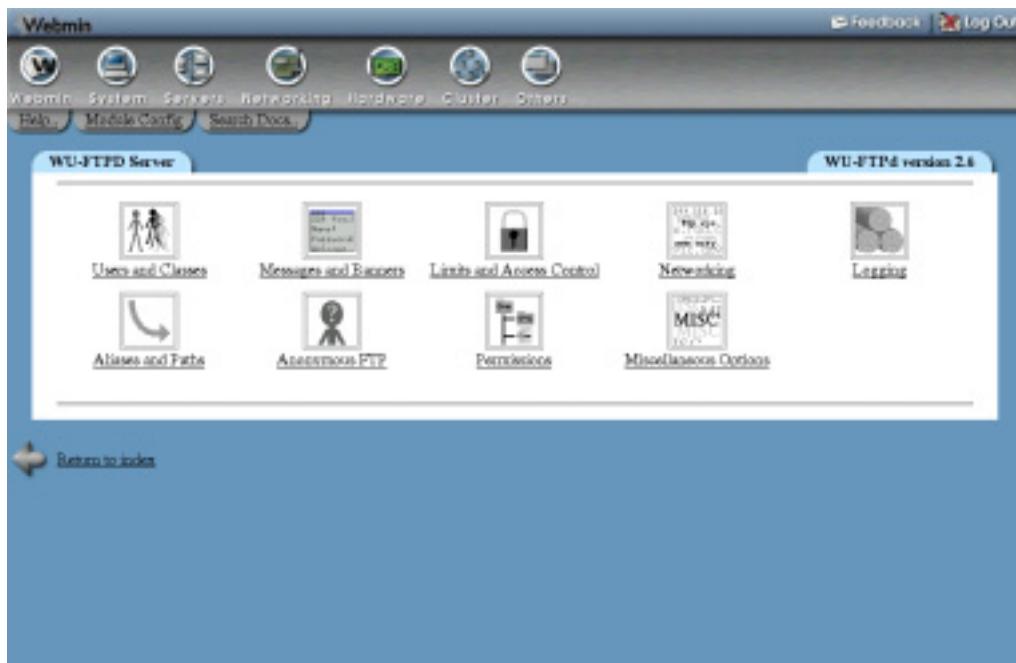


Figure 53.1 The Webmin Servers Index module.

53.2 Adding a Webmin Server

To add a new server to this module, either to provide a link to it or so that it can be managed with one of the Cluster group modules, follow these steps:

1. Click on the **Register a new server** link on the main page above or below the existing icons.
2. In the **Hostname** field, enter the Internet hostname or IP address of the other server, such as *server.example.com*.
3. In the **Port** field, enter the port that Webmin is listening on—usually *10000*.
4. From the **Server type** menu, choose the operating system that the other host runs. This only sets the icon that will be used to represent the server.
5. If the other Webmin server is in SSL mode, select **Yes** in the **SSL server?** field. This option can only be used if the master system has the Net::SSLeay Perl module and the OpenSSL library installed so it can make a client-mode SSL connection.
6. When the **Description** field is set to **From hostname and port**, the server's hostname and port number are shown under its icon on the module's main page. You can, however, select the second option and enter an alternate description to be shown instead, such as *Corporate Web Server*.
7. Servers defined in this module can be categorized into groups for easier addition in the Cluster category modules. In the **Member of server group** field you can select one of the following options:

None The system you are adding will not be in any group.

Existing group If some groups have already been defined, this server will be in the group selected from the menu next to this option. If no groups exist yet, this option will not even appear.

New group The server will be added to the new group whose name you enter in the adjacent text field.

A group will cease to exist as soon as all of the servers in it have been deleted or changed to another group.

8. The **Link type** field is possibly the most important on this form, as it determines if the new server can be used in the Cluster modules and the System and Server Status module. It also determines whether the icon is a normal link or a tunnel. Your options are:

Normal link to server RPC calls cannot be made to the other server and the icon on the module's main page will just be a normal web link. If the system is running some other web server-specified port, you should select this option.

Login via Webmin with username This option must be chosen if you want to use Webmin's RPC features to control this server, such as with the Cluster category modules. If selected, you must enter a username and password for Webmin on the remote host into the fields next to it. The user should be `root` or `admin` because other Webmin users are not, by default, allowed to receive RPC calls unless specifically authorized to do so. RPC can be used to run any command or modify any file on a server, which is why access to it must not be granted to untrusted Webmin users. If this mode is chosen, the server's icon on the main page will be a tunnel that automatically logs whoever clicks on it into the remote server as the specified user.

Login when icon is clicked on If this option is chosen, the server cannot be used for RPC, but its icon will still be a tunnel to the remote host. When first clicked on, it will prompt the user for a login and password for the remote system, which will be stored in a cookie in the user's browser. This option is handy if you want various users to be able to make use of the tunneling feature, but still log in to the remote system as themselves.

9. If **Login via Webmin with username** was selected above, the **Make fast RPC calls?** field determines whether the new fast RPC protocol will be used or the older slow protocol. You can either select **Yes** to force the use of fast mode, **No** to force slow mode, or **Decide automatically** to have Webmin use fast mode only if it is available. If the automatic option is chosen, and the server cannot be contacted or logged into, an error message will be displayed when you hit the **Create** button later.

Versions of Webmin before 0.89 did not support the fast protocol, but most systems should have been upgraded beyond that by now. You will generally want to use the faster mode all the time, unless a firewall is blocking the direct TCP connections that it uses. See Section 53.6 "How RPC Works" for more details on the differences between the two modes.

10. Finally, hit the **Create** button to add this new server. As long as there were no errors in the form you will return to the module's main page, which should include a new icon.

The icons for servers not created in **Normal link to server** mode will actually be links to a program on this master server that connects to the remote system for you. This can actually be useful

if your master server is accessible from the Internet but internal hosts are not. For example, if you only have a single Internet IP address and are using NAT. When you access those internal servers by clicking on their icons in this module on the master system, your browser is really only connecting to the master server, which is then tunneling the requests through to the chosen slave.

On a Webmin system with multiple users, you should be careful about giving access to this module to untrusted users. Anyone who can click on an icon for a server in **Login via Webmin with username** mode will be connected to the remote system as the user specified for that server, not himself. This will probably allow him to do things with `root` privileges on that remote host that he would not be able to do on the master system.

Section 53.7 “Module Access Control” explains how you can control which server icons a particular user can use, so that untrusted people can be limited to those in the safe **Normal link to server** or **Login when icon is clicked on** modes.

53.3 Editing or Deleting a Webmin Server

Once a server has been added to this module, you can edit all of its details or even delete it altogether. To edit or delete a server, follow these steps:

1. On the main page, click on the **edit** link next to the name of the server that you want to change. This will bring up an editing form almost identical to the one for adding a server.
2. All of the fields can be edited, and have the same options and meanings as explained in Section 53.2 “Adding a Webmin Server”. The only exception is the **Make fast RPC calls?** field, which will not have the **Decide automatically** selection if the module has already worked out the RPC mode that the remote server supports.
3. Hit the **Save** button to activate your new settings. Or, if you want to remove this server from the module, hit **Delete** instead. Any other modules (such as those in the Cluster category) that make use of this server will automatically remove it from their lists.

53.4 Using Server Tunnels

When you click on an icon for a Webmin server in one of the tunnel modes, you will be connected to it via this master system. The user interface of the remote host will be almost exactly the same as if you logged in normally, except that every page will include a special **Webmin Servers** link. When clicked on, this will take you back to the Webmin Servers Index module on the master system, which is more convenient than hitting the back button in your browser a few hundred times.

For tunneling to work, the master server must analyze and modify the HTML sent back by the remote host that you are logging into. Currently this works well for Webmin servers, but may fail if you are tunneling through to some other web-based application or website that uses HTML not supported by this module. Symptoms of this include links pointing to nonexistent pages on the master server and images that are not loaded properly.

Clicking on the icon for a server’s **Login when icon is clicked** mode will initially display a login form for entering a username and password for the remote system. This will appear even if the remote host does not actually run Webmin on the chosen port, but it can be used to log in to any web server that uses standard HTTP authentication. After you log in, your username and password will be remembered until you either quit your browser (thus discarding the cookie) or click on the **logout** link that appears below server’s icon on the module’s main page.

53.5 Broadcasting and Scanning for Servers

If you have a large number of Webmin servers on your network, adding them one by one to this module can be tedious. There is a better way though. The master system can broadcast on your local LAN for other Webmin servers or send requests to hosts within a specific network to probe for servers. Any servers that are found will be automatically added to this module, although only in **Normal link to server** mode. There is no way for the master system to automatically determine a login and password for a remote system—this would be a huge security hole if it were possible!

To find other Webmin servers, follow these steps:

1. If you only want to search your local LAN, click on the **Broadcast for servers** button at the bottom of the module's main page.

To search another network, enter its address into the field to the right of the **Scan for servers** button before hitting it. This must be a class C network, entered like *192.168.1.0*.

2. A page listing the URLs of servers that are found will be displayed. New ones will have **Found new server** before their URLs, those already on the main page will have **Found known server**, and responses from the master system itself will have **Found this server**.
3. When the process is complete, you can return to the main page, which will now contain an extra icon for each of the newly found servers. They can then be edited to switch to **Login via Webmin with username** mode to use them for RPC.

All versions of Webmin since 0.75 listen on UDP port 10000 for the broadcast and scan packets sent out by this module, and reply with their hostname, port number, and SSL mode. A server will not be found if a firewall is blocking this port or if UDP listening has been turned off for security reasons.

53.6 How RPC Works

RPC is a protocol that one Webmin system can use to control another. An RPC request is usually a call to a function in the library of some module, and includes the parameters to that function. There are, however, other RPC request types for transferring data to and from a server, checking to see if a module is available, getting a module's configuration, and executing a piece of Perl code. This section explains the technical details of how it works, and can be skipped if you are not a programmer and not having any trouble with RPC connections.

When you set up the System and Server Status module to fetch status information from a remote system, an RPC call is made to functions in the same module on that system to determine if a service is up or down. When a user is added in the Cluster Users and Groups module, multiple RPC calls are also made to add her to the password file, create her home directory, and copy files into it. Chapter 56 explains how to make use of RPC in your own modules, and what its limitations are.

As explained earlier, RPC has two different modes—fast and slow. Slow mode is simplest, as it uses an HTTP request from the master to the slave for each RPC function call, file transfer, or request for information. All parameters, data, and return values are included in that request and its response and no other TCP connections are made. The advantage of this mode is that it can work through firewalls and proxies, as long as HTTP requests to port 10000 are allowed.

Apart from being slow, this mode has one big negative—HTTP is a stateless protocol, but Webmin RPC calls are not stateless. It is quite possible for one function call to set a global variable

that the next function call depends upon. This means that a background process in which state is kept must be started on the remote system for each master that opens an RPC session. There is no way, however, for a slave system to automatically detect when the master CGI program has finished and shut down the background process because no direct connection between the two exists!

Webmin's solution is to have the process exit when the master makes a special RPC call, or after 30 seconds of inactivity. If a master CGI program does not invoke the `remote_finished` function, the remote process will hang around consuming memory until the timeout elapses. If for some reason more than 30 seconds pass between RPC calls to the same host, the background process will exit and future RPC calls will fail.

The newer fast RPC protocol solves these problems using only one initial HTTP request to have a background process started on the remote system. The master server then makes a TCP connection to this process (which is listening on a free port), and sends RPC requests through that connection instead. When the master program exits, this connection will be automatically torn down and the remote background process will exit. No special function calls or timeouts are needed.

Fast RPC mode has much better support for transferring large files to and from remote systems. The slow mode attempts to encode files inside an HTTP request, which can fail if they are too large. The newer mode instead transfers them unencoded through a separate TCP connection, which is quicker and far more reliable. The Cluster Software Packages and Cluster Webmin Configuration modules may fail when installing a large package in slow mode.

The only problem with fast mode is that some firewalls may block the TCP connection, which is typically made on a port that is one or two higher than the remote host's base Webmin port, such as 10001 or 10002. Multiple connections may be made if data is transferred with RPC, so any firewall on your network between the two servers must be configured to allow connections from the master to the remote host on ports in the range of 10000 up to 10100.

53.7 Module Access Control

If you have more than one Webmin user on your system, you may want to make this module available to other people without giving them access to all server icons or the ability to add servers. This is useful if you want others to only see icons for servers not in **Login via Webmin with username** mode, thus turning the module into just an index of other systems on your network.

The first step is to assign this module to a user, as explained in Chapter 52. You can then restrict him to only being able to see and use the tunnels for certain servers by following these steps:

1. Click on Webmin Servers Index next to the name of the user or group in the Webmin Users module to bring up the access control form.
2. Change the **Can edit module configuration?** field to **No**, so he cannot change the user interface for other people.
3. In the **Can use servers** field, choose **Selected** and select the ones that you want to make visible from the list provided.
4. Change the **Can edit servers?** and **Can find servers?** fields to **No**.
5. Hit the **Save** button to activate the new restrictions.

Hiding a server from a user in this module does not stop him from using it in other modules that make use of RPC.

53.8 Configuring the Webmin Servers Index Module

This module has several settings that control how its user interface appears and how scans for servers are done. You can edit them by clicking on the **Module Config** link on the main page, which will bring up a form containing the fields shown in Table 53.1.

Table 53.1 Module Configuration Options

Resolve found server addresses	When Yes is selected, the module will convert the IP addresses of any servers found by scanning or broadcasting into hostnames. You should only need to select No if reverse address resolution is slow or broken on your network.
Time to wait for scan responses	This field sets the number of seconds that the module will wait for a response to packets sent out to find other Webmin servers. It may need to be increased from the default of 5 seconds if you are scanning a remote network.
Show servers as	When Table is chosen, servers on the main page will be listed as rows in a table, which can be useful as it takes up less space than the default Icons mode. More details are also shown for each server.
Sort servers by	This field controls how servers on the main page are sorted. The options are: IP Address Servers are sorted by the IP address to which their hostnames resolve. Hostname Sorting is done by the servers' hostnames. Description Sorting is by the text that appears below the servers' icons. OS Servers are sorted by the operating system that you select for them. Group Servers are sorted into their chosen groups. Order created Servers are simply listed in the order in which they were added to this module.
Show status for servers	If Yes is selected, the editing form for a server will include a new Server status field that shows the version of Webmin it is running or an error message if the system cannot be contacted or logged into. This field will only appear in Login via Webmin with username mode.
MSC cluster groups directory	This field can be used to specify a directory containing additional server groups in the format used by MSC Software's clustering tools.

53.9 Summary

This chapter has explained the purposes of the Webmin Servers Index and how you can use it to add servers for easy linking, for making a tunneled login, or for RPC calls. It has also explained what RPC is and how other modules use it to allow multiple Webmin servers to be managed from a single interface.

Logging in Webmin

This chapter explains how and what Webmin logs, and how those logs can be searched and viewed.

54.1 Introduction to Logging

When logging is enabled, Webmin will record every action taken using it that has some effect on your system, such as the creation of a user or the changing of an Apache setting. Pages that do not actually change anything on your system, such as those that just display icons, list users, or show the current settings for some object will not write anything to the action log. In this way, it is different than the separate CLF log file that Webmin writes to `/var/webmin/miniserv.log`, which records every single page visited and image loaded.

Most actions performed in Webmin change configuration files, run commands, or execute SQL statements. When the recording of these file changes is enabled, the details of each will be included in the actions log so you can see exactly what Webmin did when you told it to create a UNIX user or delete a DNS zone. This can be helpful for understanding what is really going on behind the scenes if you are new to system administration or want to see how actions are implemented. Not all modules perform action logging, though, particularly those that are old or have been written by third-party developers.

As Section 51.4 “Setting Up Logging” explains, logging can be turned on in the Webmin Configuration module. Basic action logging is enabled by default, but the recording of file changes is not. To gain the most benefit from the Webmin Actions Log modules, file changes should be logged as well. This will slow down the program slightly though, and consume more disk space for recording the changes.

Some types of action will never have any associated file changes logged, even if this feature is enabled. Such actions might perform all their work with network connections or modify a file

so large that generating the differences between the old and new contents is impractical. Or file change logging may not have been implemented in the module at all.

The actual file in which actions are recorded is called `/var/webmin/webmin.log`. Its format is unique to Webmin but records the details of each action on a separate line in a simple text format. If the logging of file changes is enabled, the directory `/var/webmin/diffs` is used to store files containing the details of changes and commands used. Each file in this directory is named to match the ID of an action and contains the changes made to one file in `diff` format.

If you are looking for the files in `/var/webmin` on your system and cannot find them, check in `/var/log/webmin` instead. Some packaged versions of the software created by other Linux distribution vendors use this alternate directory to better fit in with the normal Linux log file layout.

54.2 The Webmin Actions Log Module

This simple module exists solely for viewing action logs created by Webmin. It can be useful for finding out what a particular user is up to or who has been doing what in some module. On a system with multiple administrators, tracking down who broke a particular server's configuration could ordinarily be tough, but this module makes it relatively easy.

The module can be found under the Webmin category on the main menu, and clicking on its icon will bring up the search page shown in Figure 54.1. Before you can view the details of a particular action, it must be found using the search form.

54.3 Displaying Logs

The form on the module's main page lets you find actions using three different search criteria. Only actions that match all three will be displayed, rather than those that match any one of the criteria. You can find actions by the Webmin user that performed them, the module in which they were carried out, and the date and time that they occurred.

To display log files, follow these steps:

1. In the first section of the form, select **By user** if you want to display only actions by a particular user and choose the user from the adjacent menu.

To exclude some user's actions from your search, use the **By any user except** option instead.

To include all users in the search, choose **By any user**.

2. In the second section, choose **In module** and select it from the menu to limit the search to actions performed in a specific module. Only modules that are currently installed will be listed.

To search all modules' actions, select **In any module** instead.

3. The final section determines which date range an action must fall into to be included in the results. If **Between** is chosen, you can select or enter one or two dates using the fields next to it. If the first date is omitted, all actions up to the second date will be included. Similarly, if the second date is missing, all actions from the first date onwards will match.

If **For today only** is selected, only actions that have occurred during the current local-time day will be included in the result.

If **At any time** is chosen, the date on which an action occurred will be ignored.

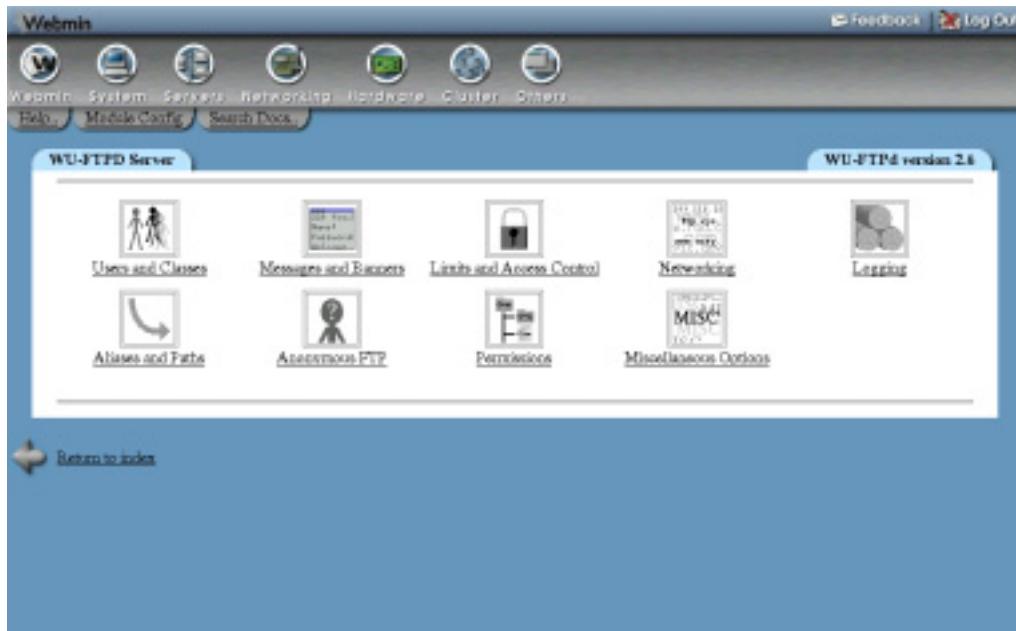


Figure 54.1 The Webmin Actions Log module.

4. Hit the **Search** button to display a page of actions that match the chosen criteria. This may take a few seconds to display if your Webmin log is large. If any were found, the resulting page will provide a short description for each action (such as **Created user fred**), the module it comes from, the Webmin user responsible, the client system from which he was connected, and the date and time it occurred.
5. Click on the description in the **Action** column to go to a page showing more details about the action. If logging of file changes was enabled at the time it occurred, the changes made to any files by the action will be shown as well, along with any commands executed or SQL statements run. Only actions from the MySQL and PostgreSQL modules will include SQL statements used to do things like creating a table or modifying a column.
6. When Webmin is in session authentication mode, a **Session ID** field will be shown in this form. Clicking on the ID will bring up a list of all actions performed by the user in a single browser instance from the time she logs in until the time she logs out.

It is possible to display every single action logged on your system by leaving the options on the search form set to their defaults. This is likely to take quite a while to generate, however, and will produce a lengthy HTML page.

54.4 Summary

After reading this chapter, you should understand how Webmin's action logging system works, and what information is recorded for each action, depending on the current configuration. You should also know how to use the Webmin Actions Log module to search for and view the details of recorded events.

Webmin Module Development

This chapter should be read if you are planning to write your own Webmin module, as it explains all the requirements for creating a usable module.

55.1 Introduction

Webmin is designed to allow the easy addition of new modules without changing any of the existing code. A module can be thought of as something like a Netscape or Photoshop plug-in. It can be written by someone other than the developers of Webmin and distributed under any license the developer chooses.

A module should be written to administer one service or server, such as the UNIX password file or the Apache Web server. Some complex system functions may even be split over several modules. For example, disk partitioning, mounting disks, and disk quota management are three separate modules in the standard Webmin distribution.

Theoretically modules can be written in any language. To make use of the Webmin API Perl, however, version 5.002 or greater should be used. A module should be written entirely in Perl, with no C functions or external binary programs. The goal is for modules to be as portable as possible across different UNIX systems and CPU types.

Modules written in other languages will not be displayed using the default theme included with recent versions of Webmin. This theme replaces the standard Perl `header` function with its own that displays category icons at the top of every page. For this reason, using Perl is strongly recommended.

At their simplest, modules are really just directories of CGI programs that Webmin's web server runs. There are certain rules, however, that should be followed to make sure that they work with the Webmin API, main menu, and access control system. Even though you can just stick any existing CGI script into a module directory, this is not a good idea.

55.2 Required Files

Every module has its own directory under the Webmin base directory, in which all the module's CGI programs and configuration files must be stored. For example, if the Webmin base was `/usr/local/webmin-1.090`, a module called `foobar` would be installed in `/usr/local/webmin-1.090/foobar`.

For a module to be displayed on the main Webmin menu, it should contain at least the following files (only `module.info` is mandatory):

`images/icon.gif` The icon displayed on the main menu for this module. The icon should be 48x48 pixels, and should use the same color scheme as the other icons on the main menu.

`module.info` This file contains information about the module and the operating systems under which it runs. Each line of the file is in the *name=value* format.

Required names and their values are:

`name` A short name for this module, such as *FooAdmin*.

`desc` A longer description for the module, such as *Foo Web Server*. This is the text that will appear below the icon on the main menu.

`os_support` A space-separated list of operating systems that this module supports. The module will only be displayed on the main menu if the OS on which Webmin is running is in the list or if there is no `os_support` line at all. Unless your module configures a particular service that only exists on a few operating systems (such as XFree86), this line should be omitted instead of trying to list all of those supported by Webmin.

The actual operating system codes used in this line can be seen in the third column of the `os_list.txt` file in the Webmin root directory, and are the same as those that can be appended to the names of `config-` files, as explained in Section 55.4 "Module Configuration". To specify only a certain version of a particular OS, add it to the OS name after a slash. For example, a `module.info` file might contain

```
os_support=redhat-linux open-linux suse-linux/8.1
```

If your module supports all Linux distributions, you can use the OS code `*-linux` in this line.

`depends` A space-separated list of other modules upon which this module depends. If module *foo* depends upon module *bar*, then Webmin will prevent module *bar* from being deleted while *foo* is still installed. Webmin will also prevent *foo* from being installed if *bar* has not yet been.

The list can also contain a Webmin version (such as *0.75*) upon which the module depends. In that case, installation of this module by an older version of Webmin will not be allowed.

`category` This value determines under which tab on the main Webmin page your module will be categorized. Supported values are `webmin`, `system`, `servers`, `net`, and `hardware`. If your `module.info` file has no category line, it will appear under the **Others** category.

lang/en The text strings used by this module, as explained in Section 56.3 “Internationalization”.

Each icon on the main menu is a link to the module directory. You must, therefore, have an `index.cgi` or `index.html` file to be displayed when the user clicks on the icon. A typical module contains many `.cgi` programs that are linked to `index.cgi`, each of which performs some function such as displaying a form or saving inputs from a form.

When you first create a new module, it will not be in the ACL of any Webmin user and you will not be able to see it in the main menu. You must first delete the `/etc/webmin/module.infos.cache` file to clear the cache of known modules. Then, to make your module visible, either edit the `/etc/webmin/webmin.acl` file, or use the Webmin Users module to grant yourself access.

55.3 Module CGI Programs

The Webmin web server treats files with the `.cgi` extension as CGI programs, just like most other web servers. All the forms, menus and other pages in your module will be generated by CGI programs, so knowledge of the basic concepts of CGI programming and HTML is necessary for writing a module.

All CGI programs are run with `root` privileges, which is generally necessary for them to be able to edit configuration files. In some cases, your code may drop those privileges by switching to another user. For example, if the module’s access control settings for some Webmin user specify it.

Assuming your module is being written in Perl, you should begin by writing a Perl script that contains functions used by the CGI programs in your module. This script is usually called something like `lib.pl` or `foobar-lib.pl`. A minimal example of such a script might look like:

```
# foobar-lib.pl
# Common functions used for managing the foobar user list

do '../web-lib.pl';
&init_config();

# list_users()
# Returns a list of all foobar users
sub list_users
{
    ...
}
```

The three important lines in this example are:

1. `do '../web-lib.pl';`

The `web-lib.pl` file in the Webmin root directory contains a large number of functions that are useful for developing Webmin modules. All CGI programs should indirectly or directly require this module. You should use `do` instead of `require`, as the latter statement will not reread a file that has already been read in the same Perl program.

This causes problems if your module is called from some other module with the `foreign_require` function.

2. `&init_config()`

This function (defined in `web-lib.pl`) initializes the following global variables:

`%config` Contains the current configuration for this module. This is typically used to store user editable options and operating system-specific information. Module config files are described in more detail in the following list.

`%gconfig` Contains the global Webmin configuration. See the following descriptions for more details.

`$module_name` The name of this module, which is just the name of the directory in which the module resides.

`$module_config_directory` The directory in which this module's `config` file is stored. If your module creates permanent files or programs for some reason (such as print driver scripts), they should be created in or under this directory.

`$tb` The background color for table headers.

`$cb` The background color for table bodies.

`$scriptname` The name of the CGI program currently being run, relative to the directory in which it is (such as `save_foo.cgi`).

`$remote_user` The username with which the current user logged into Webmin.

`$base_remote_user` The username whose permissions are currently in effect. Most of the time this will be the same as `$remote_user`, but if you have the **Configure UNIX user authentication** option set up in the Webmin Users module, this will be set to the name of the user whose permissions are used.

`$current_theme` The name of theme in effect for the current user.

`$root_directory` The root directory of the Webmin install, such as `/usr/libexec/webmin`.

`$module_root_directory` The root directory of the current module, such as `/usr/libexec/webmin/modulename`.

`%module_info` Information about the current module from its `module.info` file.

`$current_lang` The short code for the language currently being used, such as `en` or `de`.

`%current_lang_info` A hash containing information about the current language, with keys like `desc` for the language name and `titles` indicating whether graphical titles can be used. Mostly useful to theme developers.

3. The `list_users` function

This is an example of a function that might be used by various CGI programs in this module. Some module library files may also include another file containing functions specific to the current operating system or configuration. The `proc-lib.pl` file in the `proc` module is one example.

A CGI program called `list.cgi` in this same module might look something like:

```
#!/usr/bin/perl
# list.cgi
# Display the list of foobar users

require './foobar-lib.pl';
&header($text{'list_title'}, "");
print "<hr>\n";

print "<table border>\n";
print "<tr $tb>\n";
print "<td><b>$text{'list_user'}</b></td>\n";
print "<td><b>$text{'list_real'}</b></td>\n";
print "</tr>\n";

@users = &list_users();

foreach $u (@users) {
    print "<tr $cb>\n";
    print "<td><a href='edit.cgi?user=",
          "&urlize($u->{'user'}) , ''>$u->{'user'}</a></td>\n";
    print "<td>$u->{'real'}</td>\n";
    print "</tr>\n";
}

print "</table>\n";
print "<hr>\n";
&footer("", $text{'index_return'});
```

The important lines in this example are:

1. `#!/usr/bin/perl`

All CGI programs must start with a `#!` line containing the path to the Perl interpreter on your system. This should be the same as the path that Webmin itself uses, found in the `/etc/webmin/perl-path` file.

2. `require './foobar-lib.pl';`

CGI programs should include the module's library with a line like this, so that `init_config` is called and functions defined in the library are available.

3. `&header($text{'list_title'}, "");`

Any CGI that is going to produce HTML output should call the `header` function to produce a page title. In this case, the actual title is coming from a file in the `lang` directory that has been read into `%text`. Traditionally, a horizontal line is generated directly after the header as well, as in this example.

Only programs that are going to later call `redirect` should not call `header`, or produce any HTML with `print` statements.

4. The five lines starting with `print "<table border>\n";`

These lines output the HTML for the header of the table that the CGI is going to generate.

5. `@users = &list_users();`

This line is a call to the `list_users` function defined in `foobar-lib.pl`, which presumably returns an array of users.

6. The seven lines starting with `foreach $u (@users) {`

This loop creates the table rows, each of which contains a link to another CGI program in the module. You should note the use of the `urlize` function to convert the username into a form suitable for a URL parameter.

7. `print "</table>\n";`

These lines produce the HTML for the end of the table and the traditional final horizontal line.

8. `&footer("", $text{'index_return'});`

Every CGI program that calls `header` must call `footer` as well, which generates the HTML needed to properly finish the page.

The corresponding parts of the `lang/en` file for this CGI program might look like:

```
list_title=Foobar User List
list_user=Username
list_real=Real name
```

All modules that use internationalization must include a `lang/en` file, and can also include other files in `lang` for other languages. Of course, there is no requirement that you actually make use of Webmin's internationalization features. You can just put hard-coded text strings into the code instead.

55.4 Module Configuration

Almost all modules have a set of configuration parameters, available to module CGI programs in the `%config` array that is set by the `init_config` function. When Webmin or a module is installed, a configuration file appropriate for the chosen operating system is copied from the module directory to the Webmin configuration directory for that module, typically something like `/etc/webmin/foobar/config`.

The `%gconfig` associative array contains global configuration options, typically from the file `/etc/webmin/config`. Some useful entries are:

`os_type` The operating system type selected in `setup.sh`, or automatically at install time, such as `solaris` or `redhat-linux`.

`os_version` The operating system version selected at installation, such as `2.5` or `5.1`

`path` The UNIX path for this operating system, as a colon separated list of directories. This is also available in `$ENV{'PATH'}`, and therefore to any programs that your module runs.

Many modules deal with the configuration of some service that is mostly the same on different operating systems. Apache, for example, works exactly the same under Solaris and Red Hat Linux—the only difference is the standard location of the Apache configuration files. In the Webmin Apache module, the Apache config file directory is a configurable parameter itself that is initially set based on the chosen operating system.

Configuration parameters can also be used for options that the user may want to occasionally change. For example, the BIND module has a parameter that controls for format of new DNS zone serial numbers. When the fourth parameter of the header function is set, a link will be generated to a CGI program that allows the user to edit the configuration of the current module. This program reads the config.info file in the module directory to determine the possible values for each configuration parameter. A typical config.info file might look like:

```
foobar_path=Path to foobar config file,0  
display_mode=Index page display mode,1,0-Long,1-Medium,2-Short  
password_file=Foobar server users file,3,None  
file_user=Config files are owned by user,5
```

Each line is in the *config_name*=*description*,*type*[,*values*] format. The meanings of each line part are:

config_name The name of a parameter in the module configuration to which this line will apply.

description A description of this parameter for the user.

type A number that determines how this parameter can be chosen. Possible values are:

0 Free text. Any value can be entered by the user.

1 One of many. The user can choose one of several options. For this type, the *values* part of the line is a comma-separated list of *value*-*display* pairs. The *value* part of each pair is what gets stored in the config file, while the *display* part is what is shown to the user.

2 Many of many. The user can choose zero or several more options. Available options are specified in the same way as type 2.

3 Optional free text. The user can either select the default option or enter some value. The *values* part of the line is the description of the default option (typically something like *None* or *Default mode*).

4 One of many. The same as type 1, but uses a menu instead of a row of radio buttons.

5 UNIX user. Displays a list of users from the host on which Webmin is running.

6 UNIX group. Displays a list of groups from the host on which Webmin is running.

7 Directory. Like the free text input, but with a directory chooser next to it.

8 File. Like the free text input, but with a file chooser next to it.

9 Multiline free text. The first *value* after the type is the width of the input, and the second *value* is the height.

10 Like type 1, but with an additional option for entering free text of the user's choice.

11 A parameter of this type does not allow the user to enter anything, but instead puts a section header row containing the description into the configuration form at this point.

12 A field for entering a password, without actually displaying the current value.

13 Like type 2, but displays a list box for selecting multiple options instead of a row of checkboxes.

Not every configurable parameter needs an entry in `config.info`—only those that the user may want to edit.

When a module is installed (either as part of a Webmin distribution or separately), a configuration file appropriate to the OS being used is copied from the module directory to the configuration directory (usually under `/etc/webmin`). To decide which base configuration file to use, Webmin uses the OS name and version, chosen when `setup.sh` was run at installation time, to look for the following files:

```
config-osname-osversion  
config-osname  
config
```

Where `osname` is something like `redhat-linux` or `solaris`, and `osversion` is something like `2.6` or `5.0`. A typical module might have the following configuration files:

```
config-redhat-linux  
config-redhat-linux-5.0  
config-slackware-linux  
config-debian-linux  
config-solaris
```

Webmin treats each of the Linux distributions as a different OS, as each has different locations for things like the Apache configuration file, `crontab` files, and bootup scripts. The OS version number for Linux should be the distribution version (such as `4.1` or `5.0`) rather than the kernel version.

On Linux systems, the file `config-*linux` will be used if no specific file exists for the installed Linux distribution. This can be useful if you want to create settings for all variants of Linux without having to define a file for each one individually. Actually creating such a file can be a bit tricky in most shells, however, as `*` is the wildcard character. You can just precede it with a backslash to escape it.

55.5 Look and Feel

All Webmin modules should have the same general color scheme, look, and feel, as defined by the following rules:

1. All pages should be viewable on any browser that supports images, tables, and forms. Browser features such as frames, DHTML, Javascript, or Java should not be used unless

there is no other option. It should be possible to use Webmin from browsers such as Netscape 1.1 or Pocket IE.

2. All CGI programs that generate HTML output should call the `header` function. This ensures that a standard page heading is generated according to the theme in use. The only exception is a CGI that does not want any heading at all, such as one used in a pop-up selection window.
3. The `header` function creates HTML tags setting the background, text, and link colors for the generated page based on the chosen theme and color scheme. The `init_config` function sets the global variables `$tb` and `$cb`, which should be used as the background color for table headers and body rows, respectively, as in this example:

```
print "<table border width=100%>\n";
print "<tr $tb> <td><b>Foo</b></td> <td><b>Bar</b></td> </tr>\n";
print "<tr $cb> <td>some value</td> <td>another value</td> </tr>\n";
```

4. Try to avoid generating HTML forms that contain a large number of input fields. Some browsers (particularly Netscape on UNIX) slow down when rendering such pages.
5. Your module's main page (usually `index.cgi`) should set the `config` and `noindex` parameters of the `header` function to 1, assuming that the module does have a `config.info` file. This ensures that a **Module Config** link appears on the main page and that the **Module Index** link does not.
6. Other pages in the module should call `header` with the `title` parameter set to the title that you want to appear and the `image` parameter set to an empty string. This ensures that a conventional title is displayed and that a **Module Config** link back to the main page appears.

55.6 Design Goals

A typical Webmin module is written to configure some UNIX services, such as Apache, Squid, or NFS exports. Most UNIX servers are normally configured by editing a specific text file, which may have a complex format. Any Webmin module that modifies a configuration file must be able to parse all possible options in such a configuration file, even if not all options are presented to the user.

No module should ever corrupt a service configuration file or remove options that it does not understand. Modules should be able to parse any valid configuration without requiring special comments or a special format. If your module cannot deal with some option in a file, it should be left alone.

Webmin modules should be designed to be easy for novices to use but still allow the user to do almost everything that could be done by editing the configuration file directly. In some cases, however, configuration options will exist that very few users will need to edit, or that do not lend themselves to be edited through a GUI. These kind of settings should be left out of your Webmin module because they will clutter up the user interface with their presence.

55.7 Online Help

Webmin versions 0.63 and greater have support for context-sensitive help. The `hlink` function generates HTML for a link to a CGI program that processes and displays a given help page. Help pages are stored in the `help` subdirectory under the module directory, and are named simply `page.html` for those in English. So, a call to `hlink` like

```
print &hlink("Enter username:", "name"),
    "<input name=username size=20><p>\n";
```

would output a link to display the help page `help/name.html`.

If the `help` parameter to the `header` function is set, a link labeled **Help** that is directed to the specified help page is included in the heading. This can be useful if you have created specific documentation that explains what the entire page does in general, instead of (or as well as) documenting fields individually. The same rules about HTML help file selection apply.

Even though online help is not mandatory (or even common) in Webmin modules, it can be useful to provide additional information to users about what a field really means or what the purpose of a page is. In many cases, inputs are not self-explanatory and need additional documentation, so why not make it available from the page itself?

As Section 56.3 “Internationalization” explains, Webmin modules can support multiple languages through the use of alternative translation files in the `lang` subdirectory. Help pages can exist in more than one language as well by creating files named like `page.language.html` in the `help` subdirectory. If such a file exists, it will be used in preference to `page.html`, which is assumed to be in English. To add a German version of an existing `name.html` page, for example, you would need to create `name.de.html`.

55.8 Module Packaging

The Webmin Configuration module allows the user to add a new module to their existing setup. Modules must be packaged as an optionally compressed UNIX TAR file containing one or more modules. Each module in the TAR file must have all its files in one subdirectory. For example, a module TAR file might look like:

```
drwxrwxr-x 3001/10      0 Aug 12 21:53 1998 dfsadmin/
drwxrwxr-x 3001/10      0 Nov  7 01:10 1997 dfsadmin/images/
-rw-rw-r-- 3001/10     245 Aug  1 23:41 1998 dfsadmin/images/icon.gif
-rw-rw-r-- 3001/10    1438 Aug  1 23:41 1998 dfsadmin/images/dfsadmin.gif
-rw-rw-r-- 3001/10    1541 Aug  1 23:41 1998 dfsadmin/images/create_share.gif
-rw-rw-r-- 3001/10   1265 Aug  1 23:41 1998 dfsadmin/images/edit_share.gif
drwxrwxr-x 3001/10      0 May 16 18:32 1997 dfsadmin/test/
-rw-r--r-- 3001/10    493 May 16 18:32 1997 dfsadmin/test/dfstab
-rw-rw-r-- 3001/10   2774 Jul 29 22:22 1998 dfsadmin/dfs-lib.pl
-rw-rw-r-- 3001/10      49 Aug 12 21:53 1998 dfsadmin/module.info
-rwxr-xr-x 3001/10   1596 Mar 31 15:45 1998 dfsadmin/index.cgi
-rw-rw-r-- 3001/10     199 Mar  5 19:30 1998 dfsadmin/help.html
-rw-rw-r-- 3001/10     175 Mar  5 19:30 1998 dfsadmin/config.info
-rw-r--r-- 3001/10     140 Mar  5 19:30 1998 dfsadmin/config-solaris
-rwxr-xr-x 3001/10     142 Mar  5 19:30 1998 dfsadmin/delete_share.cgi
-rwxr-xr-x 3001/10    4842 Mar  5 19:30 1998 dfsadmin/edit_share.cgi
-rwxr-xr-x 3001/10    3000 Mar  5 19:30 1998 dfsadmin/save_share.cgi
-rwxr-xr-x 3001/10     573 Jun  8 15:02 1998 dfsadmin/restart_sharing.cgi
```

The standard extension for Webmin modules is `.wbm`, but any filename can be used. If you compress the module, it can have a `.wbm.gz` or `.wbm.Z` extension instead.

Webmin modules can also be packaged as RPMs, which are suitable for installing on servers on which the RPM version of Webmin itself is already installed. You can download a script called `makemodulerpm.pl` from www.webmin.com/makemodulerpm.pl that can package up a module directory into an RPM by creating the spec file automatically.

55.9 Summary and Learning More

The information and examples in this chapter should be all you need to know to start writing your own modules. Chapter 56 explains some of the more advanced possibilities in module development, such as logging, locking, and access control. Chapter 60 lists all of the functions that are available to module writers in `web-lib.pl`, which can be used to give your module a similar look and feel to standard modules, simplify the process of editing configuration files, and much more.

The best source of information about module development is the source code of Webmin itself. Have a look at it to see how the standard modules are written and read Chapter 57, which takes you through the Scheduled Cron Jobs module.

For information on creating themes, which are similar to modules in many ways, read Chapters 58 and 59. These cover the basics of theme creation and the implementation of the default MSC theme, respectively.

Advanced Module Development

This chapter carries on from Chapter 55, and explains some of the more advanced parts of module development such as access control, logging, and integration with the Users and Groups module.

56.1 Module Access Control

Webmin versions 0.72 and above support a standard method for restricting which features of a module a user can access. For example, the Apache module allows a Webmin user to be restricted to managing selected virtual servers, and the BIND module allows a user to be limited to editing records only in certain domains.

This kind of detailed access control is separate from the first level ACLs that control which users have access to which modules. As long as your module calls `init_config`, the Webmin API will automatically block users who do not have access to the entire module.

Module access control options are set in the Webmin Users module by clicking on the name of a module next to a user's name. The available options are generated by code from the module itself (except for the **Can edit module configuration?** option, which is always present). When the user clicks on **Save**, the form parameters are also parsed by code from the module being configured before being saved in the Webmin configuration directory.

A module wanting to use access control must contain a file called `acl_security.pl` in its directory. This file must contain two Perl functions:

```
acl_security_form(acl) This function takes a reference to a hash containing  
the current ACL options for this user and must output HTML for form inputs to edit  
those ACL options. Because the HTML will be inside a 4-column table, you must  
generate the appropriate <tr> and <td> tags around your input elements.
```

`acl_security_save(acl, inputs)` This function must fill in the given hash reference with values from the form created by `acl_security_form`. Form inputs are available in the global hash `%in` as generated by `ReadParse`, or from the second parameter to the function.

Because these functions are called in the context of your module, the `acl_security.pl` file can *require* the common functions file used by other CGI programs in the module. This gives you access to all the standard Webmin functions and allows you to provide more meaningful inputs. For example, when setting ACL options for the Apache module, a list of virtual servers from the Apache configuration is displayed for the user to choose from.

An example `acl_security.pl` might look like:

```
do 'foo-lib.pl';

sub acl_security_form
{
print "<tr> <td><b>Can edit users?</b></td>\n";
printf "<td><input type=radio name=edit value=1 %s> Yes\n",
      $_[0]->{'edit'} ? 'checked': '';
printf "<input type=radio name=edit value=0 %s> No</td> </tr>\n",
      $_[0]->{'edit'} ? '' : 'checked';
}

sub acl_security_save
{
$_[0]->{'edit'} = $in{'edit'};
}
```

If a user has not yet had any ACL options set for a module, a default set of options will be used. These are read from the `defaultacl` file in the module directory, which must contain `name=value` pairs, one per line. These options should allow the user to do anything so the admin or master Webmin user is not restricted by default.

To actually enforce the chosen ACL options for each user, your module programs must use the `get_module_acl` function to get the ACL for the current user and then verify that each action is allowed. When called with no parameters, this function will return a hash containing the options set for the current user in the current module, which is almost always what you want. For example:

```
#!/usr/local/bin/perl
require 'foo-lib.pl';
%access = &get_module_acl();
$access{'create'} ||
  &error("You are not allowed to create new foo users");
```

When designing a module to which some users will have limited access, remember that the user can enter **any** URL, not just those to which you link. For example, just doing ACL checking in the program that displays a form is not enough. The program that processes the form should do

all the same checks as well. CGI parameters should also never be trusted, even hidden parameters that cannot normally be input by the user.

56.2 User Update Notification

Since version 0.72, it has been possible to have the Users and Groups module notify other modules when a UNIX user is added, updated, or deleted. This can be useful if your module deals with additional information that is associated with users. For example, the Disk Quotas module sets default quotas when new users are created, and the Samba Windows File Sharing module keeps the Samba password file in sync with the UNIX user list.

To have your module notified when a user is added, updated, or deleted, you must create a Perl script called `useradmin_update.pl` in your module directory. This file must contain three functions:

`useradmin_create_user(user)` This function is called when a new UNIX user is created. The `user` parameter is a hash containing the details of the new user, described in more detail below.

`useradmin_modify_user(user, olduser)` This function is called when an existing UNIX user is modified in any way. The `user` parameter is a hash containing the new details of the user, and `olduser` contains the details of the user before he was modified.

`useradmin_delete_user(user)` This function is called when a UNIX user is deleted. Like the other functions, the `user` hash contains the user's details.

The hash reference passed to each of the three functions has the keys shown in Table 56.1.

If the system has shadow passwords enabled, other keys may also be available—it is not a good idea to rely on them.

When your functions are called, they will be in the context of your module. This means that your `useradmin_update.pl` script can require the file of common functions used by other CGI programs. The functions can perform any action you like to update other configuration files or whatever, but should not generate any output on `STDOUT` or take too long to execute. A partial example `useradmin_update.pl` might look like:

```
do 'foo-lib.pl';

sub useradmin_create_user
{
local $lref = &read_file_lines($users_file);
push(@$lref, "$_[0]->{'user'}:$_[0]->{'pass'}");
&flush_file_lines();
}
```

56.3 Internationalization

Webmin versions 0.75 and above provide module writers with functions for generating different text and messages, depending on the language selected by the user. Each module that wishes to use this feature should have a subdirectory called `lang` that contains a translation file for each

Table 56.1 Keys in the Hash Passed to the `Usermin_update.pl` Functions

user	The login name of the new or modified user
uid	The UNIX UID of the user
gid	The UNIX GID for the user's primary group
pass	The user's password, encrypted with the <code>crypt</code> function
plainpass	The user's password in plain text. This is only available when the <code>passmode</code> key is equal to 3.
passmode	This number depends on the choice made for the Password field in the create user or edit user form. 0 - No password is set for this user, typically meaning that no password is required to log in 1 - This user is not allowed to log in at all 2 - Only the encrypted password for this user is available 3 - A new or initial plain-text password is available in the <code>plainpass</code> key 4 - The user's password is unchanged. Only possible when <code>useradmin_modify_user</code> is called.
real	The user's real name
home	The user's home directory
shell	The user's login shell

supported language. Each line of a translation file defines a message in that language in the format `message_code=Message in this language`.

The default language for Webmin is English (code `en`), so every module should have at least a file called `lang/en`. If any other language is missing a message, the English one will be used instead. Check the file `lang_list.txt` for all the languages currently supported and their codes. To change the current language, go into the Webmin Configuration module and click on the **Language** icon.

When your module calls the `init_config` function, all the messages from the appropriate translation file will be read into the `%text` hash. Therefore, instead of generating hard-coded text like this:

```
print "Click here to start the server:<p>\n";
```

your module should use the `%text` hash like so:

```
print $text{'startmsg'}, "<p>\n";
```

Messages from the appropriate file in the top-level `lang` directory are also included in `%text`. Several useful messages such as `save`, `delete`, and `create` are therefore available to every module.

In some cases, you may want to include some variable text in a message. Because the position of the variable may differ depending on the language used, message strings can include place markers like `$1`, `$2`, or `$3`. The `text` function should be used to replace these place markers with actual values like:

```
print &text('servercount', $count), "<p>\n";
```

Your module's `module.info` file can also support multiple languages by adding a line like `desc_code=module description` for each language, where `code` is the language code. You can also have a separate `config.info` file for each language—called `config.info.code`—and separate help files for each language, such as `intro.code.html`. In all cases, if there is no translation for the user's chosen language then the default (English) will be used instead.

56.4 File Locking

Webmin version 0.81 introduced several new common functions for locking files to prevent multiple programs from writing to them at the same time. Module programmers should make use of these functions to prevent the corruption or overwriting of configuration files in cases where two users are using the same module at the same time.

Locking is done by the `lock_file` function, which takes the name of a file as a parameter and obtains an exclusive lock on that file by creating a file with the same name but with `.lock` appended. The `unlock_file` function also removes the lock on the file given as a parameter. Because the `.lock` file stores the PID of the process that locked the file, any locks a CGI program holds will be automatically released when it exits. It is recommended, however, that locks be properly released by calling `unlock_file` or `unlock_all_files` before exiting.

The following code shows how the locking functions might be used:

```
&lock_file("/etc/something.conf");
open(CONF, ">>/etc/something.conf");
print CONF "some new directive\n";
close(CONF);
&unlock_file("/etc/something.conf");
```

Locking should be done as soon as possible in the CGI program—ideally before reading the file to be changed and definitely before writing to it. Files can and should be locked during creation and deletion, as should directories and symbolic links before creation or removal. While this is not really necessary to prevent file corruption, it does make the logging of file changes performed by the program more complete, as explained below.

Many other programs also use `.lock` files for the same purpose, but most do not put their process ID in the file. If the `lock_file` function encounters a lock like this, it will wait until it is completely removed before obtaining its own lock, as there is no way to tell if the original process is still running or not.

56.5 Action Logging

Webmin versions 0.81 and above have support for detailed logging by CGI programs of the actions performed by users for later viewing in the Webmin Actions Log module. While previous versions wrote an HTTP log file to `/var/webmin/miniserv.log`, this did not contain the information required to work out exactly what each Webmin user had been doing. To improve on this, Webmin now logs detailed information to the file `/var/webmin/webmin.log` and optionally to files in the `/var/webmin/diffs` directory. Note that nothing will be recorded in this file if logging is not enabled in the Webmin Configuration module.

CGI programs should call the `webmin_log` function after they have successfully completed all processing and file updates. The parameters taken by the function are:

action The action the program has performed. Usually something like `save` or `delete`.

type The type of thing affected by the program. Often something like `user` or `group`, although it can be left blank if not appropriate.

object The name of the thing affected, such as `jcameron` or `root` or `www.foobar.com`.

parameters A reference to a hash containing additional information that the program wants to log. Often just passing `\%in` is useful.

All of these parameters can contain any information you want, as they are merely logged to the actions log file and not interpreted by `webmin_log` in any way. For example, a module might call the function like this:

```
&lock_file("/etc/foo.users");
open(USERRS, ">>/etc/foo.users");
print USERRS "$in{'username'} $in{'password'}\n";
close(USERRS);
&unlock_file("/etc/foo.users");
&webmin_log("create", "user", $in{'username'}, \%in);
```

Because the raw log files are not easy to understand, Webmin also provides support for converting detailed action logs into human-readable format. The Webmin Actions Log module makes use of a Perl function in the `log_parser.pl` file in each module's subdirectory to convert log records from that module into a readable message.

This file must contain the `parse_webmin_log` function, which is called once for each log record for this module. It will be called with the following parameters:

user The Webmin user who runs the program that generates the log record.

script The filename of the CGI script that generated this log, without the directory.

action Whatever was passed as the `action` parameter to `webmin_log` to create this log record.

type Whatever was passed as the `type` parameter to `webmin_log`.

object Whatever was passed as the `object` parameter to `webmin_log`.

parameters A reference to a hash, the same as the one passed to `webmin_log`.

long If non-zero, this indicates that the function is being called to create the description for the Action Details page and thus can return a longer message than normal. You can ignore this if you like.

The function should return a text string based on the parameters passed to it that converts them into a readable description for the user. For example, your `log_parser.pl` file might look like:

```
require 'foo-lib.pl';

sub parse_webmin_log
{
    local ($user, $script, $action, $type, $object, $params, $long) = @_;
    if ($action eq 'create') {
        return &text('log_create', $user);
    }
    elsif ($action eq 'delete') {
        return &text('log_delete', $user);
    }
}
```

Because the `log_parser.pl` file is read and executed in a similar way to how the `acl_security.pl` file is handled by the Webmin Users module, it can require the module's own library of functions just like any module CGI program would. This means that the `text` function and `%text` hash are available for accessing the module's translated text strings, as in the example above.

Webmin can also be configured to record exactly what file changes are made by each CGI program before calling `webmin_log`. Under **Logging** in the Webmin Configuration module is a checkbox labeled **Log changes made to files by each action** that, when enabled, will cause the `webmin_log` function to use the `diff` command to find changes made to any file locked by each program.

When logging of file changes is enabled, the action details page in the Actions Log module will show the `diffs` for all file updates, creations, and deletions by the chosen action. If locking of directories and symbolic links is done as well, it will show their creations and modifications, too.

As well as having their file changes logged, programs can also use the common `system_logged`, `kill_logged`, and `rename_logged` functions that take the same parameters as the Perl `system`, `kill`, and `rename` functions but also record the event for viewing on the action details page. There is also a `backquote_logged` function that works in a similar fashion to the Perl backquote operator (it takes a command and executes it, returning the output), but also logs the command. If these functions are used, they must be called before `webmin_log` for the logging to actually be recorded, as in this example:

```
if ($pid) {
    &kill_logged('TERM', $pid);
}
else {
```

```
&system_logged( "/etc/init.d/foo stop");
}
&webmin_log( "stop");
```

56.6 Pre- and Post-Install Scripts

Webmin versions 0.990 and above allow modules to contain Perl scripts that will be run after a module is installed and before it is uninstalled. If your module contains a file called `postinstall.pl`, the Perl function `module_install` in this file will be called after the install of your module is complete. Because it is executed in the module's directory, it can make use of the common functions library, in the following way:

```
require 'yourmodule-lib.pl';

sub module_install
{
system("cp $module_root_directory/somefile $config_directory/somefile")
      if (!-r "$config_directory/somefile");
}
```

The function will be called when a module is installed from the Webmin Configuration or Cluster Webmin Servers modules. It is not called, however, if the `install-module.pl` script is used or when the module in RPM format is installed.

If your module contains a file called `uninstall.pl`, the Perl function `module_uninstall` in that file will also be called just before the module is deleted. This can happen when it is deleted, using the Webmin Users or Cluster Webmin Servers modules, or when all of Webmin is uninstalled. The `module_uninstall` function should clean up any configuration that will no longer work when the module is uninstalled, such as Cron jobs that reference scripts in the module.

56.7 Functions in Other Modules

The standard Webmin modules contain a vast number of useful functions for parsing and manipulating the configuration files for Apache, BIND, UNIX Users, and so on. If your module needs to configure these servers as well in some way, it makes sense to make use of existing functions in the standard modules.

Because the standard modules have typically already been configured with the correct paths for files like `httpd.conf` and `squid.conf`, their functions will use those paths when you call them to read and write configuration files. The actual `%config` settings for another module can also be accessed so your module knows what commands to use to apply changes or to start some server like Apache or Squid.

When you first load the library for some other module with the `foreign_require` function, it is actually executed in a separate Perl module namespace. All of your module's CGI programs and its library will be in the main namespace, but other foreign module's functions will be put in a namespace with the same name as the Webmin module. This means that you can call those functions with code like `&useradmin::list_users()` and access global variables like `$useradmin::config{ 'passwd_file' }`. This Perl namespace separation ensures that func-

tions and global variables with the same names can exist in both your module and the foreign module, without any clashes. Some things are shared between all modules, however, such as caches used by `get_system_hostname`, `load_language`, `read_file_cached`, and `get_all_module_infos` so loading the library of a new module with `foreign_require` is not too slow.

The only way to find out which functions are available in other modules is to read the source code of their CGI programs and libraries. Unfortunately, there is no documentation at the moment as to what the functions do, apart from maybe some comments in the source code. You will also find that not every feature of another module is accessible through its functions. For example, the Apache module does not have a function for applying the current configuration. Instead, this is done by the `restart.cgi` script, which runs the appropriate commands directly.

Probably the most useful module for others is Running Processes, as it contains several functions for starting processes in the background. Many of the standard modules make use of them, which is why Running Processes should never be uninstalled unless you want to break several other Webmin modules as well.

The functions available after calling `&foreign_require("proc", "proc-lib.pl");` are:

```
proc::safe_process_exec(command, uid, gid, handle, input,
fixtags, bsmode) Executes the specified command as the given UNIX uid
and gid, and writes its output to a Perl file handle (usually STDOUT). This can be
useful when executing programs that fork their own subprocesses, which would
normally prevent Perl from detecting the end of their output when run using a piped
open function call. For example, many servers have startup scripts like /etc/
init.d/httpd that exhibit this problematic behavior.
```

If the `input` parameter is supplied, it will be fed as input to the command when run. If the `fixtags` parameter is set to 1, all output from the command will have HTML characters escaped so it can be properly displayed in a browser. And if the `bsmode` argument is set, any backspace or return characters output by the command are interpreted to remove the last output letter or clear the current line, respectively. This can be useful if the program usually uses these characters to display an incrementing counter, as `mkfs` and `cdrecord` do.

```
proc::safe_process_exec_logged(command, uid, gid, handle, input,
fixtags, bsmode) This function behaves just like safe_process_exec, but
also records the executed command so it will be logged when webmin_log is called.
```

```
proc::pty_process_exec(command, [uid, gid]) Executes the specified
command in a new PTY as either the given UNIX uid and gid or the correct user
(normally root). Many programs (such as passwd) expect to interact with a user
through the current TTY and thus input cannot simply be piped to them after being
run with the open function. You can, however, use the file handle that this function
returns to read and write to such programs as though you were supplying user input
and viewing output usually sent to the user.
```

`pty_process_exec` actually returns both a file handle and the PID of the started process so you can kill it if necessary. The standard `wait_for` function can be used to interact with the command, as the following example shows:

```

&foreign_require("proc", "proc-lib.pl");
($fh, $pid) = &proc::pty_process_exec("passwd $username");
while(1) {
    $rv = &wait_for($fh, "password:");
    if ($rv == 0) {
        &sysprint($fh, $password);
    }
    else {
        last;
    }
}
close($fh);

proc::pty_process_exec_logged(command, [uid, gid]) Just like
pty_process_exec, but records the command for later logging when
webmin_log is called.

proc::list_processes([pid]) Returns a list of all processes currently
running on the system or just the details of a single process if the pid parameter is
supplied. Each element of the returned array is a reference to a hash containing at
least the keys shown in Table 56.2.

proc::find_process(name) Searches for processes whose command or
arguments match the given name and returns an array of details for those that match.
Each element is a hash reference with the same members as the list_processes
function.

```

56.8 Remote Procedure Calls

Webmin versions 0.82 and above have several common functions for executing code on remote Webmin servers. They are used by some of the standard modules (such as those in the Cluster category) to control multiple servers from a single interface and may be useful in your own modules as well. These functions, all of which have names starting with `remote_`, let you call functions, evaluate Perl code, and transfer data to and from other systems running Webmin.

Before a *master* server can make RPC calls to a remote host, it must be registered in the Webmin Servers Index module on the master system. The **Link type** field must be set to **Login via Webmin** and a username and password must be entered. The specified user should be `root` or `admin`, as others are not, by default, allowed to accept RPC calls.

RPC is usually used to call functions in other modules on a remote system or to call common functions. This is done with the `remote_foreign_call` function, but before it can be used, `remote_foreign_require` must be called to load the library for the module that you want to call. This is very similar to calling functions in other local modules with the `foreign_` functions, as explained in Section 56.7 “Functions in Other Modules”.

A piece of code that edits a user on a remote system might look like the following:

```

$server = "www.example.com";
$user = "joe";
&remote_foreign_require($server, "useradmin", "user-lib.pl");
@users = &remote_foreign_call($server, "useradmin", "list_users");

```

Table 56.2 Keys in Each Process Information Hash

pid	The process's ID
ppid	The ID of its parent process
user	The name of the UNIX user who owns this process
cpu	The percentage of CPU time that the process is currently using
size	The amount of memory that the process is using
nice	Its current nice level
time	How long the process has been actually using the CPU
args	The process's command and any arguments

```

($joe) = grep { $__->{'user'} eq $user } @users;
if ($joe) {
    $joe->{'real'} = "Joe Bloggs";
    &remote_foreign_call($server, "useradmin", "modify_user", $joe, $joe);
}

```

Of course, you need to be familiar with the available functions in other modules and be sure that the module that you want to call is actually installed and is the right version.

All parameters passed to remote functions are converted to a serialized text form for transfer to the remote server and any return value is also sent back in serialized form. The common functions `serialize_variable` and `unserialize_variable` are used, but the process is hidden from both the caller and the remote function. They only see scalars and references in their original format. One thing to look out for is circular references, however. Trying to send a structure that contains links to itself (such as a doubly-linked list) will fail due to the shortcomings of the `serialize_variable` function. Also, try to avoid using extremely large parameters, such as strings over 1 MB in size, as serialization may make them massive.

Parameters that are references to hashes, arrays, or scalars that would normally be filled in by the function will not be transferred properly. For example, the `read_file` function normally fills in the hash referenced by its second argument with the contents of a file. This will not work when it is called remotely, as all parameters and anything that they refer to are *copied* to the other system.

The `remote_eval` function can be used to execute an arbitrary block of Perl code on a remote system, which allows you to do things that calls to remote functions cannot. It is the only way to call native Perl functions, such as `unlink`, to read and write arbitrary format files, set glo-

bal variables, and properly call functions that set their parameters. Whatever the Perl code evaluates will be returned by this function. This example shows `remote_eval` in use:

```
$data = &remote_eval($server, "useradmin",
"rename('/etc/foo', '/etc/bar');\n".
"local \%data;\n".
"&read_file('/etc/bar', \$%data);\n".
"return \$%data;\n");
&write_file('/etc/foo', $data);
```

As you can see, proper quoting is necessary when constructing the Perl code string so any variable symbols (such as \$, %, and @) are escape, as is the \ character. The second *module* parameter to `remote_eval` can be set to `undef`, which indicates that the code should be executed in the global Webmin context, rather than that of any module's.

The functions `remote_read` and `remote_write` can be used to transfer the contents of an entire file between the master and remote systems. They are much faster than reading in the file and encoding it for use in the `remote_foreign_call` or `remote_eval` functions, as the file is transferred unencoded over a TCP connection.

If your module makes RPC calls, you may want the user to select a system to make calls to or from a menu. A list of the names of all those available can be obtained from the Webmin Servers Index module with code such as the following:

```
&foreign_require("servers", "servers-lib.pl");
@allservers = &servers::list_servers();
@rpcservers = map { $_->{'host'} } grep { $_->{'user'} } @allservers;
```

In addition, all of the `remote_` functions will accept `undef` for the *server* parameter. This indicates that the local system should be used, which never needs to be defined in the Webmin Servers Index module. This is how all of the Cluster category modules can include the **this server** option in their lists of hosts to manage.

56.9 Creating Usermin Modules

Usermin has a very similar architecture to Webmin, so its modules have an almost identical design to Webmin modules. The main difference is that Usermin is designed to be used by any UNIX user on a server to perform tasks that they could perform from the command line. Any third-party Usermin modules should be written with this in mind.

By default, module CGI programs are run as `root`, just like in Webmin. This is necessary because some tasks (like changing passwords) can only be done as `root`. Most Usermin modules, however, do not need super-user privileges and so should call the standard `switch_to_remote_user` function just after calling `init_config` in order to lower privileges to those of the logged-in user.

Usermin modules can have global configuration variables that are initially set from the `config` or `config-ostype` file in the module directory, and are available in `%config`. These variables, however, are never editable by the user. They can only be set in the Usermin Configuration module of Webmin.

Per-user configurable options are supported, however, using a different mechanism. When the standard `create_user_config_dirs` function is called, the global `%userconfig` hash will be filled with values from the following sources, with later sources overriding earlier ones:

`defaultuconfig` file in the module directory This should contain the default options for this module for all users, to be used if no other settings are made by the user or system administrator.

`/etc/webmin/modulename/defaultuconfig` file This contains defaults for the module on this system, as set by the system administrator using the second form on the **Usermin module configuration** page in the Usermin Configuration Webmin module.

`~username/.usermin/modulename/config` file This contains options chosen by users themselves.

The `uconfig.info` file in the module directory defines the editors for the system-wide and per-user configuration variables. This file has the exact same format as the `config.info` file used for Webmin and Usermin global configuration, as explained elsewhere in this document.

If you create your own Usermin module, it should be packaged in exactly the same way as a Webmin module (as a `.tar` or `.tar.gz` file). The `module.info` file must contain the `usermin=1` line so that it cannot be installed in Webmin (where it would not work properly).

56.10 Summary

This chapter has been a continuation of Chapter 55 by explaining how to make use of some of the more advanced features available to Webmin module developers. It has covered access control, internationalization, logging, and the use of RPC to call functions on other servers running Webmin. It has also documented some of the functions in the Running Processes module, which can be used by other modules wanting to start or manage processes.

Inside the Scheduled Cron Jobs Module

This chapter takes module writers inside one of the standard Webmin modules and explains which parts of its design they should copy.

57.1 Module Design and CGI Programs

As Chapter 10 explains, this module lets a user view, edit, and create Cron jobs for all UNIX users on a system. It gets the lists of jobs by reading several different files, such as those in the `/var/spool/cron` directory, those in `/etc/cron.d`, and those in `/etc/crontab`. The exact paths depend upon the operating system on which Webmin is running, as every UNIX variant seems to have its own implementation of Cron.

In addition to editing jobs, the module can also be used to execute those that have already been defined and view their output. Users can also edit the files that control which users have access to Cron, usually named `/etc/cron.allow` and `/etc/cron.deny`.

The CGI programs that make up this module are:

`index.cgi` Displays a list of jobs that the current Webmin user is allowed to access, each of which is a link to the editing page created by `edit_cron.cgi` with a parameter identifying the index of the job to edit. The actual list comes from the `list_cron_jobs` function in `cron-lib.pl`.

`edit_cron.cgi` Produces HTML for a form for either editing an existing job or creating a new one, depending on the `idx` and `new` parameters. Again, the details of a job being edited are taken from `idx`.

The `list_cron_jobs` function At the bottom of the generated page are buttons that submit to either `save_cron.cgi` or `delete_cron.cgi`.

`save_cron.cgi` Calls `ReadParse` to get the form inputs from `edit_cron.cgi` and validates them to make sure all of the required fields have been filled. If so,

functions from `cron-lib.pl` are called to either create a new job or update an existing job and then a redirect is called to make the user's browser return to `index.cgi`. If an error is detected, however, the standard `error` function is called instead.

When changing the user that a job runs as, this program needs to delete and recreate it so that it ends up in the right file, instead of just changing it in place.

`delete_cron.cgi` Runs when the **Delete** button on the editing form is clicked. Just calls a function from `cron-lib.pl` to remove the job specified by the `idx` parameter and then redirects the browser to `index.cgi`.

`exec_cron.cgi` This CGI uses the `safe_process_exec` function from the Running Processes module to run the command for a specified Cron job as the user who owns it and displays the output. It also deletes any environment variables that are specific to Webmin, so that programs run by the Cron job do not get confused and think that they are being called as CGI programs when this is not really the case.

`edit_allow.cgi` Just displays a form for entering either a list of users who are allowed to use Cron or a list of those who cannot. The current settings are obtained by calling functions in `cron-lib.pl`.

`save_allow.cgi` Saves the inputs from the form created by `edit_allow.cgi` back to the original files, again by calling functions from the module's library.

This module follows a design common to many others—a single page listing objects to edit, each of which is a link to a form for editing. Your modules should use the same layout where appropriate, instead of displaying a huge table for editing multiple objects at once. It is a good idea to imitate this module's use of multiple CGI programs, as well, instead of trying to output everything in a single script. In all of the standard modules, a separate program generates each page and, if the page is a form, it is submitted to yet another program. This makes each one simpler and easier to understand instead of putting both the form generating and processing code into a single script. The `redirect` function is used by all of the form processing CGIs to return the user's browser to the module's main page, rather than back to the editing form.

57.2 The cron-lib.pl Library Script

The real work in this module is done by the functions in `cron-lib.pl`, which actually read and write the various Cron job files in their different formats. This is the way a Webmin module should be written, as it cleanly separates the user interface from the configuration file management. This prevents unnecessary duplication of code and makes it easy to add support for a specific new Cron file if one arises.

The functions in this library that CGI programs call are:

`list_cron_jobs()` Returns an array of hash references, each of which contains the details of a specific Cron job. This information is actually read from several different files, and each job hash contains the name of the file that it came from in the `file` key, the position of the `line` key in that file, and the format in the `type` key. This is used when the job is saved with `change_cron_job`, so that it gets put back in the same place with the correct format. Many other Webmin modules store

this kind of information in hashes that they create from configuration files, so they know which part of the file to update.

`create_cron_job(job)` Takes a hash reference containing Cron job details with the same keys as those returned by `list_cron_jobs`, which is then converted to a correctly formatted line, and appended to a temporary copy of the user's Cron jobs file. The `copy_crontab` function is used to activate it, using the method explained below.

`change_cron_job(job)` Takes a hash reference returned by `list_cron_jobs` but with some of the details updated and converts it to a correctly formatted line of text. If it is a user's personal Cron job, then the line must be updated in a copy of the jobs file. Otherwise, the original file that it came from can be updated directly.

`delete_cron_job(job)` Deletes the job passed in as a parameter by removing its line from the original file. If it was a user's personal Cron job, this is done in a temporary copy of his file instead of directly updating the original source.

`list_allowed` and `list_denied` Return arrays of users who are allowed or not allowed to access Cron, respectively. These functions are primarily used by `edit_allow.cgi`, and just read the contents of `/etc/cron.allow` and `/etc/cron.deny`. However, `save_cron.cgi` also uses them to check whether or not the user that you are creating a Cron job for can actually use it, as the `crontab` command will often fail if this is not the case.

`save_allowed(user, ...)` and `save_denied(user, ...)` These functions write the lists of users given as parameters to the `/etc/cron.allow` or `/etc/cron.deny` files, respectively. They are only used by `save_allow.cgi`.

`can_edit_user(access, user)` This function is used to check to see if the current Webmin user can access the Cron jobs of a particular UNIX user, based on the hash reference and username passed in as parameters. The reference is assumed to be the return value from `get_module_acl`, which contains settings made in the Webmin Users module. Most of the CGI programs use it to limit their displays and prevent attempts to access jobs belonging to unauthorized users. If your module has access control features that can limit what objects a user can access, a function like this is useful to prevent the duplication of code that checks ACL settings. It should be noted that it is called in both `edit_cron.cgi` and `save_cron.cgi` to block sneaky users who try to invoke the save program directly instead of going through the form.

`show_times_input(job)` This code prints HTML for the part of a form for editing the times at which a Cron job is run. It used to be in `edit_cron.cgi`, but was moved into the library so that other modules that set up Cron jobs (such as Filesystem Backup) can make use of the same inputs in their user interface.

`parse_times_input(job, in)` This function parses the inputs from the form created by `show_times_input`. Again, other modules use it as well as `save_cron.cgi`.

You might wonder why some of the functions above update a temporary file instead of directly editing the files in `/var/spool/cron` that contain user Cron jobs. The reason is that the `crontab` command must be used to install a modified file in order for the Cron daemon to notice the change and for it to take effect. This is done by the `copy_crontab` function, which invokes the appropriate `crontab` command for the operating system. Normally, when `crontab` is run by a user, it starts an editor like `vi` for the user to edit a temporary copy of the file, which is then moved back into `/var/spool/cron`.

This module sets the `EDITOR` environment variable to the `cron_editor.pl` script, however, which just copies the temporary file created by the module over the file passed to the script for editing by `crontab`. When it exists, the changes made by the module are properly installed and the temporary file can be deleted.

This process is not necessary for Cron jobs in `/etc/crontab` or `/etc/cron.d`, however, as the Cron daemon automatically detects when those files have been updated. For this reason, the `change_cron_job` and `delete_cron_job` functions can edit them directly.

Because Cron is a great tool for running scripts on a regular basis, several other modules make use of this one to set up jobs of their own. For example, Webmin Configuration uses it to schedule the automatic download of updated modules, Webalizer Logfile Analysis uses it to have logs processed regularly, and System and Server Status uses it to set up scheduled monitoring.

All of this is done by making foreign calls to the `cron` module, as explained in Section 56.7 “Functions in Other Modules”. If your module needs to do the same, it is advisable to make use of the code in `cron-lib.pl` that already supports a wide variety of operating systems and creates jobs in the correct way.

57.3 Module Configuration Settings

This module demonstrates how the various Cron file locations, formats, and programs on different operating systems can be supported by the same code. If you look in its directory, you will see numerous files with names starting with `config-`, such as `config-solaris` and `config-redhat-linux`. Each specifies the files to read and commands to use for particular operating systems. The code in `cron-lib.pl` makes numerous references to `%config` when listing and updating jobs, which, of course, is filled with the contents of `/etc/webmin/cron/config`. This file, in turn, comes from the appropriate `config-` file in the module’s directory, chosen at the time Webmin was installed.

If your module manages some service that differs slightly between operating systems, this method of using different default configurations makes sense. It can also be useful when writing a module for some server like Apache because the default configuration and program file paths will differ depending on the operating system or Linux distribution due to the vast number of different Apache packages out there.

The `config.info` file in this module defines inputs for editing both the operating system dependent options in the configuration file and those related only to the module’s user interface. Sometimes it makes very little sense to let users edit such settings as the location of users’ personal Cron job files as they are pretty much determined by the operating system in use. For this reason, you might think that taking those fields out of `config.info` is a good idea so that users cannot mess up the module’s configuration.

This will work fine, as it is really the entries in the appropriate config- file that get (indirectly) loaded into the %config hash. The config.info file just controls which ones are editable and what values are allowed—any others will be left unchanged when the user clicks on **Module Config**. In the Scheduled Cron Jobs module, however, all configuration settings can be edited, just in case the user upgrades the version of Cron that comes with his operating system to a totally different package.

57.4 The lang Internationalization Directory

Thanks to the generous contributions of Webmin users, the lang subdirectory for this module contains files for several different languages. The settings in the Language form of the Webmin Configuration module determines which one is loaded into the %text hash when init_config is called, as explained in Section 56.3 “Internationalization”.

This module uses no hard-coded text strings in any of its CGI programs or other scripts. Instead, references to an appropriate message for the current language, like \$text{'index_create'} or &text('exec_cmd'), are used. If your module might ever be translated into a different language, you should do the same in its CGI programs, as well. Even though it is slightly more work to put messages into a separate file, it is worth it in the long run.

57.5 The acl_security.pl Access Control Script

As Chapter 52 explains, the Webmin Users module can be used to configure detailed access control settings for a particular user and module. The actual form for editing these settings is generated by the acl_security.pl script in the module's directory, covered in Section 56.1 “Module Access Control”. Because this module lets an administrator define which UNIX users a particular Webmin user can edit Cron jobs for, it has one of these scripts as well.

As you can see by opening the file in an editor, it contains the required acl_security_form and acl_security_save functions. The first prints HTML for form inputs in a 4-column table, with their current settings based on the contents of the hash reference passed in as a parameter. The second function checks the values in %in and uses them to fill in the hash reference from its parameter, which is saved by the Webmin Users module back to /etc/webmin/cron/*username*.acl upon exiting.

The ACL settings for this module let the administrator choose allowed UNIX users by several different means. He can either grant access to all of them, to just the one whose name matches the current Webmin user, to a specific list of users, to users with some primary group, or to users with UIDs within some range. Many other modules have similar options to specify allowed users of some kind. If your module deals with some kind of UNIX user-related configuration, its acl_security.pl script should have similar inputs.

On many systems (such as those used for virtual hosting), a single subadministrator may be responsible for many UNIX accounts—possibly those with a certain primary group or with UIDs within some fixed range. This kind of access control makes it possible to safely give such a subadministrator a Webmin login to manage only those UNIX users that “belong” to him.

All of the CGI programs in this module use the get_module_acl standard function to get the access control settings for the current Webmin user. The return value is generally stored in the %access hash, which is consulted to determine if the Webmin user can access Cron jobs for

a particular UNIX user. This is mostly done by calling `can_edit_user` (explained above), and then calling `error` if access was denied.

Code in your module should do the same, and every CGI program should check to make sure that it is not being accessed inappropriately. One change that you might want to make is to put the call to `get_module_acl` into your module's library script so that the `%access` hash is available globally to every CGI program, instead of each of them having to call it explicitly.

When creating a module that can be set up to allow limited access like this, you must be very careful to stop the user from escaping its restrictions in any way. This means following all of the normal rules about programming CGI scripts, such as not passing user inputs directly to the `system` or `open` functions. Because a user who has full `root` privileges normally accesses Webmin modules, security holes like this would usually not matter. When the user has been given less privileges through the use of module access control, however, a bug could let him execute arbitrary commands or edit files as `root`.

57.6 The log_parser.pl Log Reporting Script

Like all good Webmin modules, this one logs actions taken by users so that they can be viewed later in the Webmin Actions Log module. The `save_cron.cgi`, `delete_cron.cgi`, `save_allow.cgi`, and `exec_cron.cgi` programs all call the standard `webmin_log` function with parameters indicating what action has just taken place. As Section 56.5 "Action Logging" explains, this information is then written to a log file for later reporting.

Even though just about any arguments can be passed to the `webmin_log` function, it is usually a good idea to follow the standard that this and other modules use. The first *action* parameter should be the action performed, such as `save` or `delete`. The *type* parameter should be the kind of object to which the action applies, such as `cron` or `user`. The *object* parameter should be the name of the object affected, such as `fred` or `www.foo.com`. Finally, the *params* parameter must be a hash reference containing additional information about the action, such as the structure of the object being modified or the contents of `%in`. All parameters except *action* are optional, so it is quite reasonable and common for a module to use code like `&webmin_log("stop");`.

All of these programs also make use of the `lock_file` and `unlock_file` functions to obtain locks on files that they change. This causes the actual changes to the Cron files to be captured for inclusion in the log as well, so that inexperienced administrators can see exactly what the module has been doing. Your module should make use of these functions as well, especially those for locking. They protect critical files from simultaneous access, and give you detailed file change logs for free if you decide to add calls to `action_log`.

The other aspect of logging is the conversion of the logged parameters into human-readable form, which is done by the `log_parser.pl` script. If you view the code for this module, you will see that it simply uses the parameters to decide what to pass to `text`, and returns the resulting string. Note that the `html_escape` function is used to remove any special HTML characters from Cron commands, which may otherwise cause invalid HTML to be included in the log search results. If your module includes a `log_parser.pl` script that might return text containing characters like `<`, `>`, or `&`, be sure to call `html_escape` on the appropriate parts.

Unlike the `parse_webmin_log` function in most other modules, the one in the Scheduled Cron Jobs module checks the *long* parameter to decide if a long or short action description should

be returned. The long form includes the actual command in the Cron job, which will only fit on the page displaying details of a single log entry in the Webmin Actions Log module. In most modules, however, the message is always short enough to completely ignore this parameter.

If a parameter to `webmin_log` was omitted or set to `undef` by the CGI program that created it, the actual value passed to `parse_webmin_log` will be a single dash (`-`). This happens because a `-` is used in the log file to represent a missing parameter.

57.7 The `useradmin_update.pl` User Synchronization Script

As Section 56.2 “User Update Notification” explains, other Webmin modules can choose to be notified when a user is created, modified, or deleted in the Users and Groups module. This is normally used to keep some other user list in sync (such as the Samba password file), but can be handy for other purposes as well.

The Scheduled Cron Jobs module has a `useradmin_update.pl` script so that it can detect the renaming and deletion of users, and update their Cron job files, respectively. When a UNIX user is removed, his Cron jobs will normally continue to exist, even though they will no longer work. If a user is renamed, his jobs will still be listed under the old name, which will prevent them from working properly.

Those few modules that make use of a `useradmin_update.pl` script will probably have it perform different tasks than those in this module. See the script in the `samba` directory for an example of how to synchronize a separate password file, instead. If your module’s script does something similar, it should include options somewhere (perhaps on the **Module Config** page) that will turn synchronization on or off. Any such options should be off by default, so that other configuration files are not unexpectedly updated when the user is managing UNIX users. To avoid the first problem, the `useradmin_delete_user` function removes the personal Cron jobs file for any UNIX user who is being deleted. The `useradmin_modify_user` function checks to see if the user has been renamed and, if so, renames both the user’s personal Cron file and any jobs in other files. Any other changes to the user are ignored, as they are not relevant to this module.

57.8 Summary

This chapter has explained the inner workings of the Scheduled Cron Jobs module, so anyone wanting to write his own module can copy the same design style. It has covered the purpose of each module CGI program, its configurable options, and the access control, log parsing, and user synchronization scripts. It has also covered the functions in the module’s library, which can be called by other modules that need to create or manage their own Cron jobs.

Creating Webmin Themes

This chapter explains how themes work and takes you through the process of creating your own theme for Webmin. It covers both basic features such as image replacement, and advanced capabilities like writing an alternate header function.

58.1 Introduction to Themes

Webmin versions 0.83 and above support themes, which are sets of alternate user interfaces, graphics, and color schemes. A user can choose which theme he wants by going into Webmin Configuration and clicking on Webmin Themes. Multiple themes can be installed, but only one can be active for a Webmin user at any one time. If no theme is active, the default colors and layout are used.

Like a module, a theme is a directory under the Webmin root directory that contains certain files. The most important is the `theme.info` file, which has the same *name=value* format as the `module.info` file, one per line. The only required name and its value is:

`desc` A description for this theme, such as *My Webmin Theme*. This is the text that will appear in the theme selection menu.

A theme can also contain a `config` file, also in *name=value* format. The values defined in this file control the behaviour of the standard `header` and `footer` functions. Supported names and their values are:

`cs_page` A six-character hex string in RRGGBB format for the background color of Webmin pages.

`cs_link` A six-character hex string in RRGGBB format for the color of visited and unvisited links on Webmin pages.

`cs_text` A six-character hex string in RRGGBB format for the color of normal text.

`bgimage` A relative URL (like `/images/background.gif`) for a background image to be displayed on all pages.

`noindex` If set to 1, the HTML generated by the `header` function will not include a **Webmin Index** link. This is useful if another frame is used for the main index.

`brand` HTML for an image or text to be displayed in the top-right corner of the main index page.

`brand_url` A URL to which the `brand` image is linked. These two options are usually combined to create a company icon that links to its homepage in customized versions of Webmin.

`headhtml` HTML that will be included inside the `<head>` section of each Webmin page.

`headinclude` The name of a file in your theme directory whose contents will be included inside the `<head>` section of each page.

`inbody` Text that will be included inside the `<body>` tag itself.

`prebody` HTML that will be included at the top of the `<body>` section of each page. The following substitutions will be done in the HTML:

`%HOSTNAME%` will be replaced with the system's hostname.

`%VERSION%` will be replaced with the Webmin version.

`%USER%` will be replaced with the current user's login.

`%OS%` will be replaced with the OS name and version.

`texttitles` If set to 1, the titles on all pages will be displayed as HTML text rather than using letter images.

`postbody` HTML that will be included at the bottom of the `<body>` section on each page. The same substitutions as `prebody` are used.

`tb` Text that will be included inside the `<tr>` tag in table header rows.

`cb` Text that will be included inside the `<tr>` tag in table rows below the header.

`functions` The name of a file in your theme's directory that contains Perl functions for overriding the default `header`, `footer`, and `error` functions. See Section 58.3 “Theme Functions” for more details.

`noicons` If set to 1, the standard `generate_icon` and `icons_table` functions will display only a name instead of an icon. This can be useful if your theme is designed for text-only or low bandwidth use.

Many of these options will not work automatically if your theme uses the `functions` option to create its own replacement for the `header` function. They are normally checked for and implemented by the standard `header` function, so if you define your own it will need to check the `%tconfig` global hash and interpret the values that it contains in the same way if you still want them to be configurable in the theme's `config` file.

When you have created a theme and want to distribute it to other people, it should be packaged up. Just like modules, themes are packaged as an optionally compressed tar file of the

theme directory, usually with a .wbt extension. They can then be installed through the **Webmin Themes** page in the Webmin Configuration module.

Just like modules, themes can also be packaged as RPMs, which are suitable for installing on servers on which the RPM version of Webmin itself is already installed. You can download a script called `makemodulerpm.pl` from www.webmin.com/makemodulerpm.pl that can package up a theme directory into an RPM by creating the spec file automatically.

58.2 Overriding Images and Programs

In addition to changing the default colors, a theme can be used to selectively override any icon or CGI program used by Webmin. When a theme is chosen, its directory becomes an “overlay” root directory for the Webmin web server. Thus, if your theme subdirectory contains a file called `images/newlogo.gif`, it will replace the logo on the main menu when it is displayed because the web server will look in the theme directory first before looking for `images/newlogo.gif` under the top-level directory.

In this way, any of the module icons can be overridden, as can the images used to make up the titles at the top of pages. For example, if your theme directory contains a file called `useradmin/images/icon.gif`, it will be used as the icon for the **Users and Groups** module on the main menu. Because this “replacement” does not actually change the real images, the user can switch between themes or back to the default theme easily.

CGI programs can also be overridden in exactly the same way. This can be used to do things like changing the way the main menu is displayed, by putting a custom `index.cgi` script in your theme directory. This ability, however, should be used carefully as changes to the real CGI may break your custom script if its behaviour is different from the one it replaces. It should also be noted that when a theme CGI is executed, it will be in the real directory and not the theme subdirectory. This means that a custom top-level `index.cgi` script will require `./web-lib.pl` instead of `../web-lib.pl`, just as the real `index.cgi` does.

If your theme does replace an existing script, be sure to read it carefully so that your replacement implements all of the same functionality. Some of the things to keep in mind when replacing the top-level `index.cgi` program are:

1. The `get_available_module_infos` function can be used to get a list of modules available to the current Webmin user for use when generating any tables of icons.
2. If `$gconfig{'gotoone'}` is set to 1 and the user has only one module, your `index.cgi` should redirect the browser directly to that module instead of displaying a menu. Users can set this in the Index Page Options page of the Webmin Configuration module.
3. If `$gconfig{'nohostname'}` is set, no hostname or operating system information should be displayed. This can also be set on the Index Page Options page.
4. When the global variable `$gconfig{'nofeedbackcc'}` is set to 2, no feedback link should appear. This is configured on the User Interface page of the Webmin Configuration module.
5. If your menu program normally categorizes modules, categorization should be turned off when the `$gconfig{'notabs'}` variable is set so that all modules appear on a single page. Again, this is set on the Index Page Options page.

6. If your program arranges module icons in a table and the variable `$gconfig{'nocols'}` is set, it should be used as the number of columns to display.
7. If `$gconfig{'deftab'}` is set and your program categorizes modules, it should be used to decide which category to open by default.
8. The entries in `$config_directory/webmin.catnames` should be used to get user-defined categories and different names set for standard categories. This can be done on the Edit Categories page of the Webmin Configuration module.
9. If your program displays a logout link, it should only appear if neither of the following environment variables are defined. They both indicate that a form of authentication has been used that makes logging out impossible or irrelevant.
`$ENV{'SSL_USER'}` Indicates that the current user has logged in with SSL client authentication.
`$ENV{'LOCAL_USER'}` Indicates that the user is connecting from localhost and that his UNIX username matches his Webmin login.
10. If the variable `$main::session_id` is set, Webmin is in session (or cookie) authentication mode. You should generate a link to `/session_login.cgi?logout=1` labeled **Logout** or something similar.
If that variable is not set, however, then Webmin is using HTTP authentication. Instead, your code should create a link to `/switch_user.cgi` labeled **Switch User**, as the normal logout link above will not work.

It is not mandatory to implement all of the suggestions above. However, it will make your theme behave more like those included as standard with Webmin.

58.3 Theme Functions

In Webmin versions 0.92 and above, a theme can override some of the common HTML-generating functions by adding a line like `functions=theme.pl` to the `config` file and creating a `theme.pl` script in the theme's directory that contains one or more of the following functions:

`theme_header` This function will be called instead of the standard `header` function, with all the same parameters. It must handle all those parameters properly and output whatever HTML you want to use for the page titles and so on.

`theme_footer` Called instead of the standard `footer` function, with the same parameters. If your `theme_header` function opens an HTML table for layout purposes, this function must close it so that the HTML is properly complete.

`theme_error` Called instead of the standard `error` function, with the same parameters.

These functions give you a lot of power to create themes that significantly change the Webmin layout. For them to work properly, however, they must handle all the parameters that they are passed in exactly the same way that the standard functions do.

The most complex is the `theme_header` function due to the large number of parameters that it takes and the other global variables that it can make use of. When writing it, consider the following:

1. All parameters should be checked and interpreted in the same way as the real header function.
2. There is no need for your function to call `PrintHeader`, as this will be done automatically before it is called. It can just go ahead and start producing HTML.
3. The produced HTML should be valid and complete. That means starting with `<html>` followed by a `<head>` section containing the `<title>` and then the start of the `<body>` section.
4. If the variable `$gconfig{'sysinfo'}` is set to 0, information about the current user, hostname, operating system, and version of Webmin should be displayed in the browser's status bar. This can be done by producing HTML like:

```
<script>
defaultStatus = "fred logged into Webmin 1.060 on
www.example.com";
</script>
```

5. If `$gconfig{'sysinfo'}` is set to 1, this information should be appended to the page title in the `<title>` HTML tag. If `$gconfig{'sysinfo'}` is set to 2, the information should be displayed in the page itself, somewhere just below the title. If set to 3, the host and login details should not appear anywhere.
6. In all cases, the function `get_webmin_version` can be used to get the version of Webmin and `get_system_hostname` can get the current hostname.
7. The `<body>` tag that your code produces may contain `bgcolor=`, `link=`, and `text=` parameters containing the values from the global variables `$gconfig{'cs_page'}`, `$gconfig{'cs_link'}`, and `$gconfig{'cs_text'}`, if they are set. This ensures that color preferences set by the user on the User Interface page of the Webmin Configuration module are used. Feel free, however, to ignore them if your theme only looks good with a certain color scheme.
8. If every page includes a **Logout** or **Switch User** link, it should not appear if any of the following environment variables are set:
`$ENV{'SSL_USER'}` The current user has logged in with SSL client authentication.
`$ENV{'LOCAL_USER'}` The user is connecting from localhost and his UNIX username matches his Webmin login.
`$ENV{'ANONYMOUS_USER'}` This page is being accessed without any authentication at all.
9. If the variable `$ENV{'HTTP_WEBMIN_SERVERS'}` is set, your page heading should include a link to the URL in that variable labeled **Webmin Servers**. This is set when connecting via a tunnel in the Webmin Servers Index module, and the URL refers to that module on the originating system. This gives the user an easy way to return to the list of servers.
10. If your theme generates titles made up of letter images, plain text should be used instead, if `$gconfig{'texttitles'}` is set to 1. A setting on the User Interface page of the Webmin Configuration module controls this.

11. Unless your heading is very similar to the default, the `theme_footer` function should be defined as well. It must produce closing HTML that matches that produced by `theme_header`, followed by `</body>` and `</html>` tags. Be sure to interpret and display the multiple “return” links that can be supplied to the `footer` function as well.
12. Some themes normally put all page content into a table by outputting an unclosed `<table>` tag in the header function. If the global variable `$theme_no_table` is set, this should be turned off, as it indicates that the CGI program will be slowly producing some progressive output. Many browsers will not display a table’s content until it has been completely output.

Similarly, your `theme_footer` function should not produce a closing `</table>` tag when `$theme_no_table` is set.
13. The special CGI program, `session_login.cgi`, that displays Webmin’s login form will call `theme_header` as well. At this point, however, the browser has not logged into Webmin and so will not be able to access any images to which your header refers. For this reason, when called with the first two parameters set to `undef` (as it will be in this case), your function should not produce any `` tags. Or they should always refer to the special `/unauthenticated` URL path, to which access is always allowed even to clients that have not logged in.

Reading the `header` function in `web-lib.pl` and the `theme_header` function in `mscstyle3/theme.pl` should give you a good idea of how the various parameters and global variables should be handled.

58.4 Summary

After reading this chapter (and reading Chapters 55 and 56), you will be able to create themes to change the look and feel of Webmin. You should understand how a theme can override standard graphics and CGI programs, and how it can replace the standard `header` and `footer` functions to completely change the look of every page. You should also now know the requirements that a good theme must follow so it behaves like the standard themes.

Inside the MSC Theme

This chapter explains the inner workings of Webmin’s default theme, which makes use of almost all of the features available to theme developers.

59.1 Theme Design and Graphics

This new theme has been the default in Webmin since 0.92, although the old “classic” Webmin look and feel is still available. It makes use of practically all theme-related features, such as alternate graphics, CGI program replacement, and a library that replaces the standard header and footer functions. For these reasons it is a good one to look at if you are planning on writing your own theme.

The `mscstyle3` directory that contains the theme has a subdirectory for most of the standard modules, each of which contains an `images/icons.gif` file. These override the corresponding standard module icons that appear on Webmin’s main menu. No other overriding images exist—such as for icons within modules—meaning that the original images are still used.

The theme directory also contains an `images` subdirectory under which all of the images used by the theme itself are located. The heading that appears on every page is made up of numerous images, such as those for the category icons, category title letters, shaded background, and logout button. Because they are specific to the theme, most do not have any corresponding real image under the top-level `images` directory to override.

Like all themes must, this one includes a `theme.info` file containing its description that will appear on the Themes page in the Webmin Configuration module. It also includes a `config` file, which tells the Webmin API to read `theme.pl` and use the functions that it contains to replace the standard `header` and `footer` functions. It also contains several lines starting with `cs_` that set the table background and heading color. Unlike other simpler themes, the `config`

file does not need to specify any alternate page text or background colors, as these can be set directly by its `theme_header` function.

59.2 The index.cgi Program

Because most of the graphical customization done by this theme occurs in the `theme.pl` script, its replacement `index.cgi` program does not differ much from the standard `index.cgi` in the top-level Webmin directory. The biggest difference is that it does not output any special index page heading or image. Instead, it just gets the list of modules available to the current user with `get_available_module_infos` and uses it to build a table of icons in the current category. Because the theme's `theme_header` function also calls this function, the list may already be in the global `@msc_modules` variable. Theme CGI programs execute in the “original” directory instead of the directory they are really in, therefore the `index.cgi` program in the MSC theme can use the line `require './web-lib.pl';`.

It is possible to create quite a different theme just by replacing `index.cgi`, without the need for a theme's function file. The standard Caldera theme has an `index.cgi` script that actually generates a frame set, in which the top frame displays categories and modules, and the bottom shows actual pages within modules. It is possible for a theme to include CGI programs that do not override any real program, and are used only as part of the theme's user interface. For example, the `index_top.cgi` program that the Caldera theme uses to render the top frame. Again, such programs are run in the corresponding real directory, not in the theme's subdirectory.

The MSC theme's `index.cgi` and `theme.pl` scripts all make use of `%text` and the `text` function to get messages to display to the user, instead of hard coding them into the Perl code. All of the messages come from the appropriate file in Webmin's top-level `lang` directory. If you are creating your own theme that overrides any CGI program or function, the same thing should be done to take advantage of the existing translations into many languages that Webmin already includes.

59.3 The theme_header Function

The `theme_header` function in `theme.pl` in the `mscstyle3` directory effectively replaces the standard header function that almost all Webmin CGI programs call. Unlike the standard header, this one produces HTML for a list of module category icons at the top of most pages, allowing the user to easily switch to a different category. It also outputs HTML for a link to `www.webmin.com`, logout and feedback buttons, and the standard links like **Module Index** and **Module Config**. Finally, HTML is produced that puts the rest of the page inside the white table box that you can see on almost every page.

The table of categories is generated by calling `get_available_module_infos`, checking to see which categories actually exist, reading the file `/etc/webmin/webmin.catnames` to get alternate names, then displaying an icon and name made up of letter images for each. The theme has images for all of the standard categories, plus a special question mark image to be used if a nonstandard or user-defined category is found. Just generating a fixed table of standard categories will not work, as it is possible that the user only has access to modules in some of them.

Because the category titles may be in a different language that uses characters outside of the standard English alphabet, this theme includes images for every letter with ASCII codes between 32 and 255. Any other theme that uses letter images should do the same, so that it will

work in non-English languages as well. For some languages (such as Chinese and Russian), it is impossible to create an image for every single character, due to the thousands that exist. The MSC theme checks the global variable `$current_lang_info->{'titles'}`, and if it is not set, produces plain text category labels instead.

When Webmin is in session authentication mode (determined by checking for the `$main::session_id` variable), a logout image button is added to the top-right corner of every page. If normal HTTP authentication is being used, however, this is replaced by a button for switching users, which links to a different CGI program. The old Webmin theme only has these links on the main menu. The code properly checks the `$ENV{'SSL_USER'}`, `$ENV{'LOCAL_USER'}`, and `$ENV{'ANONYMOUS_USER'}` environment variables, any of which, if set, indicates that no logout or switch button should appear.

Every page also has a feedback button in the top-right corner, unless `$ENV{'ANONYMOUS_USER'}` is set or the global or per-user configuration indicates that feedback is not allowed. This links to `feedback_form.cgi` with the current module name as a parameter, so that any feedback sent to it is automatically associated with the current module. This is a nice idea if you are writing your own theme.

Below the row of category icons are several small tabs for links like **Module Index** and **Module Config**. The `theme_header` function checks its parameters and `$ENV` variables to determine which ones to show, just like the standard `header` function does. The biggest difference is that no **Webmin Index** link is ever produced, as there is no need for it. The user can return to the module's category by just clicking on the appropriate icon at the top of any page.

Below any tabs comes the page title, supplied to the `theme_header` function as the first parameter. The MSC theme puts it in a small tab above the page body using only text, unlike the old Webmin theme that renders the title as a series of letter images.

Finally, the theme puts page content output by the CGI after `header` is called into a large box, by producing HTML tags to start a table. This is not done if the global variable `$theme_no_table` is set. Instead, the content will just be part of the page's body. CGI programs that slowly generate progressive output should set this variable, and themes that have their own custom `theme_header` function should honor it, if appropriate. Of course, if your theme doesn't use this kind of table for layout then the variable can be ignored.

59.4 The theme_footer Function

This function is much simpler and not very different from the standard `footer` function from `web-lib.pl`. It just prints HTML to close the table started by `theme_header` (unless `$theme_no_table` is set), followed by links to previous pages as specified in the parameters. Finally, the required `</body>` and `</html>` tags are produced to properly end the page.

When overriding the `footer` function in your own theme, make sure that it properly processes all of the parameters, as multiple pairs of return links and titles can be provided. If your theme's main menu categorizes modules, any link to `/` should be replaced with a link like `/?cat=$module_info->{'category'}` so that the current module's category is displayed when such a link is used to return to the main menu. Most themes put an arrow next to each of the return links, but this is not required. Yours can just use text links, form buttons, or anything that you can think of.

If the last parameter to the `footer` function is non-zero, it will not produce HTML for the end of the page and instead will only generate the return links. The MSC theme's `theme_footer` function checks for this, if the number of parameters is odd, and does the right thing. Your theme should, too, as some Webmin programs depend upon this behavior to get the correct layout.

59.5 Summary

This chapter has taken you through the inner working of the complex MSC theme, which makes use of all the features available to a theme developer. After reading it, you should understand how this theme creates the category icons that appear at the top of every page and how it changes the appearance of Webmin while still preserving all of the functions available in the old classic theme.

The Webmin API

This chapter lists all of the functions available to Webmin modules and explains the parameters and purpose of each one.

60.1 API Functions

The `web-lib.pl` file contains a number of functions useful for generating HTML, parsing files, and all the other things Webmin modules need to do. The functions available in Webmin 1.060 are listed below in alphabetical order. The possible parameters for each function are shown as well, with those that are optional surrounded by []. The text ... in the parameter list indicates that the previous one can be repeated an arbitrary number of times.

`acl_filename()` Returns the filename in which the list of users and their modules is stored. For internal use only.

`additional_log (type, object, data)` When using file-change logging, this function adds an entry to the logs for the current action. This function is useful if Webmin has not done this for you automatically, for example, if your code has run a command with a piped open statement. You might use it like this:

```
open(PIPE, "somecommand |");
&additional_log("exec", undef, "somecommand");
```

The parameters of this function are:

type Must be one of `exec` (for logging commands run), `modify` (for a file change), or `sql` (for an executed SQL statement).

object The full path to the file modified for file change logs, or `undef` otherwise.

data The shell command-run, diff output for the modified file or the executed SQL statement.

`available_usermods(allmods, usermods)` Given a list of all modules and access control information, this function returns a list of modules to which the current Usermin user has access. This is mainly for internal use by the Usermin main menu and themes. *allmods* must be a reference to an array of module details as returned by `get_all_module_infos`, while *usermods* must be a reference to the array returned by `list_usermods`.

`backquote_logged(command)` This function is similar to the Perl backquote (`) operator, in that it executes the given command and returns its output, but it also records the command executed for later logging by `webmin_log`.

`check_ipaddress(string)` Returns 1 if the given string is a valid IP address like *10.254.1.100*, and returns 0 if not.

`check_os_support(moduleinfo)` Determines whether or not the module whose details are in the *moduleinfo* parameter is supported on this operating system. This must be a hash reference of the kind returned by `get_module_info` or `get_all_module_infos`. This function is mainly for internal use. If you just want to find out if a module is available on this system, use `foreign_check` instead.

`clean_environment()` Deletes all entries set by the Webmin web server from the %ENV global environment, such as REMOTE_USER and anything starting with HTTP_. This should be called before your CGI script starts a server process such as Apache or Squid so it doesn't get confused by environment variables normally only visible to CGI programs.

`close_http_connection(handle)` Closes an HTTP session handle created by `make_http_connection`.

`complete_http_download(handle, destfile, [error], [callback])` This function is used internally by `http_download` and `ftp_download`, and should never be called directly by module developers.

`copydata(in, out)` Reads data from the *in* filehandle and writes it to *out* until there is no more to read.

`create_user_config_dirs()` In Usermin, modules cannot store persistent configuration data in their configuration directory, `$module_config_directory`, because it is only writeable by root. If your code wants to store settings on a per-user basis, it should call this function, which will set the global variable `$user_module_config_directory` to `~currentuser/.usermin/modulename` and ensure that the directory exists.

The `defaultuconfig` files in the module's directory (`uconfig` in the module configuration directory under `/etc/webmin` and `config` in the user's module configuration directory) will be read in that order into the global hash `%userconfig`. See Section 56.9 “Creating Usermin Modules” for more details.

`date_chooser_button(dayfield, monthfield, yearfield, [formno])` Returns HTML for a button that, when clicked, allows the user to select a date. The *dayfield* parameter must be the name of a text input into which the day will be placed, *monthfield* the name of select input for the month, and *yearfield* the name of a text

input for the year. The *formno*, if given, is the number of the form on the current page that contains the inputs.

`decode_base64(string)` Takes a *string* in base-64 encoded format and converts it back to the normal form. This format is often used for email attachments and passwords in HTTP requests. The opposite is `encode_base64`.

`disk_usage_kb(directory)` Returns the number of kilobytes used by the given *directory* and all the files in it.

`encode_base64(string)` Returns the given *string* encoded in base64 format, with a return at the end of each line (even if only one line is produced).

`error(message)` This function is typically used by CGI programs that process the input from a form to inform the user of invalid input or some processing error by displaying the *message* parameter and exiting. It assumes that `error_setup` has been called first to set the first part of the error message, to which the parameter will be appended. This is demonstrated in the following example.

```
&error_setup("Failed to save user");
if (!$_in{'name'}) { &error("Missing username"); }
if ($_in{'name'} =~ /a/) { &error("'"$_in{'name'}' is not a valid
username"); }
```

`error_setup(message)` Any code that calls `error` should call this function first to specify that the *message* parameter should be prepended to all future displayed errors.

`fast_wait_for(handle, string, ...)` This function works like `wait_for`, but matches exact strings instead of regular expressions.

`file_chooser_button(field, type, form)` Returns HTML for a Javascript button that allows the user to select a file or directory on the server. The parameters are:

field - The name of the HTML field into which the chosen filename will be placed.

type - 0 for a file chooser and 1 for a directory chooser.

form - The form number containing the field—typically, 0.

`filter_javascript(html)` Given the *html* for an entire web page, this function attempts to strip out or disable all Javascript and returns the result. If your module displays HTML from an untrusted source (such as an email message or file on the server), it should call this function on the HTML in order to remove the potential security risk of malicious Javascript, which could capture the session key for the current Webmin login.

`find_byname(name)` Given a *name*, searches for processes matching that name and returns their PIDs. If none are found, an empty list is returned.

`flush_file_lines()` Writes the arrays of lines for any files requested back to disk by calling the `read_file_lines` function. A new line character is added to each line.

`flush_webmin_caches()` Clears all the in-memory and on-disk caches used by `web-lib.pl`. This function is really for internal use only.

`footer(link, text, [link, text], ..., [noendbody])` This function outputs a small left-facing arrow and a link with the text **Return to** *text*. Any CGI program that calls `header` must also call this function at the end to properly complete the page's HTML.

In Webmin versions 0.92 and above, you can specify multiple link and text parameters to have the function generate multiple footer links, like so:

```
&footer("", "module index", "list.cgi", "users list");
```

If the *noendbody* parameter is set to 1, the footer function will not produce `</body>` and `</html>` tags to properly end the page. It can be used if your program needs to output something after the normal footer. You must remember to print those tags yourself, however, so the HTML page is properly completed.

`foreign_call(module, function, [arg], ...)` This calls a function in another module and returns the results. The *module* parameter must be the module name, *function* must be the name of the function to call in that module, and any remaining parameters must be the arguments to pass to that function. For example:

```
&foreign_require("proc", "proc-lib.pl");
@procs = &foreign_call("proc", "list_processes");
&foreign_call("proc", "renice_proc", $pid, -10);
```

The example calls the `proc` module to get a list of processes and then to change the priority of some process.

In Webmin versions 0.960 and above, you can use the normal Perl syntax for calling functions in other modules, like:

```
&foreign_require("proc", "proc-lib.pl");
@procs = &proc::list_processes();
&proc::renice_proc($pid, -10);
```

`foreign_check(module)` Checks if some other *module* exists and is supported under the current operating system. If yes, it returns 1, otherwise, it returns 0. You should call this before calling `foreign_call` to access functions in other modules.

`foreign_config(module)` Returns a hash containing configuration options from the specified other *module*, just like the `%config` hash set by `init_config` for this module.

`foreign_require(module, file)` Before calling functions from another module with `foreign_call`, you must use this function to bring in the appropriate library. The *module* parameter is the name of the module in which you want to call functions and the *file* parameter is the name of a library file in that module directory.

`ftp_command(command, expected, [error])` This is used by the `ftp_download` function to send a command to the connected FTP server. Your module code should never call this function.

`ftp_download(host, file, destination, [error], [callback])` Makes an FTP connection to some *host* and requests the download of a *file*. The contents of this file are then stored in the given *destination* file. If an FTP proxy is configured, the download will be made via the proxy. The optional *error* and *callback* parameters

are only supported in Webmin versions 0.93 and above, and behave in exactly the same way as in the `http_download` function documented earlier.

`generate_icon(image, title, link, [href], [width], [height])` This outputs HTML for an icon with the given `image` and `title` below it, and as a hyperlink to the relative or absolute URL in the `link` parameter (if set). If the `href` parameter is supplied, it will be included in the `` tag, so that you can point the link to a different frame. The icon's size will be set to 48 x 48, unless the `width` and `height` parameters are given.

`get_all_module_infos(cachemode)` This returns a list of hash references, each containing the details of one installed module in the same format as returned by the standard `get_module_info` function. To avoid having to read one file from each module, this function usually caches the module details in the `/etc/webmin/module.infos.cache` file, which can be read much faster.

If the `cachemode` parameter is set to 0, normal caching of known modules is done. If it is set to 1, the cache is not used at all. If it is set to 2, any existing cache will be read but not written out.

`get_available_module_infos(cachemode)` This is similar to `get_all_module_infos`, but only returns modules available to the current Webmin or Usermin user instead of every one on the system. The `cachemode` parameter is passed on to `get_all_module_infos` and thus has exactly the same meaning.

`get_miniserv_config(hashref)` This fills in the `hashref` Perl hash reference parameter with the configuration from Webmin's built-in web server, `miniserv.pl`. This is generally read from `/etc/webmin/miniserv.conf`. The opposite function is `put_miniserv_config` for writing out a new configuration.

`get_module_acl([user], [module])` This returns a hash containing the ACL for the given `user` and `module`. If no user is specified, the current user is used. If no module is specified, the calling module is `user`. See Section 56.1 "Module Access Control" for more information on module ACLs.

`get_module_info(module, noclone)` This returns a hash containing information about the given `module` with the keys `desc`, `os_support` and `dir`. If the `noclone` parameter is set to 1, details of the underlying module will be returned when requesting information on a cloned module instead of the details of the clone.

`get_system_hostname()` This returns the hostname of the system on which Webmin is running. It is more reliable than the standard Perl `hostname` function, as it tries several different methods.

`get_theme_info(theme)` This is similar to `get_module_info`, but returns the details of a `theme` from its `theme.info` file.

`get_webmin_version()` This returns the version of Webmin under which this module is running.

`group_chooser_button(field, multiple, form)` This is like the `user_chooser_button` function, but used for the selection of groups instead.

has_command(command) This searches Webmin's PATH for the given *command*. Returns 1 if found, and 0 if not. If an absolute path is specified, the function checks to see if it exists and whether or not it is executable.

header(title, image, [help], [config], [noindex], [noroot], [text], [header], [body], [below]) The header function is used by almost all programs to output the HTTP header line (Content-type: text/html), HTML title, background, and title image. The parameters passed to this function are:

title The HTML title of this page. Also used as the ALT text for the title image.

image The URL of an image to display at the top of the page instead of the text from *title*. This should always be set to an empty string.

help If this parameter is defined, then a **Help** link is added to the title on the left-hand side, linking to the given help page.

config If this parameter is non-zero, a **Module Config** link is added to the left of the title. This links to a CGI program that allows the user to edit the configuration of this module, as defined by the config.info file. See Section 55.4 "Module Configuration" for more details.

noindex By default, the header function will add a **Module Index** link to the left of the title image linking to the index for this module. If this parameter is given and non-zero, this link will not be displayed.

noroot By default, a **Webmin Index** link to the Webmin main menu will be added to the left of the title image. If given, this parameter will suppress the addition of that link.

text HTML to be displayed to the right of the title image. This can be anything you like, as long as it fits in the small area available.

header HTML to be displayed in the <head> section of the page.

body Extra HTML tags to be include in the <body> tag.

below HTML to be displayed below the header. Often this is used on modules' main pages to show the version of the server that the module is configuring.

If a theme is in use that defines its own theme_header function, all of the above parameters will be passed to that function instead. This means that they may be interpreted quite differently, and the supplied title and other information may be placed in unexpected locations depending on the theme in use.

If this function is called by a Usermin module and the config parameter is set, your code must have called create_user_config_dirs beforehand to set up the ~/.usermin/modulename directory. Instead of a **Module Config** link being included in the header, one called **Preferences** will be included instead, which allows the user to edit his own personal settings for the module. The actual settings that can be changed are determined by the uconfig.info file in the module directory, which has the same format as config.info.

Some themes have a custom header function that puts all HTML output after the heading into a table. Unfortunately, some browsers will not display a table's contents

until its ending HTML tag has been generated. This means that if your CGI program is producing some progressive output (such as the new contents of a log file), or takes a long time to run, nothing will be visible to the user until it completes. To avoid this, the special global variable `$theme_no_table` should be set to 1 before `header` is called, indicating that page content should not be put in an HTML table.

`help_file(module, page)` This returns the full path to the file containing the HTML for the given *page* in the specified *module*. Files for the user's chosen language are used in preference to English, if they exist. This function is mainly for internal use only.

`help_search_link(terms, section, ...)` This returns HTML for a link to the System Documentation module for searching for the given *terms*. The *section* parameters after the terms determine what documentation is searched and each can be one of the following:

- `man` Manual pages
- `doc` Package documentation
- `kernel` Kernel documentation
- `howto` HOWTO documents
- `kde` KDE documentation
- `perl` Perl module documentation
- `help` Webmin help
- `google` The Google search engine

The return value from this function can be passed as the 7th parameter to the `header` function on the main page of your module, creating a link to additional info in manual pages or README files. For example:

```
&header("The Foo Module", undef, undef, 1, 1, undef,  
&help_search_link("foo", "man", "doc", "google"));  
hlink(text, page)
```

This returns HTML for a link to a help page. The *text* parameter is the text of the link and *page* is the name of the help page. See Section 55.7 “Online Help” for more information.

`html_escape(string)` Given a text *string*, this converts the characters <, >, and & to <, >, and &, respectively, among others. It should be used when text from a particular source that may contain HTML characters is going to be included in a page generated by your module, so the text appears exactly as it should and potentially malicious HTML code (perhaps containing Javascript) is neutralized.

`http_download(host, port, page, destination, [error], [callback])` This makes an HTTP connection to a web server *host* and *port* request a *page*. The contents of this page are then stored in the *destination* file. If the user has configured his Webmin installation to use a proxy server, then the HTTP request will go through that proxy.

The optional *error* and *callback* functions are only supported in Webmin versions 0.93 and above. If *error* is supplied, it is a reference to a scalar that will be set with

an error message if the download fails, instead of the function simply calling the standard `error` function.

The `callback` parameter can be a reference to a function to which it will be called back at various stages of the download process. When called, the first parameter indicates the status and the second indicates specific additional information. Possible status codes are:

- 1 Server has been contacted. The second parameter is 1 if the requested URL is a redirect, or 0 if it is a normal file.
- 2 Download has started. The second parameter is the size of the file being downloaded, if it is known.
- 3 Some data has been received. The second parameter is the amount of data received so far. This will be called for every 1kb (or less) of data received.
- 4 Download complete. The URL has been totally downloaded. No second parameter is supplied.
- 5 Redirected to new URL. The second parameter is the new URL to which the request has been redirected.

If you just want to display the progress of a download in the way that some of the code Webmin modules do, the standard function `progress_callback` can be passed as the 6th parameter to `http_download`. You must, however, set the global variable `$progress_callback_url` to the URL or name of the file being downloaded, for use in the progress messages.

`include(file)` This copies the content from the given `file` to STDOUT.

`icons_table(links, titles, icons, [columns], [href], [width], [height])` The main Webmin page and many modules use grids of icons, each linking to a different option, domain, share, or something related. This function generates an icons grid based on the lists given as parameters. `links` is a reference to an array of URLs, `titles` is a reference to an array of messages that appear below icons, and `icons` is a reference to an array of image URLs.

If the `columns` parameter is given, it specifies the number of icons to display per row. The default is for each row to contain four icons. The `href`, `width`, and `height` parameters have exactly the same meaning as in the `generate_icon` function, which this function actually calls to create each icon.

`indexof(value, array)` This returns the index of some `value` in the `array` that comprises the rest of the parameters, or -1 if not found.

`init_config()` This initializes global configuration variables. See Section 55.3 “Module CGI Programs” for more details. Please note that the `init_config` function had to be passed the name of the module as a parameter prior to version 0.73. From 0.73 onwards, the module name is worked out automatically.

`is_under_directory(directory, file)` This returns 1 if the given `file` is under the specified `directory`, 0 if it is not. Both must be absolute paths. If the file is actually a symbolic link, its target must be under the directory for the function to return 1. This

can be useful in modules that enforce restrictions on the directories in which users are allowed to edit files.

`kill_byname(name, signal)` Given a name, this searches for processes matching that *name* and kills them with the given *signal*.

`kill_byname_logged(name, signal)` This is like the `kill_byname` function, but also records the command executed for later logging by `webmin_log`.

`kill_logged(signal, pid, ...)` This function is exactly the same as the Perl `kill` statement, but also records the signal and PIDs for later logging by `webmin_log`.

`list_languages()` This returns a list of hash references, each containing the details of a supported language. This function is generally for internal use only.

`list_usermods()` This returns an array containing information about which Usermin users can have access to certain modules. For internal use only.

`load_language(module)` This returns a hash containing translations for some *module*, just like the `%text` hash that `init_config` sets for this module.

`load_theme_library()` This reads the `theme.pl` file for the current theme if one exists and if it has not been loaded yet. This is another internal use only function that module writers should not call.

`lock_file(file)` This function obtains an exclusive lock on a given *file*, if necessary, waiting until the lock is released if it is held by another program. Locking is done by creating a `.lock` file contain the PID of the process, which guarantees that locks will not be held forever by dead processes. Locks can be made on files, directories, and symbolic links, and should be made before any of those are created, modified, or deleted.

`make_date(time_t)` Given a UNIX `time_t` value (seconds since 1970), this returns a date-time string in the `dd/mm/yyyy hh:mm` format.

`make_http_connection(host, port, ssl, method, page)` This is a general function for making an HTTP connection, possibly using SSL. The function will attempt to connect to the given *host* and *port* (in SSL mode if the *ssl* flag is set) through a proxy server if you have one configured in Webmin. It will then make an HTTP request using the given *method* and *page*, and return a session handle reference if no errors are encountered. If any error does occur in the connection, a scalar error string will be returned.

The returned session handle should be used with the `read_http_connection` and `write_http_connection` functions to send any additional headers and to read back the response headers and body. When done, the `close_http_connection` function should be called with the session handle.

`no_proxy(host)` This returns 1 if Webmin will connect directly to the given *host* for HTTP requests. It returns 0 if a proxy will be used. This is mainly for internal use by the various HTTP-related functions.

`open_socket(host, port, handle, [error])` This function attempts to open a TCP connection to the specified *host* and *port* using the given Perl file *handle*. Once this

function returns, the caller can read from or write to the handle to communicate with a remote system, and call `close` on it when done. If the connection cannot be made, the `error` function is called with a message explaining what went wrong, unless the `error` parameter to this function has been set as a scalar reference. If so, the error message is placed in that variable and the function returns 0 (instead of the usual return value of 1).

`other_groups(user)` This returns a list of secondary groups to which the UNIX `user` belongs.

`PrintHeader([charset])` This outputs the `Content-type: text/html` header (and possibly others) that all CGI programs that produce HTML must generate. If the `charset` parameter is given, it specifies the character set of the HTML. This function is not generally called directly. Instead, the `header` function will do it for you with the right character set for the user's language.

`progress_callback(action, details)` This function exists to be passed to `http_download` so that download progress reports will be printed. Calling it directly from your code makes no sense.

`put_miniserv_config(hashref)` This writes the configuration from the hash reference parameter `hashref` to the configuration file used by Webmin's built-in webserver. You will need to call `restart_miniserv`, however, for this change to have any effect.

`ReadParse()` This function takes any CGI parameters passed to your program (from form inputs or after the ? in the URL) and places them in the `%in` associative array. If a CGI parameter has multiple values (for example, from a list that allows multiple selections), then those values are separated by null characters (\0 in Perl).

`ReadParseMime()` When writing a CGI program that handles input from a form using `enctype=multipart/form-data`, this function must be called instead of `ReadParse` to fill the `%in` array with form inputs. You must add the `enctype` tag to any forms using file-upload inputs.

`read_acl([hashref1], [hashref2])` This function fetches the current list of those to which the Webmin user has access and stores it in the hashes referenced by the two parameters. `hashref1` will be filled in as a two-key hash, in which the first key is a username and the second a module name. `hashref2` will be filled in with usernames, each referring to an array of modules that the user has. For example:

```
&read_acl(\%hash1, \%hash2);
print "User $remote_user has access to Users and Groups<br>\n";
if ($hash1{$remote_user, 'useradmin'});
print "User $remote_user has access to ",
join(" ", @{$hash2{$remote_user}}), "<br>\n";
read_env_file(file, hash)
```

This reads a `file` of /bin/sh variable assignments in `key=value` or `key= "value"` format into the given `hash` reference.

`read_file(file, hash)` This reads a *file* in *key=value* format into the given *hash* reference.

`read_file_cached(file, hash)` This is like the standard `read_file` function, but it keeps a cache of files read in order to avoid reading them multiple times. This is mostly for internal use at the moment.

`read_file_lines(file)` This returns a reference to an array containing the lines from the given *file*, with any newline characters removed. The caller can then modify this array by adding, removing, or changing lines using functions like `push` and `splice`. `flush_file_lines` can then be called to write changes back to the original files.

`read_http_connection(handle, [amount])` This reads either a single line from the given session *handle* returned by `make_http_connection`, or the specified number of bytes if the *amount* parameter is supplied.

`redirect(url)` Given a relative or absolute *url*, this outputs an HTTP header to redirect the browser to that URL. This function will not work if called after `header`, and vice-versa.

`remote_error_setup(handler)` When one of the `remote_` functions encounters an error (such as the remote Webmin server being down), it will generally call the standard `error` function, which will cause your CGI program to exit. If you use this function to register an alternate error *handler*, however, it will be called, instead, and the `remote` function will return its return value.

`remote_eval(server, module, code)` This executes specific Perl *code* on a remote Webmin *server* in the context of the given *module*. This can be very useful if you want to do something that is not possible by calling a function with `remote_foreign_call`. You must, however, first call `remote_foreign_require` with the same server and module before using this function.

`remote_finished()` This function should be called by any CGI that makes use of any of the other `remote_` functions once it has finished calling them, to clean up connections to the remote servers. It is not strictly necessary, however, as the connections will timeout after 30 seconds of inactivity. When using fast RPC, this also never needs to be called as remote sessions will exit as soon as your CGI finishes.

`remote_foreign_call(server, module, function, [arg], ...)` This calls a *function* in some *module* on another *server* and returns the results. You must already have called `remote_foreign_require` for the same server and module before trying to use this function. The *function* parameter is the name of a function to call in the remote module, and the *arg* parameters after it are arguments that will be passed to it. For example:

```
&remote_foreign_require("www.blah.com", "apache", "apache-
lib.pl");
@servers = &remote_foreign_call("www.blah.com", "apache",
"get_config");
```

As the example shows, the `remote_foreign_call` function returns whatever is returned by the function on the remote server.

`remote_foreign_check(server, module)` This checks to see if a *module* exists and is supported on a remote Webmin *server*. If yes, it returns 1, otherwise it returns 0. If in doubt, you should call this before calling `remote_foreign_call` to access functions in other modules on other servers.

`remote_foreign_config(server, module)` This function is similar to `foreign_config`, but is for fetching a *module* configuration from a remote *server* instead. Unlike `foreign_config`, however, it returns a hash reference rather than a hash.

`remote_foreign_require(server, module, file)` Before calling functions from a module on another server with `remote_foreign_call`, you must use this function to bring in the appropriate library. The *server* parameter is the hostname of the remote Webmin server, the *module* parameter is the name of the module in which you want to call functions, and the *file* parameter is the name of a library file in that module directory.

`remote_write(server, localfile, [remotefile])` If, when making remote function calls, you need to transfer a large amount of data to a remote *server*, this function should be used instead of passing it in a scalar through `remote_foreign_call`. The *localfile* parameter is a file on the server on which the function is called, and the *remotefile* parameter is the name of a file on the remote server to which the contents of *localfile* will be copied. If *remotefile* is omitted, a random temporary filename will be chosen instead and returned from the function.

`remote_read(server, localfile, remotefile)` This is the opposite of `remote_write`, in that it copies data from a file on a remote Webmin server into a local file.

`remote_rpc_call(server, command)` This internal function is used by all of the other `remote_` functions to actually open a connection to the specified *server* and send the *command* structure to tell it what to do. You should never call it directly.

`rename_logged(old, new)` This renames a file like the Perl `rename` function, but also records the event for later logging by `webmin_log`. While you could just lock the *old* and *new* files before renaming, that would generate two large and rather useless file differences.

`replace_file_line(file, line, [newline], ...)` This function removes one *line* from a *file* and replaces it with 0 or more new lines from the *newline* parameters. This is done by reading the entire file into memory and writing out the modified version.

`reset_environment()` This returns the environment to the state it had before the last call to `clean_environment`.

`resolve_links(file)` Given a *file* name that may contain symbolic links somewhere in its path, this function returns the actual real filename to which it refers. Unlike the Perl `readlink` function, this also resolves symbolic links in directories along the path as well.

`restart_miniserv()` This restarts Webmin's built-in web server, forcing it to reload its configuration. This function is mainly used by the Webmin Configuration module and is probably of little use to the average module coder.

`same_file(file1, file2)` This returns 1 if `file1` and `file2` refer to the same actual file, by comparing inode numbers.

`save_module_acl(acl, [user], [module])` This saves the given module ACL hash. If no `user` is specified, the current user is used. If no `module` is specified, the caller's module is used. See Section 56.1 "Module Access Control" for more information on module ACLs.

`seed_random()` This seeds the Perl random number generator so calls to `rand` will return truly random results. It uses `/dev/urandom` if it is available, or the current time and process ID otherwise.

`serialise_variable(variable)` This converts the given Perl scalar or reference `variable` into a text string. This function is mainly used by the various RPC-related `remote_` functions for encoding parameters and return values, but you may find it useful for persistently storing Perl objects. The `unserialise_variable` function does the reverse.

`switch_to_remote_user()` This function should only be called by code in Usermin modules and will switch the UID and GID of the current process to those of the UNIX user whose login matches the current Usermin login. You should call this function after `init_config` in your module's library if your Usermin module can run with normal user permissions (and most can).

In addition to switching the UID, this function also sets the global variable `@remote_user_info` to the details of the UNIX user, as returned by the `getpwnam` Perl function.

`sysprint(handle, value, ...)` This calls the Perl `syswrite` function to print the `values` of the given file `handle` without any buffering.

`system_logged(command)` This function is exactly the same as the Perl `system` statement, but also records the `command` executed for later logging by `webmin_log`.

`tempname()` Returns a pathname in `/tmp` that can be used as a temporary file. The actual filename will always be under the `/tmp/.webmin` directory, which is only writeable by `root` but is world readable. If your temp file is going to contain security-critical information, it should have its UNIX permissions changed to mode 700 before writing.

`terror(string)` This is like a combination of the `error` function and the `%text` array. A call to `&terror('foo')` is exactly the same as `&error($text{'foo'})`. This function really just exists to make calling `error` more convenient in an internationalized module.

`text(message, [parameter], ...)` This function looks up the given `message` in the appropriate language translation file, replaces the text `$1`, `$2`, and so on with the rest

of the *parameters*, and returns the result. See the section on “Internationalization” in Chapter 56 for more information.

`to_ipaddress(hostname)` Given a *hostname*, this function returns a string like 10.254.1.100 representing the IP address for that hostname, or `undef` if none was found. If the parameter is already an IP address, it is returned unchanged.

`trunc(string, length)` This truncates a *string* of space-separated words so that it is less than or equal to the given *length*, without chopping off part of a word. It is useful for word-wrapping output to a fixed width.

`unique(value, ...)` This is given a list of *values* and returns an array with duplicates removed.

`unix_group_input(field, group)` This returns HTML for a text box named *field* for entering a group name, with a button next to it that pops up a window for selecting a group. The *group* parameter sets the initial value for the field.

`unix_user_input(field, user)` This returns HTML for a text box named *field* for entering a username, with a button next to it that pops up a window for selecting a user. The *user* parameter sets the initial value for the field.

`unlock_all_files()` This function releases all locks currently held by this program by calling `unlock_file` multiple times.

`unlock_file(file)` This releases a lock on *file* grabbed by the `lock_file` function. If the logging of file changes is enabled, a difference comparison between the old and new file contents will be done when this function is called. This comparison is done using the standard `diff` command.

`unserialise_variable(string)` This converts a *string* created by `serialise_variable` back into a Perl scalar or a reference of some kind.

`un_urlize(string)` This function decodes the special URL escape sequences in the given *string* and returns the original text. For example, `hello%20world%21` would be converted to `hello world!`.

`urlize(string)` This converts an arbitrary *string* to a form suitable for use in a URL. For example, `don't jump!` will be converted to `don%27t%20jump%21`.

`user_chooser_button(field, multiple, form)` This returns HTML for a **Javascript** button that allows the Webmin user to select a user or users from those on the server. The parameters are:

field The name of the HTML file into which the chosen user or users will be placed.

multiple 0 for selecting a single user, 1 for selecting multiple users.

form The form number containing the field—typically 0.

`wait_for(filehandle, regexp, ...)` Given a Perl *filehandle* and a list of regular expressions in the *regexp* parameters, this function reads from the file handle until one of the expressions matches. It then returns the regular expression number and fills the `@matches` global array with the values of any bracketed sections of the matching expression. If an error occurs while reading the input (perhaps because there is no more to read), -1 is returned instead. It can be useful for interacting with

other programs or servers that would normally take user input, as the following example shows:

```
&open_socket("somehost", 23, TELNET);
select(TELNET);
$|=1;
select(STDOUT);
while(1) {
    my $rv = &wait_for(TELNET, "login:", "password:", ">");
    if ($rv == 0) {
        print TELNET "fred\n";
    }
    elsif ($rv == 1) {
        print TELNET "mypassword\n";
    }
    elsif ($rv == 2) {
        print TELNET "ls -la\n";
        close(TELNET);
        break;
    }
    else {
        &error("Telnet failed!");
    }
}
```

`webmin_log(action, type, object, params)` As explained in the “Action Logging” section of Chapter 56, this function writes the given parameters identifying the action performed by the calling program to the detailed log file.

`write_env_file(file, hash, export)` This function writes the contents of a `hash` reference to the given `file` in /bin/sh variable assignment format. If the `export` parameter is non-zero, each variable assignment is preceded with `export`.

`write_file(file, hash)` This writes the contents of a `hash` reference to the given `file` in `key=value` format. This can be read in by the `read_file` function.

`write_http_connection(handle, [data], ...)` This function writes the given `data` strings to the HTTP session `handle`.

60.2 Summary

This chapter has documented the parameters and purposes of all of the standard functions from `web-lib.pl` that are available to Webmin modules written in Perl. Chapters 55 through 59 explain the most important of these functions in more detail and how they should be used by properly written modules.

Index

A

Access checking order field 293
access control
 Apache Webserver module 310–311
 BIND DNS Server module 346–347
 Custom Commands 230–231
 DHCP Server module 374–375
 Disk and Network Filesystems
 module 49–50
 Disk Quotas module 66–67
 File Manager 242–243
 Majordomo List Manager
 module 401
 MySQL Database Server
 module 423–424
 Network Configuration module 152–153
 Partitions on Local Disks
 module 73–74
 PostgreSQL Database Server
 module 443–444
 PPP Dialin Server module 172
 Printer Administration module 212–213
 Running Processes module 103–104
 Scheduled Cron Jobs module 96
 Sendmail Configuration module 468–469
 Squid Proxy Server module 601
 System and Server Status
 module 262
 System Logs module 118–119
 System Time module 193
 Users and Groups module 34–37
 Web Users module 697–698
 Webalizer Logfile Analysis
 module 498–499
 Webmin Server Index module 705
Access Control icon 293
Access control policy field 515
Access file shares field 574
Access files as UNIX user field 357
Access groups field 375
Access hosts field 375
Access log files table 285
Access print shares field 574

Access shared nets field 375
Access subnets field 375
access.conf file 310, 314
accounts 1
ACL name field 585, 586
acl_filename function 751
acl_security.pl script 727, 738–739
acl_security_form function 738
acl_security_save function 738
ACLs (access control lists) 51
 creating 586–587
 editing 239–240, 341–342,
 586–587, 665–666
 in Squid proxy server 584–586
 types 587–592
action_log function 739
actions 85
 adding 87–89
 configuring 85–86
 logging 726–727
 starting 86–87
 stopping 86–87
Activate CVS Server button 356
Add \$ttl to top of new zone files field 350
Add monitor of type button 252
Add virtual server to file field 270
Additional parameters condition 189
additional_log function 751
Address field 329
address mappings 456–457, 474
Address pattern field 344
Address record 316, 323, 325
Address username field 485
addresses 316
Addresses for name virtual servers
 field 272
Addresses matching regexps option 398
Addresses not matching regexps
 option 398
Administration login field 425, 444
Administration password field 425,
 442, 445
AIX operating system 37
 bootup scripts 91
 filesystem 51
 Internet service 138
 Majordomo package in 390
 system logs 119
AIX print system 572
alert priority 115
Alias to field 453, 479
aliases 276–277
All log files in report field 492
All requests condition type 293
Allow comments for records field 349
Allow editing of From:address field 634
Allow long hostnames field 350
Allow queries from field 335
Allow user sorting of column option 275
Allowed uploaded filename regex
 field 522
Alpha systems 105, 195
Also notify slaves field 335
anonymous FTP 506–507, 529–531
Answer after field 166
Answer mode field 217
answering machines 215–216
 configuring Linux systems as 216–218
 setting greeting messages 219
viewing and managing record
 messages in 218–219
Apache Options Files module
 (Usermin) 638
apache script 85
Apache servers 264
 adding and editing MIME types 288–289
 aliases 276–277
 character sets 295–296
 configuring as proxy server 301–304
 custom error messages 287–288
 directives 264–265, 308–310
 editing pages on 268–269
 encodings 295
 .htaccess files 297–298
 languages 296
 modules 264
 password protection 289–291
 redirects 278–279
 restricting access by client
 address 293–294
 root directory 313

- setting options for directories 273–276
 starting and stopping 268
 synchronizing text
 authentication files
 with UNIX user list 292
 user web directories 300–301
 versions of 264, 313
 virtual servers 269–273
- Apache Webserver module 265–268
 access control 310–311
 adding and editing MIME types 288–289
 configuring 311–314
 configuring Apache as proxy server 301–304
 configuring logging 284–285
 creating .htaccess files 298
 creating aliases 276–277
 creating redirects 278–279
 creating separate log files 285–286
 creating virtual servers 269–272
 editing .htaccess files 297
 editing pages on Apache servers 268–269
 editing virtual servers 272–273
 protecting directories 289–291
 restricting access by client address 293–294
 restricting directives in .htaccess files 299
 running CGI programs 280
 setting per-directory options 273–276
 setting up custom error messages 287–288
 setting up directory for CGI scripts 280
 setting up server-side includes 282–284
 setting up SSL (Secure Sockets Layer) 304–307
 setting up user web directories 300–301
 starting and stopping Apache servers 268
 viewing and editing character sets 295–296
 viewing and editing directives 308–310
 viewing and editing encodings 295
- viewing and editing languages 296
- Apache Websrvr monitor 254
 apachectl command 314
 AppleShare filesharing protocol 53
 Applet size field 222
 Apply to class field 540
 Approval email address field 396
 approve /tmp/email file 399
 apt-get command 106
 apt-get install webmin command 6
 Args field 133
 Arguments to calamaris command field 602
 AS numbers 588
 ASCII text 418
 at command 97
 atq command 97
 atm command 97
 attr command 239
 attributes 238
 auth facility 114
 Authentication mode field 442
 Authentication option 299
 Authentication program field 594
 Authentication realm name field 289
 authpriv facility 114
 Automatically include logfiles from field 499
 Automatically update serial numbers field 351
 automounter filesystems 47
 autoreply aliases 465–466, 480
 available_usermods function 752
- B**
- backquote_logged function 727, 752
 Backup file directory field 424
 Backup file path field 438
 Backup to file field 418
 backups 121, 417–418
 adding 122–124
 deleting 125
 editing 125
 making 124–125
 PostgreSQL 437–438
 restoring 125–126
 banner files 536–538
 Base directory for Apache documentation field 313
 Base logfile path field 497
 batch files 28–29
- Berkeley Internet Name Domain (BIND) 318
 bgimage 742
 bigint field 413
 BIND (Berkeley Internet Name Domain) 318
 BIND 4 DNS Server module 347–353
 BIND 4 DNS Server monitor 254
 BIND DNS Server module 318–320
 access control 346–347
 adding and editing records 322–325
 configuring 347–352
 configuring forwarding and transfers 340–341
 creating and editing forward zones 336–337
 creating master zones 321–322
 creating root zones 337
 creating slave zones 332–334
 editing access control lists 341–342
 editing master zones 330–332
 editing slave zones 334–336
 editing zone defaults 338–340
 setting up partial reverse delegation 342–344
 using BIND views 345–346
 BIND DNS Server monitor 254
 BIND servers 250
 setting up forwarding in 341
 setting up transfers in 341
 views 345–346
 BIOS (Basic Input Output System) 84, 195
 bits 303
 blob field 410, 415
 Block requests to domains field 303
 blocks quota 60
 Boot device field 198
 boot disk 195
 Boot file server field 366, 371
 Boot filename field 366, 371
 Boot image partition field 202
 boot loaders 84, 195–196
 boot/grub/menu.lst file 196
 BOOTP protocol 362
 Bootup and Shutdown module 85
 adding new actions 87–89
 configuring 89
 configuring actions to start at bootup 85–86
 starting and stopping actions 86–87

- Bootup commands field 88
bootup script 553
brand 742
brand_url 742
Broadcast address field 367
Broadcast field 147–148
Browser Regexp ACL (access control list) 587
browsers 233, 294
BSD print system 572
- C**
- C programming language 280
Cache Host field 599
cache manager statistics 599
Cache Port field 600
cachemgr.cgi program 599, 603
caching 303, 578, 583–584
Calamaris 600
Caldera Linux 6
Caldera Theme 684
callback parameter 758
caller ID 171
can_edit_user function 736
categories 683
category file 711
Category icons 267
\$cb variable 713
CD directory aliases field 535
CD-ROMs 47, 195
cert.pem file 305, 306
certificate authority 17, 686–687
Certificate file field 18
certificate signing request (CSR) 18, 307–308
Certificate/private key file field 306, 307
certificates 17–18, 307–308
.cgi files 280
CGI programs 712–715
 Cron jobs 734–735
 directory for 280
 errors in 282
 running 280
 susexec program for 281–282
CGI scripts 268
chacl command 239
chains 174
 actions 179–180
 changing default actions of 181
 creating 182
 exiting 179
Change Language module (Usermin) 625, 638
Change Password module (Usermin) 638
Change Theme module (Usermin) 629, 638
Change User Details module (Usermin) 639
change_cron_job function 736, 737
CHAP protocol 171
char field 410–411, 414, 433, 435
Character set for repository file field 360
character sets 295–296
chargen service 130, 132
Check condition field 381, 385
Check File monitor 254
Check Process monitor 255
check_ipaddress function 752
check_os_support function 752
chmod command 351, 541
chroot command 348, 529
Chroot directory to run BIND under field 348
CIFS (Common Internet File System) 554
class C network 317
clean_environment function 752
Clear history button 223
Client Address ACL (access control list) 587
Client Hostname ACL (access control list) 588
Client map restrictions table 161, 162
Client Regexp ACL (access control list) 588
Cloned module name field 675
close_http_connection function 752, 759
Cluster category
 Cluster Software Packages module 644
 Cluster Users and Groups module 649–650
 Cluster Webmin Configuration module 660–661
Cluster Software Packages module 644
 configuring 648
 deleting packages 647
 exploring and removing packages 647–648
installing packages 646
registering servers 645
searching for packages 646–647
Cluster Users and Groups module 649–650
 configuring 659
 creating groups 654
 creating new users 651–652
 deleting groups 656
 deleting users 653–654
 editing groups 654–655
 editing users 652–653
 editing Webmin users 662
 listing and removing servers 658
 refreshing user and group lists 656
 registering servers 650–651
 synchronizing users and groups 656–658
Cluster Webmin Configuration module 660–661
 configuring 668
 creating Webmin groups 664
 deleting modules or themes 667
 deleting Webmin groups 665
 editing ACLs (access control lists) of users and groups 665–666
 editing Webmin groups 664–665
 installing modules or themes 666–667
 listing and removing servers 668
 refreshing user and module lists 667–668
 registering servers 661
Cluster Webmin Servers module 665
clusters 643–644
CNAME (Name Alias) record 326, 329, 342–344
COAS driver option 209
Command Shell module 222–223
Command Shell module (Usermin) 639
Command to apply configuration field 377
Command to check exit status of parameter 255
Command to check for parameter 255
Command to initialize PostgreSQL field 446
Command to run after backup field 123
Command to run after disconnecting field 382
Command to run before backup field 123

- Command to run before connecting field 382
 Command to run on connect as root field 567
 Command to run on disconnect as root field 567
 Command to start apache field 314
 Command to start BIND field 352
 Command to start DHCP server field 377
 Command to start MySQL server field 426
 Command to start PostgreSQL field 446
 Command to start ProFTPD field 523
 Command to start Qmail field 490
 Command to start Samba servers field 576
 Command to start sendmail in server mode field 473
 Command to start squid field 603
 Command to start sshd field 553
 Command to stop apache field 314
 Command to stop MySQL server field 426
 Command to stop PostgreSQL field 446
 Command to stop Qmail field 490
 Command to stop Samba servers field 576
 Command to stop Sendmail field 473
 Command to stop squid field 603
 commands 224–225
 apachectl 314
 apt-get 106
 at 97
 atq 97
 atrm 97
 creating 225–227
 delete 542
 deselect 106
 dump 123
 edquota 61
 emerge 106
 get 397
 get list file 389
 getfac 239
 hostname 150
 index 397
 index list 389
 info 397
 info list 389
 intro 397
 intro list 389
 listname 394
 lists 389, 398
 locate 93
 make 246
 make install 246
 make test 246
 overwrite 542
 parameter types 227–228
 quota 61
 rename 542
 repquota 61
 restore 126
 scheduled 97–98
 sendmail 391
 setfac 239
 subscribe list address 388–389
 unmask 542
 unsubscribe list address 389
 which 397
 who 397
 who list 389
 Common Gateway Interface (CGI) 279
 Common Internet File System (CIFS) 554
 Common Name field 305, 308
 complete_http_download function 752
 Comprehensive Perl Archive Network (CPAN) 244
 concatenated RAID array 75
 concurrent logins 520
 Concurrent user limits table 540
 Concurrent Versions System (CVS) 354
 conditional blocks 612
 %config variable 713
 config file directory 9
 Configuration Engine Daemon monitor 255
 configuration files 1, 715–717
 Connection state 189
 Connection timeout parameter 258, 259
 Connection type field 221
 CONT signal 101
 Content encodings table 295
 Content handlers table 283
 Content languages table 296
 Control lines mode field 168
 cookies 512
 Copy button 234
 copydata function 752
 Country code field 16, 631
 Country name field 305, 308
 cp program 223
 CPAN (Comprehensive Perl Archive Network) 244
 cpio command 530
 CPU mode 100
 Create a New Virtual Server form 270
 CREATE statement 418
 create_cron_job function 736
 create_user_config_dirs function 733, 752, 756
 crit priority 115
 cron facility 114
 Cron jobs 93
 CGI programs 734–735
 controlling users' access to 96
 creating 94–95
 editing 95
 in other operating systems 97
 cron.allow file 734
 cron.deny file 734
 cron-lib.pl library script 735–737
 crontab command 737
 cs_link 741
 cs_page 741
 cs_text 742
 CSR (certificate signing request) 18, 307–308
 csr.pem file 308
 CUPS driver option 209
 CUPS print system 206, 209, 211–212, 572
 Current issue number field 401
 Current status field 253
 Current volume number field 401
 \$current_lang variable 713
 %current_lang_info variable 713
 \$current_theme variable 713
 custom archives 438
 Custom Commands module 224–225
 access control 230–231
 configuring 231
 creating file editor 229
 creating new commands 225–227
 Custom Commands module (Usermin) 639
 custom commands, parameter types of 227–228
 custom error messages 287–288
 Custom error responses table 287
 Custom logfiles table 517
 Custom width x height field 222
 Cut button 234

- CVS (Concurrent Versions System)
 354
CVS root directory field 355, 360
CVS Server module 354
 adding and editing users
 356–357
 browsing the repository 359
 configuring 359
 configuring CVS servers 359
 limiting user access 358–359
 setting up CVS servers 355–356
 synchronizing userlist with
 UNIX user database
 357–358
CVS servers 354
 adding and editing users
 356–357
 configuring 359
 limiting user access 358–359
 setting up 355–356
 synchronizing userlist with
 UNIX user database
 357–358
 using 356
 viewing files and directories in
 repository 359
cvspserver service 355
- D**
- daemons
 facility 114
 Fetchmail 380, 384–385
 Hostsentry 256
 monitoring status of 250–251
 Portsentry 257
Data type field 409, 432
Database file path field 431
Database name field 408, 431
databases 405–406
 adding tables 408–409
 backing up 417–418, 437–438
 creating 407–408, 431
 deleting 416–417, 437
 permissions 421–423
 restoring 418–419, 438
 viewing and editing table
 contents 412–416
Databases field 422
Date and Time ACL (access control
 list) 588
date field 410, 414, 435
date_chooser_button function 752
datetime field 410, 414
daytime service 130, 132
- DBD::mysql module 244
DBD::Pg module 244
DBI interface 426
DBM files 292
Deactivate CVS Server button 356
Debian Linux 6
 firewall configuration 175
 installing packages 108
 Majordomo package in 390
 software packages 106
 updating on 110–111
debug priority 114
decimal field 410–411, 413
decode_base64 function 753
Default backup repository directory
 field 447
Default boot option field 203
default gateways 144, 149–150
Default hostname for From:addresses
 field 634
Default kernel/partition field 199
default language 724
Default lease time field 372
Default master server(s) for slave
 zones field 351
Default MIME type field 289
Default PID file location field 352
default quotas 65
Default remote slave server field 351
Default route device field 149
Default router field 149, 366
Default Server icon 266, 268
default servers 504
Default time-to-live field 321, 330,
 339
DEFAULT variable 611
defaultuconfig file 733, 752
Delete button 234
delete command 542
Delete job command field 567
delete_cron.cgi function 734, 735
delete_cron_job function 736, 737
Delivery mode field 458
Deny client hosts field 546
Deny from address field 541
Deny groups field 515
Deny users field 515
Department field 16
depends file 711
Description width option 275
deselect commands 106
Dest AS Number ACL (access control
 list) 588
- Destination address or network
 condition 187
Destination TCP or UDP port
 condition 187
Destination TCP or UDP port field 186
DHCP (Dynamic Host Configuration
 Protocol) 145,
 361–362
DHCP server config file field 376
DHCP server executable field 376
DHCP server lease file field 377
DHCP Server module 363–365
 access control 374–375
 adding and editing groups
 373–374
 adding and editing shared
 networks 372–373
 adding fixed hosts 370–371
 adding subnets 365–367
 configuring 375–377
 deleting subnets 367
 editing fixed hosts 371
 editing global client options 370
 viewing and deleting leases 369
DHCP Server monitor 255
DHCP servers 361
 adding and editing groups
 373–374
 adding and editing shared
 networks 372–373
 adding fixed hosts 370–371
 adding subnets 365–367
 configuration files 365
 deleting subnets 367
 editing fixed hosts 371
 global client options 370
 viewing and deleting leases 369
dhcpd program 364
dhcpd.conf file 362, 376
dhcpd.leases file 362
dhcp-server program 364
dial-in modems 166
dial-up connection 3
Digest mail footer field 400
digest mailing lists 399–400
Digest title field 400
Direct TCP connection option 209
Direction to monitor parameter 257
Directive types available field 311
directives 264–265
 restricting 299
 viewing and editing 308–310
directories

- access control lists (ACLs) 239–240
 for CGI scripts 280
 creating 236
 exporting 54–55, 241–242
 mapping URL paths to 276–277
 navigating 232–233
 ownership 236
 permissions 236
 protecting 289–291
 setting options for 274
 sharing 240–241
 turning off authentication 292
 user web 300–301
- directory aliases 535–536
 Directory containing majordomo programs field 403
 Directory containing sendmail safe programs field 404
 Directory for master zone files field 351
 Directory for slave/stub zone files field 351
 Directory index footer file field 276
 Directory index header file field 276
 Directory index options field 275
 Directory index pages field 494
 Directory Indexing icon 274, 276
 directory listings 510–511
 Directory options 299
 Directory parameter 228
 Directory path field 513
 Directory README filename field 511
 Directory to backup field 123
 Directory to share field 559
 Disk and Network Filesystems module 40
 access control 49–50
 adding virtual memory 46–47
 configuring 50
 editing filesystems 48
 listing users of filesystems 48
 mounting local ext2 or ext3 hard disk filesystem 44–45
 mounting local Windows hard disk file system 45–46
 mounting NFS network filesystem 40–43
 mounting smbfs Windows networking filesystem 43–44
- support for other operating systems 51–52
 Disk Quotas module 61–62
 access control 66–67
 configuration page 12
 configuring 66
 disabling quotas for filesystems 62
 enabling quotas for filesystems 62
 example page 13
 main page 12
 setting default quotas for new users 65
- Disk Quotas module (Usermin) 639
 disk quotas. See quotas
 Disk Space monitor 255
 disk_usage_kb function 753
 Display directories first option 275
 Display fancy directory indexes option 275
 Display HTML title as description option 275
 Display lease times in field 376
 Display queue command field 567
 Display subnets and hosts as field 376
 Display virtual servers as field 312
 DNS (Domain Name System) 315–318
 DNS clients 317
 DNS domain 361
 DNS lookup cache time field 584
 DNS servers 144–145
 addresses 361
 BIND 254
 forward zones 336–337
 settings 151
 zones 316
 DNS servers field 367
 DNS servers for clients field 168
 Document directory aliases table 277
 Document Root field 270, 306
 Domain for reverse IPv6 addresses field 352
 Domain name field 367
 Domain name/Network field 321, 333
 domains 303, 315
 local 451–452, 478–479
 masquerading 452
 names 588
 parent 316
 root 316
 routing 457–458, 483
 Domains not to cache field 303
- double field 410–411, 413
 Down state 251
 dpkg command 106
 DPKG package format. 106
 dump command 123
 Dump level field 123
 dump program 121
 Dynamic DNS update style field 367
 Dynamic Host Configuration Protocol (DHCP) 145, 361–362
- E**
- echo 132
 echo service 130
 Edit Apache directive button 309
 Edit CA certificate field 687
 Edit Config Files icon 309
 Edit Directives in File button 310
 Edit NIS table button 159
 Edit Quota For field 63
 edit_allow.cgi function 735
 edit_cron.cgi function 734
 EDITOR environment variable 737
 edquota command 61
 elm program 378
 emacs command 309
 email
 address mappings 456–457
 addresses 16
 aliases 452–455, 479–480
 autoreply aliases 465–466, 480
 domain routing 457–458, 483
 downloading 384
 filter aliases 466–467, 480
 filtering 605–606
 forwarded 396–397
 queue 460–461
 reading 28, 461–462, 486–487
 relaying 455, 480
 virtual address mappings 481–483
- Email Address field 305, 308
 Email address field 16, 321, 328
 Email message subject field 124
 emerg priority 115
 emerge command 106
 Emerge package format (Gentoo) 106
 emerge webmin command 6
 emote_foreign_call function 730
 encode_base64 function 753
 encodings 295
 Encryption method for proxy
 passwords field 602

- enum field 410, 415
err priority 114
Error code field 287
error function 753
error log files 282
Error log to field 285
Error message file field 541
error messages 287–288
error_setup function 753
Escape character field 551
etc/aliases file 391
etc/cron.allow file 734
etc/cron.d directory 93
etc/cron.deny file 734
etc/crontab file 93
etc/exports file 53, 57
etc/fstab files 40
etc/ftpaccess file 525
etc/ftphosts file 525
etc/ftpusers file 525
etc/inittab file 84, 215
etc/lilo.conf file 196
etc/hsswitch.conf file 155
etc/passwd file 419
etc/ppp/chap-secret file 171
etc/procmailrc file 614
etc/shadow file 419
etc/syslog.conf file 115
etc/vfstab files 40
etc/xinetd.conf configuration file 130
etc/xinetd.d directory 130
etc/yp.conf file 155
etc/ypserv.conf file 155
Ethernet address 188, 362, 370
Ethernet Address ACL (access control list) 588
Ethernet interface 145
Ethernet network cards 144, 370
Eudora mail 131
Event types to log in history field 359
everyone views 345
Evolution mail 448
exec service 132
exec_cron.cgi function 735
executable files 236, 522
Execute as group field 236
Execute as user field 135, 236
Execute command button 222
Execute Command monitor 255
Execute previous command button 222
Exit chain action 181
Expiry date option 23
Expiry time field 321, 330, 339
exporting directories 54–55, 241–242
exports file 53
EXT attributes 237–238
ext2 filesystem 44–45, 237–238
ext3 filesystem 44–45, 50, 237–238
Extended Internet Server monitor 255
extended Internet services 139
 creating 141
 editing 140–141
 enabling 140–141
Extended Internet Services module 139, 355
extended partitions 68
External Auth ACL (access control list) 589
External Auth Rexexp ACL (access control list) 589
external modems 215
Extra character sets table 295
Extra command-line parameters field 123
Extra mail queue directories field 475
Extra MIME types table 283, 289
Extra SMTP headers for resent mail field 394
- F**
- facilities 113
Fail if process is parameter 255
Failure URL field 587
Fallback boot option field. 203
fast_wait_for function 753
fBuilder tool 176
Fetchmail 378–379, 449
fetchmail command 384
Fetchmail config file to edit field 379, 387
Fetchmail Mail Retrieval module 379–380
 access control 386
 adding mail servers to check 381–384
 configuring 386–387
 downloading email 384
 editing global settings 385
 running Fetchmail daemon 384–385
Fetchmail Mail Retrieval module (Usermin) 639
.fetchmailrc configuration file 379
Field name field 409, 431
Field options field 432
fieldPath to MySQL shared libraries directory 427
fields 409
 adding 409–411, 433
 deleting from tables 433–434
 editing 411–412, 433
 permissions 421–423
 types 412, 434–435
File Change monitor 256
file editors 225, 229
File Manager module 232
 access control 242–243
 creating directories 236
 creating files 234–235
 creating links 236
 editing EXT file attributes 237–238
 editing file ACLs 239–240
 editing file permissions 235–236
 editing files 234–235
 editing xfs filesystem 238
 exporting directories 241–242
 finding files 237
 manipulating files 234
 navigating directories 232–233
 sharing directories 240–241
 viewing files 232–233
File Manager module (Usermin) 639
File must be bigger than parameter 254
File must be smaller than parameter 254
File must exist parameter 254
File must not exist parameter 254
File or tape device field 123
File ownership field 229
File parameter 228
file permissions 564–565
 editing 229, 235–236
 mask 46
File permissions field 229
file sharing
 editing 566–567
 permissions 564–565
 with Samba 559–560
 in Windows systems 43–44, 240–241
 with NFS 53
File to add virtual servers to field 270, 312
File to check parameter 254
File to edit field 229
File to monitor parameter 256
File to upload field 234
File Transfer Protocol (FTP) 500–501
File types to report on field 494
file_chooser_button function 753

- Filename width option 275
 files
 access control lists (ACLs) 239–240
 attributes 238
 compressing 238
 copying 234
 creating 234–235
 deleting 234
 editing 234–235
 executable 236
 EXT attributes 237–238
 finding 237
 hiding 510
 manipulating 234
 mapping URL paths to 276–277
 moving 234
 quota 60
 renaming 235
 setting options for 274
 uploading 234
 viewing 232–233
 Files to ignore in directory index field 275
 Filesystem Backup module 121–122
 adding backups 122–124
 configuring 126
 editing or deleting backups 125
 making backups 124–125
 in other operating systems 128
 restoring backups 125–126
 Filesystem to check parameter 255
 filesystems 39–40
 automounter 47
 backups 121
 checking 255
 comparison of 50–51
 creating 70–71
 diskspace 255
 editing 48
 ext2 44–45, 237–238
 ext3 44–45, 237–238
 file permissions 235–236
 listing users of 48
 NFS 40–43
 quotas 60
 disabling 62
 enabling 62
 removing 48
 smbfs Windows networking 43–44
 Windows 45–46
 xfs attributes 238
 filter aliases 466–467, 480
 filter_javascript function 753
 find_byname function 753
 finger 131, 226
 firewall 173
 creating custom chains 182
 editing rules in 182
 implementations 173
 IPTables 173–174
 port forwarding 185–186
 rule conditions 186
 setting up NAT (Network Address Translation) 183–184
 setup options 177
 transparent proxying 184–185
 Fixed IP address field 371
 float field 410–411, 413
 float4 field 435
 float8 field 435
 floppy disks 47
 floppy drives 195
 flush_file_lines function 753
 flush_webmin_caches function 753
 Font size in points field 222
 foobar 711
 footer function 754
 foreign_call function 754
 foreign_check function 754
 foreign_config function 754
 foreign_require function 754
 Format for the name of forward zone
 files field 351
 Format for the name of reverse zone
 files field 351
 FORWARD chain 174
 .forward files 455
 forward zones 336–337
 forwarded email 396–397
 Forwarded mail footer field 392
 Forwarded packets chain 177
 Fragmentation condition 187
 Free Memory monitor 256
 FreeBSD operating system 2
 boot loader 195, 201–202
 bootup scripts 90
 filesystem 52
 filesystem backups 128
 Internet service 138
 network configurations 153
 NFS exports on 57–58
 package format 112
 print system 214
 system logs 119
 system time 193
 users and groups in 37
 From CPAN option 246
 From:address mapping file (Usermin module) 634
 FTP (FileTransfer Protocol) 500–501
 FTP commands field 514
 FTP proxy field 672
 FTP server PID file field 543
 FTP servers 500–501
 anonymous 506–507
 commands 514–517, 541–542
 downloading from 338
 virtual 505–506
 ftp service 132
 FTP transfers logfile field 517
 ftp_command function 754
 ftp_download function 752, 754
 ftppasswd file 525
 ftpconversions file 543
 ftpgroups file 543
 ftphosts file 525, 543
 ftpusers file 525, 543
 Full path to ftppasswd file field 543
 Full path to ftpconversions file field 543
 Full path to ftpgroups file field 543
 Full path to ftphosts file field 543
 Full path to ftpusers file field 543
 Full path to M4 config file field 473
 Full path to majordomo config file field 403
 Full path to nmbd field 575
 Full path to PID file field 603
 Full path to sendmail aliases file field 473
 Full path to sendmail pid file field 473
 Full path to sendmail.cf field 472
 Full path to smbd field 575
 Full path to smbpasswd field 575
 Full path to smbstatus field 575
 Full path to squid cache directory field 603
 Full path to squid config file field 602
 Full path to squid log directory field 603
 Full path to ssh client config file field 553
 Full path to sshd config file field 553
 Full path to sshd PID file field 553
 Full path to sshd program field 553
 Full path to ssh-keygen program field 553
 Full path to swat field 576

- Full path to the named executable field 352
Full path to the named.conf file field 352
Full path to whois command field 352
Full path to wuftpd field 543
- G**
- %gconfig variable 713
GD module 244
generate_icon function 755
generators 344
Gentoo Linux 6
 firewall configuration 175
 package system 106
get command 397
get list file command 389
Get root servers from field 338
get_all_module_infos function 729, 752, 755
get_available_module_infos function 743, 748, 755
get_miniserv_config function 755
get_module_acl function 755
get_module_info function 752, 755
get_system_hostname function 729, 755
get_theme_info function 755
get_webmin_version function 755
getfacl command 239
ghostscript PostScript rendering program 205
GID parameter 228
GIF files 277, 521
Global address book file field 635
GnuPG Encryption module (Usermin) 639
Google search engine 757
grace periods 64–65
Graphical mode runlevel 85
Greeting level field 537
greeting messages 219
Greeting Messages icon 219
Group DBM file field 292
Group ID field 441
Group name field 291, 664, 695
Group owner of uploaded files field 513
Group parameter 228
Group text file field 290
Group to start BIND as field 348
group_chooser_button function 755
groups 19, 158
- ACLs (access control lists) 665–666
adding 373–374
creating 25–26, 441
deleting 27
editing 26, 374
ID 29
ISC DHCP server 362
password protection 289–291
secondary 19, 23, 26
setting quotas for 63
- Groups visible in group chooser field 694
- GRUB boot loader 84, 200
 boot options 196
 booting another operating system 202
 booting new kernel with 201–202
 editing global options 202–203
 hardware compatibility 195
 installing 203
- GRUB Boot Loader module 200
 booting another operating with 202
 booting new Linux kernel or BSD 201–202
 configuring 203
 editing global GRUB options 202–203
 installing GRUB 203
- .gz encoding format 294
gzip command 530
- H**
- Hard Block Limit fields 63
hard disks 39
 ext2 filesystem 44–45
 ext3 filesystem 44–45
 partitions 68–69
 Windows filesystem 45–46
- Hard File Limit field 63
hard limit 60–61
Hardware address field 146, 370
Hardware category
 Linux Bootup Configuration module 196–197
Partitions on Local Disk 69
Printer Administration module 206
System Time 191
Voicemail Server module 215–216
- hardware clock 191
- hardware time 191
 changing 192–193
 synchronizing with system time 193
 synchronizing with time of another server 193
- has_command function 756
- hba.conf file 430
<head> tags 275
header function 756–757
Headers and Footers icon 394
Headers to show in mail queue field 471
- headhtml 742
headinclude 742
Heap tables 408
Help link 11
help_file function 757
help_search_link function 757
Hide files owned by groups field 510
HINFO (Host Info) record 327
hlink function 718
home directory 19, 21, 29
Home directory field 160, 485
HOME variable 611
host addresses 151
Host addresses table 161
Host assigned to field 371
Host Information (HINFO) record 327
Host name field 370
Host to connect to parameter 259
Host to ping parameter 258
HOST variable 612
hostname 150
 changing 150–151
 IP addresses for 315, 317–318
 pinging 258
 proxy servers 303
 TCP connection 259
- Hostname access control option 299
Hostname field 329
Hostname for email to local IP address field 484
Hostname for messages field 537
Hostname for resent email field 397, 401
Hostname for SMTP HELO field 484
Hostname pattern field 344
Hostname to connect to field 221
hosts 55, 158
 adding 370–371
 assignment 365
 displaying 376
 editing 371

- file structure 365
 IP addresses 365, 376
 ISC DHCP server 362
 MAC addresses 365
 permissions 421–423
 Hosts field 422
 Hostsentry Daemon monitor 256
 Hotmail 379
 HP/UX operating system 2
 bootup scripts 89–90
 filesystem 52
 Internet service 139
 package format 112
 print system 214
 system time 193–194
 users and groups in 37
 HPUX print system 572
 .htaccess files 265
 creating 298
 directory options 273
 editing 297
 restricting directives in 299
 HTML files 283
 <html> tags 225, 275
 html_escape function 757
 HTTP (Hypertext Transfer Protocol) 258
 HTTP Accel Host field 185
 HTTP Accel Port field 185
 HTTP Accel Uses Host Header field 185
 HTTP Accel With Proxy field 185
 HTTP connections 223
 HTTP proxy 301
 HTTP proxy field 672
 HTTP servers 264, 577–578
 http_download function 752, 757
 HTTP_USER_AGENT environment variable 676
 httpd script 85
 httpd.conf file 311, 314
 HTTPS protocol 304
 HUP signal 101
 Hypertext Transfer Protocol (HTTP) 258
- |
- IA64 systems 105
 IBM Journaling Filesystem (jfs) 51
 ICMP packet type condition 188
 ICMP packets 177, 258
 Icon height option 275
 Icon width option 275
 Icons in row field 376
- icons_table function 758
 ICP port field 581, 597
 IDE disks 69
 IDE drives 71, 200
 Idle time before disconnect field 168
 image files 521
 images/icon.gif file 711
 IMAP (Internet Message Access Protocol) 378
 imap service 131
 Inactive days field 23
 inbody 742
 include files 612–613
 include function 758
 Incoming interface condition 187
 Incoming packets chain 177
 Incoming UDP address field 581
 index command 397
 index list command 389
 index.cgi file 712
 index.cgi program 734, 748
 index.html files 268, 274, 712
 Indexing and index files option 299
 indexof function 758
 inetd program 193
 inetd super-server 129, 503–504,
 527–528
 Info button 235
 info command 397
 info list command 389
 info priority 114
 init program 84, 196
 &init_config function 713
 init_config function 724, 758
 Initial fields table 408, 431
 Initial physical device field 79
 Initial PostgreSQL database field 445
 Initial ramdisk file field 201
 initlocation command 431
 INPUT chain 174
 INSERT statement 418
 Install action field 246
 install-module.pl script 674
 int field 411, 413
 int2 field 435
 int4 field 435
 int8 field 435
 Interface to monitor parameter 257
 interfaces 144
 Interfaces file type field 377
 internal clocks 191
 internal modems 215
 internal views 345
 Internet and RPC Server monitor 256
- Internet Message Access Protocol (IMAP) 378
 Internet services 129–130
 creating 133–135
 enabling 133
 names 131–132
 protocols 131–132
 Internet Services and Protocols module 130
 configuring 136–137
 creating and editing RPC programs 135–136
 creating Internet services 133
 133–135
 enabling Internet services 133
 in other operating systems 138–139
 service names and protocols 131–132
 intro command 397
 intro list command 389
 Introductory message field 392, 394,
 400
 IP Access Control 14
 IP Address field 147
 IP addresses 145
 changing 146, 670–671
 for hostnames 315, 317–318
 hosts 365, 376
 pinging 258
 proxy servers 303
 restricting 14–15
 restricting access by 293–294
 restricting ProFTPD clients by 520
 restricting WU-FTP clients by 541
 subnets 365
 TCP connection 259
 IP network 303
 IP-based websites 269
 IPchains firewall 173
 IPfwadm firewall 173
 IPs and ports for DNAT field 186
 iptables command 175–176
 IPTables firewall 173–174
 iptables-restore command 175
 iptables-save command 175
 Irix operating system
 bootup scripts 89–90
 filesystem 52
 filesystem backups 128
 Internet service 138
 NFS exports on 59

- print system 214
system logs 119–120
system time 193–194
users and groups in 37
xfs filesystem 238
- is_under_directory function 758
ISAM tables 408
ISC DHCP servers 362–363
iso-9660 filesystem 39
ISPs (Internet service providers) 144
- J**
- Jabber IM Server monitor 256
Java applets 220
file manager 232
remote shell login 223
- Jaz drive 73
- JPEG files 277, 521
- K**
- Keep track of read/unread emails field 471
kern facility 114
Kernel options field 198, 201
Kernel to boot field 198
kernels 195
2.4 series of 173
booting 197–198
loading 196
system time 191
key.pem file 305
Kill Process button 101
Kill Processes button 49
kill_byname function 759
kill_byname_logged function 759
kill_logged function 727, 759
knetfilter tool 176
- L**
- lang internationalization directory 738
lang_list.txt file 724
languages 296
LANs (local area networks) 366
laptops 145
Latitude and Longitude field 328
LD_LIBRARY_PATH environment variable 676
leases 362–363
deleting 369
sorting 376
viewing 369
lib subdirectory 530
Library search path field 676
LILO boot loader 84
- boot options 196
booting another operating system with 198–199
booting new kernel with 197–198
configuring 196–197
editing global options 199–200
hardware compatibility 195
limitations of 196
- lilo command 196
Limit files to directory field 311
Limit to directory field, 507
Limit users to directories field 508
linear RAID array 75
Link type field 702, 730
links 236
Linux Boot Loader module 198–200
Linux Bootup Configuration module 196–198
- Linux distributions 6
boot loaders 196
choosing time zone 191
printer administration 206
quota commands 62
- Linux Firewall module 175–177
allowing and denying network traffic 177–178
changing default actions of chains 181
configuring 189
creating custom chains 182
editing firewall rules 182
setting up NAT (Network Address Translation) 183–184
setting up port forwarding 185–186
setting up transparent proxy 184–185
- Linux kernel 201
Linux Native Filesystem 44–45
Linux operating system
package format 111
xfs file system 238
- Linux raid 70
Linux RAID module 75–77
Linux swap 70
Linux systems 1
as answering machines 215–216
boot loader 195–196
boot process 84–85
clocks 191
filesystems 39
- hardware and system times 193
logs 113–115
MAC (Medium Access Control) address 370
networking 144–145
PPP sessions in 165–166
printing on 205–206
processes 99
rebooting 89
software packages 105–106
- List description field 394
List maintainer's address field 392
List name field 400
list.cgi program 714
list_allowed function 736
list_cron_jobs function 734, 735
list_denied function 736
list_languages function 759
list_usermods function 759
list_users function 713
Listen on addresses and ports table 270
Listen on IP address field 623, 671
Listen on port field 671
listname command 394
lists command 389, 398
Load Average monitor 256
Load average to check parameter 256
load_language function 729, 759
load_theme_library function 759
local area networks (LANs) 366
local domains 451–452, 478–479
Local host name field 484
Local user option 116
Local user(s) field 382
localfile parameter 762
Locality Name field 305, 308
locate command 93
Location (LOC) record 328
lock_file function 725, 739, 759
Log all commands for field 538
log file directory 9
Log Files icon 285
Log transfers to field 538
log_parser.pl file 727
log_parser.pl log reporting script 739–740
- Logged In Users button 28
logical partitions 68, 77
logical volume 77
creating 80–81
deleting 81
resizing 81
- Logical Volume Manager (LVM) 77–78

- Logical Volume Manager module 78
 adding and removing physical volume 80
 creating and deleting logical volume 80–81
 creating new volume groups 79–80
 creating snapshots 82–83
 resizing logical volume 81
- login 132
 name 9, 29
 password 9
 remote shell 220–223
 viewing 27–28
 viewing and disconnecting sessions 697
- Login as parameter 258
- .login file 511
- Login name field 43
- Login password field 43, 357
- Login Scripts module (Usermin) 640
- Login to MySQL as field 424
- Login username field 357
- LOGNAME variable 611
- Logout message file field 511
- logs 113–115
 access to 285
 adding 115–117
 analyzing 491–492
 configuring 284–285, 517–518, 595–596
 custom formats 518
 deleting 117–118
 displaying 708–709
 editing 117–118
 error messages 285
 format codes 518
 generating and viewing reports of 496
 predefined formats 285
 scheduled reporting of 496–497
 setting up 671
 for virtual servers 285–286
- longblob field 415
- longtext field 415
- loopback interface 145
- Lotus Notes 379
- lpr facility 114
- LPR print system 206
- LPRNG print system 206, 212, 572
- ls command 530
- ls program 223
- LVM (Logical Volume Manager) 77–78
- LVM logical volume option 45
- M**
- M4 configuration file 463–465, 606
- MAC (Medium Access Control)
 address 362
 determining 370
 hosts 365
- mail aliases 452–455, 479–480
- mail clients 448
- Mail delivery command field 387
- mail directory 479
- mail facility 114
- mail file 479
- Mail file directory style field 475, 490
- Mail file in home directory field 474
- Mail for field 456
- Mail for host or domain field 483
- Mail Forwarding module (Usermin) 640
- Mail messages to display per page field 470, 488
- mail retrieval protocol 131
- Mail Server (MX) record 316, 326, 449
- Mail server hostname field 401
- Mail server to contact field 381
- mail servers 378
 adding 381–384
 clients 448
 global settings 385
- Mail storage format field 633
- Mail subdirectory style field 633
- Mail Transfer Agent (MTA) 448–449
- Mailbox format field 488
- mailing lists 388–389
 access control 397–398
 creating 391–392
 deleting 399
 digest lists 399–400
 digest options 400–401
 footers 393–395
 forwarded email options 396–397
 global options 401
 headers 393–395
 managing member list 392–393
 moderating and maintaining 398–399
 subscription options 395–396
- Maintenance password field 392, 396, 400
- Majordomo 388–389
- Majordomo List Manager module 389–391
 access control 401
 configuring 402–404
 creating digest lists 399–400
 creating mailing lists 391–392
 deleting mailing lists 399
 editing digest options 400–401
 editing forwarded email options 396–397
 editing global options 401
 editing information, headers and footers 393–395
 editing list access control 397–398
 editing subscription options 395–396
 managing mailing list 392–393
 moderating and maintaining mailing lists 398–399
 using other mail servers 391
- Majordomo master address field 401
- Majordomo owner's address field 401
- make command 246
- make install command 246
- make script 155
- make test command 246
- make_date function 759
- make_http_connection function 752, 759
- Makefile.PL arguments field 246
- Makemap command field 473
- Manager name field 600
- Mandrake Linux 6, 106
- Mangle field 162
- Manually edit directives button 309
- Map locale to remote URLs table 279
- Map remote Location: headers to local table 279
- Masquerade option 180
- master boot record (MBR) 84, 195
- Master server field 321, 330, 333
- master servers 316, 321, 330, 333
- master zones 316
 adding records in 322–324
 converting slave zones to 336
 converting to slave zones 332
 creating 321–322
 deleting 331
 editing 330–332
 editing records in 324
- Match ACLs list 586

- Max digest size before sending field 401
Max load average for receiving field 459
Max load average for sending field 459
Max per Minute field 135
Maximum allowable message size field 397
Maximum bytes/second parameter 257
Maximum cache time field 584
Maximum cached object size field 584
Maximum concurrent logins field 520
Maximum concurrent logins per host field 520
Maximum concurrent sessions field 520
Maximum concurrent zone transfers field 341
Maximum Connections ACL (access control list) 589
Maximum days field 23
Maximum lease time field 366, 372
Maximum load average parameter 256
Maximum log lines to pass to calamaris field 602
Maximum message size field 484
Maximum number of aliases field 469
Maximum number of records to show in tables field 488
Maximum number of servers to display field 312
Maximum number of zones to display field 349
Maximum reply body size field 583
Maximum request body size field 583
Maximum transfer time field 334
Maximum users field 540
Maximum zone transfer time field 341
MBR (master boot record) 84, 195
mediumblob field 415
mediumint field 413
mediumtext field 415
Members field 291
Memory mode 100
Memory usage limit field 584
Menu parameter 228
message files 511–512, 536–538
Messages and Description icon 394
metadata 239
mgetty program 165–166, 215
MH style directory mail storage format 633
Microsoft Exchange 379
MIME types 288–289
MIME types and encodings option 299
mime.types file 314
Min free disk space field 459
Minimum days field 23
Minimum free memory (in kB) parameter 256
Minimum free space (in kB) parameter 255
Minimum free space field 567
Minimum mail file size to index field 472, 489, 635
Minimum message length field 217
mirrored RAID array 75
mkdir command 431
mod_perl module 282
mod_php module 282
mod_proxy module 302
mod_ssl module 304
modems 215
Moderation password field 396
Moderator's address field 396
Module Config link 11
Module Index 11
Module to monitor field 253
module.info file 711, 733
\$module_config_directory variable 713
%module_info variable 713
\$module_name variable 713
\$module_root_directory variable 713
module_uninstall function 728
modules 2
 access control. See access control
 Apache Web server 265–268
 BIND DNS Server 318–320
 Bootup and Shutdown 85
 Cluster Software Packages 644
 Cluster Users and Groups 649–650
 Cluster Webmin Configuration 660–661
 Command Shell 222–223
 Custom commands 224–225
 CVS Server 354
 deleting 674
 developing 710
 access control 721–722
 action logging 726–727
 CGI programs 712–715
 configuration parameters 715–717
design goals 718
file locking 725
functions 728–730
look and feel 717–718
online help 718–719
packaging 719
pre- and post-install scripts 728
required files 711–712
user update notification 723
DHCP Server 363–365
Disk and Network Filesystems 40
Disk Quotas 61–62
Extended Internet Services 139
Fetchmail Mail Retrieval 379–380
File Manager 232
Filesystem Backup 121–122
Grub Boot Loader 200
installing 673–674
internationalization 723–725
Internet Services and Protocols 130
Linux Bootup Configuration 196–197
Linux Firewall 175–177
Majordomo List Manager 389–391
MySQL Database Server 406–407
Network Configuration 145
NFS Exports 54
NFS Shares 56
NIS Client and Server 155
Partitions on Local Disks 69
Perl Modules 245
PostgreSQL Database Server 429–430
PPP Dialin Server 165–166
Printer Administration 206
Procmail Mail Filter 606
ProFTPD Server 501
Qmail Configuration 477–478
Running Processes 99–100
Samba Windows File Sharing 556
Scheduled Commands 97
Scheduled Cron Jobs 93
Sendmail Configuration 449–451
Software Packages 107
Squid Proxy Server 578–580

- SSH Server 545
 SSH/Telnet Login 220–222
 SSL Tunnels 616
 System and Server Status 250–251
 System Logs 115
 System Time 191
 SysV Init Configuration 91–92
 unauthenticated access to 685–686
 user interface 11
 Usermin Configuration 621
 Users and Groups 20
 Voicemail Server 215–216
 Webalizer Logfile Analysis 491–492
 Webmin Actions Log 708
 Webmin Configuration 669
 Webmin Servers Index 700
 Webmin Users 689
 WU-FTPD Server 526
 Modules for members field 665
 MON Service Monitor monitor 257
 monitors 250–251
 adding 252–253
 load average 256
 scheduled 260–261
 states 251–252
 types 253–260
 Mount Filesystems module (Usermin) 640
 mount points 39
 mountable partitions 74
 MSC cluster groups directory field 706
 MSC Theme 747
 index.cgi program 748
 theme_footer function 749–750
 theme_header function 748–749
 MSC.Linux Mini Theme 684
 MSC.Linux operating system 390
 MSC.Linux Theme 684
 mscstyle3 directory 747
 MTA (Mail Transfer Agent) 448–449
 MTU field 146
 Multiple NFS Servers field 51
 multiple systems 643–644
 adding groups to 654
 adding users to 651–652
 deleting groups from 656
 deleting users from 653–654
 editing groups in 654–655
 editing users in 652–653
 synchronizing users and groups in 656–658
 Multi-user mode runlevel 85
 Multi-user mode without NFS runlevel 85
 MX (Mail Server) record 316, 326, 449
 MyISAM tables 408
 MySQL 244, 405–406
 adding fields 409–411
 adding users 419–420
 backing up databases 417–418
 creating new database 407–408
 creating new tables 408–409
 deleting databases 416–417
 deleting tables 416–417
 editing fields 411–412
 executing commands 417
 field types 412
 managing permissions 421–423
 restoring databases 418–419
 synchronizing UNIX users 421
 viewing and editing table contents 412–416
 website 406
 MySQL Database module (Usermin) 640
 MySQL Database Server module 406–407
 access control 423–424
 adding fields 409–411
 adding MySQL users 419–420
 backing up databases 417–418
 configuring 424–426
 creating new database 407–408
 creating new tables 408–409
 deleting databases 416–417
 editing fields 411–412
 executing SQL command 417
 managing permissions 421–423
 restoring databases 418–419
 setting up synchronization 421
 viewing and editing table contents 412–416
 MySQL Database Server monitor 257
 MySQL host to connect to field 427
 MySQL port to connect to field 427
 MySQL socket file field 427
 mysqlshow program 427
- N**
- Name Alias (CNAME) record 326, 329, 342–344
 Name field 329
 name file 711
 Name record 326
 Name Server (NS) record 316, 326
 Name Server record 329
 name servers 319–320
 name-based websites 269
 Named log format table 285, 286
 named pipes 116, 121
 named.conf file 330, 352
 NAT (Network Address Translation) 174
 setting up 183–184
 Net::SSLeay Perl module 7, 15–16, 630
 Net::Telnet module 244
 NetBIOS name servers field 367
 NetBSD operating system
 boot loader 201–202
 bootup scripts 90
 Internet service 138
 network configurations 153
 NFS exports on 57–58
 package format 112
 print system 214
 system logs 119
 system time 193
 users and groups in 37
 netgroups 55, 158
 NetInfo database 37
 netmask 55, 147, 294
 Netmask field 147–148, 366, 585
 Netware filesharing protocol 53
 Network address field 366
 network cards 144
 Network Configuration module 145
 access control 152–153
 adding network interfaces 147–149
 changing DNS client settings 151
 changing hostname 150–151
 configuring routing 149–150
 editing host addresses 151
 in other operating systems 153
 viewing and editing network interfaces 146–147
 Network Information Service. See NIS
 network interfaces 144–145
 adding 147–149
 monitoring 257
 viewing and editing 146–147
 Network name field 372
 Network protocol condition 187
 Network Time Protocol (NTP) 193

- network traffic 173
allowing 178
blocking 177–178
- Network Traffic monitor 257
- Networking category
 Extended Internet Services
 module 139
- Internet Services and Protocols
 module 130
- Linux Firewall module 175–177
- Network Configuration module
 145
- NFS Exports module 54
- NIS Client and Server module
 155
- PPP Dialin Server module
 165–166
- SSL Tunnels module 616
- networks 55
 proxy servers 303
 security 14–15
 shared 362
- New document button 234
- New Linux Native Filesystem 44–45,
 50
- news facility 114
- NFS Directory field 41
- NFS exports
 deleting 55–56
 editing 55–56
 exporting directories 54–55
 on operating systems
 FreeBSD 57–58
 Irix 59
 NetBSD 57–58
 OpenBSD 57–58
 OpenServer 57–58
 OS X 57–58
 Solaris 56–57
- NFS Exports module 54
- NFS file sharing enabled option
 241–242
- NFS filesystem 51
- NFS Hostname field 41
- NFS network filesystem 40–43
- NFS Server monitor 257
- NFS Shares module 56
- nice level 99
- NIS (Network Information Service)
 154
 client 155–156
 creating UNIX groups in 160
 server 154–155
 master 157–158
- securing 160–163
slave 163
tables 159–160
- NIS Client and Server module 155
 configuring 163
 editing NIS tables 159–160
 securing NIS server 160–163
 setting up NIS client 155–156
 setting up NIS master server
 157–158
- setting up NIS slave server 163
- NIS domain field 367
- NIS Server icon 157
- NIS servers field 367
- NIS+ protocol 155
- NIS+ server 156
- nmbd server program 575
- no_proxy function 759
- nobody user account 19
- Not installed state 252
- notice priority 114
- Notify user of readme files matching
 field 512
- NS (Name Server) record 316, 326,
 329
- ntalk service 131
- ntfs filesystem 45–46
- NTP (Network Time Protocol) server
 193
- ntpdate program 193
- null-modem connection 166
- Number of rows to display per page
 field 425, 445
- numeric field 435
- O**
- Octal field 235
- Old Webmin Theme 683
- Oldest message age before sending
 field 400
- online help 718–719
- Only allow client hosts field 546
- Only allow group field 515
- Only allow users field 515
- open_socket function 759
- OpenBSD operating system
 boot loader 201–202
 bootup scripts 90
 filesystem 52
 Internet service 138
 network configurations 153
 package format 112
 print system 214
 system logs 119
- users and groups in 37
- OpenLinux operating system 157, 390
- OpenServer operating system 38
 bootup scripts 89–90
 Internet service 138
 NFS exports on 57–58
 package format 112
 system logs 119
 system time 193–194
- OpenSSH 544
- openssh program 545
- openssh-client program 545
- openssh-server program 545
- openssl command 17, 305, 308
- OpenSSL library 7, 15–16, 545, 630
- Opera browser 625
- operating systems 9
 boot loaders 195
 booting with GRUB 202
 booting with LILO 198–199
 changing 675–676
 filesystems 39–40
 partitions 68
 user information 37–38
- Option parameter 228
- Option title field 201
- Options file can override field 299
- Options for host field 551
- Oracle database 405
- Order to display records in field 348
- Order virtual servers by field 312
- Organization field 16
- Organization Name field 305, 308
- Organizational Unit Name field 305,
 308
- ORGMAIL variable 612
- OS X operating system 214
 bootup scripts 91
 filesystem backups 128
 Internet service 139
 NFS exports on 57–58
 system logs 119
 system time 193
 users and groups in 37
- OS/2 operating system 51
- os_support file 711
- other_groups function 760
- Others category
 Command Shell module
 222–223
- Custom Commands 224–225
- File Manager 232
- Perl Modules 245

- SSH/Telnet Login module 220–222
 System and Server Status module 250–251
 Outgoing interface condition 187
 Outgoing packets chain 177
 Outlook mail 131, 448
 OUTPUT chain 174
 Output HTML header tags option 275
 overwrite command 542
 Owned by group field 237
 Owned by user field 237
 Owner of uploaded files field 513
 Owner's email address field 396
- P**
- package formats 7
 RPM 7
 Solaris 7, 10
 tar.gz 8–10
 Package from APT option 108
 Package from Red Hat Network option 108
 package tree 109
 Packet burst rate 188
 Packet flow rate condition 188
 packets
 alteration 174
 default action 181
 filtering 174
 forwarded 177
 ICMP 177
 incoming 177
 masquerading 180
 outgoing 177
 .packlist file 247
 PAM Authentication module 529
 PAP secrets file 171
 paper size 209
 parent domains 316
 Parent group field 664
 parity RAID array 75
 parse_times_input function 736
 parse_webmin_log function 726
 partial reverse delegation 342–343
 partitions 68–69
 adding and formatting 70
 deleting 72–73
 extended 68
 labels 71–72
 logical 68, 77
 mountable 74
 primary 68
 writable 74
 Partitions on Local Disks module 69
 access control 73–74
 adding and formatting partitions 70
 creating filesystems 70–71
 deleting partitions 72–73
 editing partitions 72–73
 labeling partitions 72
 Linux vs. Solaris operating systems 74
 passwd command 223, 292
 password
 for directories 289–291
 expiry date 23
 login 9
 mailing lists 398–399
 NIS tables 158
 types of 22
 users 29
 Password field 22
 Password for proxy field 672
 Password parameter 228
 Password program field 571
 Paste button 234
 Path to access.conf field 314
 Path to calamaris log analysis program field 603
 Path to cvs executable field 355
 Path to default stunnel PEM file field 619
 Path to DHCP server PID file field 377
 Path to ftpusers file field 523
 Path to host access config file field 446
 Path to httpd executable field 313
 Path to httpd.conf field 314
 Path to majordomo wrapper field 403
 Path to mime.types field 314
 Path to mysql command field 426
 Path to mysqladmin command field 426
 Path to mysqldump command field 426
 Path to mysqlimport command field 426
 Path to mysqlshow command field 426
 Path to pg_dump command field 447
 Path to pg_restore command field 447
 Path to PostgreSQL shared libraries field 445
 Path to postmaster PID file field 446
 Path to ProFTPD config file field 523
 Path to ProFTPD executable field 523
 Path to ProFTPD PID file field 523
 Path to program field 618
 Path to psql command field 445
 Path to srm.conf field 314
 Path to stunnel executable field 619
 Path to system procmailrc file field 614
 Path to the apachectl command field 314
 Path to the fetchmail daemon PID file field 387
 Path to the fetchmail program field 387
 Path to webalizer command field 499
 Path to webalizer configuration file field 499
 Pause job command field 567
 Per-Directory Options Files icon 297
 Perform distribution upgrade option 110
 Perl executable 9
 Perl interpreter 9
 perl Makefile.PL command 246
 Perl modules 247–248
 configuring 248
 downloading 3
 installing 245–247
 names of 244
 viewing 247
 Perl Modules module 16, 245
 Perl programming language 244, 280
 Permanent URL redirects 278
 permissions 235–236
 editing 564–565
 MySQL users 421–423
 Permissions field 240
 Permissions for majordomo files field 403
 Per-share ACLs table 574
 pg_dump command 447
 pg_restore command 447
 Phone number field 171
 PHP programming language 280
 physical volume 77, 80
 PID mode 100
 pine program 378
 plain text 205
 Plan File module (Usermin) 640
 Playback volume level field 217
 Point-to-Point Protocol. See PPP
 POP (Post Office Protocol) 378
 POP3 or IMAP server name field 634
 pop3 service 129, 131
 port forwarding 185–186
 Port Number field 134
 Port speed field 166

- Port to connect to field 221, 551
Port to connect to parameter 259
Portsentry Daemon monitor 257
POSIX ACLs 239
postbody 742
Postfix mail server 391
Postfix Server monitor 257
postgres user 439
PostgreSQL 428–429
 client access 441–442
 connecting to and managing 244
 databases 431
 field types 434–435
 groups 441
 object privileges 442–443
 syntax 405
 tables 431–432
 users 439–440
PostgreSQL Database Server module 429–430
 access control 443–444
 adding and editing fields 433
 backing up databases 437–438
 configuring 444–446
 creating new databases 431
 creating new tables 431–432
 deleting databases 437
 deleting tables 436
 editing object privileges 442–443
 executing SQL commands 437
 managing groups 441
 managing users 439–440
 restoring databases 438
 restricting client access 441–442
 setting up synchronization 440
 viewing and editing table contents 436
PostgreSQL Database Server monitor 258
PostgreSQL host to connect to field 447
post-install scripts 728
postinstall.pl file 728
Post-login message file field 511
POSTROUTING chain 174
PostScript 205
PPP (Point-to-Point Protocol) 165
 accounts 169–171
 connecting to network via 144
 interfaces 144
 servers 166–169
PPP Dialin Server module 165–166
 access control 172
configuring PPP servers 166–169
managing PPP accounts 169–171
restricting access by caller ID 171
PPP IP Address field 167–168
PPP Options icon 167
pppd program 165–166
pre-encrypted password 22
pre-install scripts 728
Pre-login banner file field 537
Pre-login message file field 511
PREROUTING chain 174
primary groups 19, 23, 26
primary master drive 200
primary partitions 68
primary slave drive 200
Printcap file field 572
printconf tool 206
Printer Administration module 206
 access control 212–213
 adding printers 206–209
 changing print system used by 676
 configuring 211–212
 editing printers 209–210
 managing print jobs 210
 in other operating systems 213–214
Printer driver field 568
Printer status cache time field 572
printers 205–206
 access to 212–213
 adding 206–209
 configuring 572
 disabling 209
 drivers 209
 editing 209–210
 managing tasks in 210
 sharing 560–562
 upgrading 211–212
PrintHeader function 760
priorities 99, 113
Private key file field 18, 307
proc module 754
process ID 99
processes 99
 killing 101–102
 reprioritizing 101–102
 running 103
 searching for 102
 viewing 101–102
Procmail 605–606
Procmail Mail Filter module 606
 configuring 614
 creating and editing actions 608–610
 creating and editing variable assignments 611–612
 creating conditional blocks 612
 creating include files 612–613
 setting up Sendmail 606–608
Procmail Mail Filter module (Usermin) 640
procmailrc file 614
.profile file 511
ProFTPD server 501
 anonymous 506–507
 client restrictions by IP address 520
 concurrent logins 520
 configuration files 523
 directory listing options 510–511
 FTP commands 514–517
 inetd service 503–504
 log configurations 517–518
 login restrictions 508–509
 message and readme files 511–512
 per-directory options 512–513
 restricting access to directories in 507–508
 starting 501–502
 upload options 521–522
 virtual 505–506
 xinetd service 503
ProFTPD Server module 501
 configuring 523
 configuring logging 517–518
 creating virtual servers 505–506
 limiting concurrent logins 520
 limiting uploads 521–522
 manually editing directives 523
 restricting access to FTP commands 514–517
 restricting clients by IP address 520
 restricting users to their home directories 507–508
 setting directory listing options 510–511
 setting login restrictions 508–509
 setting message and readme files 511–512

- setting per-directory options 512–513
 setting up anonymous FTP 506–507
 setting up inetd service or xinetd service 503–504
 using 504–505
P
 ProFTPD Server monitor 258
 Program search path field 676
 progress_callback function 758, 760
 protocols 131–132
 Proxy authentication realm field 594
 Proxy IP Address ACL (access control list) 589
 Proxy Port ACL (access control list) 589
 Proxy port field 597, 598
 Proxy restrictions table 585
 proxy servers 301–304, 577–578, 672
 psql command 430, 439
 PTR (Reverse Address) record 329
 pty_process_exec function 729
 pub directory 530
 put_miniserv_config function 760
 pvftowav command 218
 Python programming language 280
- Q**
 Qmail 476–477
 actions 87
 autoreply aliases 480
 domain routing 483
 email aliases 479–480
 email relaying in 480
 filter aliases 480
 global options 484
 local domains 478–479
 mail directory 479
 mail queue 486
 mail user assignments 484–486
 reading users' email in 486–487
 using Majordomo in 391
 virtual mappings 481–483
 Qmail base directory field 489
 Qmail Configuration module 477–478
 configuring 488–490
 configuring domain routing 483
 configuring relaying 480
 editing global options 484
 editing local domains 478–479
 editing mail user assignments 484–486
 managing email aliases 479–480
- managing virtual mappings 481–483
 reading users' email 486–487
 viewing mail queue 486
 Qmail maildir in home directory field 490
 Qmail or MH directory in home directory field 633
 Qmail or MH directory location field 633
 QMail Server monitor 258
 Qmail style directory mail storage format 633
 qmail-queue program 477
 quota command 61
 quotas 60–61
 default 65
 disabling for filesystems 62
 enabling for filesystems 62
 setting for users or groups 63
 setting grace periods for 64–65
- R**
 RAID (redundant array of inexpensive disks) 74
 devices 45, 74
 levels of 75
 Read Email button 28
 Read Mail module (Usermin) 640
 read_acl function 760
 read_file function 731, 761
 read_file_cached function 729, 761
 read_file_lines function 753, 761
 read_http_connection function 759, 761
 readme files 511–512, 536–538
 README files table 536
 Read-only access field 56
 Read-only field 56
 Read-only hosts field 242
 ReadParse function 760
 ReadParseMime function 760
 Read-write access field 56
 Read-write hosts field 242
 Real hostname to connect to field 551
 real name 29
 Real name field 21
 Reboot System button 89
 rebooting 89
 Received Messages icon 218
 recompress command 530
 Re-Configure Known Modules icon 302
 Recording volume level field 217
- records
 adding 322–324
 editing 324
 generators 344
 types of 325–329
 Webmin support in reverse zones 329
 Records file field 321, 333, 334
R
 Red Hat Linux
 document root directory 268
 firewall configuration 175
 installing packages 108
 print system 206
 printer driver 209
 updating on 111
 Red Hat Network 111
 Red Hat Package Manager (RPM) 106
 redirect function 761
 redirects 278–279
 redundant RAID array 75
 referrer checking 684–685
 Refresh module list from CPAN checkbox 246
 Refresh time field 321, 330, 339
 Regexp document directory aliases table 277
 Regexp URL redirects table 278
 regular expressions 588
 reiserfs filesystem 39
 RELAYCLIENT environment variable 480
 relaying 480
 Remote HTTP Service monitor 258
 Remote IMAP server mail storage format 633
 Remote password field 382
 Remote Ping monitor 258
 Remote POP3 server mail storage format 633
 Remote Procedure Call (RPC) 643–644, 704–705, 730–732
 Remote TCP Service monitor 259
 Remote Windows server option 209
 remote_error_setup function 761
 remote_eval function 732, 761
 remote_finished function 761
 remote_foreign_call function 732, 761, 762
 remote_foreign_check function 762
 remote_foreign_config function 762
 remote_foreign_require function 730, 761, 762

- REMOTE_HOST environment variable 676
remote_read function 732, 762
remote_rpc_call function 762
REMOTE_USER environment variable 676
\$remote_user 713
remote_write function 732, 762
remotefile parameter 762
removable media 47
Remove address from list field 393
Remove Received: headers from resent email field 397
Rename button 234
rename command 542
rename function 762
rename_logged function 727, 762
replace_file_line function 762
Reply-To header 396, 397
Reply-To: address in resent email field 396
Report options field 494
Repository browser header file field 360
repquota command 61
Request from host condition type 293
Request from IP condition type 294
Request from net/CIDR condition type 294
Request from net/netmask condition type 294
Request from partial IP condition type 294
Request Method ACL (access control list) 590
Request MIME Type ACL (access control list) 590
Resent email footer field 394
Resent email header field 394
Resent email priority menu 397
reset_environment function 762
Resolve found server addresses field 706
resolve_links function 762
Responsible Person (RP) 328
restart.cgi script 729
restart_miniserv function 763
restore command 126
Restrict access by login field 290
Restrict access table 515
Restriction field 162
Resynchronize package list option 110
Reverse Address (PTR) record 329
reverse address delegation 343
reverse zones hosting 318 record types supported in 329
rexec program 132
RFC931 ident timeout field 596
RFC931 User ACL (access control list) 590
.rhosts file 123
Rieser Filesystem 50–51
Rings before answering field 217
rint command field 567
rlogin command 132
rm program 223
rndtopvf command 218
ROM (read-only memory) 195
root 19
Root access field 57
Root access hosts field 242
Root directory for ACL files field 601
Root directory for file chooser field 694
root domains 316
root filesystem 84
root privileges 3
root zones 316, 337
\$root_directory variable 713
routers 145, 173
routing 149–150
RP (Responsible Person) record 328
RPC (Remote Procedure Call) 643–644, 704–705, 730–732
RPC programs 135–136
RPM (Red Hat Package Manager) 106
.rpm files 108
RPM package 7 installing 7 SSL mode 15
rpm program 593
rpm shell command 106
rquotad 136
rules (firewall) 174 conditions 186 editing 182
Run as UNIX user field 281
Run chain option 180
Run commands on field 253
Run on host field 253
Run Webalizer as user field 496, 498
runlevels 85
Running Processes module 99–100, 625
access control 103–104
in other operating systems 104
running simple commands in 103
search feature 102
Running Processes module (Usermin) 640
rusersd program 136
- ## S
- Samba 554–555 authentication options 571 converting and setting user passwords in 557–558 encrypted password list 557 exporting directories 241–242 file naming options 565–566 file permission settings 564–565 file share 559–560 networking options 568–571 password list 20 print share 560–562 security options 563–564 sharing directories 240–241 user list 20 user synchronization in 558 Web Administration Tool 573
Samba Servers monitor 259
Samba Windows File Sharing module 556 access control 573–574 accessing SWAT 573 adding file share 559–560 configuring 574–576 configuring networking 568–571 configuring printers 572 editing file naming options 565–566 editing file permission settings 564–565 editing file sharing options 566–567 editing printer share options 567–568 editing security options 563–564 editing share defaults 568 enabling print share 560–562 managing Samba users 556–559 setting global authentication options 571 viewing and deleting client sessions 563
same_file function 763

- Sample webalizer configuration file field 499
- Save button 234
- save_allow.cgi function 735
- save_allowed function 736
- save_cron.cgi function 734
- save_denied function 736
- save_module_acl function 763
- scheduled command 97–98
- Scheduled Commands module 97
- Scheduled Commands module (Usermin) 641
- Scheduled Cron Jobs module 93
- access control 96
 - acl_security.pl access control script 738–739
 - CGI programs 734–735
 - configuration settings 737–738
 - configuring 96
 - controlling users' access 96
 - cron-lib.pl library script 735–737
 - lang internationalization directory 738
 - log_parser.pl log reporting script 739–740
 - useradmin_update.pl script 740
- See also* Cron Jobs
- Scheduled Cron Jobs module (Usermin) 641
- Scheduled report generation field 497
- scp program 544
- \$scriptname variable 713
- SCSI disks 69
- SCSI drives 71, 200
- Search APT button 108
- Search directory field 237
- Search Docs link 11
- Search For Package field 109
- Search rpmfind.net button 108
- secondary groups 19, 23, 26
- secondary servers 316
- section parameters 757
- Secure Sockets Layer (SSL) 304–307, 615–616
- seed_random function 763
- self-signed certificates 305
- Send digest when field 400
- Send email when field 253
- Send mail via connection to field 470, 488, 634
- Send outgoing mail via host field 458
- Send Signal button 101
- Send to field 456
- Send via SMTP server field 483
- Sender: address in email field 396
- Sending process group condition 189
- Sending process ID condition 189
- Sending unix group condition 189
- Sending unix user condition 189
- Sendmail
- address mappings 456–457
 - autoreply aliases 465–466
 - domain masquerading 452
 - domain routing 457–458
 - email aliases 452–455
 - email relaying 455
 - filter aliases 466–467
 - global options 458–459
 - local domains 451–452
 - M4 configuration 463–465
 - mail queue 460–461
 - reading users' email in 461–462
 - setting up to use Procmail 606–608
 - using Majordomo in 391
 - vs. Qmail 476–477
- sendmail command 391
- Sendmail command field 473, 634
- Sendmail command path field 401
- Sendmail Configuration module 449–451
- access control 468–469
- configuring 470–475
- configuring domain routing 457–458
- configuring relaying 455
- creating autoreply aliases 465–466
- creating filter aliases 466–467
- creating virtual address mappings 456–457
- editing global options 458–459
- editing local domains 451–452
- managing email aliases 452–455
- managing M4 configuration 463–465
- reading users' email 461–462
- setting up domain masquerading 452
- viewing mail queue 460–461
- Sendmail M4 base directory field 472
- Sendmail mail file in home directory field 489, 633
- Sendmail mail file location field 489, 633
- Sendmail Server monitor 259
- Sendmail style single file mail storage format 633
- SENDMAIL variable 612
- sendmail.cf file 450, 463
- SENDMAILFLAGS variable 612
- Separate window mode field 222
- Serial Port Configuration icon 166, 168, 216
- serialize_variable function 731, 763, 764
- Serve NIS domain field 157
- Server Message Block 554–555
- Server Name field 43, 270, 306, 366, 381
- Server name in URL field 630
- Server Port field 381, 385
- Server Program field 136
- Server Security icon 161
- servers
- CVS. *See* CVS servers
 - deleting 668, 703
 - DHCP 255
 - DNS. *See* DNS servers
 - editing 703
 - Jabber Instant Messaging 256
 - listing 668
 - monitoring status of 250–251
 - name 319–320
 - NFS 257
 - NIS+ 156
 - Postfix 257
 - ProFTPD 258
 - QMail 258
 - registering 645, 650–651, 661
 - Samba 259
 - scanning for 704
 - secondary 316
 - Sendmail 259
 - slave 316
 - SMTP 378
 - Squid 259
 - time zones 376
 - tunnels 703, 704
 - virtual. *See* virtual servers
- Servers category
- Apache Webserver module 265–268
 - BIND DNS Server module 318–320
 - CVS Server module 354
 - DHCP Server module 363–365
 - Fetchmail Mail Retrieval module 379–380

- Majordomo List Manager 389–391
MySQL Database Server 406–407
PostgreSQL Database Server module 429–430
Procmail Mail Filter 606
ProFTPD Server 501
Qmail Configuration module 477–478
Samba Windows File Sharing module 556
Sendmail Configuration module 449–451
Squid Proxy Server module 578–580
SSH module 545
Webalizer Logfile Analysis module 491–492
WU-FTP Server module 526
Servers to forward queries to field 341
server-side includes 282–284
Service Address (SRV) record 328–329
Service name field 503, 617
session_login.cgi program 746
set field 410, 415
setfacl command 239
Setup Firewall button 176
SGI Filesystem 51
shadow 158
Share Comment field 559
Share Name field 43, 559
shared networks 362
 adding 372–373
 editing 373
shared printers 555
shell 30, 132
shell commands 224–225
 creating 225–227
Shell in a Box module 223
SHELL variable 611
Show Apache directive names field 313
Show databases and tables as field 425
Show Directives icon 309
Show file descriptions option 275
Show servers as field 706
Show status for servers field 706
show_times_input function 736
.shtml files 283
Shutdown commands field 88
Shutdown System button 89
Silence threshold level field 217
Simple Mail Transfer Protocol (SMTP) 384, 448
Single user mode runlevel 85
slave NIS servers 163
slave servers 316
Slave servers field 158
slave zones 316
 converting master zones to 332
 converting to master zones 336
 creating 332–334
 deleting 336
 editing 334–335
smallint field 411, 413
SMB (Server Message Block) 554–555
smb.conf file 575
smbclient program 214
smbd server program 575
smbfs filesystem 40, 43–44
smbpasswd program 575
smbstatus program 575
SMRSH directory field 475
SMRSH program 404
SMTP (Simple Mail Transfer Protocol) 384, 448
 SMTP connection timeout field 484
 SMTP greeting message field 484
 SMTP incoming data timeout field 484
 SMTP outgoing response timeout field 484
 SMTP port field 483
 SMTP port options field 459
 SMTP server 378
 snapshots 82–83
 SNMP Community ACL (access control list) 590
 Socket Type field 136
 Soft Block Limit field 63
 Soft File Limit field 63
 soft limit 60–61
 software packages 105–106
 deleting 647, 647–648
 finding and removing 109
 formats 7
 installing 107–108, 646
 refreshing lists of 648
 searching for 646–647
 viewing 647–648
Software Packages module 107, 646
 finding and removing packages 109
 installing new packages 107–108
Solaris operating system 2
 boot loader 195
 bootup scripts 89–90
 filesystem 51
 filesystem backups 128
 Internet service 138
 managing disks and partitions in 74
 network configurations 153
 NFS export on 56–57
 NIS configuratiion 163
 package format 112
 print system 213
 software package 7, 10
 system logs 119
 system time 193–194
 users and groups in 37
Sort directory index by field 276
Sort leases by field 376
Sort mailing lists by field 403
Sort proxy users field 602
Sort servers by field 706
Sort tables by field 470, 488
Source address or network condition 187
Source and destination port(s) condition 188
Source AS Number ACL (access control list) 590
Source file for generics database field 474
Source file for mailertable database field 474
Source file for the access database field 474
Source file for the domains database field 474
Source file for virtusers database field 474
Source NAT option 180
Source TCP or UDP port condition 187
SpamAssassin (mail filtering software) 613
spammers 455
SPARC system 7, 195
SQL (Structured Query Language) 405, 428
Squid cachemgr.cgi executable field 603
Squid executable field 603
Squid proxy server 578
 ACLs (access control lists) 584–586

- authentication 593–594
- cache clearing 598
- cache directories 582–583
- cache manager statistics 599
- caching options 583–584
- connecting to other proxies 597–598
- initializing cache in 579
- log analysis 600
- log configurations 595–596
- ports 580–581
- restrictions 592–593
- squid.conf file 578
- user synchronization 595
- users for authentication 594
- using Webmin in 2
- vs. Apache server 301
- Squid Proxy Server module** 578–580
 - access control 601
 - adding cache directories 582–583
 - analyzing logs 600
 - changing proxy ports and addresses 580–581
 - clearing cache 598
 - configuring 601–603
 - configuring logging 595–596
 - connecting to other proxies 597–598
 - creating and editing proxy restrictions 592–593
 - editing caching and proxy options 583–584
 - setting up proxy authentication 593–595
 - viewing cache manager statistics 599
- Squid Proxy Server monitor 259
- squid.conf file 578
- srm.conf file 310, 314
- SRV (Service Address) record 328–329
- SSH Configuration module** (Usermin) 641
- ssh program 544
- SSH server** 220–222, 544
 - access restrictions 546–547
 - authentication configuration 549–550
 - client host options 551–552
 - network configuration 547–549
- SSH Server module** 545
 - configuring 553
- editing authentication settings 549–550
- editing client host options 551–552
- restricting access 546–547
- setting up new users 552–553
- SSH/Telnet Login module** 220–222
- SSH/Telnet Login module** (Usermin)
 - 641
 - ssh_config file 544
 - sshd_config file 544
 - ssh-keygen program 553
 - SSL (Secure Sockets Layer) 304–307, 615–616
 - SSL certificate 17–18, 618
 - SSL certificate and key file field 618
 - SSL encryption 15–17, 617
 - SSL key 695–697
 - SSL Tunnels module 616
 - configuring 618–619
 - creating and editing SSL tunnels 617–618
 - Standard httpd.conf file 270
 - Start Apache link 268
 - Start Fetchmail Daemon button 385
 - Start Scheduler button 206
 - State field 16
 - State or Province Name field 305, 308
 - STDERR 282
 - Stop Apache link 268
 - Stop Fetchmail Daemon button 385
 - Stop Scheduler button 206
 - STOP signal 101
 - Storage log file field 596
 - Store root servers in file field 338
 - striped RAID array 75
 - Structured Query Language (SQL) 405, 428
 - stub zones 334
 - STunnel 615–616
 - sub-domain 316
 - Subject: prefix for resent email field. 397
 - submodules 245
 - subnets 362
 - adding 365–367
 - assignment 365
 - deleting 367
 - displaying 376
 - file structure 365
 - IP addresses 365
 - names 365
 - in shared networks 372–373
 - Subnets in this shared network field 372
 - Subscribe address to list field 393
 - subscribe list address command 388–389
 - Subscribe policy field 395
 - subscription options 395–396
 - suexec program 281–282
 - suexec_log file 282
 - Sun Microsystems 154
 - super-servers 129–130
 - Support DNS for IPv6 addresses field 349
 - SuSE Linux operating system 390
 - swap space 68
 - SWAT** (Samba Web Administration Tool) 573
 - switch_to_remote_user function 732, 763
 - symbolic links 121, 243
 - syslog facility 114
 - Syslog server on option 116
 - sysprint function 763
 - System and Server Status module**
 - 250–251
 - access control 262
 - adding monitors 252–253
 - configuring 262–263
 - setting up scheduled monitoring 260–261
 - System category**
 - Disk Quotas module 61–62
 - Disks and Network Filesystems module 40
 - Filesystem Backup module 121–122
 - Running Processes module 99–100
 - Scheduled Commands module 97
 - Scheduled Cron Jobs module 93
 - Software Packages module 107
 - System Logs module 115
 - Users and Groups module 20
 - System Documentation module** 757
 - System Documentation module** (Usermin) 641
 - System Log option** 285
 - system logs 113–115
 - adding 115–117
 - deleting 117–118
 - editing 117–118
 - in versions of UNIX operating systems 119–120

- System Logs module 115
access control 118–119
adding new log files 115–117
editing or deleting log files 117–118
in other operating systems 119–120
system time 191
changing 192
synchronizing hardware time with 193
synchronizing with system time of another server 193
System Time module 191
acces control 193
changing hardware time 192–193
changing system time 192
synchronizing times with another server 193
System V package format 112
system_logged function 727, 763
SysV Init Configuration module 91–92
SYSV print system 572
- T**
- Table name field 408, 431
TABLE statement 418
tables 174
actions 179–180
adding fields to 433
creating 408–409, 431–432
deleting 416–417, 436
deleting fields from 433–434
editing contents of 436
editing fields in 433
permissions 421–423
UNIX users 161
viewing and editing contents 412–416
viewing contents of 436
Tables fields 267
Taboo body regexps field 398
Taboo header regexps field 398
Tape size field 123
tar archives 438
tar command 121, 530
tar.gz package 7
installing 8–10, 245
\$tb variable 713
TCP flags set condition 188
TCP option number is set condition 188
- TCP port field 618
tcpserver daemon 480
TCP-wrappers 356
configuration files 623
server program 129
telnet 129
clients 315
firewall blocking 222–223
server 220–222
use of 131
tempname function 763
Temporary URL redirects 278
TERM signal 101
Terminate Process button 101
terror function 763
Test telnet or SSH server field 222
Test to perform parameter 254
text authentication files 292
text field 410, 415
Text file authentication box 290
text files 234
text function 763
Text parameter 228, 757
Text record 327
Text Record Name field 328
texttitles 742
Thawte 304
theme.info file 741, 747
theme.pl file 759
theme_error function 744
theme_footer function 744, 749–750
theme_header function 744, 748–749, 756
themes
configuration files 741–742
functions 744–746
overriding images and programs 743–744
packaging 742
thrashing 46
Time before giving up field 459
Time before sending warning field 459
time field 410, 414, 435
time service 132
Time to wait for response parameter 258
Time to wait for scan responses field 706
time zones 376
Timed out state 252
Timeout before using default field 203
timestamp field 414, 435
Time-To-Live field 323
tinyblob field 415
- tinyint field 413
tinytext field 415
<title> tags 275
tmp/email file 399
to_ipaddress function 764
token ring cards 144
Too many connections message file field 511
Transfer retry time field 321, 330, 339
transparent proxying 184–185
Treat all files as MIME type field 289
Tru64/OSF1 operating system
bootup scripts 89
Internet service 138
users and groups in 37
trunc function 764
Trust remote users option 55
Trusted port restriction 162
Trusted websites field 685
Tunnel mode field 618
tunnels 617–618, 703
TXT (Text) record 327
Type of service condition 189
Type width field 409, 432
TypesConfig directive 314
- U**
- uconfig.info file 733, 756
ufsdump command 128
ufsrestore command 128
UID parameter 228
umask command 542
Umask for new files field 243
un_urllize function 764
uninstalling Webmin 13
unique function 764
UNIX groups and GIDs not to deny field 528
UNIX operating systems
backups 128
bootup scripts 89–90
Cron jobs 97
disk quotas 66
file systems 51–52
filesystems 39–40
groups 19
Internet service 138–139
network configurations 153
NFS exports on 57–59
package formats 111–112
print systems 213–214
running processes 104
system logs 119–120
system time 193–194

- user accounts 19, 37–38
- UNIX systems 1
 - backups 121
 - disk quotas 60–61
 - internal clocks 191
 - viewing logins in 27–28
- UNIX talk program 131
- UNIX user field 485
- UNIX user to connect to database as field 445
- UNIX users and UIDs not to deny field 528
- Unix users and UIDs to deny field 528
- UNIX users table 161
- Unix users to deny field 528
- UNIX_group_input function 764
- UNIX_user_input function 764
- UNIXWare operating system
 - booting with GRUB 202
 - bootup scripts 89–90
 - filesystem 52
 - Internet service 138
 - network configurations 153
 - package format 112
 - print system 213
 - system logs 119
 - users and groups in 37
- unlock_all_files function 764
- unlock_file function 725, 739, 764
- Unresume job command field 567
- unserialize_variable function 731, 763, 764
- unsubscribe list address command 389
- Unsubscribe policy field 395
- Up state 251
- Update reverse is field 349
- Upload button 234
- Upload message field 219
- Upload to directory field 234
- uploads 521–522
- URL Path Regexp ACL (access control list) 591
- URL Port ACL (access control list) 591
- URL Protocol ACL (access control list) 591
- URL redirects table 278
- URL Regexp ACL (access control list) 591
- URL to request parameter 258
- urlize function 764
- URLs (Uniform Resource Locators) 258
 - location 274
- mapping 279
- mapping paths to files or directories 276–277
- redirecting paths 278–279
- USB modems 166, 216
- Use DBI to connect if available field 426, 445
- Use random number for list alias field 403
- Use security level field 375
- Use vertical row editing interface field 425
- User DBM file field 292
- user facility 114
- user ID 21, 29
- User ID field 21
- user interface 10–13
- User mail file location field 474
- User mode 100
- User parameter 228
- User text file field 291
- User to run the fetchmail daemon field 387
- User to start BIND as field 348
- User WWW directory field 300
- user_chooser_button function 755, 764
- useradmin_update.pl script 740
- Usermin 620–621
 - access restrictions 622–623
 - authentication configurations 626–628
 - categories 628–629
 - default language 625
 - downloading 620
 - IP address 623
 - login restrictions 638
 - modules 638–641
 - access restrictions 636
 - configuring 631–635
 - creating 732–733
 - installing 624–625
 - moving 628–629
 - ports 623
 - preferences 631
 - SSL mode 630–631
 - starting and stopping 621–622
 - themes 629–630
 - upgrading 625–626
 - user interface 623–624
 - vs. Fetchmail 378
- Usermin configuration directory field 642
- Usermin Configuration module 621
- changing and installing themes 629–630
- changing default language 625
- changing port and IP address 623
- configuring 641
- configuring authentication 626–628
- configuring user interface 623–624
- editing categories 628–629
- installing modules 624–625
- restricting access 622–623
- setting login restrictions 638
- switch to SSL mode 630–631
- Usermin Web server monitor 259
- Username field 422
- Username for proxy field 672
- Username mapping field 571
- users 19
 - adding in PostgreSQL 439–440
 - batch files 28–29
 - copying to multiple users 63–64
 - creating 21–23
 - default quotas 65
 - deleting 24–25
 - editing 23–24
 - grace periods for 64–65
 - ID 29
 - login name 29
 - password 29
 - password protection 289–291
 - reading email of 28
 - real name 29
 - setting quotas for 63
 - web directories 300–301
- Users and Groups module 20
 - access control 34–37
 - before and after commands 34–35
 - configuring 30–34
 - creating new groups 25–26
 - creating new users 21–23
 - deleting groups 27
 - deleting users 24–25
 - editing groups 26
 - editing users 23–24
- Users visible in user chooser field 694
- usr/lib/sendmail file 477
- usr/sbin/sendmail file 477
- uucp facility 114

V

- valid certificates 17–18
- var/log directory 113

- var/log/mail file 113
var/log/messages file 113
var/log/secure file 113
var/log/xferlog file 517–518
var/mail file 378
var/qmail file 476
var/spool/cron directory 93
var/spool/mail file 608, 610
var/webmin/webmin.log file 708
var/yp/Makefile file 155
var/yp/securenets file 155
varchar field 410–411, 414, 433, 435
variable assignments 611–612
Variable name field 611
Verisign 304
version 9
vfat filesystem 39
vgcreate program 78
vgetty program 215
vi command 223, 309
views 318, 345–346
virtual address mappings 456–457, 481–483
virtual interfaces 148
virtual memory 40
 adding 46–47
 partitions 68, 70
 swapping 46
virtual servers 265
 adding 312
 changing log formats 286
 configuring 312
 creating 269–272, 505–506
 displaying 312
 editing 272–273
 log configurations 517–518
 ordering of 312
 separate logs files for 285–286
 server-side includes 283–284
 See also servers
virtusers 474
Voicemail Server module 215–216
 configuring Linux systems as answering machines 216–218
 setting greeting messages 219
 viewing and managing recorded messages 218–219
Voicemail Server Options icon 217
Volume group name field 79
volume groups 77, 79–80
- W**
- walld program 136
- Warning days field 23
warning priority 114
WAV files 219
web browsers 315
Web Server Address ACL (access control list) 591
Web Server Hostname ACL (access control list) 592
Web Server Regexp ACL (access control list) 592
Web servers
 Apache 254
 configuring logging 284–285
 monitors 254, 259
 port 9
 running CGI programs 280–282
 server-side includes 282–284
 SSL (Secure Sockets Layer) protocol 304–307
 Usermin 259
 Webmin 259
Webalizer 491–492
Webalizer DNS cache field 494
Webalizer history file field 494
Webalizer incremental file field 494
Webalizer Logfile Analysis module 491–492
 access control 498–499
 adding log files for reporting 497
 configuring 499
 editing global options 498
 editing report options 492–494
 generating and viewing reports 496
 setting up scheduled reporting 496–497
webalizer.conf file 498
web-lib.pl file 712, 751
Webmin 1–2
 access restrictions 669–670, 688–689
 action logging 726–727
 authentication settings 681–682
 categories 683
 certificate authority 686–687
 changing operating system 675–676
 clustering 643–644
 development of 3
 downloading 6–7
 environment variables 676
 file locking 725
 installing
 RPM package 7
 Solaris package 10
 tar.gz package 8–10
 installing updates 679–680
 IP address 670–671
 language 676
 log files 708–709
 logging in 671, 707–708
 main menu settings 677–678
 modules. *See* modules
 network security 14–15
 online help 718–719
 package formats 7
 ports 670–671
 program paths 676
 proxy servers 672
 referrer checking 684–685
 SSL encryption 15–17
 SSL mode 686
 supported operating systems 7
 themes 683–684, 741–746
 tunnels 703
 uninstalling 13
 upgrading 6, 678–679
 user interface 10–13, 672–673
 users of 2–3
 versions 6
 Web server 259
 website 6
Webmin Actions Log module 708
Webmin category
 Webmin Actions Log module 708
 Webmin Configuration module 669
 Webmin Servers Index module 700
 Webmin Users module 689
Webmin Configuration module 669
 allowing unauthenticated access to modules 685–686
 certificate authority 686–687
 changing and installing themes 683–684
 changing language 676
 changing operating system 675–676
 changing port and address 670–671
 configuring authentication 681–682
 configuring user interface 672–673
 deleting modules 674
 editing categories 683

- editing environment variables 676
 editing main menu settings 677–678
 editing program path 676
 installing modules 673–674
 installing updates 679–680
 referrer checking 684–685
 restricting access 669–670
 restricting IP addresses and networks 14–15
 setting up logging 671
 turning on SSL 686
 upgrading Webmin 678–679
 using proxy servers 672
 Webmin down state 251
 Webmin driver option 209
 Webmin Servers Index module 700
 access control 705
 adding servers 701–703
 broadcasting and scanning for servers 704
 configuring 706
 editing or deleting servers 703
 registering servers 645
 using server tunnels 703
 Webmin Users module 689
 access control 697–698
 changing themes 683
 configuring 698
 creating and editing groups 694
 creating new users 662
 creating users 689–691
 editing module access control 692–694
 editing users 691–692
 requesting client SSL key 695–697
 viewing and disconnecting login sessions 697
 Webmin Web server monitor 259
 webmin.log file 708
 webmin_log function 726, 739, 762, 765
 WebNFS clients 55
 Website hostname field 494
 websites 269
 Well Known Service (WKS) record 327
 wget program 593
 which command 397
 who command 397
 who list command 389
- Width to wrap mail messages at field 470
 wildcard characters 274
 wildcards 349
 Windows 95 filesystem 45–46
 Windows NT filesystem 45–46
 Windows operating systems
 boot loader 195
 booting with GRUB 202
 filesystems 45–46
 smbfs filesystem 43–44
 winipcfg program 370
 Winmodems 166, 215
 wireless cards 144
 WKS (Well Known Service) record 327
 Wrapping mode in mail textarea field 471
 writable partitions 74
 Write backup as UNIX user field 424
 Write key to file field 631
 Write report to directory field 493
 write_env_file function 765
 write_file function 765
 write_http_connection function 759, 765
 WU-FTP server
 client restrictions by IP address 541
 concurrent login limits 540–541
 directory aliases 535–536
 filename download restrictions 532–533
 filename upload restrictions 533–534
 FTP command restrictions 541–542
 guest users 534
 login configurations 538–540
 message and banner files 536–538
 WU-FTP Server module
 configuring 542–543
 configuring logging 538–540
 defining message and readme files 536–538
 editing directory aliases 535–536
 limiting concurrent logins 540–541
 managing user classes 531–532
 restricting access to FTP commands 541–542
- restricting clients by IP address 541
 setting up filename restrictions 532–534
 setting up guest users 534–535
 WU-FTP server 525
 anonymous 529–531
 inetd service 527–528
 login restrictions 528
 user classes 531–532
 xinetd service 527
 WU-FTP Server module 526
 setting up anonymous FTP 529–531
 setting up inetd service 527–528
 setting up login restrictions 528
 setting up xinetd service 527
- X**
- x86 code 223
 x86 systems 7, 105, 195
 xfs filesystem 39, 51, 238
 xfsdump command 124
 xinetd super-server 130, 139, 142–143, 503, 527
- Y**
- YaST tool 176
 year field 414
 YP (Yellow Pages) 154
- Z**
- .Z encoding format 294
 zcat command 530
 Zip disks 47, 123
 Zip drive 73
 Zone Name field 329
 Zone transfer format field 341
 Zone type field 333
 zones 315
 changing global options in 339–340
 editing defaults in 338–339
 forwarding 341
 master. See master zones
 partial reverse delegation 342–343
 serial number 351
 slave. See slave zones
 templates 338
 transfers 341