The Official Samba 3.0.x HOWTO and Reference Guide

Jelmer R. Vernooij, John H. Terpstra, and Gerald (Jerry) Carter

September 30, 2009

ABOUT THE COVER ARTWORK

The cover artwork of this book continues the freedom theme of the first edition of "The Official Samba-3 HOWTO and Reference Guide". We may look back upon the past to question the motives of those who have gone before us. Seldom do we realise that the past owes us no answer, and despite what we may think of the actions of those who have travelled lifes' road before us, we must feel a sense of pride and gratitude for those who, in the past, have protected our liberties.

Developments in information technology continue to move at an alarming pace. Human nature causes us to adopt and embrace new developments that appear to answer the needs of the moment, but that can entrap us at a future date. There are many examples in the short history of information technology. MS-DOS was seen as a tool that liberated users from the tyrany of large computer system operating costs, and that made possible the rapid progres we are beneficiaries of today. Yet today we are inclined to look back with disdain on MS-DOS as an obsolete and constraining technology that belongs are an era that is best forgotten.

The embrace of Windows networking, Windows NT4, and MS Active Directory in more recent times, may seem modern and progressive today, but sooner or later something better will replace them. The current preoccupation with extended identity management solutions and with directories is not unexpected. The day will come that these too will be evaluated, and what may seem refreshing and powerful may be better recogized as the chilly winds of the night. To argue against progress is unthinkable, no matter what may lie ahead.

The development of Samba is moving forwards. The changes since Samba 3.0.0 are amazing, yet many users would like to see more and faster progress. The benefits of recent developments can be realized quickly, but documentation is necessary to unlock the pandoras' box. It is our hope that this book will help the network administrator to rapidly deploy the new features with minimum effort. As you deploy and gain mileage from the new enablement,

take the time to think through what may lie ahead. Above all, take stock of the freedom of choice that Samba provides in your world, and enjoy the new potential for seamless interoperability.

ATTRIBUTION

Chapter 1, "How to Install and Test SAMBA"

- Andrew Tridgell<mailto:tridge@samba.org>
- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- John H. Terpstra<mailto:jht@samba.org>
- Karl Auer<mailto:kauer@biplane.com.au>
- Dan Shearer<mailto:dan@samba.org>

Chapter 2, "Fast Start: Cure for Impatience"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 3, "Server Types and Security Modes"

- Andrew Tridgell<mailto:tridge@samba.org>
- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- John H. Terpstra<mailto:jht@samba.org>

Chapter 4, "Domain Control"

- John H. Terpstra<mailto:jht@samba.org>
- Gerald (Jerry) Carter<mailto:jerry@samba.org>
- David Bannon<mailto:dbannon@samba.org>
- Guenther Deschner<mailto:gd@suse.de> (LDAP updates)

Chapter 5, "Backup Domain Control"

- John H. Terpstra<mailto:jht@samba.org>
- Volker Lendecke<mailto:Volker.Lendecke@SerNet.DE>
- Guenther Deschner<mailto:gd@suse.de> (LDAP updates)

Chapter 6, "Domain Membership"

- John H. Terpstra<mailto:jht@samba.org>

- Jeremy Allison<mailto:jra@samba.org>
- Gerald (Jerry) Carter<mailto:jerry@samba.org>
- Andrew Tridgell<mailto:tridge@samba.org>
- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- Guenther Deschner<mailto:gd@suse.de> (LDAP updates)

Chapter 7, "Standalone Servers"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 8, "MS Windows Network Configuration Guide"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 9, "Important and Critical Change Notes for the Samba 3.x Series"

- John H. Terpstra<mailto:jht@samba.org>
- Gerald (Jerry) Carter<mailto:jerry@samba.org>

Chapter 10, "Network Browsing"

- John H. Terpstra<mailto:jht@samba.org>
- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- Jonathan Johnson<mailto:jon@sutinen.com>

Chapter 11, "Account Information Databases"

- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- John H. Terpstra<mailto:jht@samba.org>
- Gerald (Jerry) Carter<mailto:jerry@samba.org>
- Jeremy Allison<mailto:jra@samba.org>
- Guenther Deschner<mailto:gd@suse.de> (LDAP updates)
- Olivier (lem) Lemaire<mailto:olem@IDEALX.org>

Chapter 12, "Group Mapping: MS Windows and UNIX"

- John H. Terpstra<mailto:jht@samba.org>
- Jean François Micouleau
- Gerald (Jerry) Carter<mailto:jerry@samba.org>

Chapter 13, "Remote and Local Management: The Net Command"

- John H. Terpstra<mailto:jht@samba.org>
- Volker Lendecke<mailto:Volker.Lendecke@SerNet.DE>
- Guenther Deschner<mailto:gd@suse.de>

Chapter 14, "Identity Mapping (IDMAP)"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 15, "User Rights and Privileges"

- Gerald (Jerry) Carter<mailto:jerry@samba.org>
- John H. Terpstra<mailto:jht@samba.org>

Chapter 16, "File, Directory, and Share Access Controls"

- John H. Terpstra<mailto:jht@samba.org>
- Jeremy Allison<mailto:jra@samba.org>
- Jelmer R. Vernooij<mailto:jelmer@samba.org> (drawing)

Chapter 17, "File and Record Locking"

- Jeremy Allison<mailto:jra@samba.org>
- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- John H. Terpstra<mailto:jht@samba.org>
- Eric Roseme<mailto:eric.roseme@hp.com>

Chapter 18, "Securing Samba"

- Andrew Tridgell<mailto:tridge@samba.org>
- John H. Terpstra<mailto:jht@samba.org>

Chapter 19, "Interdomain Trust Relationships"

- John H. Terpstra<mailto:jht@samba.org>
- Rafal Szczesniak<mailto:mimir@samba.org>
- Jelmer R. Vernooij<mailto:jelmer@samba.org> (drawing)
- Stephen Langasek<mailto:vorlon@netexpress.net>

Chapter 41, "Hosting a Microsoft Distributed File System Tree"

- Shirish Kalele<mailto:samba@samba.org>
- John H. Terpstra<mailto:jht@samba.org>

Chapter 21, "Classical Printing Support"

- Kurt Pfeifle<mailto:kpfeifle@danka.de>
- Gerald (Jerry) Carter<mailto:jerry@samba.org>
- John H. Terpstra<mailto:jht@samba.org>

Chapter 22, "CUPS Printing Support"

- Kurt Pfeifle<mailto:kpfeifle@danka.de>
- Ciprian Vizitiu<mailto:CVizitiu@gbif.org> (drawings)
- Jelmer R. Vernooij<mailto:jelmer@samba.org> (drawings)

Chapter 23, "Stackable VFS modules"

- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- John H. Terpstra<mailto:jht@samba.org>
- Tim Potter<mailto:tpot@samba.org>
- Simo Sorce (original vfs_skel README)
- Alexander Bokovoy (original vfs_netatalk docs)
- Stefan Metzmacher (Update for multiple modules)
- Ed Riddle (original shadow_copy docs)

Chapter 24, "Winbind: Use of Domain Accounts"

- Tim Potter<mailto:tpot@linuxcare.com.au>
- Andrew Tridgell<mailto:tridge@samba.org>
- Naag Mummaneni<mailto:getnag@rediffmail.com> (Notes for Solaris)
- John Trostel<mailto:jtrostel@snapserver.com>
- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- John H. Terpstra<mailto:jht@samba.org>

Chapter 25, "Advanced Network Management"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 26, "System and Account Policies"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 27, "Desktop Profile Management"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 28, "PAM-Based Distributed Authentication"

- John H. Terpstra<mailto:jht@samba.org>
- Stephen Langasek<mailto:vorlon@netexpress.net>

Chapter 29, "Integrating MS Windows Networks with Samba"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 30, "Unicode/Charsets"

- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- John H. Terpstra<mailto:jht@samba.org>
- TAKAHASHI Motonobu<mailto:monyo@home.monyo.com> (Japanese character support)

Chapter 31, "Backup Techniques"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 32, "High Availability"

- John H. Terpstra<mailto:jht@samba.org>
- Jeremy Allison<mailto:jra@samba.org>

Chapter 33, "Handling Large Directories"

- Jeremy Allison<mailto:jra@samba.org>
- John H. Terpstra<mailto:jht@samba.org>

Chapter 34, "Advanced Configuration Techniques"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 35, "Updating and Upgrading Samba"

- Jelmer R. Vernooij<mailto:jelmer@samba.org>

- John H. Terpstra<mailto:jht@samba.org>
- Gerald (Jerry) Carter<mailto:jerry@samba.org>

Chapter 36, "Migration from NT4 PDC to Samba-3 PDC"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 37, "SWAT: The Samba Web Administration Tool"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 38, "The Samba Checklist"

- Andrew Tridgell<mailto:tridge@samba.org>
- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- Dan Shearer<mailto:dan@samba.org>

Chapter 39, "Analyzing and Solving Samba Problems"

- Gerald (Jerry) Carter<mailto:jerry@samba.org>
- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- David Bannon<mailto:dbannon@samba.org>
- Dan Shearer<mailto:dan@samba.org>

Chapter 40, "Reporting Bugs"

- John H. Terpstra<mailto:jht@samba.org>
- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- Andrew Tridgell<mailto:tridge@samba.org>

Chapter 41, "Hosting a Microsoft Distributed File System Tree"

- John H. Terpstra<mailto:jht@samba.org>

Chapter 42, "How to Compile Samba"

- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- John H. Terpstra<mailto:jht@samba.org>
- Andrew Tridgell<mailto:tridge@samba.org>

Chapter 43, "Portability"

- Jelmer R. Vernooij<mailto:jelmer@samba.org>

- John H. Terpstra<mailto:jht@samba.org>

Chapter 44, "Samba and Other CIFS Clients"

- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- John H. Terpstra<mailto:jht@samba.org>
- Dan Shearer<mailto:dan@samba.org>
- Jim McDonough<mailto:jmcd@us.ibm.com> (OS/2)

Chapter 45, "Samba Performance Tuning"

- Paul Cochrane<mailto:paulc@dth.scot.nhs.uk>
- Jelmer R. Vernooij<mailto:jelmer@samba.org>
- John H. Terpstra<mailto:jht@samba.org>

Chapter 46, "LDAP and Transport Layer Security"

- Gavin Henry<mailto:ghenry@suretecsystems.com>

Chapter 48, "DNS and DHCP Configuration Guide"

- John H. Terpstra<mailto:jht@samba.org>

CONTENTS

Contents

ABOUT THE COVER ARTWORK **v**

ATTRIBUTION **vii**

LIST OF EXAMPLES **xliii**

LIST OF FIGURES **xlix**

LIST OF TABLES **lii**

FOREWORD **lv**

PREFACE **lvii**

INTRODUCTION **lix**

Part I General Installation **lxi**

PREPARING SAMBA FOR CONFIGURATION **1**

Chapter 1 HOW TO INSTALL AND TEST SAMBA **3**
1.1 Obtaining and Installing Samba 3
1.2 Configuring Samba (smb.conf) 3
1.2.1 Configuration File Syntax 3
1.2.2 TDB Database File Information 5
1.2.3 Starting Samba 5
1.2.4 Example Configuration 7
1.2.4.1 Test Your Config File with testparm 8
1.2.5 SWAT 9
1.3 List Shares Available on the Server 9
1.4 Connect with a UNIX Client 10
1.5 Connect from a Remote SMB Client 10
1.5.1 What If Things Don't Work? 11
1.5.2 Still Stuck? 11

1.6 Common Errors 12
1.6.1 Large Number of smbd Processes 12
1.6.2 Error Message: open_oplock_ipc 12
1.6.3 "The network name cannot be found" 13

Chapter 2 FAST START: CURE FOR IMPATIENCE 15
2.1 Features and Benefits 16
2.2 Description of Example Sites 16
2.3 Worked Examples 17
2.3.1 Standalone Server 17
2.3.1.1 Anonymous Read-Only Document Server 17
2.3.1.2 Anonymous Read-Write Document Server 20
2.3.1.3 Anonymous Print Server 20
2.3.1.4 Secure Read-Write File and Print Server 23
2.3.2 Domain Member Server 27
2.3.2.1 Example Configuration 28
2.3.3 Domain Controller 31
2.3.3.1 Example: Engineering Office 32
2.3.3.2 A Big Organization 34

Part II Server Configuration Basics 39

FIRST STEPS IN SERVER CONFIGURATION 41

Chapter 3 SERVER TYPES AND SECURITY MODES 43
3.1 Features and Benefits 43
3.2 Server Types 44
3.3 Samba Security Modes 45
3.3.1 User Level Security 46
3.3.1.1 Example Configuration 47
3.3.2 Share-Level Security 47
3.3.2.1 Example Configuration 48
3.3.3 Domain Security Mode (User-Level Security) 48
3.3.3.1 Example Configuration 49
3.3.4 ADS Security Mode (User-Level Security) 51
3.3.4.1 Example Configuration 51
3.3.5 Server Security (User Level Security) 51
3.3.5.1 Example Configuration 53
3.4 Password Checking 54

3.5 Common Errors 55
3.5.1 What Makes Samba a Server? 56
3.5.2 What Makes Samba a Domain Controller? 56
3.5.3 What Makes Samba a Domain Member? 56
3.5.4 Constantly Losing Connections to Password Server 56
3.5.5 Stand-alone Server is converted to Domain Controller — Now User accounts don't work 57

Chapter 4 DOMAIN CONTROL 59
4.1 Features and Benefits 60
4.2 Single Sign-On and Domain Security 64
4.3 Basics of Domain Control 67
4.3.1 Domain Controller Types 67
4.3.2 Preparing for Domain Control 70
4.4 Domain Control: Example Configuration 73
4.5 Samba ADS Domain Control 75
4.6 Domain and Network Logon Configuration 76
4.6.1 Domain Network Logon Service 76
4.6.1.1 Example Configuration 76
4.6.1.2 The Special Case of MS Windows XP Home Edition 76
4.6.1.3 The Special Case of Windows 9x/Me 77
4.6.2 Security Mode and Master Browsers 79
4.7 Common Errors 81
4.7.1 "$" Cannot Be Included in Machine Name 81
4.7.2 Joining Domain Fails Because of Existing Machine Account 81
4.7.3 The System Cannot Log You On (C000019B) 82
4.7.4 The Machine Trust Account Is Not Accessible 83
4.7.5 Account Disabled 83
4.7.6 Domain Controller Unavailable 83
4.7.7 Cannot Log onto Domain Member Workstation After Joining Domain 84

Chapter 5 BACKUP DOMAIN CONTROL 85
5.1 Features and Benefits 85
5.2 Essential Background Information 86
5.2.1 MS Windows NT4-style Domain Control 87
5.2.1.1 Example PDC Configuration 89
5.2.2 LDAP Configuration Notes 90

5.2.3 Active Directory Domain Control 91
5.2.4 What Qualifies a Domain Controller on the Network? 92
5.2.5 How Does a Workstation find its Domain Controller? 92
5.2.5.1 NetBIOS Over TCP/IP Enabled 92
5.2.5.2 NetBIOS Over TCP/IP Disabled 93
5.3 Backup Domain Controller Configuration 93
5.3.1 Example Configuration 94
5.4 Common Errors 96
5.4.1 Machine Accounts Keep Expiring 96
5.4.2 Can Samba Be a Backup Domain Controller to an NT4 PDC? 97
5.4.3 How Do I Replicate the smbpasswd File? 97
5.4.4 Can I Do This All with LDAP? 98

Chapter 6 DOMAIN MEMBERSHIP 99
6.1 Features and Benefits 99
6.2 MS Windows Workstation/Server Machine Trust Accounts 100
6.2.1 Manual Creation of Machine Trust Accounts 102
6.2.2 Managing Domain Machine Accounts using NT4 Server Manager 104
6.2.3 On-the-Fly Creation of Machine Trust Accounts 105
6.2.4 Making an MS Windows Workstation or Server a Domain Member 105
6.2.4.1 Windows 200x/XP Professional Client 106
6.2.4.2 Windows NT4 Client 106
6.2.4.3 Samba Client 106
6.3 Domain Member Server 107
6.3.1 Joining an NT4-type Domain with Samba-3 107
6.3.2 Why Is This Better Than security = server? 110
6.4 Samba ADS Domain Membership 111
6.4.1 Configure smb.conf 111
6.4.2 Configure /etc/krb5.conf 112
6.4.3 Create the Computer Account 115
6.4.3.1 Possible Errors 116
6.4.4 Testing Server Setup 116
6.4.5 Testing with smbclient 117
6.4.6 Notes 117
6.5 Sharing User ID Mappings between Samba Domain Members 117
6.6 Common Errors 118
6.6.1 Cannot Add Machine Back to Domain 118

6.6.2 Adding Machine to Domain Fails 118
6.6.3 I Can't Join a Windows 2003 PDC 119

Chapter 7 STANDALONE SERVERS 121
7.1 Features and Benefits 121
7.2 Background 122
7.3 Example Configuration 122
7.3.1 Reference Documentation Server 122
7.3.2 Central Print Serving 123
7.4 Common Errors 126

Chapter 8 MS WINDOWS NETWORK CONFIGURATION GUIDE 127
8.1 Features and Benefits 127
8.2 Technical Details 127
8.2.1 TCP/IP Configuration 128
8.2.1.1 MS Windows XP Professional 128
8.2.1.2 MS Windows 2000 130
8.2.1.3 MS Windows Me 132
8.2.2 Joining a Domain: Windows 2000/XP Professional 134
8.2.3 Domain Logon Configuration: Windows 9x/Me 136
8.3 Common Errors 138

Part III Advanced Configuration 145

VALUABLE NUTS AND BOLTS INFORMATION 147

Chapter 9 IMPORTANT AND CRITICAL CHANGE NOTES FOR THE SAMBA 3.X SERIES 149
9.1 Important Samba-3.2.x Change Notes 149
9.2 Important Samba-3.0.x Change Notes 149
9.2.1 User and Group Changes 150
9.2.2 Essential Group Mappings 151
9.2.3 Passdb Changes 152
9.2.4 Group Mapping Changes in Samba-3.0.23 152
9.2.5 LDAP Changes in Samba-3.0.23 152

Chapter 10 NETWORK BROWSING 153
10.1 Features and Benefits 154
10.2 What Is Browsing? 155

10.3 Discussion 156
10.3.1 NetBIOS over TCP/IP 157
10.3.2 TCP/IP without NetBIOS 159
10.3.3 DNS and Active Directory 160
10.4 How Browsing Functions 162
10.4.1 Configuring Workgroup Browsing 164
10.4.2 Domain Browsing Configuration 165
10.4.3 Forcing Samba to Be the Master 166
10.4.4 Making Samba the Domain Master 167
10.4.5 Note about Broadcast Addresses 168
10.4.6 Multiple Interfaces 168
10.4.7 Use of the Remote Announce Parameter 169
10.4.8 Use of the Remote Browse Sync Parameter 170
10.5 WINS: The Windows Internetworking Name Server 170
10.5.1 WINS Server Configuration 171
10.5.2 WINS Replication 173
10.5.3 Static WINS Entries 173
10.6 Helpful Hints 174
10.6.1 Windows Networking Protocols 174
10.6.2 Name Resolution Order 175
10.7 Technical Overview of Browsing 176
10.7.1 Browsing Support in Samba 177
10.7.2 Problem Resolution 178
10.7.3 Cross-Subnet Browsing 179
10.7.3.1 Behavior of Cross-Subnet Browsing 179
10.8 Common Errors 183
10.8.1 Flushing the Samba NetBIOS Name Cache 183
10.8.2 Server Resources Cannot Be Listed 184
10.8.3 I Get an "Unable to browse the network" Error 184
10.8.4 Browsing of Shares and Directories is Very Slow 184
10.8.5 Invalid Cached Share References Affects Network Browsing 185

Chapter 11 ACCOUNT INFORMATION DATABASES 187
11.1 Features and Benefits 188
11.1.1 Backward Compatibility Account Storage Systems 188
11.1.2 New Account Storage Systems 189
11.2 Technical Information 190
11.2.1 Important Notes About Security 190
11.2.1.1 Advantages of Encrypted Passwords 193

11.2.1.2 Advantages of Non-Encrypted Passwords 193
11.2.2 Mapping User Identifiers between MS Windows and UNIX 194
11.2.3 Mapping Common UIDs/GIDs on Distributed Machines 194
11.2.4 Comments Regarding LDAP 195
11.2.4.1 Caution Regarding LDAP and Samba 196
11.2.5 LDAP Directories and Windows Computer Accounts 197
11.3 Account Management Tools 198
11.3.1 The smbpasswd Tool 198
11.3.2 The pdbedit Tool 200
11.3.2.1 User Account Management 201
11.3.2.2 Account Import/Export 211
11.4 Password Backends 211
11.4.1 Plaintext 212
11.4.2 smbpasswd: Encrypted Password Database 212
11.4.3 tdbsam 213
11.4.4 ldapsam 213
11.4.4.1 Supported LDAP Servers 215
11.4.4.2 Schema and Relationship to the RFC 2307 posixAccount 215
11.4.4.3 OpenLDAP Configuration 216
11.4.4.4 Initialize the LDAP Database 218
11.4.4.5 Configuring Samba 220
11.4.4.6 Accounts and Groups Management 221
11.4.4.7 Security and sambaSamAccount 221
11.4.4.8 LDAP Special Attributes for sambaSamAccounts 223
11.4.4.9 Example LDIF Entries for a sambaSamAccount 224
11.4.4.10 Password Synchronization 225
11.4.4.11 Using OpenLDAP Overlay for Password Synchronization 225
11.5 Common Errors 226
11.5.1 Users Cannot Logon 226
11.5.2 Configuration of auth methods 226

Chapter 12 GROUP MAPPING: MS WINDOWS AND UNIX 229
12.1 Features and Benefits 230
12.2 Discussion 232
12.2.1 Warning: User Private Group Problems 233

12.2.2 Nested Groups: Adding Windows Domain Groups to Windows Local Groups 234
12.2.3 Important Administrative Information 236
12.2.3.1 Applicable Only to Versions Earlier than 3.0.11 236
12.2.4 Default Users, Groups, and Relative Identifiers 237
12.2.5 Example Configuration 238
12.3 Configuration Scripts 239
12.3.1 Sample smb.conf Add Group Script 239
12.3.2 Script to Configure Group Mapping 240
12.4 Common Errors 241
12.4.1 Adding Groups Fails 241
12.4.2 Adding Domain Users to the Workstation Power Users Group 241

Chapter 13 REMOTE AND LOCAL MANAGEMENT: THE NET COMMAND 243
13.1 Overview 244
13.2 Administrative Tasks and Methods 244
13.3 UNIX and Windows Group Management 245
13.3.1 Adding, Renaming, or Deletion of Group Accounts 245
13.3.1.1 Adding or Creating a New Group 246
13.3.1.2 Mapping Windows Groups to UNIX Groups 248
13.3.1.3 Deleting a Group Account 250
13.3.1.4 Rename Group Accounts 250
13.3.2 Manipulating Group Memberships 251
13.3.3 Nested Group Support 254
13.3.3.1 Managing Nest Groups on Workstations from the Samba Server 255
13.4 UNIX and Windows User Management 256
13.4.1 Adding User Accounts 257
13.4.2 Deletion of User Accounts 257
13.4.3 Managing User Accounts 258
13.4.4 User Mapping 258
13.5 Administering User Rights and Privileges 259
13.6 Managing Trust Relationships 262
13.6.1 Machine Trust Accounts 263
13.6.2 Interdomain Trusts 265
13.7 Managing Security Identifiers (SIDS) 268
13.8 Share Management 269
13.8.1 Creating, Editing, and Removing Shares 270

13.8.2 Creating and Changing Share ACLs 271
13.8.3 Share, Directory, and File Migration 271
13.8.3.1 Share Migration 272
13.8.3.2 File and Directory Migration 274
13.8.3.3 Share-ACL Migration 276
13.8.3.4 Simultaneous Share and File Migration 276
13.8.4 Printer Migration 276
13.9 Controlling Open Files 279
13.10 Session and Connection Management 279
13.11 Printers and ADS 279
13.12 Manipulating the Samba Cache 280
13.13 Managing IDMAP UID/SID Mappings 280
13.13.1 Creating an IDMAP Database Dump File 281
13.13.2 Restoring the IDMAP Database Dump File 281
13.14 Other Miscellaneous Operations 281

Chapter 14 IDENTITY MAPPING (IDMAP) 283
14.1 Samba Server Deployment Types and IDMAP 284
14.1.1 Standalone Samba Server 284
14.1.2 Domain Member Server or Domain Member Client 284
14.1.3 Primary Domain Controller 288
14.1.4 Backup Domain Controller 288
14.2 Examples of IDMAP Backend Usage 289
14.2.1 Default Winbind TDB 289
14.2.1.1 NT4-Style Domains (Includes Samba Domains) 289
14.2.1.2 ADS Domains 291
14.2.2 IDMAP_RID with Winbind 292
14.2.3 IDMAP Storage in LDAP Using Winbind 294
14.2.4 IDMAP and NSS Using LDAP from ADS with RFC2307bis Schema Extension 299
14.2.4.1 IDMAP, Active Directory, and MS Services for UNIX 3.5 300
14.2.4.2 IDMAP, Active Directory and AD4UNIX 300

Chapter 15 USER RIGHTS AND PRIVILEGES 301
15.1 Rights Management Capabilities 302
15.1.1 Using the “net rpc rights” Utility 303
15.1.2 Description of Privileges 305
15.1.3 Privileges Suppored by Windows 2000 Domain Controllers 306

15.2 The Administrator Domain SID 307
15.3 Common Errors 308
15.3.1 What Rights and Privileges Will Permit Windows Client Administration? 308

Chapter 16 FILE, DIRECTORY, AND SHARE ACCESS CONTROLS 311
16.1 Features and Benefits 312
16.2 File System Access Controls 313
16.2.1 MS Windows NTFS Comparison with UNIX File Systems 313
16.2.2 Managing Directories 315
16.2.3 File and Directory Access Control 316
16.2.3.1 Protecting Directories and Files from Deletion 318
16.3 Share Definition Access Controls 320
16.3.1 User- and Group-Based Controls 320
16.3.2 File and Directory Permissions-Based Controls 320
16.3.3 Miscellaneous Controls 320
16.4 Access Controls on Shares 321
16.4.1 Share Permissions Management 323
16.4.1.1 Windows NT4 Workstation/Server 323
16.4.1.2 Windows 200x/XP 323
16.5 MS Windows Access Control Lists and UNIX Interoperability 325
16.5.1 Managing UNIX Permissions Using NT Security Dialogs 325
16.5.2 Viewing File Security on a Samba Share 325
16.5.3 Viewing File Ownership 326
16.5.4 Viewing File or Directory Permissions 326
16.5.4.1 File Permissions 327
16.5.4.2 Directory Permissions 327
16.5.5 Modifying File or Directory Permissions 328
16.5.6 Interaction with the Standard Samba "create mask" Parameters 330
16.5.7 Interaction with the Standard Samba File Attribute Mapping 332
16.5.8 Windows NT/200X ACLs and POSIX ACLs Limitations 332
16.5.8.1 UNIX POSIX ACL Overview 333
16.5.8.2 Mapping of Windows File ACLs to UNIX POSIX ACLs 334
16.5.8.3 Mapping of Windows Directory ACLs to UNIX POSIX ACLs 335

16.6 Common Errors 335
16.6.1 Users Cannot Write to a Public Share 335
16.6.2 File Operations Done as *root* with *force user* Set 337
16.6.3 MS Word with Samba Changes Owner of File 337

Chapter 17 FILE AND RECORD LOCKING 341
17.1 Features and Benefits 341
17.2 Discussion 342
17.2.1 Opportunistic Locking Overview 343
17.2.1.1 Exclusively Accessed Shares 346
17.2.1.2 Multiple-Accessed Shares or Files 346
17.2.1.3 UNIX or NFS Client-Accessed Files 346
17.2.1.4 Slow and/or Unreliable Networks 347
17.2.1.5 Multiuser Databases 347
17.2.1.6 PDM Data Shares 347
17.2.1.7 Beware of Force User 348
17.2.1.8 Advanced Samba Oplocks Parameters 348
17.2.1.9 Mission-Critical, High-Availability 348
17.3 Samba Oplocks Control 349
17.3.1 Example Configuration 350
17.3.1.1 Disabling Oplocks 350
17.3.1.2 Disabling Kernel Oplocks 351
17.4 MS Windows Oplocks and Caching Controls 352
17.4.1 Workstation Service Entries 355
17.4.2 Server Service Entries 356
17.5 Persistent Data Corruption 357
17.6 Common Errors 357
17.6.1 locking.tdb Error Messages 358
17.6.2 Problems Saving Files in MS Office on Windows XP 358
17.6.3 Long Delays Deleting Files over Network with XP SP1 358
17.7 Additional Reading 359

Chapter 18 SECURING SAMBA 361
18.1 Introduction 361
18.2 Features and Benefits 361
18.3 Technical Discussion of Protective Measures and Issues 362
18.3.1 Using Host-Based Protection 362
18.3.2 User-Based Protection 363
18.3.3 Using Interface Protection 363
18.3.4 Using a Firewall 364

18.3.5 Using IPC$ Share-Based Denials 364
18.3.6 NTLMv2 Security 365
18.4 Upgrading Samba 366
18.5 Common Errors 366
18.5.1 Smbclient Works on Localhost, but the Network Is Dead 366
18.5.2 Why Can Users Access Other Users' Home Directories? 366

Chapter 19 INTERDOMAIN TRUST RELATIONSHIPS 369
19.1 Features and Benefits 370
19.2 Trust Relationship Background 370
19.3 Native MS Windows NT4 Trusts Configuration 371
19.3.1 Creating an NT4 Domain Trust 371
19.3.2 Completing an NT4 Domain Trust 372
19.3.3 Interdomain Trust Facilities 372
19.4 Configuring Samba NT-Style Domain Trusts 373
19.4.1 Samba as the Trusted Domain 374
19.4.2 Samba as the Trusting Domain 375
19.5 NT4-Style Domain Trusts with Windows 2000 376
19.6 Common Errors 376
19.6.1 Browsing of Trusted Domain Fails 376
19.6.2 Problems with LDAP ldapsam and Older Versions of smbldap-tools 377

Chapter 20 HOSTING A MICROSOFT DISTRIBUTED FILE SYSTEM TREE 379
20.1 Features and Benefits 379
20.2 Common Errors 380
20.2.1 MSDFS UNIX Path Is Case-Critical 381

Chapter 21 CLASSICAL PRINTING SUPPORT 383
21.1 Features and Benefits 383
21.2 Technical Introduction 384
21.2.1 Client to Samba Print Job Processing 385
21.2.2 Printing-Related Configuration Parameters 385
21.3 Simple Print Configuration 386
21.3.1 Verifying Configuration with testparm 387
21.3.2 Rapid Configuration Validation 388
21.4 Extended Printing Configuration 391
21.4.1 Detailed Explanation Settings 391
21.4.1.1 The [global] Section 392

21.4.1.2 The [printers] Section 394
21.4.1.3 Any [my_printer_name] Section 396
21.4.1.4 Print Commands 397
21.4.1.5 Default UNIX System Printing Commands 398
21.4.1.6 Custom Print Commands 398
21.5 Printing Developments Since Samba-2.2 400
21.5.1 Point'n'Print Client Drivers on Samba Servers 402
21.5.2 The Obsoleted [printer$] Section 402
21.5.3 Creating the [print$] Share 403
21.5.4 [print$] Stanza Parameters 403
21.5.5 The [print$] Share Directory 406
21.6 Installing Drivers into [print$] 407
21.6.1 Add Printer Wizard Driver Installation 407
21.6.2 Installing Print Drivers Using rpcclient 408
21.6.2.1 Identifying Driver Files 409
21.6.2.2 Obtaining Driver Files from Windows Client [print$] Shares 411
21.6.2.3 Installing Driver Files into [print$] 412
21.6.2.4 smbclient to Confirm Driver Installation 413
21.6.2.5 Running rpcclient with adddriver 415
21.6.2.6 Checking adddriver Completion 416
21.6.2.7 Check Samba for Driver Recognition 417
21.6.2.8 Specific Driver Name Flexibility 418
21.6.2.9 Running rpcclient with setdriver 419
21.7 Client Driver Installation Procedure 420
21.7.1 First Client Driver Installation 420
21.7.2 Setting Device Modes on New Printers 421
21.7.3 Additional Client Driver Installation 423
21.7.4 Always Make First Client Connection as root or "printer admin" 424
21.8 Other Gotchas 425
21.8.1 Setting Default Print Options for Client Drivers 425
21.8.2 Supporting Large Numbers of Printers 427
21.8.3 Adding New Printers with the Windows NT APW 429
21.8.4 Error Message: "Cannot connect under a different Name" 431
21.8.5 Take Care When Assembling Driver Files 432
21.8.6 Samba and Printer Ports 435
21.8.7 Avoiding Common Client Driver Misconfiguration 436
21.9 The Imprints Toolset 436

21.9.1 What Is Imprints? 436
21.9.2 Creating Printer Driver Packages 437
21.9.3 The Imprints Server 437
21.9.4 The Installation Client 437
21.10 Adding Network Printers without User Interaction 438
21.11 The addprinter Command 440
21.12 Migration of Classical Printing to Samba 441
21.13 Publishing Printer Information in Active Directory or LDAP 442
21.14 Common Errors 442
21.14.1 I Give My Root Password but I Do Not Get Access 442
21.14.2 My Print Jobs Get Spooled into the Spooling Directory, but Then Get Lost 442

Chapter 22 CUPS PRINTING SUPPORT 445
22.1 Introduction 445
22.1.1 Features and Benefits 445
22.1.2 Overview 445
22.2 Basic CUPS Support Configuration 446
22.2.1 Linking smbd with libcups.so 446
22.2.2 Simple smb.conf Settings for CUPS 447
22.2.3 More Complex CUPS smb.conf Settings 448
22.3 Advanced Configuration 449
22.3.1 Central Spooling vs. "Peer-to-Peer" Printing 450
22.3.2 Raw Print Serving: Vendor Drivers on Windows Clients 450
22.3.3 Installation of Windows Client Drivers 451
22.3.4 Explicitly Enable "raw" Printing for *application/octet-stream* 452
22.3.5 Driver Upload Methods 453
22.4 Advanced Intelligent Printing with PostScript Driver Download 454
22.4.1 GDI on Windows, PostScript on UNIX 454
22.4.2 Windows Drivers, GDI, and EMF 455
22.4.3 UNIX Printfile Conversion and GUI Basics 455
22.4.4 PostScript and Ghostscript 457
22.4.5 Ghostscript: The Software RIP for Non-PostScript Printers 458
22.4.6 PostScript Printer Description (PPD) Specification 459
22.4.7 Using Windows-Formatted Vendor PPDs 460
22.4.8 CUPS Also Uses PPDs for Non-PostScript Printers 461
22.5 The CUPS Filtering Architecture 462
22.5.1 MIME Types and CUPS Filters 463

22.5.2 MIME Type Conversion Rules 464
22.5.3 Filtering Overview 465
22.5.3.1 Filter Requirements 465
22.5.4 Prefilters 466
22.5.5 pstops 466
22.5.6 pstoraster 467
22.5.7 imagetops and imagetoraster 469
22.5.8 rasterto [printers specific] 469
22.5.9 CUPS Backends 470
22.5.10 The Role of cupsomatic/foomatic 473
22.5.11 The Complete Picture 474
22.5.12 mime.convs 474
22.5.13 "Raw" Printing 475
22.5.14 application/octet-stream Printing 475
22.5.15 PostScript Printer Descriptions for Non-PostScript Printers 477
22.5.16 *cupsomatic/foomatic-rip* Versus *Native CUPS* Printing 477
22.5.17 Examples for Filtering Chains 480
22.5.18 Sources of CUPS Drivers/PPDs 481
22.5.19 Printing with Interface Scripts 482
22.6 Network Printing (Purely Windows) 483
22.6.1 From Windows Clients to an NT Print Server 483
22.6.2 Driver Execution on the Client 483
22.6.3 Driver Execution on the Server 484
22.7 Network Printing (Windows Clients and UNIX/Samba Print Servers) 485
22.7.1 From Windows Clients to a CUPS/Samba Print Server 485
22.7.2 Samba Receiving Job-Files and Passing Them to CUPS 486
22.8 Network PostScript RIP 487
22.8.1 PPDs for Non-PS Printers on UNIX 487
22.8.2 PPDs for Non-PS Printers on Windows 488
22.9 Windows Terminal Servers (WTS) as CUPS Clients 488
22.9.1 Printer Drivers Running in "Kernel Mode" Cause Many Problems 488
22.9.2 Workarounds Impose Heavy Limitations 489
22.9.3 CUPS: A "Magical Stone"? 489
22.9.4 PostScript Drivers with No Major Problems, Even in Kernel Mode 489
22.10 Configuring CUPS for Driver Download 490
22.10.1 *cupsaddsmb*: The Unknown Utility 490

22.10.2 Prepare Your smb.conf for cupsaddsmb 491
22.10.3 CUPS "PostScript Driver for Windows NT/200x/XP" 491
22.10.4 Recognizing Different Driver Files 493
22.10.5 Acquiring the Adobe Driver Files 494
22.10.6 ESP Print Pro PostScript Driver for Windows NT/200x/XP 494
22.10.7 Caveats to Be Considered 495
22.10.8 Windows CUPS PostScript Driver Versus Adobe Driver 498
22.10.9 Run cupsaddsmb (Quiet Mode) 499
22.10.10 Run cupsaddsmb with Verbose Output 499
22.10.11 Understanding cupsaddsmb 501
22.10.12 How to Recognize If cupsaddsmb Completed Successfully 502
22.10.13 cupsaddsmb with a Samba PDC 503
22.10.14 cupsaddsmb Flowchart 504
22.10.15 Installing the PostScript Driver on a Client 505
22.10.16 Avoiding Critical PostScript Driver Settings on the Client 505
22.11 Installing PostScript Driver Files Manually Using rpcclient 506
22.11.1 A Check of the rpcclient man Page 507
22.11.2 Understanding the rpcclient man Page 508
22.11.3 Producing an Example by Querying a Windows Box 508
22.11.4 Requirements for adddriver and setdriver to Succeed 509
22.11.5 Manual Driver Installation in 15 Steps 510
22.11.6 Troubleshooting Revisited 517
22.12 The Printing *.tdb Files 518
22.12.1 Trivial Database Files 518
22.12.2 Binary Format 518
22.12.3 Losing *.tdb Files 519
22.12.4 Using tdbbackup 519
22.13 CUPS Print Drivers from Linuxprinting.org 520
22.13.1 foomatic-rip and Foomatic Explained 521
22.13.1.1 690 "Perfect" Printers 522
22.13.1.2 How the Printing HOWTO Started It All 522
22.13.1.3 Foomatic's Strange Name 522
22.13.1.4 cupsomatic, pdqomatic, lpdomatic, directomatic 523
22.13.1.5 The *Grand Unification* Achieved 524
22.13.1.6 Driver Development Outside 525
22.13.1.7 Forums, Downloads, Tutorials, Howtos (Also for Mac OS X and Commercial UNIX) 526
22.13.1.8 Foomatic Database-Generated PPDs 526

22.13.2 foomatic-rip and Foomatic PPD Download and Installation 527
22.14 Page Accounting with CUPS 530
22.14.1 Setting Up Quotas 531
22.14.2 Correct and Incorrect Accounting 531
22.14.3 Adobe and CUPS PostScript Drivers for Windows Clients 532
22.14.4 The page_log File Syntax 532
22.14.5 Possible Shortcomings 533
22.14.6 Future Developments 534
22.14.7 Other Accounting Tools 534
22.15 Additional Material 534
22.16 Autodeletion or Preservation of CUPS Spool Files 536
22.16.1 CUPS Configuration Settings Explained 536
22.16.2 Preconditions 537
22.16.3 Manual Configuration 537
22.17 Printing from CUPS to Windows-Attached Printers 538
22.18 More CUPS Filtering Chains 540
22.19 Common Errors 540
22.19.1 Windows 9x/Me Client Can't Install Driver 540
22.19.2 "cupsaddsmb" Keeps Asking for Root Password in Never-ending Loop 541
22.19.3 "cupsaddsmb" or "rpcclient addriver" Emit Error 541
22.19.4 "cupsaddsmb" Errors 541
22.19.5 Client Can't Connect to Samba Printer 541
22.19.6 New Account Reconnection from Windows 200x/XP Troubles 542
22.19.7 Avoid Being Connected to the Samba Server as the Wrong User 542
22.19.8 Upgrading to CUPS Drivers from Adobe Drivers 542
22.19.9 Can't Use "cupsaddsmb" on Samba Server, Which Is a PDC 543
22.19.10 Deleted Windows 200x Printer Driver Is Still Shown 543
22.19.11 Windows 200x/XP Local Security Policies 543
22.19.12 Administrator Cannot Install Printers for All Local Users 543
22.19.13 Print Change, Notify Functions on NT Clients 543
22.19.14 Windows XP SP1 544
22.19.15 Print Options for All Users Can't Be Set on Windows 200x/XP 544

22.19.16 Most Common Blunders in Driver Settings on Windows Clients 545
22.19.17 cupsaddsmb Does Not Work with Newly Installed Printer 545
22.19.18 Permissions on /var/spool/samba/ Get Reset After Each Reboot 546
22.19.19 Print Queue Called "lp" Mishandles Print Jobs 546
22.19.20 Location of Adobe PostScript Driver Files for "cupsaddsmb" 546
22.20 Overview of the CUPS Printing Processes 546

Chapter 23 STACKABLE VFS MODULES 549
23.1 Features and Benefits 549
23.2 Discussion 549
23.3 Included Modules 550
23.3.1 audit 550
23.3.2 default_quota 551
23.3.3 extd_audit 553
23.3.3.1 Configuration of Auditing 553
23.3.4 fake_perms 554
23.3.5 recycle 554
23.3.6 netatalk 556
23.3.7 shadow_copy 556
23.3.7.1 Shadow Copy Setup 557
23.4 VFS Modules Available Elsewhere 561
23.4.1 DatabaseFS 561
23.4.2 vscan 561
23.4.3 vscan-clamav 562

Chapter 24 WINBIND: USE OF DOMAIN ACCOUNTS 565
24.1 Features and Benefits 565
24.2 Introduction 567
24.3 What Winbind Provides 567
24.3.1 Target Uses 568
24.3.2 Handling of Foreign SIDs 568
24.4 How Winbind Works 569
24.4.1 Microsoft Remote Procedure Calls 569
24.4.2 Microsoft Active Directory Services 570
24.4.3 Name Service Switch 570
24.4.4 Pluggable Authentication Modules 571

24.4.5 User and Group ID Allocation 572
24.4.6 Result Caching 572
24.5 Installation and Configuration 573
24.5.1 Introduction 573
24.5.2 Requirements 573
24.5.3 Testing Things Out 574
24.5.3.1 Configure nsswitch.conf and the Winbind Libraries on Linux and Solaris 574
24.5.3.2 NSS Winbind on AIX 576
24.5.3.3 Configure smb.conf 577
24.5.3.4 Join the Samba Server to the PDC Domain 577
24.5.3.5 Starting and Testing the winbindd Daemon 578
24.5.3.6 Fix the init.d Startup Scripts 580
24.5.3.7 Configure Winbind and PAM 584
24.6 Conclusion 588
24.7 Common Errors 589
24.7.1 NSCD Problem Warning 589
24.7.2 Winbind Is Not Resolving Users and Groups 589

Chapter 25 ADVANCED NETWORK MANAGEMENT 591
25.1 Features and Benefits 591
25.2 Remote Server Administration 591
25.3 Remote Desktop Management 592
25.3.1 Remote Management from NoMachine.Com 592
25.3.2 Remote Management with ThinLinc 594
25.4 Network Logon Script Magic 595
25.4.1 Adding Printers without User Intervention 597
25.4.2 Limiting Logon Connections 598

Chapter 26 SYSTEM AND ACCOUNT POLICIES 601
26.1 Features and Benefits 601
26.2 Creating and Managing System Policies 602
26.2.1 Windows 9x/ME Policies 603
26.2.2 Windows NT4-Style Policy Files 603
26.2.2.1 Registry Spoiling 604
26.2.3 MS Windows 200x/XP Professional Policies 604
26.2.3.1 Administration of Windows 200x/XP Policies 605
26.2.3.2 Custom System Policy Templates 606
26.3 Managing Account/User Policies 607
26.4 Management Tools 608

26.4.1 Samba Editreg Toolset 608
26.4.2 Windows NT4/200x 609
26.4.3 Samba PDC 609
26.5 System Startup and Logon Processing Overview 609
26.6 Common Errors 610
26.6.1 Policy Does Not Work 610

Chapter 27 DESKTOP PROFILE MANAGEMENT 611
27.1 Features and Benefits 611
27.2 Roaming Profiles 611
27.2.1 Samba Configuration for Profile Handling 612
27.2.1.1 NT4/200x User Profiles 612
27.2.1.2 Windows 9x/Me User Profiles 613
27.2.1.3 Mixed Windows Windows 9x/Me and NT4/200x User Profiles 613
27.2.1.4 Disabling Roaming Profile Support 614
27.2.2 Windows Client Profile Configuration Information 615
27.2.2.1 Windows 9x/Me Profile Setup 615
27.2.2.2 Windows NT4 Workstation 618
27.2.2.3 Windows 2000/XP Professional 618
27.2.3 User Profile Hive Cleanup Service 621
27.2.4 Sharing Profiles between Windows 9x/Me and NT4/200x/XP Workstations 621
27.2.5 Profile Migration from Windows NT4/200x Server to Samba 621
27.2.5.1 Windows NT4 Profile Management Tools 622
27.2.5.2 Side Bar Notes 622
27.2.5.3 moveuser.exe 623
27.2.5.4 Get SID 623
27.3 Mandatory Profiles 623
27.4 Creating and Managing Group Profiles 624
27.5 Default Profile for Windows Users 625
27.5.1 MS Windows 9x/Me 625
27.5.1.1 User Profile Handling with Windows 9x/Me 625
27.5.2 MS Windows NT4 Workstation 626
27.5.3 MS Windows 200x/XP 629
27.6 Common Errors 632
27.6.1 Configuring Roaming Profiles for a Few Users or Groups 632
27.6.2 Cannot Use Roaming Profiles 632
27.6.3 Changing the Default Profile 634

27.6.4 Debugging Roaming Profiles and NT4-style Domain Policies 635

Chapter 28 PAM-BASED DISTRIBUTED AUTHENTICATION 637
28.1 Features and Benefits 637
28.2 Technical Discussion 639
28.2.1 PAM Configuration Syntax 639
28.2.1.1 Anatomy of /etc/pam.d Entries 640
28.2.2 Example System Configurations 645
28.2.2.1 PAM: Original Login Config 646
28.2.2.2 PAM: Login Using pam_smbpass 646
28.2.3 smb.conf PAM Configuration 648
28.2.4 Remote CIFS Authentication Using winbindd.so 649
28.2.5 Password Synchronization Using pam_smbpass.so 650
28.2.5.1 Password Synchronization Configuration 650
28.2.5.2 Password Migration Configuration 651
28.2.5.3 Mature Password Configuration 652
28.2.5.4 Kerberos Password Integration Configuration 652
28.3 Common Errors 653
28.3.1 pam_winbind Problem 653
28.3.2 Winbind Is Not Resolving Users and Groups 654

Chapter 29 INTEGRATING MS WINDOWS NETWORKS WITH SAMBA 657
29.1 Features and Benefits 657
29.2 Background Information 658
29.3 Name Resolution in a Pure UNIX/Linux World 658
29.3.1 /etc/hosts 659
29.3.2 /etc/resolv.conf 660
29.3.3 /etc/host.conf 660
29.3.4 /etc/nsswitch.conf 661
29.4 Name Resolution as Used within MS Windows Networking 662
29.4.1 The NetBIOS Name Cache 664
29.4.2 The LMHOSTS File 664
29.4.3 HOSTS File 666
29.4.4 DNS Lookup 666
29.4.5 WINS Lookup 667
29.5 Common Errors 667
29.5.1 Pinging Works Only One Way 668
29.5.2 Very Slow Network Connections 668

29.5.3 Samba Server Name-Change Problem 668

Chapter 30 UNICODE/CHARSETS 671
30.1 Features and Benefits 671
30.2 What Are Charsets and Unicode? 671
30.3 Samba and Charsets 672
30.4 Conversion from Old Names 673
30.5 Japanese Charsets 673
30.5.1 Basic Parameter Setting 674
30.5.2 Individual Implementations 677
30.5.3 Migration from Samba-2.2 Series 678
30.6 Common Errors 679
30.6.1 CP850.so Can't Be Found 679

Chapter 31 BACKUP TECHNIQUES 681
31.1 Features and Benefits 681
31.2 Discussion of Backup Solutions 681
31.2.1 BackupPC 682
31.2.2 Rsync 682
31.2.3 Amanda 683
31.2.4 BOBS: Browseable Online Backup System 683

Chapter 32 HIGH AVAILABILITY 685
32.1 Features and Benefits 685
32.2 Technical Discussion 686
32.2.1 The Ultimate Goal 686
32.2.2 Why Is This So Hard? 686
32.2.2.1 The Front-End Challenge 687
32.2.2.2 Demultiplexing SMB Requests 687
32.2.2.3 The Distributed File System Challenge 688
32.2.2.4 Restrictive Constraints on Distributed File Systems 688
32.2.2.5 Server Pool Communications 689
32.2.2.6 Server Pool Communications Demands 689
32.2.2.7 Required Modifications to Samba 689
32.2.3 A Simple Solution 690
32.2.4 High-Availability Server Products 690
32.2.5 MS-DFS: The Poor Man's Cluster 691
32.2.6 Conclusions 691

Chapter 33 HANDLING LARGE DIRECTORIES 693

Chapter 34 ADVANCED CONFIGURATION TECHNIQUES 695
34.1 Implementation 696
34.1.1 Multiple Server Hosting 696
34.1.2 Multiple Virtual Server Personalities 697
34.1.3 Multiple Virtual Server Hosting 699

Part IV Migration and Updating 701

Chapter 35 UPDATING AND UPGRADING SAMBA 703
35.1 Key Update Requirements 703
35.1.1 Upgrading from Samba-3.0.x to Samba-3.2.0 704
35.1.2 Upgrading from Samba-2.x to Samba-3.0.25 704
35.1.3 Quick Migration Guide 704
35.2 New Featuers in Samba-3.x Series 704
35.2.1 New Features in Samba-3.2.x Series 704
35.2.2 New Features in Samba-3.0.x 704
35.2.2.1 Configuration Parameter Changes 706
35.2.2.2 Removed Parameters 706
35.2.2.3 New Parameters 707
35.2.2.4 Modified Parameters (Changes in Behavior) 712
35.2.3 New Functionality 713
35.2.3.1 TDB Data Files 713
35.2.3.2 Changes in Behavior 713
35.2.3.3 Passdb Backends and Authentication 714
35.2.3.4 LDAP 715

Chapter 36 MIGRATION FROM NT4 PDC TO SAMBA-3 PDC 719
36.1 Planning and Getting Started 719
36.1.1 Objectives 719
36.1.1.1 Domain Layout 721
36.1.1.2 Server Share and Directory Layout 722
36.1.1.3 Logon Scripts 722
36.1.1.4 Profile Migration/Creation 723
36.1.1.5 User and Group Accounts 723
36.1.2 Steps in Migration Process 723
36.2 Migration Options 724

36.2.1 Planning for Success 725
36.2.2 Samba-3 Implementation Choices 725

Chapter 37 SWAT: THE SAMBA WEB ADMINISTRATION TOOL 729
37.1 Features and Benefits 729
37.2 Guidelines and Technical Tips 730
37.2.1 Validate SWAT Installation 730
37.2.1.1 Locating the SWAT File 731
37.2.1.2 Locating the SWAT Support Files 731
37.2.2 Enabling SWAT for Use 733
37.2.3 Securing SWAT through SSL 735
37.2.4 Enabling SWAT Internationalization Support 735
37.3 Overview and Quick Tour 736
37.3.1 The SWAT Home Page 736
37.3.2 Global Settings 737
37.3.3 Share Settings 738
37.3.4 Printers Settings 738
37.3.5 The SWAT Wizard 738
37.3.6 The Status Page 739
37.3.7 The View Page 739
37.3.8 The Password Change Page 739

Part V Troubleshooting 739

Chapter 38 THE SAMBA CHECKLIST 741
38.1 Introduction 741
38.2 Assumptions 741
38.3 The Tests 742

Chapter 39 ANALYZING AND SOLVING SAMBA PROBLEMS 751
39.1 Diagnostics Tools 751
39.1.1 Debugging with Samba Itself 751
39.1.2 Tcpdump 752
39.1.3 Ethereal 752
39.1.4 The Windows Network Monitor 752
39.1.4.1 Installing Network Monitor on an NT Workstation 753

39.1.4.2 Installing Network Monitor on Windows 9x/Me755
39.2 Useful URLs 755
39.3 Getting Mailing List Help 755
39.4 How to Get Off the Mailing Lists 757

Chapter 40 REPORTING BUGS 759
40.1 Introduction 759
40.2 General Information 759
40.3 Debug Levels 760
40.3.1 Debugging-Specific Operations 761
40.4 Internal Errors 761
40.5 Attaching to a Running Process 762
40.6 Patches 763

Chapter 41 MANAGING TDB FILES 765
41.1 Features and Benefits 765
41.2 Managing TDB Files 765

Part VI Reference Section 767

Chapter 42 HOW TO COMPILE SAMBA 769
42.1 Access Samba Source Code via Subversion 769
42.1.1 Introduction 769
42.1.2 Subversion Access to samba.org 769
42.1.2.1 Access via ViewCVS 770
42.1.2.2 Access via Subversion 770
42.2 Accessing the Samba Sources via rsync and ftp 771
42.3 Verifying Samba's PGP Signature 771
42.4 Building the Binaries 772
42.4.1 Compiling Samba with Active Directory Support 774
42.4.1.1 Installing the Required Packages for Debian 774
42.4.1.2 Installing the Required Packages for Red Hat Linux 774
42.4.1.3 SuSE Linux Package Requirements 775
42.5 Starting the smbd nmbd and winbindd 775
42.5.1 Starting from inetd.conf 775
42.5.2 Alternative: Starting smbd as a Daemon 777
42.5.2.1 Starting Samba for Red Hat Linux 778
42.5.2.2 Starting Samba for Novell SUSE Linux 779

Chapter 43 PORTABILITY **781**
43.1 HPUX 781
43.2 SCO UNIX 782
43.3 DNIX 782
43.4 Red Hat Linux 784
43.5 AIX: Sequential Read Ahead 784
43.6 Solaris 785
43.6.1 Locking Improvements 785
43.6.2 Winbind on Solaris 9 785

Chapter 44 SAMBA AND OTHER CIFS CLIENTS **787**
44.1 Macintosh Clients 787
44.2 OS2 Client 788
44.2.1 Configuring OS/2 Warp Connect or OS/2 Warp 4 788
44.2.2 Configuring Other Versions of OS/2 788
44.2.3 Printer Driver Download for OS/2 Clients 789
44.3 Windows for Workgroups 789
44.3.1 Latest TCP/IP Stack from Microsoft 789
44.3.2 Delete .pwl Files After Password Change 790
44.3.3 Configuring Windows for Workgroups Password Handling 790
44.3.4 Password Case Sensitivity 790
44.3.5 Use TCP/IP as Default Protocol 791
44.3.6 Speed Improvement 791
44.4 Windows 95/98 791
44.4.1 Speed Improvement 792
44.5 Windows 2000 Service Pack 2 792
44.6 Windows NT 3.1 793

Chapter 45 SAMBA PERFORMANCE TUNING **795**
45.1 Comparisons 795
45.2 Socket Options 795
45.3 Read Size 796
45.4 Max Xmit 797
45.5 Log Level 797
45.6 Read Raw 797
45.7 Write Raw 797
45.8 Slow Logins 798
45.9 Client Tuning 798
45.10 Samba Performance Problem Due to Changing Linux Kernel 798

45.11 Corrupt tdb Files 799
45.12 Samba Performance is Very Slow 799

Chapter 46 LDAP AND TRANSPORT LAYER SECURITY 801
46.1 Introduction 801
46.2 Configuring 802
46.2.1 Generating the Certificate Authority 802
46.2.2 Generating the Server Certificate 804
46.2.3 Installing the Certificates 806
46.3 Testing 807
46.4 Troubleshooting 809

Chapter 47 SAMBA SUPPORT 811
47.1 Free Support 812
47.2 Commercial Support 813

Chapter 48 DNS AND DHCP CONFIGURATION GUIDE 815
48.1 Features and Benefits 815
48.2 Example Configuration 816
48.2.1 Dynamic DNS 817
48.2.2 DHCP Server 821

Chapter A GNU GENERAL PUBLIC LICENSE VERSION 3 823

GLOSSARY 841

SUBJECT INDEX 847

LIST OF EXAMPLES

Chapter 1

1.2.1 A minimal smb.conf 4
1.2.2 Another simple smb.conf File 7

Chapter 2

2.3.1 Anonymous Read-Only Server Configuration 18
2.3.2 Modified Anonymous Read-Write smb.conf 20
2.3.3 Anonymous Print Server smb.conf 21
2.3.4 Secure Office Server smb.conf 24
2.3.5 Member Server smb.conf (Globals) 28
2.3.6 Member Server smb.conf (Shares and Services) 29
2.3.7 Engineering Office smb.conf (globals) 33
2.3.8 Engineering Office smb.conf (shares and services) 38
2.3.9 LDAP backend smb.conf for PDC 39
2.3.10 Remote LDAP BDC smb.conf 40

Chapter 3

Chapter 4

4.4.1 smb.conf for being a PDC 74
4.6.1 smb.conf for being a PDC 76

Chapter 5

5.2.1 Minimal smb.conf for a PDC in Use with a BDC — LDAP Server on PDC 90
5.2.2 Multiple LDAP Servers in smb.conf 91
5.3.1 Minimal Setup for Being a BDC 95

Chapter 6

Chapter 7

7.3.1 smb.conf for Reference Documentation Server 123
7.3.2 smb.conf for Anonymous Printing 125

Chapter 8

Chapter 9

Chapter 10

10.4.1 Domain Master Browser smb.conf 164
10.4.2 Local master browser smb.conf 165
10.4.3 smb.conf for Not Being a Master Browser 165
10.4.4 Local Master Browser smb.conf 166
10.4.5 smb.conf for Not Being a master browser 166

Chapter 11

11.2.1 Example Configuration with the LDAP idmap Backend 195
11.4.1 Configuration with LDAP 222

Chapter 12

12.3.1 smbgrpadd.sh 239
12.3.2 Configuration of smb.conf for the add group Script 240
12.3.3 Script to Set Group Mapping 240

Chapter 13

13.3.1 Script to Auto-add Domain Users to Workstation Power Users Group 255
13.3.2 A Magic Netlogon Share 256

Chapter 14

14.2.1 NT4 Domain Member Server smb.conf 289
14.2.2 ADS Domain Member Server smb.conf 291
14.2.3 ADS Domain Member smb.conf using idmap_rid 293
14.2.4 ADS Domain Member Server using LDAP 295

14.2.5 ADS Domain Member Server using RFC2307bis Schema Extension Date via NSS 299

Chapter 15

Chapter 16

16.2.1 Example File 317

Chapter 17

17.3.1 Share with Some Files Oplocked 352
17.3.2 Configuration with Oplock Break Contention Limit 352

Chapter 18

Chapter 19

Chapter 20

20.1.1 smb.conf with DFS Configured 380

Chapter 21

21.3.1 Simple Configuration with BSD Printing 386
21.4.1 Extended BSD Printing Configuration 392
21.5.1 [print$] Example 404

Chapter 22

22.2.1 Simplest Printing-Related smb.conf 448
22.2.2 Overriding Global CUPS Settings for One Printer 449
22.10.1 smb.conf for cupsaddsmb Usage 491

Chapter 23

23.2.1 smb.conf with VFS modules 550
23.2.2 smb.conf with multiple VFS modules 550
23.3.1 Share With shadow_copy VFS 560

Chapter 24

24.5.1 smb.conf for Winbind Setup 577

Chapter 25

25.4.1 Script to Enforce Single Resource Logon 599

Chapter 26

Chapter 27

Chapter 28

Chapter 29

Chapter 30

30.5.1 VFS CAP 677

Chapter 31

Chapter 32

Chapter 33

Chapter 34

34.1.1 Elastic smb.conf File 698
34.1.2 CDROM Server smb-cdserver.conf file 699
34.1.3 Master smb.conf File Global Section 700
34.1.4 MERLIN smb-merlin.conf File Share Section 701
34.1.5 SAURON smb-sauron.conf File Share Section 701

Chapter 35

Chapter 36

Chapter 37

Chapter 38

38.2.1 smb.conf with [tmp] Share 742
38.3.1 Configuration for Allowing Connections Only from a Certain Subnet 745
38.3.2 Configuration for Allowing Connections from a Certain Subnet and localhost 746

Chapter 39

Chapter 40

Chapter 41

Chapter 42

Chapter 43

Chapter 44

44.5.1 Minimal Profile Share 792

Chapter 45

Chapter 46

Chapter 47

Chapter 48

List of Figures

4 Domain Control
4.1 An Example Domain. 60

8 MS Windows Network Configuration Guide
8.1 Network Bridge Configuration. 129
8.2 Internet Protocol (TCP/IP) Properties. 130
8.3 Advanced Network Settings 131
8.4 DNS Configuration. 132
8.5 WINS Configuration 133
8.6 Local Area Connection Properties. 134
8.7 Internet Protocol (TCP/IP) Properties. 135
8.8 Advanced Network Settings. 136
8.9 DNS Configuration. 137
8.10 WINS Configuration. 138
8.11 The Windows Me Network Configuration Panel. 139
8.12 IP Address. 140
8.13 DNS Configuration. 140
8.14 WINS Configuration. 141
8.15 The General Panel. 141
8.16 The Computer Name Panel. 142
8.17 The Computer Name Changes Panel. 142
8.18 The Computer Name Changes Panel — Domain MIDEARTH. 143
8.19 Computer Name Changes — Username and Password Panel. 143
8.20 The Network Panel. 144
8.21 Client for Microsoft Networks Properties Panel. 144
8.22 Identification Panel. 145
8.23 Access Control Panel. 145

10 Network Browsing
10.1 Cross-Subnet Browsing Example. 180

11 Account Information Databases
11.1 IDMAP: Resolution of SIDs to UIDs. 191
11.2 IDMAP: Resolution of UIDs to SIDs. 192

12 Group Mapping: MS Windows and UNIX
12.1 IDMAP: Group SID-to-GID Resolution. 230
12.2 IDMAP: GID Resolution to Matching SID. 231
12.3 IDMAP Storing Group Mappings. 231

16 File, Directory, and Share Access Controls
16.1 Overview of UNIX permissions field. 317

19 Interdomain Trust Relationships
19.1 Trusts overview. 372

22 CUPS Printing Support
22.1 Windows Printing to a Local Printer. 456
22.2 Printing to a PostScript Printer. 458
22.3 Ghostscript as a RIP for Non-PostScript Printers. 458
22.4 Prefiltering in CUPS to Form PostScript. 467
22.5 Adding Device-Specific Print Options. 467
22.6 PostScript to Intermediate Raster Format. 468
22.7 CUPS-Raster Production Using Ghostscript. 469
22.8 Image Format to CUPS-Raster Format Conversion. 470
22.9 Raster to Printer-Specific Formats. 471
22.10 cupsomatic/foomatic Processing Versus Native CUPS. 479
22.11 PDF to Socket Chain. 480
22.12 PDF to USB Chain. 481
22.13 Print Driver Execution on the Client. 484
22.14 Print Driver Execution on the Server. 484
22.15 Printing via CUPS/Samba Server. 486
22.16 cupsaddsmb Flowchart. 504
22.17 Filtering Chain 1. 540
22.18 Filtering Chain with cupsomatic 547
22.19 CUPS Printing Overview. 548

24 Winbind: Use of Domain Accounts

24.1 Winbind Idmap 566

39 Analyzing and Solving Samba Problems
39.1 Starting a Capture. 753
39.2 Main Ethereal Data Window. 754

List of Tables

1 How to Install and Test SAMBA
1.1 Persistent TDB File Descriptions 6
1.2 Temporary TDB File Descriptions 14

5 Backup Domain Control
5.1 Domain Backend Account Distribution Options 87

6 Domain Membership
6.1 Assumptions 108

9 Important and Critical Change Notes for the Samba 3.x Series
9.1 Essential Domain Group Mappings 152

10 Network Browsing
10.1 Browse Subnet Example 1 181
10.2 Browse Subnet Example 2 182
10.3 Browse Subnet Example 3 182
10.4 Browse Subnet Example 4 183

11 Account Information Databases
11.1 NT4 Domain v's Samba Policy Controls 202
11.2 Samba SAM Account Control Block Flags 208
11.3 Attributes in the sambaSamAccount ObjectClass (LDAP), Part A 227
11.4 Attributes in the sambaSamAccount ObjectClass (LDAP), Part B 228
11.5 Possible ldap passwd sync Values 228

12 Group Mapping: MS Windows and UNIX
12.1 Well-Known User Default RIDs 238

15 User Rights and Privileges
15.1 Current Privilege Capabilities 303

16 File, Directory, and Share Access Controls
16.1 Managing Directories with UNIX and Windows 316
16.2 User- and Group-Based Controls 321
16.3 File and Directory Permission-Based Controls 322
16.4 Other Controls 339
16.5 How Windows File ACLs Map to UNIX POSIX File ACLs 340

21 Classical Printing Support
21.1 Default Printing Settings 399

22 CUPS Printing Support
22.1 PPDs Shipped with CUPS 478

23 Stackable VFS modules
23.1 Extended Auditing Log Information 553

27 Desktop Profile Management
27.1 User Shell Folder Registry Keys Default Values 628
27.2 Defaults of Profile Settings Registry Keys 628
27.3 Defaults of Default User Profile Paths Registry Keys 631

28 PAM-Based Distributed Authentication
28.1 Options recognized by pam_smbpass 651

29 Integrating MS Windows Networks with Samba
29.1 Unique NetBIOS Names 662
29.2 Group Names 662

30 Unicode/Charsets
30.1 Japanese Character Sets in Samba-2.2 and Samba-3 679

35 Updating and Upgrading Samba

35.1 Samba-2.2.x TDB File Descriptions 714

36 Migration from NT4 PDC to Samba-3 PDC
36.1 The Three Major Site Types 725
36.2 Nature of the Conversion Choices 726

40 Reporting Bugs
40.1 Debuggable Functions 761

41 Managing TDB Files
41.1 Samba's Trivial Database Files 767

FOREWORD

When John first asked me to write an introductory piece for his latest book, I was somewhat mystified as to why he chose me. A conversation with John provided some of the rationale, and he left it to me to fill in the *rest* of the story. So, if you are willing to endure a little bit of background, I will provide the part of the story that John wouldn't provide.

I am the Director of Corporate Standards at Sun Microsystems, and manage Sun's standards portfolio. Before that, I was the Director of Standards at Netscape, which was when I met John. Before Sun, there was Digital Equipment Corporation, also standards. I've written several books on standards, and tend to observe (and occasionally help) the technical and business trends that drive standardization as a discipline. I tend to see standardization as a management tool, not as a technical discipline and this is part of the rationale that John provided.

The book that you have before you focuses on a particular standardized way of doing something hence, it is a book about a standard. The most important thing to keep in mind about a standard is the rationale for its creation. Standards are created not for technical reasons, not for business reasons, but for a deeper and much more compelling reason. Standards are created and used to allow people to communicate in a meaningful way. Every standard, if it is a true standard, has as its entire (and only) goal set the increasing of relevant communication between people.

This primary goal cannot be met however, unless the standard is documented. I have been involved in too many standardization efforts when it became apparent that *everybody knows* was the dominant emotion of those providing documentation. *They* of the ever present *they say* and *they know* are the bane of good standards. If *they know*, why are you doing a standard?

A *good standard* survives because people know how to use it. People know how to use a standard when it is so transparent, so obvious, and so easy that it becomes invisible. And a standard becomes invisible only when the documentation describing how to deploy it is clear, unambiguous, and correct. These three elements must be present for a standard to be useful, allowing communication and interaction between two separate and distinct entities

to occur without obvious effort. As you read this book, look for the evidence of these three characteristics and notice how they are seamlessly woven into John's text. Clarity and unambiguity without *correctness* provide a technical nightmare. Correctness and clarity with ambiguity create *maybe bits,* and correctness and unambiguity without clarity provide a *muddle through* scenario.

And this is *the rest of the story* that John couldn't (or wouldn't) bring himself to state. This book provides a clear, concise, unambiguous, and technically valid presentation of Samba to make it useful to a user to someone who wants to use the standard to increase communication and the capability for communication between two or more entities whether person-machine, machine-machine, or person-person. The intent of this book is not to convince anyone of any agenda political, technical, or social. The intent is to provide documentation for users who need to know about Samba, how to use it, and how to get on with their primary responsibilities. While there is pride on John's part because of the tremendous success of the Samba documentation, he writes for the person who needs a tool to accomplish a particular job, and who has selected Samba to be that tool.

The book is a monument to John's perseverance and dedication to Samba and in my opinion to the goal of standardization. By writing this book, John has provided the users of Samba those that want to deploy it to make things better a clear, easy, and ultimately valuable resource. Additionally, he has increased the understanding and utility of a highly useful standard, and for this, as much as for the documentation, he is owed a debt of gratitude by those of us who rely on standards to make our lives more manageable.

Carl Cargill, Senior Director
Corporate Standardization, The Office of the CTO
Sun Microsystems

PREFACE

The editors wish to thank you for your decision to purchase this book. The Official Samba-3 HOWTO and Reference Guide is the result of many years of accumulation of information, feedback, tips, hints, and happy solutions.

Please note that this book is a living document, the contents of which are constantly being updated. We encourage you to contribute your tips, techniques, helpful hints, and your special insight into the Windows networking world to help make the next generation of this book even more valuable to Samba users.

We have made a concerted effort to document more comprehensively than has been done previously the information that may help you to better deploy Samba and to gain more contented network users.

This book provides example configurations, it documents key aspects of Microsoft Windows networking, provides in-depth insight into the important configuration of Samba-3, and helps to put all of these into a useful framework.

The most recent electronic versions of this document can be found at <`http://www.samba.org/`> on the "Documentation" page.

Updates, patches and corrections are most welcome. Please email your contributions to any one of the following:

Jelmer Vernooij (jelmer@samba.org)[1]
John H. Terpstra (jht@samba.org)[2]
Gerald (Jerry) Carter (jerry@samba.org)[3]

We wish to advise that only original and unencumbered material can be published. Please do not submit content that is not your own work unless proof of consent from the copyright holder accompanies your submission.

Conventions Used

The following notation conventions are used throughout this book:

- TOSHARG2 is used as an abbreviation for the book, "The Official Samba-3 HOWTO and Reference Guide, Second Edition" Editors: John H. Terpstra and Jelmer R. Vernooij, Publisher: Prentice Hall, ISBN: 0131882228.
- S3bE2 is used as an abbreviation for the book, "Samba-3 by Example, Second Edition" Editors: John H. Terpstra, Publisher: Prentice Hall, ISBN: 013188221X.
- Directories and filenames appear in mono-font. For example, `/etc/pam.conf`.
- Executable names are bolded. For example, **smbd**.
- Menu items and buttons appear in bold. For example, click **Next**.
- Selecting a menu item is indicated as: **Start** → **Control Panel** → **Administrative Tools** → **Active Directory Users and Computers**

INTRODUCTION

"A man's gift makes room for him before great men. Gifts are like hooks that can catch hold of the mind taking it beyond the reach of forces that otherwise might constrain it." — Anon.

This is a book about Samba. It is a tool, a derived work of the labors of many and of the diligence and goodwill of more than a few. This book contains material that has been contributed in a persistent belief that each of us can add value to our neighbors as well as to those who will follow us.

This book is designed to meet the needs of the Microsoft network administrator. UNIX administrators will benefit from this book also, though they may complain that it is hard to find the information they think they need. So if you are a Microsoft certified specialist, this book should meet your needs rather well. If you are a UNIX or Linux administrator, there is no need to feel badly — you should have no difficulty finding answers to your current concerns also.

What Is Samba?

Samba is a big, complex project. The Samba project is ambitious and exciting. The team behind Samba is a group of some thirty individuals who are spread the world over and come from an interesting range of backgrounds. This team includes scientists, engineers, programmers, business people, and students.

Team members were drawn into active participation through the desire to help deliver an exciting level of transparent interoperability between Microsoft Windows and the non-Microsoft information technology world.

The slogan that unites the efforts behind the Samba project says: *Samba, Opening Windows to a Wider World!* The goal behind the project is one of removing barriers to interoperability.

Samba provides file and print services for Microsoft Windows clients. These services may be hosted off any TCP/IP-enabled platform. The original

deployment platforms were UNIX and Linux, though today it is in common use across a broad variety of systems.

The Samba project includes not only an impressive feature set in file and print serving capabilities, but has been extended to include client functionality, utilities to ease migration to Samba, tools to aid interoperability with Microsoft Windows, and administration tools.

The real people behind Samba are users like you. You have inspired the developers (the Samba Team) to do more than any of them imagined could or should be done. User feedback drives Samba development. Samba-3 in particular incorporates a huge amount of work done as a result of user requests, suggestions and direct code contributions.

Why This Book?

There is admittedly a large number of Samba books on the market today and each book has its place. Despite the apparent plethora of books, Samba as a project continues to receive much criticism for failing to provide sufficient documentation. Samba is also criticized for being too complex and too difficult to configure. In many ways this is evidence of the success of Samba as there would be no complaints if it was not successful.

The Samba Team members work predominantly with UNIX and Linux, so it is hardly surprising that existing Samba documentation should reflect that orientation. The original HOWTO text documents were intended to provide some tips, a few golden nuggets, and if they helped anyone then that was just wonderful. But the HOWTO documents lacked structure and context. They were isolated snapshots of information that were written to pass information on to someone else who might benefit. They reflected a need to transmit more information that could be conveniently put into manual pages.

The original HOWTO documents were written by different authors. Most HOWTO documents are the result of feedback and contributions from numerous authors. In this book we took care to preserve as much original content as possible. As you read this book you will note that chapters were written by multiple authors, each of whom has his own style. This demonstrates the nature of the Open Source software development process.

Out of the original HOWTO documents sprang a collection of unofficial HOWTO documents that are spread over the Internet. It is sincerely in-

tended that this work will *not* replace the valuable unofficial HOWTO work that continues to flourish. If you are involved in unofficial HOWTO production then please continue your work!

Those of you who have dedicated your labors to the production of unofficial HOWTOs, to Web page information regarding Samba, or to answering questions on the mailing lists or elsewhere, may be aware that this is a labor of love. We would like to know about your contribution and willingly receive the precious pearls of wisdom you have collected. Please email your contribution to John H. Terpstra (jht@samba.org)[4]. As a service to other users we will gladly adopt material that is technically accurate.

Existing Samba books are largely addressed to the UNIX administrator. From the perspective of this target group the existing books serve an adequate purpose, with one exception — now that Samba-3 is out they need to be updated!

This book, the *Official Samba-3 HOWTO and Reference Guide*, includes the Samba-HOWTO-Collection.pdf that ships with Samba. These documents have been written with a new design intent and purpose.

Over the past two years many Microsoft network administrators have adopted Samba and have become interested in its deployment. Their information needs are very different from that of the UNIX administrator. This book has been arranged and the information presented from the perspective of someone with previous Microsoft Windows network administrative training and experience.

Book Structure and Layout

This book is presented in six parts:

General Installation Designed to help you get Samba-3 running quickly. The Fast Start chapter is a direct response to requests from Microsoft network administrators for some sample configurations that *just work*.

Server Configuration Basics The purpose of this section is to aid the transition from existing Microsoft Windows network knowledge to

[4] <mailto:jht@samba.org>

Samba terminology and norms. The chapters in this part each cover the installation of one type of Samba server.

Advanced Configuration The mechanics of network browsing have long been the Achilles heel of all Microsoft Windows users. Samba-3 introduces new user and machine account management facilities, a new way to map UNIX groups and Windows groups, Interdomain trusts, new loadable file system drivers (VFS), and more. New with this document is expanded printing documentation, as well as a wealth of information regarding desktop and user policy handling, use of desktop profiles, and techniques for enhanced network integration. This section makes up the core of the book. Read and enjoy.

Migration and Updating A much requested addition to the book is information on how to migrate from Microsoft Windows NT4 to Samba-3, as well as an overview of what the issues are when moving from Samba-2.x to Samba-3.

Troubleshooting This short section should help you when all else fails.

Reference Section Here you will find a collection of things that are either too peripheral for most users, or are a little left of field to be included in the main body of information.

Welcome to Samba-3 and the first published document to help you and your users to enjoy a whole new world of interoperability between Microsoft Windows and the rest of the world.

Part I

General Installation

PREPARING SAMBA FOR CONFIGURATION

This section of the Samba-HOWTO-Collection contains general info on how to install Samba and how to configure the parts of Samba you will most likely need. PLEASE read this.

Chapter 1

HOW TO INSTALL AND TEST SAMBA

1.1 Obtaining and Installing Samba

Binary packages of Samba are included in almost any Linux or UNIX distribution. There are also some packages available at the Samba home page[1]. Refer to the manual of your operating system for details on installing packages for your specific operating system.

If you need to compile Samba from source, check Chapter 42, "How to Compile Samba".

1.2 Configuring Samba (smb.conf)

Samba's configuration is stored in the `smb.conf` file, which usually resides in `/etc/samba/smb.conf` or `/usr/local/samba/lib/smb.conf`. You can either edit this file yourself or do it using one of the many graphical tools that are available, such as the Web-based interface SWAT, that is included with Samba.

1.2.1 Configuration File Syntax

The `smb.conf` file uses the same syntax as the various old `.ini` files in Windows 3.1: Each file consists of various sections, which are started by putting the section name between brackets (`[]`) on a new line. Each contains

[1] <http://samba.org/>

zero or more key/value pairs separated by an equality sign (=). The file is just a plaintext file, so you can open and edit it with your favorite editing tool.

Each section in the `smb.conf` file represents either a share or a meta-service on the Samba server. The section `[global]` is special, since it contains settings that apply to the whole Samba server. Samba supports a number of meta-services, each of which serves its own purpose. For example, the `[homes]` share is a meta-service that causes Samba to provide a personal home share for each user. The `[printers]` share is a meta-service that establishes print queue support and that specifies the location of the intermediate spool directory into which print jobs are received from Windows clients prior to being dispatched to the UNIX/Linux print spooler.

The `printers` meta-service will cause every printer that is either specified in a `printcap` file, via the **lpstat**, or via the CUPS API, to be published as a shared print queue. The `printers` stanza in the `smb.conf` file can be set as not browseable. If it is set to be browseable, then it will be visible as if it is a share. That makes no sense given that this meta-service is responsible only for making UNIX system printers available as Windows print queues. If a `comment` parameter is specified, the value of it will be displayed as part of the printer name in Windows Explorer browse lists.

Each section of the `smb.conf` file that specifies a share, or a meta-service, is called a stanza. The `global` stanza specifies settings that affect all the other stanzas in the `smb.conf` file. Configuration parameters are documented in the `smb.conf` man page. Some parameters can be used only in the `global` stanza, some only in share or meta-service stanzas, and some can be used globally or just within a share or meta-service stanza.

Example 1.2.1 contains a very minimal `smb.conf`.

Example 1.2.1 A minimal smb.conf

```
[global]
        workgroup = WKG
        netbios name = MYNAME
 [share1]
        path = /tmp
 [share2]
        path = /my_shared_folder
        comment = Some random files
```

1.2.2 TDB Database File Information

This section contains brief descriptions of the databases that are used by Samba-3.

The directory in which Samba stores the tdb files is determined by compile-time directives. Samba-3 stores tdb files in two locations. The best way to determine these locations is to execute the following command:

```
root#  smbd -b | grep PRIVATE_DIR
   PRIVATE_DIR: /etc/samba/private
```

This means that the confidential tdb files are stored in the `/etc/samba/private` directory. Samba-3 also uses a number of tdb files that contain more mundane data. The location of these files can be found by executing:

```
root#  smbd -b | grep LOCKDIR
   LOCKDIR: /var/lib/samba
```

Therefore the remaining control files will, in the example shown, be stored in the `/var/lib/samba` directory.

The persistent tdb files are described in Table 1.1. All persistent tdb files should be regularly backed up. Use the **tdbbackup** utility to backup the tdb files. All persistent tdb files must be preserved during machine migrations, updates and upgrades.

The temporary tdb files do not need to be backed up, nor do they need to be preseved across machine migrations, updates or upgrades. The temporary tdb files are described in Table 1.2.

1.2.3 Starting Samba

Samba essentially consists of two or three daemons. A daemon is a UNIX application that runs in the background and provides services. An example of a service is the Apache Web server for which the daemon is called **httpd**. In the case of Samba there are three daemons, two of which are needed as a minimum.

Table 1.1 Persistent TDB File Descriptions

Name	Description
account_policy	Samba/NT account policy settings, includes password expiration settings.
group_mapping	Mapping table from Windows groups/SID to UNIX groups.
ntdrivers	Stores per-printer installed driver information.
ntforms	Stores per-printer installed forms information.
ntprinters	Stores the per-printer devmode configuration settings.
passdb	Exists only when the tdbsam passwd backend is used. This file stores the SambaSAMAccount information. Note: This file requires that user POSIX account information is availble from either the /etc/passwd file, or from an alternative system source.
registry	Read-only Samba database of a Windows registry skeleton that provides support for exporting various database tables via the winreg RPCs.
secrets	This file stores the Workgroup/Domain/Machine SID, the LDAP directory update password, and a further collection of critical environmental data that is necessary for Samba to operate correctly. This file contains very sensitive information that must be protected. It is stored in the PRIVATE_DIR directory.
share_info	Stores per-share ACL information.
winbindd_idmap	Winbindd's local IDMAP database.

The Samba server is made up of the following daemons:

nmbd This daemon handles all name registration and resolution requests. It is the primary vehicle involved in network browsing. It handles all UDP-based protocols. The **nmbd** daemon should be the first command started as part of the Samba startup process.

smbd This daemon handles all TCP/IP-based connection services for file- and print-based operations. It also manages local authentication. It should be started immediately following the startup of **nmbd**.

winbindd This daemon should be started when Samba is a member of a Windows NT4 or ADS domain. It is also needed when Samba has trust relationships with another domain. The **winbindd** daemon will check the `smb.conf` file for the presence of the *idmap uid* and *idmap gid* parameters. If they are are found, **winbindd** will use the values specified for for UID and GID allocation. If these parameters are not specified, **winbindd** will start but it will not be able to allocate UIDs or GIDs.

When Samba has been packaged by an operating system vendor, the startup process is typically a custom feature of its integration into the platform as a whole. Please refer to your operating system platform administration manuals for specific information pertaining to correct management of Samba startup.

1.2.4 Example Configuration

There are sample configuration files in the examples subdirectory in the source code distribution tarball package. It is suggested you read them carefully so you can see how the options go together in practice. See the man page for all the options. It might be worthwhile to start out with the `smb.conf.default` configuration file and adapt it to your needs. It contains plenty of comments.

The simplest useful configuration file would contain something like that shown in Example 1.2.2.

Example 1.2.2 Another simple smb.conf File

```
[global]
        workgroup = MIDEARTH
 [homes]
        guest ok = no
        read only = no
```

This will allow connections by anyone with an account on the server, using either their login name or *homes* as the service name. (Note: The workgroup that Samba should appear in must also be set. The default workgroup name is WORKGROUP.)

Make sure you put the `smb.conf` file in the correct place. Note, the correct location of this file depends on how the binary files were built. You can discover the correct location by executing from the directory that contains the **smbd** command file:

```
root#  smbd -b | grep smb.conf
```

For more information about security settings for the *[homes]* share, please refer to Chapter 18, "Securing Samba".

1.2.4.1 Test Your Config File with testparm

It's important to validate the contents of the `smb.conf` file using the testparm program. If testparm runs correctly, it will list the loaded services. If not, it will give an error message. Make sure it runs correctly and that the services look reasonable before proceeding. Enter the command:

```
root#  testparm /etc/samba/smb.conf
```

Testparm will parse your configuration file and report any unknown parameters or incorrect syntax. It also performs a check for common misconfigurations and will issue a warning if one is found.

Always run testparm again whenever the `smb.conf` file is changed!

The `smb.conf` file is constantly checked by the Samba daemons **smbd** and every instance of itself that it spawns, **nmbd** and **winbindd**. It is good practice to keep this file as small as possible. Many administrators prefer to document Samba configuration settings and thus the need to keep this file small goes against good documentation wisdom. One solution that may be adopted is to do all documentation and configuration in a file that has another name, such as `smb.conf.master`. The **testparm** utility can be used to generate a fully optimized `smb.conf` file from this master configuration and documentation file as shown here:

```
root#  testparm -s smb.conf.master > smb.conf
```

This administrative method makes it possible to maintain detailed configuration change records while at the same time keeping the working `smb.conf` file size to the minimum necessary.

1.2.5 SWAT

SWAT is a Web-based interface that can be used to facilitate the configuration of Samba. SWAT might not be available in the Samba package that shipped with your platform, but in a separate package. If you need to build SWAT please read the SWAT man page regarding compilation, installation, and configuration of SWAT from the source code.

To launch SWAT, just run your favorite Web browser and point it to <`http://localhost:901/`>. Replace *`localhost`* with the name of the computer on which Samba is running if that is a different computer than your browser.

SWAT can be used from a browser on any IP-connected machine, but be aware that connecting from a remote machine leaves your connection open to password sniffing because passwords will be sent over the wire in the clear.

Please note that re-writing the configuration file using SWAT will remove all comments! More information about SWAT can be found in Chapter 37, "SWAT: The Samba Web Administration Tool".

1.3 List Shares Available on the Server

To list shares that are available from the configured Samba server, execute the following command:

```
$ smbclient -L yourhostname
```

You should see a list of shares available on your server. If you do not, then something is incorrectly configured. This method can also be used to see what shares are available on other SMB servers, such as Windows 2000.

If you choose user-level security, you may find that Samba requests a password before it will list the shares. See the **smbclient** man page for details.

You can force it to list the shares without a password by adding the option `-N` to the command line.

1.4 Connect with a UNIX Client

Enter the following command:

```
$ smbclient  //yourhostname/aservice
```

Typically *yourhostname* is the name of the host on which smbd has been installed. The *aservice* is any service that has been defined in the `smb.conf` file. Try your username if you just have a *[homes]* section in the `smb.conf` file.

Example: If the UNIX host is called *bambi* and a valid login name is *fred*, you would type:

```
$ smbclient //bambi/fred
```

1.5 Connect from a Remote SMB Client

Now that Samba is working correctly locally, you can try to access it from other clients. Within a few minutes, the Samba host should be listed in the Network Neighborhood on all Windows clients of its subnet. Try browsing the server from another client or "mounting" it.

Mounting disks from a DOS, Windows, or OS/2 client can be done by running a command such as:

```
C:\> net use m: \\servername\service
```

Where the drive letter m: is any available drive letter. It is important to double-check that the service (share) name that you used does actually exist.

Try printing, for example,

```
C:\> net use lpt1:   \\servername\spoolservice
```

The `spoolservice` is the name of the printer (actually the print queue) on the target server. This will permit all print jobs that are captured by the lpt1: port on the Windows client to be sent to the printer that owns the spoolservice that has been specified.

```
C:\> print filename
```

1.5.1 What If Things Don't Work?

You might want to read Chapter 38, "The Samba Checklist". If you are still stuck, refer to Chapter 39, "Analyzing and Solving Samba Problems". Samba has been successfully installed at thousands of sites worldwide. It is unlikely that your particular problem is unique, so it might be productive to perform an Internet search to see if someone else has encountered your problem and has found a way to overcome it.

If you are new to Samba, and particularly if you are new to Windows networking, or to UNIX/Linux, the book "Samba-3 by Example" will help you to create a validated network environment. Simply choose from the first five chapters the network design that most closely matches site needs, then follow the simple step-by-step procedure to deploy it. Later, when you have a working network you may well want to refer back to this book for further insight into opportunities for improvement.

1.5.2 Still Stuck?

The best advice under the stress of abject frustration is to cool down! That may be challenging of itself, but while you are angry or annoyed your ability to seek out a solution is somewhat undermined. A cool head clears the way to finding the answer you are looking for. Just remember, every problem has a solution — there is a good chance that someone else has found it even though you can't right now. That will change with time, patience and learning.

Now that you have cooled down a bit, please refer to Chapter 38, "The Samba Checklist" for a process that can be followed to identify the cause of your problem.

1.6 Common Errors

The following questions and issues are raised repeatedly on the Samba mailing list.

1.6.1 Large Number of smbd Processes

Samba consists of three core programs: nmbd, smbd, and winbindd. nmbd is the name server message daemon, smbd is the server message daemon, and winbindd is the daemon that handles communication with domain controllers.

If Samba is *not* running as a WINS server, then there will be one single instance of nmbd running on your system. If it is running as a WINS server, then there will be two instances — one to handle the WINS requests.

smbd handles all connection requests. It spawns a new process for each client connection made. That is why you may see so many of them, one per client connection.

winbindd will run as one or two daemons, depending on whether or not it is being run in *split mode* (in which case there will be two instances).

1.6.2 Error Message: open_oplock_ipc

An error message is observed in the log files when smbd is started: "open_oplock_ipc: Failed to get local UDP socket for address 100007f. Error was Cannot assign requested."

Your loopback device isn't working correctly. Make sure it is configured correctly. The loopback device is an internal (virtual) network device with the IP address *127.0.0.1*. Read your OS documentation for details on how to configure the loopback on your system.

1.6.3 "The network name cannot be found"

This error can be caused by one of these misconfigurations:

- You specified a nonexisting path for the share in `smb.conf`.
- The user you are trying to access the share with does not have sufficient permissions to access the path for the share. Both read (r) and access (x) should be possible.
- The share you are trying to access does not exist.

Table 1.2 Temporary TDB File Descriptions

Name	Description	Backup
brlock	Byte-range locking information.	No
connections	A temporary cache for current connection information used to enforce max connections.	no
eventlog/*tdb	Records of eventlog entries. In most circumstances this is just a cache of system logs.	no
gencache	Generic caching database for dead WINS servers and trusted domain data.	no
login_cache	A temporary cache for login information, in particular bad password attempts.	no
messages	Temporary storage of messages being processed by smbd.	no
netsamlogon_cache	Caches user net_info_3 structure data from net_samlogon requests (as a domain member).	no
perfmon/*.tdb	Performance counter information.	no
printing/*.tdb	Cached output from lpq command created on a per-print-service basis.	no
schannel_store	A confidential file, stored in the PRIVATE_DIR, containing crytographic connection information so that clients that have temporarily disconnected can reconnect without needing to renegotiate the connection setup process.	no
sessionid	Temporary cache for miscellaneous session information and for utmp handling.	no
unexpected	Stores packets received for which no process is actively listening.	no
winbindd_cache	Cache of Identity information received from an NT4 domain or from ADS. Includes user lists, etc.	yes

Chapter 2

FAST START: CURE FOR IMPATIENCE

When we first asked for suggestions for inclusion in the Samba HOWTO documentation, someone wrote asking for example configurations — and lots of them. That is remarkably difficult to do without losing a lot of value that can be derived from presenting many extracts from working systems. That is what the rest of this document does. It does so with extensive descriptions of the configuration possibilities within the context of the chapter that covers it. We hope that this chapter is the medicine that has been requested.

The information in this chapter is very sparse compared with the book "Samba-3 by Example" that was written after the original version of this book was nearly complete. "Samba-3 by Example" was the result of feedback from reviewers during the final copy editing of the first edition. It was interesting to see that reader feedback mirrored that given by the original reviewers. In any case, a month and a half was spent in doing basic research to better understand what new as well as experienced network administrators would best benefit from. The book "Samba-3 by Example" is the result of that research. What is presented in the few pages of this book is covered far more comprehensively in the second edition of "Samba-3 by Example". The second edition of both books will be released at the same time.

So in summary, the book "The Official Samba-3 HOWTO & Reference Guide" is intended as the equivalent of an auto mechanic's repair guide. The book "Samba-3 by Example" is the equivalent of the driver's guide that explains how to drive the car. If you want complete network configuration examples, go to Samba-3 by Example[1].

[1] <http://www.samba.org/samba/docs/Samba3-ByExample.pdf>

2.1 Features and Benefits

Samba needs very little configuration to create a basic working system. In this chapter we progress from the simple to the complex, for each providing all steps and configuration file changes needed to make each work. Please note that a comprehensively configured system will likely employ additional smart features. These additional features are covered in the remainder of this document.

The examples used here have been obtained from a number of people who made requests for example configurations. All identities have been obscured to protect the guilty, and any resemblance to unreal nonexistent sites is deliberate.

2.2 Description of Example Sites

In the first set of configuration examples we consider the case of exceptionally simple system requirements. There is a real temptation to make something that should require little effort much too complex.

Section 2.3.1.1 documents the type of server that might be sufficient to serve CD-ROM images, or reference document files for network client use. This configuration is also discussed in Chapter 7, "Standalone Servers", Section 7.3.1. The purpose for this configuration is to provide a shared volume that is read-only that anyone, even guests, can access.

The second example shows a minimal configuration for a print server that anyone can print to as long as they have the correct printer drivers installed on their computer. This is a mirror of the system described in Chapter 7, "Standalone Servers", Section 7.3.2.

The next example is of a secure office file and print server that will be accessible only to users who have an account on the system. This server is meant to closely resemble a workgroup file and print server, but has to be more secure than an anonymous access machine. This type of system will typically suit the needs of a small office. The server provides no network logon facilities, offers no domain control; instead it is just a network-attached storage (NAS) device and a print server.

The later example consider more complex systems that will either integrate into existing MS Windows networks or replace them entirely. These cover

domain member servers as well as Samba domain control (PDC/BDC) and finally describes in detail a large distributed network with branch offices in remote locations.

2.3 Worked Examples

The configuration examples are designed to cover everything necessary to get Samba running. They do not cover basic operating system platform configuration, which is clearly beyond the scope of this text.

It is also assumed that Samba has been correctly installed, either by way of installation of the packages that are provided by the operating system vendor or through other means.

2.3.1 Standalone Server

A standalone server implies no more than the fact that it is not a domain controller and it does not participate in domain control. It can be a simple, workgroup-like server, or it can be a complex server that is a member of a domain security context.

As the examples are developed, every attempt is made to progress the system toward greater capability, just as one might expect would happen in a real business office as that office grows in size and its needs change.

2.3.1.1 Anonymous Read-Only Document Server

The purpose of this type of server is to make available to any user any documents or files that are placed on the shared resource. The shared resource could be a CD-ROM drive, a CD-ROM image, or a file storage area.

- The file system share point will be `/export`.
- All files will be owned by a user called Jack Baumbach. Jack's login name will be *jackb*. His password will be *m0r3pa1n* — of course, that's just the example we are using; do not use this in a production environment because all readers of this document will know it.

Installation Procedure: Read-Only Server

1. Add user to system (with creation of the user's home directory):

Example 2.3.1 Anonymous Read-Only Server Configuration

```
# Global parameters
 [global]
        workgroup = MIDEARTH
        netbios name = HOBBIT
        security = share
 [data]
        comment = Data
        path = /export
        read only = Yes
        guest ok = Yes
```

```
root# useradd -c "Jack Baumbach" -m -g users -p m0r3pa1n jackb
```

2. Create directory, and set permissions and ownership:

```
root# mkdir /export
root# chmod u+rwx,g+rx,o+rx /export
root# chown jackb.users /export
```

3. Copy the files that should be shared to the `/export` directory.
4. Install the Samba configuration file (`/etc/samba/smb.conf`) as shown in Example 2.3.1.
5. Test the configuration file by executing the following command:

```
root# testparm
```

Alternatively, where you are operating from a master configuration file called `smb.conf.master`, the following sequence of commands might prove more appropriate:

```
root#  cd /etc/samba
root#  testparm -s smb.conf.master > smb.conf
```

```
root#  testparm
```

Note any error messages that might be produced. Proceed only if error-free output has been obtained. An example of typical output that should be generated from the above configuration file is shown here:

```
Load smb config files from /etc/samba/smb.conf
Processing section "[data]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
[Press enter]

# Global parameters
[global]
   workgroup = MIDEARTH
   netbios name = HOBBIT
   security = share

[data]
   comment = Data
   path = /export
   read only = Yes
   guest only = Yes
```

6. Start Samba using the method applicable to your operating system platform. The method that should be used is platform dependent. Refer to Section 42.5 for further information regarding the starting of Samba.

7. Configure your MS Windows client for workgroup *MIDEARTH*, set the machine name to ROBBINS, reboot, wait a few (2 - 5) minutes, then open Windows Explorer and visit the Network Neighborhood. The machine HOBBIT should be visible. When you click this machine icon, it should open up to reveal the *data* share. After you click the share, it should open up to reveal the files previously placed in the `/export` directory.

The information above (following # Global parameters) provides the complete contents of the `/etc/samba/smb.conf` file.

2.3.1.2 Anonymous Read-Write Document Server

We should view this configuration as a progression from the previous example. The difference is that shared access is now forced to the user identity of jackb and to the primary group jackb belongs to. One other refinement we can make is to add the user *jackb* to the `smbpasswd` file. To do this, execute:

```
root# smbpasswd -a jackb
New SMB password: m0r3pa1n
Retype new SMB password: m0r3pa1n
Added user jackb.
```

Addition of this user to the `smbpasswd` file allows all files to be displayed in the Explorer Properties boxes as belonging to *jackb* instead of to *User Unknown*.

The complete, modified `smb.conf` file is as shown in Example 2.3.2.

Example 2.3.2 Modified Anonymous Read-Write smb.conf

```
# Global parameters
 [global]
        workgroup = MIDEARTH
        netbios name = HOBBIT
        security = SHARE
 [data]
        comment = Data
        path = /export
        force user = jackb
        force group = users
        read only = No
        guest ok = Yes
```

2.3.1.3 Anonymous Print Server

An anonymous print server serves two purposes:

- It allows printing to all printers from a single location.
- It reduces network traffic congestion due to many users trying to access a limited number of printers.

In the simplest of anonymous print servers, it is common to require the installation of the correct printer drivers on the Windows workstation. In this case the print server will be designed to just pass print jobs through to the spooler, and the spooler should be configured to do raw pass-through to the printer. In other words, the print spooler should not filter or process the data stream being passed to the printer.

In this configuration, it is undesirable to present the Add Printer Wizard, and we do not want to have automatic driver download, so we disable it in the following configuration. Example 2.3.3 is the resulting `smb.conf` file.

Example 2.3.3 Anonymous Print Server smb.conf

```
# Global parameters
 [global]
        workgroup = MIDEARTH
        netbios name = LUTHIEN
        security = share
        printcap name = cups
        disable spoolss = Yes
        show add printer wizard = No
        printing = cups
 [printers]
        comment = All Printers
        path = /var/spool/samba
        guest ok = Yes
        printable = Yes
        use client driver = Yes
        browseable = No
```

The above configuration is not ideal. It uses no smart features, and it deliberately presents a less than elegant solution. But it is basic, and it does print. Samba makes use of the direct printing application program interface that is provided by CUPS. When Samba has been compiled and linked with the CUPS libraries, the default printing system will be CUPS. By specifying that the printcap name is CUPS, Samba will use the CUPS library API to communicate directly with CUPS for all printer functions. It is possible to force the use of external printing commands by setting the

value of the *printing* to either SYSV or BSD, and thus the value of the parameter *printcap name* must be set to something other than CUPS. In such case, it could be set to the name of any file that contains a list of printers that should be made available to Windows clients.

NOTE

Windows users will need to install a local printer and then change the print to device after installation of the drivers. The print to device can then be set to the network printer on this machine.

Make sure that the directory `/var/spool/samba` is capable of being used as intended. The following steps must be taken to achieve this:

- The directory must be owned by the superuser (root) user and group:

  ```
  root# chown root.root /var/spool/samba
  ```

- Directory permissions should be set for public read-write with the sticky bit set as shown:

  ```
  root# chmod a+twrx /var/spool/samba
  ```

  The purpose of setting the sticky bit is to prevent who does not own the temporary print file from being able to take control of it with the potential for devious misuse.

NOTE

On CUPS-enabled systems there is a facility to pass raw data directly to the printer without intermediate processing via CUPS print filters. Where use of this mode of operation is desired, it is necessary to configure a raw printing device. It is also necessary to enable the raw mime handler in the `/etc/mime.conv` and `/etc/mime.types` files. Refer to Section 22.3.4.

2.3.1.4 Secure Read-Write File and Print Server

We progress now from simple systems to a server that is slightly more complex.

Our new server will require a public data storage area in which only authenticated users (i.e., those with a local account) can store files, as well as a home directory. There will be one printer that should be available for everyone to use.

In this hypothetical environment (no espionage was conducted to obtain this data), the site is demanding a simple environment that is *secure enough* but not too difficult to use.

Site users will be Jack Baumbach, Mary Orville, and Amed Sehkah. Each will have a password (not shown in further examples). Mary will be the printer administrator and will own all files in the public share.

This configuration will be based on *user-level security* that is the default, and for which the default is to store Microsoft Windows-compatible encrypted passwords in a file called `/etc/samba/smbpasswd`. The default `smb.conf` entry that makes this happen is *passdb backend* = smbpasswd, guest. Since this is the default, it is not necessary to enter it into the configuration file. Note that the guest backend is added to the list of active passdb backends no matter whether it specified directly in Samba configuration file or not. Installing the Secure Office Server

1. Add all users to the operating system:

Example 2.3.4 Secure Office Server smb.conf

```
# Global parameters
 [global]
        workgroup = MIDEARTH
        netbios name = OLORIN
        printcap name = cups
        disable spoolss = Yes
        show add printer wizard = No
        printing = cups
 [homes]
        comment = Home Directories
        valid users = %S
        read only = No
        browseable = No
 [public]
        comment = Data
        path = /export
        force user = maryo
        force group = users
        read only = No
 [printers]
        comment = All Printers
        path = /var/spool/samba
        printer admin = root, maryo
        create mask = 0600
        guest ok = Yes
        printable = Yes
        use client driver = Yes
        browseable = No
```

```
root# useradd -c "Jack Baumbach" -m -g users -p m0r3pa1n jackb
root# useradd -c "Mary Orville" -m -g users -p secret maryo
root# useradd -c "Amed Sehkah" -m -g users -p secret ameds
```

2. Configure the Samba `smb.conf` file as shown in Example 2.3.4.

3. Initialize the Microsoft Windows password database with the new users:

```
root# smbpasswd -a root
New SMB password: bigsecret
Reenter smb password: bigsecret
Added user root.

root# smbpasswd -a jackb
New SMB password: m0r3pa1n
Retype new SMB password: m0r3pa1n
Added user jackb.

root# smbpasswd -a maryo
New SMB password: secret
Reenter smb password: secret
Added user maryo.

root# smbpasswd -a ameds
New SMB password: mysecret
Reenter smb password: mysecret
Added user ameds.
```

4. Install printer using the CUPS Web interface. Make certain that all printers that will be shared with Microsoft Windows clients are installed as raw printing devices.

5. Start Samba using the operating system administrative interface. Alternately, this can be done manually by executing:

```
root#  nmbd; smbd;
```

   Both applications automatically execute as daemons. Those who are paranoid about maintaining control can add the `-D` flag to coerce them to start up in daemon mode.

6. Configure the `/export` directory:

```
root# mkdir /export
root# chown maryo.users /export
root# chmod u=rwx,g=rwx,o-rwx /export
```

7. Check that Samba is running correctly:

```
root# smbclient -L localhost -U%
Domain=[MIDEARTH] OS=[UNIX] Server=[Samba-3.0.20]

Sharename       Type      Comment
---------       ----      -------
public          Disk      Data
IPC$            IPC       IPC Service (Samba-3.0.20)
ADMIN$          IPC       IPC Service (Samba-3.0.20)
hplj4           Printer   hplj4

Server               Comment
---------            -------
OLORIN               Samba-3.0.20

Workgroup            Master
---------            -------
MIDEARTH             OLORIN
```

The following error message indicates that Samba was not running:

```
root#  smbclient -L olorin -U%
Error connecting to 192.168.1.40 (Connection refused)
Connection to olorin failed
```

8. Connect to OLORIN as maryo:

```
root# smbclient //olorin/maryo -Umaryo%secret
OS=[UNIX] Server=[Samba-3.0.20]
smb: \> dir
.                              D        0  Sat Jun 21 10:58:16 2003
..                             D        0  Sat Jun 21 10:54:32 2003
Documents                      D        0  Fri Apr 25 13:23:58 2003
DOCWORK                        D        0  Sat Jun 14 15:40:34 2003
OpenOffice.org                 D        0  Fri Apr 25 13:55:16 2003
.bashrc                        H     1286  Fri Apr 25 13:23:58 2003
```

```
  .netscape6                    DH        0  Fri Apr 25 13:55:13 2003
  .mozilla                      DH        0  Wed Mar  5 11:50:50 2003
  .kermrc                        H      164  Fri Apr 25 13:23:58 2003
  .acrobat                      DH        0  Fri Apr 25 15:41:02 2003

        55817 blocks of size 524288. 34725 blocks available
  smb: \> q
```

By now you should be getting the hang of configuration basics. Clearly, it is time to explore slightly more complex examples. For the remainder of this chapter we abbreviate instructions, since there are previous examples.

2.3.2 Domain Member Server

In this instance we consider the simplest server configuration we can get away with to make an accounting department happy. Let's be warned, the users are accountants and they do have some nasty demands. There is a budget for only one server for this department.

The network is managed by an internal Information Services Group (ISG), to which we belong. Internal politics are typical of a medium-sized organization; Human Resources is of the opinion that they run the ISG because they are always adding and disabling users. Also, departmental managers have to fight tooth and nail to gain basic network resources access for their staff. Accounting is different, though, they get exactly what they want. So this should set the scene.

We use the users from the last example. The accounting department has a general printer that all departmental users may use. There is also a check printer that may be used only by the person who has authority to print checks. The chief financial officer (CFO) wants that printer to be completely restricted and for it to be located in the private storage area in her office. It therefore must be a network printer.

The accounting department uses an accounting application called *SpytFull* that must be run from a central application server. The software is licensed to run only off one server, there are no workstation components, and it is run off a mapped share. The data store is in a UNIX-based SQL backend. The UNIX gurus look after that, so this is not our problem.

The accounting department manager (maryo) wants a general filing system as well as a separate file storage area for form letters (nastygrams). The form letter area should be read-only to all accounting staff except the manager. The general filing system has to have a structured layout with a general area for all staff to store general documents as well as a separate file area for each member of her team that is private to that person, but she wants full access to all areas. Users must have a private home share for personal work-related files and for materials not related to departmental operations.

2.3.2.1 Example Configuration

The server *valinor* will be a member server of the company domain. Accounting will have only a local server. User accounts will be on the domain controllers, as will desktop profiles and all network policy files.

Example 2.3.5 Member Server smb.conf (Globals)

```
# Global parameters
 [global]
        workgroup = MIDEARTH
        netbios name = VALINOR
        security = DOMAIN
        printcap name = cups
        disable spoolss = Yes
        show add printer wizard = No
        idmap uid = 15000-20000
        idmap gid = 15000-20000
        winbind use default domain = Yes
        printing = cups
```

1. Do not add users to the UNIX/Linux server; all of this will run off the central domain.
2. Configure `smb.conf` according to Example 2.3.5 and Example 2.3.6.
3. Join the domain. Note: Do not start Samba until this step has been completed!

```
root# net rpc join -Uroot%'bigsecret'
Joined domain MIDEARTH.
```

Example 2.3.6 Member Server smb.conf (Shares and Services)

```
[homes]
        comment = Home Directories
        valid users = %S
        read only = No
        browseable = No
[spytfull]
        comment = Accounting Application Only
        path = /export/spytfull
        valid users = @Accounts
        admin users = maryo
        read only = Yes
[public]
        comment = Data
        path = /export/public
        read only = No
[printers]
        comment = All Printers
        path = /var/spool/samba
        printer admin = root, maryo
        create mask = 0600
        guest ok = Yes
        printable = Yes
        use client driver = Yes
        browseable = No
```

4. Make absolutely certain that you disable (shut down) the **nscd** daemon on any system on which **winbind** is configured to run.

5. Start Samba following the normal method for your operating system platform. If you wish to do this manually, execute as root:

   ```
   root# nmbd; smbd; winbindd;
   ```

6. Configure the name service switch (NSS) control file on your system to resolve user and group names via winbind. Edit the following lines in `/etc/nsswitch.conf`:

   ```
   passwd: files winbind
   ```

```
group:  files winbind
hosts:  files dns winbind
```

7. Set the password for **wbinfo** to use:

```
root# wbinfo --set-auth-user=root%'bigsecret'
```

8. Validate that domain user and group credentials can be correctly resolved by executing:

```
root# wbinfo -u
MIDEARTH\maryo
MIDEARTH\jackb
MIDEARTH\ameds
...
MIDEARTH\root

root# wbinfo -g
MIDEARTH\Domain Users
MIDEARTH\Domain Admins
MIDEARTH\Domain Guests
...
MIDEARTH\Accounts
```

9. Check that **winbind** is working. The following demonstrates correct username resolution via the **getent** system utility:

```
root# getent passwd maryo
maryo:x:15000:15003:Mary Orville:/home/MIDEARTH/maryo:/bin/false
```

10. A final test that we have this under control might be reassuring:

```
root# touch /export/a_file
root# chown maryo /export/a_file
root# ls -al /export/a_file
...
```

```
-rw-r--r--    1 maryo    users       11234 Jun 21 15:32 a_file
...

root# rm /export/a_file
```

11. Configuration is now mostly complete, so this is an opportune time to configure the directory structure for this site:

```
root# mkdir -p /export/{spytfull,public}
root# chmod ug=rwxS,o=x /export/{spytfull,public}
root# chown maryo.Accounts /export/{spytfull,public}
```

2.3.3 Domain Controller

For the remainder of this chapter the focus is on the configuration of domain control. The examples that follow are for two implementation strategies. Remember, our objective is to create a simple but working solution. The remainder of this book should help to highlight opportunity for greater functionality and the complexity that goes with it.

A domain controller configuration can be achieved with a simple configuration using the new tdbsam password backend. This type of configuration is good for small offices, but has limited scalability (cannot be replicated), and performance can be expected to fall as the size and complexity of the domain increases.

The use of tdbsam is best limited to sites that do not need more than a Primary Domain Controller (PDC). As the size of a domain grows the need for additional domain controllers becomes apparent. Do not attempt to under-resource a Microsoft Windows network environment; domain controllers provide essential authentication services. The following are symptoms of an under-resourced domain control environment:

- Domain logons intermittently fail.
- File access on a domain member server intermittently fails, giving a permission denied error message.

A more scalable domain control authentication backend option might use Microsoft Active Directory or an LDAP-based backend. Samba-3 provides

for both options as a domain member server. As a PDC, Samba-3 is not able to provide an exact alternative to the functionality that is available with Active Directory. Samba-3 can provide a scalable LDAP-based PDC/BDC solution.

The tdbsam authentication backend provides no facility to replicate the contents of the database, except by external means (i.e., there is no self-contained protocol in Samba-3 for Security Account Manager database [SAM] replication).

NOTE

If you need more than one domain controller, do not use a tdbsam authentication backend.

2.3.3.1 Example: Engineering Office

The engineering office network server we present here is designed to demonstrate use of the new tdbsam password backend. The tdbsam facility is new to Samba-3. It is designed to provide many user and machine account controls that are possible with Microsoft Windows NT4. It is safe to use this in smaller networks.

1. A working PDC configuration using the tdbsam password backend can be found in Example 2.3.7 together with Example 2.3.8:
2. Create UNIX group accounts as needed using a suitable operating system tool:

```
root# groupadd ntadmins
root# groupadd designers
root# groupadd engineers
root# groupadd qateam
```

3. Create user accounts on the system using the appropriate tool provided with the operating system. Make sure all user home directories are

Example 2.3.7 Engineering Office smb.conf (globals)

```
[global]
        workgroup = MIDEARTH
        netbios name = FRODO
        passdb backend = tdbsam
        printcap name = cups
        add user script = /usr/sbin/useradd -m %u
        delete user script = /usr/sbin/userdel -r %u
        add group script = /usr/sbin/groupadd %g
        delete group script = /usr/sbin/groupdel %g
        add user to group script = /usr/sbin/groupmod -A ↩
            %u %g
        delete user from group script = /usr/sbin/ ↩
            groupmod -R %u %g
        add machine script = /usr/sbin/useradd -s /bin/ ↩
            false -d /var/lib/nobody %u
# Note: The following specifies the default logon script ↩
    .
# Per user logon scripts can be specified in the user ↩
    account using pdbedit
        logon script = scripts\logon.bat
# This sets the default profile path. Set per user paths ↩
     with pdbedit
        logon path = \\%L\Profiles\%U
        logon drive = H:
        logon home = \\%L\%U
        domain logons = Yes
        os level = 35
        preferred master = Yes
        domain master = Yes
        idmap uid = 15000-20000
        idmap gid = 15000-20000
        printing = cups
```

created also. Add users to groups as required for access control on files, directories, printers, and as required for use in the Samba environment.

4. Assign each of the UNIX groups to NT groups by executing this shell script (You could name the script `initGroups.sh`):

```
#!/bin/bash
```

```
#### Keep this as a shell script for future re-use

# First assign well known groups
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmins rid=512 type=d
net groupmap add ntgroup="Domain Users"  unixgroup=users rid=513 type=
net groupmap add ntgroup="Domain Guests" unixgroup=nobody rid=514 type=d

# Now for our added Domain Groups
net groupmap add ntgroup="Designers" unixgroup=designers type=d
net groupmap add ntgroup="Engineers" unixgroup=engineers type=d
net groupmap add ntgroup="QA Team"   unixgroup=qateam    type=d
```

5. Create the `scripts` directory for use in the *[NETLOGON]* share:

   ```
   root# mkdir -p /var/lib/samba/netlogon/scripts
   ```

   Place the logon scripts that will be used (batch or cmd scripts) in this directory.

The above configuration provides a functional PDC system to which must be added file shares and printers as required.

2.3.3.2 A Big Organization

In this section we finally get to review in brief a Samba-3 configuration that uses a Lightweight Directory Access (LDAP)-based authentication backend. The main reasons for this choice are to provide the ability to host primary and Backup Domain Control (BDC), as well as to enable a higher degree of scalability to meet the needs of a very distributed environment.

The Primary Domain Controller This is an example of a minimal configuration to run a Samba-3 PDC using an LDAP authentication backend. It is assumed that the operating system has been correctly configured.

The Idealx scripts (or equivalent) are needed to manage LDAP-based POSIX and/or SambaSamAccounts. The Idealx scripts may be downloaded from the Idealx[2] Web site. They may also be obtained from the Samba tarball.

[2] <http://www.idealx.org>

Linux distributions tend to install the Idealx scripts in the `/usr/share/doc/packages/sambaXXXXXX/examples/LDAP/smbldap-tools` directory. Idealx scripts version `smbldap-tools-0.9.1` are known to work well.

1. Obtain from the Samba sources `~/examples/LDAP/samba.schema` and copy it to the `/etc/openldap/schema/` directory.

2. Set up the LDAP server. This example is suitable for OpenLDAP 2.1.x. The `/etc/openldap/slapd.conf` file. Example slapd.conf File

```
# Note commented out lines have been removed
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/samba.schema

pidfile         /var/run/slapd/slapd.pid
argsfile        /var/run/slapd/slapd.args

database        bdb
suffix          "dc=quenya,dc=org"
rootdn          "cn=Manager,dc=quenya,dc=org"
rootpw          {SSHA}06qDkonA8hk6W6SSnRzWj0/pBcU3m0/P
# The password for the above is 'nastyon3'

directory     /var/lib/ldap

index   objectClass     eq
index cn                      pres,sub,eq
index sn                      pres,sub,eq
index uid                     pres,sub,eq
index displayName             pres,sub,eq
index uidNumber               eq
index gidNumber               eq
index memberUid               eq
index   sambaSID              eq
index   sambaPrimaryGroupSID  eq
index   sambaDomainName       eq
index   default               sub
```

3. Create the following file `initdb.ldif`:

```
# Organization for SambaXP Demo
dn: dc=quenya,dc=org
objectclass: dcObject
objectclass: organization
dc: quenya
o: SambaXP Demo
description: The SambaXP Demo LDAP Tree

# Organizational Role for Directory Management
dn: cn=Manager,dc=quenya,dc=org
objectclass: organizationalRole
cn: Manager
description: Directory Manager

# Setting up the container for users
dn: ou=People, dc=quenya, dc=org
objectclass: top
objectclass: organizationalUnit
ou: People

# Set up an admin handle for People OU
dn: cn=admin, ou=People, dc=quenya, dc=org
cn: admin
objectclass: top
objectclass: organizationalRole
objectclass: simpleSecurityObject
userPassword: {SSHA}0jBHgQ1vp4EDX2rEMMfIudvRMJoGwjVb
# The password for above is 'mordonL8'
```

4. Load the initial data above into the LDAP database:

```
root# slapadd -v -l initdb.ldif
```

5. Start the LDAP server using the appropriate tool or method for the

operating system platform on which it is installed.

6. Install the Idealx script files in the `/usr/local/sbin` directory, then configure the smbldap_conf.pm file to match your system configuration.

7. The `smb.conf` file that drives this backend can be found in example Example 2.3.9. Add additional stanzas as required.

8. Add the LDAP password to the `secrets.tdb` file so Samba can update the LDAP database:

```
root# smbpasswd -w mordonL8
```

9. Add users and groups as required. Users and groups added using Samba tools will automatically be added to both the LDAP backend and the operating system as required.

Backup Domain Controller Example 2.3.10 shows the example configuration for the BDC. Note that the `smb.conf` file does not specify the smbldap-tools scripts — they are not needed on a BDC. Add additional stanzas for shares and printers as required.

1. Decide if the BDC should have its own LDAP server or not. If the BDC is to be the LDAP server, change the following `smb.conf` as indicated. The default configuration in Example 2.3.10 uses a central LDAP server.

2. Configure the NETLOGON and PROFILES directory as for the PDC in Example 2.3.10.

Example 2.3.8 Engineering Office smb.conf (shares and services)

```
[homes]
        comment = Home Directories
        valid users = %S
        read only = No
        browseable = No
# Printing auto-share (makes printers available thru ↩
    CUPS)
[printers]
        comment = All Printers
        path = /var/spool/samba
        printer admin = root, maryo
        create mask = 0600
        guest ok = Yes
        printable = Yes
        browseable = No
[print$]
        comment = Printer Drivers Share
        path = /var/lib/samba/drivers
        write list = maryo, root
        printer admin = maryo, root
# Needed to support domain logons
[netlogon]
        comment = Network Logon Service
        path = /var/lib/samba/netlogon
        admin users = root, maryo
        guest ok = Yes
        browseable = No
# For profiles to work, create a user directory under ↩
    the path
# shown. i.e., mkdir -p /var/lib/samba/profiles/maryo
[Profiles]
        comment = Roaming Profile Share
        path = /var/lib/samba/profiles
        read only = No
        profile acls = Yes
# Other resource (share/printer) definitions would ↩
    follow below.
```

Example 2.3.9 LDAP backend smb.conf for PDC

```
# Global parameters
 [global]
        workgroup = MIDEARTH
        netbios name = FRODO
        passdb backend = ldapsam:ldap://localhost
        username map = /etc/samba/smbusers
        printcap name = cups
        add user script = /usr/local/sbin/smbldap-useradd ↩
            -m '%u'
        delete user script = /usr/local/sbin/smbldap- ↩
            userdel %u
        add group script = /usr/local/sbin/smbldap- ↩
            groupadd -p '%g'
        delete group script = /usr/local/sbin/smbldap- ↩
            groupdel '%g'
        add user to group script = /usr/local/sbin/ ↩
            smbldap-groupmod -m '%u' '%g'
        delete user from group script = /usr/local/sbin/ ↩
            smbldap-groupmod -x '%u' '%g'
        set primary group script = /usr/local/sbin/ ↩
            smbldap-usermod -g '%g' '%u'
        add machine script = /usr/local/sbin/smbldap- ↩
            useradd -w '%u'
        logon script = scripts\logon.bat
        logon path = \\%L\Profiles\%U
        logon drive = H:
        logon home = \\%L\%U
        domain logons = Yes
        os level = 35
        preferred master = Yes
        domain master = Yes
        ldap suffix = dc=quenya,dc=org
        ldap machine suffix = ou=People
        ldap user suffix = ou=People
        ldap group suffix = ou=People
        ldap idmap suffix = ou=People
        ldap admin dn = cn=Manager
        ldap ssl = no
        ldap passwd sync = Yes
        idmap uid = 15000-20000
        idmap gid = 15000-20000
        printing = cups
```

Example 2.3.10 Remote LDAP BDC smb.conf

```
# Global parameters
 [global]
        workgroup = MIDEARTH
        netbios name = GANDALF
        passdb backend = ldapsam:ldap://frodo.quenya.org
        username map = /etc/samba/smbusers
        printcap name = cups
        logon script = scripts\logon.bat
        logon path = \\%L\Profiles\%U
        logon drive = H:
        logon home = \\%L\%U
        domain logons = Yes
        os level = 33
        preferred master = Yes
        domain master = No
        ldap suffix = dc=quenya,dc=org
        ldap machine suffix = ou=People
        ldap user suffix = ou=People
        ldap group suffix = ou=People
        ldap idmap suffix = ou=People
        ldap admin dn = cn=Manager
        ldap ssl = no
        ldap passwd sync = Yes
        idmap uid = 15000-20000
        idmap gid = 15000-20000
        printing = cups
```

Part II

Server Configuration Basics

FIRST STEPS IN SERVER CONFIGURATION

Samba can operate in various modes within SMB networks. This HOWTO section contains information on configuring Samba to function as the type of server your network requires. Please read this section carefully.

Chapter 3

SERVER TYPES AND SECURITY MODES

This chapter provides information regarding the types of server that Samba may be configured to be. A Microsoft network administrator who wishes to migrate to or use Samba will want to know the meaning, within a Samba context, of terms familiar to the MS Windows administrator. This means that it is essential also to define how critical security modes function before we get into the details of how to configure the server itself.

This chapter provides an overview of the security modes of which Samba is capable and how they relate to MS Windows servers and clients.

A question often asked is, "Why would I want to use Samba?" Most chapters contain a section that highlights features and benefits. We hope that the information provided will help to answer this question. Be warned though, we want to be fair and reasonable, so not all features are positive toward Samba. The benefit may be on the side of our competition.

3.1 Features and Benefits

Two men were walking down a dusty road, when one suddenly kicked up a small red stone. It hurt his toe and lodged in his sandal. He took the stone out and cursed it with a passion and fury befitting his anguish. The other looked at the stone and said, "This is a garnet. I can turn that into a precious gem and some day it will make a princess very happy!"

The moral of this tale: Two men, two very different perspectives regarding the same stone. Like it or not, Samba is like that stone. Treat it the right

way and it can bring great pleasure, but if you are forced to use it and have no time for its secrets, then it can be a source of discomfort.

Samba started out as a project that sought to provide interoperability for MS Windows 3.x clients with a UNIX server. It has grown up a lot since its humble beginnings and now provides features and functionality fit for large-scale deployment. It also has some warts. In sections like this one, we tell of both.

So, what are the benefits of the features mentioned in this chapter?

- Samba-3 can replace an MS Windows NT4 domain controller.
- Samba-3 offers excellent interoperability with MS Windows NT4-style domains as well as natively with Microsoft Active Directory domains.
- Samba-3 permits full NT4-style interdomain trusts.
- Samba has security modes that permit more flexible authentication than is possible with MS Windows NT4 domain controllers.
- Samba-3 permits use of multiple concurrent account database backends. (Encrypted passwords that are stored in the account database are in formats that are unique to Windows networking).
- The account database backends can be distributed and replicated using multiple methods. This gives Samba-3 greater flexibility than MS Windows NT4 and in many cases a significantly higher utility than Active Directory domains with MS Windows 200x.

3.2 Server Types

Administrators of Microsoft networks often refer to three different types of servers:

- Domain Controller
  - Primary Domain Controller (PDC)
  - Backup Domain Controller (BDC)
  - ADS Domain Controller
- Domain Member Server
  - Active Directory Domain Server

  - NT4 Style Domain Domain Server

- Standalone Server

The chapters covering domain control (Chapter 4, "Domain Control"), backup domain control (Chapter 5, "Backup Domain Control"), and domain membership (Chapter 6, "Domain Membership") provide pertinent information regarding Samba configuration for each of these server roles. You are strongly encouraged to become intimately familiar with these chapters because they lay the foundation for deployment of Samba domain security.

A Standalone server is autonomous in respect of the source of its account backend. Refer to Chapter 7, "Standalone Servers" to gain a wider appreciation of what is meant by a server being configured as a *standalone* server.

3.3 Samba Security Modes

In this section, the function and purpose of Samba's security modes are described. An accurate understanding of how Samba implements each security mode as well as how to configure MS Windows clients for each mode will significantly reduce user complaints and administrator heartache.

Microsoft Windows networking uses a protocol that was originally called the Server Message Block (SMB) protocol. Since some time around 1996 the protocol has been better known as the Common Internet Filesystem (CIFS) protocol.

In the SMB/CIFS networking world, there are only two types of security: *user-level* and *share level*. We refer to these collectively as *security levels*. In implementing these two security levels, Samba provides flexibilities that are not available with MS Windows NT4/200x servers. In fact, Samba implements *share-level* security only one way, but has four ways of implementing *user-level* security. Collectively, we call the Samba implementations of the security levels *security modes*. They are known as *share*, *user*, *domain*, *ADS*, and *server* modes. They are documented in this chapter.

An SMB server informs the client, at the time of a session setup, the security level the server is running. There are two options: share-level and user-level. Which of these two the client receives affects the way the client then tries to authenticate itself. It does not directly affect (to any great extent) the way the Samba server does security. This may sound strange, but it fits

in with the client/server approach of SMB. In SMB everything is initiated and controlled by the client, and the server can only tell the client what is available and whether an action is allowed.

The term `client` refers to all agents whether it is a Windows workstation, a Windows server, another Samba server, or any vanilla SMB or CIFS client application (e.g., **smbclient**) that make use of services provided by an SMB/CIFS server.

3.3.1 User Level Security

We describe user-level security first because its simpler. In user-level security, the client sends a session setup request directly following protocol negotiation. This request provides a username and password. The server can either accept or reject that username/password combination. At this stage the server has no idea what share the client will eventually try to connect to, so it can't base the *accept/reject* on anything other than:

1. the username/password.
2. the name of the client machine.

If the server accepts the username/password credentials, the client expects to be able to mount shares (using a *tree connection*) without further specifying a password. It expects that all access rights will be as the username/password credentials set that was specified in the initial *session setup*.

It is also possible for a client to send multiple *session setup* requests. When the server responds, it gives the client a *uid* to use as an authentication tag for that username/password. The client can maintain multiple authentication contexts in this way (WinDD is an example of an application that does this).

Windows networking user account names are case-insensitive, meaning that upper-case and lower-case characters in the account name are considered equivalent. They are said to be case-preserving, but not case significant. Windows and LanManager systems previous to Windows NT version 3.10 have case-insensitive passwords that were not necessarilty case-preserving. All Windows NT family systems treat passwords as case-preserving and case-sensitive.

3.3.1.1 Example Configuration

The `smb.conf` parameter that sets user-level security is:

```
        security = user
```

This is the default setting since Samba-2.2.x.

3.3.2 Share-Level Security

In share-level security, the client authenticates itself separately for each share. It sends a password along with each tree connection request (share mount), but it does not explicitly send a username with this operation. The client expects a password to be associated with each share, independent of the user. This means that Samba has to work out what username the client probably wants to use, the SMB server is not explicitly sent the username. Some commercial SMB servers such as NT actually associate passwords directly with shares in share-level security, but Samba always uses the UNIX authentication scheme where it is a username/password pair that is authenticated, not a share/password pair.

To understand the MS Windows networking parallels, think in terms of MS Windows 9x/Me where you can create a shared folder that provides read-only or full access, with or without a password.

Many clients send a session setup request even if the server is in share-level security. They normally send a valid username but no password. Samba records this username in a list of possible usernames. When the client then issues a tree connection request, it also adds to this list the name of the share they try to connect to (useful for home directories) and any users listed in the *user* parameter in the `smb.conf` file. The password is then checked in turn against these possible usernames. If a match is found, then the client is authenticated as that user.

Where the list of possible user names is not provided, Samba makes a UNIX system call to find the user account that has a password that matches the one provided from the standard account database. On a system that has no name service switch (NSS) facility, such lookups will be from the `/etc/passwd` database. On NSS enabled systems, the lookup will go to the libraries that have been specified in the `nsswitch.conf` file. The entries in that file in which the libraries are specified are:

```
passwd: files nis ldap
shadow: files nis ldap
group: files nis ldap
```

In the example shown here (not likely to be used in practice) the lookup will check `/etc/passwd` and `/etc/group`, if not found it will check NIS, then LDAP.

3.3.2.1 Example Configuration

The `smb.conf` parameter that sets share-level security is:

```
        security = share
```

3.3.3 Domain Security Mode (User-Level Security)

Domain security provides a mechanism for storing all user and group accounts in a central, shared, account repository. The centralized account repository is shared between domain (security) controllers. Servers that act as domain controllers provide authentication and validation services to all machines that participate in the security context for the domain. A primary domain controller (PDC) is a server that is responsible for maintaining the integrity of the security account database. Backup domain controllers (BDCs) provide only domain logon and authentication services. Usually, BDCs will answer network logon requests more responsively than will a PDC.

When Samba is operating in *security* = domain mode, the Samba server has a domain security trust account (a machine account) and causes all authentication requests to be passed through to the domain controllers. In other words, this configuration makes the Samba server a domain member server, even when it is in fact acting as a domain controller. All machines that participate in domain security must have a machine account in the security database.

Within the domain security environment, the underlying security architecture uses user-level security. Even machines that are domain members must authenticate on startup. The machine account consists of an account entry

in the accounts database, the name of which is the NetBIOS name of the machine and of which the password is randomly generated and known to both the domain controllers and the member machine. If the machine account cannot be validated during startup, users will not be able to log on to the domain using this machine because it cannot be trusted. The machine account is referred to as a machine trust account.

There are three possible domain member configurations:

1. Primary domain controller (PDC) - of which there is one per domain.
2. Backup domain controller (BDC) - of which there can be any number per domain.
3. Domain member server (DMS) - of which there can be any number per domain.

We will discuss each of these in separate chapters. For now, we are most interested in basic DMS configuration.

3.3.3.1 Example Configuration

Samba as a Domain Member Server

This method involves addition of the following parameters in the `smb.conf` file:

```
        security = domain
        workgroup = MIDEARTH
```

In order for this method to work, the Samba server needs to join the MS Windows NT security domain. This is done as follows:

1. On the MS Windows NT domain controller, using the Server Manager, add a machine account for the Samba server.
2. On the UNIX/Linux system execute:

   `root# net rpc join -U administrator%password`

NOTE

Samba-2.2.4 and later Samba 2.2.x series releases can autojoin a Windows NT4-style domain just by executing:

```
root# smbpasswd -j DOMAIN_NAME -r PDC_NAME \
    -U Administrator%password
```

Samba-3 can do the same by executing:

```
root# net rpc join -U Administrator%password
```

It is not necessary with Samba-3 to specify the *DOMAIN_NAME* or the *PDC_NAME*, as it figures this out from the `smb.conf` file settings.

Use of this mode of authentication requires there to be a standard UNIX account for each user in order to assign a UID once the account has been authenticated by the Windows domain controller. This account can be blocked to prevent logons by clients other than MS Windows through means such as setting an invalid shell in the `/etc/passwd` entry. The best way to allocate an invalid shell to a user account is to set the shell to the file `/bin/false`.

Domain controllers can be located anywhere that is convenient. The best advice is to have a BDC on every physical network segment, and if the PDC is on a remote network segment the use of WINS (see Chapter 10, "Network Browsing" for more information) is almost essential.

An alternative to assigning UIDs to Windows users on a Samba member server is presented in Chapter 24, "Winbind: Use of Domain Accounts", Chapter 24, "Winbind: Use of Domain Accounts".

For more information regarding domain membership, Chapter 6, "Domain Membership".

3.3.4 ADS Security Mode (User-Level Security)

Both Samba-2.2, and Samba-3 can join an Active Directory domain using NT4 style RPC based security. This is possible if the domain is run in native mode. Active Directory in native mode perfectly allows NT4-style domain members. This is contrary to popular belief.

If you are using Active Directory, starting with Samba-3 you can join as a native AD member. Why would you want to do that? Your security policy might prohibit the use of NT-compatible authentication protocols. All your machines are running Windows 2000 and above and all use Kerberos. In this case, Samba, as an NT4-style domain, would still require NT-compatible authentication data. Samba in AD-member mode can accept Kerberos tickets.

Sites that use Microsoft Windows active directory services (ADS) should be aware of the significance of the terms: `native mode` and `mixed mode` ADS operation. The term `realm` is used to describe a Kerberos-based security architecture (such as is used by Microsoft ADS).

3.3.4.1 Example Configuration

```
        realm = your.kerberos.REALM
        security = ADS
```

The following parameter may be required:

```
        password server = your.kerberos.server
```

Please refer to Chapter 6, “Domain Membership”, and Section 6.4 for more information regarding this configuration option.

3.3.5 Server Security (User Level Security)

Server security mode is left over from the time when Samba was not capable of acting as a domain member server. It is highly recommended not to use this feature. Server security mode has many drawbacks that include:

- Potential account lockout on MS Windows NT4/200x password servers.
- Lack of assurance that the password server is the one specified.

- Does not work with Winbind, which is particularly needed when storing profiles remotely.

- This mode may open connections to the password server and keep them open for extended periods.

- Security on the Samba server breaks badly when the remote password server suddenly shuts down.

- With this mode there is NO security account in the domain that the password server belongs to for the Samba server.

In server security mode the Samba server reports to the client that it is in user-level security. The client then does a session setup as described earlier. The Samba server takes the username/password that the client sends and attempts to log into the *password server* by sending exactly the same username/password that it got from the client. If that server is in user-level security and accepts the password, then Samba accepts the client's connection. This parameter allows the Samba server to use another SMB server as the *password server*.

You should also note that at the start of all this, when the server tells the client what security level it is in, it also tells the client if it supports encryption. If it does, it supplies the client with a random cryptkey. The client will then send all passwords in encrypted form. Samba supports this type of encryption by default.

The parameter *security* = server means that Samba reports to clients that it is running in *user mode* but actually passes off all authentication requests to another user mode server. This requires an additional parameter *password server* that points to the real authentication server. The real authentication server can be another Samba server, or it can be a Windows NT server, the latter being natively capable of encrypted password support.

NOTE

When Samba is running in *server security mode*, it is essential that the parameter *password server* is set to the precise NetBIOS machine name of the target authentication server. Samba cannot determine this from NetBIOS name lookups because the choice of the target authentication server is arbitrary and cannot be determined from a domain name. In essence, a Samba server that is in *server security mode* is operating in what used to be known as workgroup mode.

3.3.5.1 Example Configuration

Using MS Windows NT as an Authentication Server

This method involves the additions of the following parameters in the `smb.conf` file:

```
encrypt passwords = Yes
security = server
password server = "NetBIOS_name_of_a_DC"
```

There are two ways of identifying whether or not a username and password pair is valid. One uses the reply information provided as part of the authentication messaging process, the other uses just an error code.

The downside of this mode of configuration is that for security reasons Samba will send the password server a bogus username and a bogus password, and if the remote server fails to reject the bogus username and password pair, then an alternative mode of identification or validation is used. Where a site uses password lockout, after a certain number of failed authentication attempts, this will result in user lockouts.

Use of this mode of authentication requires a standard UNIX account for the user. This account can be blocked to prevent logons by non-SMB/CIFS clients.

3.4 Password Checking

MS Windows clients may use encrypted passwords as part of a challenge/response authentication model (a.k.a. NTLMv1 and NTLMv2) or alone, or clear-text strings for simple password-based authentication. It should be realized that with the SMB protocol, the password is passed over the network either in plaintext or encrypted, but not both in the same authentication request.

When encrypted passwords are used, a password that has been entered by the user is encrypted in two ways:

- An MD4 hash of the unicode of the password string. This is known as the NT hash.
- The password is converted to uppercase, and then padded or truncated to 14 bytes. This string is then appended with 5 bytes of NULL characters and split to form two 56-bit DES keys to encrypt a "magic" 8-byte value. The resulting 16 bytes form the LanMan hash.

MS Windows 95 pre-service pack 1 and MS Windows NT versions 3.x and version 4.0 pre-service pack 3 will use either mode of password authentication. All versions of MS Windows that follow these versions no longer support plain-text passwords by default.

MS Windows clients have a habit of dropping network mappings that have been idle for 10 minutes or longer. When the user attempts to use the mapped drive connection that has been dropped, the client re-establishes the connection using a cached copy of the password.

When Microsoft changed the default password mode, support was dropped for caching of the plaintext password. This means that when the registry parameter is changed to re-enable use of plaintext passwords, it appears to work, but when a dropped service connection mapping attempts to revalidate, this will fail if the remote authentication server does not support encrypted passwords. It is definitely not a good idea to re-enable plaintext password support in such clients.

The following parameters can be used to work around the issue of Windows 9x/Me clients uppercasing usernames and passwords before transmitting them to the SMB server when using clear-text authentication:

```
        password level
        username level
```

By default Samba will convert to lowercase the username before attempting to lookup the user in the database of local system accounts. Because UNIX usernames conventionally only contain lowercase characters, the *username-level* parameter is rarely needed.

However, passwords on UNIX systems often make use of mixed-case characters. This means that in order for a user on a Windows 9x/Me client to connect to a Samba server using clear-text authentication, the *password level* must be set to the maximum number of uppercase letters that *could* appear in a password. Note that if the Server OS uses the traditional DES version of crypt(), a *password level* of 8 will result in case-insensitive passwords as seen from Windows users. This will also result in longer login times because Samba has to compute the permutations of the password string and try them one by one until a match is located (or all combinations fail).

The best option to adopt is to enable support for encrypted passwords wherever Samba is used. Most attempts to apply the registry change to re-enable plaintext passwords will eventually lead to user complaints and unhappiness.

3.5 Common Errors

We all make mistakes. It is okay to make mistakes, as long as they are made in the right places and at the right time. A mistake that causes lost productivity is seldom tolerated; however, a mistake made in a developmental test lab is expected.

Here we look at common mistakes and misapprehensions that have been the subject of discussions on the Samba mailing lists. Many of these are avoidable by doing your homework before attempting a Samba implementation. Some are the result of a misunderstanding of the English language, which has many phrases that are potentially vague and may be highly confusing to those for whom English is not their native tongue.

3.5.1 What Makes Samba a Server?

To some, the nature of the Samba security mode is obvious, but entirely wrong all the same. It is assumed that *security* = server means that Samba will act as a server. Not so! This setting means that Samba will *try* to use another SMB server as its source for user authentication alone.

Samba is a server regardless of which security mode is chosen. When Samba is used outside of a domain security context, it is best to leave the security mode at the default setting. By default Samba-3 uses user-mode security.

3.5.2 What Makes Samba a Domain Controller?

The `smb.conf` parameter *security* = domain does not really make Samba behave as a domain controller. This setting means we want Samba to be a domain member. See Chapter 4, "Domain Control" for more information.

3.5.3 What Makes Samba a Domain Member?

Guess! So many others do. But whatever you do, do not think that *security* = user makes Samba act as a domain member. Read the manufacturer's manual before the warranty expires. See Chapter 6, "Domain Membership", for more information.

3.5.4 Constantly Losing Connections to Password Server

"Why does server_validate() simply give up rather than re-establish its connection to the password server? Though I am not fluent in the SMB protocol, perhaps the cluster server process passes along to its client workstation the session key it receives from the password server, which means the password hashes submitted by the client would not work on a subsequent connection whose session key would be different. So server_validate() must give up."

Indeed. That's why *security* = server is at best a nasty hack. Please use *security* = domain; *security* = server mode is also known as pass-through authentication.

3.5.5 Stand-alone Server is converted to Domain Controller — Now User accounts don't work

"When I try to log in to the DOMAIN, the eventlog shows *tried credentials DOMAIN/username; effective credentials SERVER/username*"

Usually this is due to a user or machine account being created before the Samba server is configured to be a domain controller. Accounts created before the server becomes a domain controller will be *local* accounts and authenticated as what looks like a member in the SERVER domain, much like local user accounts in Windows 2000 and later. Accounts created after the Samba server becomes a domain controller will be *domain* accounts and will be authenticated as a member of the DOMAIN domain.

This can be verified by issuing the command **pdbedit -L -v username**. If this reports DOMAIN then the account is a domain account, if it reports SERVER then the account is a local account.

The easiest way to resolve this is to remove and recreate the account; however this may cause problems with established user profiles. You can also use **pdbedit -u username -I DOMAIN**. You may also need to change the User SID and Primary Group SID to match the domain.

Chapter 4

DOMAIN CONTROL

There are many who approach MS Windows networking with incredible misconceptions. That's okay, because it gives the rest of us plenty of opportunity to be of assistance. Those who really want help are well advised to become familiar with information that is already available.

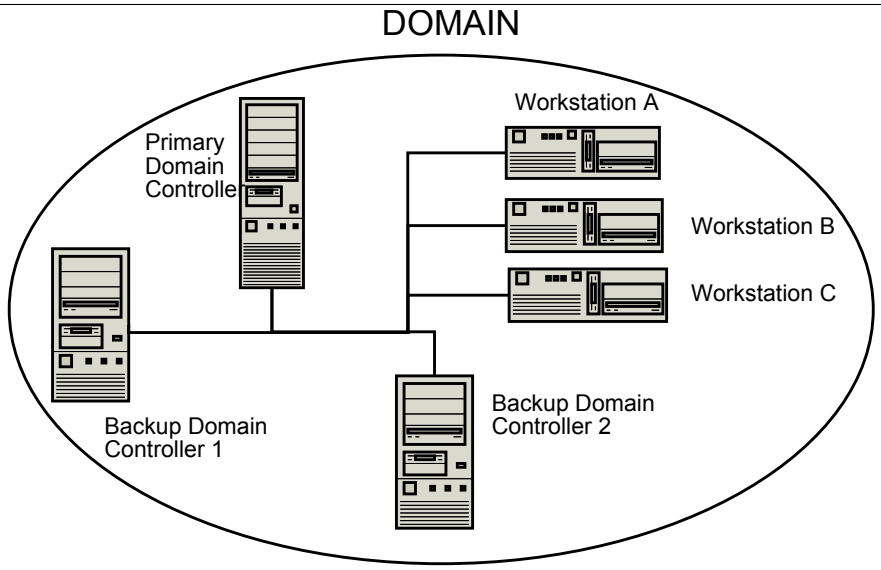
You are advised not to tackle this section without having first understood and mastered some basics. MS Windows networking is not particularly forgiving of misconfiguration. Users of MS Windows networking are likely to complain of persistent niggles that may be caused by a broken network configuration. To a great many people, however, MS Windows networking starts with a domain controller that in some magical way is expected to solve all network operational ills.

Figure 4.1 shows a typical MS Windows domain security network environment. Workstations A, B, and C are representative of many physical MS Windows network clients.

From the Samba mailing list we can readily identify many common networking issues. If you are not clear on the following subjects, then it will do much good to read the sections of this HOWTO that deal with it. These are the most common causes of MS Windows networking problems:

- Basic TCP/IP configuration.
- NetBIOS name resolution.
- Authentication configuration.
- User and group configuration.
- Basic file and directory permission control in UNIX/Linux.

Figure 4.1 An Example Domain.



- Understanding how MS Windows clients interoperate in a network environment.

Do not be put off; on the surface of it MS Windows networking seems so simple that anyone can do it. In fact, it is not a good idea to set up an MS Windows network with inadequate training and preparation. But let's get our first indelible principle out of the way: *It is perfectly okay to make mistakes!* In the right place and at the right time, mistakes are the essence of learning. It is very much not okay to make mistakes that cause loss of productivity and impose an avoidable financial burden on an organization.

Where is the right place to make mistakes? Only out of harms way. If you are going to make mistakes, then please do it on a test network, away from users, and in such a way as to not inflict pain on others. Do your learning on a test network.

4.1 Features and Benefits

What is the key benefit of Microsoft Domain Security?

In a word, *single sign-on*, or SSO for short. To many, this is the Holy Grail of MS Windows NT and beyond networking. SSO allows users in

a well-designed network to log onto any workstation that is a member of the domain that contains their user account (or in a domain that has an appropriate trust relationship with the domain they are visiting) and they will be able to log onto the network and access resources (shares, files, and printers) as if they are sitting at their home (personal) workstation. This is a feature of the domain security protocols.

The benefits of domain security are available to those sites that deploy a Samba PDC. A domain provides a unique network security identifier (SID). Domain user and group security identifiers are comprised of the network SID plus a relative identifier (RID) that is unique to the account. User and group SIDs (the network SID plus the RID) can be used to create access control lists (ACLs) attached to network resources to provide organizational access control. UNIX systems recognize only local security identifiers.

A SID represents a security context. For example, every Windows machine has local accounts within the security context of the local machine which has a unique SID. Every domain (NT4, ADS, Samba) contains accounts that exist within the domain security context which is defined by the domain SID.

A domain member server will have a SID that differs from the domain SID. The domain member server can be configured to regard all domain users as local users. It can also be configured to recognize domain users and groups as non-local. SIDs are persistent. A typical domain of user SID looks like this:

```
S-1-5-21-726309263-4128913605-1168186429
```

Every account (user, group, machine, trust, etc.) is assigned a RID. This is done automatically as an account is created. Samba produces the RID algorithmically. The UNIX operating system uses a separate name space for user and group identifiers (the UID and GID) but Windows allocates the RID from a single name space. A Windows user and a Windows group can not have the same RID. Just as the UNIX user `root` has the UID=0, the Windows Administrator has the well-known RID=500. The RID is catenated to the Windows domain SID, so Administrator account for a domain that has the above SID will have the user SID

```
S-1-5-21-726309263-4128913605-1168186429-500
```

The result is that every account in the Windows networking world has a globally unique security identifier.

NOTE

Network clients of an MS Windows domain security environment must be domain members to be able to gain access to the advanced features provided. Domain membership involves more than just setting the workgroup name to the domain name. It requires the creation of a domain trust account for the workstation (called a machine account). Refer to Chapter 6, "Domain Membership" for more information.

The following functionalities are new to the Samba-3 release:

- Samba-3 supports the use of a choice of backends that may be used in which user, group and machine accounts may be stored. Multiple passwd backends can be used in combination, either as additive backend data sets, or as fail-over data sets.

  An LDAP passdb backend confers the benefit that the account backend can be distributed and replicated, which is of great value because it confers scalability and provides a high degree of reliability.

- Windows NT4 domain trusts. Samba-3 supports workstation and server (machine) trust accounts. It also supports Windows NT4 style interdomain trust accounts, which further assists in network scalability and interoperability.

- Operation without NetBIOS over TCP/IP, rather using the raw SMB over TCP/IP. Note, this is feasible only when operating as a Microsoft active directory domain member server. When acting as a Samba domain controller the use of NetBIOS is necessary to provide network browsing support.

- Samba-3 provides NetBIOS name services (WINS), NetBIOS over TCP/IP (TCP port 139) session services, SMB over TCP/IP (TCP port 445)

session services, and Microsoft compatible ONC DCE RPC services (TCP port 135) services.

- Management of users and groups via the User Manager for Domains. This can be done on any MS Windows client using the `Nexus.exe` toolkit for Windows 9x/Me, or using the SRVTOOLS.EXE package for MS Windows NT4/200x/XP platforms. These packages are available from Microsoft's Web site.

- Implements full Unicode support. This simplifies cross-locale internationalization support. It also opens up the use of protocols that Samba-2.2.x had but could not use due to the need to fully support Unicode.

The following functionalities are not provided by Samba-3:

- SAM replication with Windows NT4 domain controllers (i.e., a Samba PDC and a Windows NT BDC, or vice versa). This means Samba cannot operate as a BDC when the PDC is Microsoft-based Windows NT PDC. Samba-3 can not participate in replication of account data to Windows PDCs and BDCs.

- Acting as a Windows 2000 active directory domain controller (i.e., Kerberos and Active Directory). In point of fact, Samba-3 does have some Active Directory domain control ability that is at this time purely experimental. Active directory domain control is one of the features that is being developed in Samba-4, the next generation Samba release. At this time there are no plans to enable active directory domain control support during the Samba-3 series life-cycle.

- The Windows 200x/XP Microsoft Management Console (MMC) cannot be used to manage a Samba-3 server. For this you can use only the MS Windows NT4 Domain Server Manager and the MS Windows NT4 Domain User Manager. Both are part of the SVRTOOLS.EXE package mentioned later.

Windows 9x/Me/XP Home clients are not true members of a domain for reasons outlined in this chapter. The protocol for support of Windows 9x/Me-style network (domain) logons is completely different from NT4/Windows 200x-type domain logons and has been officially supported for some time. These clients use the old LanMan network logon facilities that are supported in Samba since approximately the Samba-1.9.15 series.

Samba-3 implements group mapping between Windows NT groups and UNIX groups (this is really quite complicated to explain in a short space). This is discussed more fully in Chapter 12, "Group Mapping: MS Windows and UNIX".

Samba-3, like an MS Windows NT4 PDC or a Windows 200x Active Directory, needs to store user and Machine Trust Account information in a suitable backend data-store. Refer to Section 6.2. With Samba-3 there can be multiple backends for this. A complete discussion of account database backends can be found in Chapter 11, "Account Information Databases".

4.2 Single Sign-On and Domain Security

When network administrators are asked to describe the benefits of Windows NT4 and active directory networking the most often mentioned feature is that of single sign-on (SSO). Many companies have implemented SSO solutions. The mode of implementation of a single sign-on solution is an important factor in the practice of networking in general, and is critical in respect of Windows networking. A company may have a wide variety of information systems, each of which requires a form of user authentication and validation, thus it is not uncommon that users may need to remember more than ten login IDs and passwords. This problem is compounded when the password for each system must be changed at regular intervals, and particularly so where password uniqueness and history limits are applied.

There is a broadly held perception that SSO is the answer to the problem of users having to deal with too many information system access credentials (username/password pairs). Many elaborate schemes have been devised to make it possible to deliver a user-friendly SSO solution. The trouble is that if this implementation is not done correctly, the site may end up paying dearly by way of complexity and management overheads. Simply put, many SSO solutions are an administrative nightmare.

SSO implementations utilize centralization of all user account information. Depending on environmental complexity and the age of the systems over which a SSO solution is implemented, it may not be possible to change the solution architecture so as to accomodate a new identity management and user authentication system. Many SSO solutions involving legacy systems consist of a new super-structure that handles authentication on behalf of the user. The software that gets layered over the old system may simply

implement a proxy authentication system. This means that the addition of SSO increases over-all information systems complexity. Ideally, the implementation of SSO should reduce complexity and reduce administative overheads.

The initial goal of many network administrators is often to create and use a centralized identity management system. It is often assumed that such a centralized system will use a single authentication infrastructure that can be used by all information systems. The Microsoft Windows NT4 security domain architecture and the Micrsoft active directory service are often put forward as the ideal foundation for such a system. It is conceptually simple to install an external authentication agent on each of the disparate infromation systems that can then use the Microsoft (NT4 domain or ads service) for user authentication and access control. The wonderful dream of a single centralized authentication service is commonly broken when realities are realized. The problem with legacy systems is often the inability to externalize the authentication and access control system it uses because its implementation will be excessively invasive from a re-engineering perspective, or because application software has built-in dependencies on particular elements of the way user authentication and access control were designed and built.

Over the past decade an industry has been developed around the various methods that have been built to get around the key limitations of legacy information technology systems. One approach that is often used involves the use of a meta-directory. The meta-directory stores user credentials for all disparate information systems in the format that is particular to each system. An elaborate set of management procedures is coupled with a rigidly enforced work-flow protocol for managing user rights and privileges within the maze of systems that are provisioned by the new infrastructure makes possible user access to all systems using a single set of user credentials.

The Organization for the Advancement of Structured Information Standards (OASIS) has developed the Security Assertion Markup Language (SAML), a structured method for communication of authentication information. The over-all umbrella name for the technologies and methods that deploy SAML is called Federated Identity Management (FIM). FIM depends on each system in the complex maze of disparate information systems to authenticate their respective users and vouch for secure access to the services each provides.

SAML documents can be wrapped in a Simple Object Access Protocol

(SOAP) message for the computer-to-computer communications needed for Web services. Or they may be passed between Web servers of federated organizations that share live services. The Liberty Alliance, an industry group formed to promote federated-identity standards, has adopted SAML 1.1 as part of its application framework. Microsoft and IBM have proposed an alternative specification called WS-Security. Some believe that the competing technologies and methods may converge when the SAML 2.0 standard is introduced. A few Web access-management products support SAML today, but implemention of the technology mostly requires customization to integrate applications and develop user interfaces. In a nust-shell, that is why FIM is a big and growing industry.

Ignoring the bigger picture, which is beyond the scope of this book, the migration of all user and group management to a centralized system is a step in the right direction. It is essential for interoperability reasons to locate the identity management system data in a directory such as Microsoft Active Directory Service (ADS), or any proprietary or open source system that provides a standard protocol for information access (such as LDAP) and that can be coupled with a flexible array of authentication mechanisms (such as kerberos) that use the protocols that are defined by the various general security service application programming interface (GSSAPI) services.

A growing number of companies provide authentication agents for disparate legacy platforms to permit the use of LDAP systems. Thus the use of OpenLDAP, the dominant open source software implementation of the light weight directory access protocol standard. This fact, means that by providing support in Samba for the use of LDAP and Microsoft ADS make Samba a highly scalable and forward reaching organizational networking technology.

Microsoft ADS provides purely proprietary services that, with limitation, can be extended to provide a centralized authentication infrastructure. Samba plus LDAP provides a similar opportunity for extension of a centralized authentication architecture, but it is the fact that the Samba Team are proactive in introducing the extension of authentication services, using LDAP or otherwise, to applications such as SQUID (the open source proxy server) through tools such as the **ntlm_auth** utility, that does much to create sustainable choice and competition in the FIM market place.

Primary domain control, if it is to be scalable to meet the needs of large sites, must therefore be capable of using LDAP. The rapid adoption of OpenLDAP, and Samba configurations that use it, is ample proof that the era of the directory has started. Samba-3 does not demand the use of LDAP, but the

demand for a mechanism by which user and group identity information can be distributed makes it an an unavoidable option.

At this time, the use of Samba based BDCs, necessitates the use of LDAP. The most commonly used LDAP implementation used by Samba sites is OpenLDAP. It is possible to use any standards compliant LDAP server. Those known to work includes those manufactured by: IBM, CA, Novell (e-Directory), and others.

4.3 Basics of Domain Control

Over the years, public perceptions of what domain control really is has taken on an almost mystical nature. Before we branch into a brief overview of domain control, there are three basic types of domain controllers.

4.3.1 Domain Controller Types

- NT4 style Primary Domain Controller
- NT4 style Backup Domain Controller
- ADS Domain Controller

The *Primary Domain Controller* or PDC plays an important role in MS Windows NT4. In Windows 200x domain control architecture, this role is held by domain controllers. Folklore dictates that because of its role in the MS Windows network, the domain controller should be the most powerful and most capable machine in the network. As strange as it may seem to say this here, good overall network performance dictates that the entire infrastructure needs to be balanced. It is advisable to invest more in standalone (domain member) servers than in the domain controllers.

In the case of MS Windows NT4-style domains, it is the PDC that initiates a new domain control database. This forms a part of the Windows registry called the Security Account Manager (SAM). It plays a key part in NT4-type domain user authentication and in synchronization of the domain authentication database with BDCs.

With MS Windows 200x Server-based Active Directory domains, one domain controller initiates a potential hierarchy of domain controllers, each with its own area of delegated control. The master domain controller has the ability

to override any downstream controller, but a downline controller has control only over its downline. With Samba-3, this functionality can be implemented using an LDAP-based user and machine account backend.

New to Samba-3 is the ability to use a backend database that holds the same type of data as the NT4-style SAM database (one of the registry files)[1]

The *Backup Domain Controller* or BDC plays a key role in servicing network authentication requests. The BDC is biased to answer logon requests in preference to the PDC. On a network segment that has a BDC and a PDC, the BDC will most likely service network logon requests. The PDC will answer network logon requests when the BDC is too busy (high load). When a user logs onto a Windows domain member client the workstation will query the network to locate the nearest network logon server. Where a WINS server is used, this is done via a query to the WINS server. If a netlogon server can not be found from the WINS query, or in the absence of a WINS server, the workstation will perform a NetBIOS name lookup via a mailslot broadcast over the UDP broadcast protocol. This means that the netlogon server that the windows client will use is influenced by a number of variables, thus there is no simple determinant of whether a PDC or a BDC will serve a particular logon authentication request.

A Windows NT4 BDC can be promoted to a PDC. If the PDC is online at the time that a BDC is promoted to PDC, the previous PDC is automatically demoted to a BDC. With Samba-3, this is not an automatic operation; the PDC and BDC must be manually configured, and other appropriate changes also need to be made.

With MS Windows NT4, a decision is made at installation to determine what type of machine the server will be. It is possible to promote a BDC to a PDC, and vice versa. The only method Microsoft provide to convert a Windows NT4 domain controller to a domain member server or a standalone server is to reinstall it. The install time choices offered are:

- *Primary Domain Controller* — the one that seeds the domain SAM.
- *Backup Domain Controller* — one that obtains a copy of the domain SAM.
- *Domain Member Server* — one that has no copy of the domain SAM; rather it obtains authentication from a domain controller for all access controls.

[1] See also Chapter 11, "Account Information Databases". .

- *Standalone Server* — one that plays no part in SAM synchronization, has its own authentication database, and plays no role in domain security.

NOTE

Algin Technology LLC provide a commercial tool that makes it possible to promote a Windows NT4 standalone server to a PDC or a BDC, and also permits this process to be reversed. Refer to the Algin[a] web site for further information.

[a] <http://utools.com/UPromote.asp>

Samba-3 servers can readily be converted to and from domain controller roles through simple changes to the `smb.conf` file. Samba-3 is capable of acting fully as a native member of a Windows 200x server Active Directory domain.

For the sake of providing a complete picture, MS Windows 2000 domain control configuration is done after the server has been installed. Please refer to Microsoft documentation for the procedures that should be followed to convert a domain member server to or from a domain control, and to install or remove active directory service support.

New to Samba-3 is the ability to function fully as an MS Windows NT4-style domain controller, excluding the SAM replication components. However, please be aware that Samba-3 also supports the MS Windows 200x domain control protocols.

At this time any appearance that Samba-3 is capable of acting as a *domain controller* in native ADS mode is limited and experimental in nature. This functionality should not be used until the Samba Team offers formal support for it. At such a time, the documentation will be revised to duly reflect all configuration and management requirements. Samba can act as a NT4-style domain controller in a Windows 2000/XP environment. However, there are certain compromises:

- No machine policy files.

- No Group Policy Objects.
- No synchronously executed Active Directory logon scripts.
- Can't use Active Directory management tools to manage users and machines.
- Registry changes tattoo the main registry, while with Active Directory they do not leave permanent changes in effect.
- Without Active Directory you cannot perform the function of exporting specific applications to specific users or groups.

4.3.2 Preparing for Domain Control

There are two ways that MS Windows machines may interact with each other, with other servers, and with domain controllers: either as *standalone* systems, more commonly called *workgroup* members, or as full participants in a security system, more commonly called *domain* members.

It should be noted that workgroup membership involves no special configuration other than the machine being configured so the network configuration has a commonly used name for its workgroup entry. It is not uncommon for the name WORKGROUP to be used for this. With this mode of configuration, there are no Machine Trust Accounts, and any concept of membership as such is limited to the fact that all machines appear in the network neighborhood to be logically grouped together. Again, just to be clear: *workgroup mode does not involve security machine accounts.*

Domain member machines have a machine trust account in the domain accounts database. A special procedure must be followed on each machine to effect domain membership. This procedure, which can be done only by the local machine Administrator account, creates the domain machine account (if it does not exist), and then initializes that account. When the client first logs onto the domain, a machine trust account password change will be automatically triggered.

NOTE

When Samba is configured as a domain controller, secure network operation demands that all MS Windows NT4/200x/XP Professional clients should be configured as domain members. If a machine is not made a member of the domain, then it will operate like a workgroup (standalone) machine. Please refer to Chapter 6, "Domain Membership", for information regarding domain membership.

The following are necessary for configuring Samba-3 as an MS Windows NT4-style PDC for MS Windows NT4/200x/XP clients:

- Configuration of basic TCP/IP and MS Windows networking.
- Correct designation of the server role (*security* = user).
- Consistent configuration of name resolution.[2]
- Domain logons for Windows NT4/200x/XP Professional clients.
- Configuration of roaming profiles or explicit configuration to force local profile usage.
- Configuration of network/system policies.
- Adding and managing domain user accounts.
- Configuring MS Windows NT4/2000 Professional and Windows XP Professional client machines to become domain members.

The following provisions are required to serve MS Windows 9x/Me clients:

- Configuration of basic TCP/IP and MS Windows networking.
- Correct designation of the server role (*security* = user).
- Network logon configuration (since Windows 9x/Me/XP Home are not technically domain members, they do not really participate in the security aspects of Domain logons as such).

[2]See Chapter 10, "Network Browsing", and Chapter 29, "Integrating MS Windows Networks with Samba".

- Roaming profile configuration.
- Configuration of system policy handling.
- Installation of the network driver "Client for MS Windows Networks" and configuration to log onto the domain.
- Placing Windows 9x/Me clients in user-level security — if it is desired to allow all client-share access to be controlled according to domain user/group identities.
- Adding and managing domain user accounts.

NOTE

Roaming profiles and system/network policies are advanced network administration topics that are covered in Chapter 27, "Desktop Profile Management" and Chapter 26, "System and Account Policies" of this document. However, these are not necessarily specific to a Samba PDC as much as they are related to Windows NT networking concepts.

A domain controller is an SMB/CIFS server that:

- Registers and advertises itself as a domain controller (through NetBIOS broadcasts as well as by way of name registrations either by Mailslot Broadcasts over UDP broadcast, to a WINS server over UDP unicast, or via DNS and Active Directory).
- Provides the NETLOGON service. (This is actually a collection of services that runs over multiple protocols. These include the LanMan logon service, the Netlogon service, the Local Security Account service, and variations of them.)
- Provides a share called NETLOGON.

It is rather easy to configure Samba to provide these. Each Samba domain controller must provide the NETLOGON service that Samba calls the *domain logons* functionality (after the name of the parameter in the `smb.conf`

file). Additionally, one server in a Samba-3 domain must advertise itself as the domain master browser.[3] This causes the PDC to claim a domain-specific NetBIOS name that identifies it as a DMB for its given domain or workgroup. Local master browsers (LMBs) in the same domain or workgroup on broadcast-isolated subnets then ask for a complete copy of the browse list for the whole wide-area network. Browser clients then contact their LMB, and will receive the domain-wide browse list instead of just the list for their broadcast-isolated subnet.

4.4 Domain Control: Example Configuration

The first step in creating a working Samba PDC is to understand the parameters necessary in `smb.conf`. An example `smb.conf` for acting as a PDC can be found in Example 4.4.1.

The basic options shown in Example 4.4.1 are explained as follows:

passdb backend This contains all the user and group account information. Acceptable values for a PDC are: *smbpasswd, tdbsam, and ldapsam.* The "guest" entry provides default accounts and is included by default; there is no need to add it explicitly.

Where use of BDCs is intended, the only logical choice is to use LDAP so the passdb backend can be distributed. The tdbsam and smbpasswd files cannot effectively be distributed and therefore should not be used.

Domain Control Parameters The parameters *os level, preferred master, domain master, security, encrypt passwords*, and *domain logons* play a central role in assuring domain control and network logon support.

The *os level* must be set at or above a value of 32. A domain controller must be the DMB, must be set in *user* mode security, must support Microsoft-compatible encrypted passwords, and must provide the network logon service (domain logons). Encrypted passwords must be enabled. For more details on how to do this, refer to Chapter 11, "Account Information Databases".

[3] See Chapter 10, "Network Browsing".

Example 4.4.1 smb.conf for being a PDC

```
[global]
        netbios name
        workgroup
        passdb backend = tdbsam
        os level = 33
        preferred master = auto
        domain master = yes
        local master = yes
        security = user
        domain logons = yes
        logon path = \\%N\profiles\%U
        logon drive = H:
        logon home = \\homeserver\%U\winprofile
        logon script = logon.cmd
 [netlogon]
        path = /var/lib/samba/netlogon
        read only = yes
        write list
 [profiles]
        path = /var/lib/samba/profiles
        read only = no
        create mask = 0600
        directory mask = 0700
```

Environment Parameters The parameters *logon path, logon home, logon drive*, and *logon script* are environment support settings that help to facilitate client logon operations and that help to provide automated control facilities to ease network management overheads. Please refer to the man page information for these parameters.

NETLOGON Share The NETLOGON share plays a central role in domain logon and domain membership support. This share is provided on all Microsoft domain controllers. It is used to provide logon scripts, to store group policy files (NTConfig.POL), as well as to locate other common tools that may be needed for logon processing. This is an essential share on a domain controller.

PROFILE Share This share is used to store user desktop profiles. Each

user must have a directory at the root of this share. This directory must be write-enabled for the user and must be globally read-enabled. Samba-3 has a VFS module called "fake_permissions" that may be installed on this share. This will allow a Samba administrator to make the directory read-only to everyone. Of course this is useful only after the profile has been properly created.

NOTE

The above parameters make for a full set of functionality that may define the server's mode of operation. The following `smb.conf` parameters are the essentials alone:

```
netbios name = BELERIAND
workgroup = MIDEARTH
domain logons = Yes
domain master = Yes
security = User
```

The additional parameters shown in the longer listing in this section just make for a more complete explanation.

4.5 Samba ADS Domain Control

Samba-3 is not, and cannot act as, an Active Directory server. It cannot truly function as an Active Directory PDC. The protocols for some of the functionality of Active Directory domain controllers has been partially implemented on an experimental only basis. Please do not expect Samba-3 to support these protocols. Do not depend on any such functionality either now or in the future. The Samba Team may remove these experimental features or may change their behavior. This is mentioned for the benefit of those who have discovered secret capabilities in Samba-3 and who have asked when this functionality will be completed. The answer is maybe someday or maybe never!

To be sure, Samba-3 is designed to provide most of the functionality that

Microsoft Windows NT4-style domain controllers have. Samba-3 does not have all the capabilities of Windows NT4, but it does have a number of features that Windows NT4 domain controllers do not have. In short, Samba-3 is not NT4 and it is not Windows Server 200x: it is not an Active Directory server. We hope this is plain and simple enough for all to understand.

4.6 Domain and Network Logon Configuration

The subject of network or domain logons is discussed here because it forms an integral part of the essential functionality that is provided by a domain controller.

4.6.1 Domain Network Logon Service

All domain controllers must run the netlogon service (*domain logons* in Samba). One domain controller must be configured with *domain master* = Yes (the PDC); on all BDCs set the parameter *domain master* = No.

4.6.1.1 Example Configuration

Example 4.6.1 smb.conf for being a PDC

```
[global]
        domain logons = Yes
        domain master = (Yes on PDC, No on BDCs)
 [netlogon]
        comment = Network Logon Service
        path = /var/lib/samba/netlogon
        guest ok = Yes
        browseable = No
```

4.6.1.2 The Special Case of MS Windows XP Home Edition

To be completely clear: If you want MS Windows XP Home Edition to integrate with your MS Windows NT4 or Active Directory domain security, understand it cannot be done. The only option is to purchase the upgrade from MS Windows XP Home Edition to MS Windows XP Professional.

NOTE

MS Windows XP Home Edition does not have the ability to join any type of domain security facility. Unlike MS Windows 9x/Me, MS Windows XP Home Edition also completely lacks the ability to log onto a network.

Now that this has been said, please do not ask the mailing list or email any of the Samba Team members with your questions asking how to make this work. It can't be done. If it can be done, then to do so would violate your software license agreement with Microsoft, and we recommend that you do not do that.

4.6.1.3 The Special Case of Windows 9x/Me

A domain and a workgroup are exactly the same in terms of network browsing. The difference is that a distributable authentication database is associated with a domain, for secure login access to a network. Also, different access rights can be granted to users if they successfully authenticate against a domain logon server. Samba-3 does this now in the same way as MS Windows NT/200x.

The SMB client logging on to a domain has an expectation that every other server in the domain should accept the same authentication information. Network browsing functionality of domains and workgroups is identical and is explained in this documentation under the browsing discussions. It should be noted that browsing is totally orthogonal to logon support.

Issues related to the single-logon network model are discussed in this section. Samba supports domain logons, network logon scripts, and user profiles for MS Windows for Workgroups and MS Windows 9x/Me clients, which are the focus of this section.

When an SMB client in a domain wishes to log on, it broadcasts requests for a logon server. The first one to reply gets the job and validates its password using whatever mechanism the Samba administrator has installed. It is possible (but ill advised) to create a domain where the user database is not shared between servers; that is, they are effectively workgroup servers

advertising themselves as participating in a domain. This demonstrates how authentication is quite different from but closely involved with domains.

Using these features, you can make your clients verify their logon via the Samba server, make clients run a batch file when they log on to the network and download their preferences, desktop, and start menu.

MS Windows XP Home edition is not able to join a domain and does not permit the use of domain logons.

Before launching into the configuration instructions, it is worthwhile to look at how a Windows 9x/Me client performs a logon:

1. The client broadcasts (to the IP broadcast address of the subnet it is in) a NetLogon request. This is sent to the NetBIOS name DOMAIN<1C> at the NetBIOS layer. The client chooses the first response it receives, which contains the NetBIOS name of the logon server to use in the format of `\\SERVER`. The `1C` name is the name type that is registered by domain controllers (SMB/CIFS servers that provide the netlogon service).

2. The client connects to that server, logs on (does an SMBsessetupX) and then connects to the IPC$ share (using an SMBtconX).

3. The client does a NetWkstaUserLogon request, which retrieves the name of the user's logon script.

4. The client then connects to the NetLogon share and searches for said script. If it is found and can be read, it is retrieved and executed by the client. After this, the client disconnects from the NetLogon share.

5. The client sends a NetUserGetInfo request to the server to retrieve the user's home share, which is used to search for profiles. Since the response to the NetUserGetInfo request does not contain much more than the user's home share, profiles for Windows 9x clients must reside in the user home directory.

6. The client connects to the user's home share and searches for the user's profile. As it turns out, you can specify the user's home share as a share name and path. For example, `\\server\fred\.winprofile`. If the profiles are found, they are implemented.

7. The client then disconnects from the user's home share and reconnects to the NetLogon share and looks for `CONFIG.POL`, the policies file. If this is found, it is read and implemented.

The main difference between a PDC and a Windows 9x/Me logon server configuration is:

- Password encryption is not required for a Windows 9x/Me logon server. But note that beginning with MS Windows 98 the default setting is that plaintext password support is disabled. It can be re-enabled with the registry changes that are documented in Chapter 26, "System and Account Policies".
- Windows 9x/Me clients do not require and do not use Machine Trust Accounts.

A Samba PDC will act as a Windows 9x/Me logon server; after all, it does provide the network logon services that MS Windows 9x/Me expect to find.

NOTE

Use of plaintext passwords is strongly discouraged. Where used they are easily detected using a sniffer tool to examine network traffic.

4.6.2 Security Mode and Master Browsers

There are a few comments to make in order to tie up some loose ends. There has been much debate over the issue of whether it is okay to configure Samba as a domain controller that operates with security mode other than user-mode. The only security mode that will not work due to technical reasons is share-mode security. Domain and server mode security are really just a variation on SMB user-level security.

Actually, this issue is also closely tied to the debate on whether Samba must be the DMB for its workgroup when operating as a domain controller. In a pure Microsoft Windows NT domain, the PDC wins the election to be the DMB, and then registers the DOMAIN<1B> NetBIOS name. This is not the name used by Windows clients to locate the domain controller, all domain controllers register the DOMAIN<1C> name and Windows clients locate a network logon server by seraching for the DOMAIN<1C> name. A DMB is a Domain Master Browser — see Chapter 10, "Network Browsing",

Section 10.4.1; Microsoft PDCs expect to win the election to become the DMB, if it loses that election it will report a continuous and rapid sequence of warning messages to its Windows event logger complaining that it has lost the election to become a DMB. For this reason, in networks where a Samba server is the PDC it is wise to configure the Samba domain controller as the DMB.

NOTE

SMB/CIFS servers that register the DOMAIN<1C> name do so because they provide the network logon service. Server that register the DOMAIN<1B> name are DMBs — meaning that they are responsible for browse list synchronization across all machines that have registered the DOMAIN<1D> name. The later are LMBs that have the responsibility to listen to all NetBIOS name registrations that occur locally to their own network segment. The network logon service (NETLOGON) is germane to domain control and has nothing to do with network browsing and browse list management. The 1C and 1B/1D name services are orthogonal to each other.

Now back to the issue of configuring a Samba domain controller to use a mode other than *security* = user. If a Samba host is configured to use another SMB server or domain controller in order to validate user connection requests, it is a fact that some other machine on the network (the *password server*) knows more about the user than the Samba host. About 99 percent of the time, this other host is a domain controller. Now to operate in domain mode security, the *workgroup* parameter must be set to the name of the Windows NT domain (which already has a domain controller). If the domain does not already have a domain controller, you do not yet have a domain.

Configuring a Samba box as a domain controller for a domain that already by definition has a PDC is asking for trouble. Therefore, you should always configure the Samba domain controller to be the DMB for its domain and set *security* = user. This is the only officially supported mode of operation.

4.7 Common Errors

4.7.1 "$" Cannot Be Included in Machine Name

A machine account, typically stored in `/etc/passwd`, takes the form of the machine name with a "$" appended. Some BSD systems will not create a user with a "$" in the name. Recent versions of FreeBSD have removed this limitation, but older releases are still in common use.

The problem is only in the program used to make the entry. Once made, it works perfectly. Create a user without the "$". Then use **vipw** to edit the entry, adding the "$". Or create the whole entry with vipw if you like; make sure you use a unique user login ID.

> NOTE
>
> The machine account must have the exact name that the workstation has.

> NOTE
>
> The UNIX tool **vipw** is a common tool for directly editing the `/etc/passwd` file. The use of vipw will ensure that shadow files (where used) will remain current with the passwd file. This is important for security reasons.

4.7.2 Joining Domain Fails Because of Existing Machine Account

"I get told, 'You already have a connection to the Domain....' or 'Cannot join domain, the credentials supplied conflict with an existing set...' when creating a Machine Trust Account."

This happens if you try to create a Machine Trust Account from the machine itself and already have a connection (e.g., mapped drive) to a share (or IPC$)

on the Samba PDC. The following command will remove all network drive connections:

```
C:\> net use * /d
```

This will break all network connections.

Further, if the machine is already a "member of a workgroup" that is the same name as the domain you are joining (bad idea), you will get this message. Change the workgroup name to something else — it does not matter what — reboot, and try again.

4.7.3 The System Cannot Log You On (C000019B)

"I joined the domain successfully but after upgrading to a newer version of the Samba code I get the message, 'The system cannot log you on (C000019B). Please try again or consult your system administrator when attempting to logon.'"

This occurs when the domain SID stored in the secrets.tdb database is changed. The most common cause of a change in domain SID is when the domain name and/or the server name (NetBIOS name) is changed. The only way to correct the problem is to restore the original domain SID or remove the domain client from the domain and rejoin. The domain SID may be reset using either the net or rpcclient utilities.

To reset or change the domain SID you can use the net command as follows:

```
root# net getlocalsid 'OLDNAME'
root# net setlocalsid 'SID'
```

Workstation Machine Trust Accounts work only with the domain (or network) SID. If this SID changes, domain members (workstations) will not be able to log onto the domain. The original domain SID can be recovered from the secrets.tdb file. The alternative is to visit each workstation to rejoin it to the domain.

4.7.4 The Machine Trust Account Is Not Accessible

"When I try to join the domain I get the message, "The machine account for this computer either does not exist or is not accessible." What's wrong?"

This problem is caused by the PDC not having a suitable Machine Trust Account. If you are using the *add machine script* method to create accounts, then this would indicate that it has not worked. Ensure the domain admin user system is working.

Alternately, if you are creating account entries manually, then they have not been created correctly. Make sure that you have the entry correct for the Machine Trust Account in `smbpasswd` file on the Samba PDC. If you added the account using an editor rather than using the smbpasswd utility, make sure that the account name is the machine NetBIOS name with a "$" appended to it (i.e., computer_name$). There must be an entry in both the POSIX UNIX system account backend as well as in the SambaSAMAccount backend. The default backend for Samba-3 (i.e., the parameter ***passdb backend*** is not specified in the `smb.conf` file, or if specified is set to `smbpasswd`, are respectively the `/etc/passwd` and `/etc/samba/smbpasswd` (or `/usr/local/samba/lib/private/smbpasswd` if compiled using Samba Team default settings). The use of the `/etc/passwd` can be overridden by alternative settings in the NSS `/etc/nsswitch.conf` file.

Some people have also reported that inconsistent subnet masks between the Samba server and the NT client can cause this problem. Make sure that these are consistent for both client and server.

4.7.5 Account Disabled

"When I attempt to log in to a Samba domain from a NT4/W200x workstation, I get a message about my account being disabled."

Enable the user accounts with `smbpasswd -e` ***username***. This is normally done as an account is created.

4.7.6 Domain Controller Unavailable

"Until a few minutes after Samba has started, clients get the error 'Domain Controller Unavailable'"

A domain controller has to announce its role on the network. This usually takes a while. Be patient for up to 15 minutes, then try again.

4.7.7 Cannot Log onto Domain Member Workstation After Joining Domain

After successfully joining the domain, user logons fail with one of two messages: one to the effect that the domain controller cannot be found; the other claims that the account does not exist in the domain or that the password is incorrect. This may be due to incompatible settings between the Windows client and the Samba-3 server for *schannel* (secure channel) settings or *smb signing* settings. Check your Samba settings for *client schannel*, *server schannel*, *client signing*, *server signing* by executing:

```
testparm -v | grep channel and looking for the value of these parameters.
```

Also use the MMC — Local Security Settings. This tool is available from the Control Panel. The Policy settings are found in the Local Policies/Security Options area and are prefixed by *Secure Channel:..., and Digitally sign....*

It is important that these be set consistently with the Samba-3 server settings.

Chapter 5

BACKUP DOMAIN CONTROL

Before you continue reading this section, please make sure that you are comfortable with configuring a Samba domain controller as described in Chapter 4, "Domain Control".

5.1 Features and Benefits

This is one of the most difficult chapters to summarize. It does not matter what we say here, for someone will still draw conclusions and/or approach the Samba Team with expectations that are either not yet capable of being delivered or that can be achieved far more effectively using a totally different approach. In the event that you should have a persistent concern that is not addressed in this book, please email John H. Terpstra[1] clearly setting out your requirements and/or question, and we will do our best to provide a solution.

Samba-3 can act as a Backup Domain Controller (BDC) to another Samba Primary Domain Controller (PDC). A Samba-3 PDC can operate with an LDAP account backend. The LDAP backend can be either a common master LDAP server or a slave server. The use of a slave LDAP server has the benefit that when the master is down, clients may still be able to log onto the network. This effectively gives Samba a high degree of scalability and is an effective solution for large organizations. If you use an LDAP slave server for a PDC, you will need to ensure the master's continued availability — if

[1] <mailto:jht@samba.org>

the slave finds its master down at the wrong time, you will have stability and operational problems.

While it is possible to run a Samba-3 BDC with a non-LDAP backend, that backend must allow some form of "two-way" propagation of changes from the BDC to the master. At this time only LDAP delivers the capability to propagate identity database changes from the BDC to the PDC. The BDC can use a slave LDAP server, while it is preferable for the PDC to use as its primary an LDAP master server.

The use of a non-LDAP backend SAM database is particularly problematic because domain member servers and workstations periodically change the Machine Trust Account password. The new password is then stored only locally. This means that in the absence of a centrally stored accounts database (such as that provided with an LDAP-based solution) if Samba-3 is running as a BDC, the BDC instance of the domain member trust account password will not reach the PDC (master) copy of the SAM. If the PDC SAM is then replicated to BDCs, this results in overwriting the SAM that contains the updated (changed) trust account password with resulting breakage of the domain trust.

Considering the number of comments and questions raised concerning how to configure a BDC, let's consider each possible option and look at the pros and cons for each possible solution. Table 5.1 lists possible design configurations for a PDC/BDC infrastructure.

5.2 Essential Background Information

A domain controller is a machine that is able to answer logon requests from network workstations. Microsoft LanManager and IBM LanServer were two early products that provided this capability. The technology has become known as the LanMan Netlogon service.

When MS Windows NT3.10 was first released, it supported a new style of Domain Control and with it a new form of the network logon service that has extended functionality. This service became known as the NT NetLogon Service. The nature of this service has changed with the evolution of MS Windows NT and today provides a complex array of services that are implemented over an intricate spectrum of technologies.

Table 5.1 Domain Backend Account Distribution Options

PDC Backend	BDC Backend	Notes/Discussion
Master LDAP Server	Slave LDAP Server	The optimal solution that provides high integrity. The SAM will be replicated to a common master LDAP server.
Single Central LDAP Server	Single Central LDAP Server	A workable solution without failover ability. This is a usable solution, but not optimal.
tdbsam	tdbsam + **net rpc vampire**	Does not work with Samba-3.0; Samba does not implement the server-side protocols required.
tdbsam	tdbsam + **rsync**	Do not use this configuration. Does not work because the TDB files are live and data may not have been flushed to disk. Furthermore, this will cause domain trust breakdown.
smbpasswd file	smbpasswd file	Do not use this configuration. Not an elegant solution due to the delays in synchronization and also suffers from the issue of domain trust breakdown.

5.2.1 MS Windows NT4-style Domain Control

Whenever a user logs into a Windows NT4/200x/XP Professional workstation, the workstation connects to a domain controller (authentication server) to validate that the username and password the user entered are valid. If the information entered does not match account information that has been stored in the domain control database (the SAM, or Security Account Manager database), a set of error codes is returned to the workstation that has made the authentication request.

When the username/password pair has been validated, the domain controller (authentication server) will respond with full enumeration of the account information that has been stored regarding that user in the user and machine accounts database for that domain. This information contains a complete network access profile for the user but excludes any information that is particular to the user's desktop profile, or for that matter it excludes all desktop profiles for groups that the user may belong to. It does include

password time limits, password uniqueness controls, network access time limits, account validity information, machine names from which the user may access the network, and much more. All this information was stored in the SAM in all versions of MS Windows NT (3.10, 3.50, 3.51, 4.0).

The account information (user and machine) on domain controllers is stored in two files, one containing the security information and the other the SAM. These are stored in files by the same name in the `%SystemRoot%\System32\config` directory. This normally translates to the path `C:\WinNT\System32\config`. These are the files that are involved in replication of the SAM database where BDCs are present on the network.

There are two situations in which it is desirable to install BDCs:

- On the local network that the PDC is on, if there are many workstations and/or where the PDC is generally very busy. In this case the BDCs will pick up network logon requests and help to add robustness to network services.
- At each remote site, to reduce wide-area network traffic and to add stability to remote network operations. The design of the network, and the strategic placement of BDCs, together with an implementation that localizes as much of network to client interchange as possible, will help to minimize wide-area network bandwidth needs (and thus costs).

The interoperation of a PDC and its BDCs in a true Windows NT4 environment is worth mentioning here. The PDC contains the master copy of the SAM. In the event that an administrator makes a change to the user account database while physically present on the local network that has the PDC, the change will likely be made directly to the PDC instance of the master copy of the SAM. In the event that this update may be performed in a branch office, the change will likely be stored in a delta file on the local BDC. The BDC will then send a trigger to the PDC to commence the process of SAM synchronization. The PDC will then request the delta from the BDC and apply it to the master SAM. The PDC will then contact all the BDCs in the domain and trigger them to obtain the update and then apply that to their own copy of the SAM.

Samba-3 cannot participate in true SAM replication and is therefore not able to employ precisely the same protocols used by MS Windows NT4. A Samba-3 BDC will not create SAM update delta files. It will not interoperate with a PDC (NT4 or Samba) to synchronize the SAM from delta files that are held by BDCs.

Samba-3 cannot function as a BDC to an MS Windows NT4 PDC, and Samba-3 cannot function correctly as a PDC to an MS Windows NT4 BDC. Both Samba-3 and MS Windows NT4 can function as a BDC to its own type of PDC.

The BDC is said to hold a *read-only* of the SAM from which it is able to process network logon requests and authenticate users. The BDC can continue to provide this service, particularly while, for example, the wide-area network link to the PDC is down. A BDC plays a very important role in both the maintenance of domain security as well as in network integrity.

In the event that the NT4 PDC should need to be taken out of service, or if it dies, one of the NT4 BDCs can be promoted to a PDC. If this happens while the original NT4 PDC is online, it is automatically demoted to an NT4 BDC. This is an important aspect of domain controller management. The tool that is used to effect a promotion or a demotion is the Server Manager for Domains. It should be noted that Samba-3 BDCs cannot be promoted in this manner because reconfiguration of Samba requires changes to the `smb.conf` file. It is easy enough to manuall change the `smb.conf` file and then restart relevant Samba network services.

5.2.1.1 Example PDC Configuration

Beginning with Version 2.2, Samba officially supports domain logons for all current Windows clients, including Windows NT4, 2003, and XP Professional. For Samba to be enabled as a PDC, some parameters in the *[global]* section of the `smb.conf` have to be set. Refer to Example 5.2.1 for an example of the minimum required settings.

Several other things like a *[homes]* and a *[netlogon]* share also need to be set along with settings for the profile path, the user's home drive, and so on. This is not covered in this chapter; for more information please refer to Chapter 4, "Domain Control". Refer to Chapter 4, "Domain Control" for specific recommendations for PDC configuration. Alternately, fully documented working example network configurations using OpenLDAP and Samba as available in the book[2] "Samba-3 by Example" that may be obtained from local and on-line book stores.

[2] <http://www.samba.org/samba/docs/Samba3-ByExample>

Example 5.2.1 Minimal smb.conf for a PDC in Use with a BDC — LDAP Server on PDC

```
        workgroup = MIDEARTH
        passdb backend = ldapsam://localhost:389
        domain master = yes
        domain logons = yes
        ldap suffix = dc=quenya,dc=org
        ldap user suffix = ou=Users
        ldap group suffix = ou=Groups
        ldap machine suffix = ou=Computers
        ldap idmap suffix = ou=Idmap
        ldap admin dn = cn=sambadmin,dc=quenya,dc=org
```

5.2.2 LDAP Configuration Notes

When configuring a master and a slave LDAP server, it is advisable to use the master LDAP server for the PDC and slave LDAP servers for the BDCs. It is not essential to use slave LDAP servers; however, many administrators will want to do so in order to provide redundant services. Of course, one or more BDCs may use any slave LDAP server. Then again, it is entirely possible to use a single LDAP server for the entire network.

When configuring a master LDAP server that will have slave LDAP servers, do not forget to configure this in the `/etc/openldap/slapd.conf` file. It must be noted that the DN of a server certificate must use the CN attribute to name the server, and the CN must carry the servers' fully qualified domain name. Additional alias names and wildcards may be present in the subjectAltName certificate extension. More details on server certificate names are in RFC2830.

It does not really fit within the scope of this document, but a working LDAP installation is basic to LDAP-enabled Samba operation. When using an OpenLDAP server with Transport Layer Security (TLS), the machine name in `/etc/ssl/certs/slapd.pem` must be the same as in `/etc/openldap/sldap.conf`. The Red Hat Linux startup script creates the `slapd.pem` file with hostname "localhost.localdomain." It is impossible to access this LDAP server from a slave LDAP server (i.e., a Samba BDC) unless the certificate is re-created with a correct hostname.

Do not install a Samba PDC so that is uses an LDAP slave server. Joining client machines to the domain will fail in this configuration because the

change to the machine account in the LDAP tree must take place on the master LDAP server. This is not replicated rapidly enough to the slave server that the PDC queries. It therefore gives an error message on the client machine about not being able to set up account credentials. The machine account is created on the LDAP server, but the password fields will be empty. Unfortunately, some sites are unable to avoid such configurations, and these sites should review the *ldap replication sleep* parameter, intended to slow down Samba sufficiently for the replication to catch up. This is a kludge, and one that the administrator must manually duplicate in any scripts (such as the *add machine script*) that they use.

Possible PDC/BDC plus LDAP configurations include:

- PDC+BDC -> One Central LDAP Server.
- PDC -> LDAP master server, BDC -> LDAP slave server.
- PDC -> LDAP master, with secondary slave LDAP server.

  BDC -> LDAP master, with secondary slave LDAP server.
- PDC -> LDAP master, with secondary slave LDAP server.

  BDC -> LDAP slave server, with secondary master LDAP server.

In order to have a fallback configuration (secondary) LDAP server, you would specify the secondary LDAP server in the `smb.conf` file as shown in Example 5.2.2.

Example 5.2.2 Multiple LDAP Servers in smb.conf

```
        passdb backend = ldapsam:"ldap://master.quenya. ↩
           org ldap://slave.quenya.org"
```

5.2.3 Active Directory Domain Control

As of the release of MS Windows 2000 and Active Directory, this information is now stored in a directory that can be replicated and for which partial or full administrative control can be delegated. Samba-3 is not able to be a domain controller within an Active Directory tree, and it cannot be an Active Directory server. This means that Samba-3 also cannot act as a BDC to an Active Directory domain controller.

5.2.4 What Qualifies a Domain Controller on the Network?

Every machine that is a domain controller for the domain MIDEARTH has to register the NetBIOS group name MIDEARTH<1C> with the WINS server and/or by broadcast on the local network. The PDC also registers the unique NetBIOS name MIDEARTH<1B> with the WINS server. The name type <1B> name is normally reserved for the Domain Master Browser (DMB), a role that has nothing to do with anything related to authentication, but the Microsoft domain implementation requires the DMB to be on the same machine as the PDC.

Where a WINS server is not used, broadcast name registrations alone must suffice. Refer to Chapter 10, "Network Browsing",Section 10.3 for more information regarding TCP/IP network protocols and how SMB/CIFS names are handled.

5.2.5 How Does a Workstation find its Domain Controller?

There are two different mechanisms to locate a domain controller: one method is used when NetBIOS over TCP/IP is enabled and the other when it has been disabled in the TCP/IP network configuration.

Where NetBIOS over TCP/IP is disabled, all name resolution involves the use of DNS, broadcast messaging over UDP, as well as Active Directory communication technologies. In this type of environment all machines require appropriate DNS entries. More information may be found in Section 10.3.3.

5.2.5.1 NetBIOS Over TCP/IP Enabled

An MS Windows NT4/200x/XP Professional workstation in the domain MIDEARTH that wants a local user to be authenticated has to find the domain controller for MIDEARTH. It does this by doing a NetBIOS name query for the group name MIDEARTH<1C>. It assumes that each of the machines it gets back from the queries is a domain controller and can answer logon requests. To not open security holes, both the workstation and the selected domain controller authenticate each other. After that the workstation sends the user's credentials (name and password) to the local domain controller for validation.

5.2.5.2 NetBIOS Over TCP/IP Disabled

An MS Windows NT4/200x/XP Professional workstation in the realm `quenya.org` that has a need to affect user logon authentication will locate the domain controller by re-querying DNS servers for the `_ldap._tcp.pdc._msdcs.quenya.org` record. More information regarding this subject may be found in Section 10.3.3.

5.3 Backup Domain Controller Configuration

The creation of a BDC requires some steps to prepare the Samba server before smbd is executed for the first time. These steps are as follows:

- The domain SID has to be the same on the PDC and the BDC. In Samba versions pre-2.2.5, the domain SID was stored in the file `private/MACHINE.SID`. For all versions of Samba released since 2.2.5 the domain SID is stored in the file `private/secrets.tdb`. This file is unique to each server and cannot be copied from a PDC to a BDC; the BDC will generate a new SID at startup. It will overwrite the PDC domain SID with the newly created BDC SID. There is a procedure that will allow the BDC to aquire the domain SID. This is described here.

  To retrieve the domain SID from the PDC or an existing BDC and store it in the `secrets.tdb`, execute:

  ```
  root# net rpc getsid
  ```

- Specification of the *ldap admin dn* is obligatory. This also requires the LDAP administration password to be set in the `secrets.tdb` using the **smbpasswd -w** *mysecret*.
- The *ldap suffix* parameter and the *ldap idmap suffix* parameter must be specified in the `smb.conf` file.
- The UNIX user database has to be synchronized from the PDC to the BDC. This means that both the `/etc/passwd` and `/etc/group` have to be replicated from the PDC to the BDC. This can be done manually whenever changes are made. Alternately, the PDC is set up as an NIS master server and the BDC as an NIS slave server. To set up the BDC

as a mere NIS client would not be enough, as the BDC would not be able to access its user database in case of a PDC failure. NIS is by no means the only method to synchronize passwords. An LDAP solution would also work.

- The Samba password database must be replicated from the PDC to the BDC. Although it is possible to synchronize the `smbpasswd` file with **rsync** and **ssh**, this method is broken and flawed, and is therefore not recommended. A better solution is to set up slave LDAP servers for each BDC and a master LDAP server for the PDC. The use of rsync is inherently flawed by the fact that the data will be replicated at timed intervals. There is no guarantee that the BDC will be operating at all times with correct and current machine and user account information. This means that this method runs the risk of users being inconvenienced by discontinuity of access to network services due to inconsistent security data. It must be born in mind that Windows workstations update (change) the machine trust account password at regular intervals — administrators are not normally aware that this is happening or when it takes place.

  The use of LDAP for both the POSIX (UNIX user and group) accounts and for the SambaSAMAccount data automatically ensures that all account change information will be written to the shared directory. This eliminates the need for any special action to synchronize account information because LDAP will meet that requirement.

- The netlogon share has to be replicated from the PDC to the BDC. This can be done manually whenever login scripts are changed, or it can be done automatically using a **cron** job that will replicate the directory structure in this share using a tool like **rsync**. The use of **rsync** for replication of the netlogon data is not critical to network security and is one that can be manually managed given that the administrator will make all changes to the netlogon share as part of a conscious move.

5.3.1 Example Configuration

Finally, the BDC has to be capable of being found by the workstations. This can be done by configuring the Samba `smb.conf` file *[global]* section as shown in Example 5.3.1.

Example 5.3.1 Minimal Setup for Being a BDC

```
        workgroup = MIDEARTH
        passdb backend = ldapsam:ldap://slave-ldap.quenya ↩
            .org
        domain master = no
        domain logons = yes
        ldap suffix = dc=abmas,dc=biz
        ldap user suffix = ou=Users
        ldap group suffix = ou=Groups
        ldap machine suffix = ou=Computers
        ldap idmap suffix = ou=Idmap
        ldap admin dn = cn=sambadmin,dc=quenya,dc=org
        idmap backend = ldap:ldap://master-ldap.quenya. ↩
            org
        idmap uid = 10000-20000
        idmap gid = 10000-20000
```

Fully documented working example network configurations using OpenLDAP and Samba as available in the book[3] "Samba-3 by Example" that may be obtained from local and on-line book stores.

This configuration causes the BDC to register only the name MIDEARTH<1C> with the WINS server. This is not a problem, as the name MIDEARTH<1C> is a NetBIOS group name that is meant to be registered by more than one machine. The parameter *domain master* = no forces the BDC not to register MIDEARTH<1B>, which is a unique NetBIOS name that is reserved for the PDC.

The *idmap backend* will redirect the **winbindd** utility to use the LDAP database to store all mappings for Windows SIDs to UIDs and GIDs for UNIX accounts in a repository that is shared. The BDC will however depend on local resolution of UIDs and GIDs via NSS and the **nss_ldap** utility.

[3] <http://www.samba.org/samba/docs/Samba3-ByExample>

NOTE

Samba-3 has introduced a new ID mapping facility. One of the features of this facility is that it allows greater flexibility in how user and group IDs are handled in respect to NT domain user and group SIDs. One of the new facilities provides for explicitly ensuring that UNIX/Linux UID and GID values will be consistent on the PDC, all BDCs, and all domain member servers. The parameter that controls this is called *idmap backend*. Please refer to the man page for `smb.conf` for more information regarding its behavior.

The use of the *idmap backend* = ldap:ldap://master.quenya.org option on a BDC only makes sense where ldapsam is used on a PDC. The purpose of an LDAP-based idmap backend is also to allow a domain member (without its own passdb backend) to use winbindd to resolve Windows network users and groups to common UID/GIDs. In other words, this option is generally intended for use on BDCs and on domain member servers.

5.4 Common Errors

Domain control was a new area for Samba, but there are now many examples that we may refer to. Updated information will be published as they become available and may be found in later Samba releases or from the Samba Web site[4]; refer in particular to the `WHATSNEW.txt` in the Samba release tarball. The book, "Samba-3 by Example" documents well tested and proven configuration examples. You can obtain a copy of this book[5] for the Samba web site.

5.4.1 Machine Accounts Keep Expiring

This problem will occur when the passdb (SAM) files are copied from a central server but the local BDC is acting as a PDC. This results in the

[4] <http://samba.org>
[5] <http://www.samba.org/samba/docs/Samba3-ByExample.pdf>

application of Local Machine Trust Account password updates to the local SAM. Such updates are not copied back to the central server. The newer machine account password is then overwritten when the SAM is recopied from the PDC. The result is that the domain member machine on startup will find that its passwords do not match the one now in the database, and since the startup security check will now fail, this machine will not allow logon attempts to proceed and the account expiry error will be reported.

The solution is to use a more robust passdb backend, such as the ldapsam backend, setting up a slave LDAP server for each BDC and a master LDAP server for the PDC.

5.4.2 Can Samba Be a Backup Domain Controller to an NT4 PDC?

No. The native NT4 SAM replication protocols have not yet been fully implemented.

Can I get the benefits of a BDC with Samba? Yes, but only to a Samba PDC.The main reason for implementing a BDC is availability. If the PDC is a Samba machine, a second Samba machine can be set up to service logon requests whenever the PDC is down.

5.4.3 How Do I Replicate the smbpasswd File?

Replication of the smbpasswd file is sensitive. It has to be done whenever changes to the SAM are made. Every user's password change is done in the smbpasswd file and has to be replicated to the BDC. So replicating the smbpasswd file very often is necessary.

As the smbpasswd file contains plaintext password equivalents, it must not be sent unencrypted over the wire. The best way to set up smbpasswd replication from the PDC to the BDC is to use the utility rsync. rsync can use ssh as a transport. **ssh** itself can be set up to accept *only* **rsync** transfer without requiring the user to type a password.

As said a few times before, use of this method is broken and flawed. Machine trust accounts will go out of sync, resulting in a broken domain. This method is *not* recommended. Try using LDAP instead.

5.4.4 Can I Do This All with LDAP?

The simple answer is yes. Samba's pdb_ldap code supports binding to a replica LDAP server and will also follow referrals and rebind to the master if it ever needs to make a modification to the database. (Normally BDCs are read-only, so this will not occur often).

Chapter 6

DOMAIN MEMBERSHIP

Domain membership is a subject of vital concern. Samba must be able to participate as a member server in a Microsoft domain security context, and Samba must be capable of providing domain machine member trust accounts; otherwise it would not be able to offer a viable option for many users.

This chapter covers background information pertaining to domain membership, the Samba configuration for it, and MS Windows client procedures for joining a domain. Why is this necessary? Because both are areas in which there exists within the current MS Windows networking world, and particularly in the UNIX/Linux networking and administration world, a considerable level of misinformation, incorrect understanding, and lack of knowledge. Hopefully this chapter will fill the voids.

6.1 Features and Benefits

MS Windows workstations and servers that want to participate in domain security need to be made domain members. Participating in domain security is often called *single sign-on*, or SSO for short. This chapter describes the process that must be followed to make a workstation (or another server — be it an MS Windows NT4/200x server) or a Samba server a member of an MS Windows domain security context.

Samba-3 can join an MS Windows NT4-style domain as a native member server, an MS Windows Active Directory domain as a native member server, or a Samba domain control network. Domain membership has many advantages:

- MS Windows workstation users get the benefit of SSO.
- Domain user access rights and file ownership/access controls can be set from the single Domain Security Account Manager (SAM) database (works with domain member servers as well as with MS Windows workstations that are domain members).
- Only MS Windows NT4/200x/XP Professional workstations that are domain members can use network logon facilities.
- Domain member workstations can be better controlled through the use of policy files (`NTConfig.POL`) and desktop profiles.
- Through the use of logon scripts, users can be given transparent access to network applications that run off application servers.
- Network administrators gain better application and user access management abilities because there is no need to maintain user accounts on any network client or server other than the central domain database (either NT4/Samba SAM-style domain, NT4 domain that is backended with an LDAP directory, or via an Active Directory infrastructure).

6.2 MS Windows Workstation/Server Machine Trust Accounts

A Machine Trust Account is an account that is used to authenticate a client machine (rather than a user) to the domain controller server. In Windows terminology, this is known as a "computer account." The purpose of the machine trust account is to prevent a rogue user and domain controller from colluding to gain access to a domain member workstation.

The password of a Machine Trust Account acts as the shared secret for secure communication with the domain controller. This is a security feature to prevent an unauthorized machine with the same NetBIOS name from joining the domain, participating in domain security operations, and gaining access to domain user/group accounts. Windows NT/200x/XP Professional clients use machine trust accounts, but Windows 9x/Me/XP Home clients do not. Hence, a Windows 9x/Me/XP Home client is never a true member of a domain because it does not possess a Machine Trust Account, and, thus, has no shared secret with the domain controller.

A Windows NT4 PDC stores each Machine Trust Account in the Windows Registry. The introduction of MS Windows 2000 saw the introduction of Active Directory, the new repository for Machine Trust Accounts. A Samba PDC, however, stores each Machine Trust Account in two parts, as follows:

- A domain security account (stored in the *passdb backend*) that has been configured in the `smb.conf` file. The precise nature of the account information that is stored depends on the type of backend database that has been chosen.

  The older format of this data is the `smbpasswd` database that contains the UNIX login ID, the UNIX user identifier (UID), and the LanMan and NT-encrypted passwords. There is also some other information in this file that we do not need to concern ourselves with here.

  The two newer database types are called ldapsam and tdbsam. Both store considerably more data than the older `smbpasswd` file did. The extra information enables new user account controls to be implemented.

- A corresponding UNIX account, typically stored in `/etc/passwd`. Work is in progress to allow a simplified mode of operation that does not require UNIX user accounts, but this has not been a feature of the early releases of Samba-3, and is not currently planned for release either.

There are three ways to create Machine Trust Accounts:

- Manual creation from the UNIX/Linux command line. Here, both the Samba and corresponding UNIX account are created by hand.
- Using the MS Windows NT4 Server Manager, either from an NT4 domain member server or using the Nexus toolkit available from the Microsoft Web site. This tool can be run from any MS Windows machine as long as the user is logged on as the administrator account.
- "On-the-fly" creation. The Samba Machine Trust Account is automatically created by Samba at the time the client is joined to the domain. (For security, this is the recommended method.) The corresponding UNIX account may be created automatically or manually.

Neither MS Windows NT4/200x/XP Professional, nor Samba, provide any method for enforcing the method of machine trust account creation. This is a matter of the administrator's choice.

6.2.1 Manual Creation of Machine Trust Accounts

The first step in manually creating a Machine Trust Account is to manually create the corresponding UNIX account in `/etc/passwd`. This can be done using **vipw** or another "adduser" command that is normally used to create new UNIX accounts. The following is an example for a Linux-based Samba server:

```
root# /usr/sbin/useradd -g machines -d /var/lib/nobody \
   -c "machine nickname" \
   -s /bin/false machine_name$

root# passwd -l machine_name$
```

In the example above there is an existing system group "machines" which is used as the primary group for all machine accounts. In the following examples the "machines" group numeric GID is 100.

On *BSD systems, this can be done using the **chpass** utility:

```
root# chpass -a \
```

```
'machine_name$:*:101:100::0:0:Windows machine_name:/dev/null:/sbin/nologin'
```

The `/etc/passwd` entry will list the machine name with a "$" appended, and will not have a password, will have a null shell and no home directory. For example, a machine named "doppy" would have an `/etc/passwd` entry like this:

```
doppy$:x:505:100:machine_nickname:/dev/null:/bin/false
```

in which *machine_nickname* can be any descriptive name for the client, such as BasementComputer. *machine_name* absolutely must be the NetBIOS name of the client to be joined to the domain. The "$" must be appended to the NetBIOS name of the client or Samba will not recognize this as a Machine Trust Account.

Now that the corresponding UNIX account has been created, the next step is to create the Samba account for the client containing the well-known initial Machine Trust Account password. This can be done using the **smbpasswd** command as shown here:

```
root# smbpasswd -a -m machine_name
```

where *machine_name* is the machine's NetBIOS name. The RID of the new machine account is generated from the UID of the corresponding UNIX account.

JOIN THE CLIENT TO THE DOMAIN IMMEDIATELY

Manually creating a Machine Trust Account using this method is the equivalent of creating a Machine Trust Account on a Windows NT PDC using the Server Manager. From the time at which the account is created to the time the client joins the domain and changes the password, your domain is vulnerable to an intruder joining your domain using a machine with the same NetBIOS name. A PDC inherently trusts members of the domain and will serve out a large degree of user information to such clients. You have been warned!

6.2.2 Managing Domain Machine Accounts using NT4 Server Manager

A working *add machine script* is essential for machine trust accounts to be automatically created. This applies no matter whether you use automatic account creation or the NT4 Domain Server Manager.

If the machine from which you are trying to manage the domain is an MS Windows NT4 workstation or MS Windows 200x/XP Professional, the tool of choice is the package called **SRVTOOLS.EXE**. When executed in the target directory it will unpack **SrvMgr.exe** and **UsrMgr.exe** (both are domain management tools for MS Windows NT4 workstation).

If your workstation is a Microsoft Windows 9x/Me family product, you should download the **Nexus.exe** package from the Microsoft Web site. When executed from the target directory, it will unpack the same tools but for use on this platform.

Further information about these tools may be obtained from Knowledge Base articles 173673[1], and 172540[2]

Launch the **srvmgr.exe** (Server Manager for Domains) and follow these steps: Server Manager Account Machine Account Management

[1] <http://support.microsoft.com/default.aspx?scid=kb;en-us;173673>
[2] <http://support.microsoft.com/default.aspx?scid=kb;en-us;172540>

1. From the menu select **Computer**.
2. Click **Select Domain**.
3. Click the name of the domain you wish to administer in the **Select Domain** panel and then click **OK**.
4. Again from the menu select **Computer**.
5. Select **Add to Domain**.
6. In the dialog box, click the radio button to **Add NT Workstation of Server**, then enter the machine name in the field provided, and click the **Add** button.

6.2.3 On-the-Fly Creation of Machine Trust Accounts

The third (and recommended) way of creating Machine Trust Accounts is simply to allow the Samba server to create them as needed when the client is joined to the domain.

Since each Samba Machine Trust Account requires a corresponding UNIX account, a method for automatically creating the UNIX account is usually supplied; this requires configuration of the add machine script option in `smb.conf`. This method is not required; however, corresponding UNIX accounts may also be created manually.

Here is an example for a Red Hat Linux system:

```
[global]
        add machine script = /usr/sbin/useradd -d /var/ ↩
            lib/nobody -g 100 -s /bin/false -M %u
```

6.2.4 Making an MS Windows Workstation or Server a Domain Member

The procedure for making an MS Windows workstation or server a member of the domain varies with the version of Windows.

6.2.4.1 Windows 200x/XP Professional Client

When the user elects to make the client a domain member, Windows 200x prompts for an account and password that has privileges to create machine accounts in the domain. A Samba administrator account (i.e., a Samba account that has `root` privileges on the Samba server) must be entered here; the operation will fail if an ordinary user account is given.

For security reasons, the password for this administrator account should be set to a password that is other than that used for the root user in `/etc/passwd`.

The name of the account that is used to create domain member machine trust accounts can be anything the network administrator may choose. If it is other than `root`, then this is easily mapped to `root` in the file named in the `smb.conf` parameter *username map* = /etc/samba/smbusers.

The session key of the Samba administrator account acts as an encryption key for setting the password of the machine trust account. The Machine Trust Account will be created on-the-fly, or updated if it already exists.

6.2.4.2 Windows NT4 Client

If the Machine Trust Account was created manually, on the Identification Changes menu enter the domain name, but do not check the box **Create a Computer Account in the Domain**. In this case, the existing Machine Trust Account is used to join the machine to the domain.

If the Machine Trust Account is to be created on the fly, on the Identification Changes menu enter the domain name and check the box **Create a Computer Account in the Domain**. In this case, joining the domain proceeds as above for Windows 2000 (i.e., you must supply a Samba administrator account when prompted).

6.2.4.3 Samba Client

Joining a Samba client to a domain is documented in Section 6.3.

6.3 Domain Member Server

This mode of server operation involves the Samba machine being made a member of a domain security context. This means by definition that all user authentication will be done from a centrally defined authentication regime. The authentication regime may come from an NT3/4-style (old domain technology) server, or it may be provided from an Active Directory server (ADS) running on MS Windows 2000 or later.

Of course it should be clear that the authentication backend itself could be from any distributed directory architecture server that is supported by Samba. This can be LDAP (from OpenLDAP), or Sun's iPlanet, or Novell e-Directory Server, and so on.

NOTE

When Samba is configured to use an LDAP or other identity management and/or directory service, it is Samba that continues to perform user and machine authentication. It should be noted that the LDAP server does not perform authentication handling in place of what Samba is designed to do.

Please refer to Chapter 4, "Domain Control", for more information regarding how to create a domain machine account for a domain member server as well as for information on how to enable the Samba domain member machine to join the domain and be fully trusted by it.

6.3.1 Joining an NT4-type Domain with Samba-3

Table 6.1 lists names that are used in the remainder of this chapter.

First, you must edit your `smb.conf` file to tell Samba it should now use domain security.

Change (or add) your *security* line in the [global] section of your `smb.conf` to read:

Table 6.1 Assumptions

Samba DMS NetBIOS name:	SERV1
Windows 200x/NT domain name:	MIDEARTH
Domain's PDC NetBIOS name:	DOMPDC
Domain's BDC NetBIOS names:	DOMBDC1 and DOMBDC2

```
        security = domain
```

Note that if the parameter ***security = user*** is used, this machine would function as a standalone server and not as a domain member server. Domain security mode causes Samba to work within the domain security context.

Next change the *workgroup* line in the *[global]* section to read:

```
        workgroup = MIDEARTH
```

This is the name of the domain we are joining.

You must also have the parameter *encrypt passwords* set to `yes` in order for your users to authenticate to the NT PDC. This is the default setting if this parameter is not specified. There is no need to specify this parameter, but if it is specified in the `smb.conf` file, it must be set to `Yes`.

Finally, add (or modify) a *password server* line in the [global] section to read:

```
        password server = DOMPDC DOMBDC1 DOMBDC2
```

These are the PDC and BDCs Samba will attempt to contact in order to authenticate users. Samba will try to contact each of these servers in order, so you may want to rearrange this list in order to spread out the authentication load among Domain Controllers.

Alternatively, if you want smbd to determine automatically the list of domain controllers to use for authentication, you may set this line to be:

```
        password server = *
```

This method allows Samba to use exactly the same mechanism that NT does. The method either uses broadcast-based name resolution, performs a

WINS database lookup in order to find a domain controller against which to authenticate, or locates the domain controller using DNS name resolution.

To join the domain, run this command:

```
root# net rpc join -S DOMPDC -UAdministrator%password
```

If the `-S DOMPDC` argument is not given, the domain name will be obtained from `smb.conf` and the NetBIOS name of the PDC will be obtained either using a WINS lookup or via NetBIOS broadcast based name look up.

The machine is joining the domain DOM, and the PDC for that domain (the only machine that has write access to the domain SAM database) is DOMPDC; therefore, use the `-S` option. The *`Administrator%password`* is the login name and password for an account that has the necessary privilege to add machines to the domain. If this is successful, you will see the following message in your terminal window. Where the older NT4-style domain architecture is used:

```
Joined domain DOM.
```

Where Active Directory is used, the command used to join the ADS domain is:

```
root#  net ads join -UAdministrator%password
```

And the following output is indicative of a successful outcome:

```
Joined SERV1 to realm MYREALM.
```

Refer to the **net** man page and to Chapter 13, "Remote and Local Management: The Net Command" for further information.

This process joins the server to the domain without separately having to create the machine trust account on the PDC beforehand.

This command goes through the machine account password change protocol, then writes the new (random) machine account password for this Samba

server into a file in the same directory in which a smbpasswd file would be normally stored. The trust account information that is needed by the DMS is written into the file `/usr/local/samba/private/secrets.tdb` or `/etc/samba/secrets.tdb`.

This file is created and owned by root and is not readable by any other user. It is the key to the domain-level security for your system and should be treated as carefully as a shadow password file.

Finally, restart your Samba daemons and get ready for clients to begin using domain security. The way you can restart your Samba daemons depends on your distribution, but in most cases the following will suffice:

```
root# /etc/init.d/samba restart
```

6.3.2 Why Is This Better Than security = server?

Currently, domain security in Samba does not free you from having to create local UNIX users to represent the users attaching to your server. This means that if domain user `DOM\fred` attaches to your domain security Samba server, there needs to be a local UNIX user fred to represent that user in the UNIX file system. This is similar to the older Samba security mode *security* = server, where Samba would pass through the authentication request to a Windows NT server in the same way as a Windows 95 or Windows 98 server would.

Please refer to Chapter 24, "Winbind: Use of Domain Accounts", for information on a system to automatically assign UNIX UIDs and GIDs to Windows NT domain users and groups.

The advantage of domain-level security is that the authentication in domain-level security is passed down the authenticated RPC channel in exactly the same way that an NT server would do it. This means Samba servers now participate in domain trust relationships in exactly the same way NT servers do (i.e., you can add Samba servers into a resource domain and have the authentication passed on from a resource domain PDC to an account domain PDC).

In addition, with *security* = server, every Samba daemon on a server has to keep a connection open to the authenticating server for as long as that

daemon lasts. This can drain the connection resources on a Microsoft NT server and cause it to run out of available connections. With *security* = domain, however, the Samba daemons connect to the PDC or BDC only for as long as is necessary to authenticate the user and then drop the connection, thus conserving PDC connection resources.

Finally, acting in the same manner as an NT server authenticating to a PDC means that as part of the authentication reply, the Samba server gets the user identification information such as the user SID, the list of NT groups the user belongs to, and so on.

NOTE

Much of the text of this document was first published in the Web magazine *LinuxWorld*[a] as the article <`http://www.linuxworld.com/linuxworld/lw-1998-10/lw-10-samba.html`> *Doing the NIS/NT Samba*.

[a] <`http://www.linuxworld.com`>

6.4 Samba ADS Domain Membership

This is a rough guide to setting up Samba-3 with Kerberos authentication against a Windows 200x KDC. A familiarity with Kerberos is assumed.

6.4.1 Configure smb.conf

You must use at least the following three options in `smb.conf`:

```
        realm = your.kerberos.REALM
        security = ADS
# The following parameter need only be specified if ↩
    present.
# The default setting if not present is Yes.
        encrypt passwords = yes
```

In case samba cannot correctly identify the appropriate ADS server using the realm name, use the *password server* option in `smb.conf`:

```
password server = your.kerberos.server
```

The most common reason for which Samba may not be able to locate the ADS domain controller is a consequence of sites maintaining some DNS servers on UNIX systems without regard for the DNS requirements of the ADS infrastructure. There is no harm in specifying a preferred ADS domain controller using the *password server*.

NOTE

You do *not* need an smbpasswd file, and older clients will be authenticated as if *security* = domain, although it will not do any harm and allows you to have local users not in the domain.

6.4.2 Configure /etc/krb5.conf

With both MIT and Heimdal Kerberos, it is unnecessary to configure the `/etc/krb5.conf`, and it may be detrimental.

Microsoft ADS automatically create SRV records in the DNS zone *_kerberos._tcp.REALM.NAME* for each KDC in the realm. This is part of the installation and configuration process used to create an Active Directory domain. A KDC is a Kerberos Key Distribution Center and forms an integral part of the Microsoft active directory infrastructure.

UNIX systems can use kinit and the DES-CBC-MD5 or DES-CBC-CRC encryption types to authenticate to the Windows 2000 KDC. For further information regarding Windows 2000 ADS kerberos interoperability please refer to the Microsoft Windows 2000 Kerberos Interoperability[3] guide. Another very useful document that may be referred to for general information

[3] <http://www.microsoft.com/windows2000/techinfo/planning/security/kerbsteps.asp>

regarding Kerberos interoperability is RFC1510[4]. This RFC explains much of the magic behind the operation of Kerberos.

MIT's, as well as Heimdal's, recent KRB5 libraries default to checking for SRV records, so they will automatically find the KDCs. In addition, `krb5.conf` only allows specifying a single KDC, even there if there may be more than one. Using the DNS lookup allows the KRB5 libraries to use whichever KDCs are available.

When manually configuring `krb5.conf`, the minimal configuration is:

```
[libdefaults]
   default_realm = YOUR.KERBEROS.REALM

[realms]
   YOUR.KERBEROS.REALM = {
   kdc = your.kerberos.server
   }

[domain_realms]
   .kerberos.server = YOUR.KERBEROS.REALM
```

When using Heimdal versions before 0.6, use the following configuration settings:

```
[libdefaults]
   default_realm      = YOUR.KERBEROS.REALM
   default_etypes     = des-cbc-crc des-cbc-md5
   default_etypes_des = des-cbc-crc des-cbc-md5

[realms]
        YOUR.KERBEROS.REALM = {
        kdc = your.kerberos.server
   }

[domain_realms]
        .kerberos.server = YOUR.KERBEROS.REALM
```

[4] <http://www.ietf.org/rfc/rfc1510.txt?number=1510>

Test your config by doing a `kinit` *USERNAME@REALM* and making sure that your password is accepted by the Win2000 KDC.

With Heimdal versions earlier than 0.6.x you can use only newly created accounts in ADS or accounts that have had the password changed once after migration, or in case of `Administrator` after installation. At the moment, a Windows 2003 KDC can only be used with Heimdal releases later than 0.6 (and no default etypes in krb5.conf). Unfortunately, this whole area is still in a state of flux.

> **Note**
>
> The realm must be in uppercase or you will get a "Cannot find KDC for requested realm while getting initial credentials" error (Kerberos is case-sensitive!).

> **Note**
>
> Time between the two servers must be synchronized. You will get a "kinit(v5): Clock skew too great while getting initial credentials" if the time difference (clock skew) is more than five minutes.

Clock skew limits are configurable in the Kerberos protocols. The default setting is five minutes.

You also must ensure that you can do a reverse DNS lookup on the IP address of your KDC. Also, the name that this reverse lookup maps to must either be the NetBIOS name of the KDC (i.e., the hostname with no domain attached) or it can be the NetBIOS name followed by the realm.

The easiest way to ensure you get this right is to add a `/etc/hosts` entry mapping the IP address of your KDC to its NetBIOS name. If you do not get this correct, then you will get a local error when you try to join the realm.

If all you want is Kerberos support in smbclient, then you can skip directly to Section 6.4.5 now. Section 6.4.3 and Section 6.4.4 are needed only if you want Kerberos support for smbd and winbindd.

6.4.3 Create the Computer Account

As a user who has write permission on the Samba private directory (usually root), run:

```
root#  net ads join -U Administrator%password
```

The Administrator account can be any account that has been designated in the ADS domain security settings with permission to add machines to the ADS domain. It is, of course, a good idea to use an account other than Administrator. On the UNIX/Linux system, this command must be executed by an account that has UID=0 (root).

When making a Windows client a member of an ADS domain within a complex organization, you may want to create the machine trust account within a particular organizational unit. Samba-3 permits this to be done using the following syntax:

```
root#  kinit Administrator@your.kerberos.REALM
root#  net ads join createcomputer="organizational_unit"
```

Your ADS manager will be able to advise what should be specified for the "organizational_unit" parameter.

For example, you may want to create the machine trust account in a container called "Servers" under the organizational directory "Computers/BusinessUnit/Department," like this:

```
root#  net ads join "Computers/BusinessUnit/Department/Servers"
```

This command will place the Samba server machine trust account in the container `Computers/BusinessUnit/Department/Servers`. The container should exist in the ADS directory before executing this command. Please

note that forward slashes must be used, because backslashes are both valid characters in an OU name and used as escapes for other characters. If you need a backslash in an OU name, it may need to be quadrupled to pass through the shell escape and ldap escape.

6.4.3.1 Possible Errors

ADS support not compiled in Samba must be reconfigured (remove config.cache) and recompiled (make clean all install) after the Kerberos libraries and headers files are installed.

net ads join prompts for user name You need to log in to the domain using `kinit` *USERNAME@REALM*. *USERNAME* must be a user who has rights to add a machine to the domain.

Unsupported encryption/or checksum types Make sure that the `/etc/krb5.conf` is correctly configured for the type and version of Kerberos installed on the system.

6.4.4 Testing Server Setup

If the join was successful, you will see a new computer account with the NetBIOS name of your Samba server in Active Directory (in the "Computers" folder under Users and Computers.

On a Windows 2000 client, try `net use * \\server\share`. You should be logged in with Kerberos without needing to know a password. If this fails, then run `klist tickets`. Did you get a ticket for the server? Does it have an encryption type of DES-CBC-MD5?

NOTE

Samba can use both DES-CBC-MD5 encryption as well as ARCFOUR-HMAC-MD5 encoding.

6.4.5 Testing with smbclient

On your Samba server try to log in to a Windows 2000 server or your Samba server using smbclient and Kerberos. Use smbclient as usual, but specify the `-k` option to choose Kerberos authentication.

6.4.6 Notes

You must change the administrator password at least once after installing a domain controller, to create the right encryption types.

Windows 200x does not seem to create the *_kerberos._udp* and *_ldap._tcp* in the default DNS setup. Perhaps this will be fixed later in service packs.

6.5 Sharing User ID Mappings between Samba Domain Members

Samba maps UNIX users and groups (identified by UIDs and GIDs) to Windows users and groups (identified by SIDs). These mappings are done by the *idmap* subsystem of Samba.

In some cases it is useful to share these mappings between Samba domain members, so *name->id* mapping is identical on all machines. This may be needed in particular when sharing files over both CIFS and NFS.

To use the *LDAP* *ldap idmap suffix*, set:

```
        ldap idmap suffix = ou=Idmap
```

See the `smb.conf` man page entry for the *ldap idmap suffix* parameter for further information.

Do not forget to specify also the *ldap admin dn* and to make certain to set the LDAP administrative password into the `secrets.tdb` using:

```
root#  smbpasswd -w ldap-admin-password
```

In place of `ldap-admin-password`, substitute the LDAP administration password for your system.

6.6 Common Errors

In the process of adding/deleting/re-adding domain member machine trust accounts, there are many traps for the unwary player and many "little" things that can go wrong. It is particularly interesting how often subscribers on the Samba mailing list have concluded after repeated failed attempts to add a machine account that it is necessary to "reinstall" MS Windows on the machine. In truth, it is seldom necessary to reinstall because of this type of problem. The real solution is often quite simple, and with an understanding of how MS Windows networking functions, it is easy to overcome.

6.6.1 Cannot Add Machine Back to Domain

"A Windows workstation was reinstalled. The original domain machine trust account was deleted and added immediately. The workstation will not join the domain if I use the same machine name. Attempts to add the machine fail with a message that the machine already exists on the network — I know it does not. Why is this failing?"

The original name is still in the NetBIOS name cache and must expire after machine account deletion before adding that same name as a domain member again. The best advice is to delete the old account and then add the machine with a new name. Alternately, the name cache can be flushed and reloaded with current data using the **nbtstat** command on the Windows client:

```
C:\>  nbtstat -R
```

6.6.2 Adding Machine to Domain Fails

"Adding a Windows 200x or XP Professional machine to the Samba PDC Domain fails with a message that says, "The machine could not be added at this time, there is a network problem. Please try again later." Why?"

You should check that there is an *add machine script* in your `smb.conf` file. If there is not, please add one that is appropriate for your OS platform. If a script has been defined, you will need to debug its operation. Increase

the *log level* in the `smb.conf` file to level 10, then try to rejoin the domain. Check the logs to see which operation is failing.

Possible causes include:

- The script does not actually exist, or could not be located in the path specified.

  Corrective action: Fix it. Make sure when run manually that the script will add both the UNIX system account and the Samba SAM account.

- The machine could not be added to the UNIX system accounts file `/etc/passwd`.

  Corrective action: Check that the machine name is a legal UNIX system account name. If the UNIX utility **useradd** is called, then make sure that the machine name you are trying to add can be added using this tool. **Useradd** on some systems will not allow any uppercase characters nor will it allow spaces in the name.

The *add machine script* does not create the machine account in the Samba backend database; it is there only to create a UNIX system account to which the Samba backend database account can be mapped.

6.6.3 I Can't Join a Windows 2003 PDC

Windows 2003 requires SMB signing. Client-side SMB signing has been implemented in Samba-3.0. Set *client use spnego* = yes when communicating with a Windows 2003 server. This will not interfere with other Windows clients that do not support the more advanced security features of Windows 2003 because the client will simply negotiate a protocol that both it and the server suppport. This is a well-known fall-back facility that is built into the SMB/CIFS protocols.

Chapter 7

STANDALONE SERVERS

Standalone servers are independent of domain controllers on the network. They are not domain members and function more like workgroup servers. In many cases a standalone server is configured with a minimum of security control with the intent that all data served will be readily accessible to all users.

7.1 Features and Benefits

Standalone servers can be as secure or as insecure as needs dictate. They can have simple or complex configurations. Above all, despite the hoopla about domain security, they remain a common installation.

If all that is needed is a server for read-only files, or for printers alone, it may not make sense to effect a complex installation. For example, a drafting office needs to store old drawings and reference standards. Noone can write files to the server because it is legislatively important that all documents remain unaltered. A share-mode read-only standalone server is an ideal solution.

Another situation that warrants simplicity is an office that has many printers that are queued off a single central server. Everyone needs to be able to print to the printers, there is no need to effect any access controls, and no files will be served from the print server. Again, a share-mode standalone server makes a great solution.

7.2 Background

The term *standalone server* means that it will provide local authentication and access control for all resources that are available from it. In general this means that there will be a local user database. In more technical terms, it means resources on the machine will be made available in either *share* mode or in *user* mode.

No special action is needed other than to create user accounts. Standalone servers do not provide network logon services. This means that machines that use this server do not perform a domain logon to it. Whatever logon facility the workstations are subject to is independent of this machine. It is, however, necessary to accommodate any network user so the logon name he or she uses will be translated (mapped) locally on the standalone server to a locally known user name. There are several ways this can be done.

Samba tends to blur the distinction a little in defining a standalone server. This is because the authentication database may be local or on a remote server, even if from the SMB protocol perspective the Samba server is not a member of a domain security context.

Through the use of Pluggable Authentication Modules (PAM) (see Chapter 28, "PAM-Based Distributed Authentication") and the name service switcher (NSS), which maintains the UNIX-user database, the source of authentication may reside on another server. We would be inclined to call this the authentication server. This means that the Samba server may use the local UNIX/Linux system password database (`/etc/passwd` or `/etc/shadow`), may use a local smbpasswd file, or may use an LDAP backend, or even via PAM and Winbind another CIFS/SMB server for authentication.

7.3 Example Configuration

Example 7.3.1 and Section 7.3.2 are designed to inspire simplicity. It is too easy to attempt a high level of creativity and to introduce too much complexity in server and network design.

7.3.1 Reference Documentation Server

Configuration of a read-only data server that everyone can access is very simple. By default, all shares are read-only, unless set otherwise in the `smb.`

conf file. Example 7.3.1 is the smb.conf file that will do this. Assume that all the reference documents are stored in the directory /export, and the documents are owned by a user other than nobody. No home directories are shared, and there are no users in the /etc/passwd UNIX system database. This is a simple system to administer.

Example 7.3.1 smb.conf for Reference Documentation Server

```
# Global parameters
 [global]
        workgroup = MIDEARTH
        netbios name = GANDALF
        security = SHARE
        passdb backend = guest
        wins server = 192.168.1.1
 [data]
        comment = Data
        path = /export
        guest only = Yes
```

> I would have spoken more briefly, if I'd had more time to prepare.
>
> —Mark Twain

In Example 7.3.1, the machine name is set to GANDALF, and the workgroup is set to the name of the local workgroup (MIDEARTH) so the machine will appear together with systems with which users are familiar. The only password backend required is the "guest" backend to allow default unprivileged account names to be used. As there is a WINS server on this network, we of course make use of it.

A US Air Force Colonel was renowned for saying: "Better is the enemy of good enough!" There are often sound reasons for avoiding complexity as well as for avoiding a technically perfect solution. Unfortunately, many network administrators still need to learn the art of doing just enough to keep out of trouble.

7.3.2 Central Print Serving

Configuration of a simple print server is easy if you have all the right tools on your system.

Assumptions

1. The print server must require no administration.
2. The print spooling and processing system on our print server will be CUPS. (Please refer to Chapter 22, "CUPS Printing Support", for more information).
3. The print server will service only network printers. The network administrator will correctly configure the CUPS environment to support the printers.
4. All workstations will use only PostScript drivers. The printer driver of choice is the one shipped with the Windows OS for the Apple Color LaserWriter.

In this example our print server will spool all incoming print jobs to `/var/spool/samba` until the job is ready to be submitted by Samba to the CUPS print processor. Since all incoming connections will be as the anonymous (guest) user, two things will be required to enable anonymous printing.

Enabling Anonymous Printing

- The UNIX/Linux system must have a **guest** account. The default for this is usually the account **nobody**. To find the correct name to use for your version of Samba, do the following:

  ```
  $ testparm -s -v | grep "guest account"
  ```

  Make sure that this account exists in your system password database (`/etc/passwd`).

  It is a good idea either to set a password on this account, or else to lock it from UNIX use. Assuming that the guest account is called `pcguest`, it can be locked by executing:

  ```
  root#  passwd -l pcguest
  ```

  The exact command may vary depending on your UNIX/Linux distribution.

- The directory into which Samba will spool the file must have write access for the guest account. The following commands will ensure that this directory is available for use:

```
root# mkdir /var/spool/samba
root# chown nobody.nobody /var/spool/samba
root# chmod a+rwt /var/spool/samba
```

The contents of the `smb.conf` file is shown in Example 7.3.2.

Example 7.3.2 smb.conf for Anonymous Printing

```
# Global parameters
 [global]
        workgroup = MIDEARTH
        netbios name = GANDALF
        security = SHARE
        passdb backend = guest
        printing = cups
        printcap name = cups
 [printers]
        comment = All Printers
        path = /var/spool/samba
        printer admin = root
        guest ok = Yes
        printable = Yes
        use client driver = Yes
        browseable = No
```

NOTE

On CUPS-enabled systems there is a facility to pass raw data directly to the printer without intermediate processing via CUPS print filters. Where use of this mode of operation is desired, it is necessary to configure a raw printing device. It is also necessary to enable the raw mime handler in the `/etc/mime.conv` and `/etc/mime.types` files. Refer to Chapter 22, "CUPS Printing Support", Section 22.3.4.

The example in Example 7.3.2 uses CUPS for direct printing via the CUPS libarary API. This means that all printers will be exposed to Windows users without need to configure a printcap file. If there is necessity to expose only a sub-set of printers, or to define a special type of printer (for example, a PDF filter) the *printcap name = cups* can be replaced with the entry *printcap name = /etc/samba/myprintcap*. In this case the file specified should contain a list of the printer names that should be exposed to Windows network users.

7.4 Common Errors

The greatest mistake so often made is to make a network configuration too complex. It pays to use the simplest solution that will meet the needs of the moment.

Chapter 8

MS WINDOWS NETWORK CONFIGURATION GUIDE

8.1 Features and Benefits

Occasionally network administrators report difficulty getting Microsoft Windows clients to interoperate correctly with Samba servers. It seems that some folks just cannot accept the fact that the right way to configure an MS Windows network client is precisely as one would do when using MS Windows NT4 or 200x servers. Yet there is repetitious need to provide detailed Windows client configuration instructions.

The purpose of this chapter is to graphically illustrate MS Windows client configuration for the most common critical aspects of such configuration. An experienced network administrator will not be interested in the details of this chapter.

8.2 Technical Details

This chapter discusses TCP/IP protocol configuration as well as network membership for the platforms that are in common use today. These are:

- Microsoft Windows XP Professional
- Windows 2000 Professional
- Windows Millennium edition (Me)

8.2.1 TCP/IP Configuration

The builder of a house must ensure that all construction takes place on a firm foundation. The same is true for the builder of a TCP/IP-based networking system. Fundamental network configuration problems will plague all network users until they are resolved.

MS Windows workstations and servers can be configured either with fixed IP addresses or via DHCP. The examples that follow demonstrate the use of DHCP and make only passing reference to those situations where fixed IP configuration settings can be effected.

It is possible to use shortcuts or abbreviated keystrokes to arrive at a particular configuration screen. The decision was made to base all examples in this chapter on use of the **Start** button.

8.2.1.1 MS Windows XP Professional

There are two paths to the Windows XP TCP/IP configuration panel. Choose the access method that you prefer:

Click **Start** -> **Control Panel** -> **Network Connections**.

Alternately, click **Start** ->, and right-click **My Network Places** then select **Properties**.

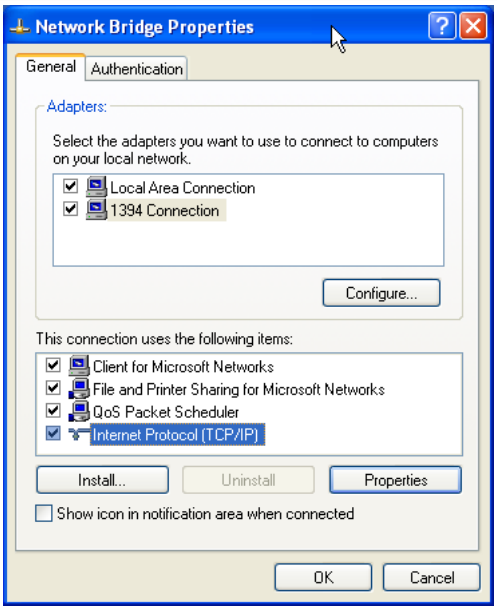
The following procedure steps through the Windows XP Professional TCP/IP configuration process:

1. On some installations the interface will be called **Local Area Connection** and on others it will be called **Network Bridge**. On our system it is called **Network Bridge**. Right-click on **Network Bridge** -> **Properties**. See Figure 8.1.

2. The Network Bridge Configuration, or Local Area Connection, panel is used to set TCP/IP protocol settings. In **This connection uses the following items:** box, click on **Internet Protocol (TCP/IP)**, then click on **Properties**. The default setting is DHCP-enabled operation (i.e., "Obtain an IP address automatically"). See Figure 8.2.

   Many network administrators will want to use DHCP to configure all client TCP/IP protocol stack settings. (For information on how to configure the ISC DHCP server for Windows client support see

Figure 8.1 Network Bridge Configuration.



Section 48.2.2, Section 48.2.2. If it is necessary to provide a fixed IP address, click on "Use the following IP address" and enter the IP Address, the subnet mask, and the default gateway address in the boxes provided.

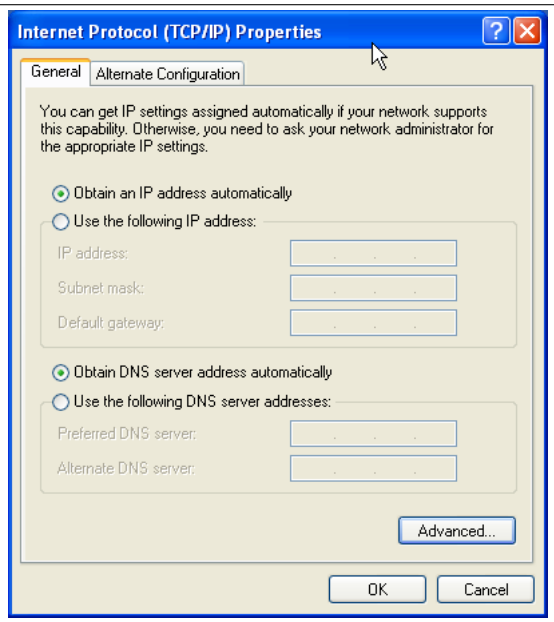
3. Click the **Advanced** button to proceed with TCP/IP configuration. This opens a panel in which it is possible to create additional IP addresses for this interface. The technical name for the additional addresses is *IP aliases*, and additionally this panel permits the setting of more default gateways (routers). In most cases where DHCP is used, it will not be necessary to create additional settings. See Figure 8.3 to see the appearance of this panel.

   Fixed settings may be required for DNS and WINS if these settings are not provided automatically via DHCP.

4. Click the **DNS** tab to add DNS server settings. The example system uses manually configured DNS settings. When finished making changes, click the **OK** to commit the settings. See Figure 8.4.

5. Click the **WINS** tab to add manual WINS server entries. This step demonstrates an example system that uses manually configured WINS

Figure 8.2 Internet Protocol (TCP/IP) Properties.



settings. When finished making changes, click **OK** to commit the settings. See Figure 8.5.

8.2.1.2 MS Windows 2000

There are two paths to the Windows 2000 Professional TCP/IP configuration panel. Choose the access method that you prefer:

Click **Start** -> **Control Panel** -> **Network and Dial-up Connections**.

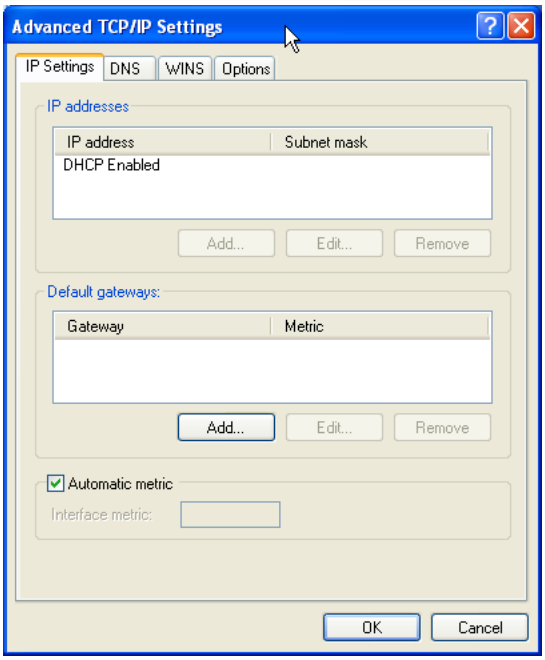
Alternatively, click **Start**, then right-click **My Network Places**, and select **Properties**.

The following procedure steps through the Windows XP Professional TCP/IP configuration process:

1. Right-click on **Local Area Connection**, then click **Properties**. See Figure 8.6.

2. The Local Area Connection Properties is used to set TCP/IP protocol settings. Click on **Internet Protocol (TCP/IP)** in the **Components**

Figure 8.3 Advanced Network Settings



checked are used by this connection: box, then click the **Properties** button.

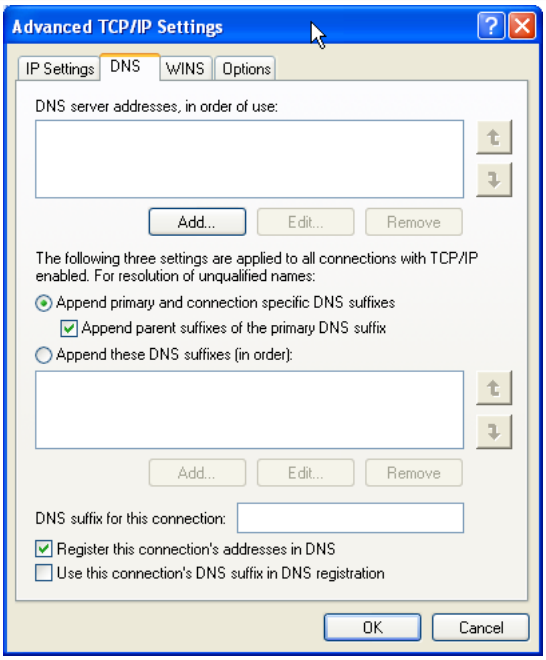
3. The default setting is DHCP-enabled operation (i.e., "Obtain an IP address automatically"). See Figure 8.7.

   Many network administrators will want to use DHCP to configure all client TCP/IP protocol stack settings. (For information on how to configure the ISC DHCP server for Windows client support, see, Section 48.2.2. If it is necessary to provide a fixed IP address, click on "Use the following IP address" and enter the IP Address, the subnet mask, and the default gateway address in the boxes provided. For this example we are assuming that all network clients will be configured using DHCP.

4. Click the **Advanced** button to proceed with TCP/IP configuration. Refer to Figure 8.8.

   Fixed settings may be required for DNS and WINS if these settings are not provided automatically via DHCP.

Figure 8.4 DNS Configuration.



5. Click the **DNS** tab to add DNS server settings. The example system uses manually configured DNS settings. When finished making changes, click **OK** to commit the settings. See Figure 8.9.

6. Click the **WINS** tab to add manual WINS server entries. This step demonstrates an example system that uses manually configured WINS settings. When finished making changes, click **OK** to commit the settings. See Figure 8.10.

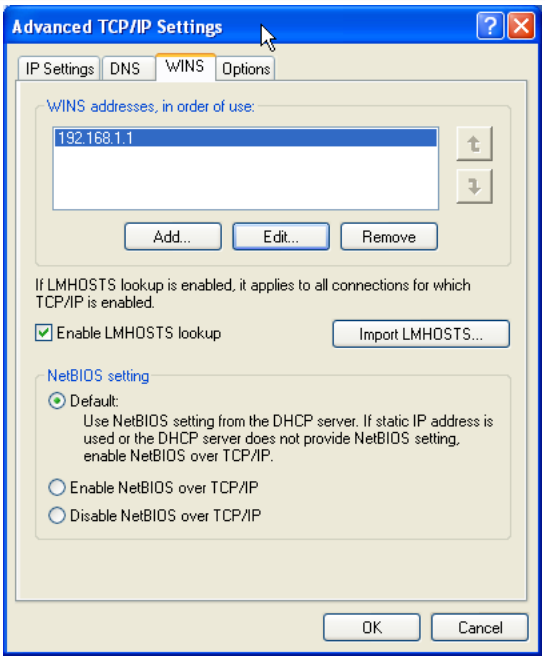
8.2.1.3 MS Windows Me

There are two paths to the Windows Millennium edition (Me) TCP/IP configuration panel. Choose the access method that you prefer:

Click **Start** -> **Control Panel** -> **Network Connections**.

Alternatively, click on **Start** ->, and right click on **My Network Places** then select **Properties**.

Figure 8.5 WINS Configuration



The following procedure steps through the Windows Me TCP/IP configuration process:

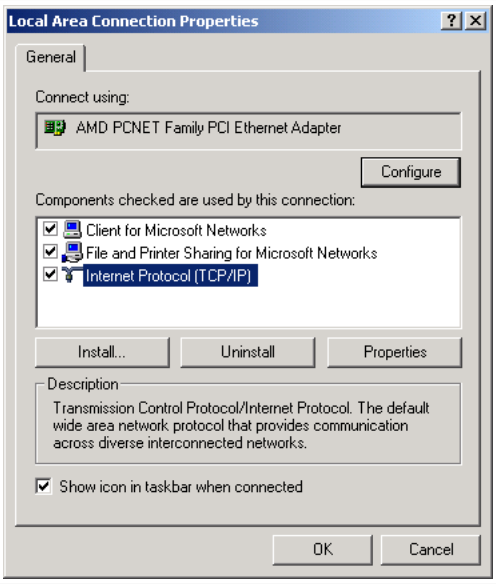
1. In the box labeled **The following network components are installed:**, click on **Internet Protocol TCP/IP**, then click on the **Properties** button. See Figure 8.11.

2. Many network administrators will want to use DHCP to configure all client TCP/IP protocol stack settings. (For information on how to configure the ISC DHCP server for Windows client support see Section 48.2.2, Section 48.2.2. The default setting on Windows Me workstations is for DHCP-enabled operation (i.e., **Obtain IP address automatically** is enabled). See Figure 8.12.

   If it is necessary to provide a fixed IP address, click on **Specify an IP address** and enter the IP Address and the subnet mask in the boxes provided. For this example we are assuming that all network clients will be configured using DHCP.

3. Fixed settings may be required for DNS and WINS if these settings are not provided automatically via DHCP.

Figure 8.6 Local Area Connection Properties.



4. If necessary, click the **DNS Configuration** tab to add DNS server settings. Click the **WINS Configuration** tab to add WINS server settings. The **Gateway** tab allows additional gateways (router addresses) to be added to the network interface settings. In most cases where DHCP is used, it will not be necessary to create these manual settings.

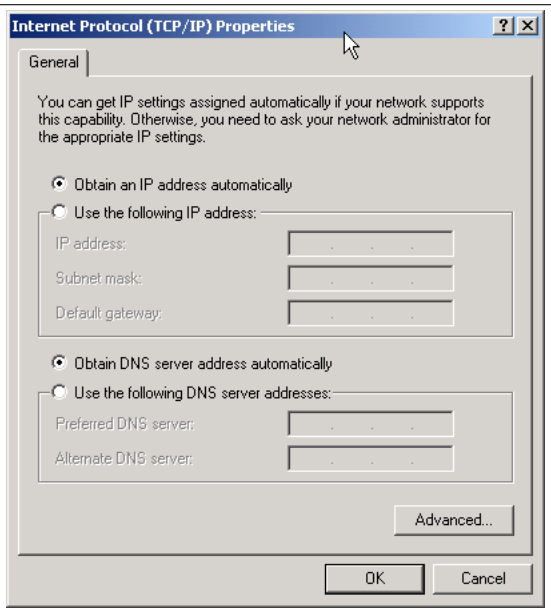
5. The following example uses manually configured WINS settings. See Figure 8.13. When finished making changes, click **OK** to commit the settings.

   This is an example of a system that uses manually configured WINS settings. One situation where this might apply is on a network that has a single DHCP server that provides settings for multiple Windows workgroups or domains. See Figure 8.14.

8.2.2 Joining a Domain: Windows 2000/XP Professional

Microsoft Windows NT/200x/XP Professional platforms can participate in domain security. This section steps through the process for making a Windows 200x/XP Professional machine a member of a domain security envi-

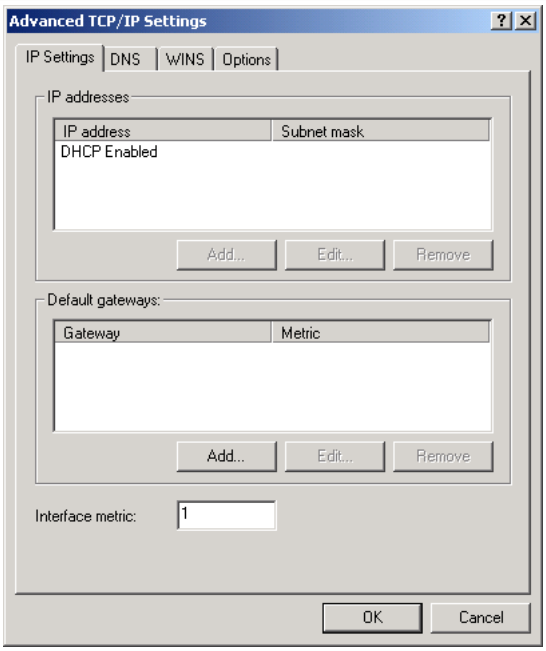
Figure 8.7 Internet Protocol (TCP/IP) Properties.



ronment. It should be noted that this process is identical when joining a domain that is controlled by Windows NT4/200x as well as a Samba PDC.

1. Click **Start**.
2. Right-click **My Computer**, then select **Properties**.
3. The opening panel is the same one that can be reached by clicking **System** on the Control Panel. See Figure 8.15.
4. Click the **Computer Name** tab. This panel shows the **Computer Description**, the **Full computer name**, and the **Workgroup** or **Domain name**. Clicking the **Network ID** button will launch the configuration wizard. Do not use this with Samba-3. If you wish to change the computer name or join or leave the domain, click the **Change** button. See Figure 8.16.
5. Click on **Change**. This panel shows that our example machine (TEMPTATION) is in a workgroup called WORKGROUP. We will join the domain called MIDEARTH. See Figure 8.17.
6. Enter the name **MIDEARTH** in the field below the domain radio button. This panel shows that our example machine (TEMPTATION) is

Figure 8.8 Advanced Network Settings.



set to join the domain called MIDEARTH. See Figure 8.18.

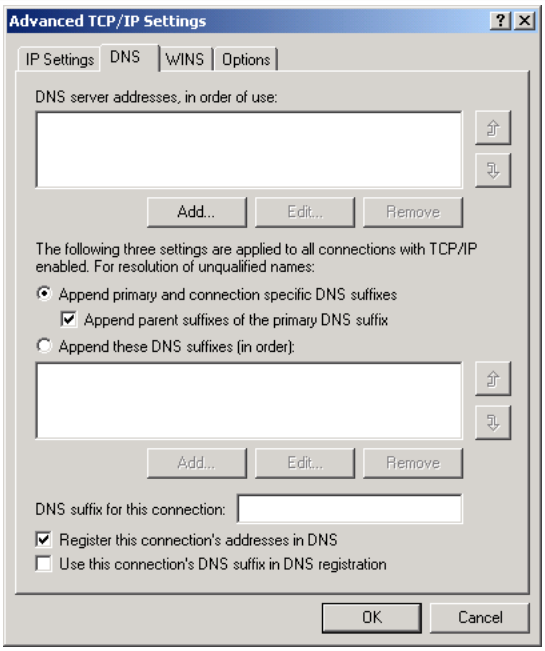
7. Now click the **OK** button. A dialog box should appear to allow you to provide the credentials (username and password) of a domain administrative account that has the rights to add machines to the domain. Enter the name "root" and the root password from your Samba-3 server. See Figure 8.19.

8. Click on **OK**. The "Welcome to the MIDEARTH domain." dialog box should appear. At this point the machine must be rebooted. Joining the domain is now complete.

8.2.3 Domain Logon Configuration: Windows 9x/Me

We follow the convention used by most in saying that Windows 9x/Me machines can participate in domain logons. The truth is that these platforms can use only the LanManager network logon protocols.

Figure 8.9 DNS Configuration.



NOTE

Windows XP Home edition cannot participate in domain or LanManager network logons.

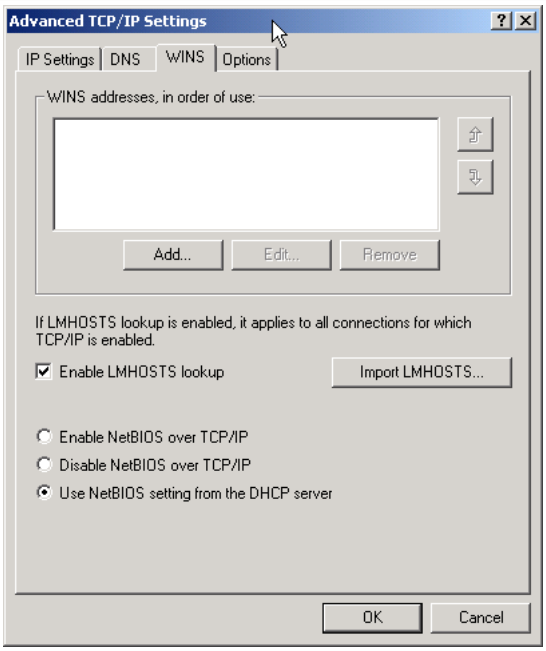
1. Right-click on the **Network Neighborhood** icon.

2. The Network Configuration Panel allows all common network settings to be changed. See Figure 8.20.

   Make sure that the **Client for Microsoft Networks** driver is installed as shown. Click on the **Client for Microsoft Networks** entry in **The following network components are installed:** box. Then click the **Properties** button.

3. The Client for Microsoft Networks Properties panel is the correct location to configure network logon settings. See Figure 8.21.

Figure 8.10 WINS Configuration.



Enter the Windows NT domain name, check the **Log on to Windows NT domain** box, and click **OK**.

4. Click on the **Identification** button. This is the location at which the workgroup (domain) name and the machine name (computer name) need to be set. See Figure 8.22.

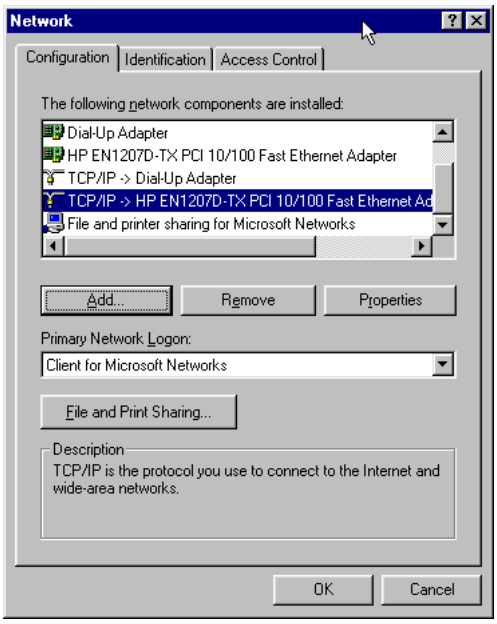
5. Now click the **Access Control** button. If you want to be able to assign share access permissions using domain user and group accounts, it is necessary to enable **User-level access control** as shown in this panel. See Figure 8.23.

8.3 Common Errors

The most common errors that can afflict Windows networking systems include:

- Incorrect IP address.
- Incorrect or inconsistent netmasks.

Figure 8.11 The Windows Me Network Configuration Panel.



- Incorrect router address.
- Incorrect DNS server address.
- Incorrect WINS server address.
- Use of a Network Scope setting — watch out for this one!

The most common reasons for which a Windows NT/200x/XP Professional client cannot join the Samba controlled domain are:

- `smb.conf` does not have correct *add machine script* settings.
- "root" account is not in password backend database.
- Attempt to use a user account instead of the "root" account to join a machine to the domain.
- Open connections from the workstation to the server.
- Firewall or filter configurations in place on either the client or the Samba server.

Figure 8.12 IP Address.

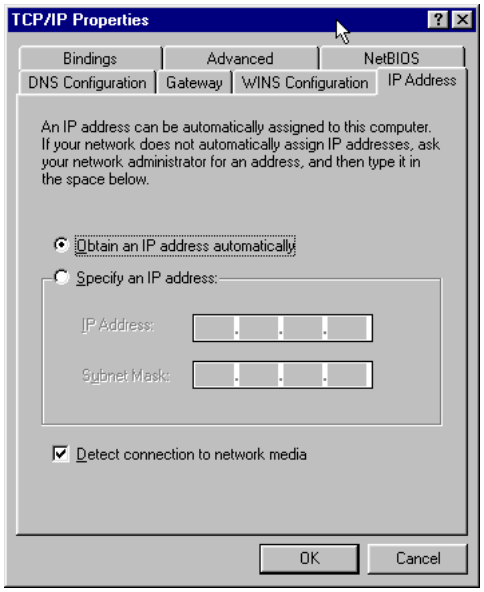


Figure 8.13 DNS Configuration.

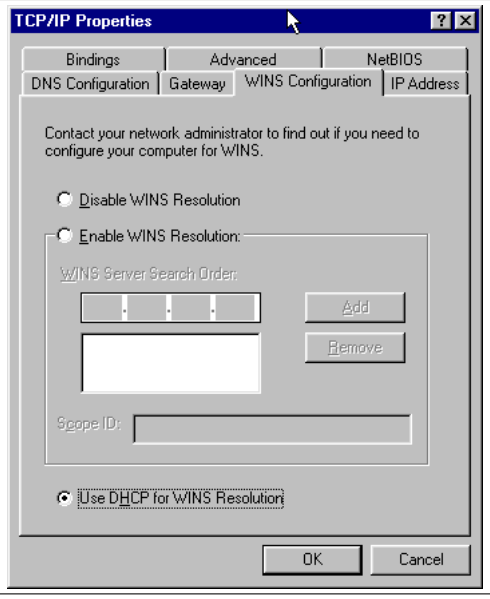


Figure 8.14 WINS Configuration.

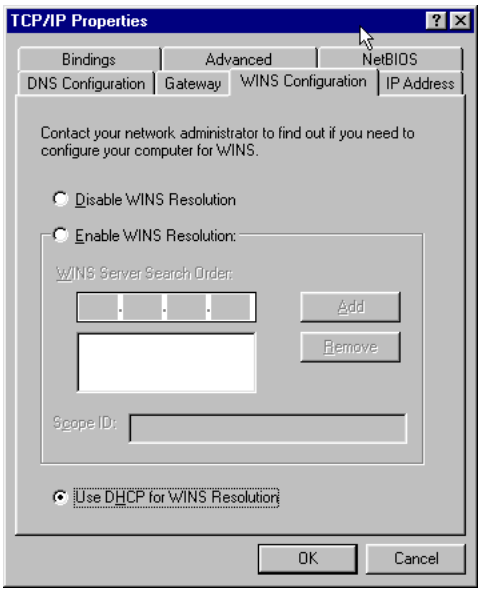


Figure 8.15 The General Panel.

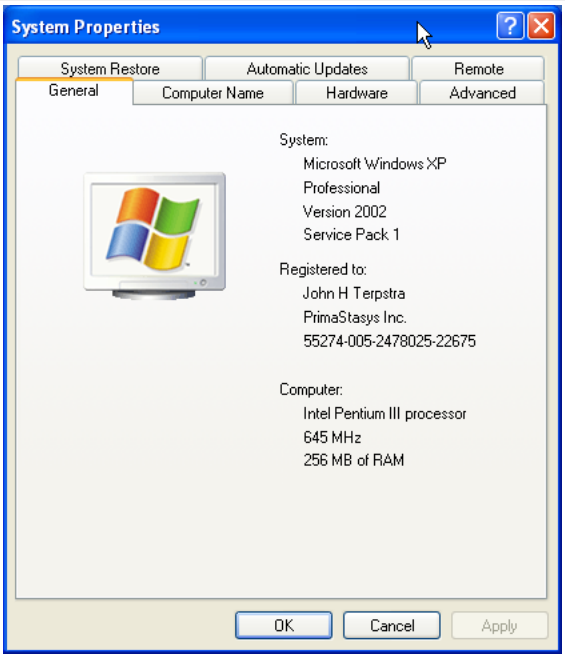


Figure 8.16 The Computer Name Panel.

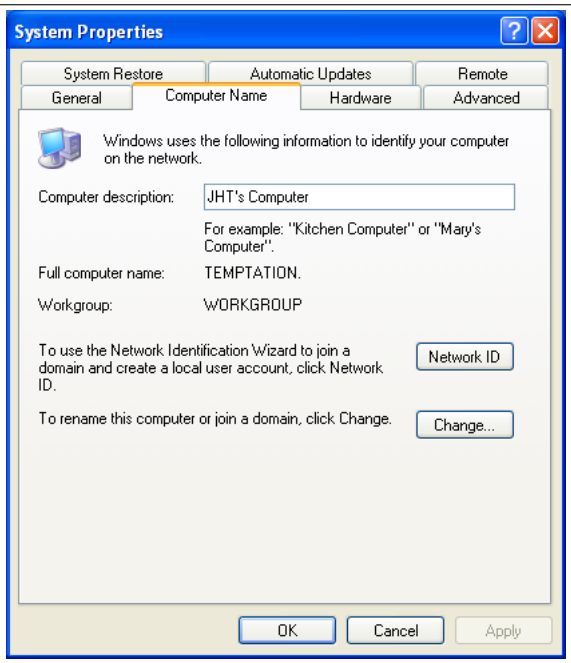


Figure 8.17 The Computer Name Changes Panel.

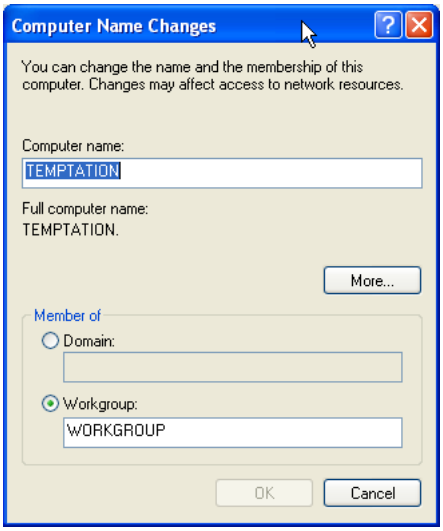


Figure 8.18 The Computer Name Changes Panel — Domain MIDEARTH.

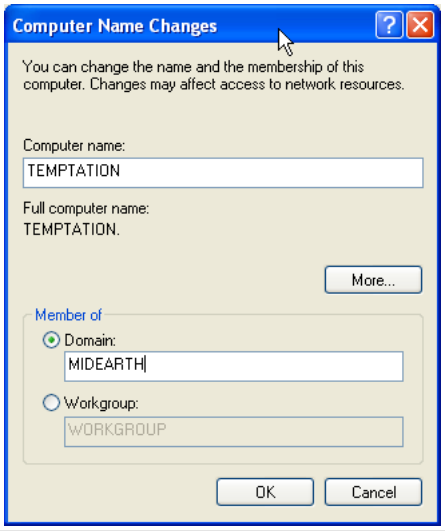


Figure 8.19 Computer Name Changes — Username and Password Panel.

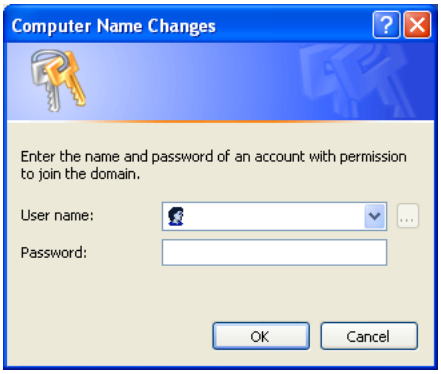


Figure 8.20 The Network Panel.

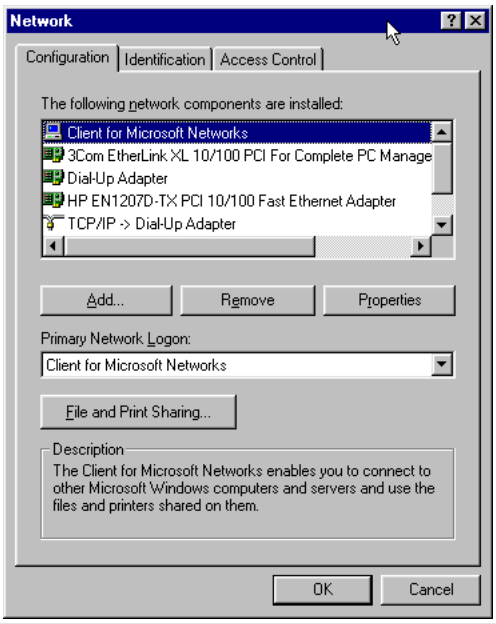


Figure 8.21 Client for Microsoft Networks Properties Panel.

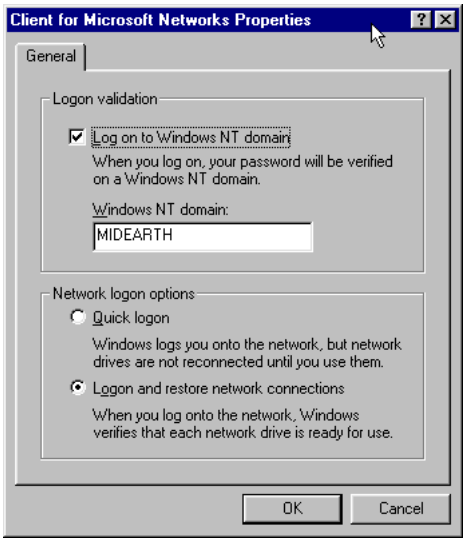


Figure 8.22 Identification Panel.

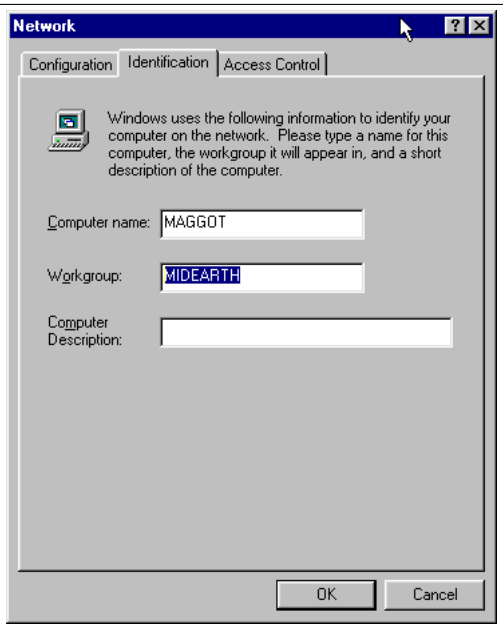
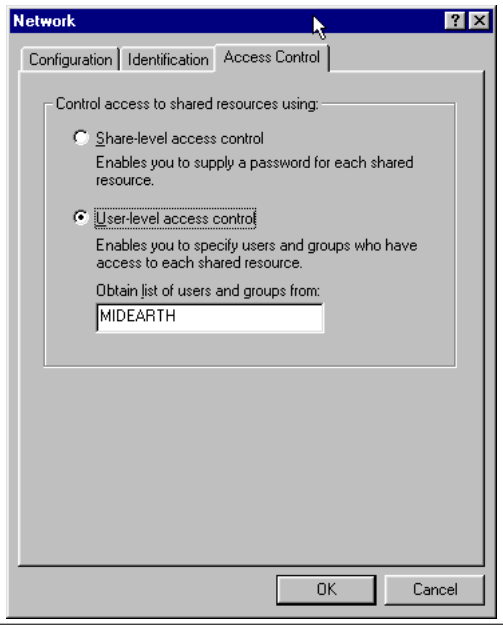


Figure 8.23 Access Control Panel.



Part III

Advanced Configuration

VALUABLE NUTS AND BOLTS INFORMATION

Samba has several features that you might want or might not want to use. The chapters in this part each cover specific Samba features.

Chapter 9

IMPORTANT AND CRITICAL CHANGE NOTES FOR THE SAMBA 3.X SERIES

Please read this chapter carefully before update or upgrading Samba. You should expect to find only critical or very important information here. Comprehensive change notes and guidance information can be found in the section Chapter 35, "Updating and Upgrading Samba".

9.1 Important Samba-3.2.x Change Notes

!!!!!!!!!!!!Add all critical update notes here!!!!!!!!!!!!!

9.2 Important Samba-3.0.x Change Notes

These following notes pertain in particular to Samba 3.0.23 through Samba 3.0.25c (or more recent 3.0.25 update). Samba is a fluid and ever changing project. Changes throughout the 3.0.x series release are documented in this documention - See Section 35.1.2.

Sometimes it is difficult to figure out which part, or parts, of the HOWTO documentation should be updated to reflect the impact of new or modified features. At other times it becomes clear that the documentation is in need of being restructured.

In recent times a group of Samba users has joined the thrust to create a new Samba Wiki[1] that is slated to become the all-singing and all-dancing new face of Samba documentation. Hopefully, the Wiki will benefit from greater community input and thus may be kept more up to date. Until that golden dream materializes and matures it is necessary to continue to maintain the HOWTO. This chapter will document major departures from earlier behavior until such time as the body of this HOWTO is restructured or modified.

This chapter is new to the release of the HOWTO for Samba 3.0.23. It includes much of the notes provided in the `WHATSNEW.txt` file that is included with the Samba source code release tarball.

9.2.1 User and Group Changes

The change documented here affects unmapped user and group accounts only.

The user and group internal management routines have been rewritten to prevent overlaps of assigned Relative Identifiers (RIDs). In the past the has been a potential problem when either manually mapping Unix groups with the **net groupmap** command or when migrating a Windows domain to a Samba domain by executing: **net rpc vampire**.

Unmapped users are now assigned a SID in the `S-1-22-1` domain and unmapped groups are assigned a SID in the `S-1-22-2` domain. Previously they were assigned a RID within the SAM on the Samba server. For a domain controller this would have been under the authority of the domain SID where as on a member server or standalone server, this would have been under the authority of the local SAM (see the man page for **net getlocalsid**).

The result is that any unmapped users or groups on an upgraded Samba domain controller may be assigned a new SID. Because the SID rather than a name is stored in Windows security descriptors, this can cause a user to no longer have access to a resource for example if a file was copied from a Samba file server to a local Windows client NTFS partition. Any files stored on the Samba server itself will continue to be accessible because UNIX stores the UNIX GID and not the SID for authorization checks.

An example helps to illustrate the change:

[1] <http://wiki.samba.org/>

Assume that a group named *developers* exists with a UNIX GID of 782. In this case this user does not exist in Samba's group mapping table. It would be perfectly normal for this group to be appear in an ACL editor. Prior to Samba-3.0.23, the group SID might appear as `S-1-5-21-647511796-4126122067-3123570092-2565`.

With the release of Samba-3.0.23, the group SID would be reported as `S-1-22-2-782`. Any security descriptors associated with files stored on a Windows NTFS disk partition will not allow access based on the group permissions if the user was not a member of the `S-1-5-21-647511796-4126122067-3123570092-2565` group. Because this group SID is `S-1-22-2-782` and not reported in a user's token, Windows would fail the authorization check even though both SIDs in some respect refer to the same UNIX group.

The workaround for versions of Samba prior to 3.0.23, is to create a manual domain group mapping entry for the group *developers* to point at the `S-1-5-21-647511796-4126122067-3123570092-2565` SID. With the release of Samba-3.0.23 this workaround is no longer needed.

9.2.2 Essential Group Mappings

Samba 3.0.x series releases before 3.0.23 automatically created group mappings for the essential Windows domain groups `Domain Admins, Domain Users, Domain Guests`. Commencing with Samba 3.0.23 these mappings need to be created by the Samba administrator. Failure to do this may result in a failure to correctly authenticate and recoognize valid domain users. When this happens users will not be able to log onto the Windows client.

> **Note**
>
> Group mappings are essentail only if the Samba servers is running as a PDC/BDC. Stand-alone servers do not require these group mappings.

The following mappings are required:

Table 9.1 Essential Domain Group Mappings

Domain Group	RID	Example UNIX Group
Domain Admins	512	root
Domain Users	513	users
Domain Guests	514	nobody

When the POSIX (UNIX) groups are stored in LDAP, it may be desirable to call these `domadmins, domusers, domguests` respectively.

For further information regarding group mappings see Chapter 12, "Group Mapping: MS Windows and UNIX".

9.2.3 Passdb Changes

The *passdb backend* parameter no long accepts multiple passdb backends in a chained configuration. Also be aware that the SQL and XML based passdb modules have been removed in the Samba-3.0.23 release. More information regarding external support for a SQL passdb module can be found on the pdbsql[2] web site.

9.2.4 Group Mapping Changes in Samba-3.0.23

The default mapping entries for groups such as `Domain Admins` are no longer created when using an `smbpasswd` file or a `tdbsam` passdb backend. This means that it is necessary to explicitly execute the **net groupmap add** to create group mappings, rather than use the **net groupmap modify** method to create the Windows group SID to UNIX GID mappings. This change has no effect on winbindd's IDMAP functionality for domain groups.

9.2.5 LDAP Changes in Samba-3.0.23

There has been a minor update the Samba LDAP schema file. A substring matching rule has been added to the `sambaSID` attribute definition. For OpenLDAP servers, this will require the addition of `index sambaSID sub` to the `slapd.conf` configuration file. It will be necessary to execute the **slapindex** command after making this change. There has been no change to the actual data storage schema.

[2] <http://pdbsql.sourceforge.net/>

Chapter 10

NETWORK BROWSING

This chapter contains detailed information as well as a fast-track guide to implementing browsing across subnets and/or across workgroups (or domains). WINS is the best tool for resolution of NetBIOS names to IP addresses; however, WINS is not involved in browse list handling except by way of name-to-address resolution.

NOTE

What is WINS?

WINS is a facility that provides resolution of a NetBIOS name to its IP address. WINS is like a Dynamic-DNS service for NetBIOS networking names.

NOTE

MS Windows 2000 and later versions can be configured to operate with no NetBIOS over TCP/IP. Samba-3 and later versions also support this mode of operation. When the use of NetBIOS over TCP/IP has been disabled, the primary means for resolution of MS Windows machine names is via DNS and Active Directory. The following information assumes that your site is running NetBIOS over TCP/IP.

10.1 Features and Benefits

Charles Dickens once referred to the past in these words: "*It was the best of times, it was the worst of times.*" The more we look back, the more we long for what was and hope it never returns.

For many MS Windows network administrators, that statement sums up their feelings about NetBIOS networking precisely. For those who mastered NetBIOS networking, its fickle nature was just par for the course. For those who never quite managed to tame its lusty features, NetBIOS is like Paterson's Curse.

For those not familiar with botanical problems in Australia, Paterson's Curse, *Echium plantagineum*, was introduced to Australia from Europe during the mid-19th century. Since then it has spread rapidly. The high seed production, with densities of thousands of seeds per square meter, a seed longevity of more than 7 years, and an ability to germinate at any time of year, given the right conditions, are some of the features that make it such a persistent weed.

In this chapter we explore vital aspects of Server Message Block (SMB) networking with a particular focus on SMB as implemented through running NetBIOS (Network Basic Input/Output System) over TCP/IP. Since Samba does not implement SMB or NetBIOS over any other protocols, we need to know how to configure our network environment and simply remember to use nothing but TCP/IP on all our MS Windows network clients.

Samba provides the ability to implement a WINS (Windows Internetworking Name Server) and implements extensions to Microsoft's implementation of WINS. These extensions help Samba to effect stable WINS operations beyond the normal scope of MS WINS.

WINS is exclusively a service that applies only to those systems that run NetBIOS over TCP/IP. MS Windows 200x/XP have the capacity to operate with support for NetBIOS disabled, in which case WINS is of no relevance. Samba supports this also.

For those networks on which NetBIOS has been disabled (i.e., WINS is not required), the use of DNS is necessary for hostname resolution.

10.2 What Is Browsing?

To most people, browsing means they can see the MS Windows and Samba servers in the Network Neighborhood, and when the computer icon for a particular server is clicked, it opens up and shows the shares and printers available on the target server.

What seems so simple is in fact a complex interaction of different technologies. The technologies (or methods) employed in making all of this work include:

- MS Windows machines register their presence to the network.
- Machines announce themselves to other machines on the network.
- One or more machines on the network collate the local announcements.
- The client machine finds the machine that has the collated list of machines.
- The client machine is able to resolve the machine names to IP addresses.
- The client machine is able to connect to a target machine.

The Samba application that controls browse list management and name resolution is called `nmbd`. The configuration parameters involved in nmbd's operation are:

Browsing options:

- *os level*

- *lm announce*
- *lm interval*
- *preferred master*(*)
- *local master*(*)
- *domain master*(*)
- *browse list*
- *enhanced browsing*

Name Resolution Method:

- *name resolve order*(*)

WINS options:

- *dns proxy*
- *wins proxy*
- *wins server*(*)
- *wins support*(*)
- *wins hook*

Those marked with an (*) are the only options that commonly may need to be modified. Even if none of these parameters is set, `nmbd` will still do its job.

For Samba, the WINS Server and WINS Support are mutually exclusive options. When **nmbd** is started it will fail to execute if both options are set in the `smb.conf` file. The **nmbd** understands that when it spawns an instance of itself to run as a WINS server that it has to use its own WINS server also.

10.3 Discussion

All MS Windows networking uses SMB-based messaging. SMB messaging may be implemented with or without NetBIOS. MS Windows 200x supports NetBIOS over TCP/IP for backwards compatibility. Microsoft appears intent on phasing out NetBIOS support.

10.3.1 NetBIOS over TCP/IP

Samba implements NetBIOS, as does MS Windows NT/200x/XP, by encapsulating it over TCP/IP. NetBIOS-based networking uses broadcast messaging to effect browse list management. When running NetBIOS over TCP/IP, this uses UDP-based messaging. UDP messages can be broadcast or unicast.

Normally, only unicast UDP messaging can be forwarded by routers. The *remote announce* parameter to smb.conf helps to project browse announcements to remote network segments via unicast UDP. Similarly, the *remote browse sync* parameter of `smb.conf` implements browse list collation using unicast UDP.

The methods used by MS Windows to perform name lookup requests (name resolution) is determined by a configuration parameter called the NetBIOS node-type. There are four basic NetBIOS node types:

- *b-node (type 0x01):* The Windows client will use only NetBIOS broadcast requests using UDP broadcast.
- *p-node (type 0x02):* The Windows client will use point-to-point (NetBIOS unicast) requests using UDP unicast directed to a WINS server.
- *m-node (type 0x04):* The Windows client will first use NetBIOS broadcast requests using UDP broadcast, then it will use (NetBIOS unicast) requests using UDP unicast directed to a WINS server.
- *h-node (type 0x08):* The Windows client will use (NetBIOS unicast) requests using UDP unicast directed to a WINS server, then it will use NetBIOS broadcast requests using UDP broadcast.

The default Windows network client (or server) network configuration enables NetBIOS over TCP/IP and b-node configuration. The use of WINS makes most sense with h-node (hybrid mode) operation so that in the event of a WINS breakdown or non-availability, the client can use broadcast-based name resolution.

In those networks where Samba is the only SMB server technology, wherever possible `nmbd` should be configured on one machine as the WINS server. This makes it easy to manage the browsing environment. If each network segment is configured with its own Samba WINS server, then the only way to get cross-segment browsing to work is by using the *remote announce* and the *remote browse sync* parameters to your `smb.conf` file.

If only one WINS server is used for an entire multisegment network, then the use of the *remote announce* and the *remote browse sync* parameters should not be necessary.

As of Samba-3, WINS replication is being worked on. The bulk of the code has been committed, but it still needs maturation. This is not a supported feature of the Samba-3.0.20 release. Hopefully, this will become a supported feature of one of the Samba-3 release series. The delay is caused by the fact that this feature has not been of sufficient significance to inspire someone to pay a developer to complete it.

Right now Samba WINS does not support MS-WINS replication. This means that when setting up Samba as a WINS server, there must only be one `nmbd` configured as a WINS server on the network. Some sites have used multiple Samba WINS servers for redundancy (one server per subnet) and then used *remote browse sync* and *remote announce* to effect browse list collation across all segments. Note that this means clients will only resolve local names and must be configured to use DNS to resolve names on other subnets in order to resolve the IP addresses of the servers they can see on other subnets. This setup is not recommended but is mentioned as a practical consideration (i.e., an "if all else fails" scenario). NetBIOS over TCP/IP is an ugly and difficult to manage protocol. Its replacement, NetBIOSless SMB over TCP/IP is not without its own manageability concerns. NetBIOS based networking is a life of compromise and trade-offs. WINS stores information that cannot be stored in DNS; consequently, DNS is a poor substitute for WINS given that when NetBIOS over TCP/IP is used, Windows clients are designed to use WINS.

Lastly, take note that browse lists are a collection of unreliable broadcast messages that are repeated at intervals of not more than 15 minutes. This means that it will take time to establish a browse list, and it can take up to 45 minutes to stabilize, particularly across network segments.

When an MS Windows 200x/XP system attempts to resolve a host name to an IP address, it follows a defined path:

1. Checks the `hosts` file. It is located in `%SystemRoot%\System32\Drivers\etc`.
2. Does a DNS lookup.
3. Checks the NetBIOS name cache.
4. Queries the WINS server.

5. Does a broadcast name lookup over UDP.
6. Looks up entries in LMHOSTS, located in `%SystemRoot%\System32\Drivers\etc`.

Given the nature of how the NetBIOS over TCP/IP protocol is implemented, only WINS is capable of resolving with any reliability name lookups for service-oriented names such as TEMPTATION<1C> — a NetBIOS name query that seeks to find network logon servers. DNS has no concept of service-oriented names such as this. In fact, the Microsoft ADS implementation specifically manages a whole range of extended service-oriented DNS entries. This type of facility is not implemented and is not supported for the NetBIOS over TCP/IP protocol namespace.

10.3.2 TCP/IP without NetBIOS

All TCP/IP-enabled systems use various forms of hostname resolution. The primary methods for TCP/IP hostname resolution involve either a static file (`/etc/hosts`) or the Domain Name System (DNS). DNS is the technology that makes the Internet usable. DNS-based hostname resolution is supported by nearly all TCP/IP-enabled systems. Only a few embedded TCP/IP systems do not support DNS.

Windows 200x/XP can register its hostname with a Dynamic DNS server (DDNS). It is possible to force register with a dynamic DNS server in Windows 200x/XP using **ipconfig /registerdns**.

With Active Directory, a correctly functioning DNS server is absolutely essential. In the absence of a working DNS server that has been correctly configured, MS Windows clients and servers will be unable to locate each other, so network services consequently will be severely impaired.

Use of raw SMB over TCP/IP (No NetBIOS layer) can be done only with Active Directory domains. Samba is not an Active Directory domain controller: ergo, it is not possible to run Samba as a domain controller and at the same time *not* use NetBIOS. Where Samba is used as an Active Directory domain member server (DMS) it is possible to configure Samba to not use NetBIOS over TCP/IP. A Samba DMS can integrate fully into an Active Directory domain, however, if NetBIOS over TCP/IP is disabled, it is necessary to manually create appropriate DNS entries for the Samba DMS because they will not be automatically generated either by Samba, or by the ADS environment.

10.3.3 DNS and Active Directory

Occasionally we hear from UNIX network administrators who want to use a UNIX-based DDNS server in place of the Microsoft DNS server. While this might be desirable to some, the MS Windows 200x DNS server is autoconfigured to work with Active Directory. It is possible to use BIND version 8 or 9, but it will almost certainly be necessary to create service records (SRV records) so MS Active Directory clients can resolve hostnames to locate essential network services. The following are some of the default service records that Active Directory requires:

The use of DDNS is highly recommended with Active Directory, in which case the use of BIND9 is preferred for its ability to adequately support the SRV (service) records that are needed for Active Directory. Of course, when running ADS, it makes sense to use Microsoft's own DDNS server because of the natural affinity between ADS and MS DNS.

_ldap._tcp.pdc._msdcs.*Domain* This provides the address of the Windows NT PDC for the domain.

_ldap._tcp.pdc._msdcs.*DomainTree* Resolves the addresses of global catalog servers in the domain.

_ldap._tcp.*site*.sites.writable._msdcs.*Domain* Provides list of domain controllers based on sites.

_ldap._tcp.writable._msdcs.*Domain* Enumerates list of domain controllers that have the writable copies of the Active Directory data store.

_ldap._tcp.*GUID*.domains._msdcs.*DomainTree* Entry used by MS Windows clients to locate machines using the global unique identifier.

_ldap._tcp.*Site*.gc._msdcs.*DomainTree* Used by Microsoft Windows clients to locate the site configuration-dependent global catalog server.

Specific entries used by Microsoft clients to locate essential services for an example domain called `quenya.org` include:

- _kerberos._udp.quenya.org — Used to contact the KDC server via UDP. This entry must list port 88 for each KDC.
- _kpasswd._udp.quenya.org — Used to locate the `kpasswd` server when a user password change must be processed. This record must list port 464 on the master KDC.
- _kerberos._tcp.quenya.org — Used to locate the KDC server via TCP. This entry must list port 88 for each KDC.
- _ldap._tcp.quenya.org — Used to locate the LDAP service on the PDC. This record must list port 389 for the PDC.
- _kpasswd._tcp.quenya.org — Used to locate the `kpasswd` server to permit user password changes to be processed. This must list port 464.
- _gc._tcp.quenya.org — Used to locate the global catalog server for the top of the domain. This must list port 3268.

The following records are also used by the Windows domain member client to locate vital services on the Windows ADS domain controllers.

- _ldap._tcp.pdc._msdcs.quenya.org
- _ldap.gc._msdcs.quenya.org
- _ldap.default-first-site-name._sites.gc._msdcs.quenya.org
- _ldap.{SecID}.domains._msdcs.quenya.org
- _ldap._tcp.dc._msdcs.quenya.org
- _kerberos._tcp.dc._msdcs.quenya.org
- _ldap.default-first-site-name._sites.dc._msdcs.quenya.org
- _kerberos.default-first-site-name._sites.dc._msdcs.queyna.org
- SecID._msdcs.quenya.org

Presence of the correct DNS entries can be validated by executing:

```
root#  dig @frodo -t any _ldap._tcp.dc._msdcs.quenya.org

; <lt;>> DiG 9.2.2 <lt;>> @frodo -t any _ldap._tcp.dc._msdcs.quenya.org
;; global options:  printcmd
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3072
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 2


;; QUESTION SECTION:
;_ldap._tcp.dc._msdcs.quenya.org. IN        ANY


;; ANSWER SECTION:
_ldap._tcp.dc._msdcs.quenya.org. 600 IN SRV 0 100 389 frodo.quenya.org.
_ldap._tcp.dc._msdcs.quenya.org. 600 IN SRV 0 100 389 noldor.quenya.org.


;; ADDITIONAL SECTION:
frodo.quenya.org.  3600  IN      A       10.1.1.16
noldor.quenya.org. 1200  IN      A       10.1.1.17


;; Query time: 0 msec
;; SERVER: frodo#53(10.1.1.16)
;; WHEN: Wed Oct  7 14:39:31 2004
;; MSG SIZE  rcvd: 171
```

10.4 How Browsing Functions

MS Windows machines register their NetBIOS names (i.e., the machine name for each service type in operation) on startup. The exact method by which this name registration takes place is determined by whether or not the MS Windows client/server has been given a WINS server address, whether or not LMHOSTS lookup is enabled, whether or not DNS for NetBIOS name resolution is enabled, and so on.

In the case where there is no WINS server, all name registrations as well as name lookups are done by UDP broadcast. This isolates name resolution to the local subnet, unless LMHOSTS is used to list all names and IP addresses. In such situations, Samba provides a means by which the Samba server name may be forcibly injected into the browse list of a remote MS Windows network (using the *remote announce* parameter).

Where a WINS server is used, the MS Windows client will use UDP unicast to register with the WINS server. Such packets can be routed, and thus WINS allows name resolution to function across routed networks.

During the startup process, an election takes place to create a local master browser (LMB) if one does not already exist. On each NetBIOS network one machine will be elected to function as the domain master browser (DMB). This domain browsing has nothing to do with MS security Domain Control. Instead, the DMB serves the role of contacting each LMB (found by asking WINS or from LMHOSTS) and exchanging browse list contents. This way every master browser will eventually obtain a complete list of all machines that are on the network. Every 11 to 15 minutes an election is held to determine which machine will be the master browser. By the nature of the election criteria used, the machine with the highest uptime, or the most senior protocol version or other criteria, will win the election as DMB.

Where a WINS server is used, the DMB registers its IP address with the WINS server using the name of the domain and the NetBIOS name type 1B (e.g., DOMAIN<1B>). All LMBs register their IP addresses with the WINS server, also with the name of the domain and the NetBIOS name type of 1D. The 1B name is unique to one server within the domain security context, and only one 1D name is registered for each network segment. Machines that have registered the 1D name will be authoritive browse list maintainers for the network segment they are on. The DMB is responsible for synchronizing the browse lists it obtains from the LMBs.

Clients wishing to browse the network make use of this list but also depend on the availability of correct name resolution to the respective IP address or addresses.

Any configuration that breaks name resolution and/or browsing intrinsics will annoy users because they will have to put up with protracted inability to use the network services.

Samba supports a feature that allows forced synchronization of browse lists across routed networks using the *remote browse sync* parameter in the `smb.conf` file. This causes Samba to contact the LMB on a remote network and to request browse list synchronization. This effectively bridges two networks that are separated by routers. The two remote networks may use either broadcast-based name resolution or WINS-based name resolution, but it should be noted that the *remote browse sync* parameter provides browse list synchronization — and that is distinct from name-to-address resolution.

In other words, for cross-subnet browsing to function correctly, it is essential that a name-to-address resolution mechanism be provided. This mechanism could be via DNS, `/etc/hosts`, and so on.

10.4.1 Configuring Workgroup Browsing

To configure cross-subnet browsing on a network containing machines in a workgroup, not an NT domain, you need to set up one Samba server to be the DMB (note that this is not the same as a Primary Domain Controller, although in an NT domain the same machine plays both roles). The role of a DMB is to collate the browse lists from LMB on all the subnets that have a machine participating in the workgroup. Without one machine configured as a DMB, each subnet would be an isolated workgroup unable to see any machines on another subnet. It is the presence of a DMB that makes cross-subnet browsing possible for a workgroup.

In a workgroup environment the DMB must be a Samba server, and there must only be one DMB per workgroup name. To set up a Samba server as a DMB, set the following option in the *[global]* section of the `smb.conf` file:

```
        domain master = yes
```

The DMB should preferably be the LMB for its own subnet. In order to achieve this, set the following options in the *[global]* section of the `smb.conf` file as shown in Example 10.4.1

Example 10.4.1 Domain Master Browser smb.conf

```
[global]
        domain master = yes
        local master = yes
        preferred master = yes
        os level = 65
```

The DMB may be the same machine as the WINS server, if necessary.

Next, you should ensure that each of the subnets contains a machine that can act as an LMB for the workgroup. Any MS Windows NT/200x/XP machine should be able to do this, as will Windows 9x/Me machines (although these tend to get rebooted more often, so it is not such a good idea to use them).

Example 10.4.2 Local master browser smb.conf

```
[global]
        domain master = no
        local master = yes
        preferred master = yes
        os level = 65
```

To make a Samba server an LMB, set the following options in the *[global]* section of the `smb.conf` file as shown in Example 10.4.2

Do not do this for more than one Samba server on each subnet, or they will war with each other over which is to be the LMB.

The *local master* parameter allows Samba to act as a LMB. The *preferred master* causes **nmbd** to force a browser election on startup and the *os level* parameter sets Samba high enough so it should win any browser elections.

If you have an NT machine on the subnet that you wish to be the LMB, you can disable Samba from becoming an LMB by setting the following options in the *[global]* section of the `smb.conf` file as shown in Example 10.4.3.

Example 10.4.3 smb.conf for Not Being a Master Browser

```
[global]
        domain master = no
        local master = no
        preferred master = no
        os level = 0
```

10.4.2 Domain Browsing Configuration

If you are adding Samba servers to a Windows NT domain, then you must not set up a Samba server as a DMB. By default, a Windows NT PDC for a domain is also the DMB for that domain. Network browsing may break if a Samba server other than the PDC registers the DMB NetBIOS name (*DOMAIN*<1B>) with WINS.

For subnets other than the one containing the Windows NT PDC, you may set up Samba servers as LMBs as described. To make a Samba server a

Local Master Browser, set the following options in the *[global]* section of the `smb.conf` file as shown in Example 10.4.4

Example 10.4.4 Local Master Browser smb.conf

```
[global]
        domain master = no
        local master = yes
        preferred master = yes
        os level = 65
```

If you wish to have a Samba server fight the election with machines on the same subnet, you may set the *os level* parameter to lower levels. By doing this you can tune the order of machines that will become LMBs if they are running. For more details on this, refer to Section 10.4.3.

If you have Windows NT machines that are members of the domain on all subnets and you are sure they will always be running, you can disable Samba from taking part in browser elections and ever becoming an LMB by setting the following options in the *[global]* section of the `smb.conf` file as shown in Example 10.4.5

Example 10.4.5 smb.conf for Not Being a master browser

[global]domain master = no*local master* = no*preferred master* = no*os level* = 0

10.4.3 Forcing Samba to Be the Master

Who becomes the master browser is determined by an election process using broadcasts. Each election packet contains a number of parameters that determine what precedence (bias) a host should have in the election. By default Samba uses a low precedence and thus loses elections to just about every Windows network server or client.

If you want Samba to win elections, set the *os level* global option in `smb.conf` to a higher number. It defaults to 20. Using 34 would make it win all elections over every other system (except other Samba systems).

An *os level* of two would make it beat Windows for Workgroups and Windows 9x/Me, but not MS Windows NT/200x Server. An MS Windows

NT/200x Server domain controller uses level 32. The maximum os level is 255.

If you want Samba to force an election on startup, set the *preferred master* global option in `smb.conf` to `yes`. Samba will then have a slight advantage over other potential master browsers that are not preferred master browsers. Use this parameter with care, because if you have two hosts (whether they are Windows 9x/Me or NT/200x/XP or Samba) on the same local subnet both set with *preferred master* to `yes`, then periodically and continually they will force an election in order to become the LMB.

If you want Samba to be a *DMB*, then it is recommended that you also set *preferred master* to `yes`, because Samba will not become a DMB for the whole of your LAN or WAN if it is not also a LMB on its own broadcast isolated subnet.

It is possible to configure two Samba servers to attempt to become the DMB for a domain. The first server that comes up will be the DMB. All other Samba servers will attempt to become the DMB every 5 minutes. They will find that another Samba server is already the DMB and will fail. This provides automatic redundancy should the current DMB fail. The network bandwidth overhead of browser elections is relatively small, requiring approximately four UDP packets per machine per election. The maximum size of a UDP packet is 576 bytes.

10.4.4 Making Samba the Domain Master

The domain master browser is responsible for collating the browse lists of multiple subnets so browsing can occur between subnets. You can make Samba act as the domain master browser by setting *domain master* = yes in `smb.conf`. By default it will not be a domain master browser.

Do not set Samba to be the domain master for a workgroup that has the same name as an NT/200x domain. If Samba is configured to be the domain master for a workgroup that is present on the same network as a Windows NT/200x domain that has the same name, network browsing problems will certainly be experienced.

When Samba is the domain master and the master browser, it will listen for master announcements (made roughly every 12 minutes) from LMBs on other subnets and then contact them to synchronize browse lists.

If you want Samba to be the domain master, you should also set the *os level* high enough to make sure it wins elections, and set *preferred master* to `yes`, to get Samba to force an election on startup.

All servers (including Samba) and clients should be using a WINS server to resolve NetBIOS names. If your clients are only using broadcasting to resolve NetBIOS names, then two things will occur:

1. LMBs will be unable to find a DMB because they will be looking only on the local subnet.
2. If a client happens to get hold of a domain-wide browse list and a user attempts to access a host in that list, it will be unable to resolve the NetBIOS name of that host.

If, however, both Samba and your clients are using a WINS server, then:

1. LMBs will contact the WINS server and, as long as Samba has registered that it is a DMB with the WINS server, the LMB will receive Samba's IP address as its DMB.
2. When a client receives a domain-wide browse list and a user attempts to access a host in that list, it will contact the WINS server to resolve the NetBIOS name of that host. As long as that host has registered its NetBIOS name with the same WINS server, the user will be able to see that host..

10.4.5 Note about Broadcast Addresses

If your network uses a zero-based broadcast address (for example, if it ends in a 0), then you will strike problems. Windows for Workgroups does not seem to support a zeros broadcast, and you will probably find that browsing and name lookups will not work.

10.4.6 Multiple Interfaces

Samba supports machines with multiple network interfaces. If you have multiple interfaces, you will need to use the *interfaces* option in `smb.conf` to configure them. For example, the machine you are working with has 4 network interfaces; `eth0`, `eth1`, `eth2`, `eth3` and only interfaces `eth1` and `eth4` should be used by Samba. In this case, the following `smb.conf` file entries would permit that intent:

```
        interfaces = eth1, eth4
        bind interfaces only = Yes
```

The *bind interfaces only* = Yes is necessary to exclude TCP/IP session services (ports 135, 139, and 445) over the interfaces that are not specified. Please be aware that **nmbd** will listen for incoming UDP port 137 packets on the unlisted interfaces, but it will not answer them. It will, however, send its broadcast packets over the unlisted interfaces. Total isolation of ethernet interface requires the use of a firewall to block ports 137 and 138 (UDP), and ports 135, 139, and 445 (TCP) on all network interfaces that must not be able to access the Samba server.

10.4.7 Use of the Remote Announce Parameter

The *remote announce* parameter of `smb.conf` can be used to forcibly ensure that all the NetBIOS names on a network get announced to a remote network. The syntax of the *remote announce* parameter is:

```
        remote announce = 192.168.12.23 [172.16.21.255]  ↩
           ...
```

or

```
        remote announce = 192.168.12.23/MIDEARTH  ↩
           [172.16.21.255/ELVINDORF]  ...
```

where:

`192.168.12.23` **and** *`172.16.21.255`* is either the LMB IP address or the broadcast address of the remote network. That is, the LMB is at 192.168.1.23, or the address could be given as 172.16.21.255 where the netmask is assumed to be 24 bits (255.255.255.0). When the remote announcement is made to the broadcast address of the remote network, every host will receive our announcements. This is noisy and therefore undesirable but may be necessary if we do not know the IP address of the remote LMB.

`WORKGROUP` is optional and can be either our own workgroup or that of the remote network. If you use the workgroup name of the remote network,

our NetBIOS machine names will end up looking like they belong to that workgroup. This may cause name resolution problems and should be avoided.

10.4.8 Use of the Remote Browse Sync Parameter

The *remote browse sync* parameter of `smb.conf` is used to announce to another LMB that it must synchronize its NetBIOS name list with our Samba LMB. This works only if the Samba server that has this option is simultaneously the LMB on its network segment.

The syntax of the *remote browse sync* parameter is:

```
        remote browse sync
```

where *192.168.10.40* is either the IP address of the remote LMB or the network broadcast address of the remote segment.

10.5 WINS: The Windows Internetworking Name Server

Use of WINS (either Samba WINS or MS Windows NT Server WINS) is highly recommended. Every NetBIOS machine registers its name together with a name_type value for each of several types of service it has available. It registers its name directly as a unique (the type 0x03) name. It also registers its name if it is running the LanManager-compatible server service (used to make shares and printers available to other users) by registering the server (the type 0x20) name.

All NetBIOS names are up to 15 characters in length. The name_type variable is added to the end of the name, thus creating a 16 character name. Any name that is shorter than 15 characters is padded with spaces to the 15th character. Thus, all NetBIOS names are 16 characters long (including the name_type information).

WINS can store these 16-character names as they get registered. A client that wants to log onto the network can ask the WINS server for a list of all names that have registered the NetLogon service name_type. This saves broadcast traffic and greatly expedites logon processing. Since broadcast name resolution cannot be used across network segments, this type of information can only be provided via WINS or via a statically configured `lmhosts` file that must reside on all clients in the absence of WINS.

WINS also forces browse list synchronization by all LMBs. LMBs must synchronize their browse list with the DMB, and WINS helps the LMB to identify its DMB. By definition this will work only within a single workgroup. Note that the DMB has nothing to do with what is referred to as an MS Windows NT domain. The latter is a reference to a security environment, while the DMB refers to the master controller for browse list information only.

WINS will work correctly only if every client TCP/IP protocol stack is configured to use the WINS servers. Any client that is not configured to use the WINS server will continue to use only broadcast-based name registration, so WINS may never get to know about it. In any case, machines that have not registered with a WINS server will fail name-to-address lookup attempts by other clients and will therefore cause workstation access errors.

To configure Samba as a WINS server, just add *wins support* = yes to the `smb.conf` file [global] section.

To configure Samba to register with a WINS server, just add *wins server* = 10.0.0.18 to your `smb.conf` file *[global]* section.

IMPORTANT

Never use *wins support* = yes together with *wins server* = 10.0.0.18 particularly not using its own IP address. Specifying both will cause nmbd to refuse to start!

10.5.1 WINS Server Configuration

Either a Samba server or a Windows NT server machine may be set up as a WINS server. To configure a Samba server to be a WINS server, you must add to the `smb.conf` file on the selected Server the following line to the *[global]* section:

```
        wins support = yes
```

Versions of Samba prior to 1.9.17 had this parameter default to yes. If you have any older versions of Samba on your network, it is strongly suggested

you upgrade to a recent version, or at the very least set the parameter to "no" on all these machines.

Machines configured with *wins support* = yes will keep a list of all NetBIOS names registered with them, acting as a DNS for NetBIOS names.

It is strongly recommended to set up only one WINS server. Do not set the *wins support* = yes option on more than one Samba server on a network.

To configure Windows NT/200x Server as a WINS server, install and configure the WINS service. See the Windows NT/200x documentation for details. Windows NT/200x WINS servers can replicate to each other, allowing more than one to be set up in a complex subnet environment. Because Microsoft refuses to document the replication protocols, Samba cannot currently participate in these replications. It is possible that a Samba-to-Samba WINS replication protocol may be defined in the future, in which case more than one Samba machine could be set up as a WINS server. Currently only one Samba server should have the *wins support* = yes parameter set.

After the WINS server has been configured, you must ensure that all machines participating on the network are configured with the address of this WINS server. If your WINS server is a Samba machine, fill in the Samba machine IP address in the **Primary WINS Server** field of the **Control Panel->Network->Protocols->TCP->WINS Server** dialogs in Windows 9x/Me or Windows NT/200x. To tell a Samba server the IP address of the WINS server, add the following line to the *[global]* section of all `smb.conf` files:

```
        wins server = <name or IP address>
```

where <name or IP address> is either the DNS name of the WINS server machine or its IP address.

This line must not be set in the `smb.conf` file of the Samba server acting as the WINS server itself. If you set both the *wins support* = yes option and the *wins server* = <name> option then **nmbd** will fail to start.

There are two possible scenarios for setting up cross-subnet browsing. The first details setting up cross-subnet browsing on a network containing Windows 9x/Me, Samba, and Windows NT/200x machines that are not configured as part of a Windows NT domain. The second details setting up cross-subnet browsing on networks that contain NT domains.

10.5.2 WINS Replication

Samba-3 does not support native WINS replication. There was an approach to implement it, called `wrepld`, but it was never ready for action and the development is now discontinued.

Meanwhile, there is a project named `samba4WINS`, which makes it possible to run the Samba-4 WINS server parallel to Samba-3 since version 3.0.21. More information about `samba4WINS` are available at http://ftp.sernet.de/pub/samba4WINS.

10.5.3 Static WINS Entries

Adding static entries to your Samba WINS server is actually fairly easy. All you have to do is add a line to `wins.dat`, typically located in `/usr/local/samba/var/locks` or `/var/run/samba`.

Entries in `wins.dat` take the form of:

```
"NAME#TYPE" TTL ADDRESS+ FLAGS
```

where NAME is the NetBIOS name, TYPE is the NetBIOS type, TTL is the time-to-live as an absolute time in seconds, ADDRESS+ is one or more addresses corresponding to the registration, and FLAGS are the NetBIOS flags for the registration.

> **Note**
>
> A change that has been made to the `wins.dat` will not take effect until nmbd has been restarted. It should be noted that since the `wins.dat` file changes dynamically, nmbd should be stopped before editting this file. Do not forget to restart nmbd when this file has been editted.

A typical dynamic entry looks like this:

```
"MADMAN#03" 1155298378 192.168.1.2 66R
```

To make a NetBIOS name static (permanent), simply set the TTL to 0, like this:

```
"MADMAN#03" 0 192.168.1.2 66R
```

The NetBIOS flags may be interpreted as additive hexadecimal values: 00 - Broadcast node registration, 20 - Peer node registration, 40 - Meta node registration, 60 - Hybrid node registration, 02 - Permanent name, 04 - Active name, 80 - Group name. The 'R' indicates this is a registration record. Thus 66R means: Hybrid node active and permanent NetBIOS name. These values may be found in the `nameserv.h` header file from the Samba source code repository. These are the values for the NB flags.

Though this method works with early Samba-3 versions, there is a possibility that it may change in future versions if WINS replication is added.

10.6 Helpful Hints

The following hints should be carefully considered because they are stumbling points for many new network administrators.

10.6.1 Windows Networking Protocols

A common cause of browsing problems results from the installation of more than one protocol on an MS Windows machine.

WARNING

Do not use more than one protocol on MS Windows clients.

Every NetBIOS machine takes part in a process of electing the LMB (and DMB) every 15 minutes. A set of election criteria is used to determine the order of precedence for winning this election process. A machine running Samba or Windows NT will be biased, so the most suitable machine will predictably win and thus retain its role.

The election process is *fought out, so to speak* over every NetBIOS network interface. In the case of a Windows 9x/Me machine that has both TCP/IP and IPX installed and has NetBIOS enabled over both protocols, the election will be decided over both protocols. As often happens, if the Windows 9x/Me machine is the only one with both protocols, then the LMB may be won on the NetBIOS interface over the IPX protocol. Samba will then lose the LMB role because Windows 9x/Me will insist it knows who the LMB is. Samba will then cease to function as an LMB, and browse list operation on all TCP/IP-only machines will therefore fail.

Windows 95, 98, 98se, and Me are referred to generically as Windows 9x/Me. The Windows NT4, 200x, and XP use common protocols. These are roughly referred to as the Windows NT family, but it should be recognized that 2000 and XP/2003 introduce new protocol extensions that cause them to behave differently from MS Windows NT4. Generally, where a server does not support the newer or extended protocol, these will fall back to the NT4 protocols.

The safest rule of all to follow is: Use only one protocol!

10.6.2 Name Resolution Order

Resolution of NetBIOS names to IP addresses can take place using a number of methods. The only ones that can provide NetBIOS name_type information are:

- WINS — the best tool.
- LMHOSTS — static and hard to maintain.
- Broadcast — uses UDP and cannot resolve names across remote segments.

Alternative means of name resolution include:

- Static `/etc/hosts` — hard to maintain and lacks name_type info.

- DNS — is a good choice but lacks essential NetBIOS name_type information.

Many sites want to restrict DNS lookups and avoid broadcast name resolution traffic. The *name resolve order* parameter is of great help here. The syntax of the *name resolve order* parameter is:

```
        name resolve order = wins lmhosts bcast host
```

or

```
        name resolve order = wins lmhosts       ( ↩
           eliminates bcast and host)
```

The default is:

```
        name resolve order = host lmhost wins bcast
,
```

where "host" refers to the native methods used by the UNIX system to implement the gethostbyname() function call. This is normally controlled by `/etc/host.conf`, `/etc/nsswitch.conf` and `/etc/resolv.conf`.

10.7 Technical Overview of Browsing

SMB networking provides a mechanism by which clients can access a list of machines in a network called *browse list*. This list contains machines that are ready to offer file and/or print services to other machines within the network. It therefore does not include machines that aren't currently able to do server tasks. The browse list is heavily used by all SMB clients. Configuration of SMB browsing has been problematic for some Samba users, hence this document.

MS Windows 2000 and later versions, as with Samba-3 and later versions, can be configured to not use NetBIOS over TCP/IP. When configured this way, it is imperative that name resolution (using DNS/LDAP/ADS) be correctly configured and operative. Browsing will not work if name resolution from SMB machine names to IP addresses does not function correctly.

Where NetBIOS over TCP/IP is enabled, use of a WINS server is highly recommended to aid the resolution of NetBIOS (SMB) names to IP ad-

dresses. WINS allows remote segment clients to obtain NetBIOS name_type information that cannot be provided by any other means of name resolution.

10.7.1 Browsing Support in Samba

Samba facilitates browsing. The browsing is supported by nmbd and is also controlled by options in the `smb.conf` file. Samba can act as an LMB for a workgroup, and the ability to support domain logons and scripts is now available.

Samba can also act as a DMB for a workgroup. This means that it will collate lists from LMBs into a wide-area network server list. In order for browse clients to resolve the names they may find in this list, it is recommended that both Samba and your clients use a WINS server.

Do not set Samba to be the domain master for a workgroup that has the same name as an NT Domain. On each wide-area network, you must only ever have one DMB per workgroup, regardless of whether it is NT, Samba, or any other type of domain master that is providing this service.

NOTE

nmbd can be configured as a WINS server, but it is not necessary to specifically use Samba as your WINS server. MS Windows NT4, Server or Advanced Server 200x can be configured as your WINS server. In a mixed NT/200x server and Samba environment on a WAN, it is recommended that you use the Microsoft WINS server capabilities. In a Samba-only environment, it is recommended that you use one and only one Samba server as the WINS server.

To get browsing to work, you need to run **nmbd** as usual, but must use the *workgroup* option in `smb.conf` to control what workgroup Samba becomes a part of.

Samba also has a useful option for a Samba server to offer itself for browsing on another subnet. It is recommended that this option is used only for

"unusual" purposes: announcements over the Internet, for example. See *remote announce* in the `smb.conf` man page.

10.7.2 Problem Resolution

If something does not work, the `log.nmbd` file will help to track down the problem. Try a *log level* of 2 or 3 for finding problems. Also note that the current browse list usually gets stored in text form in a file called `browse.dat`.

If it does not work, you should still be able to type the server name as `\\SERVER` in **filemanager**, then press enter, and **filemanager** should display the list of available shares.

Some people find browsing fails because they do not have the global *guest account* set to a valid account. Remember that the IPC$ connection that lists the shares is done as guest and so you must have a valid guest account.

NOTE

The `IPC$` share is used by all SMB/CIFS clients to obtain the list of resources that is available on the server. This is the source of the list of shares and printers when browsing an SMB/CIFS server (also Windows machines) using the Windows Explorer to browse resources through the Windows Network Neighborhood (also called My Network Places) through to a Windows server. At this point, the client has opened a connection to the `\\server\IPC4` resource. Clicking on a share will then open up a connection to the `\\server\share`.

MS Windows 2000 and later (as with Samba) can be configured to disallow anonymous (i.e., guest account) access to the IPC$ share. In that case, the MS Windows 2000/XP/2003 machine acting as an SMB/CIFS client will use the name of the currently logged-in user to query the IPC$ share. MS Windows 9x/Me clients are not able to do this and thus will not be able to browse server resources.

The other big problem people have is that their broadcast address, netmask, or IP address is wrong (specified with the *interfaces* option in `smb.conf`)

10.7.3 Cross-Subnet Browsing

Since the release of Samba 1.9.17 (alpha1), Samba has supported the replication of browse lists across subnet boundaries. This section describes how to set this feature up in different settings.

To see browse lists that span TCP/IP subnets (i.e., networks separated by routers that do not pass broadcast traffic), you must set up at least one WINS server. The WINS server acts as a DNS for NetBIOS names. This will allow NetBIOS name-to-IP address translation to be completed by a direct query of the WINS server. This is done via a directed UDP packet on port 137 to the WINS server machine. The WINS server avoids the necessity of default NetBIOS name-to-IP address translation, which is done using UDP broadcasts from the querying machine. This means that machines on one subnet will not be able to resolve the names of machines on another subnet without using a WINS server. The Samba hacks, *`remote browse sync`*, and *`remote announce`* are designed to get around the natural limitations that prevent UDP broadcast propagation. The hacks are not a universal solution and they should not be used in place of WINS, they are considered last resort methods.

Remember, for browsing across subnets to work correctly, all machines, be they Windows 95, Windows NT, or Samba servers, must have the IP address of a WINS server given to them by a DHCP server or by manual configuration: for Windows 9x/Me and Windows NT/200x/XP, this is in the TCP/IP Properties, under Network settings; for Samba, this is in the `smb.conf` file.

It is possible to operate Samba-3 without NetBIOS over TCP/IP. If you do this, be warned that if used outside of MS ADS, this will forgo network browsing support. ADS permits network browsing support through DNS, providing appropriate DNS records are inserted for all Samba servers.

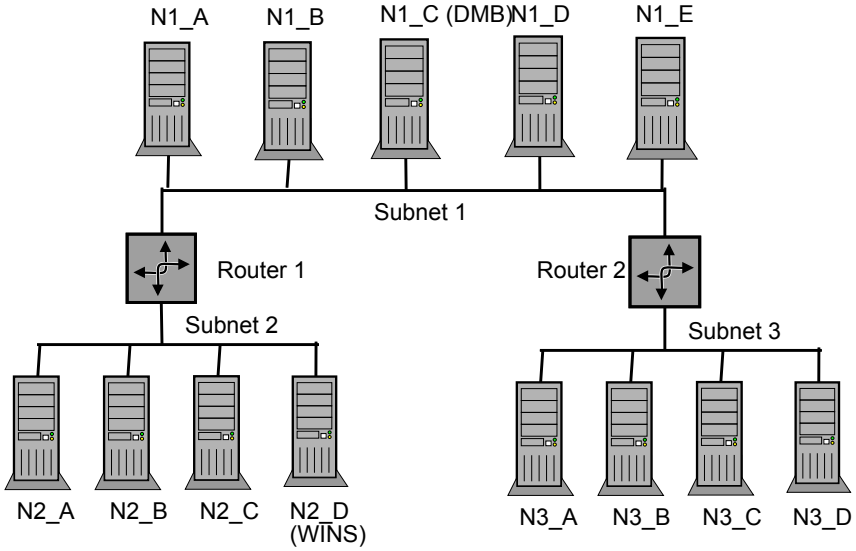
10.7.3.1 Behavior of Cross-Subnet Browsing

Cross-subnet browsing is a complicated dance, containing multiple moving parts. It has taken Microsoft several years to get the code that correctly

achieves this, and Samba lags behind in some areas. Samba is capable of cross-subnet browsing when configured correctly.

Consider a network set up as in Figure 10.1.

Figure 10.1 Cross-Subnet Browsing Example.



This consists of three subnets (1, 2, 3) connected by two routers (R1, R2), which do not pass broadcasts. Subnet 1 has five machines on it, subnet 2 has four machines, and subnet 3 has four machines. Assume for the moment that all machines are configured to be in the same workgroup (for simplicity's sake). Machine N1_C on subnet 1 is configured as the DMB (i.e., it will collate the browse lists for the workgroup). Machine N2_D is configured as a WINS server, and all the other machines are configured to register their NetBIOS names with it.

As these machines are booted up, elections for master browsers take place on each of the three subnets. Assume that machine N1_C wins on subnet 1, N2_B wins on subnet 2, and N3_D wins on subnet 3. These machines are known as LMBs for their particular subnet. N1_C has an advantage in winning as the LMB on subnet 1 because it is set up as DMB.

On each of the three networks, machines that are configured to offer sharing services will broadcast that they are offering these services. The LMB on each subnet will receive these broadcasts and keep a record of the fact that the machine is offering a service. This list of records is the basis of the

browse list. For this case, assume that all the machines are configured to offer services, so all machines will be on the browse list.

For each network, the LMB on that network is considered *authoritative* for all the names it receives via local broadcast. This is because a machine seen by the LMB via a local broadcast must be on the same network as the Local Master Browser and thus is a *trusted* and *verifiable* resource. Machines on other networks that the LMBs learn about when collating their browse lists have not been directly seen. These records are called *non-authoritative.*

At this point the browse lists appear as shown in Table 10.1 (these are the machines you would see in your network neighborhood if you looked in it on a particular network right now).

Table 10.1 Browse Subnet Example 1

Subnet	**Browse Master**	**List**
Subnet1	N1_C	N1_A, N1_B, N1_C, N1_D, N1_E
Subnet2	N2_B	N2_A, N2_B, N2_C, N2_D
Subnet3	N3_D	N3_A, N3_B, N3_C, N3_D

At this point all the subnets are separate, and no machine is seen across any of the subnets.

Now examine subnet 2 in Table 10.2. As soon as N2_B has become the LMB, it looks for a DMB with which to synchronize its browse list. It does this by querying the WINS server (N2_D) for the IP address associated with the NetBIOS name WORKGROUP<1B>. This name was registered by the DMB (N1_C) with the WINS server as soon as it was started.

Once N2_B knows the address of the DMB, it tells it that is the LMB for subnet 2 by sending a *MasterAnnouncement* packet as a UDP port 138 packet. It then synchronizes with it by doing a *NetServerEnum2* call. This tells the DMB to send it all the server names it knows about. Once the DMB receives the *MasterAnnouncement* packet, it schedules a synchronization request to the sender of that packet. After both synchronizations are complete, the browse lists look like those in Table 10.2

Servers with an (*) after them are non-authoritative names.

At this point users looking in their Network Neighborhood on subnets 1 or 2 will see all the servers on both; users on subnet 3 will still see only the servers on their own subnet.

Table 10.2 Browse Subnet Example 2

Subnet	Browse Master	List
Subnet1	N1_C	N1_A, N1_B, N1_C, N1_D, N1_E, N2_A(*), N2_B(*), N2_C(*), N2_D(*)
Subnet2	N2_B	N2_A, N2_B, N2_C, N2_D, N1_A(*), N1_B(*), N1_C(*), N1_D(*), N1_E(*)
Subnet3	N3_D	N3_A, N3_B, N3_C, N3_D

The same sequence of events that occurred for N2_B now occurs for the LMB on subnet 3 (N3_D). When it synchronizes browse lists with the DMB (N1_A) it gets both the server entries on subnet 1 and those on subnet 2. After N3_D has synchronized with N1_C and vica versa, the browse lists will appear as shown in Table 10.3

Table 10.3 Browse Subnet Example 3

Subnet	Browse Master	List
Subnet1	N1_C	N1_A, N1_B, N1_C, N1_D, N1_E, N2_A(*), N2_B(*), N2_C(*), N2_D(*), N3_A(*), N3_B(*), N3_C(*), N3_D(*)
Subnet2	N2_B	N2_A, N2_B, N2_C, N2_D, N1_A(*), N1_B(*), N1_C(*), N1_D(*), N1_E(*)
Subnet3	N3_D	N3_A, N3_B, N3_C, N3_D, N1_A(*), N1_B(*), N1_C(*), N1_D(*), N1_E(*), N2_A(*), N2_B(*), N2_C(*), N2_D(*)

Servers with an (*) after them are non-authoritative names.

At this point, users looking in their Network Neighborhood on subnets 1 or 3 will see all the servers on all subnets, while users on subnet 2 will still see only the servers on subnets 1 and 2, but not 3.

Finally, the LMB for subnet 2 (N2_B) will sync again with the DMB (N1_C) and will receive the missing server entries. Finally, as when a steady state (if no machines are removed or shut off) has been achieved, the browse lists will appear as shown in Table 10.4.

Servers with an (*) after them are non-authoritative names.

Synchronizations between the DMB and LMBs will continue to occur, but this should remain a steady-state operation.

Table 10.4 Browse Subnet Example 4

Subnet	Browse Master	List
Subnet1	N1_C	N1_A, N1_B, N1_C, N1_D, N1_E, N2_A(*), N2_B(*), N2_C(*), N2_D(*), N3_A(*), N3_B(*), N3_C(*), N3_D(*)
Subnet2	N2_B	N2_A, N2_B, N2_C, N2_D, N1_A(*), N1_B(*), N1_C(*), N1_D(*), N1_E(*), N3_A(*), N3_B(*), N3_C(*), N3_D(*)
Subnet3	N3_D	N3_A, N3_B, N3_C, N3_D, N1_A(*), N1_B(*), N1_C(*), N1_D(*), N1_E(*), N2_A(*), N2_B(*), N2_C(*), N2_D(*)

If either router R1 or R2 fails, the following will occur:

1. Names of computers on each side of the inaccessible network fragments will be maintained for as long as 36 minutes in the Network Neighborhood lists.
2. Attempts to connect to these inaccessible computers will fail, but the names will not be removed from the Network Neighborhood lists.
3. If one of the fragments is cut off from the WINS server, it will only be able to access servers on its local subnet using subnet-isolated broadcast NetBIOS name resolution. The effect is similar to that of losing access to a DNS server.

10.8 Common Errors

Many questions are asked on the mailing lists regarding browsing. The majority of browsing problems originate from incorrect configuration of NetBIOS name resolution. Some are of particular note.

10.8.1 Flushing the Samba NetBIOS Name Cache

How Can One Flush the Samba NetBIOS Name Cache without Restarting Samba?

Samba's **nmbd** process controls all browse list handling. Under normal circumstances it is safe to restart **nmbd**. This will effectively flush the

Samba NetBIOS name cache and cause it to be rebuilt. This does not make certain that a rogue machine name will not reappear in the browse list. When **nmbd** is taken out of service, another machine on the network will become the browse master. This new list may still have the rogue entry in it. If you really want to clear a rogue machine from the list, every machine on the network must be shut down and restarted after all machines are down. Failing a complete restart, the only other thing you can do is wait until the entry times out and is then flushed from the list. This may take a long time on some networks (perhaps months).

10.8.2 Server Resources Cannot Be Listed

"My Client Reports "'This server is not configured to list shared resources."""

Your guest account is probably invalid for some reason. Samba uses the guest account for browsing in **smbd**. Check that your guest account is valid.

Also see *guest account* in the `smb.conf` man page.

10.8.3 I Get an "Unable to browse the network" Error

This error can have multiple causes:

- There is no LMB. Configure nmbd or any other machine to serve as LMB.
- You cannot log onto the machine that is the LMB. Can you log on to it as a guest user?
- There is no IP connectivity to the LMB. Can you reach it by broadcast?

10.8.4 Browsing of Shares and Directories is Very Slow

" There are only two machines on a test network. One is a Samba server, the other a Windows XP machine. Authentication and logons work perfectly, but when I try to explore shares on the Samba server, the Windows XP client becomes unresponsive. Sometimes it does not respond for some minutes. Eventually, Windows Explorer will respond and displays files and directories without problem."

“ But, the share is immediately available from a command shell (**cmd**, followed by exploration with DOS command. Is this a Samba problem, or is it a Windows problem? How can I solve this?”

Here are a few possibilities:

Bad Networking Hardware Most common defective hardware problems center around low cost or defective hubs, routers, network interface controllers (NICs), and bad wiring. If one piece of hardware is defective, the whole network may suffer. Bad networking hardware can cause data corruption. Most bad networking hardware problems are accompanied by an increase in apparent network traffic, but not all.

The Windows XP WebClient A number of sites have reported similar slow network browsing problems and found that when the WebClient service is turned off, the problem disappears. This is certainly something that should be explored because it is a simple solution — if it works.

Inconsistent WINS Configuration This type of problem is common when one client is configured to use a WINS server (that is a TCP/IP configuration setting) and there is no WINS server on the network. Alternatively, this will happen if there is a WINS server and Samba is not configured to use it. The use of WINS is highly recommended if the network is using NetBIOS over TCP/IP protocols. If use of NetBIOS over TCP/IP is disabled on all clients, Samba should not be configured as a WINS server, nor should it be configured to use one.

Incorrect DNS Configuration If use of NetBIOS over TCP/IP is disabled, Active Directory is in use and the DNS server has been incorrectly configured. For further information refer to Section 10.3.3.

10.8.5 Invalid Cached Share References Affects Network Browsing

Cached references on your MS Windows client (workstation or server) to shares or servers that no longer exist can cause MS Windows Explorer to appear unresponsive as it tries to connect to these shares. After a delay

(can take a long time) it times out and browsing will appear to be mostly normal again.

To eliminate the problem the stale cached references should be removed. This does not happen automatically and requires manual intervention. This is a design feature of MS Windows and not anything that Samba can change. To remove the stale shortcuts found in *My Network Places* which refer to what are now invalid shares or servers it is necessary to edit the Windows Registry under `HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\`. Edit the entry `MountPoints2` (on Windows XP and later, or `MountPoints` on Windows 2000 and earlier). Remove all keys named `\\server\share` (where 'server' and 'share' refer to a non-existent server or share).

NOTE

Removal of stale network links needs to be done on a per-user basis. Alternately, you can delete the shortcuts from the MS Windows Explorer in `My Network Places` just by right-clicking them and selecting *Delete.*

Samba users have reported that these stale references negatively affect network browsing with Windows, Samba, and Novell servers. It is suspected to be a universal problem not directly related to the Samba server. Samba users may experience this more often due to Samba being somewhat viewed as an experimenter's toolkit. This results from the fact that a user might go through several reconfigurations and incarnations of their Samba server, by different names, with different shares, increasing the chances for having stale (invalid) cached share references. Windows clients do not expire these references thus necessitating manual removal.

It is common for *Open* dialog boxes (for example; in Word and Excel) to respond very slowly, as they attempt to locate all of the cached references, even if they are not in the current directory being accessed.

Chapter 11

ACCOUNT INFORMATION DATABASES

Early releases of Samba-3 implemented new capability to work concurrently with multiple account backends. This capability was removed beginning with release of Samba 3.0.23. Commencing with Samba 3.0.23 it is possible to work with only one specified passwd backend.

The three passdb backends that are fully maintained (actively supported) by the Samba Team are: `smbpasswd` (being obsoleted), `tdbsam` (a tdb-based binary file format), and `ldapsam` (LDAP directory). Of these, only the `ldapsam` backend stores both POSIX (UNIX) and Samba user and group account information in a single repository. The `smbpasswd` and `tdbsam` backends store only Samba user accounts.

In a strict sense, there are three supported account storage and access systems. One of these is considered obsolete (smbpasswd). It is recommended to use the `tdbsam` method for all simple systems. Use `ldapsam` for larger and more complex networks.

In a strict and literal sense, the passdb backends are account storage mechanisms (or methods) alone. The choice of terminology can be misleading, however we are stuck with this choice of wording. This chapter documents the nature of the account storage system with a focus on user and trust accounts. Trust accounts have two forms, machine trust accounts (computer accounts) and interdomain trust accounts. These are all treated as user-like entities.

11.1 Features and Benefits

Samba-3 provides for complete backward compatibility with Samba-2.2.x functionality as follows:

11.1.1 Backward Compatibility Account Storage Systems

Plaintext This isn't really a backend at all, but is listed here for simplicity. Samba can be configured to pass plaintext authentication requests to the traditional UNIX/Linux `/etc/passwd` and `/etc/shadow`-style subsystems. On systems that have Pluggable Authentication Modules (PAM) support, all PAM modules are supported. The behavior is just as it was with Samba-2.2.x, and the protocol limitations imposed by MS Windows clients apply likewise. Please refer to Section 11.2, for more information regarding the limitations of plaintext password usage.

smbpasswd This option allows continued use of the `smbpasswd` file that maintains a plain ASCII (text) layout that includes the MS Windows LanMan and NT-encrypted passwords as well as a field that stores some account information. This form of password backend does not store any of the MS Windows NT/200x SAM (Security Account Manager) information required to provide the extended controls that are needed for more comprehensive interoperation with MS Windows NT4/200x servers.

This backend should be used only for backward compatibility with older versions of Samba. It may be deprecated in future releases.

ldapsam_compat (Samba-2.2 LDAP Compatibility) There is a password backend option that allows continued operation with an existing OpenLDAP backend that uses the Samba-2.2.x LDAP schema extension. This option is provided primarily as a migration tool, although there is no reason to force migration at this time. This tool will eventually be deprecated.

11.1.2 New Account Storage Systems

Samba-3 introduces a number of new password backend capabilities.

tdbsam This backend provides a rich database backend for local servers. This backend is not suitable for multiple domain controllers (i.e., PDC + one or more BDC) installations.

The *tdbsam* password backend stores the old *smbpasswd* information plus the extended MS Windows NT/200x SAM information into a binary format TDB (trivial database) file. The inclusion of the extended information makes it possible for Samba-3 to implement the same account and system access controls that are possible with MS Windows NT4/200x-based systems.

The inclusion of the *tdbsam* capability is a direct response to user requests to allow simple site operation without the overhead of the complexities of running OpenLDAP. It is recommended to use this only for sites that have fewer than 250 users. For larger sites or implementations, the use of OpenLDAP or of Active Directory integration is strongly recommended.

ldapsam This provides a rich directory backend for distributed account installation.

Samba-3 has a new and extended LDAP implementation that requires configuration of OpenLDAP with a new format Samba schema. The new format schema file is included in the `examples/LDAP` directory of the Samba distribution.

The new LDAP implementation significantly expands the control abilities that were possible with prior versions of Samba. It is now possible to specify "per-user" profile settings, home directories, account access controls, and much more. Corporate sites will see that the Samba Team has listened to their requests both for capability and greater scalability.

11.2 Technical Information

Old Windows clients send plaintext passwords over the wire. Samba can check these passwords by encrypting them and comparing them to the hash stored in the UNIX user database.

Newer Windows clients send encrypted passwords (LanMan and NT hashes) instead of plaintext passwords over the wire. The newest clients will send only encrypted passwords and refuse to send plaintext passwords unless their registry is tweaked.

Many people ask why Samba cannot simply use the UNIX password database. Windows requires passwords that are encrypted in its own format. The UNIX passwords can't be converted to UNIX-style encrypted passwords. Because of that, you can't use the standard UNIX user database, and you have to store the LanMan and NT hashes somewhere else.

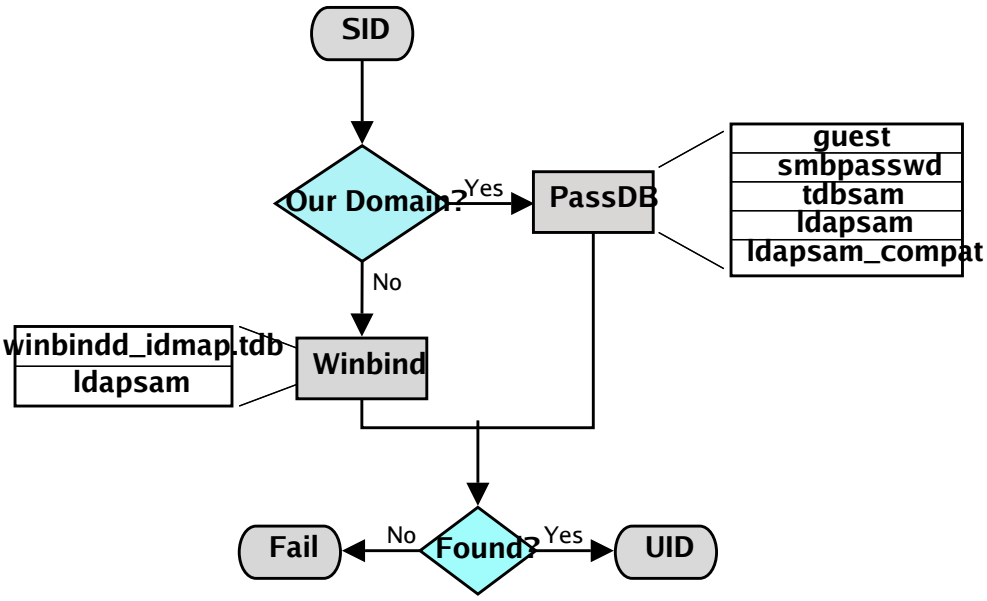
In addition to differently encrypted passwords, Windows also stores certain data for each user that is not stored in a UNIX user database: for example, workstations the user may logon from, the location where the user's profile is stored, and so on. Samba retrieves and stores this information using a *passdb backend*. Commonly available backends are LDAP, tdbsam, and plain text file. For more information, see the man page for `smb.conf` regarding the *passdb backend* parameter.

The resolution of SIDs to UIDs is fundamental to correct operation of Samba. In both cases shown, if winbindd is not running or cannot be contacted, then only local SID/UID resolution is possible. See Figure 11.1 and Figure 11.2 diagrams.

11.2.1 Important Notes About Security

The UNIX and SMB password encryption techniques seem similar on the surface. This similarity is, however, only skin deep. The UNIX scheme typically sends clear-text passwords over the network when logging in. This is bad. The SMB encryption scheme never sends the clear-text password over the network, but it does store the 16-byte hashed values on disk. This is also bad. Why? Because the 16 byte hashed values are a "password equivalent." You cannot derive the user's password from them, but they could potentially be used in a modified client to gain access to a server. This would require considerable technical knowledge on behalf of the attacker but

Figure 11.1 IDMAP: Resolution of SIDs to UIDs.



is perfectly possible. You should therefore treat the data stored in whatever passdb backend you use (smbpasswd file, LDAP) as though it contained the clear-text passwords of all your users. Its contents must be kept secret, and the file should be protected accordingly.

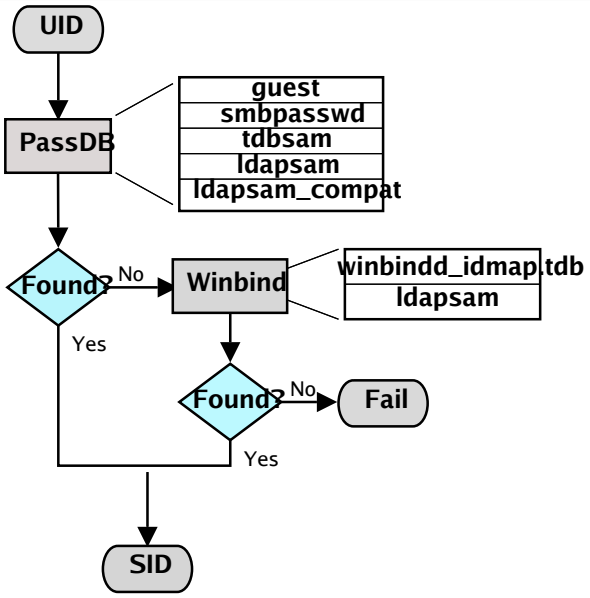
Ideally, we would like a password scheme that involves neither plaintext passwords on the network nor plaintext passwords on disk. Unfortunately, this is not available because Samba is stuck with having to be compatible with other SMB systems (Windows NT, Windows for Workgroups, Windows 9x/Me).

Windows NT 4.0 Service Pack 3 changed the default setting so plaintext passwords are disabled from being sent over the wire. This mandates either the use of encrypted password support or editing the Windows NT registry to re-enable plaintext passwords.

The following versions of Microsoft Windows do not support full domain security protocols, although they may log onto a domain environment:

- MS DOS Network client 3.0 with the basic network redirector installed.
- Windows 95 with the network redirector update installed.

Figure 11.2 IDMAP: Resolution of UIDs to SIDs.



- Windows 98 [Second Edition].
- Windows Me.

NOTE

MS Windows XP Home does not have facilities to become a domain member, and it cannot participate in domain logons.

The following versions of MS Windows fully support domain security protocols.

- Windows NT 3.5x.
- Windows NT 4.0.
- Windows 2000 Professional.
- Windows 200x Server/Advanced Server.

- Windows XP Professional.

All current releases of Microsoft SMB/CIFS clients support authentication via the SMB challenge/response mechanism described here. Enabling clear-text authentication does not disable the ability of the client to participate in encrypted authentication. Instead, it allows the client to negotiate either plaintext or encrypted password handling.

MS Windows clients will cache the encrypted password alone. Where plaintext passwords are re-enabled through the appropriate registry change, the plaintext password is never cached. This means that in the event that a network connections should become disconnected (broken), only the cached (encrypted) password will be sent to the resource server to effect an auto-reconnect. If the resource server does not support encrypted passwords, the auto-reconnect will fail. Use of encrypted passwords is strongly advised.

11.2.1.1 Advantages of Encrypted Passwords

- Plaintext passwords are not passed across the network. Someone using a network sniffer cannot just record passwords going to the SMB server.
- Plaintext passwords are not stored anywhere in memory or on disk.
- Windows NT does not like talking to a server that does not support encrypted passwords. It will refuse to browse the server if the server is also in user-level security mode. It will insist on prompting the user for the password on each connection, which is very annoying. The only thing you can do to stop this is to use SMB encryption.
- Encrypted password support allows automatic share (resource) reconnects.
- Encrypted passwords are essential for PDC/BDC operation.

11.2.1.2 Advantages of Non-Encrypted Passwords

- Plaintext passwords are not kept on disk and are not cached in memory.
- Plaintext passwords use the same password file as other UNIX services, such as Login and FTP.

- Use of other services (such as Telnet and FTP) that send plaintext passwords over the network makes sending them for SMB not such a big deal.

11.2.2 Mapping User Identifiers between MS Windows and UNIX

Every operation in UNIX/Linux requires a user identifier (UID), just as in MS Windows NT4/200x this requires a security identifier (SID). Samba provides two means for mapping an MS Windows user to a UNIX/Linux UID.

First, all Samba SAM database accounts require a UNIX/Linux UID that the account will map to. As users are added to the account information database, Samba will call the *add user script* interface to add the account to the Samba host OS. In essence all accounts in the local SAM require a local user account.

The second way to map Windows SID to UNIX UID is via the *idmap uid* and *idmap gid* parameters in `smb.conf`. Please refer to the man page for information about these parameters. These parameters are essential when mapping users from a remote (non-member Windows client or a member of a foreign domain) SAM server.

11.2.3 Mapping Common UIDs/GIDs on Distributed Machines

Samba-3 has a special facility that makes it possible to maintain identical UIDs and GIDs on all servers in a distributed network. A distributed network is one where there exists a PDC, one or more BDCs, and/or one or more domain member servers. Why is this important? This is important if files are being shared over more than one protocol (e.g., NFS) and where users are copying files across UNIX/Linux systems using tools such as **rsync**.

The special facility is enabled using a parameter called *`idmap backend`*. The default setting for this parameter is an empty string. Technically it is possible to use an LDAP-based idmap backend for UIDs and GIDs, but it makes most sense when this is done for network configurations that also use LDAP for the SAM backend. Example 11.2.1 shows that configuration.

A network administrator who wants to make significant use of LDAP backends will sooner or later be exposed to the excellent work done by PADL

Example 11.2.1 Example Configuration with the LDAP idmap Backend

```
[global]
        idmap backend = ldap:ldap://ldap-server.quenya. ↩
            org:636
 # Alternatively, this could be specified as:
        idmap backend = ldap:ldaps://ldap-server.quenya. ↩
            org
```

Software. PADL <http://www.padl.com> have produced and released to open source an array of tools that might be of interest. These tools include:

- *nss_ldap:* An LDAP name service switch (NSS) module to provide native name service support for AIX, Linux, Solaris, and other operating systems. This tool can be used for centralized storage and retrieval of UIDs and GIDs.
- *pam_ldap:* A PAM module that provides LDAP integration for UNIX/Linux system access authentication.
- *idmap_ad:* An IDMAP backend that supports the Microsoft Services for UNIX RFC 2307 schema available from the PADL Web site[1].

11.2.4 Comments Regarding LDAP

There is much excitement and interest in LDAP directories in the information technology world today. The LDAP architecture was designed to be highly scalable. It was also designed for use across a huge number of potential areas of application encompassing a wide range of operating systems and platforms. LDAP technologies are at the heart of the current generations of Federated Identity Management (FIM) solutions that can underlie a corporate Single Sign-On (SSO) environment.

LDAP implementations have been built across a wide variety of platforms. It lies at the core of Microsoft Windows Active Directory services (ADS), Novell's eDirectory, as well as many others. Implementation of the directory services LDAP involves interaction with legacy as well as new generation applications, all of which depend on some form of authentication services.

[1] <http://www.padl.com/download/xad_oss_plugins.tar.gz>

UNIX services can utilize LDAP directory information for authentication and access controls through intermediate tools and utilities. The total environment that consists of the LDAP directory and the middle-ware tools and utilities makes it possible for all user access to the UNIX platform to be managed from a central environment and yet distributed to wherever the point of need may be physically located. Applications that benefit from this infrastructure include: UNIX login shells, mail and messaging systems, quota controls, printing systems, DNS servers, DHCP servers, and also Samba.

Many sites are installing LDAP for the first time in order to provide a scalable passdb backend for Samba. Others are faced with the need to adapt an existing LDAP directory to new uses such as for the Samba SAM backend. Whatever your particular need and attraction to Samba may be, decisions made in respect of the design of the LDAP directory structure and its implementation are of a durable nature for the site. These have far-reaching implications that affect long-term information systems management costs.

Do not rush into an LDAP deployment. Take the time to understand how the design of the Directory Information Tree (DIT) may impact current and future site needs, as well as the ability to meet them. The way that Samba SAM information should be stored within the DIT varies from site to site and with each implementation new experience is gained. It is well understood by LDAP veterans that first implementations create awakening, second implementations of LDAP create fear, and third-generation deployments bring peace and tranquility.

11.2.4.1 Caution Regarding LDAP and Samba

Samba requires UNIX POSIX identity information as well as a place to store information that is specific to Samba and the Windows networking environment. The most used information that must be dealt with includes: user accounts, group accounts, machine trust accounts, interdomain trust accounts, and intermediate information specific to Samba internals.

The example deployment guidelines in this book, as well as other books and HOWTO documents available from the internet may not fit with established directory designs and implementations. The existing DIT may not be able to accommodate the simple information layout proposed in common sources. Additionally, you may find that the common scripts and tools that are used to provision the LDAP directory for use with Samba may not suit your needs.

It is not uncommon, for sites that have existing LDAP DITs to find necessity to generate a set of site-specific scripts and utilities to make it possible to deploy Samba within the scope of site operations. The way that user and group accounts are distributed throughout the DIT may make this a challenging matter. The solution will, of course, be rewarding, but the journey to it may be challenging. Take time to understand site needs and do not rush into deployment.

Above all, do not blindly use scripts and tools that are not suitable for your site. Check and validate all scripts before you execute them to make sure that the existing infrastructure will not be damaged by inadvertent use of an inappropriate tool.

11.2.5 LDAP Directories and Windows Computer Accounts

Samba doesn't provide a turnkey solution to LDAP. It is best to deal with the design and configuration of an LDAP directory prior to integration with Samba. A working knowledge of LDAP makes Samba integration easy, and the lack of a working knowledge of LDAP can make it a frustrating experience.

Computer (machine) accounts can be placed wherever you like in an LDAP directory subject to some constraints that are described in this chapter.

The POSIX and sambaSamAccount components of computer (machine) accounts are both used by Samba. Thus, machine accounts are treated inside Samba in the same way that Windows NT4/200X treats them. A user account and a machine account are indistinguishable from each other, except that the machine account ends in a $ character, as do trust accounts.

The need for Windows user, group, machine, trust, and other accounts to be tied to a valid UNIX UID is a design decision that was made a long way back in the history of Samba development. It is unlikely that this decision will be reversed or changed during the remaining life of the Samba-3.x series.

The resolution of a UID from the Windows SID is achieved within Samba through a mechanism that must refer back to the host operating system on which Samba is running. The NSS is the preferred mechanism that shields applications (like Samba) from the need to know everything about every host OS it runs on.

Samba asks the host OS to provide a UID via the "passwd", "shadow", and "group" facilities in the NSS control (configuration) file. The best tool for achieving this is left up to the UNIX administrator to determine. It is not imposed by Samba. Samba provides winbindd with its support libraries as one method. It is possible to do this via LDAP, and for that Samba provides the appropriate hooks so that all account entities can be located in an LDAP directory.

For many the weapon of choice is to use the PADL nss_ldap utility. This utility must be configured so that computer accounts can be resolved to a POSIX/UNIX account UID. That is fundamentally an LDAP design question. The information provided on the Samba list and in the documentation is directed at providing working examples only. The design of an LDAP directory is a complex subject that is beyond the scope of this documentation.

11.3 Account Management Tools

Samba provides two tools for management of user and machine accounts: **smbpasswd** and **pdbedit**.

The **pdbedit** can be used to manage account policies in addition to Samba user account information. The policy management capability is used to administer domain default settings for password aging and management controls to handle failed login attempts.

Some people are confused when reference is made to `smbpasswd` because the name refers to a storage mechanism for SambaSAMAccount information, but it is also the name of a utility tool. That tool is destined to eventually be replaced by new functionality that is being added to the **net** toolset (see Chapter 13, "Remote and Local Management: The Net Command".

11.3.1 The smbpasswd Tool

The **smbpasswd** utility is similar to the **passwd** and **yppasswd** programs. It maintains the two 32 byte password fields in the passdb backend. This utility operates independently of the actual account and password storage methods used (as specified by the *`passdb backend`* in the `smb.conf` file.

smbpasswd works in a client-server mode where it contacts the local smbd to change the user's password on its behalf. This has enormous benefits.

smbpasswd has the capability to change passwords on Windows NT servers (this only works when the request is sent to the NT PDC if changing an NT domain user's password).

smbpasswd can be used to:

- *add* user or machine accounts.
- *delete* user or machine accounts.
- *enable* user or machine accounts.
- *disable* user or machine accounts.
- *set to NULL* user passwords.
- *manage* interdomain trust accounts.

To run smbpasswd as a normal user, just type:

```
$ smbpasswd
Old SMB password: secret
```

For *secret*, type the old value here or press return if there is no old password.

```
New SMB Password: new secret
Repeat New SMB Password: new secret
```

If the old value does not match the current value stored for that user, or the two new values do not match each other, then the password will not be changed.

When invoked by an ordinary user, the command will allow only the user to change his or her own SMB password.

When run by root, **smbpasswd** may take an optional argument specifying the username whose SMB password you wish to change. When run as root, **smbpasswd** does not prompt for or check the old password value, thus allowing root to set passwords for users who have forgotten their passwords.

smbpasswd is designed to work in the way familiar to UNIX users who use the **passwd** or **yppasswd** commands. While designed for administrative use, this tool provides essential user-level password change capabilities.

For more details on using **smbpasswd**, refer to the man page (the definitive reference).

11.3.2 The pdbedit Tool

pdbedit is a tool that can be used only by root. It is used to manage the passdb backend, as well as domain-wide account policy settings. **pdbedit** can be used to:

- add, remove, or modify user accounts.
- list user accounts.
- migrate user accounts.
- migrate group accounts.
- manage account policies.
- manage domain access policy settings.

Under the terms of the Sarbanes-Oxley Act of 2002, American businesses and organizations are mandated to implement a series of `internal controls` and procedures to communicate, store, and protect financial data. The Sarbanes-Oxley Act has far reaching implications in respect of:

1. Who has access to information systems that store financial data.
2. How personal and financial information is treated among employees and business partners.
3. How security vulnerabilities are managed.
4. Security and patch level maintenance for all information systems.
5. How information systems changes are documented and tracked.
6. How information access controls are implemented and managed.
7. Auditability of all information systems in respect of change and security.
8. Disciplinary procedures and controls to ensure privacy.

In short, the Sarbanes-Oxley Act of 2002 is an instrument that enforces accountability in respect of business related information systems so as to ensure the compliance of all information systems that are used to store personal information and particularly for financial records processing. Similar accountabilities are being demanded around the world.

The need to be familiar with the Samba tools and facilities that permit information systems operation in compliance with government laws and regulations is clear to all. The **pdbedit** is currently the only Samba tool that provides the capacity to manage account and systems access controls and policies. During the remaining life-cycle of the Samba-3 series it is possible the new tools may be implemented to aid in this important area.

Domain global policy controls available in Windows NT4 compared with Samba is shown in Table 11.1.

The **pdbedit** tool is the only one that can manage the account security and policy settings. It is capable of all operations that smbpasswd can do as well as a superset of them.

One particularly important purpose of the **pdbedit** is to allow the import/-export of account information from one passdb backend to another.

11.3.2.1 User Account Management

The **pdbedit** tool, like the **smbpasswd** tool, requires that a POSIX user account already exists in the UNIX/Linux system accounts database (backend). Neither tool will call out to the operating system to create a user account because this is considered to be the responsibility of the system administrator. When the Windows NT4 domain user manager is used to add an account, Samba will implement the `add user script` (as well as the other interface scripts) to ensure that user, group and machine accounts are correctly created and changed. The use of the **pdbedit** tool does not make use of these interface scripts.

Before attempting to use the **pdbedit** tool to manage user and machine accounts, make certain that a system (POSIX) account has already been created.

Listing User and Machine Accounts The following is an example of the user account information that is stored in a tdbsam password backend. This listing was produced by running:

Table 11.1 NT4 Domain v's Samba Policy Controls

NT4 policy Name	Samba Policy Name	NT4 Range	Samba Range	Samba Default
Maximum Password Age	maximum password age	0 - 999 (days)	0 - 4294967295 (sec)	4294967295
Minimum Password Age	minimum password age	0 - 999 (days)	0 - 4294967295 (sec)	0
Minimum Password Length	min password length	1 - 14 (Chars)	0 - 4294967295 (Chars)	5
Password Uniqueness	password history	0 - 23 (#)	0 - 4294967295 (#)	0
Account Lockout - Reset count after	reset count minutes	1 - 99998 (min)	0 - 4294967295 (min)	30
Lockout after bad logon attempts	bad lockout attempt	0 - 998 (#)	0 - 4294967295 (#)	0
*** Not Known ***	disconnect time	TBA	0 - 4294967295	0
Lockout Duration	lockout duration	1 - 99998 (min)	0 - 4294967295 (min)	30
Users must log on in order to change password	user must logon to change password	0/1	0 - 4294967295	0
*** Registry Setting ***	refuse machine password change	0/1	0 - 4294967295	0

```
$ pdbedit -Lv met
UNIX username:        met
NT username:          met
Account Flags:        [U          ]
User SID:             S-1-5-21-1449123459-1407424037-3116680435-2004
```

```
Primary Group SID:    S-1-5-21-1449123459-1407424037-3116680435-1201
Full Name:            Melissa E Terpstra
Home Directory:       \\frodo\met\Win9Profile
HomeDir Drive:        H:
Logon Script:         scripts\logon.bat
Profile Path:         \\frodo\Profiles\met
Domain:               MIDEARTH
Account desc:
Workstations:         melbelle
Munged dial:
Logon time:           0
Logoff time:          Mon, 18 Jan 2038 20:14:07 GMT
Kickoff time:         Mon, 18 Jan 2038 20:14:07 GMT
Password last set:    Sat, 14 Dec 2002 14:37:03 GMT
Password can change:  Sat, 14 Dec 2002 14:37:03 GMT
Password must change: Mon, 18 Jan 2038 20:14:07 GMT
```

Accounts can also be listed in the older `smbpasswd` format:

```
root# pdbedit -Lw
root:0:84B0D8E14D158FF8417EAF50CFAC29C3:
     AF6DD3FD4E2EA8BDE1695A3F05EFBF52:[U          ]:LCT-42681AB8:
jht:1000:6BBC4159020A52741486235A2333E4D2:
     CC099521AD554A3C3CF2556274DBCFBC:[U          ]:LCT-40D75B5B:
rcg:1002:E95D4331A6F23AF8AAD3B435B51404EE:
     BB0F2C39B04CA6100F0E535DF8314B43:[U          ]:LCT-40D7C5A3:
afw:1003:1AAFA7F9F6DC1DEAAAD3B435B51404EE:
     CE92C2F9471594CDC4E7860CA6BC62DB:[T          ]:LCT-40DA501F:
met:1004:A2848CB7E076B435AAD3B435B51404EE:
     F25F5D3405085C555236B80B7B22C0D2:[U          ]:LCT-4244FAB8:
aurora$:1005:060DE593EA638B8ACC4A19F14D2FF2BB:
     060DE593EA638B8ACC4A19F14D2FF2BB:[W          ]:LCT-4173E5CC:
temptation$:1006:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:
     A96703C014E404E33D4049F706C45EE9:[W          ]:LCT-42BF0C57:
vaioboss$:1001:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:
     88A30A095160072784C88F811E89F98A:[W          ]:LCT-41C3878D:
frodo$:1008:15891DC6B843ECA41249940C814E316B:
     B68EADCCD18E17503D3DAD3E6B0B9A75:[W          ]:LCT-42B7979F:
marvel$:1011:BF709959C3C94E0B3958B7B84A3BB6F3:
```

```
C610EFE9A385A3E8AA46ADFD576E6881:[W          ]:LCT-40F07A4
```

The account information that was returned by this command in order from left to right consists of the following colon separated data:

- Login ID.
- UNIX UID.
- Microsoft LanManager password hash (password converted to uppercase then hashed.
- Microsoft NT password hash (hash of the case-preserved password).
- Samba SAM Account Flags.
- The LCT data (password last change time).

The Account Flags parameters are documented in the **pdbedit** man page, and are briefly documented in Section 11.3.2.1.

The LCT data consists of 8 hexadecimal characters representing the time since January 1, 1970, of the time when the password was last changed.

Adding User Accounts The **pdbedit** can be used to add a user account to a standalone server or to a domain. In the example shown here the account for the user `vlaan` has been created before attempting to add the SambaSAMAccount.

```
root#  pdbedit -a vlaan
new password: secretpw
retype new password: secretpw
Unix username:        vlaan
NT username:          vlaan
Account Flags:        [U          ]
User SID:             S-1-5-21-726309263-4128913605-1168186429-3014
Primary Group SID:    S-1-5-21-726309263-4128913605-1168186429-513
Full Name:            Victor Laan
Home Directory:       \\frodo\vlaan
HomeDir Drive:        H:
Logon Script:         scripts\logon.bat
Profile Path:         \\frodo\profiles\vlaan
Domain:               MIDEARTH
```

```
Account desc:         Guest User
Workstations:
Munged dial:
Logon time:           0
Logoff time:          Mon, 18 Jan 2038 20:14:07 GMT
Kickoff time:         Mon, 18 Jan 2038 20:14:07 GMT
Password last set:    Wed, 29 Jun 2005 19:35:12 GMT
Password can change:  Wed, 29 Jun 2005 19:35:12 GMT
Password must change: Mon, 18 Jan 2038 20:14:07 GMT
Last bad password   : 0
Bad password count  : 0
Logon hours         : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

Deleting Accounts An account can be deleted from the SambaSAMAccount database

```
root#  pdbedit -x vlaan
```

The account is removed without further screen output. The account is removed only from the SambaSAMAccount (passdb backend) database, it is not removed from the UNIX account backend.

The use of the NT4 domain user manager to delete an account will trigger the *delete user script*, but not the **pdbedit** tool.

Changing User Accounts Refer to the **pdbedit** man page for a full synopsis of all operations that are available with this tool.

An example of a simple change in the user account information is the change of the full name information shown here:

```
root#  pdbedit -r --fullname="Victor Aluicious Laan" vlaan
...
Primary Group SID:    S-1-5-21-726309263-4128913605-1168186429-513
Full Name:            Victor Aluicious Laan
Home Directory:       \\frodo\vlaan
...
```

Let us assume for a moment that a user's password has expired and the user is unable to change the password at this time. It may be necessary to give the user additional grace time so that it is possible to continue to work with the account and the original password. This demonstrates how the password expiration settings may be updated

```
root#  pdbedit -Lv vlaan
...
Password last set:    Sun, 09 Sep 2001 22:21:40 GMT
Password can change:  Thu, 03 Jan 2002 15:08:35 GMT
Password must change: Thu, 03 Jan 2002 15:08:35 GMT
Last bad password   : Thu, 03 Jan 2002 15:08:35 GMT
Bad password count  : 2
...
```

The user has recorded 2 bad logon attempts and the next will lock the account, but the password is also expired. Here is how this account can be reset:

```
root#  pdbedit -z vlaan
...
Password last set:    Sun, 09 Sep 2001 22:21:40 GMT
Password can change:  Thu, 03 Jan 2002 15:08:35 GMT
Password must change: Thu, 03 Jan 2002 15:08:35 GMT
Last bad password   : 0
Bad password count  : 0
...
```

The `Password must change:` parameter can be reset like this:

```
root#  pdbedit --pwd-must-change-time=1200000000 vlaan
...
Password last set:    Sun, 09 Sep 2001 22:21:40 GMT
Password can change:  Thu, 03 Jan 2002 15:08:35 GMT
Password must change: Thu, 10 Jan 2008 14:20:00 GMT
...
```

Another way to use this tools is to set the date like this:

```
root#  pdbedit --pwd-must-change-time="2010-01-01" \
              --time-format="%Y-%m-%d" vlaan
...
Password last set:    Sun, 09 Sep 2001 22:21:40 GMT
Password can change:  Thu, 03 Jan 2002 15:08:35 GMT
Password must change: Fri, 01 Jan 2010 00:00:00 GMT
...
```

Refer to the strptime man page for specific time format information.

Please refer to the pdbedit man page for further information relating to SambaSAMAccount management.

Account Flags Management The Samba SAM account flags are properly called the ACB (account control block) within the Samba source code. In some parts of the Samba source code they are referred to as the account encode_bits, and also as the account control flags.

The manual adjustment of user, machine (workstation or server) or an inter-domain trust account account flgas should not be necessary under normal conditions of use of Samba. On the other hand, where this information becomes corrupted for some reason, the ability to correct the damaged data is certainly useful. The tool of choice by which such correction can be affected is the **pdbedit** utility.

There have been a few requests for information regarding the account flags from developers who are creating their own Samba management tools. An example of a need for information regarding the proper management of the account flags is evident when developing scripts that will be used to manage an LDAP directory.

The account flag field can contain up to 16 characters. Presently, only 11 are in use. These are listed in Table 11.2. The order in which the flags are specified to the **pdbedit** command is not important. In fact, they can be set without problem in any order in the SambaAcctFlags record in the LDAP directory.

An example of use of the **pdbedit** utility to set the account control flags is shown here:

Table 11.2 Samba SAM Account Control Block Flags

Flag	Description
D	Account is disabled.
H	A home directory is required.
I	An inter-domain trust account.
L	Account has been auto-locked.
M	An MNS (Microsoft network service) logon account.
N	Password not required.
S	A server trust account.
T	Temporary duplicate account entry.
U	A normal user account.
W	A workstation trust account.
X	Password does not expire.

```
root#  pdbedit -r -c "[DLX]" jht
Unix username:        jht
NT username:          jht
Account Flags:        [DHULX      ]
User SID:             S-1-5-21-729263-4123605-1186429-3000
Primary Group SID:    S-1-5-21-729263-4123605-1186429-513
Full Name:            John H Terpstra,Utah Office
Home Directory:       \\aurora\jht
HomeDir Drive:        H:
Logon Script:         scripts\logon.bat
Profile Path:         \\aurora\profiles\jht
Domain:               MIDEARTH
Account desc:         BluntObject
Workstations:
Logon time:           0
Logoff time:          Mon, 18 Jan 2038 20:14:07 GMT
Kickoff time:         0
Password last set:    Sun, 03 Jul 2005 23:19:18 GMT
Password can change:  Sun, 03 Jul 2005 23:19:18 GMT
Password must change: Mon, 18 Jan 2038 20:14:07 GMT
Last bad password   : 0
Bad password count  : 0
Logon hours         : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

The flags can be reset to the default settings by executing:

```
root#  pdbedit -r -c "[]" jht
Unix username:        jht
NT username:          jht
Account Flags:        [U          ]
User SID:             S-1-5-21-729263-4123605-1186429-3000
Primary Group SID:    S-1-5-21-729263-4123605-1186429-513
Full Name:            John H Terpstra,Utah Office
Home Directory:       \\aurora\jht
HomeDir Drive:        H:
Logon Script:         scripts\logon.bat
Profile Path:         \\aurora\profiles\jht
Domain:               MIDEARTH
Account desc:         BluntObject
Workstations:
Logon time:           0
Logoff time:          Mon, 18 Jan 2038 20:14:07 GMT
Kickoff time:         0
Password last set:    Sun, 03 Jul 2005 23:19:18 GMT
Password can change:  Sun, 03 Jul 2005 23:19:18 GMT
Password must change: Mon, 18 Jan 2038 20:14:07 GMT
Last bad password   : 0
Bad password count  : 0
Logon hours         : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

Domain Account Policy Managment To view the domain account access policies that may be configured execute:

```
root#  pdbedit -P ?
No account policy by that name
Account policy names are :
min password length
password history
user must logon to change password
maximum password age
```

```
minimum password age
lockout duration
reset count minutes
bad lockout attempt
disconnect time
refuse machine password change
```

Commands will be executed to establish controls for our domain as follows:

1. min password length = 8 characters.
2. password history = last 4 passwords.
3. maximum password age = 90 days.
4. minimum password age = 7 days.
5. bad lockout attempt = 8 bad logon attempts.
6. lockout duration = forever, account must be manually reenabled.

The following command execution will achieve these settings:

```
root#  pdbedit -P "min password length" -C 8
account policy value for min password length was 5
account policy value for min password length is now 8
root#  pdbedit -P "password history" -C 4
account policy value for password history was 0
account policy value for password history is now 4
root#  pdbedit -P "maximum password age" -C 7776000
account policy value for maximum password age was 4294967295
account policy value for maximum password age is now 7776000
root#  pdbedit -P "minimum password age" -C 604800
account policy value for minimum password age was 0
account policy value for minimum password age is now 7
root#  pdbedit -P "bad lockout attempt" -C 8
account policy value for bad lockout attempt was 0
account policy value for bad lockout attempt is now 8
root#  pdbedit -P "lockout duration" -C -1
account policy value for lockout duration was 30
account policy value for lockout duration is now 4294967295
```

NOTE

To set the maximum (infinite) lockout time use the value of -1.

WARNING

Account policies must be set individually on each PDC and BDC. At this time (Samba 3.0.11 to Samba 3.0.14a) account policies are not replicated automatically. This may be fixed before Samba 3.0.20 ships or some time there after. Please check the WHATSNEW.txt file in the Samba-3 tarball for specific update notiations regarding this facility.

11.3.2.2 Account Import/Export

The **pdbedit** tool allows import/export of authentication (account) databases from one backend to another. For example, to import/export accounts from an old `smbpasswd` database to a *`tdbsam`* backend:

1.
```
root# pdbedit -i smbpasswd -e tdbsam
```

2. Replace the *`smbpasswd`* with *`tdbsam`* in the *`passdb backend`* configuration in `smb.conf`.

11.4 Password Backends

Samba offers flexibility in backend account database design. The flexibility is immediately obvious as one begins to explore this capability. Recent changes to Samba (since 3.0.23) have removed the mulitple backend feature in order

to simplify problems that broke some installations. This removal has made the internal operation of Samba-3 more consistent and predictable.

Beginning with Samba 3.0.23 it is no longer possible to specify use of mulitple passdb backends. Earlier versions of Samba-3 made it possible to specify multiple password backends, and even multiple backends of the same type. The multiple passdb backend capability caused many problems with name to SID and SID to name ID resolution. The Samba team wrestled with the challenges and decided that this feature needed to be removed.

11.4.1 Plaintext

Older versions of Samba retrieved user information from the UNIX user database and eventually some other fields from the file `/etc/samba/smbpasswd` or `/etc/smbpasswd`. When password encryption is disabled, no SMB-specific data is stored at all. Instead, all operations are conducted via the way that the Samba host OS will access its `/etc/passwd` database. On most Linux systems, for example, all user and group resolution is done via PAM.

11.4.2 smbpasswd: Encrypted Password Database

Traditionally, when configuring *encrypt passwords* = yes in Samba's `smb.conf` file, user account information such as username, LM/NT password hashes, password change times, and account flags have been stored in the `smbpasswd(5)` file. There are several disadvantages to this approach for sites with large numbers of users (counted in the thousands).

- The first problem is that all lookups must be performed sequentially. Given that there are approximately two lookups per domain logon (one during initial logon validation and one for a session connection setup, such as when mapping a network drive or printer), this is a performance bottleneck for large sites. What is needed is an indexed approach such as that used in databases.
- The second problem is that administrators who desire to replicate an smbpasswd file to more than one Samba server are left to use external tools such as **rsync(1)** and **ssh(1)** and write custom, in-house scripts.
- Finally, the amount of information that is stored in an smbpasswd entry leaves no room for additional attributes such as a home directory, password expiration time, or even a relative identifier (RID).

As a result of these deficiencies, a more robust means of storing user attributes used by smbd was developed. The API that defines access to user accounts is commonly referred to as the samdb interface (previously, this was called the passdb API and is still so named in the Samba source code trees).

Samba provides an enhanced set of passdb backends that overcome the deficiencies of the smbpasswd plaintext database. These are tdbsam and ldapsam. Of these, ldapsam will be of most interest to large corporate or enterprise sites.

11.4.3 tdbsam

Samba can store user and machine account data in a "TDB" (trivial database). Using this backend does not require any additional configuration. This backend is recommended for new installations that do not require LDAP.

As a general guide, the Samba Team does not recommend using the tdbsam backend for sites that have 250 or more users. Additionally, tdbsam is not capable of scaling for use in sites that require PDB/BDC implementations that require replication of the account database. Clearly, for reason of scalability, the use of ldapsam should be encouraged.

The recommendation of a 250-user limit is purely based on the notion that this would generally involve a site that has routed networks, possibly spread across more than one physical location. The Samba Team has not at this time established the performance-based scalability limits of the tdbsam architecture.

There are sites that have thousands of users and yet require only one server. One site recently reported having 4,500 user accounts on one UNIX system and reported excellent performance with the `tdbsam` passdb backend. The limitation of where the `tdbsam` passdb backend can be used is not one pertaining to a limitation in the TDB storage system, it is based only on the need for a reliable distribution mechanism for the SambaSAMAccount backend.

11.4.4 ldapsam

There are a few points to stress that the ldapsam does not provide. The LDAP support referred to in this documentation does not include:

- A means of retrieving user account information from a Windows 200x Active Directory server.
- A means of replacing /etc/passwd.

The second item can be accomplished by using LDAP NSS and PAM modules. LGPL versions of these libraries can be obtained from PADL Software[2]. More information about the configuration of these packages may be found in *LDAP, System Administration* by Gerald Carter, Chapter 6, Replacing NIS"[3].

This document describes how to use an LDAP directory for storing Samba user account information traditionally stored in the smbpasswd(5) file. It is assumed that the reader already has a basic understanding of LDAP concepts and has a working directory server already installed. For more information on LDAP architectures and directories, please refer to the following sites:

- OpenLDAP[4]
- Sun One Directory Server[5]
- Novell eDirectory[6]
- IBM Tivoli Directory Server[7]
- Red Hat Directory Server[8]
- Fedora Directory Server[9]

Two additional Samba resources that may prove to be helpful are:

- The Samba-PDC-LDAP-HOWTO[10] maintained by Ignacio Coupeau.
- The NT migration scripts from IDEALX[11] that are geared to manage users and groups in such a Samba-LDAP domain controller configu-

[2] <http://www.padl.com/>
[3] <http://safari.oreilly.com/?XmlId=1-56592-491-6>
[4] <http://www.openldap.org/>
[5] <http://www.sun.com/software/products/directory_srvr_ee/index.xml>
[6] <http://www.novell.com/products/edirectory/>
[7] <http://www-306.ibm.com/software/tivoli/products/directory-server/>
[8] <http://www.redhat.com/software/rha/directory/>
[9] <http://www.linuxsecurity.com/content/view/119229>
[10] <http://www.unav.es/cti/ldap-smb/ldap-smb-3-howto.html>
[11] <http://samba.idealx.org/>

ration. Idealx also produced the smbldap-tools and the Interactive Console Management tool.

11.4.4.1 Supported LDAP Servers

The LDAP ldapsam code was developed and tested using the OpenLDAP 2.x server and client libraries. The same code should work with Netscape's Directory Server and client SDK. However, there are bound to be compile errors and bugs. These should not be hard to fix. Please submit fixes via the process outlined in Chapter 40, "Reporting Bugs".

Samba is capable of working with any standards-compliant LDAP server.

11.4.4.2 Schema and Relationship to the RFC 2307 posixAccount

Samba-3.0 includes the necessary schema file for OpenLDAP 2.x in the `examples/LDAP/samba.schema` directory of the source code distribution tarball. The schema entry for the sambaSamAccount ObjectClass is shown here:

```
ObjectClass (1.3.6.1.4.1.7165.2.2.6 NAME 'sambaSamAccount' SUP top AUXILIARY
    DESC 'Samba-3.0 Auxiliary SAM Account'
    MUST ( uid $ sambaSID )
    MAY  ( cn $ sambaLMPassword $ sambaNTPassword $ sambaPwdLastSet $
           sambaLogonTime $ sambaLogoffTime $ sambaKickoffTime $
           sambaPwdCanChange $ sambaPwdMustChange $ sambaAcctFlags $
           displayName $ sambaHomePath $ sambaHomeDrive $ sambaLogonScript $
           sambaProfilePath $ description $ sambaUserWorkstations $
           sambaPrimaryGroupSID $ sambaDomainName ))
```

The `samba.schema` file has been formatted for OpenLDAP 2.0/2.1. The Samba Team owns the OID space used by the above schema and recommends its use. If you translate the schema to be used with Netscape DS, please submit the modified schema file as a patch to jerry@samba.org[12].

Just as the smbpasswd file is meant to store information that provides information additional to a user's `/etc/passwd` entry, so is the sambaSamAccount object meant to supplement the UNIX user account information. A

[12] <mailto:jerry@samba.org>

sambaSamAccount is an `AUXILIARY` ObjectClass, so it can be used to augment existing user account information in the LDAP directory, thus providing information needed for Samba account handling. However, there are several fields (e.g., uid) that overlap with the posixAccount ObjectClass outlined in RFC 2307. This is by design.

In order to store all user account information (UNIX and Samba) in the directory, it is necessary to use the sambaSamAccount and posixAccount ObjectClasses in combination. However, **smbd** will still obtain the user's UNIX account information via the standard C library calls, such as getpwnam(). This means that the Samba server must also have the LDAP NSS library installed and functioning correctly. This division of information makes it possible to store all Samba account information in LDAP, but still maintain UNIX account information in NIS while the network is transitioning to a full LDAP infrastructure.

11.4.4.3 OpenLDAP Configuration

To include support for the sambaSamAccount object in an OpenLDAP directory server, first copy the samba.schema file to slapd's configuration directory. The samba.schema file can be found in the directory `examples/LDAP` in the Samba source distribution.

```
root# cp samba.schema /etc/openldap/schema/
```

Next, include the `samba.schema` file in `slapd.conf`. The sambaSamAccount object contains two attributes that depend on other schema files. The *`uid`* attribute is defined in `cosine.schema` and the *`displayName`* attribute is defined in the `inetorgperson.schema` file. Both of these must be included before the `samba.schema` file.

```
## /etc/openldap/slapd.conf

## schema files (core.schema is required by default)
include             /etc/openldap/schema/core.schema

## needed for sambaSamAccount
include            /etc/openldap/schema/cosine.schema
```

```
include            /etc/openldap/schema/inetorgperson.schema
include            /etc/openldap/schema/nis.schema
include            /etc/openldap/schema/samba.schema
....
```

It is recommended that you maintain some indices on some of the most useful attributes, as in the following example, to speed up searches made on sambaSamAccount ObjectClasses (and possibly posixAccount and posixGroup as well):

```
# Indices to maintain
## required by OpenLDAP
index objectclass             eq

index cn                      pres,sub,eq
index sn                      pres,sub,eq
## required to support pdb_getsampwnam
index uid                     pres,sub,eq
## required to support pdb_getsambapwrid()
index displayName             pres,sub,eq

## uncomment these if you are storing posixAccount and
## posixGroup entries in the directory as well
##index uidNumber               eq
##index gidNumber               eq
##index memberUid               eq

index   sambaSID              eq
index   sambaPrimaryGroupSID  eq
index   sambaDomainName       eq
index   default               sub
```

Create the new index by executing:

```
root# ./sbin/slapindex -f slapd.conf
```

Remember to restart slapd after making these changes:

```
root# /etc/init.d/slapd restart
```

11.4.4.4 Initialize the LDAP Database

Before you can add accounts to the LDAP database, you must create the account containers that they will be stored in. The following LDIF file should be modified to match your needs (DNS entries, and so on):

```
# Organization for Samba Base
dn: dc=quenya,dc=org
objectclass: dcObject
objectclass: organization
dc: quenya
o: Quenya Org Network
description: The Samba-3 Network LDAP Example

# Organizational Role for Directory Management
dn: cn=Manager,dc=quenya,dc=org
objectclass: organizationalRole
cn: Manager
description: Directory Manager

# Setting up container for Users OU
dn: ou=People,dc=quenya,dc=org
objectclass: top
objectclass: organizationalUnit
ou: People

# Setting up admin handle for People OU
dn: cn=admin,ou=People,dc=quenya,dc=org
cn: admin
objectclass: top
objectclass: organizationalRole
objectclass: simpleSecurityObject
userPassword: {SSHA}c3ZM9tBaBo9autm1dL3waDS21+JSfQVz

# Setting up container for groups
```

```
dn: ou=Groups,dc=quenya,dc=org
objectclass: top
objectclass: organizationalUnit
ou: Groups

# Setting up admin handle for Groups OU
dn: cn=admin,ou=Groups,dc=quenya,dc=org
cn: admin
objectclass: top
objectclass: organizationalRole
objectclass: simpleSecurityObject
userPassword: {SSHA}c3ZM9tBaBo9autm1dL3waDS21+JSfQVz

# Setting up container for computers
dn: ou=Computers,dc=quenya,dc=org
objectclass: top
objectclass: organizationalUnit
ou: Computers

# Setting up admin handle for Computers OU
dn: cn=admin,ou=Computers,dc=quenya,dc=org
cn: admin
objectclass: top
objectclass: organizationalRole
objectclass: simpleSecurityObject
userPassword: {SSHA}c3ZM9tBaBo9autm1dL3waDS21+JSfQVz
```

The userPassword shown above should be generated using **slappasswd**.

The following command will then load the contents of the LDIF file into the LDAP database.

```
$ slapadd -v -l initldap.dif
```

Do not forget to secure your LDAP server with an adequate access control list as well as an admin password.

NOTE

Before Samba can access the LDAP server, you need to store the LDAP admin password in the Samba-3 `secrets.tdb` database by:

```
root# smbpasswd -w secret
```

11.4.4.5 Configuring Samba

The following parameters are available in `smb.conf` only if your version of Samba was built with LDAP support. Samba automatically builds with LDAP support if the LDAP libraries are found. The best method to verify that Samba was built with LDAP support is:

```
root#  smbd -b | grep LDAP
   HAVE_LDAP_H
   HAVE_LDAP
   HAVE_LDAP_DOMAIN2HOSTLIST
   HAVE_LDAP_INIT
   HAVE_LDAP_INITIALIZE
   HAVE_LDAP_SET_REBIND_PROC
   HAVE_LIBLDAP
   LDAP_SET_REBIND_PROC_ARGS
```

If the build of the **smbd** command you are using does not produce output that includes `HAVE_LDAP_H` it is necessary to discover why the LDAP headers and libraries were not found during compilation.

LDAP-related smb.conf options include these:

```
passdb backend = ldapsam:url
ldap admin dn
ldap delete dn
ldap filter
ldap group suffix
```

```
        ldap idmap suffix
        ldap machine suffix
        ldap passwd sync
        ldap ssl
        ldap suffix
        ldap user suffix
        ldap replication sleep
        ldap timeout
        ldap page size
```

These are described in the `smb.conf` man page and so are not repeated here. However, an example for use with an LDAP directory is shown in Example 11.4.1

11.4.4.6 Accounts and Groups Management

Because user accounts are managed through the sambaSamAccount ObjectClass, you should modify your existing administration tools to deal with sambaSamAccount attributes.

Machine accounts are managed with the sambaSamAccount ObjectClass, just like user accounts. However, it is up to you to store those accounts in a different tree of your LDAP namespace. You should use "ou=Groups,dc=quenya,dc=org" to store groups and "ou=People,dc=quenya,dc=org" to store users. Just configure your NSS and PAM accordingly (usually, in the `/etc/openldap/sldap.conf` configuration file).

In Samba-3, the group management system is based on POSIX groups. This means that Samba makes use of the posixGroup ObjectClass. For now, there is no NT-like group system management (global and local groups). Samba-3 knows only about `Domain Groups` and, unlike MS Windows 2000 and Active Directory, Samba-3 does not support nested groups.

11.4.4.7 Security and sambaSamAccount

There are two important points to remember when discussing the security of sambaSAMAccount entries in the directory.

- *Never* retrieve the SambaLMPassword or SambaNTPassword attribute values over an unencrypted LDAP session.

Example 11.4.1 Configuration with LDAP

```
[global]
        security = user
        encrypt passwords = yes
        netbios name = MORIA
        workgroup = NOLDOR
# LDAP related parameters:
# Define the DN used when binding to the LDAP servers.
# The password for this DN is not stored in smb.conf
# Set it using 'smbpasswd -w secret' to store the
# passphrase in the secrets.tdb file.
# If the "ldap admin dn" value changes, it must be reset ↩
    .
        ldap admin dn = "cn=Manager,dc=quenya,dc=org"
# SSL directory connections can be configured by:
# ('off', 'start tls', or 'on' (default))
        ldap ssl = start tls
# syntax: passdb backend = ldapsam:ldap://server-name[: ↩
    port]
        passdb backend = ldapsam:ldap://frodo.quenya.org
# smbpasswd -x delete the entire dn-entry
        ldap delete dn = no
# The machine and user suffix are added to the base ↩
    suffix
# wrote WITHOUT quotes. NULL suffixes by default
        ldap user suffix = ou=People
        ldap group suffix = ou=Groups
        ldap machine suffix = ou=Computers
# Trust UNIX account information in LDAP
# (see the smb.conf man page for details)
# Specify the base DN to use when searching the ↩
    directory
        ldap suffix = dc=quenya,dc=org
```

- *Never* allow non-admin users to view the SambaLMPassword or SambaNTPassword attribute values.

These password hashes are clear-text equivalents and can be used to impersonate the user without deriving the original clear-text strings. For more information on the details of LM/NT password hashes, refer to Chapter 11, "Account Information Databases".

To remedy the first security issue, the *ldap ssl* `smb.conf` parameter defaults to require an encrypted session (*ldap ssl* = on) using the default port of `636` when contacting the directory server. When using an OpenLDAP server, it is possible to use the StartTLS LDAP extended operation in the place of LDAPS. In either case, you are strongly encouraged to use secure communications protocols (so do not set *ldap ssl* = off).

Note that the LDAPS protocol is deprecated in favor of the LDAPv3 StartTLS extended operation. However, the OpenLDAP library still provides support for the older method of securing communication between clients and servers.

The second security precaution is to prevent non-administrative users from harvesting password hashes from the directory. This can be done using the following ACL in `slapd.conf`:

```
## allow the "ldap admin dn" access, but deny everyone else
access to attrs=SambaLMPassword,SambaNTPassword
     by dn="cn=Samba Admin,ou=People,dc=quenya,dc=org" write
     by * none
```

11.4.4.8 LDAP Special Attributes for sambaSamAccounts

The sambaSamAccount ObjectClass is composed of the attributes shown in next tables: Table 11.3, and Table 11.4.

The majority of these parameters are only used when Samba is acting as a PDC of a domain (refer to Chapter 4, "Domain Control", for details on how to configure Samba as a PDC). The following four attributes are only stored with the sambaSamAccount entry if the values are non-default values:

- sambaHomePath
- sambaLogonScript
- sambaProfilePath
- sambaHomeDrive

These attributes are only stored with the sambaSamAccount entry if the values are non-default values. For example, assume MORIA has now been configured as a PDC and that *logon home* = \\%L\%u was defined in its

smb.conf file. When a user named "becky" logs on to the domain, the *logon home* string is expanded to \\MORIA\becky. If the smbHome attribute exists in the entry "uid=becky,ou=People,dc=samba,dc=org", this value is used. However, if this attribute does not exist, then the value of the *logon home* parameter is used in its place. Samba will only write the attribute value to the directory entry if the value is something other than the default (e.g., \\MOBY\becky).

11.4.4.9 Example LDIF Entries for a sambaSamAccount

The following is a working LDIF that demonstrates the use of the SambaSamAccount ObjectClass:

```
dn: uid=guest2, ou=People,dc=quenya,dc=org
sambaLMPassword: 878D8014606CDA29677A44EFA1353FC7
sambaPwdMustChange: 2147483647
sambaPrimaryGroupSID: S-1-5-21-2447931902-1787058256-3961074038-513
sambaNTPassword: 552902031BEDE9EFAAD3B435B51404EE
sambaPwdLastSet: 1010179124
sambaLogonTime: 0
objectClass: sambaSamAccount
uid: guest2
sambaKickoffTime: 2147483647
sambaAcctFlags: [UX         ]
sambaLogoffTime: 2147483647
sambaSID: S-1-5-21-2447931902-1787058256-3961074038-5006
sambaPwdCanChange: 0
```

The following is an LDIF entry for using both the sambaSamAccount and posixAccount ObjectClasses:

```
dn: uid=gcarter, ou=People,dc=quenya,dc=org
sambaLogonTime: 0
displayName: Gerald Carter
sambaLMPassword: 552902031BEDE9EFAAD3B435B51404EE
sambaPrimaryGroupSID: S-1-5-21-2447931902-1787058256-3961074038-1201
objectClass: posixAccount
objectClass: sambaSamAccount
```

```
sambaAcctFlags: [UX         ]
userPassword: {crypt}BpM2ej8Rkzogo
uid: gcarter
uidNumber: 9000
cn: Gerald Carter
loginShell: /bin/bash
logoffTime: 2147483647
gidNumber: 100
sambaKickoffTime: 2147483647
sambaPwdLastSet: 1010179230
sambaSID: S-1-5-21-2447931902-1787058256-3961074038-5004
homeDirectory: /home/moria/gcarter
sambaPwdCanChange: 0
sambaPwdMustChange: 2147483647
sambaNTPassword: 878D8014606CDA29677A44EFA1353FC7
```

11.4.4.10 Password Synchronization

Samba-3 and later can update the non-Samba (LDAP) password stored with an account. When using pam_ldap, this allows changing both UNIX and Windows passwords at once.

The *ldap passwd sync* options can have the values shown in Table 11.5.

More information can be found in the `smb.conf` man page.

11.4.4.11 Using OpenLDAP Overlay for Password Synchronization

Howard Chu has written a special overlay called **smbk5pwd**. This tool modifies the `SambaNTPassword`, `SambaLMPassword` and `Heimdal` hashes in an OpenLDAP entry when an LDAP_EXOP_X_MODIFY_PASSWD operation is performed.

The overlay is shipped with OpenLDAP-2.3 and can be found in the `contrib/slapd-modules/smbk5pwd` subdirectory. This module can also be used with OpenLDAP-2.2.

11.5 Common Errors

11.5.1 Users Cannot Logon

"I've installed Samba, but now I can't log on with my UNIX account!"

Make sure your user has been added to the current Samba *passdb backend*. Read the Section 11.3 for details.

11.5.2 Configuration of auth methods

When explicitly setting an *auth methods* parameter, `guest` must be specified as the first entry on the line — for example, *auth methods* = guest sam.

Table 11.3 Attributes in the sambaSamAccount ObjectClass (LDAP), Part A

`sambaLMPassword`	The LanMan password 16-byte hash stored as a character representation of a hexadecimal string.
`sambaNTPassword`	The NT password 16-byte hash stored as a character representation of a hexadecimal string.
`sambaPwdLastSet`	The integer time in seconds since 1970 when the `sambaLMPassword` and `sambaNTPassword` attributes were last set.
`sambaAcctFlags`	String of 11 characters surrounded by square brackets [ ] representing account flags such as U (user), W (workstation), X (no password expiration), I (domain trust account), H (home dir required), S (server trust account), and D (disabled).
`sambaLogonTime`	Integer value currently unused.
`sambaLogoffTime`	Integer value currently unused.
`sambaKickoffTime`	Specifies the time (UNIX time format) when the user will be locked down and cannot login any longer. If this attribute is omitted, then the account will never expire. Using this attribute together with shadowExpire of the shadowAccount ObjectClass will enable accounts to expire completely on an exact date.
`sambaPwdCanChange`	Specifies the time (UNIX time format) after which the user is allowed to change his password. If this attribute is not set, the user will be free to change his password whenever he wants.
`sambaPwdMustChange`	Specifies the time (UNIX time format) when the user is forced to change his password. If this value is set to 0, the user will have to change his password at first login. If this attribute is not set, then the password will never expire.
`sambaHomeDrive`	Specifies the drive letter to which to map the UNC path specified by sambaHomePath. The drive letter must be specified in the form "X:" where X is the letter of the drive to map. Refer to the "logon drive" parameter in the smb.conf(5) man page for more information.
`sambaLogonScript`	The sambaLogonScript property specifies the path of the user's logon script, .CMD, .EXE, or .BAT file. The string can be null. The path is relative to the netlogon share. Refer to the *logon script* parameter in the `smb.conf` man page for more information.
`sambaProfilePath`	Specifies a path to the user's profile. This value can be a null string, a local absolute path, or a UNC path. Refer to the *logon path* parameter in

Table 11.4 Attributes in the sambaSamAccount ObjectClass (LDAP), Part B

`sambaUserWorkstations`	Here you can give a comma-separated list of machines on which the user is allowed to login. You may observe problems when you try to connect to a Samba domain member. Because domain members are not in this list, the domain controllers will reject them. Where this attribute is omitted, the default implies no restrictions.
`sambaSID`	The security identifier(SID) of the user. The Windows equivalent of UNIX UIDs.
`sambaPrimaryGroupSID`	The security identifier (SID) of the primary group of the user.
`sambaDomainName`	Domain the user is part of.

Table 11.5 Possible ldap passwd sync Values

Value	Description
yes	When the user changes his password, update `SambaNTPassword`, `SambaLMPassword`, and the `password` fields.
no	Only update `SambaNTPassword` and `SambaLMPassword`.
only	Only update the LDAP password and let the LDAP server worry about the other fields. This option is only available on some LDAP servers and only when the LDAP server supports LDAP_EXOP_X_MODIFY_PASSWD.

Chapter 12

GROUP MAPPING: MS WINDOWS AND UNIX

Starting with Samba-3, new group mapping functionality is available to create associations between Windows group SIDs and UNIX group GIDs. The **groupmap** subcommand included with the net tool can be used to manage these associations.

The new facility for mapping NT groups to UNIX system groups allows the administrator to decide which NT domain groups are to be exposed to MS Windows clients. Only those NT groups that map to a UNIX group that has a value other than the default (`-1`) will be exposed in group selection lists in tools that access domain users and groups.

WARNING

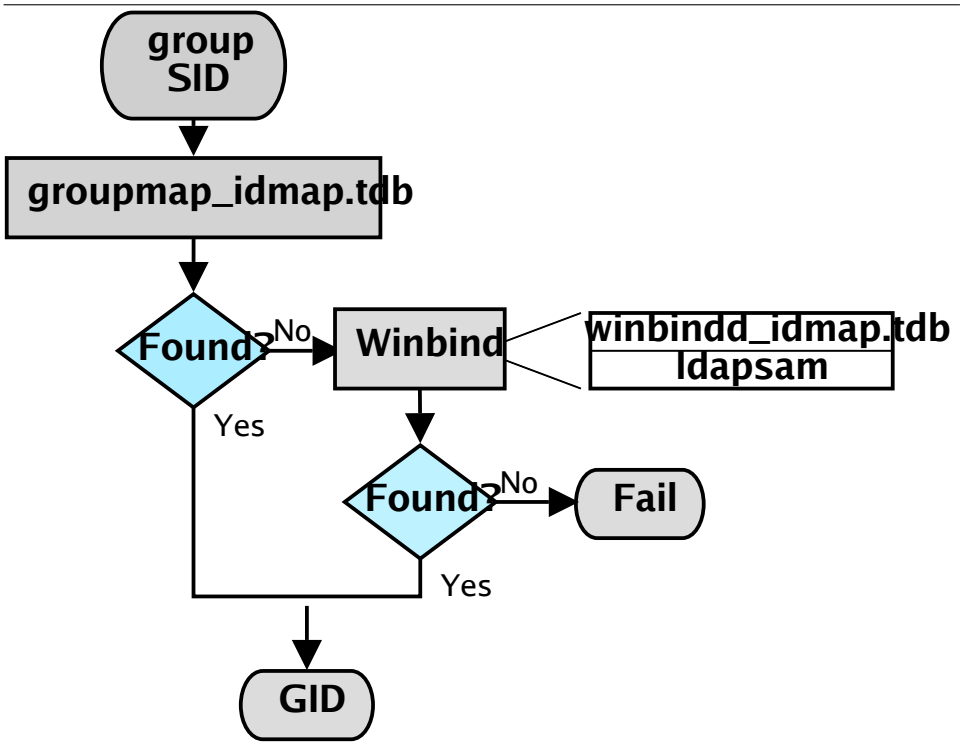
The *`domain admin group`* parameter has been removed in Samba-3 and should no longer be specified in `smb.conf`. In Samba-2.2.x, this parameter was used to give the listed users membership in the `Domain Admins` Windows group, which gave local admin rights on their workstations (in default configurations).

12.1 Features and Benefits

Samba allows the administrator to create MS Windows NT4/200x group accounts and to arbitrarily associate them with UNIX/Linux group accounts.

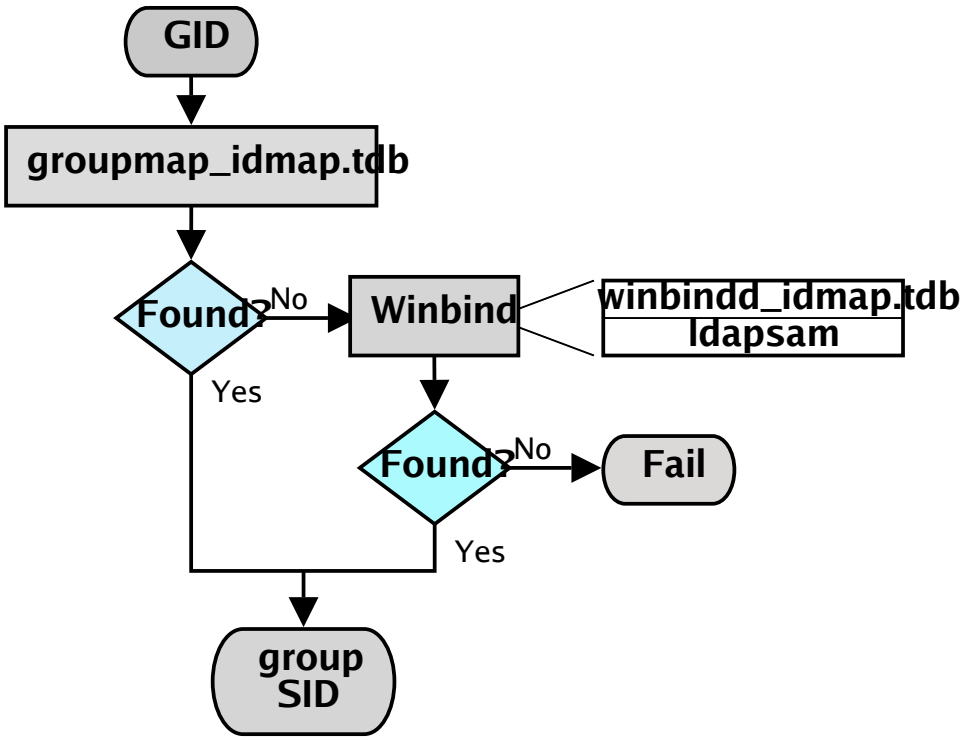
Group accounts can be managed using the MS Windows NT4 or MS Windows 200x/XP Professional MMC tools. Appropriate interface scripts should be provided in `smb.conf` if it is desired that UNIX/Linux system accounts should be automatically created when these tools are used. In the absence of these scripts, and so long as **winbindd** is running, Samba group accounts that are created using these tools will be allocated UNIX UIDs and GIDs from the ID range specified by the *idmap uid*/*idmap gid* parameters in the `smb.conf` file.

Figure 12.1 IDMAP: Group SID-to-GID Resolution.



In both cases, when winbindd is not running, only locally resolvable groups can be recognized. Please refer to Figure 12.1 and Figure 12.2. The **net**

Figure 12.2 IDMAP: GID Resolution to Matching SID.



groupmap is used to establish UNIX group to NT SID mappings as shown in Figure 12.3.

Figure 12.3 IDMAP Storing Group Mappings.

Administrators should be aware that where `smb.conf` group interface scripts make direct calls to the UNIX/Linux system tools (the shadow utilities, **groupadd**, **groupdel**, and **groupmod**), the resulting UNIX/Linux group names will be subject to any limits imposed by these tools. If the tool does not allow uppercase characters or space characters, then the creation of an MS Windows NT4/200x-style group of `Engineering Managers` will attempt to create an identically named UNIX/Linux group, an attempt that will of course fail.

There are several possible workarounds for the operating system tools limitation. One method is to use a script that generates a name for the UNIX/Linux system group that fits the operating system limits and that then just passes the UNIX/Linux group ID (GID) back to the calling Samba interface. This will provide a dynamic workaround solution.

Another workaround is to manually create a UNIX/Linux group, then manually create the MS Windows NT4/200x group on the Samba server, and then use the **net groupmap** tool to connect the two to each other.

12.2 Discussion

When you install MS Windows NT4/200x on a computer, the installation program creates default users and groups, notably the `Administrators` group, and gives that group privileges necessary to perform essential system tasks, such as the ability to change the date and time or to kill (or close) any process running on the local machine.

The `Administrator` user is a member of the `Administrators` group, and thus inherits `Administrators` group privileges. If a `joe` user is created to be a member of the `Administrators` group, `joe` has exactly the same rights as the user `Administrator`.

When an MS Windows NT4/200x/XP machine is made a domain member, the "Domain Admins" group of the PDC is added to the local `Administrators` group of the workstation. Every member of the `Domain Admins` group inherits the rights of the local `Administrators` group when logging on the workstation.

The following steps describe how to make Samba PDC users members of the `Domain Admins` group.

1. Create a UNIX group (usually in `/etc/group`); let's call it `domadm`.
2. Add to this group the users that must be "Administrators". For example, if you want `joe, john`, and `mary` to be administrators, your entry in `/etc/group` will look like this:

```
domadm:x:502:joe,john,mary
```

3. Map this domadm group to the "Domain Admins" group by executing the command:

   ```
   root# net groupmap add ntgroup="Domain Admins" unixgroup=domadm rid=512 type=d
   ```

   The quotes around "Domain Admins" are necessary due to the space in the group name. Also make sure to leave no white space surrounding the equal character (=).

Now `joe, john`, and `mary` are domain administrators.

It is possible to map any arbitrary UNIX group to any Windows NT4/200x group as well as to make any UNIX group a Windows domain group. For example, if you wanted to include a UNIX group (e.g., acct) in an ACL on a local file or printer on a Domain Member machine, you would flag that group as a domain group by running the following on the Samba PDC:

```
root# net groupmap add rid=1000 ntgroup="Accounting" unixgroup=acct type=d
```

The `ntgroup` value must be in quotes if it contains space characters to prevent the space from being interpreted as a command delimiter.

Be aware that the RID parameter is an unsigned 32-bit integer that should normally start at 1000. However, this RID must not overlap with any RID assigned to a user. Verification for this is done differently depending on the passdb backend you are using. Future versions of the tools may perform the verification automatically, but for now the burden is on you.

12.2.1 Warning: User Private Group Problems

Windows does not permit user and group accounts to have the same name. This has serious implications for all sites that use private group accounts. A private group account is an administrative practice whereby users are each given their own group account. Red Hat Linux, as well as several free distributions of Linux, by default create private groups.

When mapping a UNIX/Linux group to a Windows group account, all conflict can be avoided by assuring that the Windows domain group name does not overlap with any user account name.

12.2.2 Nested Groups: Adding Windows Domain Groups to Windows Local Groups

This functionality is known as `nested groups` and was first added to Samba-3.0.3.

All MS Windows products since the release of Windows NT 3.10 support the use of nested groups. Many Windows network administrators depend on this capability because it greatly simplifies security administration.

The nested group architecture was designed with the premise that day-to-day user and group membership management should be performed on the domain security database. The application of group security should be implemented on domain member servers using only local groups. On the domain member server, all file system security controls are then limited to use of the local groups, which will contain domain global groups and domain global users.

You may ask, What are the benefits of this arrangement? The answer is obvious to those who have plumbed the dark depths of Windows networking architecture. Consider for a moment a server on which are stored 200,000 files, each with individual domain user and domain group settings. The company that owns the file server is bought by another company, resulting in the server being moved to another location, and then it is made a member of a different domain. Who would you think now owns all the files and directories? Answer: Account Unknown.

Unraveling the file ownership mess is an unenviable administrative task that can be avoided simply by using local groups to control all file and directory access control. In this case, only the members of the local groups will have been lost. The files and directories in the storage subsystem will still be owned by the local groups. The same goes for all ACLs on them. It is administratively much simpler to delete the `Account Unknown` membership entries inside local groups with appropriate entries for domain global groups in the new domain that the server has been made a member of.

Another prominent example of the use of nested groups involves implementation of administrative privileges on domain member workstations and servers. Administrative privileges are given to all members of the built-in local group `Administrators` on each domain member machine. To ensure that all domain administrators have full rights on the member server or workstation, on joining the domain, the `Domain Admins` group is added to the local

Administrators group. Thus everyone who is logged into the domain as a member of the Domain Admins group is also granted local administrative privileges on each domain member.

UNIX/Linux has no concept of support for nested groups, and thus Samba has for a long time not supported them either. The problem is that you would have to enter UNIX groups as auxiliary members of a group in `/etc/group`. This does not work because it was not a design requirement at the time the UNIX file system security model was implemented. Since Samba-2.2, the winbind daemon can provide `/etc/group` entries on demand by obtaining user and group information from the domain controller that the Samba server is a member of.

In effect, Samba supplements the `/etc/group` data via the dynamic **libnss_winbind** mechanism. Beginning with Samba-3.0.3, this facility is used to provide local groups in the same manner as Windows. It works by expanding the local groups on the fly as they are accessed. For example, the `Domain Users` group of the domain is made a member of the local group `demo`. Whenever Samba needs to resolve membership of the `demo` local (alias) group, winbind asks the domain controller for demo members of the Domain Users group. By definition, it can only contain user objects, which can then be faked to be member of the UNIX/Linux group `demo`.

To enable the use of nested groups, **winbindd** must be used with NSS winbind. Creation and administration of the local groups is done best via the Windows Domain User Manager or its Samba equivalent, the utility **net rpc group**. Creating the local group `demo` is achieved by executing:

```
root#  net rpc group add demo -L -Uroot%not24get
```

Here the -L switch means that you want to create a local group. It may be necessary to add -S and -U switches for accessing the correct host with appropriate user or root privileges. Adding and removing group members can be done via the `addmem` and `delmem` subcommands of **net rpc group** command. For example, addition of "DOM\Domain Users" to the local group `demo` is done by executing:

```
net rpc group addmem demo "DOM\Domain Users"
```

Having completed these two steps, the execution of **getent group demo** will show demo members of the global `Domain Users` group as members of the group `demo`. This also works with any local or domain user. In case the domain DOM trusts another domain, it is also possible to add global users and groups of the trusted domain as members of `demo`. The users from the foreign domain who are members of the group that has been added to the `demo` group now have the same local access permissions as local domain users have.

12.2.3 Important Administrative Information

Administrative rights are necessary in two specific forms:

1. For Samba-3 domain controllers and domain member servers/clients.
2. To manage domain member Windows workstations.

Versions of Samba up to and including 3.0.10 do not provide a means for assigning rights and privileges that are necessary for system administration tasks from a Windows domain member client machine, so domain administration tasks such as adding, deleting, and changing user and group account information, and managing workstation domain membership accounts, can be handled by any account other than root.

Samba-3.0.11 introduced a new privilege management interface (see Chapter 15, "User Rights and Privileges") that permits these tasks to be delegated to non-root (i.e., accounts other than the equivalent of the MS Windows Administrator) accounts.

Administrative tasks on a Windows domain member workstation can be done by anyone who is a member of the `Domain Admins` group. This group can be mapped to any convenient UNIX group.

12.2.3.1 Applicable Only to Versions Earlier than 3.0.11

Administrative tasks on UNIX/Linux systems, such as adding users or groups, requires `root`-level privilege. The addition of a Windows client to a Samba domain involves the addition of a user account for the Windows client.

Many UNIX administrators continue to request that the Samba Team make it possible to add Windows workstations, or the ability to add, delete, or

modify user accounts, without requiring `root` privileges. Such a request violates every understanding of basic UNIX system security.

There is no safe way to provide access on a UNIX/Linux system without providing `root`-level privileges. Provision of `root` privileges can be done either by logging on to the Domain as the user `root` or by permitting particular users to use a UNIX account that has a UID=0 in the `/etc/passwd` database. Users of such accounts can use tools like the NT4 Domain User Manager and the NT4 Domain Server Manager to manage user and group accounts as well as domain member server and client accounts. This level of privilege is also needed to manage share-level ACLs.

12.2.4 Default Users, Groups, and Relative Identifiers

When first installed, Windows NT4/200x/XP are preconfigured with certain user, group, and alias entities. Each has a well-known RID. These must be preserved for continued integrity of operation. Samba must be provisioned with certain essential domain groups that require the appropriate RID value. When Samba-3 is configured to use `tdbsam`, the essential domain groups are automatically created. It is the LDAP administrator's responsibility to create (provision) the default NT groups.

Each essential domain group must be assigned its respective well-known RID. The default users, groups, aliases, and RIDs are shown in Table 12.1.

> **Note**
>
> It is the administrator's responsibility to create the essential domain groups and to assign each its default RID.

It is permissible to create any domain group that may be necessary; just make certain that the essential domain groups (well known) have been created and assigned their default RIDs. Other groups you create may be assigned any arbitrary RID you care to use.

Be sure to map each domain group to a UNIX system group. That is the only way to ensure that the group will be available for use as an NT domain

group.

Table 12.1 Well-Known User Default RIDs

Well-Known Entity	RID	Type	Essential
Domain Administrator	500	User	No
Domain Guest	501	User	No
Domain KRBTGT	502	User	No
Domain Admins	512	Group	Yes
Domain Users	513	Group	Yes
Domain Guests	514	Group	Yes
Domain Computers	515	Group	No
Domain Controllers	516	Group	No
Domain Certificate Admins	517	Group	No
Domain Schema Admins	518	Group	No
Domain Enterprise Admins	519	Group	No
Domain Policy Admins	520	Group	No
Builtin Admins	544	Alias	No
Builtin users	545	Alias	No
Builtin Guests	546	Alias	No
Builtin Power Users	547	Alias	No
Builtin Account Operators	548	Alias	No
Builtin System Operators	549	Alias	No
Builtin Print Operators	550	Alias	No
Builtin Backup Operators	551	Alias	No
Builtin Replicator	552	Alias	No
Builtin RAS Servers	553	Alias	No

12.2.5 Example Configuration

You can list the various groups in the mapping database by executing **net groupmap list**. Here is an example:

```
root#  net groupmap list
Domain Admins (S-1-5-21-2547222302-1596225915-2414751004-512) -> domadmin
Domain Users (S-1-5-21-2547222302-1596225915-2414751004-513) -> domuser
Domain Guests (S-1-5-21-2547222302-1596225915-2414751004-514) -> domguest
```

For complete details on **net groupmap**, refer to the net(8) man page.

12.3 Configuration Scripts

Everyone needs tools. Some of us like to create our own, others prefer to use canned tools (i.e., prepared by someone else for general use).

12.3.1 Sample smb.conf Add Group Script

A script to create complying group names for use by the Samba group interfaces is provided in Example 12.3.1. This script adds a temporary entry in the `/etc/group` file and then renames it to the desired name. This is an example of a method to get around operating system maintenance tool limitations such as those present in some version of the **groupadd** tool.

Example 12.3.1 smbgrpadd.sh

```
#!/bin/bash

# Add the group using normal system groupadd tool.
groupadd smbtmpgrp00

thegid=`cat /etc/group | grep ^smbtmpgrp00 | cut -d ":" -f3`

# Now change the name to what we want for the MS Windows networking end
cp /etc/group /etc/group.bak
cat /etc/group.bak | sed "s/^smbtmpgrp00/$1/g" > /etc/group
rm /etc/group.bak

# Now return the GID as would normally happen.
echo $thegid
exit 0
```

The `smb.conf` entry for the above script shown in Example 12.3.2 demonstrates how it may be used.

Example 12.3.2 Configuration of smb.conf for the add group Script

```
[global]
        add group script = /path_to_tool/smbgrpadd.sh "%g ↩
            "
```

12.3.2 Script to Configure Group Mapping

In our example we have created a UNIX/Linux group called `ntadmin`. Our script will create the additional groups `Orks`, `Elves`, and `Gnomes`. It is a good idea to save this shell script for later use just in case you ever need to rebuild your mapping database. For the sake of convenience we elect to save this script as a file called `initGroups.sh`. This script is given in Example 12.3.3.

Example 12.3.3 Script to Set Group Mapping

```
#!/bin/bash

net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin rid=512 type=d
net groupmap add ntgroup="Domain Users" unixgroup=users rid=513 type=d
net groupmap add ntgroup="Domain Guests" unixgroup=nobody rid=514 type=d

groupadd Orks
groupadd Elves
groupadd Gnomes

net groupmap add ntgroup="Orks"   unixgroup=Orks   type=d
net groupmap add ntgroup="Elves"  unixgroup=Elves  type=d
net groupmap add ntgroup="Gnomes" unixgroup=Gnomes type=d
```

Of course it is expected that the administrator will modify this to suit local needs. For information regarding the use of the **net groupmap** tool please refer to the man page.

NOTE

Versions of Samba-3 prior to 3.0.23 automatically create default group mapping for the `Domain Admins`, `Domain Users` and `Domain Guests` Windows groups, but do not map them to UNIX GIDs. This was a cause of administrative confusion and trouble. Commencing with Samba-3.0.23 this annomaly has been fixed - thus all Windows groups must now be manually and explicitly created and mapped to a valid UNIX GID by the Samba administrator.

12.4 Common Errors

At this time there are many little surprises for the unwary administrator. In a real sense it is imperative that every step of automated control scripts be carefully tested manually before putting it into active service.

12.4.1 Adding Groups Fails

This is a common problem when the **groupadd** is called directly by the Samba interface script for the *add group script* in the `smb.conf` file.

The most common cause of failure is an attempt to add an MS Windows group account that has an uppercase character and/or a space character in it.

There are three possible workarounds. First, use only group names that comply with the limitations of the UNIX/Linux **groupadd** system tool. Second, it involves the use of the script mentioned earlier in this chapter, and third is the option is to manually create a UNIX/Linux group account that can substitute for the MS Windows group name, then use the procedure listed above to map that group to the MS Windows group.

12.4.2 Adding Domain Users to the Workstation Power Users Group

"What must I do to add domain users to the Power Users group?"

The Power Users group is a group that is local to each Windows 200x/XP Professional workstation. You cannot add the Domain Users group to the Power Users group automatically, it must be done on each workstation by logging in as the local workstation *administrator* and then using the following procedure:

1. Click **Start** -> **Control Panel** -> **Users and Passwords**.
2. Click the **Advanced** tab.
3. Click the **Advanced** button.
4. Click `Groups`.
5. Double-click `Power Users`. This will launch the panel to add users or groups to the local machine `Power Users` group.
6. Click the **Add** button.
7. Select the domain from which the `Domain Users` group is to be added.
8. Double-click the `Domain Users` group.
9. Click the **OK** button. If a logon box is presented during this process, please remember to enter the connect as `DOMAIN\UserName`, that is, for the domain `MIDEARTH` and the user `root` enter `MIDEARTH\root`.

Chapter 13

REMOTE AND LOCAL MANAGEMENT: THE NET COMMAND

The **net** command is one of the new features of Samba-3 and is an attempt to provide a useful tool for the majority of remote management operations necessary for common tasks. The **net** tool is flexible by design and is intended for command-line use as well as for scripted control application.

Originally introduced with the intent to mimic the Microsoft Windows command that has the same name, the **net** command has morphed into a very powerful instrument that has become an essential part of the Samba network administrator's toolbox. The Samba Team has introduced tools, such as **smbgroupedit** and **rpcclient**, from which really useful capabilities have been integrated into the **net**. The **smbgroupedit** command was absorbed entirely into the **net**, while only some features of the **rpcclient** command have been ported to it. Anyone who finds older references to these utilities and to the functionality they provided should look at the **net** command before searching elsewhere.

A Samba-3 administrator cannot afford to gloss over this chapter because to do so will almost certainly cause the infliction of self-induced pain, agony, and desperation. Be warned: this is an important chapter.

13.1 Overview

The tasks that follow the installation of a Samba-3 server, whether standalone or domain member, of a domain controller (PDC or BDC) begins with the need to create administrative rights. Of course, the creation of user and group accounts is essential for both a standalone server and a PDC. In the case of a BDC or a Domain Member server (DMS), domain user and group accounts are obtained from the central domain authentication backend.

Regardless of the type of server being installed, local UNIX groups must be mapped to the Windows networking domain global group accounts. Do you ask why? Because Samba always limits its access to the resources of the host server by way of traditional UNIX UID and GID controls. This means that local groups must be mapped to domain global groups so that domain users who are members of the domain global groups can be given access rights based on UIDs and GIDs local to the server that is hosting Samba. Such mappings are implemented using the **net** command.

UNIX systems that are hosting a Samba-3 server that is running as a member (PDC, BDC, or DMS) must have a machine security account in the domain authentication database (or directory). The creation of such security (or trust) accounts is also handled using the **net** command.

The establishment of interdomain trusts is achieved using the **net** command also, as may a plethora of typical administrative duties such as user management, group management, share and printer management, file and printer migration, security identifier management, and so on.

The overall picture should be clear now: the **net** command plays a central role on the Samba-3 stage. This role will continue to be developed. The inclusion of this chapter is evidence of its importance, one that has grown in complexity to the point that it is no longer considered prudent to cover its use fully in the online UNIX man pages.

13.2 Administrative Tasks and Methods

The basic operations of the **net** command are documented here. This documentation is not exhaustive, and thus it is incomplete. Since the primary focus is on migration from Windows servers to a Samba server, the emphasis is on the use of the Distributed Computing Environment Remote Procedure

Call (DCE RPC) mode of operation. When used against a server that is a member of an Active Directory domain, it is preferable (and often necessary) to use ADS mode operations. The **net** command supports both, but not for every operation. For most operations, if the mode is not specified, **net** will automatically fall back via the `ads`, `rpc`, and `rap` modes. Please refer to the man page for a more comprehensive overview of the capabilities of this utility.

13.3 UNIX and Windows Group Management

As stated, the focus in most of this chapter is on use of the **net rpc** family of operations that are supported by Samba. Most of them are supported by the **net ads** mode when used in connection with Active Directory. The **net rap** operating mode is also supported for some of these operations. RAP protocols are used by IBM OS/2 and by several earlier SMB servers.

Samba's **net** tool implements sufficient capability to permit all common administrative tasks to be completed from the command line. In this section each of the essential user and group management facilities are explored.

Samba-3 recognizes two types of groups: *domain groups* and *local groups*. Domain groups can contain (have as members) only domain user accounts. Local groups can contain local users, domain users, and domain groups as members.

The purpose of a local group is to permit file permission to be set for a group account that, like the usual UNIX/Linux group, is persistent across redeployment of a Windows file server.

13.3.1 Adding, Renaming, or Deletion of Group Accounts

Samba provides file and print services to Windows clients. The file system resources it makes available to the Windows environment must, of necessity, be provided in a manner that is compatible with the Windows networking environment. UNIX groups are created and deleted as required to serve operational needs in the UNIX operating system and its file systems.

In order to make available to the Windows environment, Samba has a facility by which UNIX groups can be mapped to a logical entity, called a Windows (or domain) group. Samba supports two types of Windows groups, local and

global. Global groups can contain as members, global users. This membership is affected in the normal UNIX manner, but adding UNIX users to UNIX groups. Windows user accounts consist of a mapping between a user SambaSAMAccount (logical entity) and a UNIX user account. Therefore, a UNIX user is mapped to a Windows user (i.e., is given a Windows user account and password) and the UNIX groups to which that user belongs, is mapped to a Windows group account. The result is that in the Windows account environment that user is also a member of the Windows group account by virtue of UNIX group memberships.

The following sub-sections that deal with management of Windows groups demonstrates the relationship between the UNIX group account and its members to the respective Windows group accounts. It goes on to show how UNIX group members automatically pass-through to Windows group membership as soon as a logical mapping has been created.

13.3.1.1 Adding or Creating a New Group

Before attempting to add a Windows group account, the currently available groups can be listed as shown here:

```
root#  net rpc group list -Uroot%not24get
Password:
Domain Admins
Domain Users
Domain Guests
Print Operators
Backup Operators
Replicator
Domain Computers
Engineers
```

A Windows group account called "SupportEngrs" can be added by executing the following command:

```
root#  net rpc group add "SupportEngrs" -Uroot%not24get
```

The addition will result in immediate availability of the new group account as validated by executing this command:

```
root#  net rpc group list -Uroot%not24get
Password:
Domain Admins
Domain Users
Domain Guests
Print Operators
Backup Operators
Replicator
Domain Computers
Engineers
SupportEngrs
```

The following demonstrates that the POSIX (UNIX/Linux system account) group has been created by calling the *add group script* = /opt/IDEALX/sbin/smbldap-groupadd -p "%g" interface script:

```
root#  getent group
...
Domain Admins:x:512:root
Domain Users:x:513:jht,lct,ajt,met
Domain Guests:x:514:
Print Operators:x:550:
Backup Operators:x:551:
Replicator:x:552:
Domain Computers:x:553:
Engineers:x:1002:jht
SupportEngrs:x:1003:
```

The following demonstrates that the use of the **net** command to add a group account results in immediate mapping of the POSIX group that has been created to the Windows group account as shown here:

```
root#  net groupmap list
Domain Admins (S-1-5-21-72630-4128915-11681869-512) -> Domain Admins
Domain Users (S-1-5-21-72630-4128915-11681869-513) -> Domain Users
Domain Guests (S-1-5-21-72630-4128915-11681869-514) -> Domain Guests
Print Operators (S-1-5-21-72630-4128915-11681869-550) -> Print Operators
Backup Operators (S-1-5-21-72630-4128915-11681869-551) -> Backup Operators
Replicator (S-1-5-21-72630-4128915-11681869-552) -> Replicator
Domain Computers (S-1-5-21-72630-4128915-11681869-553) -> Domain Computers
Engineers (S-1-5-21-72630-4128915-11681869-3005) -> Engineers
SupportEngrs (S-1-5-21-72630-4128915-11681869-3007) -> SupportEngrs
```

13.3.1.2 Mapping Windows Groups to UNIX Groups

Windows groups must be mapped to UNIX system (POSIX) groups so that file system access controls can be asserted in a manner that is consistent with the methods appropriate to the operating system that is hosting the Samba server.

All file system (file and directory) access controls, within the file system of a UNIX/Linux server that is hosting a Samba server, are implemented using a UID/GID identity tuple. Samba does not in any way override or replace UNIX file system semantics. Thus it is necessary that all Windows networking operations that access the file system provide a mechanism that maps a Windows user to a particular UNIX/Linux group account. The user account must also map to a locally known UID. Note that the **net** command does not call any RPC-functions here but directly accesses the passdb.

Samba depends on default mappings for the `Domain Admins, Domain Users`, and `Domain Guests` global groups. Additional groups may be added as shown in the examples just given. There are times when it is necessary to map an existing UNIX group account to a Windows group. This operation, in effect, creates a Windows group account as a consequence of creation of the mapping.

The operations that are permitted include: `add`, `modify`, and `delete`. An example of each operation is shown here.

> **Note**
>
> Commencing with Samba-3.0.23 Windows Domain Groups must be explicitly created. By default, all UNIX groups are exposed to Windows networking as Windows local groups.

An existing UNIX group may be mapped to an existing Windows group by this example:

```
root#  net groupmap modify ntgroup="Domain Users" unixgroup=users
```

An existing UNIX group may be mapped to a new Windows group as shown here:

```
root#  net groupmap add ntgroup="EliteEngrs" unixgroup=Engineers type=d
```

Supported mapping types are 'd' (domain global) and 'l' (domain local). A Windows group may be deleted, and then a new Windows group can be mapped to the UNIX group by executing these commands:

```
root#  net groupmap delete ntgroup=Engineers
root#  net groupmap add ntgroup=EngineDrivers unixgroup=Engineers type=d
```

The deletion and addition operations affected only the logical entities known as Windows groups, or domain groups. These operations are inert to UNIX system groups, meaning that they neither delete nor create UNIX system groups. The mapping of a UNIX group to a Windows group makes the UNIX group available as Windows groups so that files and folders on domain member clients (workstations and servers) can be given domain-wide access controls for domain users and groups.

Two types of Windows groups can be created: `domain (global)` and `local`. In the previous examples the Windows groups created were of type `domain` or global. The following command will create a Windows group of type `local`.

```
root#  net groupmap add ntgroup=Pixies unixgroup=pixies type=l
```

Supported mapping types are 'd' (domain global) and 'l' (domain local), a domain local group in Samba is treated as local to the individual Samba server. Local groups can be used with Samba to enable multiple nested group support.

13.3.1.3 Deleting a Group Account

A group account may be deleted by executing the following command:

```
root#  net rpc group delete SupportEngineers -Uroot%not24get
```

Validation of the deletion is advisable. The same commands may be executed as shown above.

13.3.1.4 Rename Group Accounts

> NOTE
>
> This command is not documented in the man pages; it is implemented in the source code, but it does not work at this time. The example given documents, from the source code, how it should work. Watch the release notes of a future release to see when this may have been fixed.

Sometimes it is necessary to rename a group account. Good administrators know how painful some managers' demands can be if this simple request

is ignored. The following command demonstrates how the Windows group "SupportEngrs" can be renamed to "CustomerSupport":

```
root#  net rpc group rename SupportEngrs \
    CustomerSupport -Uroot%not24get
```

13.3.2 Manipulating Group Memberships

Three operations can be performed regarding group membership. It is possible to (1) add Windows users to a Windows group, to (2) delete Windows users from Windows groups, and to (3) list the Windows users that are members of a Windows group.

To avoid confusion, it makes sense to check group membership before attempting to make any changes. The **getent group** will list UNIX/Linux group membership. UNIX/Linux group members are seen also as members of a Windows group that has been mapped using the **net groupmap** command (see Chapter 12, "Group Mapping: MS Windows and UNIX"). The following list of UNIX/Linux group membership shows that the user `ajt` is a member of the UNIX/Linux group `Engineers`.

```
root#  getent group
...
Domain Admins:x:512:root
Domain Users:x:513:jht,lct,ajt,met,vlendecke
Domain Guests:x:514:
Print Operators:x:550:
Backup Operators:x:551:
Replicator:x:552:
Domain Computers:x:553:
Engineers:x:1000:jht,ajt
```

The UNIX/Linux groups have been mapped to Windows groups, as is shown here:

```
root#  net groupmap list
Domain Admins (S-1-5-21-72630-412605-116429-512) -> Domain Admins
```

```
Domain Users (S-1-5-21-72630-412605-116429-513) -> Domain Users
Domain Guests (S-1-5-21-72630-412605-116429-514) -> Domain Guests
Print Operators (S-1-5-21-72630-412605-116429-550) -> Print Operators
Backup Operators (S-1-5-21-72630-412605-116429-551) -> Backup Operators
Replicator (S-1-5-21-72630-412605-116429-552) -> Replicator
Domain Computers (S-1-5-21-72630-412605-116429-553) -> Domain Computers
Engineers (S-1-5-21-72630-412605-116429-3001) -> Engineers
```

Given that the user `ajt` is already a member of the UNIX/Linux group and, via the group mapping, a member of the Windows group, an attempt to add this account again should fail. This is demonstrated here:

```
root#  net rpc group addmem "MIDEARTH\Engineers" ajt -Uroot%not24get
Could not add ajt to MIDEARTH\Engineers: NT_STATUS_MEMBER_IN_GROUP
```

This shows that the group mapping between UNIX/Linux groups and Windows groups is effective and transparent.

To permit the user `ajt` to be added using the **net rpc group** utility, this account must first be removed. The removal and confirmation of its effect is shown here:

```
root#  net rpc group delmem "MIDEARTH\Engineers" ajt -Uroot%not24get
root#  getent group Engineers
Engineers:x:1000:jht
root#  net rpc group members Engineers -Uroot%not24get
MIDEARTH\jht
```

In this example both at the UNIX/Linux system level, the group no longer has the `ajt` as a member. The above also shows this to be the case for Windows group membership.

The account is now added again, using the **net rpc group** utility:

```
root#  net rpc group addmem "MIDEARTH\Engineers" ajt -Uroot%not24get
root#  getent group Engineers
Engineers:x:1000:jht,ajt
root#  net rpc group members Engineers -Uroot%not24get
```

```
MIDEARTH\jht
MIDEARTH\ajt
```

In this example the members of the Windows `Domain Users` account are validated using the **net rpc group** utility. Note the this contents of the UNIX/Linux group was shown four paragraphs earlier. The Windows (domain) group membership is shown here:

```
root#  net rpc group members "Domain Users" -Uroot%not24get
MIDEARTH\jht
MIDEARTH\lct
MIDEARTH\ajt
MIDEARTH\met
MIDEARTH\vlendecke
```

This express example shows that Windows group names are treated by Samba (as with MS Windows) in a case-insensitive manner:

```
root#  net rpc group members "DomAiN USerS" -Uroot%not24get
MIDEARTH\jht
MIDEARTH\lct
MIDEARTH\ajt
MIDEARTH\met
MIDEARTH\vlendecke
```

NOTE

An attempt to specify the group name as `MIDEARTH\Domain Users` in place of just simply `Domain Users` will fail. The default behavior of the net rpc group is to direct the command at the local machine. The Windows group is treated as being local to the machine. If it is necessary to query another machine, its name can be specified using the `-S servername` parameter to the **net** command.

13.3.3 Nested Group Support

It is possible in Windows (and now in Samba also) to create a local group that has members (contains), domain users, and domain global groups. Creation of the local group `demo` is achieved by executing:

```
root#  net rpc group add demo -L -S MORDON -Uroot%not24get
```

The -L switch means create a local group. Use the -S argument to direct the operation to a particular server. The parameters to the -U argument should be for a user who has appropriate administrative right and privileges on the machine.

Addition and removal of group members can be achieved using the `addmem` and `delmem` subcommands of **net rpc group** command. For example, addition of "DOM\Domain Users" to the local group `demo` would be done by executing:

```
root#  net rpc group addmem demo "DOM\Domain Users" -Uroot%not24get
```

The members of a nested group can be listed by executing the following:

```
root#  net rpc group members demo -Uroot%not24get
DOM\Domain Users
```

```
DOM\Engineers
DOM\jamesf
DOM\jht
```

Nested group members can be removed (deleted) as shown here:

```
root#  net rpc group delmem demo "DOM\jht" -Uroot%not24get
```

13.3.3.1 Managing Nest Groups on Workstations from the Samba Server

Windows network administrators often ask on the Samba mailing list how it is possible to grant everyone administrative rights on their own workstation. This is of course a very bad practice, but commonly done to avoid user complaints. Here is how it can be done remotely from a Samba PDC or BDC:

```
root#  net rpc group addmem "Administrators" "Domain Users" \
    -S WINPC032 -Uadministrator%secret
```

This can be scripted, and can therefore be performed as a user logs onto the domain from a Windows workstation. Here is a simple example that shows how this can be done. Automating User Addition to the Workstation Power Users Group

Example 13.3.1 Script to Auto-add Domain Users to Workstation Power Users Group

```
#!/bin/bash

/usr/bin/net rpc group addmem "Power Users" "DOMAIN_NAME\$1" \
                    -UAdministrator%secret -S $2

exit 0
```

1. Create the script shown in Example 13.3.1 and locate it in the directory `/etc/samba/scripts`, named as `autopoweruser.sh`.

Example 13.3.2 A Magic Netlogon Share

```
[netlogon]
        comment = Netlogon Share
        path = /var/lib/samba/netlogon
        root preexec = /etc/samba/scripts/autopoweruser. ↩
            sh %U %m
        read only = Yes
        guest ok = Yes
```

2. Set the permissions on this script to permit it to be executed as part of the logon process:

```
root#  chown root:root /etc/samba/autopoweruser.sh
root#  chmod 755 /etc/samba/autopoweruser.sh
```

3. Modify the `smb.conf` file so the `NETLOGON` stanza contains the parameters shown in Example 13.3.2.
4. Ensure that every Windows workstation Administrator account has the same password that you have used in the script shown in Example 13.3.2

This script will be executed every time a user logs on to the network. Therefore every user will have local Windows workstation management rights. This could of course be assigned using a group, in which case there is little justification for the use of this procedure. The key justification for the use of this method is that it will guarantee that all users have appropriate rights on the workstation.

13.4 UNIX and Windows User Management

Every Windows network user account must be translated to a UNIX/Linux user account. In actual fact, the only account information the UNIX/Linux Samba server needs is a UID. The UID is available either from a system (POSIX) account or from a pool (range) of UID numbers that is set aside for the purpose of being allocated for use by Windows user accounts. In the case of the UID pool, the UID for a particular user will be allocated by **winbindd**.

Although this is not the appropriate place to discuss the *username map* facility, this interface is an important method of mapping a Windows user account to a UNIX account that has a different name. Refer to the man page for the `smb.conf` file for more information regarding this facility. User name mappings cannot be managed using the **net** utility.

13.4.1 Adding User Accounts

The syntax for adding a user account via the **net** (according to the man page) is shown here:

```
net [<method>] user ADD <name> [-c container] [-F user flags] \
    [misc. options] [targets]
```

The user account password may be set using this syntax:

```
net rpc password <username> [<password>] -Uadmin_username%admin_pass
```

The following demonstrates the addition of an account to the server `FRODO`:

```
root#  net rpc user add jacko -S FRODO -Uroot%not24get
Added user jacko
```

The account password can be set with the following methods (all show the same operation):

```
root#  net rpc password jacko f4sth0rse -S FRODO -Uroot%not24get
root#  net rpc user password jacko f4sth0rse \
    -S FRODO -Uroot%not24get
```

13.4.2 Deletion of User Accounts

Deletion of a user account can be done using the following syntax:

```
net [<method>] user DELETE <name> [misc. options] [targets]
```

The following command will delete the user account `jacko`:

```
root#  net rpc user delete jacko -Uroot%not24get
Deleted user account
```

13.4.3 Managing User Accounts

Two basic user account operations are routinely used: change of password and querying which groups a user is a member of. The change of password operation is shown in Section 13.4.1.

The ability to query Windows group membership can be essential. Here is how a remote server may be interrogated to find which groups a user is a member of:

```
root#  net rpc user info jacko -S SAURON -Uroot%not24get
net rpc user info jacko -S SAURON -Uroot%not24get
Domain Users
Domain Admins
Engineers
TorridGroup
BOP Shop
Emergency Services
```

It is also possible to rename user accounts: oldusername newusername Note that this operation does not yet work against Samba Servers. It is, however, possible to rename useraccounts on Windows Servers.

13.4.4 User Mapping

In some situations it is unavoidable that a user's Windows logon name will differ from the login ID that user has on the Samba server. It is possible to create a special file on the Samba server that will permit the Windows

user name to be mapped to a different UNIX/Linux user name. The `smb.conf` file must also be amended so that the `[global]` stanza contains the parameter:

```
username map = /etc/samba/smbusers
```

The content of the `/etc/samba/smbusers` file is shown here:

```
parsonsw: "William Parsons"
marygee: geeringm
```

In this example the Windows user account "William Parsons" will be mapped to the UNIX user `parsonsw`, and the Windows user account "geeringm" will be mapped to the UNIX user `marygee`.

13.5 Administering User Rights and Privileges

With all versions of Samba earlier than 3.0.11 the only account on a Samba server that could manage users, groups, shares, printers, and such was the `root` account. This caused problems for some users and was a frequent source of scorn over the necessity to hand out the credentials for the most security-sensitive account on a UNIX/Linux system.

New to Samba version 3.0.11 is the ability to delegate administrative privileges as necessary to either a normal user or to groups of users. The significance of the administrative privileges is documented in Chapter 15, "User Rights and Privileges". Examples of use of the **net** for user rights and privilege management is appropriate to this chapter.

NOTE

When user rights and privileges are correctly set, there is no longer a need for a Windows network account for the `root` user (nor for any synonym of it) with a UNIX UID=0. Initial user rights and privileges can be assigned by any account that is a member of the `Domain Admins` group. Rights can be assigned to user as well as group accounts.

By default, no privileges and rights are assigned. This is demonstrated by executing the command shown here:

```
root#  net rpc rights list accounts -U root%not24get
BUILTIN\Print Operators
No privileges assigned

BUILTIN\Account Operators
No privileges assigned

BUILTIN\Backup Operators
No privileges assigned

BUILTIN\Server Operators
No privileges assigned

BUILTIN\Administrators
No privileges assigned

Everyone
No privileges assigned
```

The **net** command can be used to obtain the currently supported capabilities for rights and privileges using this method:

```
root#  net rpc rights list -U root%not24get
     SeMachineAccountPrivilege  Add machines to domain
```

```
      SePrintOperatorPrivilege  Manage printers
           SeAddUsersPrivilege  Add users and groups to the domain
     SeRemoteShutdownPrivilege  Force shutdown from a remote system
       SeDiskOperatorPrivilege  Manage disk shares
             SeBackupPrivilege  Back up files and directories
            SeRestorePrivilege  Restore files and directories
      SeTakeOwnershipPrivilege  Take ownership of files or other objects
```

Machine account privilege is necessary to permit a Windows NT4 or later network client to be added to the domain. The disk operator privilege is necessary to permit the user to manage share ACLs and file and directory ACLs for objects not owned by the user.

In this example, all rights are assigned to the `Domain Admins` group. This is a good idea since members of this group are generally expected to be all-powerful. This assignment makes that the reality:

```
root#  net rpc rights grant "MIDEARTH\Domain Admins" \
    SeMachineAccountPrivilege SePrintOperatorPrivilege \
    SeAddUsersPrivilege SeRemoteShutdownPrivilege \
    SeDiskOperatorPrivilege  -U root%not24get
Successfully granted rights.
```

Next, the domain user `jht` is given the privileges needed for day-to-day administration:

```
root#  net rpc rights grant "MIDEARTH\jht" \
    SeMachineAccountPrivilege SePrintOperatorPrivilege \
    SeAddUsersPrivilege SeDiskOperatorPrivilege \
    -U root%not24get
Successfully granted rights.
```

The following step permits validation of the changes just made:

```
root#  net rpc rights list accounts -U root%not24get
MIDEARTH\jht
SeMachineAccountPrivilege
```

```
SePrintOperatorPrivilege
SeAddUsersPrivilege
SeDiskOperatorPrivilege

BUILTIN\Print Operators
No privileges assigned

BUILTIN\Account Operators
No privileges assigned

BUILTIN\Backup Operators
No privileges assigned

BUILTIN\Server Operators
No privileges assigned

BUILTIN\Administrators
No privileges assigned

Everyone
No privileges assigned

MIDEARTH\Domain Admins
SeMachineAccountPrivilege
SePrintOperatorPrivilege
SeAddUsersPrivilege
SeRemoteShutdownPrivilege
SeDiskOperatorPrivilege
```

13.6 Managing Trust Relationships

There are essentially two types of trust relationships: the first is between domain controllers and domain member machines (network clients), the second is between domains (called interdomain trusts). All Samba servers that participate in domain security require a domain membership trust account, as do like Windows NT/200x/XP workstations.

13.6.1 Machine Trust Accounts

The net command looks in the `smb.conf` file to obtain its own configuration settings. Thus, the following command 'knows' which domain to join from the `smb.conf` file.

A Samba server domain trust account can be validated as shown in this example:

```
root#  net rpc testjoin
Join to 'MIDEARTH' is OK
```

Where there is no domain membership account, or when the account credentials are not valid, the following results will be observed:

```
net rpc testjoin -S DOLPHIN
Join to domain 'WORLDOCEAN' is not valid
```

The equivalent command for joining a Samba server to a Windows ADS domain is shown here:

```
root#  net ads testjoin
Using short domain name -- TAKEAWAY
Joined 'LEMONADE' to realm 'TAKEAWAY.BIZ'
```

In the event that the ADS trust was not established, or is broken for one reason or another, the following error message may be obtained:

```
root#  net ads testjoin -UAdministrator%secret
Join to domain is not valid
```

The following demonstrates the process of creating a machine trust account in the target domain for the Samba server from which the command is executed:

```
root#  net rpc join -S FRODO -Uroot%not24get
```

```
Joined domain MIDEARTH.
```

The joining of a Samba server to a Samba domain results in the creation of a machine account. An example of this is shown here:

```
root#  pdbedit -Lw merlin\$
merlin$:1009:9B4489D6B90461FD6A3EC3AB96147E16:\
176D8C554E99914BDF3407DEA2231D80:[S          ]:LCT-42891919:
```

The S in the square brackets means this is a server (PDC/BDC) account. The domain join can be cast to join purely as a workstation, in which case the S is replaced with a W (indicating a workstation account). The following command can be used to affect this:

```
root#  net rpc join member -S FRODO -Uroot%not24get
Joined domain MIDEARTH.
```

Note that the command-line parameter `member` makes this join specific. By default the type is deduced from the `smb.conf` file configuration. To specifically join as a PDC or BDC, the command-line parameter will be `[PDC | BDC]`. For example:

```
root#  net rpc join bdc -S FRODO -Uroot%not24get
Joined domain MIDEARTH.
```

It is best to let Samba figure out the domain join type from the settings in the `smb.conf` file.

The command to join a Samba server to a Windows ADS domain is shown here:

```
root#  net ads join -UAdministrator%not24get
Using short domain name -- GDANSK
Joined 'FRANDIMITZ' to realm 'GDANSK.ABMAS.BIZ'
```

There is no specific option to remove a machine account from an NT4 domain. When a domain member that is a Windows machine is withdrawn from the domain, the domain membership account is not automatically removed either. Inactive domain member accounts can be removed using any convenient tool. If necessary, the machine account can be removed using the following **net** command:

```
root#  net rpc user delete HERRING\$ -Uroot%not24get
Deleted user account.
```

The removal is made possible because machine accounts are just like user accounts with a trailing $ character. The account management operations treat user and machine accounts in like manner.

A Samba-3 server that is a Windows ADS domain member can execute the following command to detach from the domain:

```
root#  net ads leave
```

Detailed information regarding an ADS domain can be obtained by a Samba DMS machine by executing the following:

```
root#  net ads status
```

The volume of information is extensive. Please refer to the book "Samba-3 by Example", Chapter 7 for more information regarding its use. This book may be obtained either in print or online from the Samba-3 by Example[1].

13.6.2 Interdomain Trusts

Interdomain trust relationships form the primary mechanism by which users from one domain can be granted access rights and privileges in another domain.

To discover what trust relationships are in effect, execute this command:

[1] <http://www.samba.org/samba/docs/Samba3-ByExample.pdf>

```
root#  net rpc trustdom list -Uroot%not24get
Trusted domains list:

none

Trusting domains list:

none
```

There are no interdomain trusts at this time; the following steps will create them.

It is necessary to create a trust account in the local domain. A domain controller in a second domain can create a trusted connection with this account. That means that the foreign domain is being trusted to access resources in the local domain. This command creates the local trust account:

```
root#  net rpc trustdom add DAMNATION f00db4r -Uroot%not24get
```

The account can be revealed by using the **pdbedit** as shown here:

```
root#  pdbedit -Lw DAMNATION\$
DAMNATION$:1016:9AC1F121DF897688AAD3B435B51404EE: \
7F845808B91BB9F7FEF44B247D9DC9A6:[I         ]:LCT-428934B1:
```

A trust account will always have an I in the field within the square brackets.

If the trusting domain is not capable of being reached, the following command will fail:

```
root#  net rpc trustdom list -Uroot%not24get
Trusted domains list:

none

Trusting domains list:
```

```
DAMNATION           S-1-5-21-1385457007-882775198-1210191635
```

The above command executed successfully; a failure is indicated when the following response is obtained:

```
net rpc trustdom list -Uroot%not24get
Trusted domains list:

DAMNATION           S-1-5-21-1385457007-882775198-1210191635

Trusting domains list:

DAMNATION           domain controller is not responding
```

Where a trust account has been created on a foreign domain, Samba is able to establish the trust (connect with) the foreign account. In the process it creates a one-way trust to the resources on the remote domain. This command achieves the objective of joining the trust relationship:

```
root#  net rpc trustdom establish DAMNATION
Password: xxxxxxx == f00db4r
Could not connect to server TRANSGRESSION
Trust to domain DAMNATION established
```

Validation of the two-way trust now established is possible as shown here:

```
root#  net rpc trustdom list -Uroot%not24get
Trusted domains list:

DAMNATION           S-1-5-21-1385457007-882775198-1210191635

Trusting domains list:

DAMNATION           S-1-5-21-1385457007-882775198-1210191635
```

Sometimes it is necessary to remove the ability for local users to access a foreign domain. The trusting connection can be revoked as shown here:

```
root#  net rpc trustdom revoke DAMNATION -Uroot%not24get
```

At other times it becomes necessary to remove the ability for users from a foreign domain to be able to access resources in the local domain. The command shown here will do that:

```
root#  net rpc trustdom del DAMNATION -Uroot%not24get
```

13.7 Managing Security Identifiers (SIDS)

The basic security identifier that is used by all Windows networking operations is the Windows security identifier (SID). All Windows network machines (servers and workstations), users, and groups are identified by their respective SID. All desktop profiles are also encoded with user and group SIDs that are specific to the SID of the domain to which the user belongs.

It is truly prudent to store the machine and/or domain SID in a file for safekeeping. Why? Because a change in hostname or in the domain (workgroup) name may result in a change in the SID. When you have the SID on hand, it is a simple matter to restore it. The alternative is to suffer the pain of having to recover user desktop profiles and perhaps rejoin all member machines to the domain.

First, do not forget to store the local SID in a file. It is a good idea to put this in the directory in which the `smb.conf` file is also stored. Here is a simple action to achieve this:

```
root#  net getlocalsid > /etc/samba/my-sid
```

Good, there is now a safe copy of the local machine SID. On a PDC/BDC this is the domain SID also.

The following command reveals what the former one should have placed into the file called `my-sid`:

```
root#  net getlocalsid
SID for domain MERLIN is: S-1-5-21-726309263-4128913605-1168186429
```

If ever it becomes necessary to restore the SID that has been stored in the `my-sid` file, simply copy the SID (the string of characters that begins with `S-1-5-21`) to the command line shown here:

```
root#  net setlocalsid S-1-5-21-1385457007-882775198-1210191635
```

Restoration of a machine SID is a simple operation, but the absence of a backup copy can be very problematic.

The following operation is useful only for machines that are being configured as a PDC or a BDC. DMS and workstation clients should have their own machine SID to avoid any potential namespace collision. Here is the way that the BDC SID can be synchronized to that of the PDC (this is the default NT4 domain practice also):

```
root#  net rpc getsid -S FRODO -Uroot%not24get
Storing SID S-1-5-21-726309263-4128913605-1168186429 \
    for Domain MIDEARTH in secrets.tdb
```

Usually it is not necessary to specify the target server (-S FRODO) or the administrator account credentials (-Uroot%not24get).

13.8 Share Management

Share management is central to all file serving operations. Typical share operations include:

- Creation/change/deletion of shares
- Setting/changing ACLs on shares
- Moving shares from one server to another
- Change of permissions of share contents

Each of these are dealt with here insofar as they involve the use of the **net** command. Operations outside of this command are covered elsewhere in this document.

13.8.1 Creating, Editing, and Removing Shares

A share can be added using the **net rpc share** command capabilities. The target machine may be local or remote and is specified by the -S option. It must be noted that the addition and deletion of shares using this tool depends on the availability of a suitable interface script. The interface scripts Sambas **smbd** uses are called *add share command*, *delete share command* and *change share command* A set of example scripts are provided in the Samba source code tarball in the directory `~samba/examples/scripts`.

The following steps demonstrate the use of the share management capabilities of the **net** utility. In the first step a share called `Bulge` is added. The sharepoint within the file system is the directory `/data`. The command that can be executed to perform the addition of this share is shown here:

```
root#  net rpc share add Bulge=/data -S MERLIN -Uroot%not24get
```

Validation is an important process, and by executing the command **net rpc share** with no other operators it is possible to obtain a listing of available shares, as shown here:

```
root#  net rpc share -S MERLIN -Uroot%not24get
profdata
archive
Bulge   <--- This one was added
print$
netlogon
profiles
IPC$
kyocera
ADMIN$
```

Often it is desirable also to permit a share to be removed using a command-line tool. The following step permits the share that was previously added to be removed:

```
root#  net rpc share delete Bulge -S MERLIN -Uroot%not24get
```

A simple validation shown here demonstrates that the share has been removed:

```
root#  net rpc share -S MERLIN -Uroot%not24get
profdata
archive
print$
netlogon
profiles
IPC$
ADMIN$
kyocera
```

13.8.2 Creating and Changing Share ACLs

At this time the **net** tool cannot be used to manage ACLs on Samba shares. In MS Windows language this is called Share Permissions.

It is possible to set ACLs on Samba shares using either the SRVTOOLS NT4 Domain Server Manager or using the Computer Management MMC snap-in. Neither is covered here, but see Chapter 16, "File, Directory, and Share Access Controls".

13.8.3 Share, Directory, and File Migration

Shares and files can be migrated in the same manner as user, machine, and group accounts. It is possible to preserve access control settings (ACLs) as well as security settings throughout the migration process. The **net rpc vampire** facility is used to migrate accounts from a Windows NT4 (or later) domain to a Samba server. This process preserves passwords and account security settings and is a precursor to the migration of shares and files.

The **net rpc share** command may be used to migrate shares, directories, files, and all relevant data from a Windows server to a Samba server.

A set of command-line switches permit the creation of almost direct clones of Windows file servers. For example, when migrating a fileserver, file ACLs and DOS file attributes from the Windows server can be included in the migration process and will reappear, almost identically, on the Samba server when the migration has been completed.

The migration process can be completed only with the Samba server already being fully operational. The user and group accounts must be migrated before attempting to migrate data share, files, and printers. The migration of files and printer configurations involves the use of both SMB and MS DCE RPC services. The benefit of the manner in which the migration process has been implemented is that the possibility now exists to use a Samba server as a man-in-middle migration service that affects a transfer of data from one server to another. For example, if the Samba server is called MESSER, the source Windows NT4 server is called PEPPY, and the target Samba server is called GONZALES, the machine MESSER can be used to effect the migration of all data (files and shares) from PEPPY to GONZALES. If the target machine is not specified, the local server is assumed by default - as net's general rule of thumb .

The success of server migration requires a firm understanding of the structure of the source server (or domain) as well as the processes on which the migration is critically dependant.

There are two known limitations to the migration process:

1. The **net** command requires that the user credentials provided exist on both the migration source and the migration target.
2. Printer settings may not be fully or may be incorrectly migrated. This might in particular happen when migrating a Windows 2003 print server to Samba.

13.8.3.1 Share Migration

The **net rpc share migrate** command operation permits the migration of plain share stanzas. A stanza contains the parameters within which a file or print share are defined. The use of this migration method will create share stanzas that have as parameters the file system directory path, an optional

description, and simple security settings that permit write access to files. One of the first steps necessary following migration is to review the share stanzas to ensure that the settings are suitable for use.

The shares are created on the fly as part of the migration process. The **smbd** application does this by calling on the operating system to execute the script specified by the `smb.conf` parameter *`add share command`*.

There is a suitable example script for the *`add share command`* in the `$SAMBA_SOURCES/examples/scripts` directory. It should be noted that the account that is used to drive the migration must, of necessity, have appropriate file system access privileges and have the right to create shares and to set ACLs on them. Such rights are conferred by these rights: *`SeAddUsersPrivilege`* and *`SeDiskOperatorPrivilege`*. For more information regarding rights and privileges please refer to Chapter 15, "User Rights and Privileges".

The syntax of the share migration command is shown here:

```
net rpc share MIGRATE SHARES <share-name> -S <source>
        [--destination=localhost] [--exclude=share1,share2] [-v]
```

When the parameter <share-name> is omitted, all shares will be migrated. The potentially large list of available shares on the system that is being migrated can be limited using the *`--exclude`* switch. For example:

```
root#  net rpc share migrate shares myshare\
         -S win2k -U administrator%secret"
```

This will migrate the share `myshare` from the server `win2k` to the Samba Server using the permissions that are tied to the account `administrator` with the password `secret`. The account that is used must be the same on both the migration source server and the target Samba server. The use of the **net rpc vampire**, prior to attempting the migration of shares, will ensure that accounts will be identical on both systems. One precaution worth taking before commencement of migration of shares is to validate that the migrated accounts (on the Samba server) have the needed rights and privileges. This can be done as shown here:

```
root#  net rpc right list accounts -Uroot%not24get
```

The steps taken so far perform only the migration of shares. Directories and directory contents are not migrated by the steps covered up to this point.

13.8.3.2 File and Directory Migration

Everything covered to this point has been done in preparation for the migration of file and directory data. For many people preparation is potentially boring and the real excitement only begins when file data can be used. The next steps demonstrate the techniques that can be used to transfer (migrate) data files using the **net** command.

Transfer of files from one server to another has always been a challenge for MS Windows administrators because Windows NT and 200X servers do not always include the tools needed. The **xcopy** from Windows NT is not capable of preserving file and directory ACLs, it does so only with Windows 200x. Microsoft does provide a utility that can copy ACLs (security settings) called **scopy**, but it is provided only as part of the Windows NT or 200X Server Resource Kit.

There are several tools, both commercial and freeware, that can be used from a Windows server to copy files and directories with full preservation of security settings. One of the best known of the free tools is called **robocopy**.

The **net** utility can be used to copy files and directories with full preservation of ACLs as well as DOS file attributes. Note that including ACLs makes sense only where the destination system will operate within the same security context as the source system. This applies both to a DMS and to domain controllers that result from a vampired domain. Before file and directory migration, all shares must already exist.

The syntax for the migration commands is shown here:

```
net rpc share MIGRATE FILES <share-name> -S <source>
    [--destination=localhost] [--exclude=share1,share2]
    [--acls] [--attrs] [--timestamps] [-v]
```

If the <share-name> parameter is omitted, all shares will be migrated. The potentially large list of shares on the source system can be restricted using the *--exclude* command switch.

Where it is necessary to preserve all file ACLs, the `--acls` switch should be added to the above command line. Original file timestamps can be preserved by specifying the `--timestamps` switch, and the DOS file attributes (i.e., hidden, archive, etc.) can be preserved by specifying the `--attrs` switch.

> **Note**
>
> The ability to preserve ACLs depends on appropriate support for ACLs as well as the general file system semantics of the host operating system on the target server. A migration from one Windows file server to another will perfectly preserve all file attributes. Because of the difficulty of mapping Windows ACLs onto a POSIX ACLs-supporting system, there can be no perfect migration of Windows ACLs to a Samba server.

The ACLs that result on a Samba server will most probably not match the originating ACLs. Windows supports the possibility of files that are owned only by a group. Group-alone file ownership is not possible under UNIX/Linux. Errors in migrating group-owned files can be avoided by using the `smb.conf` file *force unknown acl user* = yes parameter. This facility will automatically convert group-owned files into correctly user-owned files on the Samba server.

An example for migration of files from a machine called `nt4box` to the Samba server from which the process will be handled is shown here:

```
root#  net rpc share migrate files -S nt4box --acls \
    --attrs -U administrator%secret
```

This command will migrate all files and directories from all file shares on the Windows server called `nt4box` to the Samba server from which migration is initiated. Files that are group-owned will be owned by the user account `administrator`.

13.8.3.3 Share-ACL Migration

It is possible to have share-ACLs (security descriptors) that won't allow you, even as Administrator, to copy any files or directories into it. Therefor the migration of the share-ACLs has been put into a separate function:

```
root#  net rpc share migrate security -S nt4box -U administrator%secret
```

This command will only copy the share-ACL of each share on nt4box to your local samba-system.

13.8.3.4 Simultaneous Share and File Migration

The operating mode shown here is just a combination of the previous three. It first migrates share definitions and then all shared files and directories and finally migrates the share-ACLs:

```
net rpc share MIGRATE ALL <share-name> -S <source>
    [--exclude=share1, share2] [--acls] [--attrs] [--timestamps] [-v]
```

An example of simultaneous migration is shown here:

```
root#  net rpc share migrate all -S w2k3server -U administrator%secret
```

This will generate a complete server clone of the *w2k3server* server.

13.8.4 Printer Migration

The installation of a new server, as with the migration to a new network environment, often is similar to building a house; progress is very rapid from the laying of foundations up to the stage at which the house can be locked up, but the finishing off appears to take longer and longer as building approaches completion.

Printing needs vary greatly depending on the network environment and may be very simple or complex. If the need is very simple, the best solution to the

implementation of printing support may well be to re-install everything from a clean slate instead of migrating older configurations. On the other hand, a complex network that is integrated with many international offices and a multiplexity of local branch offices, each of which form an inter-twined maze of printing possibilities, the ability to migrate all printer configurations is decidedly beneficial. To manually re-establish a complex printing network will take much time and frustration. Often it will not be possible to find driver files that are currently in use, necessitating the installation of newer drivers. Newer drivers often implement printing features that will necessitate a change in the printer usage. Additionally, with very complex printer configurations it becomes almost impossible to re-create the same environment — no matter how extensively it has been documented.

The migration of an existing printing architecture involves the following:

- Establishment of print queues.
- Installation of printer drivers (both for the print server and for Windows clients.
- Configuration of printing forms.
- Implementation of security settings.
- Configuration of printer settings.

The Samba **net** utility permits printer migration from one Windows print server to another. When this tool is used to migrate printers to a Samba server **smbd**, the application that receives the network requests to create the necessary services must call out to the operating system in order to create the underlying printers. The call-out is implemented by way of an interface script that can be specified by the `smb.conf` file parameter . This script is essential to the migration process. A suitable example script may be obtained from the `$SAMBA_SOURCES/examples/scripts` directory. Take note that this script must be customized to suit the operating system environment and may use its tools to create a print queue.

Each of the components listed above can be completed separately, or they can be completed as part of an automated operation. Many network administrators prefer to deal with migration issues in a manner that gives them the most control, particularly when things go wrong. The syntax for each operation is now briefly described.

Printer migration from a Windows print server (NT4 or 200x) is shown. This instruction causes the printer share to be created together with the underlying print queue:

```
net rpc printer MIGRATE PRINTERS [printer] [misc. options] [targets]
```

Printer drivers can be migrated from the Windows print server to the Samba server using this command-line instruction:

```
net rpc printer MIGRATE DRIVERS [printer] [misc. options] [targets]
```

Printer forms can be migrated with the following operation:

```
net rpc printer MIGRATE FORMS [printer] [misc. options] [targets]
```

Printer security settings (ACLs) can be migrated from the Windows server to the Samba server using this command:

```
net rpc printer MIGRATE SECURITY [printer] [misc. options] [targets]
```

Printer configuration settings include factors such as paper size and default paper orientation. These can be migrated from the Windows print server to the Samba server with this command:

```
net rpc printer MIGRATE SETTINGS [printer] [misc. options] [targets]
```

Migration of printers including the above-mentioned sets of information may be completed with a single command using this syntax:

```
net rpc printer MIGRATE ALL [printer] [misc. options] [targets]
```

13.9 Controlling Open Files

The man page documents the **net file** function suite, which provides the tools to close open files using either RAP or RPC function calls. Please refer to the man page for specific usage information.

13.10 Session and Connection Management

The session management interface of the **net session** command uses the old RAP method to obtain the list of connections to the Samba server, as shown here:

```
root#  net rap session -S MERLIN -Uroot%not24get
Computer             User name            Client Type        Opens Idle time
------------------------------------------------------------------------------
\\merlin             root                 Unknown Client         0 00:00:00
\\marvel             jht                  Unknown Client         0 00:00:00
\\maggot             jht                  Unknown Client         0 00:00:00
\\marvel             jht                  Unknown Client         0 00:00:00
```

A session can be closed by executing a command as shown here:

```
root#  net rap session close marvel -Uroot%not24get
```

13.11 Printers and ADS

When Samba-3 is used within an MS Windows ADS environment, printers shared via Samba will not be browseable until they have been published to the ADS domain. Information regarding published printers may be obtained from the ADS server by executing the **net ads print info** command following this syntax:

```
net ads printer info <printer_name> <server_name> -Uadministrator%secret
```

If the asterisk (*) is used in place of the printer_name argument, a list of all printers will be returned.

To publish (make available) a printer to ADS, execute the following command:

```
net ads printer publish <printer_name> -Uadministrator%secret
```

This publishes a printer from the local Samba server to ADS.

Removal of a Samba printer from ADS is achieved by executing this command:

```
net ads printer remove <printer_name> -Uadministrator%secret
```

A generic search (query) can also be made to locate a printer across the entire ADS domain by executing:

```
net ads printer search <printer_name> -Uadministrator%secret
```

13.12 Manipulating the Samba Cache

Please refer to the **net** command man page for information regarding cache management.

13.13 Managing IDMAP UID/SID Mappings

The IDMAP UID to SID, and SID to UID, mappings that are created by **winbindd** can be backed up to a text file. The text file can be manually edited, although it is highly recommended that you attempt this only if you know precisely what you are doing.

An IDMAP text dump file can be restored (or reloaded). There are two situations that may necessitate this action: a) The existing IDMAP file is corrupt, b) It is necessary to install an editted version of the mapping information.

Winbind must be shut down to dump the IDMAP file. Before restoring a dump file, shut down **winbindd** and delete the old `winbindd_idmap.tdb` file.

13.13.1 Creating an IDMAP Database Dump File

The IDMAP database can be dumped to a text file as shown here:

```
net idmap dump <full_path_and_tdb_filename> > dumpfile.txt
```

Where a particular build of Samba the run-time tdb files are stored in the `/var/lib/samba` directory the following commands to create the dump file will suffice:

```
net idmap dump /var/lib/samba/winbindd_idmap.tdb > idmap_dump.txt
```

13.13.2 Restoring the IDMAP Database Dump File

The IDMAP dump file can be restored using the following command:

```
net idmap restore idmap_dump.txt
```

Where the Samba run-time tdb files are stored in the `/var/lib/samba` directory the following command can be used to restore the data to the tdb file:

```
net idmap restore /var/lib/samba/winbindd_idmap.tdb < idmap_dump.txt
```

13.14 Other Miscellaneous Operations

The following command is useful for obtaining basic statistics regarding a Samba domain. This command does not work with current Windows XP Professional clients.

```
root#  net rpc info
Domain Name: RAPIDFLY
Domain SID: S-1-5-21-399034208-633907489-3292421255
Sequence number: 1116312355
Num users: 720
Num domain groups: 27
Num local groups: 6
```

Another useful tool is the **net time** tool set. This tool may be used to query the current time on the target server as shown here:

```
root#  net time -S SAURON
Tue May 17 00:50:43 2005
```

In the event that it is the intent to pass the time information obtained to the UNIX **/bin/time**, it is a good idea to obtain the time from the target server in a format that is ready to be passed through. This may be done by executing:

```
root#  net time system -S FRODO
051700532005.16
```

The time can be set on a target server by executing:

```
root#  net time set -S MAGGOT -U Administrator%not24get
Tue May 17 00:55:30 MDT 2005
```

It is possible to obtain the time zone of a server by executing the following command against it:

```
root#  net time zone -S SAURON
-0600
```

Chapter 14

IDENTITY MAPPING (IDMAP)

The Microsoft Windows operating system has a number of features that impose specific challenges to interoperability with the operating systems on which Samba is implemented. This chapter deals explicitly with the mechanisms Samba-3 (version 3.0.8 and later) uses to overcome one of the key challenges in the integration of Samba servers into an MS Windows networking environment. This chapter deals with identity mapping (IDMAP) of Windows security identifiers (SIDs) to UNIX UIDs and GIDs.

To ensure sufficient coverage, each possible Samba deployment type is discussed. This is followed by an overview of how the IDMAP facility may be implemented.

The IDMAP facility is of concern where more than one Samba server (or Samba network client) is installed in a domain. Where there is a single Samba server, do not be too concerned regarding the IDMAP infrastructure — the default behavior of Samba is nearly always sufficient. Where mulitple Samba servers are used it is often necessary to move data off one server and onto another, and that is where the fun begins!

Where user and group account information is stored in an LDAP directory every server can have the same consistent UID and GID for users and groups. This is achieved using NSS and the nss_ldap tool. Samba can be configured to use only local accounts, in which case the scope of the IDMAP problem is somewhat reduced. This works reasonably well if the servers belong to a single domain, and interdomain trusts are not needed. On the other hand, if the Samba servers are NT4 domain members, or ADS domain members, or if there is a need to keep the security name-space separate (i.e., the user

`DOMINICUS\FJones` must not be given access to the account resources of the user `FRANCISCUS\FJones`[1] free from inadvertent cross-over, close attention should be given to the way that the IDMAP facility is configured.

The use of IDMAP is important where the Samba server will be accessed by workstations or servers from more than one domain, in which case it is important to run winbind so it can handle the resolution (ID mapping) of foreign SIDs to local UNIX UIDs and GIDs.

The use of the IDMAP facility requires the execution of the **winbindd** upon Samba startup.

14.1 Samba Server Deployment Types and IDMAP

There are four basic server deployment types, as documented in Chapter 3, "Server Types and Security Modes".

14.1.1 Standalone Samba Server

A standalone Samba server is an implementation that is not a member of a Windows NT4 domain, a Windows 200X Active Directory domain, or a Samba domain.

By definition, this means that users and groups will be created and controlled locally, and the identity of a network user must match a local UNIX/Linux user login. The IDMAP facility is therefore of little to no interest, winbind will not be necessary, and the IDMAP facility will not be relevant or of interest.

14.1.2 Domain Member Server or Domain Member Client

Samba-3 can act as a Windows NT4 PDC or BDC, thereby providing domain control protocols that are compatible with Windows NT4. Samba-3 file and print sharing protocols are compatible with all versions of MS Windows products. Windows NT4, as with MS Active Directory, extensively makes use of Windows SIDs.

[1] Samba local account mode results in both `DOMINICUS\FJones` and `FRANCISCUS\FJones` mapping to the UNIX user `FJones`.

Samba-3 domain member servers and clients must interact correctly with MS Windows SIDs. Incoming Windows SIDs must be translated to local UNIX UIDs and GIDs. Outgoing information from the Samba server must provide to MS Windows clients and servers appropriate SIDs.

A Samba member of a Windows networking domain (NT4-style or ADS) can be configured to handle identity mapping in a variety of ways. The mechanism it uses depends on whether or not the **winbindd** daemon is used and how the winbind functionality is configured. The configuration options are briefly described here:

Winbind is not used; users and groups are local: Where **winbindd** is not used Samba (**smbd**) uses the underlying UNIX/Linux mechanisms to resolve the identity of incoming network traffic. This is done using the LoginID (account name) in the session setup request and passing it to the getpwnam() system function call. This call is implemented using the name service switch (NSS) mechanism on modern UNIX/Linux systems. By saying "users and groups are local," we are implying that they are stored only on the local system, in the `/etc/passwd` and `/etc/group` respectively.

For example, when the user `BERYLIUM\WambatW` tries to open a connection to a Samba server the incoming SessionSetupAndX request will make a system call to look up the user `WambatW` in the `/etc/passwd` file.

This configuration may be used with standalone Samba servers, domain member servers (NT4 or ADS), and for a PDC that uses either an smbpasswd or a tdbsam-based Samba passdb backend.

Winbind is not used; users and groups resolved via NSS: In this situation user and group accounts are treated as if they are local accounts. The only way in which this differs from having local accounts is that the accounts are stored in a repository that can be shared. In practice this means that they will reside in either an NIS-type database or else in LDAP.

This configuration may be used with standalone Samba servers, domain member servers (NT4 or ADS), and for a PDC that uses either an smbpasswd or a tdbsam-based Samba passdb backend.

Winbind/NSS with the default local IDMAP table: There are many sites that require only a simple Samba server or a single Samba server that is a member of a Windows NT4 domain or an ADS domain. A typical example is an appliance like file server on which no local accounts are configured and winbind is used to obtain account credentials from the domain controllers for the domain. The domain control can be provided by Samba-3, MS Windows NT4, or MS Windows Active Directory.

Winbind is a great convenience in this situation. All that is needed is a range of UID numbers and GID numbers that can be defined in the `smb.conf` file. The `/etc/nsswitch.conf` file is configured to use **winbind**, which does all the difficult work of mapping incoming SIDs to appropriate UIDs and GIDs. The SIDs are allocated a UID/GID in the order in which winbind receives them.

This configuration is not convenient or practical in sites that have more than one Samba server and that require the same UID or GID for the same user or group across all servers. One of the hazards of this method is that in the event that the winbind IDMAP file becomes corrupted or lost, the repaired or rebuilt IDMAP file may allocate UIDs and GIDs to different users and groups from what was there previously with the result that MS Windows files that are stored on the Samba server may now not belong to the rightful owners.

Winbind/NSS uses RID based IDMAP: The IDMAP_RID facility is new to Samba version 3.0.8. It was added to make life easier for a number of sites that are committed to use of MS ADS, that do not apply an ADS schema extension, and that do not have an installed an LDAP directory server just for the purpose of maintaining an IDMAP table. If you have a single ADS domain (not a forest of domains, and not multiple domain trees) and you want a simple cookie-cutter solution to the IDMAP table problem, then IDMAP_RID is an obvious choice.

This facility requires the allocation of the *idmap uid* and the *idmap gid* ranges, and within the *idmap uid* it is possible to allocate a subset of this range for automatic mapping of the relative identifier (RID) portion of the SID directly to the base of the UID plus the RID value. For example, if the *idmap uid* range is `1000-100000000` and the *idmap backend = idmap_rid:DOMAIN_NAME=1000-50000000*, and

a SID is encountered that has the value `S-1-5-21-34567898-12529001-32973135-1234`, the resulting UID will be `1000 + 1234 = 2234`.

Winbind with an NSS/LDAP backend-based IDMAP facility: In this configuration **winbind** resolved SIDs to UIDs and GIDs from the *`idmap uid`* and *`idmap gid`* ranges specified in the `smb.conf` file, but instead of using a local winbind IDMAP table, it is stored in an LDAP directory so that all domain member machines (clients and servers) can share a common IDMAP table.

It is important that all LDAP IDMAP clients use only the master LDAP server because the *`idmap backend`* facility in the `smb.conf` file does not correctly handle LDAP redirects.

Winbind with NSS to resolve UNIX/Linux user and group IDs: The use of LDAP as the passdb backend is a smart solution for PDC, BDC, and domain member servers. It is a neat method for assuring that UIDs, GIDs, and the matching SIDs are consistent across all servers.

The use of the LDAP-based passdb backend requires use of the PADL nss_ldap utility or an equivalent. In this situation winbind is used to handle foreign SIDs, that is, SIDs from standalone Windows clients (i.e., not a member of our domain) as well as SIDs from another domain. The foreign UID/GID is mapped from allocated ranges (idmap uid and idmap gid) in precisely the same manner as when using winbind with a local IDMAP table.

The nss_ldap tool set can be used to access UIDs and GIDs via LDAP as well as via Active Directory. In order to use Active Directory, it is necessary to modify the ADS schema by installing either the AD4UNIX schema extension or using the Microsoft Services for UNIX version 3.5 or later to extend the ADS schema so it maintains UNIX account credentials. Where the ADS schema is extended, a Microsoft Management Console (MMC) snap-in is also installed to permit the UNIX credentials to be set and managed from the ADS User and Computer Management tool. Each account must be separately UNIX-enabled before the UID and GID data can be used by Samba.

14.1.3 Primary Domain Controller

Microsoft Windows domain security systems generate the user and group SID as part of the process of creation of an account. Windows does not have a concept of the UNIX UID or a GID; rather, it has its own type of security descriptor. When Samba is used as a domain controller, it provides a method of producing a unique SID for each user and group. Samba generates a machine and a domain SID to which it adds an RID that is calculated algorithmically from a base value that can be specified in the `smb.conf` file, plus twice (2x) the UID or GID. This method is called "algorithmic mapping".

For example, if a user has a UID of 4321, and the algorithmic RID base has a value of 1000, the RID will be `1000 + (2 x 4321) = 9642`. Thus, if the domain SID is `S-1-5-21-89238497-92787123-12341112`, the resulting SID is `S-1-5-21-89238497-92787123-12341112-9642`.

The foregoing type of SID is produced by Samba as an automatic function and is either produced on the fly (as is the case when using a *`passdb backend = [tdbsam | smbpasswd]`*), or may be stored as a permanent part of an account in an LDAP-based ldapsam.

ADS uses a directory schema that can be extended to accommodate additional account attributes such as UIDs and GIDs. The installation of Microsoft Service for UNIX 3.5 will expand the normal ADS schema to include UNIX account attributes. These must of course be managed separately through a snap-in module to the normal ADS account management MMC interface.

Security identifiers used within a domain must be managed to avoid conflict and to preserve itegrity. In an NT4 domain context, the PDC manages the distribution of all security credentials to the backup domain controllers (BDCs). At this time the only passdb backend for a Samba domain controller that is suitable for such information is an LDAP backend.

14.1.4 Backup Domain Controller

BDCs have read-only access to security credentials that are stored in LDAP. Changes in user or group account information are passed by the BDC to the PDC. Only the PDC can write changes to the directory.

IDMAP information can be written directly to the LDAP server so long as all domain controllers have access to the master (writable) LDAP server. Samba-3 at this time does not handle LDAP redirects in the IDMAP backend. This means that it is is unsafe to use a slave (replicate) LDAP server with the IDMAP facility.

14.2 Examples of IDMAP Backend Usage

Anyone who wishes to use **winbind** will find the following example configurations helpful. Remember that in the majority of cases **winbind** is of primary interest for use with domain member servers (DMSs) and domain member clients (DMCs).

14.2.1 Default Winbind TDB

Two common configurations are used:

- Networks that have an NT4 PDC (with or without BDCs) or a Samba PDC (with or without BDCs).
- Networks that use MS Windows 200x ADS.

14.2.1.1 NT4-Style Domains (Includes Samba Domains)

Example 14.2.1 is a simple example of an NT4 DMS `smb.conf` file that shows only the global section.

Example 14.2.1 NT4 Domain Member Server smb.conf

```
# Global parameters
 [global]
        workgroup = MEGANET2
        security = DOMAIN
        idmap uid = 10000-20000
        idmap gid = 10000-20000
        template primary group = "Domain Users"
        template shell = /bin/bash
```

The use of **winbind** requires configuration of NSS. Edit the `/etc/nsswitch.conf` so it includes the following entries:

```
...
passwd: files winbind
shadow: files winbind
group:  files winbind
...
hosts:  files [dns] wins
...
```

The use of DNS in the hosts entry should be made only if DNS is used on site.

The creation of the DMS requires the following steps:

1. Create or install an `smb.conf` file with the above configuration.

2. Execute:

   ```
   root#  net rpc join -UAdministrator%password
   Joined domain MEGANET2.
   ```

   The success of the join can be confirmed with the following command:

   ```
   root#  net rpc testjoin
   Join to 'MIDEARTH' is OK
   ```

   A failed join would report an error message like the following:

   ```
   root#  net rpc testjoin
   [2004/11/05 16:34:12, 0] utils/net_rpc_join.c:net_rpc_join_ok(66)
   Join to domain 'MEGANET2' is not valid
   ```

3. Start the **nmbd, winbind,** and **smbd** daemons in the order shown.

14.2.1.2 ADS Domains

The procedure for joining an ADS domain is similar to the NT4 domain join, except the `smb.conf` file will have the contents shown in Example 14.2.2

Example 14.2.2 ADS Domain Member Server smb.conf

```
# Global parameters
 [global]
        workgroup = BUTTERNET
        netbios name = GARGOYLE
        realm = BUTTERNET.BIZ
        security = ADS
        template shell = /bin/bash
        idmap uid = 500-10000000
        idmap gid = 500-10000000
        winbind use default domain = Yes
        winbind nested groups = Yes
        printer admin = "BUTTERNET\Domain Admins"
```

ADS DMS operation requires use of kerberos (KRB). For this to work, the `krb5.conf` must be configured. The exact requirements depends on which version of MIT or Heimdal Kerberos is being used. It is sound advice to use only the latest version, which at this time are MIT Kerberos version 1.3.5 and Heimdal 0.61.

The creation of the DMS requires the following steps:

1. Create or install an `smb.conf` file with the above configuration.
2. Edit the `/etc/nsswitch.conf` file as shown above.
3. Execute:

   ```
   root#  net ads join -UAdministrator%password
   Joined domain BUTTERNET.
   ```

   The success or failure of the join can be confirmed with the following command:

   ```
   root#  net ads testjoin
   ```

```
Using short domain name -- BUTTERNET
Joined 'GARGOYLE' to realm 'BUTTERNET.BIZ'
```

An invalid or failed join can be detected by executing:

```
root#  net ads testjoin
GARGOYLE$@'s password:
[2004/11/05 16:53:03, 0] utils/net_ads.c:ads_startup(186)
  ads_connect: No results returned
Join to domain is not valid
```

The specific error message may differ from the above because it depends on the type of failure that may have occurred. Increase the *log level* to 10, repeat the test, and then examine the log files produced to identify the nature of the failure.

4. Start the **nmbd**, **winbind**, and **smbd** daemons in the order shown.

14.2.2 IDMAP_RID with Winbind

The **idmap_rid** facility is a new tool that, unlike native winbind, creates a predictable mapping of MS Windows SIDs to UNIX UIDs and GIDs. The key benefit of this method of implementing the Samba IDMAP facility is that it eliminates the need to store the IDMAP data in a central place. The downside is that it can be used only within a single ADS domain and is not compatible with trusted domain implementations.

This alternate method of SID to UID/GID mapping can be achieved using the idmap_rid plug-in. This plug-in uses the RID of the user SID to derive the UID and GID by adding the RID to a base value specified. This utility requires that the parameter "allow trusted domains = No" be specified, as it is not compatible with multiple domain environments. The *idmap uid* and *idmap gid* ranges must be specified.

The idmap_rid facility can be used both for NT4/Samba-style domains and Active Directory. To use this with an NT4 domain, do not include the *realm* parameter; additionally, the method used to join the domain uses the `net rpc join` process.

An example `smb.conf` file for and ADS domain environment is shown in Example 14.2.3.

Example 14.2.3 ADS Domain Member smb.conf using idmap_rid

```
# Global parameters
 [global]
        workgroup = KPAK
        netbios name = BIGJOE
        realm = CORP.KPAK.COM
        server string = Office Server
        security = ADS
        allow trusted domains = No
        idmap backend = idmap_rid:KPAK=500-100000000
        idmap uid = 500-100000000
        idmap gid = 500-100000000
        template shell = /bin/bash
        winbind use default domain = Yes
        winbind enum users = No
        winbind enum groups = No
        winbind nested groups = Yes
        printer admin = "Domain Admins"
```

In a large domain with many users it is imperative to disable enumeration of users and groups. For example, at a site that has 22,000 users in Active Directory the winbind-based user and group resolution is unavailable for nearly 12 minutes following first startup of **winbind**. Disabling enumeration resulted in instantaneous response. The disabling of user and group enumeration means that it will not be possible to list users or groups using the **getent passwd** and **getent group** commands. It will be possible to perform the lookup for individual users, as shown in the following procedure.

The use of this tool requires configuration of NSS as per the native use of winbind. Edit the `/etc/nsswitch.conf` so it has the following parameters:

```
...
passwd: files winbind
shadow: files winbind
group:  files winbind
...
hosts:  files wins
```

...

The following procedure can use the idmap_rid facility:

1. Create or install an `smb.conf` file with the above configuration.
2. Edit the `/etc/nsswitch.conf` file as shown above.
3. Execute:

   ```
   root#  net ads join -UAdministrator%password
   Using short domain name -- KPAK
   Joined 'BIGJOE' to realm 'CORP.KPAK.COM'
   ```

   An invalid or failed join can be detected by executing:

   ```
   root#  net ads testjoin
   BIGJOE$@'s password:
   [2004/11/05 16:53:03, 0] utils/net_ads.c:ads_startup(186)
     ads_connect: No results returned
   Join to domain is not valid
   ```

   The specific error message may differ from the above because it depends on the type of failure that may have occurred. Increase the *`log level`* to 10, repeat the test, and then examine the log files produced to identify the nature of the failure.
4. Start the **nmbd**, **winbind**, and **smbd** daemons in the order shown.
5. Validate the operation of this configuration by executing:

   ```
   root#  getent passwd administrator
   administrator:x:1000:1013:Administrator:/home/BE/administrator:/bin/bash
   ```

14.2.3 IDMAP Storage in LDAP Using Winbind

The storage of IDMAP information in LDAP can be used with both NT4/Samba-3-style domains and ADS domains. OpenLDAP is a commonly used LDAP

server for this purpose, although any standards-complying LDAP server can be used. It is therefore possible to deploy this IDMAP configuration using the Sun iPlanet LDAP server, Novell eDirectory, Microsoft ADS plus ADAM, and so on.

An example is for an ADS domain is shown in Example 14.2.4.

Example 14.2.4 ADS Domain Member Server using LDAP

```
# Global parameters
 [global]
        workgroup = SNOWSHOW
        netbios name = GOODELF
        realm = SNOWSHOW.COM
        server string = Samba Server
        security = ADS
        log level = 1 ads:10 auth:10 sam:10 rpc:10
        ldap admin dn = cn=Manager,dc=SNOWSHOW,dc=COM
        ldap idmap suffix = ou=Idmap
        ldap suffix = dc=SNOWSHOW,dc=COM
        idmap backend = ldap:ldap://ldap.snowshow.com
        idmap uid = 150000-550000
        idmap gid = 150000-550000
        template shell = /bin/bash
        winbind use default domain = Yes
```

In the case of an NT4 or Samba-3-style domain the *realm* is not used, and the command used to join the domain is **net rpc join**. The above example also demonstrates advanced error-reporting techniques that are documented in Section 40.3.

Where MIT kerberos is installed (version 1.3.4 or later), edit the `/etc/krb5.conf` file so it has the following contents:

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 default_realm = SNOWSHOW.COM
 dns_lookup_realm = false
```

```
 dns_lookup_kdc = true

[appdefaults]
 pam = {
   debug = false
   ticket_lifetime = 36000
   renew_lifetime = 36000
   forwardable = true
   krb4_convert = false
 }
```

Where Heimdal kerberos is installed, edit the `/etc/krb5.conf` file so it is either empty (i.e., no contents) or it has the following contents:

```
[libdefaults]
        default_realm = SNOWSHOW.COM
        clockskew = 300

[realms]
        SNOWSHOW.COM = {
                kdc = ADSDC.SHOWSHOW.COM
        }

[domain_realm]
        .snowshow.com = SNOWSHOW.COM
```

NOTE

Samba cannot use the Heimdal libraries if there is no `/etc/krb5.conf` file. So long as there is an empty file, the Heimdal kerberos libraries will be usable. There is no need to specify any settings because Samba, using the Heimdal libraries, can figure this out automatically.

Edit the NSS control file `/etc/nsswitch.conf` so it has the following entries:

```
...
passwd: files ldap
shadow: files ldap
group:  files ldap
...
hosts:  files wins
...
```

You will need the PADL[2] **nss_ldap** tool set for this solution. Configure the `/etc/ldap.conf` file so it has the information needed. The following is an example of a working file:

```
host    192.168.2.1
base    dc=snowshow,dc=com
binddn  cn=Manager,dc=snowshow,dc=com
bindpw  not24get

pam_password exop

nss_base_passwd ou=People,dc=snowshow,dc=com?one
nss_base_shadow ou=People,dc=snowshow,dc=com?one
nss_base_group  ou=Groups,dc=snowshow,dc=com?one
ssl     no
```

The following procedure may be followed to effect a working configuration:

1. Configure the `smb.conf` file as shown above.
2. Create the `/etc/krb5.conf` file as shown above.
3. Configure the `/etc/nsswitch.conf` file as shown above.
4. Download, build, and install the PADL nss_ldap tool set. Configure the `/etc/ldap.conf` file as shown above.
5. Configure an LDAP server and initialize the directory with the top-level entries needed by IDMAP, shown in the following LDIF file:

[2] <http://www.padl.com>

```
dn: dc=snowshow,dc=com
objectClass: dcObject
objectClass: organization
dc: snowshow
o: The Greatest Snow Show in Singapore.
description: Posix and Samba LDAP Identity Database

dn: cn=Manager,dc=snowshow,dc=com
objectClass: organizationalRole
cn: Manager
description: Directory Manager

dn: ou=Idmap,dc=snowshow,dc=com
objectClass: organizationalUnit
ou: idmap
```

6. Execute the command to join the Samba DMS to the ADS domain as shown here:

```
root#  net ads testjoin
Using short domain name -- SNOWSHOW
Joined 'GOODELF' to realm 'SNOWSHOW.COM'
```

7. Store the LDAP server access password in the Samba `secrets.tdb` file as follows:

```
root#  smbpasswd -w not24get
```

8. Start the **nmbd**, **winbind**, and **smbd** daemons in the order shown.

Follow the diagnositic procedures shown earlier in this chapter to identify success or failure of the join. In many cases a failure is indicated by a silent return to the command prompt with no indication of the reason for failure.

14.2.4 IDMAP and NSS Using LDAP from ADS with RFC2307bis Schema Extension

The use of this method is messy. The information provided in the following is for guidance only and is very definitely not complete. This method does work; it is used in a number of large sites and has an acceptable level of performance.

An example `smb.conf` file is shown in Example 14.2.5.

Example 14.2.5 ADS Domain Member Server using RFC2307bis Schema Extension Date via NSS

```
# Global parameters
 [global]
        workgroup = BOBBY
        realm = BOBBY.COM
        security = ADS
        idmap uid = 150000-550000
        idmap gid = 150000-550000
        template shell = /bin/bash
        winbind cache time = 5
        winbind use default domain = Yes
        winbind trusted domains only = Yes
        winbind nested groups = Yes
```

The DMS must be joined to the domain using the usual procedure. Additionally, it is necessary to build and install the PADL nss_ldap tool set. Be sure to build this tool set with the following:

```
./configure --enable-rfc2307bis --enable-schema-mapping
make install
```

The following `/etc/nsswitch.conf` file contents are required:

```
...
passwd: files ldap
shadow: files ldap
group:  files ldap
...
```

```
hosts:  files wins
...
```

The /etc/ldap.conf file must be configured also. Refer to the PADL documentation and source code for nss_ldap to specific instructions.

The next step involves preparation of the ADS schema. This is briefly discussed in the remaining part of this chapter.

14.2.4.1 IDMAP, Active Directory, and MS Services for UNIX 3.5

The Microsoft Windows Service for UNIX (SFU) version 3.5 is available for free download[3] from the Microsoft Web site. You will need to download this tool and install it following Microsoft instructions.

14.2.4.2 IDMAP, Active Directory and AD4UNIX

Instructions for obtaining and installing the AD4UNIX tool set can be found from the Geekcomix[4] Web site.

[3] <http://www.microsoft.com/windows/sfu/>

[4] <http://www.geekcomix.com/cgi-bin/classnotes/wiki.pl?LDAP01/An_Alternative_Approach>

Chapter 15

USER RIGHTS AND PRIVILEGES

The administration of Windows user, group, and machine accounts in the Samba domain-controlled network necessitates interfacing between the MS Windows networking environment and the UNIX operating system environment. The right (permission) to add machines to the Windows security domain can be assigned (set) to non-administrative users both in Windows NT4 domains and Active Directory domains.

The addition of Windows NT4/2kX/XPPro machines to the domain necessitates the creation of a machine account for each machine added. The machine account is a necessity that is used to validate that the machine can be trusted to permit user logons.

Machine accounts are analogous to user accounts, and thus in implementing them on a UNIX machine that is hosting Samba (i.e., on which Samba is running), it is necessary to create a special type of user account. Machine accounts differ from normal user accounts in that the account name (login ID) is terminated with a `$` sign. An additional difference is that this type of account should not ever be able to log into the UNIX environment as a system user and therefore is set to have a shell of **/bin/false** and a home directory of **/dev/null.** The machine account is used only to authenticate domain member machines during start-up. This security measure is designed to block man-in-the-middle attempts to violate network integrity.

NOTE

Machine (computer) accounts are used in the Windows NT OS family to store security credentials for domain member servers and workstations. When the domain member starts up, it goes through a validation process that includes an exchange of credentials with a domain controller. If the domain member fails to authenticate using the credentials known for it by domain controllers, the machine will be refused all access by domain users. The computer account is essential to the way that MS Windows secures authentication.

The creation of UNIX system accounts has traditionally been the sole right of the system administrator, better known as the `root` account. It is possible in the UNIX environment to create multiple users who have the same UID. Any UNIX user who has a UID=0 is inherently the same as the `root` account user.

All versions of Samba call system interface scripts that permit CIFS function calls that are used to manage users, groups, and machine accounts in the UNIX environment. All versions of Samba up to and including version 3.0.10 required the use of a Windows administrator account that unambiguously maps to the UNIX `root` account to permit the execution of these interface scripts. The requirement to do this has understandably met with some disdain and consternation among Samba administrators, particularly where it became necessary to permit people who should not possess `root`-level access to the UNIX host system.

15.1 Rights Management Capabilities

Samba 3.0.11 introduced support for the Windows privilege model. This model allows certain rights to be assigned to a user or group SID. In order to enable this feature, *enable privileges* = yes must be defined in the *global* section of the `smb.conf` file.

Currently, the rights supported in Samba-3 are listed in Table 15.1. The remainder of this chapter explains how to manage and use these privileges

on Samba servers.

Table 15.1 Current Privilege Capabilities

Privilege	Description
SeMachineAccountPrivilege	Add machines to domain
SePrintOperatorPrivilege	Manage printers
SeAddUsersPrivilege	Add users and groups to the domain
SeRemoteShutdownPrivilege	Force shutdown from a remote system
SeDiskOperatorPrivilege	Manage disk share
SeTakeOwnershipPrivilege	Take ownership of files or other objects

15.1.1 Using the "net rpc rights" Utility

There are two primary means of managing the rights assigned to users and groups on a Samba server. The **NT4 User Manager for Domains** may be used from any Windows NT4, 2000, or XP Professional domain member client to connect to a Samba domain controller and view/modify the rights assignments. This application, however, appears to have bugs when run on a client running Windows 2000 or later; therefore, Samba provides a command-line utility for performing the necessary administrative actions.

The **net rpc rights** utility in Samba 3.0.11 has three new subcommands:

list [name|accounts] When called with no arguments, **net rpc list** simply lists the available rights on the server. When passed a specific user or group name, the tool lists the privileges currently assigned to the specified account. When invoked using the special string `accounts`, **net rpc rights list** returns a list of all privileged accounts on the server and the assigned rights.

grant <user> <right [right ...]> When called with no arguments, this function is used to assign a list of rights to a specified user or group. For example, to grant the members of the Domain Admins group on a Samba domain controller, the capability to add client machines to the domain, one would run:

```
root#  net -S server -U domadmin rpc rights grant \
```

```
'DOMAIN\Domain Admins' SeMachineAccountPrivilege
```

The following syntax has the same result:

```
root#  net rpc rights grant 'DOMAIN\Domain Admins' \
     SeMachineAccountPrivilege -S server -U domadmin
```

More than one privilege can be assigned by specifying a list of rights separated by spaces. The parameter 'Domain\Domain Admins' must be quoted with single ticks or using double-quotes to prevent the backslash and the space from being interpreted by the system shell.

revoke <user> <right [right ...]> This command is similar in format to **net rpc rights grant**. Its effect is to remove an assigned right (or list of rights) from a user or group.

NOTE

You must be connected as a member of the Domain Admins group to be able to grant or revoke privileges assigned to an account. This capability is inherent to the Domain Admins group and is not configurable. There are no default rights and privileges, except the ability for a member of the Domain Admins group to assign them. This means that all administrative rights and privileges (other than the ability to assign them) must be explicitly assigned, even for the Domain Admins group.

By default, no privileges are initially assigned to any account because certain actions will be performed as root once smbd determines that a user has the necessary rights. For example, when joining a client to a Windows domain, *add machine script* must be executed with superuser rights in most cases. For this reason, you should be very careful about handing out privileges to accounts.

Access as the root user (UID=0) bypasses all privilege checks.

15.1.2 Description of Privileges

The privileges that have been implemented in Samba-3.0.11 are shown below. It is possible, and likely, that additional privileges may be implemented in later releases of Samba. It is also likely that any privileges currently implemented but not used may be removed from future releases as a housekeeping matter, so it is important that the successful as well as unsuccessful use of these facilities should be reported on the Samba mailing lists.

SeAddUsersPrivilege This right determines whether or not smbd will allow the user to create new user or group accounts via such tools as **net rpc user add** or **NT4 User Manager for Domains.**

SeDiskOperatorPrivilege Accounts that possess this right will be able to execute scripts defined by the **add/delete/change** share command in `smb.conf` file as root. Such users will also be able to modify the ACL associated with file shares on the Samba server.

SeMachineAccountPrivilege This right controls whether or not the user can join client machines to a Samba-controlled domain.

SePrintOperatorPrivilege This privilege operates identically to the *printer admin* option in the `smb.conf` file (see section 5 man page for `smb.conf`) except that it is a global right (not on a per-printer basis). Eventually the smb.conf option will be deprecated and administrative rights to printers will be controlled exclusively by this right and the security descriptor associated with the printer object in the `ntprinters.tdb` file.

SeRemoteShutdownPrivilege Samba provides two hooks for shutting down or rebooting the server and for aborting a previously issued shutdown command. Since this is an operation normally limited by the operating system to the root user, an account must possess this right to be able to execute either of these hooks.

SeTakeOwnershipPrivilege This right permits users to take ownership of files and directories.

15.1.3 Privileges Suppored by Windows 2000 Domain Controllers

For reference purposes, a Windows NT4 Primary Domain Controller reports support for the following privileges:

```
         SeCreateTokenPrivilege  Create a token object
  SeAssignPrimaryTokenPrivilege  Replace a process level token
          SeLockMemoryPrivilege  Lock pages in memory
       SeIncreaseQuotaPrivilege  Increase quotas
      SeMachineAccountPrivilege  Add workstations to domain
                 SeTcbPrivilege  Act as part of the operating system
            SeSecurityPrivilege  Manage auditing and security log
       SeTakeOwnershipPrivilege  Take ownership of files or other objects
          SeLoadDriverPrivilege  Load and unload device drivers
       SeSystemProfilePrivilege  Profile system performance
          SeSystemtimePrivilege  Change the system time
SeProfileSingleProcessPrivilege  Profile single process
SeIncreaseBasePriorityPrivilege  Increase scheduling priority
      SeCreatePagefilePrivilege  Create a pagefile
     SeCreatePermanentPrivilege  Create permanent shared objects
              SeBackupPrivilege  Back up files and directories
             SeRestorePrivilege  Restore files and directories
            SeShutdownPrivilege  Shut down the system
               SeDebugPrivilege  Debug programs
               SeAuditPrivilege  Generate security audits
   SeSystemEnvironmentPrivilege  Modify firmware environment values
        SeChangeNotifyPrivilege  Bypass traverse checking
      SeRemoteShutdownPrivilege  Force shutdown from a remote system
```

And Windows 200x/XP Domain Controllers and workstations reports to support the following privileges:

```
         SeCreateTokenPrivilege  Create a token object
  SeAssignPrimaryTokenPrivilege  Replace a process level token
          SeLockMemoryPrivilege  Lock pages in memory
```

```
       SeIncreaseQuotaPrivilege  Increase quotas
      SeMachineAccountPrivilege  Add workstations to domain
                 SeTcbPrivilege  Act as part of the operating system
            SeSecurityPrivilege  Manage auditing and security log
       SeTakeOwnershipPrivilege  Take ownership of files or other objects
          SeLoadDriverPrivilege  Load and unload device drivers
       SeSystemProfilePrivilege  Profile system performance
          SeSystemtimePrivilege  Change the system time
SeProfileSingleProcessPrivilege  Profile single process
SeIncreaseBasePriorityPrivilege  Increase scheduling priority
      SeCreatePagefilePrivilege  Create a pagefile
     SeCreatePermanentPrivilege  Create permanent shared objects
             SeBackupPrivilege  Back up files and directories
            SeRestorePrivilege  Restore files and directories
           SeShutdownPrivilege  Shut down the system
              SeDebugPrivilege  Debug programs
              SeAuditPrivilege  Generate security audits
   SeSystemEnvironmentPrivilege  Modify firmware environment values
        SeChangeNotifyPrivilege  Bypass traverse checking
      SeRemoteShutdownPrivilege  Force shutdown from a remote system
             SeUndockPrivilege  Remove computer from docking station
          SeSyncAgentPrivilege  Synchronize directory service data
    SeEnableDelegationPrivilege  Enable computer and user accounts to
                                 be trusted for delegation
        SeManageVolumePrivilege  Perform volume maintenance tasks
         SeImpersonatePrivilege  Impersonate a client after authentication
        SeCreateGlobalPrivilege  Create global objects
```

The Samba Team is implementing only those privileges that are logical and useful in the UNIX/Linux environment. Many of the Windows 200X/XP privileges have no direct equivalence in UNIX.

15.2 The Administrator Domain SID

Please note that every Windows NT4 and later server requires a domain Administrator account. Samba versions commencing with 3.0.11 permit Administrative duties to be performed via assigned rights and privileges (see Chapter 15, "User Rights and Privileges"). An account in the server's passdb

backend can be set to the well-known RID of the default administrator account. To obtain the domain SID on a Samba domain controller, run the following command:

```
root#  net getlocalsid
SID for domain FOO is: S-1-5-21-4294955119-3368514841-2087710299
```

You may assign the domain administrator RID to an account using the **pdbedit** command as shown here:

```
root#  pdbedit -U S-1-5-21-4294955119-3368514841-2087710299-500 -u root -r
```

NOTE

The RID 500 is the well known standard value of the default Administrator account. It is the RID that confers the rights and privileges that the Administrator account has on a Windows machine or domain. Under UNIX/Linux the equivalent is UID=0 (the root account).

Releases of Samba version 3.0.11 and later make it possible to operate without an Administrator account provided equivalent rights and privileges have been established for a Windows user or a Windows group account.

15.3 Common Errors

15.3.1 What Rights and Privileges Will Permit Windows Client Administration?

When a Windows NT4 (or later) client joins a domain, the domain global `Domain Admins` group is added to the membership of the local `Administra-`

`tors` group on the client. Any user who is a member of the domain global `Domain Admins` group will have administrative rights on the Windows client.

This is often not the most desirable solution because it means that the user will have administrative rights and privileges on domain servers also. The `Power Users` group on Windows client workstations permits local administration of the workstation alone. Any domain global user or domain global group can be added to the membership of the local workstation group `Power Users`.

See Section 13.3.3 for an example of how to add domain users and groups to a local group that is on a Windows workstation. The use of the **net** command permits this to be done from the Samba server.

Another way this can be done is to log onto the Windows workstation as the user `Administrator`, then open a **cmd** shell, then execute:

```
C:\>  net localgroup administrators /add domain_name\entity
```

where `entity` is either a domain user or a domain group account name.

Chapter 16

FILE, DIRECTORY, AND SHARE ACCESS CONTROLS

Advanced MS Windows users are frequently perplexed when file, directory, and share manipulation of resources shared via Samba do not behave in the manner they might expect. MS Windows network administrators are often confused regarding network access controls and how to provide users with the access they need while protecting resources from unauthorized access.

Many UNIX administrators are unfamiliar with the MS Windows environment and in particular have difficulty in visualizing what the MS Windows user wishes to achieve in attempts to set file and directory access permissions.

The problem lies in the differences in how file and directory permissions and controls work between the two environments. This difference is one that Samba cannot completely hide, even though it does try to bridge the chasm to a degree.

POSIX Access Control List technology has been available (along with extended attributes) for UNIX for many years, yet there is little evidence today of any significant use. This explains to some extent the slow adoption of ACLs into commercial Linux products. MS Windows administrators are astounded at this, given that ACLs were a foundational capability of the now decade-old MS Windows NT operating system.

The purpose of this chapter is to present each of the points of control that are possible with Samba-3 in the hope that this will help the network administrator to find the optimum method for delivering the best environment for MS Windows desktop users.

This is an opportune point to mention that Samba was created to provide a means of interoperability and interchange of data between differing operating environments. Samba has no intent to change UNIX/Linux into a platform like MS Windows. Instead the purpose was and is to provide a sufficient level of exchange of data between the two environments. What is available today extends well beyond early plans and expectations, yet the gap continues to shrink.

16.1 Features and Benefits

Samba offers much flexibility in file system access management. These are the key access control facilities present in Samba today:

Samba Access Control Facilities

- *UNIX File and Directory Permissions*

  Samba honors and implements UNIX file system access controls. Users who access a Samba server will do so as a particular MS Windows user. This information is passed to the Samba server as part of the logon or connection setup process. Samba uses this user identity to validate whether or not the user should be given access to file system resources (files and directories). This chapter provides an overview for those to whom the UNIX permissions and controls are a little strange or unknown.

- *Samba Share Definitions*

  In configuring share settings and controls in the `smb.conf` file, the network administrator can exercise overrides to native file system permissions and behaviors. This can be handy and convenient to effect behavior that is more like what MS Windows NT users expect, but it is seldom the *best* way to achieve this. The basic options and techniques are described herein.

- *Samba Share ACLs*

  Just as it is possible in MS Windows NT to set ACLs on shares themselves, so it is possible to do in Samba. Few people make use of this facility, yet it remains one of the easiest ways to affect access controls (restrictions) and can often do so with minimum invasiveness compared with other methods.

- *MS Windows ACLs through UNIX POSIX ACLs*

  The use of POSIX ACLs on UNIX/Linux is possible only if the underlying operating system supports them. If not, then this option will not be available to you. Current UNIX technology platforms have native support for POSIX ACLs. There are patches for the Linux kernel that also provide this support. Sadly, few Linux platforms ship today with native ACLs and extended attributes enabled. This chapter has pertinent information for users of platforms that support them.

16.2 File System Access Controls

Perhaps the most important recognition to be made is the simple fact that MS Windows NT4/200x/XP implement a totally divergent file system technology from what is provided in the UNIX operating system environment. First we consider what the most significant differences are, then we look at how Samba helps to bridge the differences.

16.2.1 MS Windows NTFS Comparison with UNIX File Systems

Samba operates on top of the UNIX file system. This means it is subject to UNIX file system conventions and permissions. It also means that if the MS Windows networking environment requires file system behavior, that differs from UNIX file system behavior then somehow Samba is responsible for emulating that in a transparent and consistent manner.

It is good news that Samba does this to a large extent, and on top of that, provides a high degree of optional configuration to override the default behavior. We look at some of these overrides, but for the greater part we stay within the bounds of default behavior. Those wishing to explore the depths of control ability should review the `smb.conf` man page.

The following compares file system features for UNIX with those of MS Windows NT/200x:

Name Space MS Windows NT4/200x/XP file names may be up to 254 characters long, and UNIX file names may be 1023 characters long. In MS Windows, file extensions indicate particular file types; in UNIX

this is not so rigorously observed because all names are considered arbitrary.

What MS Windows calls a folder, UNIX calls a directory.

Case Sensitivity MS Windows file names are generally uppercase if made up of 8.3 (8-character file name and 3 character extension. File names that are longer than 8.3 are case preserving and case insensitive.

UNIX file and directory names are case sensitive and case preserving. Samba implements the MS Windows file name behavior, but it does so as a user application. The UNIX file system provides no mechanism to perform case-insensitive file name lookups. MS Windows does this by default. This means that Samba has to carry the processing overhead to provide features that are not native to the UNIX operating system environment.

Consider the following. All are unique UNIX names but one single MS Windows file name:

```
MYFILE.TXT
MyFile.txt
myfile.txt
```

So clearly, in an MS Windows file namespace these three files cannot co-exist, but in UNIX they can.

So what should Samba do if all three are present? That which is lexically first will be accessible to MS Windows users; the others are invisible and unaccessible — any other solution would be suicidal. The Windows client will ask for a case-insensitive file lookup, and that is the reason for which Samba must offer a consistent selection in the event that the UNIX directory contains multiple files that would match a case insensitive file listing.

Directory Separators MS Windows and DOS use the backslash \ as a directory delimiter, and UNIX uses the forward-slash / as its directory delimiter. This is handled transparently by Samba.

Drive Identification MS Windows products support a notion of drive letters, like **C:**, to represent disk partitions. UNIX has no concept of separate identifiers for file partitions; each such file system is mounted to become part of the overall directory tree. The UNIX directory tree begins at `/` just as the root of a DOS drive is specified as `C:\`.

File Naming Conventions MS Windows generally never experiences file names that begin with a dot (`.`), while in UNIX these are commonly found in a user's home directory. Files that begin with a dot (`.`) are typically startup files for various UNIX applications, or they may be files that contain startup configuration data.

Links and Short-Cuts MS Windows make use of *links and shortcuts* that are actually special types of files that will redirect an attempt to execute the file to the real location of the file. UNIX knows of file and directory links, but they are entirely different from what MS Windows users are used to.

Symbolic links are files in UNIX that contain the actual location of the data (file or directory). An operation (like read or write) will operate directly on the file referenced. Symbolic links are also referred to as "soft links." A hard link is something that MS Windows is not familiar with. It allows one physical file to be known simultaneously by more than one file name.

There are many other subtle differences that may cause the MS Windows administrator some temporary discomfort in the process of becoming familiar with UNIX/Linux. These are best left for a text that is dedicated to the purpose of UNIX/Linux training and education.

16.2.2 Managing Directories

There are three basic operations for managing directories: **create**, **delete**, **rename**. Table 16.1 compares the commands in Windows and UNIX that implement these operations.

Table 16.1 Managing Directories with UNIX and Windows

Action	MS Windows Command	UNIX Command
create	md folder	mkdir folder
delete	rd folder	rmdir folder
rename	rename oldname newname	mv oldname newname

16.2.3 File and Directory Access Control

The network administrator is strongly advised to read basic UNIX training manuals and reference materials regarding file and directory permissions maintenance. Much can be achieved with the basic UNIX permissions without having to resort to more complex facilities like POSIX ACLs or extended attributes (EAs).

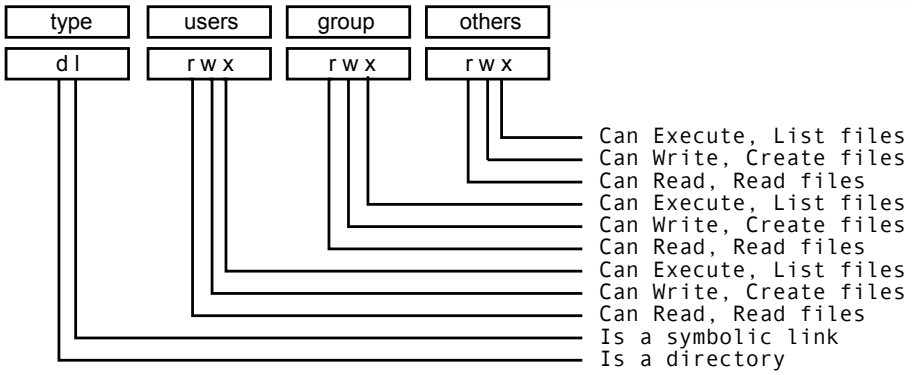
UNIX/Linux file and directory access permissions involves setting three primary sets of data and one control set. A UNIX file listing looks as follows:

```
$ ls -la
total 632
drwxr-xr-x   13 maryo   gnomes      816 2003-05-12 22:56 .
drwxrwxr-x   37 maryo   gnomes     3800 2003-05-12 22:29 ..
dr-xr-xr-x    2 maryo   gnomes       48 2003-05-12 22:29 muchado02
drwxrwxrwx    2 maryo   gnomes       48 2003-05-12 22:29 muchado03
drw-rw-rw-    2 maryo   gnomes       48 2003-05-12 22:29 muchado04
d-w--w--w-    2 maryo   gnomes       48 2003-05-12 22:29 muchado05
dr--r--r--    2 maryo   gnomes       48 2003-05-12 22:29 muchado06
drwsrwsrwx    2 maryo   gnomes       48 2003-05-12 22:29 muchado08
----------    1 maryo   gnomes     1242 2003-05-12 22:31 mydata00.lst
--w--w--w-    1 maryo   gnomes     7754 2003-05-12 22:33 mydata02.lst
-r--r--r--    1 maryo   gnomes    21017 2003-05-12 22:32 mydata04.lst
-rw-rw-rw-    1 maryo   gnomes    41105 2003-05-12 22:32 mydata06.lst
$
```

The columns represent (from left to right) permissions, number of hard links to file, owner, group, size (bytes), access date, time of last modification, and file name.

An overview of the permissions field is shown in Figure 16.1.

Figure 16.1 Overview of UNIX permissions field.



Any bit flag may be unset. An unset bit flag is the equivalent of "cannot" and is represented as a "-" character (see Example 16.2.1)

Example 16.2.1 Example File

```
-rwxr-x---   Means:
 ^^^                 The owner (user) can read, write, execute
    ^^^              the group can read and execute
       ^^^           everyone else cannot do anything with it.
```

Additional possibilities in the [type] field are c = character device, b = block device, p = pipe device, s = UNIX Domain Socket.

The letters `rwxXst` set permissions for the user, group, and others as read (r), write (w), execute (or access for directories) (x), execute only if the file is a directory or already has execute permission for some user (X), set user (SUID) or group ID (SGID) on execution (s), sticky (t).

When the sticky bit is set on a directory, files in that directory may be unlinked (deleted) or renamed only by root or their owner. Without the sticky bit, anyone able to write to the directory can delete or rename files. The sticky bit is commonly found on directories, such as `/tmp`, that are world-writable.

When the set user or group ID bit (s) is set on a directory, then all files created within it will be owned by the user and/or group whose 'set user or group' bit is set. This can be helpful in setting up directories for which it is desired that all users who are in a group should be able to write to and read

from a file, particularly when it is undesirable for that file to be exclusively owned by a user whose primary group is not the group that all such users belong to.

When a directory is set `d-wx--x---`, the owner can read and create (write) files in it, but because the (r) read flags are not set, files cannot be listed (seen) in the directory by anyone. The group can read files in the directory but cannot create new files. If files in the directory are set to be readable and writable for the group, then group members will be able to write to (or delete) them.

16.2.3.1 Protecting Directories and Files from Deletion

People have asked on the Samba mailing list how is it possible to protect files or directories from deletion by users. For example, Windows NT/2K/XP provides the capacity to set access controls on a directory into which people can write files but not delete them. It is possible to set an ACL on a Windows file that permits the file to be written to but not deleted. Such concepts are foreign to the UNIX operating system file space. Within the UNIX file system anyone who has the ability to create a file can write to it. Anyone who has write permission on the directory that contains a file and has write permission for it has the capability to delete it.

For the record, in the UNIX environment the ability to delete a file is controlled by the permissions on the directory that the file is in. In other words, a user can delete a file in a directory to which that user has write access, even if that user does not own the file.

Of necessity, Samba is subject to the file system semantics of the host operating system. Samba is therefore limited in the file system capabilities that can be made available through Windows ACLs, and therefore performs a "best fit" translation to POSIX ACLs. Some UNIX file systems do, however support, a feature known as extended attributes. Only the Windows concept of *inheritance* is implemented by Samba through the appropriate extended attribute.

The specific semantics of the extended attributes are not consistent across UNIX and UNIX-like systems such as Linux. For example, it is possible on some implementations of the extended attributes to set a flag that prevents the directory or file from being deleted. The extended attribute that may achieve this is called the `immutible` bit. Unfortunately, the implementation

of the immutible flag is NOT consistent with published documentation. For example, the man page for the **chattr** on SUSE Linux 9.2 says:

```
A file with the i attribute cannot be modified: it cannot be deleted
or renamed, no link can be created to this file and no data can be
written to the file. Only the superuser or a process possessing the
CAP_LINUX_IMMUTABLE capability can set or clear this attribute.
```

A simple test can be done to check if the immutible flag is supported on files in the file system of the Samba host server. Test for File Immutibility Support

1. Create a file called `filename`.

2. Login as the `root` user, then set the immutibile flag on a test file as follows:

   ```
   root#  chattr +i 'filename'
   ```

3. Login as the user who owns the file (not root) and attempt to remove the file as follows:

   ```
   mystic:/home/hannibal > rm filename
   ```

   It will not be possible to delete the file if the immutible flag is correctly honored.

On operating systems and file system types that support the immutible bit, it is possible to create directories that cannot be deleted. Check the man page on your particular host system to determine whether or not immutable directories are writable. If they are not, then the entire directory and its contents will effectively be protected from writing (file creation also) and deletion.

16.3 Share Definition Access Controls

The following parameters in the `smb.conf` file sections define a share control or affect access controls. Before using any of the following options, please refer to the man page for `smb.conf`.

16.3.1 User- and Group-Based Controls

User- and group-based controls can prove quite useful. In some situations it is distinctly desirable to force all file system operations as if a single user were doing so. The use of the *force user* and *force group* behavior will achieve this. In other situations it may be necessary to use a paranoia level of control to ensure that only particular authorized persons will be able to access a share or its contents. Here the use of the *valid users* or the *invalid users* parameter may be useful.

As always, it is highly advisable to use the easiest to maintain and the least ambiguous method for controlling access. Remember, when you leave the scene, someone else will need to provide assistance, and if he or she finds too great a mess or does not understand what you have done, there is risk of Samba being removed and an alternative solution being adopted.

Table 16.2 enumerates these controls.

16.3.2 File and Directory Permissions-Based Controls

Directory permission-based controls, if misused, can result in considerable difficulty in diagnosing the causes of misconfiguration. Use them sparingly and carefully. By gradually introducing each, one at a time, undesirable side effects may be detected. In the event of a problem, always comment all of them out and then gradually reintroduce them in a controlled way.

Refer to Table 16.3 for information regarding the parameters that may be used to set file and directory permission-based access controls.

16.3.3 Miscellaneous Controls

The parameter documented in Table 16.4 are often used by administrators in ways that create inadvertent barriers to file access. Such are the consequences of not understanding the full implications of `smb.conf` file settings.

Table 16.2 User- and Group-Based Controls

Control Parameter	Description, Action, Notes
admin users	List of users who will be granted administrative privileges on the share. They will do all file operations as the superuser (root). Users in this list will be able to do anything they like on the share, irrespective of file permissions.
force group	Specifies a UNIX group name that will be assigned as the default primary group for all users connecting to this service.
force user	Specifies a UNIX username that will be assigned as the default user for all users connecting to this service. This is useful for sharing files. Incorrect use can cause security problems.
guest ok	If this parameter is set for a service, then no password is required to connect to the service. Privileges will be those of the guest account.
invalid users	List of users that should not be allowed to login to this service.
only user	Controls whether connections with usernames not in the user list will be allowed.
read list	List of users that are given read-only access to a service. Users in this list will not be given write access, no matter what the read-only option is set to.
username	Refer to the `smb.conf` man page for more information; this is a complex and potentially misused parameter.
valid users	List of users that should be allowed to login to this service.
write list	List of users that are given read-write access to a service.

16.4 Access Controls on Shares

This section deals with how to configure Samba per-share access control restrictions. By default, Samba sets no restrictions on the share itself. Restrictions on the share itself can be set on MS Windows NT4/200x/XP

Table 16.3 File and Directory Permission-Based Controls

Control Parameter	Description, Action, Notes
create mask	Refer to the `smb.conf` man page.
directory mask	The octal modes used when converting DOS modes to UNIX modes when creating UNIX directories. See also directory security mask.
dos filemode	Enabling this parameter allows a user who has write access to the file to modify the permissions on it.
force create mode	This parameter specifies a set of UNIX-mode bit permissions that will always be set on a file created by Samba.
force directory mode	This parameter specifies a set of UNIX-mode bit permissions that will always be set on a directory created by Samba.
force directory security mode	Controls UNIX permission bits modified when a Windows NT client is manipulating UNIX permissions on a directory.
force security mode	Controls UNIX permission bits modified when a Windows NT client manipulates UNIX permissions.
hide unreadable	Prevents clients from seeing the existence of files that cannot be read.
hide unwriteable files	Prevents clients from seeing the existence of files that cannot be written to. Unwritable directories are shown as usual.
nt acl support	This parameter controls whether smbd will attempt to map UNIX permissions into Windows NT ACLs.
security mask	Controls UNIX permission bits modified when a Windows NT client is manipulating the UNIX permissions on a file.

shares. This can be an effective way to limit who can connect to a share. In the absence of specific restrictions, the default setting is to allow the global user `Everyone - Full Control` (full control, change and read).

At this time Samba does not provide a tool for configuring access control

settings on the share itself the only way to create those settings is to use either the NT4 Server Manager or the Windows 200x Microsoft Management Console (MMC) for Computer Management. There are currently no plans to provide this capability in the Samba command-line tool set.

Samba stores the per-share access control settings in a file called `share_info.tdb`. The location of this file on your system will depend on how Samba was compiled. The default location for Samba's tdb files is under `/usr/local/samba/var`. If the `tdbdump` utility has been compiled and installed on your system, then you can examine the contents of this file by executing **tdbdump share_info.tdb** in the directory containing the tdb files.

16.4.1 Share Permissions Management

The best tool for share permissions management is platform-dependent. Choose the best tool for your environment.

16.4.1.1 Windows NT4 Workstation/Server

The tool you need to manage share permissions on a Samba server from a Windows NT4 Workstation or Server is the NT Server Manager. Server Manager is shipped with Windows NT4 Server products but not with Windows NT4 Workstation. You can obtain the NT Server Manager for MS Windows NT4 Workstation from the Microsoft web site support[1] section. Instructions

1. Launch the NT4 Server Manager and click on the Samba server you want to administer. From the menu select **Computer**, then click on **Shared Directories**.
2. Click on the share that you wish to manage and click the **Properties** tab, then click the **Permissions** tab. Now you can add or change access control settings as you wish.

16.4.1.2 Windows 200x/XP

On MS Windows NT4/200x/XP system, ACLs on the share itself are set using native tools, usually from File Manager. For example, in Windows 200x,

[1] <http://support.microsoft.com/default.aspx?scid=kb;en-us;173673>

right-click on the shared folder, then select **Sharing**, then click on **Permissions**. The default Windows NT4/200x permission allows "Everyone" full control on the share.

MS Windows 200x and later versions come with a tool called the Computer Management snap-in for the MMC. This tool is located by clicking on **Control Panel** -> **Administrative Tools** -> **Computer Management**. Instructions

1. After launching the MMC with the Computer Management snap-in, click the menu item **Action** and select **Connect to another computer**. If you are not logged onto a domain you will be prompted to enter a domain login user identifier and a password. This will authenticate you to the domain. If you are already logged in with administrative privilege, this step is not offered.

2. If the Samba server is not shown in the **Select Computer** box, type in the name of the target Samba server in the field **Name:**. Now click the on **[+]** next to **System Tools**, then on the **[+]** next to **Shared Folders** in the left panel.

3. In the right panel, double-click on the share on which you wish to set access control permissions. Then click the tab **Share Permissions**. It is now possible to add access control entities to the shared folder. Remember to set what type of access (full control, change, read) you wish to assign for each entry.

WARNING

Be careful. If you take away all permissions from the `Everyone` user without removing this user, effectively no user will be able to access the share. This is a result of what is known as ACL precedence. Everyone with *no access* means that `MaryK` who is part of the group `Everyone` will have no access even if she is given explicit full control access.

16.5 MS Windows Access Control Lists and UNIX Interoperability

16.5.1 Managing UNIX Permissions Using NT Security Dialogs

Windows NT clients can use their native security settings dialog box to view and modify the underlying UNIX permissions.

This ability is careful not to compromise the security of the UNIX host on which Samba is running and still obeys all the file permission rules that a Samba administrator can set.

Samba does not attempt to go beyond POSIX ACLs, so the various finer-grained access control options provided in Windows are actually ignored.

NOTE

All access to UNIX/Linux system files via Samba is controlled by the operating system file access controls. When trying to figure out file access problems, it is vitally important to find the identity of the Windows user as it is presented by Samba at the point of file access. This can best be determined from the Samba log files.

16.5.2 Viewing File Security on a Samba Share

From an NT4/2000/XP client, right-click on any file or directory in a Samba-mounted drive letter or UNC path. When the menu pops up, click on the **Properties** entry at the bottom of the menu. This brings up the file `Properties` dialog box. Click on the **Security** tab and you will see three buttons: **Permissions**, **Auditing**, and **Ownership**. The **Auditing** button will cause either an error message "A requested privilege is not held by the client" to appear if the user is not the NT administrator, or a dialog intended to allow an administrator to add auditing requirements to a file if the user is logged on as the NT administrator. This dialog is nonfunctional with a Samba share at this time, because the only useful button, the **Add** button, will not currently allow a list of users to be seen.

16.5.3 Viewing File Ownership

Clicking on the **Ownership** button brings up a dialog box telling you who owns the given file. The owner name will be displayed like this:

```
SERVER\user (Long name)
```

SERVER is the NetBIOS name of the Samba server, *user* is the username of the UNIX user who owns the file, and *(Long name)* is the descriptive string identifying the user (normally found in the GECOS field of the UNIX password database). Click on the **Close** button to remove this dialog.

If the parameter *nt acl support* is set to `false`, the file owner will be shown as the NT user *Everyone*.

The **Take Ownership** button will not allow you to change the ownership of this file to yourself (clicking it will display a dialog box complaining that the user as whom you are currently logged onto the NT client cannot be found). The reason for this is that changing the ownership of a file is a privileged operation in UNIX, available only to the *root* user. Because clicking on this button causes NT to attempt to change the ownership of a file to the current user logged into the NT client, this will not work with Samba at this time.

There is an NT **chown** command that will work with Samba and allow a user with administrator privilege connected to a Samba server as root to change the ownership of files on both a local NTFS file system or remote mounted NTFS or Samba drive. This is available as part of the Seclib NT security library written by Jeremy Allison of the Samba Team and is downloadable from the main Samba FTP site.

16.5.4 Viewing File or Directory Permissions

The third button is the **Permissions** button. Clicking on it brings up a dialog box that shows both the permissions and the UNIX owner of the file or directory. The owner is displayed like this:

SERVER\user (Long name)

SERVER is the NetBIOS name of the Samba server, *user* is the username of the UNIX user who owns the file, and *(Long name)* is the descriptive

string identifying the user (normally found in the GECOS field of the UNIX password database).

If the parameter *nt acl support* is set to `false`, the file owner will be shown as the NT user `Everyone`, and the permissions will be shown as NT *Full Control*.

The permissions field is displayed differently for files and directories. Both are discussed next.

16.5.4.1 File Permissions

The standard UNIX user/group/world triplet and the corresponding `read, write, execute` permissions triplets are mapped by Samba into a three-element NT ACL with the "r", "w", and "x" bits mapped into the corresponding NT permissions. The UNIX world permissions are mapped into the global NT group `Everyone`, followed by the list of permissions allowed for the UNIX world. The UNIX owner and group permissions are displayed as an NT **user** icon and an NT **local group** icon, respectively, followed by the list of permissions allowed for the UNIX user and group.

Because many UNIX permission sets do not map into common NT names such as `read`, `change`, or `full control`, usually the permissions will be prefixed by the words `Special Access` in the NT display list.

But what happens if the file has no permissions allowed for a particular UNIX user group or world component? In order to allow *no permissions* to be seen and modified, Samba then overloads the NT `Take Ownership` ACL attribute (which has no meaning in UNIX) and reports a component with no permissions as having the NT **O** bit set. This was chosen, of course, to make it look like a zero, meaning zero permissions. More details on the decision behind this action are given below.

16.5.4.2 Directory Permissions

Directories on an NT NTFS file system have two different sets of permissions. The first set is the ACL set on the directory itself, which is usually displayed in the first set of parentheses in the normal `RW` NT style. This first set of permissions is created by Samba in exactly the same way as normal file permissions are, described above, and is displayed in the same way.

The second set of directory permissions has no real meaning in the UNIX permissions world and represents the `inherited` permissions that any file created within this directory would inherit.

Samba synthesizes these inherited permissions for NT by returning as an NT ACL the UNIX permission mode that a new file created by Samba on this share would receive.

16.5.5 Modifying File or Directory Permissions

Modifying file and directory permissions is as simple as changing the displayed permissions in the dialog box and clicking on **OK**. However, there are limitations that a user needs to be aware of, and also interactions with the standard Samba permission masks and mapping of DOS attributes that also need to be taken into account.

If the parameter *nt acl support* is set to `false`, any attempt to set security permissions will fail with an "Access Denied" message.

The first thing to note is that the **Add** button will not return a list of users in Samba (it will give an error message saying "The remote procedure call failed and did not execute"). This means that you can only manipulate the current user/group/world permissions listed in the dialog box. This actually works quite well because these are the only permissions that UNIX actually has.

If a permission triplet (either user, group, or world) is removed from the list of permissions in the NT dialog box, then when the **OK** button is pressed, it will be applied as *no permissions* on the UNIX side. If you view the permissions again, the *no permissions* entry will appear as the NT **O** flag, as described above. This allows you to add permissions back to a file or directory once you have removed them from a triplet component.

Because UNIX supports only the "r", "w", and "x" bits of an NT ACL, if other NT security attributes such as `Delete Access` are selected, they will be ignored when applied on the Samba server.

When setting permissions on a directory, the second set of permissions (in the second set of parentheses) is by default applied to all files within that directory. If this is not what you want, you must uncheck the **Replace permissions on existing files** checkbox in the NT dialog before clicking on **OK**.

If you wish to remove all permissions from a user/group/world component, you may either highlight the component and click on the **Remove** button or set the component to only have the special `Take Ownership` permission (displayed as **O**) highlighted.

16.5.6 Interaction with the Standard Samba "create mask" Parameters

There are four parameters that control interaction with the standard Samba *create mask* parameters:

- *security mask*
- *force security mode*
- *directory security mask*
- *force directory security mode*

When a user clicks on **OK** to apply the permissions, Samba maps the given permissions into a user/group/world r/w/x triplet set, and then checks the changed permissions for a file against the bits set in the *security mask* parameter. Any bits that were changed that are not set to *1* in this parameter are left alone in the file permissions.

Essentially, zero bits in the *security mask* may be treated as a set of bits the user is *not* allowed to change, and one bits are those the user is allowed to change.

If not explicitly set, this parameter defaults to the same value as the *create mask* parameter. To allow a user to modify all the user/group/world permissions on a file, set this parameter to 0777.

Next Samba checks the changed permissions for a file against the bits set in the *force security mode* parameter. Any bits that were changed that correspond to bits set to *1* in this parameter are forced to be set.

Essentially, bits set in the *force security mode* parameter may be treated as a set of bits that, when modifying security on a file, the user has always set to be *on*.

If not explicitly set, this parameter defaults to the same value as the *force create mode* parameter. To allow a user to modify all the user/group/world permissions on a file with no restrictions, set this parameter to 000. The *security mask* and *force security mode* parameters are applied to the change request in that order.

For a directory, Samba performs the same operations as described above for a file except it uses the parameter *directory security mask* instead of

security mask, and *force directory security mode* parameter instead of *force security mode*.

The *directory security mask* parameter by default is set to the same value as the *directory mask* parameter and the *force directory security mode* parameter by default is set to the same value as the *force directory mode* parameter. In this way Samba enforces the permission restrictions that an administrator can set on a Samba share, while still allowing users to modify the permission bits within that restriction.

If you want to set up a share that allows users full control in modifying the permission bits on their files and directories and does not force any particular bits to be set *on*, then set the following parameters in the `smb.conf` file in that share-specific section:

```
        security mask = 0777
        force security mode = 0
        directory security mask = 0777
        force directory security mode = 0
```

16.5.7 Interaction with the Standard Samba File Attribute Mapping

NOTE

Samba maps some of the DOS attribute bits (such as "read-only") into the UNIX permissions of a file. This means there can be a conflict between the permission bits set via the security dialog and the permission bits set by the file attribute mapping.

If a file has no UNIX read access for the owner, it will show up as "read-only" in the standard file attributes tabbed dialog. Unfortunately, this dialog is the same one that contains the security information in another tab.

What this can mean is that if the owner changes the permissions to allow himself or herself read access using the security dialog, clicks on **OK** to get back to the standard attributes tab dialog, and clicks on **OK** on that dialog, then NT will set the file permissions back to read-only (as that is what the attributes still say in the dialog). This means that after setting permissions and clicking on **OK** to get back to the attributes dialog, you should always press **Cancel** rather than **OK** to ensure that your changes are not overridden.

16.5.8 Windows NT/200X ACLs and POSIX ACLs Limitations

Windows administrators are familiar with simple ACL controls, and they typically consider that UNIX user/group/other (ugo) permissions are inadequate and not sufficiently fine-grained.

Competing SMB implementations differ in how they handle Windows ACLs. Samba handles Windows ACLs from the perspective of UNIX file system ad-

ministration and thus adopts the limitations of POSIX ACLs. Therefore, where POSIX ACLs lack a capability of the Windows NT/200X ACLs, the POSIX semantics and limitations are imposed on the Windows administrator.

POSIX ACLs present an interesting challenge to the UNIX administrator and therefore force a compromise to be applied to Windows ACLs administration. POSIX ACLs are not covered by an official standard; rather, the latest standard is a draft standard 1003.1e revision 17. This is the POSIX document on which the Samba implementation has been implemented.

UNIX vendors differ in the manner in which POSIX ACLs are implemented. There are a number of Linux file systems that support ACLs. Samba has to provide a way to make transparent all the differences between the various implementations of POSIX ACLs. The pressure for ACLs support in Samba has noticeably increased the pressure to standardize ACLs support in the UNIX world.

Samba has to deal with the complicated matter of handling the challenge of the Windows ACL that implements *inheritance*, a concept not anticipated by POSIX ACLs as implemented in UNIX file systems. Samba provides support for *masks* that permit normal ugo and ACLs functionality to be overrided. This further complicates the way in which Windows ACLs must be implemented.

16.5.8.1 UNIX POSIX ACL Overview

In examining POSIX ACLs we must consider the manner in which they operate for both files and directories. File ACLs have the following significance:

```
# file: testfile      <- the file name
# owner: jeremy       <-- the file owner
# group: users        <-- the POSIX group owner
user::rwx             <-- perms for the file owner (user)
user:tpot:r-x         <-- perms for the additional user 'tpot'
group::r--            <-- perms for the file group owner (group)
group:engrs:r--       <-- perms for the additonal group 'engineers'
mask:rwx              <-- the mask that is 'ANDed' with groups
```

```
other::---            <-- perms applied to everyone else (other)
```

Directory ACLs have the following signficance:

```
# file: testdir       <-- the directory name
# owner: jeremy       <-- the directory owner
# group: jeremy       <-- the POSIX group owner
user::rwx             <-- directory perms for owner (user)
group::rwx            <-- directory perms for owning group (group)
mask::rwx             <-- the mask that is 'ANDed' with group perms
other:r-x             <-- perms applied to everyone else (other)
default:user::rwx     <-- inherited owner perms
default:user:tpot:rwx <-- inherited extra perms for user 'tpot'
default:group::r-x    <-- inherited group perms
default:mask:rwx      <-- inherited default mask
default:other:---     <-- inherited permissions for everyone (other)
```

16.5.8.2 Mapping of Windows File ACLs to UNIX POSIX ACLs

Microsoft Windows NT4/200X ACLs must of necessity be mapped to POSIX ACLs. The mappings for file permissions are shown in Table 16.5. The # character means this flag is set only when the Windows administrator sets the `Full Control` flag on the file.

As can be seen from the mapping table, there is no one-to-one mapping capability, and therefore Samba must make a logical mapping that will permit Windows to operate more-or-less the way that is intended by the administrator.

In general the mapping of UNIX POSIX user/group/other permissions will be mapped to Windows ACLs. This has precedence over the creation of POSIX ACLs. POSIX ACLs are necessary to establish access controls for users and groups other than the user and group that own the file or directory.

The UNIX administrator can set any directory permission from within the UNIX environment. The Windows administrator is more restricted in that it is not possible from within Windows Explorer to remove read permission for the file owner.

16.5.8.3 Mapping of Windows Directory ACLs to UNIX POSIX ACLs

Interesting things happen in the mapping of UNIX POSIX directory permissions and UNIX POSIX ACLs to Windows ACEs (Access Control Entries, the discrete components of an ACL) are mapped to Windows directory ACLs.

Directory permissions function in much the same way as shown for file permissions, but there are some notable exceptions and a few peculiarities that the astute administrator will want to take into account in the setting up of directory permissions.

16.6 Common Errors

File, directory, and share access problems are common topics on the mailing list. The following are examples recently taken from the mailing list.

16.6.1 Users Cannot Write to a Public Share

The following complaint has frequently been voiced on the Samba mailing list: "We are facing some troubles with file/directory permissions. I can log on the domain as admin user (root), and there's a public share on which everyone needs to have permission to create/modify files, but only root can change the file, no one else can. We need to constantly go to the server to `chgrp -R users *` and `chown -R nobody *` to allow other users to change the file."

Here is one way the problem can be solved:

1. Go to the top of the directory that is shared.
2. Set the ownership to whatever public user and group you want

```
$ find 'directory_name' -type d -exec chown user:group {}\;
$ find 'directory_name' -type d -exec chmod 2775 {}\;
$ find 'directory_name' -type f -exec chmod 0775 {}\;
$ find 'directory_name' -type f -exec chown user:group {}\;
```

> **NOTE**
>
> The above will set the `SGID bit` on all directories. Read your UNIX/Linux man page on what that does. This ensures that all files and directories that are created in the directory tree will be owned by the current user and will be owned by the group that owns the directory in which it is created.

3. Directory is */foodbar*:

```
$ chown jack:engr /foodbar
```

> **NOTE**
>
> This is the same as doing:
>
> ```
> $ chown jack /foodbar
> $ chgrp engr /foodbar
> ```

4. Now type:

```
$ chmod 2775 /foodbar
$ ls -al /foodbar/..
```

You should see:

```
drwxrwsr-x  2 jack  engr    48 2003-02-04 09:55 foodbar
```

5. Now type:

```
$ su - jill
$ cd /foodbar
$ touch Afile
$ ls -al
```

   You should see that the file `Afile` created by Jill will have ownership and permissions of Jack, as follows:

```
-rw-r--r--  1 jill  engr     0 2007-01-18 19:41 Afile
```

6. If the user that must have write permission in the directory is not a member of the group *engr* set in the `smb.conf` entry for the share:

```
        force group = engr
```

16.6.2 File Operations Done as *root* with *force user* Set

When you have a user in *admin users*, Samba will always do file operations for this user as *root*, even if *force user* has been set.

16.6.3 MS Word with Samba Changes Owner of File

Question: "When user B saves a word document that is owned by user A, the updated file is now owned by user B. Why is Samba doing this? How do I fix this?"

Answer: Word does the following when you modify/change a Word document: MS Word creates a new document with a temporary name. Word then closes the old document and deletes it, then renames the new document to the original document name. There is no mechanism by which Samba can in any way know that the new document really should be owned by the owners of the original file. Samba has no way of knowing that the file will be renamed by MS Word. As far as Samba is able to tell, the file that gets created is a new file, not one that the application (Word) is updating.

There is a workaround to solve the permissions problem. It involves understanding how you can manage file system behavior from within the `smb.conf` file, as well as understanding how UNIX file systems work. Set on the directory in which you are changing Word documents: **chmod g+s 'directory_name'.** This ensures that all files will be created with the group that owns the directory. In `smb.conf` share declaration section set:

```
        force create mode = 0660
        force directory mode = 0770
```

These two settings will ensure that all directories and files that get created in the share will be readable/writable by the owner and group set on the directory itself.

Table 16.4 Other Controls

Control Parameter	Description, Action, Notes
case sensitive, *default case*, *short preserve case*	This means that all file name lookup will be done in a case-sensitive manner. Files will be created with the precise file name Samba received from the MS Windows client.
csc policy	Client-side caching policy parallels MS Windows client-side file caching capabilities.
dont descend	Allows specifying a comma-delimited list of directories that the server should always show as empty.
dos filetime resolution	This option is mainly used as a compatibility option for Visual C++ when used against Samba shares.
dos filetimes	DOS and Windows allow users to change file timestamps if they can write to the file. POSIX semantics prevent this. This option allows DOS and Windows behavior.
fake oplocks	Oplocks are the way that SMB clients get permission from a server to locally cache file operations. If a server grants an oplock, the client is free to assume that it is the only one accessing the file, and it will aggressively cache file data.
hide dot files, *hide files*, *veto files*	Note: MS Windows Explorer allows override of files marked as hidden so they will still be visible.
read only	If this parameter is yes, then users of a service may not create or modify files in the service's directory.
veto files	List of files and directories that are neither visible nor accessible.

Table 16.5 How Windows File ACLs Map to UNIX POSIX File ACLs

Windows ACE	**File Attribute Flag**
Full Control	#
Traverse Folder/Execute File	x
List Folder/Read Data	r
Read Attributes	r
Read Extended Attribures	r
Create Files/Write Data	w
Create Folders/Append Data	w
Write Attributes	w
Write Extended Attributes	w
Delete Subfolders and Files	w
Delete	#
Read Permissions	all
Change Permissions	#
Take Ownership	#

Chapter 17

FILE AND RECORD LOCKING

One area that causes trouble for many network administrators is locking. The extent of the problem is readily evident from searches over the Internet.

17.1 Features and Benefits

Samba provides all the same locking semantics that MS Windows clients expect and that MS Windows NT4/200x servers also provide.

The term *locking* has exceptionally broad meaning and covers a range of functions that are all categorized under this one term.

Opportunistic locking is a desirable feature when it can enhance the perceived performance of applications on a networked client. However, the opportunistic locking protocol is not robust and therefore can encounter problems when invoked beyond a simplistic configuration or on extended slow or faulty networks. In these cases, operating system management of opportunistic locking and/or recovering from repetitive errors can offset the perceived performance advantage that it is intended to provide.

The MS Windows network administrator needs to be aware that file and record locking semantics (behavior) can be controlled either in Samba or by way of registry settings on the MS Windows client.

NOTE

Sometimes it is necessary to disable locking control settings on the Samba server as well as on each MS Windows client!

17.2 Discussion

There are two types of locking that need to be performed by an SMB server. The first is *record locking* that allows a client to lock a range of bytes in an open file. The second is the *deny modes* that are specified when a file is open.

Record locking semantics under UNIX are very different from record locking under Windows. Versions of Samba before 2.2 have tried to use the native fcntl() UNIX system call to implement proper record locking between different Samba clients. This cannot be fully correct for several reasons. The simplest is that a Windows client is allowed to lock a byte range up to 2ˆ32 or 2ˆ64, depending on the client OS. The UNIX locking only supports byte ranges up to 2ˆ31. So it is not possible to correctly satisfy a lock request above 2ˆ31. There are many more differences, too many to be listed here.

Samba 2.2 and above implement record locking completely independently of the underlying UNIX system. If a byte-range lock that the client requests happens to fall into the range of 0 to 2ˆ31, Samba hands this request down to the UNIX system. No other locks can be seen by UNIX, anyway.

Strictly speaking, an SMB server should check for locks before every read and write call on a file. Unfortunately, with the way fcntl() works, this can be slow and may overstress the **rpc.lockd**. This is almost always unnecessary because clients are independently supposed to make locking calls before reads and writes if locking is important to them. By default, Samba only makes locking calls when explicitly asked to by a client, but if you set *strict locking* = yes, it will make lock checking calls on *every* read and write call.

You can also disable byte-range locking completely by using *locking* = no. This is useful for those shares that do not support locking or do not need it (such as CD-ROMs). In this case, Samba fakes the return codes of locking calls to tell clients that everything is okay.

The second class of locking is the *deny modes*. These are set by an application when it opens a file to determine what types of access should be allowed simultaneously with its open. A client may ask for `DENY_NONE`, `DENY_READ`, `DENY_WRITE`, or `DENY_ALL`. There are also special compatibility modes called `DENY_FCB` and `DENY_DOS`.

17.2.1 Opportunistic Locking Overview

Opportunistic locking (oplocks) is invoked by the Windows file system (as opposed to an API) via registry entries (on the server and the client) for the purpose of enhancing network performance when accessing a file residing on a server. Performance is enhanced by caching the file locally on the client that allows the following:

Read-ahead: The client reads the local copy of the file, eliminating network latency.

Write caching: The client writes to the local copy of the file, eliminating network latency.

Lock caching: The client caches application locks locally, eliminating network latency.

The performance enhancement of oplocks is due to the opportunity of exclusive access to the file — even if it is opened with deny-none — because Windows monitors the file's status for concurrent access from other processes.

Windows Defines Four Kinds of Oplocks:

Level1 Oplock The redirector sees that the file was opened with deny none (allowing concurrent access), verifies that no other process is accessing the file, checks that oplocks are enabled, then grants deny-all/read-write/exclusive access to the file. The client now performs operations on the cached local file.

If a second process attempts to open the file, the open is deferred while the redirector "breaks" the original oplock. The oplock break signals the caching client to write the local file back to the server, flush the

local locks, and discard read-ahead data. The break is then complete, the deferred open is granted, and the multiple processes can enjoy concurrent file access as dictated by mandatory or byte-range locking options. However, if the original opening process opened the file with a share mode other than deny-none, then the second process is granted limited or no access, despite the oplock break.

Level2 Oplock Performs like a Level1 oplock, except caching is only operative for reads. All other operations are performed on the server disk copy of the file.

Filter Oplock Does not allow write or delete file access.

Batch Oplock Manipulates file openings and closings and allows caching of file attributes.

An important detail is that oplocks are invoked by the file system, not an application API. Therefore, an application can close an oplocked file, but the file system does not relinquish the oplock. When the oplock break is issued, the file system then simply closes the file in preparation for the subsequent open by the second process.

Opportunistic locking is actually an improper name for this feature. The true benefit of this feature is client-side data caching, and oplocks is merely a notification mechanism for writing data back to the networked storage disk. The limitation of oplocks is the reliability of the mechanism to process an oplock break (notification) between the server and the caching client. If this exchange is faulty (usually due to timing out for any number of reasons), then the client-side caching benefit is negated.

The actual decision that a user or administrator should consider is whether it is sensible to share among multiple users data that will be cached locally on a client. In many cases the answer is no. Deciding when to cache or not cache data is the real question, and thus oplocks should be treated as a toggle for client-side caching. Turn it "on" when client-side caching is desirable and reliable. Turn it "off" when client-side caching is redundant, unreliable, or counterproductive.

Oplocks is by default set to "on" by Samba on all configured shares, so careful attention should be given to each case to determine if the potential

benefit is worth the potential for delays. The following recommendations will help to characterize the environment where oplocks may be effectively configured.

Windows oplocks is a lightweight performance-enhancing feature. It is not a robust and reliable protocol. Every implementation of oplocks should be evaluated as a trade-off between perceived performance and reliability. Reliability decreases as each successive rule above is not enforced. Consider a share with oplocks enabled, over a wide-area network, to a client on a South Pacific atoll, on a high-availability server, serving a mission-critical multiuser corporate database during a tropical storm. This configuration will likely encounter problems with oplocks.

Oplocks can be beneficial to perceived client performance when treated as a configuration toggle for client-side data caching. If the data caching is likely to be interrupted, then oplock usage should be reviewed. Samba enables oplocks by default on all shares. Careful attention should be given to the client usage of shared data on the server, the server network reliability, and the oplocks configuration of each share. In mission-critical, high-availability environments, data integrity is often a priority. Complex and expensive configurations are implemented to ensure that if a client loses connectivity with a file server, a failover replacement will be available immediately to provide continuous data availability.

Windows client failover behavior is more at risk of application interruption than other platforms because it is dependent upon an established TCP transport connection. If the connection is interrupted — as in a file server failover — a new session must be established. It is rare for Windows client applications to be coded to recover correctly from a transport connection loss; therefore, most applications will experience some sort of interruption — at worst, abort and require restarting.

If a client session has been caching writes and reads locally due to oplocks, it is likely that the data will be lost when the application restarts or recovers from the TCP interrupt. When the TCP connection drops, the client state is lost. When the file server recovers, an oplock break is not sent to the client. In this case, the work from the prior session is lost. Observing this scenario with oplocks disabled and with the client writing data to the file server real-time, the failover will provide the data on disk as it existed at the time of the disconnect.

In mission-critical, high-availability environments, careful attention should be given to oplocks. Ideally, comprehensive testing should be done with all affected applications with oplocks enabled and disabled.

17.2.1.1 Exclusively Accessed Shares

Oplocks is most effective when it is confined to shares that are exclusively accessed by a single user, or by only one user at a time. Because the true value of oplocks is the local client caching of data, any operation that interrupts the caching mechanism will cause a delay.

Home directories are the most obvious examples of where the performance benefit of oplocks can be safely realized.

17.2.1.2 Multiple-Accessed Shares or Files

As each additional user accesses a file in a share with oplocks enabled, the potential for delays and resulting perceived poor performance increases. When multiple users are accessing a file on a share that has oplocks enabled, the management impact of sending and receiving oplock breaks and the resulting latency while other clients wait for the caching client to flush data offset the performance gains of the caching user.

As each additional client attempts to access a file with oplocks set, the potential performance improvement is negated and eventually results in a performance bottleneck.

17.2.1.3 UNIX or NFS Client-Accessed Files

Local UNIX and NFS clients access files without a mandatory file-locking mechanism. Thus, these client platforms are incapable of initiating an oplock break request from the server to a Windows client that has a file cached. Local UNIX or NFS file access can therefore write to a file that has been cached by a Windows client, which exposes the file to likely data corruption.

If files are shared between Windows clients and either local UNIX or NFS users, turn oplocks off.

17.2.1.4 Slow and/or Unreliable Networks

The biggest potential performance improvement for oplocks occurs when the client-side caching of reads and writes delivers the most differential over sending those reads and writes over the wire. This is most likely to occur when the network is extremely slow, congested, or distributed (as in a WAN). However, network latency also has a high impact on the reliability of the oplock break mechanism, and thus increases the likelihood of encountering oplock problems that more than offset the potential perceived performance gain. Of course, if an oplock break never has to be sent, then this is the most advantageous scenario in which to utilize oplocks.

If the network is slow, unreliable, or a WAN, then do not configure oplocks if there is any chance of multiple users regularly opening the same file.

17.2.1.5 Multiuser Databases

Multiuser databases clearly pose a risk due to their very nature — they are typically heavily accessed by numerous users at random intervals. Placing a multiuser database on a share with oplocks enabled will likely result in a locking management bottleneck on the Samba server. Whether the database application is developed in-house or a commercially available product, ensure that the share has oplocks disabled.

17.2.1.6 PDM Data Shares

Process data management (PDM) applications such as IMAN, Enovia, and Clearcase are increasing in usage with Windows client platforms and therefore with SMB datastores. PDM applications manage multiuser environments for critical data security and access. The typical PDM environment is usually associated with sophisticated client design applications that will load data locally as demanded. In addition, the PDM application will usually monitor the data state of each client. In this case, client-side data caching is best left to the local application and PDM server to negotiate and maintain. It is appropriate to eliminate the client OS from any caching tasks, and the server from any oplocks management, by disabling oplocks on the share.

17.2.1.7 Beware of Force User

Samba includes an `smb.conf` parameter called *force user* that changes the user accessing a share from the incoming user to whatever user is defined by the `smb.conf` variable. If oplocks is enabled on a share, the change in user access causes an oplock break to be sent to the client, even if the user has not explicitly loaded a file. In cases where the network is slow or unreliable, an oplock break can become lost without the user even accessing a file. This can cause apparent performance degradation as the client continually reconnects to overcome the lost oplock break.

Avoid the combination of the following:

- *force user* in the `smb.conf` share configuration.
- Slow or unreliable networks.
- Oplocks enabled.

17.2.1.8 Advanced Samba Oplocks Parameters

Samba provides oplock parameters that allow the administrator to adjust various properties of the oplock mechanism to account for timing and usage levels. These parameters provide good versatility for implementing oplocks in environments where they would likely cause problems. The parameters are *oplock break wait time*, and *oplock contention limit*.

For most users, administrators, and environments, if these parameters are required, then the better option is simply to turn oplocks off. The Samba SWAT help text for both parameters reads: "Do not change this parameter unless you have read and understood the Samba oplock code." This is good advice.

17.2.1.9 Mission-Critical, High-Availability

In mission-critical, high-availability environments, data integrity is often a priority. Complex and expensive configurations are implemented to ensure that if a client loses connectivity with a file server, a failover replacement will be available immediately to provide continuous data availability.

Windows client failover behavior is more at risk of application interruption than other platforms because it is dependent upon an established TCP

transport connection. If the connection is interrupted — as in a file server failover — a new session must be established. It is rare for Windows client applications to be coded to recover correctly from a transport connection loss; therefore, most applications will experience some sort of interruption — at worst, abort and require restarting.

If a client session has been caching writes and reads locally due to oplocks, it is likely that the data will be lost when the application restarts or recovers from the TCP interrupt. When the TCP connection drops, the client state is lost. When the file server recovers, an oplock break is not sent to the client. In this case, the work from the prior session is lost. Observing this scenario with oplocks disabled, if the client was writing data to the file server real-time, then the failover will provide the data on disk as it existed at the time of the disconnect.

In mission-critical, high-availability environments, careful attention should be given to oplocks. Ideally, comprehensive testing should be done with all affected applications with oplocks enabled and disabled.

17.3 Samba Oplocks Control

Oplocks is a unique Windows file locking feature. It is not really file locking, but is included in most discussions of Windows file locking, so is considered a de facto locking feature. Oplocks is actually part of the Windows client file caching mechanism. It is not a particularly robust or reliable feature when implemented on the variety of customized networks that exist in enterprise computing.

Like Windows, Samba implements oplocks as a server-side component of the client caching mechanism. Because of the lightweight nature of the Windows feature design, effective configuration of oplocks requires a good understanding of its limitations, and then applying that understanding when configuring data access for each particular customized network and client usage state.

Oplocks essentially means that the client is allowed to download and cache a file on its hard drive while making changes; if a second client wants to access the file, the first client receives a break and must synchronize the file back to the server. This can give significant performance gains in some cases; some programs insist on synchronizing the contents of the entire file back to the server for a single change.

Level1 Oplocks (also known as just plain "oplocks") is another term for opportunistic locking.

Level2 Oplocks provides opportunistic locking for a file that will be treated as *read only*. Typically this is used on files that are read-only or on files that the client has no initial intention to write to at time of opening the file.

Kernel Oplocks are essentially a method that allows the Linux kernel to co-exist with Samba's oplocked files, although this has provided better integration of MS Windows network file locking with the underlying OS. SGI IRIX and Linux are the only two OSs that are oplock-aware at this time.

Unless your system supports kernel oplocks, you should disable oplocks if you are accessing the same files from both UNIX/Linux and SMB clients. Regardless, oplocks should always be disabled if you are sharing a database file (e.g., Microsoft Access) between multiple clients, because any break the first client receives will affect synchronization of the entire file (not just the single record), which will result in a noticeable performance impairment and, more likely, problems accessing the database in the first place. Notably, Microsoft Outlook's personal folders (*.pst) react quite badly to oplocks. If in doubt, disable oplocks and tune your system from that point.

If client-side caching is desirable and reliable on your network, you will benefit from turning on oplocks. If your network is slow and/or unreliable, or you are sharing your files among other file sharing mechanisms (e.g., NFS) or across a WAN, or multiple people will be accessing the same files frequently, you probably will not benefit from the overhead of your client sending oplock breaks and will instead want to disable oplocks for the share.

Another factor to consider is the perceived performance of file access. If oplocks provide no measurable speed benefit on your network, it might not be worth the hassle of dealing with them.

17.3.1 Example Configuration

In the following section we examine two distinct aspects of Samba locking controls.

17.3.1.1 Disabling Oplocks

You can disable oplocks on a per-share basis with the following:

```
[acctdata]
        oplocks = False
        level2 oplocks = False
```

The default oplock type is Level1. Level2 oplocks are enabled on a per-share basis in the `smb.conf` file.

Alternately, you could disable oplocks on a per-file basis within the share:

```
        veto oplock files = /*.mdb/*.MDB/*.dbf/*.DBF/
```

If you are experiencing problems with oplocks, as apparent from Samba's log entries, you may want to play it safe and disable oplocks and Level2 oplocks.

17.3.1.2 Disabling Kernel Oplocks

Kernel oplocks is an `smb.conf` parameter that notifies Samba (if the UNIX kernel has the capability to send a Windows client an oplock break) when a UNIX process is attempting to open the file that is cached. This parameter addresses sharing files between UNIX and Windows with oplocks enabled on the Samba server: the UNIX process can open the file that is Oplocked (cached) by the Windows client and the smbd process will not send an oplock break, which exposes the file to the risk of data corruption. If the UNIX kernel has the ability to send an oplock break, then the kernel oplocks parameter enables Samba to send the oplock break. Kernel oplocks are enabled on a per-server basis in the `smb.conf` file.

```
        kernel oplocks = yes
```

The default is no.

Veto oplocks is an `smb.conf` parameter that identifies specific files for which oplocks are disabled. When a Windows client opens a file that has been configured for veto oplocks, the client will not be granted the oplock, and all operations will be executed on the original file on disk instead of a client-cached file copy. By explicitly identifying files that are shared with UNIX processes and disabling oplocks for those files, the server-wide oplock configuration can be enabled to allow Windows clients to utilize the performance benefit of file caching without the risk of data corruption. Veto oplocks can

be enabled on a per-share basis, or globally for the entire server, in the `smb.conf` file as shown in Example 17.3.1.

Example 17.3.1 Share with Some Files Oplocked

```
[global]
        veto oplock files = /filename.htm/*.txt/
 [share_name]
        veto oplock files = /*.exe/filename.ext/
```

oplock break wait time is an `smb.conf` parameter that adjusts the time interval for Samba to reply to an oplock break request. Samba recommends: "Do not change this parameter unless you have read and understood the Samba oplock code." Oplock break wait time can only be configured globally in the `smb.conf` file as shown:

```
        oplock break wait time =  0 (default)
```

Oplock break contention limit is an `smb.conf` parameter that limits the response of the Samba server to grant an oplock if the configured number of contending clients reaches the limit specified by the parameter. Samba recommends "Do not change this parameter unless you have read and understood the Samba oplock code." Oplock break contention limit can be enabled on a per-share basis, or globally for the entire server, in the `smb.conf` file as shown in Example 17.3.2.

Example 17.3.2 Configuration with Oplock Break Contention Limit

```
[global]
        oplock break contention limit =  2 (default)
 [share_name]
        oplock break contention limit =  2 (default)
```

17.4 MS Windows Oplocks and Caching Controls

There is a known issue when running applications (like Norton Antivirus) on a Windows 2000/ XP workstation computer that can affect any application attempting to access shared database files across a network. This is a result of a default setting configured in the Windows 2000/XP operating system.

When a workstation attempts to access shared data files located on another Windows 2000/XP computer, the Windows 2000/XP operating system will attempt to increase performance by locking the files and caching information locally. When this occurs, the application is unable to properly function, which results in an "Access Denied" error message being displayed during network operations.

All Windows operating systems in the NT family that act as database servers for data files (meaning that data files are stored there and accessed by other Windows PCs) may need to have oplocks disabled in order to minimize the risk of data file corruption. This includes Windows 9x/Me, Windows NT, Windows 200x, and Windows XP. [1]

If you are using a Windows NT family workstation in place of a server, you must also disable oplocks on that workstation. For example, if you use a PC with the Windows NT Workstation operating system instead of Windows NT Server, and you have data files located on it that are accessed from other Windows PCs, you may need to disable oplocks on that system.

The major difference is the location in the Windows registry where the values for disabling oplocks are entered. Instead of the LanManServer location, the LanManWorkstation location may be used.

You can verify (change or add, if necessary) this registry value using the Windows Registry Editor. When you change this registry value, you will have to reboot the PC to ensure that the new setting goes into effect.

The location of the client registry entry for oplocks has changed in Windows 2000 from the earlier location in Microsoft Windows NT.

NOTE

Windows 2000 will still respect the EnableOplocks registry value used to disable oplocks in earlier versions of Windows.

You can also deny the granting of oplocks by changing the following registry entries:

[1] Microsoft has documented this in Knowledge Base article 300216.

```
HKEY_LOCAL_MACHINE\System\
   CurrentControlSet\Services\MRXSmb\Parameters\

   OplocksDisabled REG_DWORD 0 or 1
   Default: 0 (not disabled)
```

NOTE

The OplocksDisabled registry value configures Windows clients to either request or not request oplocks on a remote file. To disable oplocks, the value of OplocksDisabled must be set to 1.

```
HKEY_LOCAL_MACHINE\System\
   CurrentControlSet\Services\LanmanServer\Parameters

   EnableOplocks REG_DWORD 0 or 1
   Default: 1 (Enabled by Default)

   EnableOpLockForceClose REG_DWORD 0 or 1
   Default: 0 (Disabled by Default)
```

NOTE

The EnableOplocks value configures Windows-based servers (including Workstations sharing files) to allow or deny oplocks on local files.

To force closure of open oplocks on close or program exit, EnableOpLockForceClose must be set to 1.

An illustration of how Level2 oplocks work follows:

- Station 1 opens the file requesting oplock.
- Since no other station has the file open, the server grants station 1 exclusive oplock.
- Station 2 opens the file requesting oplock.
- Since station 1 has not yet written to the file, the server asks station 1 to break to Level2 oplock.
- Station 1 complies by flushing locally buffered lock information to the server.
- Station 1 informs the server that it has broken to level2 Oplock (alternately, station 1 could have closed the file).
- The server responds to station 2's open request, granting it Level2 oplock. Other stations can likewise open the file and obtain Level2 oplock.
- Station 2 (or any station that has the file open) sends a write request SMB. The server returns the write response.
- The server asks all stations that have the file open to break to none, meaning no station holds any oplock on the file. Because the workstations can have no cached writes or locks at this point, they need not respond to the break-to-none advisory; all they need do is invalidate locally cashed read-ahead data.

17.4.1 Workstation Service Entries

```
\HKEY_LOCAL_MACHINE\System\
   CurrentControlSet\Services\LanmanWorkstation\Parameters

UseOpportunisticLocking   REG_DWORD   0 or 1
Default: 1 (true)
```

This indicates whether the redirector should use oplocks performance enhancement. This parameter should be disabled only to isolate problems.

17.4.2 Server Service Entries

```
\HKEY_LOCAL_MACHINE\System\
   CurrentControlSet\Services\LanmanServer\Parameters

EnableOplocks   REG_DWORD   0 or 1
Default: 1 (true)
```

This specifies whether the server allows clients to use oplocks on files. Oplocks are a significant performance enhancement, but have the potential to cause lost cached data on some networks, particularly WANs.

```
MinLinkThroughput   REG_DWORD   0 to infinite bytes per second
Default: 0
```

This specifies the minimum link throughput allowed by the server before it disables raw I/O and oplocks for this connection.

```
MaxLinkDelay   REG_DWORD   0 to 100,000 seconds
Default: 60
```

This specifies the maximum time allowed for a link delay. If delays exceed this number, the server disables raw I/O and oplocks for this connection.

```
OplockBreakWait   REG_DWORD   10 to 180 seconds
Default: 35
```

This specifies the time that the server waits for a client to respond to an oplock break request. Smaller values can allow detection of crashed clients more quickly but can potentially cause loss of cached data.

17.5 Persistent Data Corruption

If you have applied all of the settings discussed in this chapter but data corruption problems and other symptoms persist, here are some additional things to check out.

We have credible reports from developers that faulty network hardware, such as a single faulty network card, can cause symptoms similar to read caching and data corruption. If you see persistent data corruption even after repeated re-indexing, you may have to rebuild the data files in question. This involves creating a new data file with the same definition as the file to be rebuilt and transferring the data from the old file to the new one. There are several known methods for doing this that can be found in our knowledge base.

17.6 Common Errors

In some sites locking problems surface as soon as a server is installed; in other sites locking problems may not surface for a long time. Almost without exception, when a locking problem does surface, it will cause embarrassment and potential data corruption.

Over the past few years there have been a number of complaints on the Samba mailing lists that have claimed that Samba caused data corruption. Three causes have been identified so far:

- Incorrect configuration of oplocks (incompatible with the application being used). This is a common problem even where MS Windows NT4 or MS Windows 200x-based servers were in use. It is imperative that the software application vendors' instructions for configuration of file locking should be followed. If in doubt, disable oplocks on both the server and the client. Disabling of all forms of file caching on the MS Windows client may be necessary also.
- Defective network cards, cables, or hubs/switches. This is generally a more prevalent factor with low-cost networking hardware, although occasionally there have also been problems with incompatibilities in more up-market hardware.
- There have been some random reports of Samba log files being written over data files. This has been reported by very few sites (about five

in the past 3 years) and all attempts to reproduce the problem have failed. The Samba Team has been unable to catch this happening and thus unable to isolate any particular cause. Considering the millions of systems that use Samba, for the sites that have been affected by this as well as for the Samba Team, this is a frustrating and vexing challenge. If you see this type of thing happening, please create a bug report on Samba Bugzilla[2] without delay. Make sure that you give as much information as you possibly can to help isolate the cause and to allow replication of the problem (an essential step in problem isolation and correction).

17.6.1 locking.tdb Error Messages

"We are seeing lots of errors in the Samba logs, like:"

```
tdb(/usr/local/samba_2.2.7/var/locks/locking.tdb): rec_read bad magic
 0x4d6f4b61 at offset=36116
```

"What do these mean?"

This error indicates a corrupted tdb. Stop all instances of smbd, delete locking.tdb, and restart smbd.

17.6.2 Problems Saving Files in MS Office on Windows XP

This is a bug in Windows XP. More information can be found in Microsoft Knowledge Base article 812937[3] .

17.6.3 Long Delays Deleting Files over Network with XP SP1

"It sometimes takes approximately 35 seconds to delete files over the network after XP SP1 has been applied." This is a bug in Windows XP. More information can be found in Microsoft Knowledge Base article 811492[4] .

[2] <https://bugzilla.samba.org>
[3] <http://support.microsoft.com/?id=812937>
[4] <http://support.microsoft.com/?id=811492>

17.7 Additional Reading

You may want to check for an updated documentation regarding file and record locking issues on the Microsoft Support[5] web site. Additionally, search for the word `locking` on the Samba web[6] site.

Section of the Microsoft MSDN Library on opportunistic locking:

Microsoft Knowledge Base, "Maintaining Transactional Integrity with OPLOCKS", Microsoft Corporation, April 1999, Microsoft KB Article 224992[7].

Microsoft Knowledge Base, "Configuring Opportunistic Locking in Windows 2000", Microsoft Corporation, April 2001 Microsoft KB Article 296264[8].

Microsoft Knowledge Base, "PC Ext: Explanation of Opportunistic Locking on Windows NT", Microsoft Corporation, April 1995 Microsoft KB Article 129202[9].

[5] <http://support.microsoft.com/>
[6] <http://www.samba.org/>
[7] <http://support.microsoft.com/?id=224992>
[8] <http://support.microsoft.com/?id=296264>
[9] <http://support.microsoft.com/?id=129202>

Chapter 18

SECURING SAMBA

18.1 Introduction

The information contained in this chapter applies in general to all Samba installations. Security is everyone's concern in the information technology world. A surprising number of Samba servers are being installed on machines that have direct internet access, thus security is made more critical than it would have been had the server been located behind a firewall and on a private network. Paranoia regarding server security is causing some network administrators to insist on the installation of robust firewalls even on servers that are located inside secured networks. This chapter provides information to assist the administrator who understands how to create the needed barriers and deterents against "the enemy", no matter where [s]he may come from.

> A new apprentice reported for duty to the chief engineer of a boiler house. He said, "Here I am, if you will show me the boiler I'll start working on it." Then engineer replied, "You're leaning on it!"

Security concerns are just like that. You need to know a little about the subject to appreciate how obvious most of it really is. The challenge for most of us is to discover that first morsel of knowledge with which we may unlock the secrets of the masters.

18.2 Features and Benefits

There are three levels at which security principles must be observed in order to render a site at least moderately secure. They are the perimeter firewall,

the configuration of the host server that is running Samba, and Samba itself.

Samba permits a most flexible approach to network security. As far as possible Samba implements the latest protocols to permit more secure MS Windows file and print operations.

Samba can be secured from connections that originate from outside the local network. This can be done using *host-based protection*, using Samba's implementation of a technology known as "tcpwrappers," or it may be done be using *interface-based exclusion* so smbd will bind only to specifically permitted interfaces. It is also possible to set specific share- or resource-based exclusions, for example, on the *[IPC$]* autoshare. The *[IPC$]* share is used for browsing purposes as well as to establish TCP/IP connections.

Another method by which Samba may be secured is by setting Access Control Entries (ACEs) in an Access Control List (ACL) on the shares themselves. This is discussed in Chapter 16, "File, Directory, and Share Access Controls".

18.3 Technical Discussion of Protective Measures and Issues

The key challenge of security is that protective measures suffice at best only to close the door on known exploits and breach techniques. Never assume that because you have followed these few measures, the Samba server is now an impenetrable fortress! Given the history of information systems so far, it is only a matter of time before someone will find yet another vulnerability.

18.3.1 Using Host-Based Protection

In many installations of Samba, the greatest threat comes from outside your immediate network. By default, Samba accepts connections from any host, which means that if you run an insecure version of Samba on a host that is directly connected to the Internet, you can be especially vulnerable.

One of the simplest fixes in this case is to use the *hosts allow* and *hosts deny* options in the Samba `smb.conf` configuration file to allow access to your server only from a specific range of hosts. An example might be:

```
hosts allow = 127.0.0.1 192.168.2.0/24 ↩
    192.168.3.0/24
```

```
        hosts deny = 0.0.0.0/0
```

The above will allow SMB connections only from `localhost` (your own computer) and from the two private networks 192.168.2 and 192.168.3. All other connections will be refused as soon as the client sends its first packet. The refusal will be marked as `not listening on called name` error.

18.3.2 User-Based Protection

If you want to restrict access to your server to valid users only, then the following method may be of use. In the `smb.conf` *[global]* section put:

```
        valid users = @smbusers, jacko
```

This restricts all server access either to the user *jacko* or to members of the system group *smbusers*.

18.3.3 Using Interface Protection

By default, Samba accepts connections on any network interface that it finds on your system. That means if you have an ISDN line or a PPP connection to the Internet then Samba will accept connections on those links. This may not be what you want.

You can change this behavior using options like this:

```
        interfaces = eth* lo
        bind interfaces only = yes
```

This tells Samba to listen for connections only on interfaces with a name starting with `eth` such as `eth0` or `eth1`, plus on the loopback interface called `lo`. The name you will need to use depends on what OS you are using. In the above, I used the common name for Ethernet adapters on Linux.

If you use the above and someone tries to make an SMB connection to your host over a PPP interface called `ppp0`, then [s]he will get a TCP connection refused reply. In that case, no Samba code is run at all, because the operating system has been told not to pass connections from that interface to any Samba process. However, the refusal helps a would-be cracker by confirming that the IP address provides valid active services.

A better response would be to ignore the connection (from, for example, ppp0) altogether. The advantage of ignoring the connection attempt, as compared with refusing it, is that it foils those who probe an interface with the sole intention of finding valid IP addresses for later use in exploitation or denial of service attacks. This method of dealing with potential malicious activity demands the use of appropriate firewall mechanisms.

18.3.4 Using a Firewall

Many people use a firewall to deny access to services they do not want exposed outside their network. This can be a good idea, although I recommend using it in conjunction with the above methods so you are protected even if your firewall is not active for some reason.

If you are setting up a firewall, you need to know what TCP and UDP ports to allow and block. Samba uses the following:

Port 135/TCP - used by smbd
Port 137/UDP - used by nmbd
Port 138/UDP - used by nmbd
Port 139/TCP - used by smbd
Port 445/TCP - used by smbd

The last one is important because many older firewall setups may not be aware of it, given that this port was only added to the protocol in recent years.

When configuring a firewall, the high order ports (1024-65535) are often used for outgoing connections and therefore should be permitted through the firewall. It is prudent to block incoming packets on the high order ports except for established connections.

18.3.5 Using IPC$ Share-Based Denials

If the above methods are not suitable, then you could also place a more specific deny on the IPC$ share that is used in the recently discovered security hole. This allows you to offer access to other shares while denying access to IPC$ from potentially untrustworthy hosts.

To do this you could use:

```
[IPC$]
```

```
        hosts allow = 192.168.115.0/24 127.0.0.1
        hosts deny = 0.0.0.0/0
```

This instructs Samba that IPC$ connections are not allowed from anywhere except the two listed network addresses (localhost and the 192.168.115 subnet). Connections to other shares are still allowed. Because the IPC$ share is the only share that is always accessible anonymously, this provides some level of protection against attackers who do not know a valid username/-password for your host.

If you use this method, then clients will be given an `'access denied'` reply when they try to access the IPC$ share. Those clients will not be able to browse shares and may also be unable to access some other resources. This is not recommended unless for some reason you cannot use one of the other methods just discussed.

18.3.6 NTLMv2 Security

To configure NTLMv2 authentication, the following registry keys are worth knowing about:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa]
"lmcompatibilitylevel"=dword:00000003
```

The value 0x00000003 means to send NTLMv2 response only. Clients will use NTLMv2 authentication; use NTLMv2 session security if the server supports it. Domain controllers accept LM, NTLM, and NTLMv2 authentication.

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0]
"NtlmMinClientSec"=dword:00080000
```

The value 0x00080000 means permit only NTLMv2 session security. If either NtlmMinClientSec or NtlmMinServerSec is set to 0x00080000, the connection will fail if NTLMv2 session security is negotiated.

18.4 Upgrading Samba

Please check regularly on <http://www.samba.org/> for updates and important announcements. Occasionally security releases are made, and it is highly recommended to upgrade Samba promptly when a security vulnerability is discovered. Check with your OS vendor for OS-specific upgrades.

18.5 Common Errors

If all Samba and host platform configurations were really as intuitive as one might like them to be, this chapter would not be necessary. Security issues are often vexing for a support person to resolve, not because of the complexity of the problem, but because most administrators who post what turns out to be a security problem request are totally convinced that the problem is with Samba.

18.5.1 Smbclient Works on Localhost, but the Network Is Dead

This is a common problem. Linux vendors tend to install a default firewall. With the default firewall in place, only traffic on the loopback adapter (IP address 127.0.0.1) is allowed through the firewall.

The solution is either to remove the firewall (stop it) or modify the firewall script to allow SMB networking traffic through. See Section 18.3.4 section.

18.5.2 Why Can Users Access Other Users' Home Directories?

" We are unable to keep individual users from mapping to any other user's home directory once they have supplied a valid password! They only need to enter their own password. I have not found any method to configure Samba so that users may map only their own home directory."

"User xyzzy can map his home directory. Once mapped, user xyzzy can also map anyone else's home directory."

This is not a security flaw, it is by design. Samba allows users to have exactly the same access to the UNIX file system as when they were logged on to the UNIX box, except that it only allows such views onto the file system as are allowed by the defined shares.

If your UNIX home directories are set up so that one user can happily **cd** into another user's directory and execute **ls**, the UNIX security solution is to change file permissions on the user's home directories so that the **cd** and **ls** are denied.

Samba tries very hard not to second guess the UNIX administrator's security policies and trusts the UNIX admin to set the policies and permissions he or she desires.

Samba allows the behavior you require. Simply put the *only user* = %S option in the *[homes]* share definition.

The *only user* works in conjunction with the *users* = list, so to get the behavior you require, add the line:

```
        users = %S
```

This is equivalent to adding

```
        valid users = %S
```

to the definition of the *[homes]* share, as recommended in the `smb.conf` man page.

Chapter 19

INTERDOMAIN TRUST RELATIONSHIPS

Samba-3 supports NT4-style domain trust relationships. This is a feature that many sites will want to use if they migrate to Samba-3 from an NT4-style domain and do not want to adopt Active Directory or an LDAP-based authentication backend. This chapter explains some background information regarding trust relationships and how to create them. It is now possible for Samba-3 to trust NT4 (and vice versa), as well as to create Samba-to-Samba trusts.

The use of interdomain trusts requires use of **winbind**, so the **winbindd** daemon must be running. Winbind operation in this mode is dependent on the specification of a valid UID range and a valid GID range in the `smb.conf` file. These are specified respectively using:

```
        idmap uid = 10000-20000
        idmap gid = 10000-20000
```

The range of values specified must not overlap values used by the host operating system and must not overlap values used in the passdb backend for POSIX user accounts. The maximum value is limited by the uppermost value permitted by the host operating system. This is a UNIX kernel limited parameter. Linux kernel 2.6-based systems support a maximum value of 4294967295 (32-bit unsigned variable).

NOTE

The use of winbind is necessary only when Samba is the trusting domain, not when it is the trusted domain.

19.1 Features and Benefits

Samba-3 can participate in Samba-to-Samba as well as in Samba-to-MS Windows NT4-style trust relationships. This imparts to Samba scalability similar to that with MS Windows NT4.

Given that Samba-3 can function with a scalable backend authentication database such as LDAP, and given its ability to run in primary as well as backup domain control modes, the administrator would be well-advised to consider alternatives to the use of interdomain trusts simply because, by the very nature of how trusts function, this system is fragile. That was, after all, a key reason for the development and adoption of Microsoft Active Directory.

19.2 Trust Relationship Background

MS Windows NT3/4-type security domains employ a nonhierarchical security structure. The limitations of this architecture as it effects the scalability of MS Windows networking in large organizations is well known. Additionally, the flat namespace that results from this design significantly impacts the delegation of administrative responsibilities in large and diverse organizations.

Microsoft developed Active Directory Service (ADS), based on Kerberos and LDAP, as a means of circumventing the limitations of the older technologies. Not every organization is ready or willing to embrace ADS. For small companies the older NT4-style domain security paradigm is quite adequate, and so there remains an entrenched user base for whom there is no direct desire to go through a disruptive change to adopt ADS.

With Windows NT, Microsoft introduced the ability to allow different security domains to effect a mechanism so users from one domain may be given

access rights and privileges in another domain. The language that describes this capability is couched in terms of *trusts*. Specifically, one domain will *trust* the users from another domain. The domain from which users can access another security domain is said to be a trusted domain. The domain in which those users have assigned rights and privileges is the trusting domain. With NT3.x/4.0 all trust relationships are always in one direction only, so if users in both domains are to have privileges and rights in each others' domain, then it is necessary to establish two relationships, one in each direction.

Further, in an NT4-style MS security domain, all trusts are nontransitive. This means that if there are three domains (let's call them red, white, and blue), where red and white have a trust relationship, and white and blue have a trust relationship, then it holds that there is no implied trust between the red and blue domains. Relationships are explicit and not transitive.

New to MS Windows 2000 ADS security contexts is the fact that trust relationships are two-way by default. Also, all inter-ADS domain trusts are transitive. In the case of the red, white, and blue domains, with Windows 2000 and ADS, the red and blue domains can trust each other. This is an inherent feature of ADS domains. Samba-3 implements MS Windows NT4-style interdomain trusts and interoperates with MS Windows 200x ADS security domains in similar manner to MS Windows NT4-style domains.

19.3 Native MS Windows NT4 Trusts Configuration

There are two steps to creating an interdomain trust relationship. To effect a two-way trust relationship, it is necessary for each domain administrator to create a trust account for the other domain to use in verifying security credentials.

19.3.1 Creating an NT4 Domain Trust

For MS Windows NT4, all domain trust relationships are configured using the Domain User Manager. This is done from the Domain User Manager Policies entry on the menu bar. From the **Policy** menu, select **Trust Relationships**. Next to the lower box labeled **Permitted to Trust this Domain** are two buttons, **Add** and **Remove**. The **Add** button will open a panel in which to enter the name of the remote domain that will be able to assign

access rights to users in your domain. You will also need to enter a password for this trust relationship, which the trusting domain will use when authenticating users from the trusted domain. The password needs to be typed twice (for standard confirmation).

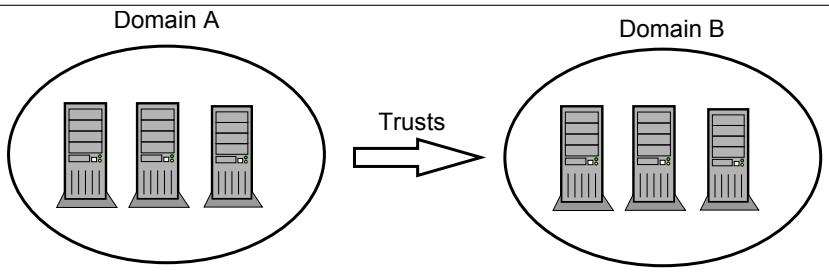
19.3.2 Completing an NT4 Domain Trust

A trust relationship will work only when the other (trusting) domain makes the appropriate connections with the trusted domain. To consummate the trust relationship, the administrator launches the Domain User Manager from the menu selects **Policies**, then select **Trust Relationships**, and clicks on the **Add** button next to the box that is labeled **Trusted Domains**. A panel opens in which must be entered the name of the remote domain as well as the password assigned to that trust.

19.3.3 Interdomain Trust Facilities

A two-way trust relationship is created when two one-way trusts are created, one in each direction. Where a one-way trust has been established between two MS Windows NT4 domains (let's call them DomA and DomB), the following facilities are created:

Figure 19.1 Trusts overview.



- DomA (completes the trust connection) *Trusts* DomB.
- DomA is the *Trusting* domain.
- DomB is the *Trusted* domain (originates the trust account).
- Users in DomB can access resources in DomA.
- Users in DomA cannot access resources in DomB.

- Global groups from DomB can be used in DomA.
- Global groups from DomA cannot be used in DomB.
- DomB does appear in the logon dialog box on client workstations in DomA.
- DomA does not appear in the logon dialog box on client workstations in DomB.
- Users and groups in a trusting domain cannot be granted rights, permissions, or access to a trusted domain.
- The trusting domain can access and use accounts (users/global groups) in the trusted domain.
- Administrators of the trusted domain can be granted administrative rights in the trusting domain.
- Users in a trusted domain can be given rights and privileges in the trusting domain.
- Trusted domain global groups can be given rights and permissions in the trusting domain.
- Global groups from the trusted domain can be made members in local groups on MS Windows domain member machines.

19.4 Configuring Samba NT-Style Domain Trusts

This description is meant to be a fairly short introduction about how to set up a Samba server so that it can participate in interdomain trust relationships. Trust relationship support in Samba is at an early stage, so do not be surprised if something does not function as it should.

Each of the procedures described next assumes the peer domain in the trust relationship is controlled by a Windows NT4 server. However, the remote end could just as well be another Samba-3 domain. It can be clearly seen, after reading this document, that combining Samba-specific parts of what's written in the following sections leads to trust between domains in a purely Samba environment.

19.4.1 Samba as the Trusted Domain

In order to set the Samba PDC to be the trusted party of the relationship, you first need to create a special account for the domain that will be the trusting party. To do that, you can use the **smbpasswd** utility. Creating the trusted domain account is similar to creating a trusted machine account. Suppose, your domain is called SAMBA, and the remote domain is called RUMBA. The first step will be to issue this command from your favorite shell:

```
root#  smbpasswd -a -i rumba
New SMB password: XXXXXXXX
Retype SMB password: XXXXXXXX
Added user rumba$
```

where `-a` means to add a new account into the passdb database and `-i` means to "create this account with the Interdomain trust flag".

The account name will be "rumba$" (the name of the remote domain). If this fails, you should check that the trust account has been added to the system password database (`/etc/passwd`). If it has not been added, you can add it manually and then repeat the previous step.

After issuing this command, you will be asked to enter the password for the account. You can use any password you want, but be aware that Windows NT will not change this password until 7 days following account creation. After the command returns successfully, you can look at the entry for the new account (in the standard way as appropriate for your configuration) and see that the account's name is really RUMBA$ and it has the "I" flag set in the flags field. Now you are ready to confirm the trust by establishing it from Windows NT Server.

Open User Manager for Domains and from the **Policies** menu, select **Trust Relationships...**. Beside the **Trusted domains** list box, click the **Add...** button. You will be prompted for the trusted domain name and the relationship password. Type in SAMBA, as this is the name of the remote domain and the password used at the time of account creation. Click on **OK** and, if everything went without incident, you will see the `Trusted domain relationship successfully established` message.

19.4.2 Samba as the Trusting Domain

This time activities are somewhat reversed. Again, we'll assume that your domain controlled by the Samba PDC is called SAMBA and the NT-controlled domain is called RUMBA.

The very first step is to add an account for the SAMBA domain on RUMBA's PDC.

Launch the Domain User Manager, then from the menu select **Policies**, **Trust Relationships**. Now, next to the **Trusting Domains** box, press the **Add** button and type in the name of the trusted domain (SAMBA) and the password to use in securing the relationship.

The password can be arbitrarily chosen. It is easy to change the password from the Samba server whenever you want. After you confirm the password, your account is ready for use. Now its Samba's turn.

Using your favorite shell while logged in as root, issue this command:

```
root#net rpc trustdom establish rumba
```

You will be prompted for the password you just typed on your Windows NT4 Server box. An error message, `"NT_STATUS_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT,"` that may be reported periodically is of no concern and may safely be ignored. It means the password you gave is correct and the NT4 server says the account is ready for interdomain connection and not for ordinary connection. After that, be patient; it can take a while (especially in large networks), but eventually you should see the `Success` message. Congratulations! Your trust relationship has just been established.

NOTE

You have to run this command as root because you must have write access to the `secrets.tdb` file.

19.5 NT4-Style Domain Trusts with Windows 2000

Although Domain User Manager is not present in Windows 2000, it is also possible to establish an NT4-style trust relationship with a Windows 2000 domain controller running in mixed mode as the trusting server. It should also be possible for Samba to trust a Windows 2000 server; however, more testing is still needed in this area.

After Section 19.4.1 as described previously, open Active Directory Domains and Trusts on the AD controller of the domain whose resources you wish Samba users to have access to. Remember that since NT4-style trusts are not transitive, if you want your users to have access to multiple mixed-mode domains in your AD forest, you will need to repeat this process for each of those domains. With Active Directory domains and trusts open, right-click on the name of the Active Directory domain that will trust our Samba domain and choose **Properties**, then click on the **Trusts** tab. In the upper part of the panel, you will see a list box labeled **Domains trusted by this domain:** and an **Add...** button next to it. Press this button and, just as with NT4, you will be prompted for the trusted domain name and the relationship password. Press *OK* and after a moment, Active Directory will respond with `The trusted domain has been added and the trust has been verified.` Your Samba users can now be granted access to resources in the AD domain.

19.6 Common Errors

Interdomain trust relationships should not be attempted on networks that are unstable or that suffer regular outages. Network stability and integrity are key concerns with distributed trusted domains.

19.6.1 Browsing of Trusted Domain Fails

Browsing from a machine in a trusted Windows 200x domain to a Windows 200x member of a trusting Samba domain, I get the following error:

```
The system detected a possible attempt to compromise security. Please
ensure that you can contact the server that authenticated you.
```

The event logs on the box I'm trying to connect to have entries regarding group policy not being applied because it is a member of a down-level domain.

If there is a computer account in the Windows 200x domain for the machine in question, and it is disabled, this problem can occur. If there is no computer account (removed or never existed), or if that account is still intact (i.e., you just joined it to another domain), everything seems to be fine. By default, when you unjoin a domain (the Windows 200x domain), the computer tries to automatically disable the computer account in the domain. If you are running as an account that has privileges to do this when you unjoin the machine, it is done; otherwise it is not done.

19.6.2 Problems with LDAP ldapsam and Older Versions of smbldap-tools

If you use the **smbldap-useradd** script to create a trust account to set up interdomain trusts, the process of setting up the trust will fail. The account that was created in the LDAP database will have an account flags field that has `[W ]`, when it must have `[I ]` for interdomain trusts to work.

Here is a simple solution. Create a machine account as follows:

```
root#  smbldap-useradd -w domain_name
```

Then set the desired trust account password as shown here:

```
root#  smbldap-passwd domain_name\$
```

Using a text editor, create the following file:

```
dn: uid=domain_name$,ou=People,dc={your-domain},dc={your-top-level-domain}
changetype: modify
sambaAcctFlags: [I         ]
```

Then apply the text file to the LDAP database as follows:

```
root#  ldapmodify -x -h localhost \
```

```
 -D "cn=Manager,dc={your-domain},dc={your-top-level-domain}" \
 -W -f /path-to/foobar
```

Create a single-sided trust under the NT4 Domain User Manager, then execute:

```
root#  net rpc trustdom establish domain_name
```

It works with Samba-3 and NT4 domains, and also with Samba-3 and Windows 200x ADS in mixed mode. Both domain controllers, Samba and NT must have the same WINS server; otherwise, the trust will never work.

Chapter 20

HOSTING A MICROSOFT DISTRIBUTED FILE SYSTEM TREE

20.1 Features and Benefits

The distributed file system (DFS) provides a means of separating the logical view of files and directories that users see from the actual physical locations of these resources on the network. It allows for higher availability, smoother storage expansion, load balancing, and so on.

For information about DFS, refer to the Microsoft documentation[1]. This document explains how to host a DFS tree on a UNIX machine (for DFS-aware clients to browse) using Samba.

A Samba server can be made a DFS server by setting the global Boolean *host msdfs* parameter in the `smb.conf` file. You designate a share as a DFS root using the share-level Boolean *msdfs root* parameter. A DFS root directory on Samba hosts DFS links in the form of symbolic links that point to other servers. For example, a symbolic link `junction->msdfs:storage1\share1` in the share directory acts as the DFS junction. When DFS-aware clients attempt to access the junction link, they are redirected to the storage location (in this case, *storage1**share1*).

DFS trees on Samba work with all DFS-aware clients ranging from Windows 95 to 200x. Example 20.1.1 shows how to setup a DFS tree on a Samba

[1] <http://www.microsoft.com/NTServer/nts/downloads/winfeatures/NTSDistrFile/AdminGuide.asp>

server. In the `/export/dfsroot` directory, you set up your DFS links to other servers on the network.

```
root# cd /export/dfsroot
root# chown root /export/dfsroot
root# chmod 755 /export/dfsroot
root# ln -s msdfs:storageA\\shareA linka
root# ln -s msdfs:serverB\\share,serverC\\share linkb
```

Example 20.1.1 smb.conf with DFS Configured

```
[global]
        netbios name = GANDALF
        host msdfs   = yes
 [dfs]
        path = /export/dfsroot
        msdfs root = yes
```

You should set up the permissions and ownership of the directory acting as the DFS root so that only designated users can create, delete, or modify the msdfs links. Also note that symlink names should be all lowercase. This limitation exists to have Samba avoid trying all the case combinations to get at the link name. Finally, set up the symbolic links to point to the network shares you want and start Samba.

Users on DFS-aware clients can now browse the DFS tree on the Samba server at \\`samba`\`dfs`. Accessing links linka or linkb (which appear as directories to the client) takes users directly to the appropriate shares on the network.

20.2 Common Errors

- Windows clients need to be rebooted if a previously mounted non-DFS share is made a DFS root, or vice versa. A better way is to introduce a new share and make it the DFS root.
- Currently, there's a restriction that msdfs symlink names should all be lowercase.

- For security purposes, the directory acting as the root of the DFS tree should have ownership and permissions set so only designated users can modify the symbolic links in the directory.

20.2.1 MSDFS UNIX Path Is Case-Critical

A network administrator sent advice to the Samba mailing list after long sessions trying to determine why DFS was not working. His advice is worth noting.

"I spent some time trying to figure out why my particular DFS root wasn't working. I noted in the documentation that the symlink should be in all lowercase. It should be amended that the entire path to the symlink should all be in lowercase as well."

"For example, I had a share defined as such:"

```
[pub]
        path = /export/home/Shares/public_share
        msdfs root = yes
```

"and I could not make my Windows 9x/Me (with the dfs client installed) follow this symlink:"

```
damage1 -> msdfs:damage\test-share
```

"Running a debug level of 10 reveals:"

```
[2003/08/20 11:40:33, 5] msdfs/msdfs.c:is_msdfs_link(176)
  is_msdfs_link: /export/home/shares/public_share/* does not exist.
```

"Curious. So I changed the directory name from `.../Shares/...` to `.../shares/...` (along with my service definition) and it worked!"

Chapter 21

CLASSICAL PRINTING SUPPORT

21.1 Features and Benefits

Printing is often a mission-critical service for the users. Samba can provide this service reliably and seamlessly for a client network consisting of Windows workstations.

A Samba print service may be run on a standalone or domain member server, side by side with file serving functions, or on a dedicated print server. It can be made as tightly or as loosely secured as needs dictate. Configurations may be simple or complex. Available authentication schemes are essentially the same as described for file services in previous chapters. Overall, Samba's printing support is now able to replace an NT or Windows 2000 print server full-square, with additional benefits in many cases. Clients may download and install drivers and printers through their familiar `Point'n'Print` mechanism. Printer installations executed by `Logon Scripts` are no problem. Administrators can upload and manage drivers to be used by clients through the familiar `Add Printer Wizard`. As an additional benefit, driver and printer management may be run from the command line or through scripts, making it more efficient in case of large numbers of printers. If a central accounting of print jobs (tracking every single page and supplying the raw data for all sorts of statistical reports) is required, this function is best supported by the newer Common UNIX Printing System (CUPS) as the print subsystem underneath the Samba hood.

This chapter outlines the fundamentals of Samba printing as implemented by the more traditional UNIX BSD- and System V-style printing systems.

Much of the information in this chapter applies also to CUPS. If you use CUPS, you may be tempted to jump to the next chapter, but you will certainly miss a few things if you do. For further information refer to Chapter 22, "CUPS Printing Support".

NOTE

Most of the following examples have been verified on Windows XP Professional clients. Where this document describes the responses to commands given, bear in mind that Windows 200x/XP clients are quite similar but may differ in minor details. Windows NT4 is somewhat different again.

21.2 Technical Introduction

Samba's printing support always relies on the installed print subsystem of the UNIX OS it runs on. Samba is a `middleman`. It takes print files from Windows (or other SMB) clients and passes them to the real printing system for further processing; therefore, it needs to communicate with both sides: the Windows print clients and the UNIX printing system. Hence, we must differentiate between the various client OS types, each of which behave differently, as well as the various UNIX print subsystems, which themselves have different features and are accessed differently.

This chapter deals with the traditional way of UNIX printing. The next chapter covers in great detail the more modern CUPS.

IMPORTANT

CUPS users, be warned: do not just jump on to the next chapter. You might miss important information only found here!

It is apparent from postings on the Samba mailing list that print configuration is one of the most problematic aspects of Samba administration today. Many new Samba administrators have the impression that Samba performs some sort of print processing. Rest assured, Samba does not perform any type of print processing. It does not do any form of print filtering.

Samba obtains from its clients a data stream (print job) that it spools to a local spool area. When the entire print job has been received, Samba invokes a local UNIX/Linux print command and passes the spooled file to it. It is up to the local system printing subsystems to correctly process the print job and to submit it to the printer.

21.2.1 Client to Samba Print Job Processing

Successful printing from a Windows client via a Samba print server to a UNIX printer involves six (potentially seven) stages:

1. Windows opens a connection to the printer share.
2. Samba must authenticate the user.
3. Windows sends a copy of the print file over the network into Samba's spooling area.
4. Windows closes the connection.
5. Samba invokes the print command to hand the file over to the UNIX print subsystem's spooling area.
6. The UNIX print subsystem processes the print job.
7. The print file may need to be explicitly deleted from the Samba spooling area. This item depends on your print spooler configuration settings.

21.2.2 Printing-Related Configuration Parameters

There are a number of configuration parameters to control Samba's printing behavior. Please refer to the man page for `smb.conf` for an overview of these. As with other parameters, there are global-level (tagged with a *G* in the listings) and service-level (*S*) parameters.

Global Parameters These *may not* go into individual share definitions. If they go in by error, the **testparm** utility can discover this (if you run it) and tell you so.

Service-Level Parameters These may be specified in the *[global]* section of `smb.conf`. In this case they define the default behavior of all individual or service-level shares (provided they do not have a different setting defined for the same parameter, thus overriding the global default).

21.3 Simple Print Configuration

Example 21.3.1 shows a simple printing configuration. If you compare this with your own, you may find additional parameters that have been preconfigured by your OS vendor. Following is a discussion and explanation of the parameters. This example does not use many parameters. However, in many environments these are enough to provide a valid `smb.conf` file that enables all clients to print.

Example 21.3.1 Simple Configuration with BSD Printing

```
[global]
        printing = bsd
        load printers = yes
 [printers]
        path = /var/spool/samba
        printable = yes
        public = yes
        writable = no
```

This is only an example configuration. Samba assigns default values to all configuration parameters. The defaults are conservative and sensible. When a parameter is specified in the `smb.conf` file, this overwrites the default value. The **testparm** utility when run as root is capable of reporting all settings, both default as well as `smb.conf` file settings. **Testparm** gives warnings for all misconfigured settings. The complete output is easily 360 lines and more, so you may want to pipe it through a pager program.

The syntax for the configuration file is easy to grasp. You should know that is not very picky about its syntax. As has been explained elsewhere in this

book, Samba tolerates some spelling errors (such as *browseable* instead of *browsable*), and spelling is case-insensitive. It is permissible to use *Yes/No* or *True/False* for Boolean settings. Lists of names may be separated by commas, spaces, or tabs.

21.3.1 Verifying Configuration with testparm

To see all (or at least most) printing-related settings in Samba, including the implicitly used ones, try the command outlined below. This command greps for all occurrences of `lp`, `print`, `spool`, `driver`, `ports`, and `[` in **testparm**'s output. This provides a convenient overview of the running **smbd** print configuration. This command does not show individually created printer shares or the spooling paths they may use. Here is the output of my Samba setup, with settings shown in Example 21.3.1:

```
root# testparm -s -v | egrep "(lp|print|spool|driver|ports|\[)"
 Load smb config files from /etc/samba/smb.conf
 Processing section "[homes]"
 Processing section "[printers]"

 [global]
        smb ports = 139 445
        lpq cache time = 10
        load printers = Yes
        printcap name = /etc/printcap
        disable spoolss = No
        enumports command =
        addprinter command =
        deleteprinter command =
        show add printer wizard = Yes
        os2 driver map =
        printer admin =
        min print space = 0
        max print jobs = 1000
        printable = No
        printing = bsd
        print command = lpr -r -P'%p' %s
        lpq command = lpq -P'%p'
        lprm command = lprm -P'%p' %j
```

```
        lppause command =
        lpresume command =
        printer name =
        use client driver = No

 [homes]

 [printers]
        path = /var/spool/samba
        printable = Yes
```

You can easily verify which settings were implicitly added by Samba's default behavior. *Remember: it may be important in your future dealings with Samba.*

NOTE

The **testparm** in Samba-3 behaves differently from that in 2.2.x: used without the "-v" switch, it only shows you the settings actually written into! To see the complete configuration used, add the "-v" parameter to testparm.

21.3.2 Rapid Configuration Validation

Should you need to troubleshoot at any stage, please always come back to this point first and verify if **testparm** shows the parameters you expect. To give you a warning from personal experience, try to just comment out the *load printers* parameter. If your 2.2.x system behaves like mine, you'll see this:

```
root# grep "load printers" /etc/samba/smb.conf
        #  load printers = Yes
        # This setting is commented out!!
```

```
root# testparm -v /etc/samba/smb.conf | egrep "(load printers)"
        load printers = Yes
```

I assumed that commenting out of this setting should prevent Samba from publishing my printers, but it still did. It took some time to figure out the reason. But I am no longer fooled ... at least not by this.

```
root# grep -A1 "load printers" /etc/samba/smb.conf
        load printers = No
        # The above setting is what I want!
        #  load printers = Yes
        # This setting is commented out!

root# testparm -s -v smb.conf.simpleprinting | egrep "(load printers)"
        load printers = No
```

Only when the parameter is explicitly set to *load printers* = No would Samba conform with my intentions. So, my strong advice is:

- Never rely on commented-out parameters.
- Always set parameters explicitly as you intend them to behave.
- Use **testparm** to uncover hidden settings that might not reflect your intentions.

The following is the most minimal configuration file:

```
root# cat /etc/samba/smb.conf-minimal
        [printers]
```

This example should show that you can use **testparm** to test any Samba configuration file. Actually, we encourage you *not* to change your working system (unless you know exactly what you are doing). Don't rely on the assumption that changes will only take effect after you restart smbd! This is not the case. Samba rereads it every 60 seconds and on each new client connection. You might have to face changes for your production clients that you didn't intend to apply. You will now note a few more interesting things; **testparm** is useful to identify what the Samba print configuration would

be if you used this minimalistic configuration. Here is what you can expect to find:

```
root# testparm -v smb.conf-minimal | egrep "(print|lpq|spool|driver|ports|[)"
 Processing section "[printers]"
 WARNING: [printers] service MUST be printable!
 No path in service printers - using /tmp

        lpq cache time = 10
        load printers = Yes
        printcap name = /etc/printcap
        disable spoolss = No
        enumports command =
        addprinter command =
        deleteprinter command =
        show add printer wizard = Yes
        os2 driver map =
        printer admin =
        min print space = 0
        max print jobs = 1000
        printable = No
        printing = bsd
        print command = lpr -r -P%p %s
        lpq command = lpq -P%p
        printer name =
        use client driver = No

 [printers]
        printable = Yes
```

testparm issued two warnings:

- We did not specify the *[printers]* section as printable.
- We did not tell Samba which spool directory to use.

However, this was not fatal, and Samba will default to values that will work. Please, do not rely on this and do not use this example. This was included to encourage you to be careful to design and specify your setup to do precisely what you require. The outcome on your system may vary

for some parameters given, since Samba may have been built with different compile-time options. *Warning:* do not put a comment sign *at the end* of a valid line. It will cause the parameter to be ignored (just as if you had put the comment sign at the front). At first I regarded this as a bug in my Samba versions. But the man page clearly says: `Internal whitespace in a parameter value is retained verbatim.` This means that a line consisting of, for example,

```
# This defines LPRng as the printing system
        printing = lprng
```

will regard the whole of the string after the `=` sign as the value you want to define. This is an invalid value that will be ignored, and a default value will be used in its place.

21.4 Extended Printing Configuration

Example 21.4.1 shows a more verbose configuration for print-related settings in a BSD-style printing environment. What follows is a discussion and explanation of the various parameters. We chose to use BSD-style printing here because it is still the most commonly used system on legacy UNIX/Linux installations. New installations predominantly use CUPS, which is discussed in a separate chapter. The example explicitly names many parameters that do not need to be specified because they are set by default. You could use a much leaner `smb.conf` file, or you can use **testparm** or **SWAT** to optimize the `smb.conf` file to remove all parameters that are set at default.

This is an example configuration. You may not find all the settings that are in the configuration file that was provided by the OS vendor. Samba configuration parameters, if not explicitly set, default to a sensible value. To see all settings, as `root` use the **testparm** utility. **testparm** gives warnings for misconfigured settings.

21.4.1 Detailed Explanation Settings

The following is a discussion of the settings from Example 21.4.1 Example 21.4.1.

Example 21.4.1 Extended BSD Printing Configuration

```
[global]
        printing = bsd
        load printers = yes
        show add printer wizard = yes
        printcap name = /etc/printcap
        printer admin = @ntadmin, root
        max print jobs = 100
        lpq cache time = 20
        use client driver = no
 [printers]
        comment = All Printers
        printable = yes
        path = /var/spool/samba
        browseable = no
        guest ok = yes
        public = yes
        read only = yes
        writable = no
 [my_printer_name]
        comment = Printer with Restricted Access
        path = /var/spool/samba_my_printer
        printer admin = kurt
        browseable = yes
        printable = yes
        writable = no
        hosts allow = 0.0.0.0
        hosts deny = turbo_xp, 10.160.50.23, 10.160.51.60
        guest ok = no
```

21.4.1.1 The [global] Section

The *[global]* section is one of four special sections (along with *[homes]*, *[printers]*, and *[print$]*). The *[global]* contains all parameters that apply to the server as a whole. It is the place for parameters that have only a global meaning. It may also contain service-level parameters that define default settings for all other sections and shares. This way you can simplify the configuration and avoid setting the same value repeatedly. (Within each individual section or share, you may, however, override these globally set share settings and specify other values).

***printing* = bsd** Causes Samba to use default print commands applicable for the BSD (also known as RFC 1179 style or LPR/LPD) printing system. In general, the *printing* parameter informs Samba about the print subsystem it should expect. Samba supports CUPS, LPD, LPRNG, SYSV, HPUX, AIX, QNX, and PLP. Each of these systems defaults to a different *print command* (and other queue control commands).

> CAUTION
>
> The *printing* parameter is normally a service-level parameter. Since it is included here in the *[global]* section, it will take effect for all printer shares that are not defined differently. Samba-3 no longer supports the SOFTQ printing system.

***load printers* = yes** Tells Samba to create automatically all available printer shares. Available printer shares are discovered by scanning the printcap file. All created printer shares are also loaded for browsing. If you use this parameter, you do not need to specify separate shares for each printer. Each automatically created printer share will clone the configuration options found in the *[printers]* section. (The *load printers = no* setting will allow you to specify each UNIX printer you want to share separately, leaving out some you do not want to be publicly visible and available).

***show add printer wizard* = yes** Setting is normally enabled by default (even if the parameter is not specified in `smb.conf`). It causes the **Add Printer Wizard** icon to appear in the **Printers** folder of the Samba host's share listing (as shown in **Network Neighborhood** or by the **net view** command). To disable it, you need to explicitly set it to `no` (commenting it out will not suffice). The *Add Printer Wizard* lets you upload a printer driver to the *[print$]* share and associate it with a printer (if the respective queue exists before the action), or exchange a printer's driver for any other previously uploaded driver.

***max print jobs* = 100** Sets the upper limit to 100 print jobs being active on the Samba server at any one time. Should a client submit a job that exceeds this number, a "no more space available on server" type of error message will be returned by Samba to the client. A setting of zero (the default) means there is *no* limit at all.

***printcap name* = /etc/printcap** Tells Samba where to look for a list of available printer names. Where CUPS is used, make sure that a printcap file is written. This is controlled by the `Printcap` directive in the `cupsd.conf` file.

***printer admin* = @ntadmin** Members of the ntadmin group should be able to add drivers and set printer properties (`ntadmin` is only an example name; it needs to be a valid UNIX group name); root is implicitly always a *printer admin*. The `@` sign precedes group names in the `/etc/group`. A printer admin can do anything to printers via the remote administration interfaces offered by MS-RPC (see Section 21.5). In larger installations, the *printer admin* parameter is normally a per-share parameter. This permits different groups to administer each printer share.

***lpq cache time* = 20** Controls the cache time for the results of the lpq command. It prevents the lpq command being called too often and reduces the load on a heavily used print server.

***use client driver* = no** If set to `yes`, only takes effect for Windows NT/200x/XP clients (and not for Win 95/98/ME). Its default value is `No` (or `False`). It must *not* be enabled on print shares (with a `yes` or `true` setting) that have valid drivers installed on the Samba server. For more detailed explanations, see the `smb.conf` man page.

21.4.1.2 The [printers] Section

The printers section is the second special section. If a section with this name appears in the `smb.conf`, users are able to connect to any printer specified in the Samba host's printcap file, because Samba on startup then creates a printer share for every printer name it finds in the printcap file. You

could regard this section as a convenient shortcut to share all printers with minimal configuration. It is also a container for settings that should apply as default to all printers. (For more details, see the `smb.conf` man page.) Settings inside this container must be share-level parameters.

***comment* = All printers** The *comment* is shown next to the share if a client queries the server, either via **Network Neighborhood** or with the **net view** command, to list available shares.

***printable* = yes** The *[printers]* service *must* be declared as printable. If you specify otherwise, smbd will refuse to load at startup. This parameter allows connected clients to open, write to, and submit spool files into the directory specified with the *path* parameter for this service. It is used by Samba to differentiate printer shares from file shares.

***path* = /var/spool/samba** Must point to a directory used by Samba to spool incoming print files. *It must not be the same as the spool directory specified in the configuration of your UNIX print subsystem!* The path typically points to a directory that is world writable, with the *sticky* bit set to it.

***browseable* = no** Is always set to `no` if *printable* = yes. It makes the *[printer]* share itself invisible in the list of available shares in a **net view** command or in the Explorer browse list. (You will of course see the individual printers.)

***guest ok* = yes** If this parameter is set to `yes`, no password is required to connect to the printer's service. Access will be granted with the privileges of the *guest account*. On many systems the guest account will map to a user named "nobody." This user will usually be found in the UNIX passwd file with an empty password, but with no valid UNIX login. On some systems the guest account might not have the privilege to be able to print. Test this by logging in as your guest user using **su - guest** and run a system print command like:

```
lpr -P printername /etc/motd
```

***public* = yes** Is a synonym for *guest ok* = yes. Since we have *guest ok* = yes, it really does not need to be here. (This leads to the interesting question, "What if I by accident have two contradictory settings for the same share?" The answer is that the last one encountered by Samba wins. **testparm** does not complain about different settings of the same parameter for the same share. You can test this by setting up multiple lines for the *guest account* parameter with different usernames, and then run testparm to see which one is actually used by Samba.)

***read only* = yes** Normally (for other types of shares) prevents users from creating or modifying files in the service's directory. However, in a *printable* service, it is *always* allowed to write to the directory (if user privileges allow the connection), but only via print spooling operations. Normal write operations are not permitted.

***writable* = no** Is a synonym for *read only* = yes.

21.4.1.3 Any [my_printer_name] Section

If a *[my_printer_name]* section appears in the `smb.conf` file, which includes the parameter *printable* = yes Samba will configure it as a printer share. Windows 9x/Me clients may have problems with connecting or loading printer drivers if the share name has more than eight characters. Do not name a printer share with a name that may conflict with an existing user or file share name. On client connection requests, Samba always tries to find file shares with that name first. If it finds one, it will connect to this and will not connect to a printer with the same name!

***comment* = Printer with Restricted Access** The comment says it all.

***path* = /var/spool/samba_my_printer** Sets the spooling area for this printer to a directory other than the default. It is not necessary to set it differently, but the option is available.

***printer admin* = kurt** The printer admin definition is different for this explicitly defined printer share from the general *[printers]* share. It is not a requirement; we did it to show that it is possible.

***browseable* = yes** This makes the printer browseable so the clients may conveniently find it when browsing the **Network Neighborhood**.

***printable* = yes** See Section 21.4.1.2.

***writable* = no** See Section 21.4.1.2.

***hosts allow* = 10.160.50.,10.160.51.** Here we exercise a certain degree of access control by using the *hosts allow* and *hosts deny* parameters. This is not by any means a safe bet. It is not a way to secure your printers. This line accepts all clients from a certain subnet in a first evaluation of access control.

***hosts deny* = turbo_xp,10.160.50.23,10.160.51.60** All listed hosts are not allowed here (even if they belong to the allowed subnets). As you can see, you could name IP addresses as well as NetBIOS hostnames here.

***guest ok* = no** This printer is not open for the guest account.

21.4.1.4 Print Commands

In each section defining a printer (or in the *[printers]* section), a *print command* parameter may be defined. It sets a command to process the files

that have been placed into the Samba print spool directory for that printer. (That spool directory was, if you remember, set up with the *path* parameter). Typically, this command will submit the spool file to the Samba host's print subsystem, using the suitable system print command. But there is no requirement that this needs to be the case. For debugging or some other reason, you may want to do something completely different than print the file. An example is a command that just copies the print file to a temporary location for further investigation when you need to debug printing. If you craft your own print commands (or even develop print command shell scripts), make sure you pay attention to the need to remove the files from the Samba spool directory. Otherwise, your hard disk may soon suffer from shortage of free space.

21.4.1.5 Default UNIX System Printing Commands

You learned earlier that Samba, in most cases, uses its built-in settings for many parameters if it cannot find an explicitly stated one in its configuration file. The same is true for the *print command*. The default print command varies depending on the *printing* parameter setting. In the commands listed in Table 21.1 , you will notice some parameters of the form *%X* where *X* is *p, s, J*, and so on. These letters stand for printer name, spool file, and job ID, respectively. They are explained in more detail in Table 21.1 presents an overview of key printing options but excludes the special case of CUPS, is discussed in Chapter 22, "CUPS Printing Support".

For *`printing = CUPS`*, if Samba is compiled against libcups, it uses the CUPS API to submit jobs. (It is a good idea also to set *printcap* = cups in case your `cupsd.conf` is set to write its autogenerated printcap file to an unusual place). Otherwise, Samba maps to the System V printing commands with the -oraw option for printing; that is, it uses **lp -c -d%p -oraw; rm %s**. With *`printing = cups`*, and if Samba is compiled against libcups, any manually set print command will be ignored!

21.4.1.6 Custom Print Commands

After a print job has finished spooling to a service, the *print command* will be used by Samba via a system() call to process the spool file. Usually the command specified will submit the spool file to the host's printing subsystem. But there is no requirement at all that this must be the case. The print

Table 21.1 Default Printing Settings

Setting	Default Printing Commands
printing = bsd|aix|lprng|plp	print command is **lpr -r -P%p %s**
printing = sysv|hpux	print command is **lp -c -P%p %s; rm %s**
printing = qnx	print command is **lp -r -P%p -s %s**
printing = bsd|aix|lprng|plp	lpq command is **lpq -P%p**
printing = sysv|hpux	lpq command is **lpstat -o%p**
printing = qnx	lpq command is **lpq -P%p**
printing = bsd|aix|lprng|plp	lprm command is **lprm -P%p %j**
printing = sysv|hpux	lprm command is **cancel %p-%j**
printing = qnx	lprm command is **cancel %p-%j**
printing = bsd|aix|lprng|plp	lppause command is **lp -i %p-%j -H hold**
printing = sysv|hpux	lppause command (...is empty)
printing = qnx	lppause command (...is empty)
printing = bsd|aix|lprng|plp	lpresume command is **lp -i %p-%j -H resume**
printing = sysv|hpux	lpresume command (...is empty)
printing = qnx	lpresume command (...is empty)

subsystem may not remove the spool file on its own, so whatever command you specify, you should ensure that the spool file is deleted after it has been processed.

There is no difficulty with using your own customized print commands with the traditional printing systems. However, if you do not wish to roll your own, you should be well informed about the default built-in commands that Samba uses for each printing subsystem (see Table 21.1). In all the commands listed in the last paragraphs, you see parameters of the form *%X*. These are *macros*, or shortcuts, used as placeholders for the names of real objects. At the time of running a command with such a placeholder, Samba will insert the appropriate value automatically. Print commands can handle all Samba macro substitutions. In regard to printing, the following ones do have special relevance:

- *%s, %f* — the path to the spool file name.
- *%p* — the appropriate printer name.
- *%J* — the job name as transmitted by the client.
- *%c* — the number of printed pages of the spooled job (if known).

- *%z* — the size of the spooled print job (in bytes).

The print command must contain at least one occurrence of *%s* or *%f*. The *%p* is optional. If no printer name is supplied, the *%p* will be silently removed from the print command. In this case, the job is sent to the default printer.

If specified in the *[global]* section, the print command given will be used for any printable service that does not have its own print command specified. If there is neither a specified print command for a printable service nor a global print command, spool files will be created but not processed! Most importantly, print files will not be removed, so they will consume disk space.

Printing may fail on some UNIX systems when using the *nobody* account. If this happens, create an alternative guest account and give it the privilege to print. Set up this guest account in the *[global]* section with the *guest account* parameter.

You can form quite complex print commands. You need to realize that print commands are just passed to a UNIX shell. The shell is able to expand the included environment variables as usual. (The syntax to include a UNIX environment variable *$variable* in the Samba print command is *%$variable*.) To give you a working *print command* example, the following will log a print job to `/tmp/print.log`, print the file, then remove it. The semicolon (";" is the usual separator for commands in shell scripts:

```
        print command = echo Printing %s >> /tmp/print. ↩
            log; lpr -P %p %s; rm %s
```

You may have to vary your own command considerably from this example depending on how you normally print files on your system. The default for the *print command* parameter varies depending on the setting of the *printing* parameter. Another example is:

```
        print command = /usr/local/samba/bin/ ↩
            myprintscript %p %s
```

21.5 Printing Developments Since Samba-2.2

Prior to Samba-2.2.x, print server support for Windows clients was limited to *LanMan* printing calls. This is the same protocol level as Windows 9x/Me PCs offer when they share printers. Beginning with the 2.2.0 release, Samba

started to support the native Windows NT printing mechanisms. These are implemented via *MS-RPC* (Remote Procedure Calls). MS-RPCs use the *SPOOLSS* named pipe for all printing.

The additional functionality provided by the new SPOOLSS support includes:

- Support for downloading printer driver files to Windows 95/98/NT/2000 clients upon demand (*Point'n'Print*).
- Uploading of printer drivers via the Windows NT *Add Printer Wizard* (APW) or the Imprints[1] tool set.
- Support for the native MS-RPC printing calls such as StartDocPrinter, EnumJobs(), and so on. (See the MSDN documentation[2] for more information on the Win32 printing API).
- Support for NT Access Control Lists (ACL) on printer objects.
- Improved support for printer queue manipulation through the use of internal databases for spooled job information (implemented by various `*.tdb` files).

A benefit of updating is that Samba-3 is able to publish its printers to Active Directory (or LDAP).

A fundamental difference exists between MS Windows NT print servers and Samba operation. Windows NT permits the installation of local printers that are not shared. This is an artifact of the fact that any Windows NT machine (server or client) may be used by a user as a workstation. Samba will publish all printers that are made available, either by default or by specific declaration via printer-specific shares.

Windows NT/200x/XP Professional clients do not have to use the standard SMB printer share; they can print directly to any printer on another Windows NT host using MS-RPC. This, of course, assumes that the client has the necessary privileges on the remote host that serves the printer resource. The default permissions assigned by Windows NT to a printer gives the print permissions to the well-known *Everyone* group. (The older clients of type Windows 9x/Me can only print to shared printers.)

[1] <http://imprints.sourceforge.net/>
[2] <http://msdn.microsoft.com/>

21.5.1 Point'n'Print Client Drivers on Samba Servers

There is much confusion about what all this means. The question is often asked, "Is it or is it not necessary for printer drivers to be installed on a Samba host in order to support printing from Windows clients?" The answer to this is no, it is not necessary.

Windows NT/2000 clients can, of course, also run their APW to install drivers *locally* (which then connect to a Samba-served print queue). This is the same method used by Windows 9x/Me clients. (However, a bug existed in Samba 2.2.0 that made Windows NT/2000 clients require that the Samba server possess a valid driver for the printer. This was fixed in Samba 2.2.1).

But it is a new capability to install the printer drivers into the *[print$]* share of the Samba server, and a big convenience, too. Then *all* clients (including 95/98/ME) get the driver installed when they first connect to this printer share. The *uploading* or *depositing* of the driver into this *[print$]* share and the following binding of this driver to an existing Samba printer share can be achieved by different means:

- Running the *APW* on an NT/200x/XP Professional client (this does not work from 95/98/ME clients).
- Using the *Imprints* toolset.
- Using the *smbclient* and *rpcclient* command-line tools.
- Using *cupsaddsmb* (only works for the CUPS printing system, not for LPR/LPD, LPRng, and so on).

Samba does not use these uploaded drivers in any way to process spooled files. These drivers are utilized entirely by the clients who download and install them via the "Point'n'Print" mechanism supported by Samba. The clients use these drivers to generate print files in the format the printer (or the UNIX print system) requires. Print files received by Samba are handed over to the UNIX printing system, which is responsible for all further processing, as needed.

21.5.2 The Obsoleted [printer$] Section

Versions of Samba prior to 2.2 made it possible to use a share named *[printer$]*. This name was taken from the same named service created by Windows 9x/Me clients when a printer was shared by them. Windows

9x/Me printer servers always have a *[printer$]* service that provides read-only access (with no password required) to support printer driver downloads. However, Samba's initial implementation allowed for a parameter named ***printer driver location*** to be used on a per-share basis. This specified the location of the driver files associated with that printer. Another parameter named ***printer driver*** provided a means of defining the printer driver name to be sent to the client.

These parameters, including the ***printer driver file*** parameter, are now removed and cannot be used in installations of Samba-3. The share name *[print$]* is now used for the location of downloadable printer drivers. It is taken from the *[print$]* service created by Windows NT PCs when a printer is shared by them. Windows NT print servers always have a *[print$]* service that provides read-write access (in the context of its ACLs) to support printer driver downloads and uploads. This does not mean Windows 9x/Me clients are now thrown aside. They can use Samba's *[print$]* share support just fine.

21.5.3 Creating the [print$] Share

In order to support the uploading and downloading of printer driver files, you must first configure a file share named *[print$]*. The public name of this share is hard coded in the MS Windows clients. It cannot be renamed, since Windows clients are programmed to search for a service of exactly this name if they want to retrieve printer driver files.

You should modify the server's file to add the global parameters and create the *[print$]* file share (of course, some of the parameter values, such as *path*, are arbitrary and should be replaced with appropriate values for your site). See Example 21.5.1.

Of course, you also need to ensure that the directory named by the *path* parameter exists on the UNIX file system.

21.5.4 [print$] Stanza Parameters

The *[print$]* is a special section in `smb.conf`. It contains settings relevant to potential printer driver download and is used by Windows clients for local print driver installation. The following parameters are frequently needed in this share section:

Example 21.5.1 [print$] Example

```
[global]
 # members of the ntadmin group should be able to add ↩
    drivers and set
 # printer properties. root is implicitly always a ' ↩
    printer admin'.
        printer admin = @ntadmin
 # ...
 [printers]
 # ...
 [print$]
        comment = Printer Driver Download Area
        path = /etc/samba/drivers
        browseable = yes
        guest ok = yes
        read only = yes
        write list = @ntadmin, root
```

***comment* = Printer Driver Download Area** The comment appears next to the share name if it is listed in a share list (usually Windows clients will not see it, but it will also appear up in a **smbclient -L sambaserver** output).

***path* = /etc/samba/printers** The path to the location of the Windows driver file deposit from the UNIX point of view.

***browseable* = no** Makes the *[print$]* share invisible to clients from the **Network Neighborhood**. By excuting from a **cmd** shell:

```
C:\>  net use g:\\sambaserver\print$
```

you can still mount it from any client. This can also be done from the **Connect network drive menu**> from Windows Explorer.

***guest ok* = yes** Gives read-only access to this share for all guest users.

Access may be granted to download and install printer drivers on clients. The requirement for *guest ok = yes* depends on how your site is configured. If users will be guaranteed to have an account on the Samba host, then this is a non-issue.

> **Note**
>
> If all your Windows NT users are guaranteed to be authenticated by the Samba server (for example, if Samba authenticates via an NT domain server and the user has already been validated by the domain controller in order to log on to the Windows NT session), then guest access is not necessary. Of course, in a workgroup environment where you just want to print without worrying about silly accounts and security, then configure the share for guest access. You should consider adding *map to guest* = Bad User in the *[global]* section as well. Make sure you understand what this parameter does before using it.

***read only* = yes** Because we do not want everybody to upload driver files (or even change driver settings), we tagged this share as not writable.

***write list* = @ntadmin, root** The *[print$]* was made read-only by the previous setting so we should create a *write list* entry also. UNIX groups are denoted with a leading "@" character. Users listed here are allowed write-access (as an exception to the general public's read-only access), which they need to update files on the share. Normally, you will want to name only administrative-level user account in this setting. Check the file system permissions to make sure these accounts can copy files to the share. If this is a non-root account, then the account should also be mentioned in the global *printer admin* parameter. See the `smb.conf` man page for more information on configuring file shares.

21.5.5 The [print$] Share Directory

In order for a Windows NT print server to support the downloading of driver files by multiple client architectures, you must create several subdirectories within the *[print$]* service (i.e., the UNIX directory named by the *path* parameter). These correspond to each of the supported client architectures. Samba follows this model as well. Just like the name of the *[print$]* share itself, the subdirectories must be exactly the names listed below (you may leave out the subdirectories of architectures you do not need to support).

Therefore, create a directory tree below the *[print$]* share for each architecture you wish to support like this:

```
[print$]--+
          |--W32X86           # serves drivers to Windows NT x86
          |--WIN40            # serves drivers to Windows 95/98
          |--W32ALPHA         # serves drivers to Windows NT Alpha_AXP
          |--W32MIPS          # serves drivers to Windows NT R4000
          |--W32PPC           # serves drivers to Windows NT PowerPC
```

Required Permissions

In order to add a new driver to your Samba host, one of two conditions must hold true:

- The account used to connect to the Samba host must have a UID of 0 (i.e., a root account).
- The account used to connect to the Samba host must be named in the *printer admin* list.

Of course, the connected account must still have write access to add files to the subdirectories beneath *[print$]*. Remember that all file shares are set to "read-only" by default.

Once you have created the required *[print$]* service and associated subdirectories, go to a Windows NT 4.0/200x/XP client workstation. Open

Network Neighborhood or **My Network Places** and browse for the Samba host. Once you have located the server, navigate to its **Printers and Faxes** folder. You should see an initial listing of printers that matches the printer shares defined on your Samba host.

21.6 Installing Drivers into [print$]

Have you successfully created the *[print$]* share in `smb.conf`, and have you forced Samba to reread its `smb.conf` file? Good. But you are not yet ready to use the new facility. The client driver files need to be installed into this share. So far, it is still an empty share. Unfortunately, it is not enough to just copy the driver files over. They need to be correctly installed so that appropriate records for each driver will exist in the Samba internal databases so it can provide the correct drivers as they are requested from MS Windows clients. And that is a bit tricky, to say the least. We now discuss two alternative ways to install the drivers into *[print$]*:

- Using the Samba command-line utility **rpcclient** with its various subcommands (here, **adddriver** and **setdriver**) from any UNIX workstation.
- Running a GUI (**Printer Properties** and **Add Printer Wizard**) from any Windows NT/200x/XP client workstation.

The latter option is probably the easier one (even if the process may seem a little bit weird at first).

21.6.1 Add Printer Wizard Driver Installation

The printers initially listed in the Samba host's **Printers** folder accessed from a client's Explorer will have no real printer driver assigned to them. By default this driver name is set to a null string. This must be changed now. The local **Add Printer Wizard** (APW), run from NT/2000/XP clients, will help us in this task.

Installation of a valid printer driver is not straightforward. You must attempt to view the printer properties for the printer to which you want the driver assigned. Open Windows Explorer, open **Network Neighborhood**, browse to the Samba host, open Samba's **Printers** folder, right-click on the printer icon, and select **Properties...**. You are now trying to view printer

and driver properties for a queue that has this default `NULL` driver assigned. This will result in the following error message: "Device settings cannot be displayed. The driver for the specified printer is not installed, only spooler properties will be displayed. Do you want to install the driver now?"

Do *not* click on **Yes**! Instead, click on **No** in the error dialog. Now you will be presented with the printer properties window. From here, the way to assign a driver to a printer is open. You now have the choice of:

- Select a driver from the pop-up list of installed drivers. Initially this list will be empty.
- Click on **New Driver** to install a new printer driver (which will start up the APW).

Once the APW is started, the procedure is exactly the same as the one you are familiar with in Windows (we assume here that you are familiar with the printer driver installations procedure on Windows NT). Make sure your connection is, in fact, set up as a user with *printer admin* privileges (if in doubt, use **smbstatus** to check for this). If you wish to install printer drivers for client operating systems other than Windows NT x86, you will need to use the **Sharing** tab of the printer properties dialog.

Assuming you have connected with an administrative (or root) account (as named by the *printer admin* parameter), you will also be able to modify other printer properties such as ACLs and default device settings using this dialog. For the default device settings, please consider the advice given further in Section 21.6.2.

21.6.2 Installing Print Drivers Using rpcclient

The second way to install printer drivers into *[print$]* and set them up in a valid way is to do it from the UNIX command line. This involves four distinct steps:

1. Gather information about required driver files and collect the files.
2. Deposit the driver files into the *[print$]* share's correct subdirectories (possibly by using **smbclient**).
3. Run the **rpcclient** command-line utility once with the **adddriver** subcommand.
4. Run **rpcclient** a second time with the **setdriver** subcommand.

We provide detailed hints for each of these steps in the paragraphs that follow.

21.6.2.1 Identifying Driver Files

To find out about the driver files, you have two options. You can check the contents of the driver CDROM that came with your printer. Study the `*.inf` files located on the CD-ROM. This may not be possible, since the `*.inf` file might be missing. Unfortunately, vendors have now started to use their own installation programs. These installations packages are often in some Windows platform archive format. Additionally, the files may be re-named during the installation process. This makes it extremely difficult to identify the driver files required.

Then you have the second option. Install the driver locally on a Windows client and investigate which filenames and paths it uses after they are installed. (You need to repeat this procedure for every client platform you want to support. We show it here for the W32X86 platform only, a name used by Microsoft for all Windows NT/200x/XP clients.)

A good method to recognize the driver files is to print the test page from the driver's **Properties** dialog (**General** tab). Then look at the list of driver files named on the printout. You'll need to recognize what Windows (and Samba) are calling the **Driver File**, **Data File**, **Config File**, **Help File**, and (optionally) **Dependent Driver Files** (this may vary slightly for Windows NT). You need to note all filenames for the next steps.

Another method to quickly test the driver filenames and related paths is provided by the **rpcclient** utility. Run it with **enumdrivers** or with the **getdriver** subcommand, each at the `3` info level. In the following example, *TURBO_XP* is the name of the Windows PC (in this case it was a Windows XP Professional laptop). I installed the driver locally to TURBO_XP from a Samba server called `KDE-BITSHOP`. We could run an interactive **rpcclient** session; then we would get an **rpcclient />** prompt and would type the subcommands at this prompt. This is left as a good exercise for you. For now, we use **rpcclient** with the `-c` parameter to execute a single subcommand line and exit again. This is the method you use if you want to create scripts to automate the procedure for a large number of printers and drivers. Note the different quotation marks used to overcome the different spaces between words:

```
root# rpcclient -U'Danka%xxxx' -c \
   'getdriver "Heidelberg Digimaster 9110 (PS)" 3' TURBO_XP
cmd = getdriver "Heidelberg Digimaster 9110 (PS)" 3

[Windows NT x86]
Printer Driver Info 3:
  Version: [2]
  Driver Name: [Heidelberg Digimaster 9110 (PS)]
  Architecture: [Windows NT x86]
  Driver Path: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\HDNIS01_de.DLL]
  Datafile: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\Hddm91c1_de.ppd]
  Configfile: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\HDNIS01U_de.DLL]
  Helpfile: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\HDNIS01U_de.HLP]

  Dependentfiles: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\Hddm91c1_de.DLL]
  Dependentfiles: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\Hddm91c1_de.INI]
  Dependentfiles: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\Hddm91c1_de.dat]
  Dependentfiles: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\Hddm91c1_de.cat]
  Dependentfiles: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\Hddm91c1_de.def]
  Dependentfiles: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\Hddm91c1_de.hre]
  Dependentfiles: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\Hddm91c1_de.vnd]
  Dependentfiles: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\Hddm91c1_de.hlp]
  Dependentfiles: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\HDNIS01Aux.dll]
  Dependentfiles: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\HDNIS01_de.NTF]

  Monitorname: []
  Defaultdatatype: []
```

You may notice that this driver has quite a large number of **Dependent files** (there are worse cases, however). Also, strangely, the **Driver File** is tagged here **Driver Path**. We do not yet have support for the so-called WIN40 architecture installed. This name is used by Microsoft for the Windows 9x/Me platforms. If we want to support these, we need to install the Windows 9x/Me driver files in addition to those for W32X86 (i.e., the Windows NT 2000/XP clients) onto a Windows PC. This PC can also host the Windows 9x/Me drivers, even if it runs on Windows NT, 2000, or XP.

Since the *[print$]* share is usually accessible through the **Network Neighborhood**, you can also use the UNC notation from Windows Explorer to poke

at it. The Windows 9x/Me driver files will end up in subdirectory `0` of the `WIN40` directory. The full path to access them is `\\WINDOWSHOST\print$\WIN40\0\`.

> **Note**
>
> More recent drivers on Windows 2000 and Windows XP are installed into the "3" subdirectory instead of the "2". The version 2 of drivers, as used in Windows NT, were running in kernel mode. Windows 2000 changed this. While it still can use the kernel mode drivers (if this is enabled by the Admin), its native mode for printer drivers is user mode execution. This requires drivers designed for this purpose. These types of drivers install into the "3" subdirectory.

21.6.2.2 Obtaining Driver Files from Windows Client [print$] Shares

Now we need to collect all the driver files we identified in our previous step. Where do we get them from? Well, why not retrieve them from the very PC and the same *[print$]* share that we investigated in our last step to identify the files? We can use **smbclient** to do this. We will use the paths and names that were leaked to us by **getdriver**. The listing is edited to include line breaks for readability:

```
root# smbclient //TURBO_XP/print\$ -U'Danka%xxxx' \
   -c 'cd W32X86/2;mget HD*_de.* hd*ppd Hd*_de.* Hddm*dll HDN*Aux.DLL'

added interface ip=10.160.51.60 bcast=10.160.51.255 nmask=255.255.252.0
Got a positive name query response from 10.160.50.8 ( 10.160.50.8 )
Domain=[DEVELOPMENT] OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
Get file Hddm91c1_de.ABD? n
Get file Hddm91c1_de.def? y
getting file \W32X86\2\Hddm91c1_de.def of size 428 as Hddm91c1_de.def
Get file Hddm91c1_de.DLL? y
getting file \W32X86\2\Hddm91c1_de.DLL of size 876544 as Hddm91c1_de.DLL
```

```
[...]
```

After this command is complete, the files are in our current local directory. You probably have noticed that this time we passed several commands to the `-c` parameter, separated by semicolons. This ensures that all commands are executed in sequence on the remote Windows server before **smbclient** exits again.

Remember to repeat the procedure for the WIN40 architecture should you need to support Windows 9x/Me/XP clients. Remember too, the files for these architectures are in the `WIN40/0/` subdirectory. Once this is complete, we can run **smbclient. . .put** to store the collected files on the Samba server's *[print$]* share.

21.6.2.3 Installing Driver Files into [print$]

We are now going to locate the driver files into the *[print$]* share. Remember, the UNIX path to this share has been defined previously in your `smb.conf` file. You also have created subdirectories for the different Windows client types you want to support. If, for example, your *[print$]* share maps to the UNIX path `/etc/samba/drivers/`, your driver files should now go here:

- For all Windows NT, 2000, and XP clients, `/etc/samba/drivers/W32X86/` but not (yet) into the `2` subdirectory.
- For all Windows 95, 98, and Me clients, `/etc/samba/drivers/WIN40/` but not (yet) into the `0` subdirectory.

We again use smbclient to transfer the driver files across the network. We specify the same files and paths as were leaked to us by running **getdriver** against the original *Windows* install. However, now we are going to store the files into a *Samba/UNIX* print server's *[print$]* share.

```
root# smbclient //SAMBA-CUPS/print\$ -U'root%xxxx' -c \
  'cd W32X86; put HDNIS01_de.DLL; \
  put Hddm91c1_de.ppd; put HDNIS01U_de.DLL;        \
  put HDNIS01U_de.HLP; put Hddm91c1_de.DLL;        \
  put Hddm91c1_de.INI; put Hddm91c1KMMin.DLL;      \
  put Hddm91c1_de.dat; put Hddm91c1_de.dat;        \
```

```
  put Hddm91c1_de.def; put Hddm91c1_de.hre;        \
  put Hddm91c1_de.vnd; put Hddm91c1_de.hlp;        \
  put Hddm91c1_de_reg.HLP; put HDNIS01Aux.dll;     \
  put HDNIS01_de.NTF'

added interface ip=10.160.51.60 bcast=10.160.51.255 nmask=255.255.252.0
Got a positive name query response from 10.160.51.162 ( 10.160.51.162 )
Domain=[CUPS-PRINT] OS=[UNIX] Server=[Samba 2.2.7a]
putting file HDNIS01_de.DLL as \W32X86\HDNIS01_de.DLL
putting file Hddm91c1_de.ppd as \W32X86\Hddm91c1_de.ppd
putting file HDNIS01U_de.DLL as \W32X86\HDNIS01U_de.DLL
putting file HDNIS01U_de.HLP as \W32X86\HDNIS01U_de.HLP
putting file Hddm91c1_de.DLL as \W32X86\Hddm91c1_de.DLL
putting file Hddm91c1_de.INI as \W32X86\Hddm91c1_de.INI
putting file Hddm91c1KMMin.DLL as \W32X86\Hddm91c1KMMin.DLL
putting file Hddm91c1_de.dat as \W32X86\Hddm91c1_de.dat
putting file Hddm91c1_de.dat as \W32X86\Hddm91c1_de.dat
putting file Hddm91c1_de.def as \W32X86\Hddm91c1_de.def
putting file Hddm91c1_de.hre as \W32X86\Hddm91c1_de.hre
putting file Hddm91c1_de.vnd as \W32X86\Hddm91c1_de.vnd
putting file Hddm91c1_de.hlp as \W32X86\Hddm91c1_de.hlp
putting file Hddm91c1_de_reg.HLP as \W32X86\Hddm91c1_de_reg.HLP
putting file HDNIS01Aux.dll as \W32X86\HDNIS01Aux.dll
putting file HDNIS01_de.NTF as \W32X86\HDNIS01_de.NTF
```

Whew — that was a lot of typing! Most drivers are a lot smaller — many have only three generic PostScript driver files plus one PPD. While we did retrieve the files from the `2` subdirectory of the `W32X86` directory from the Windows box, we do not put them (for now) in this same subdirectory of the Samba box. This relocation will automatically be done by the **adddriver** command, which we will run shortly (and do not forget to also put the files for the Windows 9x/Me architecture into the `WIN40/` subdirectory should you need them).

21.6.2.4 smbclient to Confirm Driver Installation

For now we verify that our files are there. This can be done with **smbclient**, too (but, of course, you can log in via SSH also and do this through a standard UNIX shell access):

```
root# smbclient //SAMBA-CUPS/print\$ -U 'root%xxxx' \
   -c 'cd W32X86; pwd; dir; cd 2; pwd; dir'
 added interface ip=10.160.51.60 bcast=10.160.51.255 nmask=255.255.252.0
Got a positive name query response from 10.160.51.162 ( 10.160.51.162 )
Domain=[CUPS-PRINT] OS=[UNIX] Server=[Samba 2.2.8a]

Current directory is \\SAMBA-CUPS\print$\W32X86\
.                                   D        0  Sun May  4 03:56:35 2003
..                                  D        0  Thu Apr 10 23:47:40 2003
2                                    D        0  Sun May  4 03:56:18 2003
HDNIS01Aux.dll                       A    15356  Sun May  4 03:58:59 2003
Hddm91c1KMMin.DLL                    A    46966  Sun May  4 03:58:59 2003
HDNIS01_de.DLL                       A   434400  Sun May  4 03:58:59 2003
HDNIS01_de.NTF                       A   790404  Sun May  4 03:56:35 2003
Hddm91c1_de.DLL                      A   876544  Sun May  4 03:58:59 2003
Hddm91c1_de.INI                      A      101  Sun May  4 03:58:59 2003
Hddm91c1_de.dat                      A     5044  Sun May  4 03:58:59 2003
Hddm91c1_de.def                      A      428  Sun May  4 03:58:59 2003
Hddm91c1_de.hlp                      A    37699  Sun May  4 03:58:59 2003
Hddm91c1_de.hre                      A   323584  Sun May  4 03:58:59 2003
Hddm91c1_de.ppd                      A    26373  Sun May  4 03:58:59 2003
Hddm91c1_de.vnd                      A    45056  Sun May  4 03:58:59 2003
HDNIS01U_de.DLL                      A   165888  Sun May  4 03:58:59 2003
HDNIS01U_de.HLP                      A    19770  Sun May  4 03:58:59 2003
Hddm91c1_de_reg.HLP                  A   228417  Sun May  4 03:58:59 2003
              40976 blocks of size 262144. 709 blocks available

Current directory is \\SAMBA-CUPS\print$\W32X86\2\
.                                   D        0  Sun May  4 03:56:18 2003
..                                  D        0  Sun May  4 03:56:35 2003
ADOBEPS5.DLL                         A   434400  Sat May  3 23:18:45 2003
laserjet4.ppd                        A     9639  Thu Apr 24 01:05:32 2003
ADOBEPSU.DLL                         A   109568  Sat May  3 23:18:45 2003
ADOBEPSU.HLP                         A    18082  Sat May  3 23:18:45 2003
PDFcreator2.PPD                      A    15746  Sun Apr 20 22:24:07 2003
              40976 blocks of size 262144. 709 blocks available
```

Notice that there are already driver files present in the 2 subdirectory (probably from a previous installation). Once the files for the new driver are there too, you are still a few steps away from being able to use them on the clients. The only thing you could do now is retrieve them from a client just like you retrieve ordinary files from a file share, by opening print$ in Windows Explorer. But that wouldn't install them per Point'n'Print. The reason is that Samba does not yet know that these files are something special, namely *printer driver files*, and it does not know to which print queue(s) these driver files belong.

21.6.2.5 Running rpcclient with adddriver

Next, you must tell Samba about the special category of the files you just uploaded into the *[print$]* share. This is done by the **adddriver** command. It will prompt Samba to register the driver files into its internal TDB database files. The following command and its output has been edited for readability:

```
root# rpcclient -Uroot%xxxx -c 'adddriver "Windows NT x86" \
  "dm9110:HDNIS01_de.DLL: \
  Hddm91c1_de.ppd:HDNIS01U_de.DLL:HDNIS01U_de.HLP:   \
  NULL:RAW:Hddm91c1_de.DLL,Hddm91c1_de.INI,          \
  Hddm91c1_de.dat,Hddm91c1_de.def,Hddm91c1_de.hre,   \
  Hddm91c1_de.vnd,Hddm91c1_de.hlp,Hddm91c1KMMin.DLL, \
  HDNIS01Aux.dll,HDNIS01_de.NTF,                     \
  Hddm91c1_de_reg.HLP' SAMBA-CUPS

cmd = adddriver "Windows NT x86" \
  "dm9110:HDNIS01_de.DLL:Hddm91c1_de.ppd:HDNIS01U_de.DLL:   \
  HDNIS01U_de.HLP:NULL:RAW:Hddm91c1_de.DLL,Hddm91c1_de.INI, \
  Hddm91c1_de.dat,Hddm91c1_de.def,Hddm91c1_de.hre,          \
  Hddm91c1_de.vnd,Hddm91c1_de.hlp,Hddm91c1KMMin.DLL,        \
  HDNIS01Aux.dll,HDNIS01_de.NTF,Hddm91c1_de_reg.HLP"

Printer Driver dm9110 successfully installed.
```

After this step, the driver should be recognized by Samba on the print server. You need to be very careful when typing the command. Don't exchange the order of the fields. Some changes would lead to an `NT_STATUS_UNSUCCESSFUL`

error message. These become obvious. Other changes might install the driver files successfully but render the driver unworkable. So take care! Hints about the syntax of the adddriver command are in the man page. provides a more detailed description, should you need it.

21.6.2.6 Checking adddriver Completion

One indication for Samba's recognition of the files as driver files is the `successfully installed` message. Another one is the fact that our files have been moved by the **adddriver** command into the 2 subdirectory. You can check this again with **smbclient**:

```
root# smbclient //SAMBA-CUPS/print\$ -Uroot%xx \
   -c 'cd W32X86;dir;pwd;cd 2;dir;pwd'
 added interface ip=10.160.51.162 bcast=10.160.51.255 nmask=255.255.252.0
 Domain=[CUPS-PRINT] OS=[UNIX] Server=[Samba 2.2.7a]

  Current directory is \\SAMBA-CUPS\print$\W32X86\
  .                                   D        0  Sun May  4 04:32:48 2003
  ..                                  D        0  Thu Apr 10 23:47:40 2003
  2                                    D        0  Sun May  4 04:32:48 2003
                 40976 blocks of size 262144. 731 blocks available

  Current directory is \\SAMBA-CUPS\print$\W32X86\2\
  .                                   D        0  Sun May  4 04:32:48 2003
  ..                                  D        0  Sun May  4 04:32:48 2003
  DigiMaster.PPD                       A   148336  Thu Apr 24 01:07:00 2003
  ADOBEPS5.DLL                         A   434400  Sat May  3 23:18:45 2003
  laserjet4.ppd                        A     9639  Thu Apr 24 01:05:32 2003
  ADOBEPSU.DLL                         A   109568  Sat May  3 23:18:45 2003
  ADOBEPSU.HLP                         A    18082  Sat May  3 23:18:45 2003
  PDFcreator2.PPD                      A    15746  Sun Apr 20 22:24:07 2003
  HDNIS01Aux.dll                       A    15356  Sun May  4 04:32:18 2003
  Hddm91c1KMMin.DLL                    A    46966  Sun May  4 04:32:18 2003
  HDNIS01_de.DLL                       A   434400  Sun May  4 04:32:18 2003
  HDNIS01_de.NTF                       A   790404  Sun May  4 04:32:18 2003
  Hddm91c1_de.DLL                      A   876544  Sun May  4 04:32:18 2003
  Hddm91c1_de.INI                      A      101  Sun May  4 04:32:18 2003
  Hddm91c1_de.dat                      A     5044  Sun May  4 04:32:18 2003
```

```
  Hddm91c1_de.def                     A      428  Sun May  4 04:32:18 2003
  Hddm91c1_de.hlp                     A    37699  Sun May  4 04:32:18 2003
  Hddm91c1_de.hre                     A   323584  Sun May  4 04:32:18 2003
  Hddm91c1_de.ppd                     A    26373  Sun May  4 04:32:18 2003
  Hddm91c1_de.vnd                     A    45056  Sun May  4 04:32:18 2003
  HDNIS01U_de.DLL                     A   165888  Sun May  4 04:32:18 2003
  HDNIS01U_de.HLP                     A    19770  Sun May  4 04:32:18 2003
  Hddm91c1_de_reg.HLP                 A   228417  Sun May  4 04:32:18 2003
                40976 blocks of size 262144. 731 blocks available
```

Another verification is that the timestamp of the printing TDB files is now updated (and possibly their file size has increased).

21.6.2.7 Check Samba for Driver Recognition

Now the driver should be registered with Samba. We can easily verify this and will do so in a moment. However, this driver is not yet associated with a particular printer. We may check the driver status of the files by at least three methods:

- From any Windows client browse Network Neighborhood, find the Samba host, and open the Samba **Printers and Faxes** folder. Select any printer icon, right-click and select the printer **Properties**. Click the **Advanced** tab. Here is a field indicating the driver for that printer. A drop-down menu allows you to change that driver (be careful not to do this unwittingly). You can use this list to view all drivers known to Samba. Your new one should be among them. (Each type of client will see only its own architecture's list. If you do not have every driver installed for each platform, the list will differ if you look at it from Windows95/98/ME or Windows NT/2000/XP.)
- From a Windows 200x/XP client (not Windows NT) browse **Network Neighborhood**, search for the Samba server, open the server's **Printers** folder, and right-click on the white background (with no printer highlighted). Select **Server Properties**. On the **Drivers** tab you will see the new driver listed. This view enables you to also inspect the list of files belonging to that driver (this does not work on Windows NT, but only on Windows 2000 and Windows XP; Windows NT does not provide the **Drivers** tab). An alternative and much quicker method for Windows 2000/XP to start this dialog is by typing into a DOS box

(you must of course adapt the name to your Samba server instead of *SAMBA-CUPS*):

```
rundll32 printui.dll,PrintUIEntry /s /t2 /n\\SAMBA-CUPS
```

- From a UNIX prompt, run this command (or a variant thereof), where *SAMBA-CUPS* is the name of the Samba host and xxxx represents the actual Samba password assigned to root:

```
rpcclient -U'root%xxxx' -c 'enumdrivers' SAMBA-CUPS
```

You will see a listing of all drivers Samba knows about. Your new one should be among them. But it is only listed under the *[Windows NT x86]* heading, not under *[Windows 4.0]*, since you didn't install that part. Or did you? In our example it is named `dm9110`. Note that the third column shows the other installed drivers twice, one time for each supported architecture. Our new driver only shows up for Windows NT 4.0 or 2000. To have it present for Windows 95, 98, and Me, you'll have to repeat the whole procedure with the WIN40 architecture and subdirectory.

21.6.2.8 Specific Driver Name Flexibility

You can name the driver as you like. If you repeat the **adddriver** step with the same files as before but with a different driver name, it will work the same:

```
root# rpcclient -Uroot%xxxx         \
  -c 'adddriver "Windows NT x86"                      \
  "mydrivername:HDNIS01_de.DLL:              \
  Hddm91c1_de.ppd:HDNIS01U_de.DLL:HDNIS01U_de.HLP:   \
  NULL:RAW:Hddm91c1_de.DLL,Hddm91c1_de.INI,           \
  Hddm91c1_de.dat,Hddm91c1_de.def,Hddm91c1_de.hre,    \
  Hddm91c1_de.vnd,Hddm91c1_de.hlp,Hddm91c1KMMin.DLL, \
  HDNIS01Aux.dll,HDNIS01_de.NTF,Hddm91c1_de_reg.HLP' SAMBA-CUPS
```

```
cmd = adddriver "Windows NT x86" \
 "mydrivername:HDNIS01_de.DLL:Hddm91c1_de.ppd:HDNIS01U_de.DLL:\
  HDNIS01U_de.HLP:NULL:RAW:Hddm91c1_de.DLL,Hddm91c1_de.INI,          \
  Hddm91c1_de.dat,Hddm91c1_de.def,Hddm91c1_de.hre,                   \
  Hddm91c1_de.vnd,Hddm91c1_de.hlp,Hddm91c1KMMin.DLL,                 \
  HDNIS01Aux.dll,HDNIS01_de.NTF,Hddm91c1_de_reg.HLP"

Printer Driver mydrivername successfully installed.
```

You will be able to bind that driver to any print queue (however, you are responsible that you associate drivers to queues that make sense with respect to target printers). You cannot run the **rpcclient adddriver** command repeatedly. Each run consumes the files you had put into the *[print$]* share by moving them into the respective subdirectories, so you must execute an **smbclient ... put** command before each **rpcclient ... adddriver** command.

21.6.2.9 Running rpcclient with setdriver

Samba needs to know which printer owns which driver. Create a mapping of the driver to a printer, and store this information in Samba's memory, the TDB files. The **rpcclient setdriver** command achieves exactly this:

```
root# rpcclient -U'root%xxxx' -c 'setdriver dm9110 mydrivername' SAMBA-CUPS
 cmd = setdriver dm9110 mydrivername

Successfully set dm9110 to driver mydrivername.
```

Ah, no, I did not want to do that. Repeat, this time with the name I intended:

```
root# rpcclient -U'root%xxxx' -c 'setdriver dm9110 dm9110' SAMBA-CUPS
 cmd = setdriver dm9110 dm9110
Successfully set dm9110 to driver dm9110.
```

The syntax of the command is:

```
rpcclient -U'root%sambapassword' -c 'setdriver printername \
 drivername' SAMBA-Hostname.
```

Now we have done most of the work, but not all of it.

> **Note**
>
> The **setdriver** command will only succeed if the printer is already known to Samba. A bug in 2.2.x prevented Samba from recognizing freshly installed printers. You had to restart Samba, or at least send an HUP signal to all running smbd processes to work around this: `kill -HUP ‘pidof smbd‘`.

21.7 Client Driver Installation Procedure

As Don Quixote said, “The proof of the pudding is in the eating.” The proof for our setup lies in the printing. So let’s install the printer driver onto the client PCs. This is not as straightforward as it may seem. Read on.

21.7.1 First Client Driver Installation

Especially important is the installation onto the first client PC (for each architectural platform separately). Once this is done correctly, all further clients are easy to set up and shouldn’t need further attention. What follows is a description for the recommended first procedure. You now work from a client workstation. You should check that your connection is not unwittingly mapped to *bad user* nobody. In a DOS box type:

```
net use \\SAMBA-SERVER\print$ /user:root
```

Replace root, if needed, by another valid *printer admin* user as given in the definition. Should you already be connected as a different user, you will get an error message. There is no easy way to get rid of that connection, because Windows does not seem to know a concept of logging off from a share

connection (do not confuse this with logging off from the local workstation; that is a different matter). On Windows NT/200x, you can force a logoff from all smb/cifs connections by restarting the *workstation* service. You can try to close all Windows file explorers and Internet Explorer for Windows. As a last resort, you may have to reboot. Make sure there is no automatic reconnection set up. It may be easier to go to a different workstation and try from there. After you have made sure you are connected as a printer admin user (you can check this with the **smbstatus** command on Samba), do this from the Windows workstation:

1. Open **Network Neighborhood**.
2. Browse to Samba server.
3. Open its **Printers and Faxes** folder.
4. Highlight and right-click on the printer.
5. Select **Connect** (for Windows NT4/200x it is possibly **Install**).

A new printer (named *`printername`* on Samba server) should now have appeared in your *local* Printer folder (check **Start** -> **Settings** -> **Control Panel** -> **Printers and Faxes**).

Most likely you are tempted to try to print a test page. After all, you now can open the printer properties, and on the **General** tab there is a button offering to do just that. But chances are that you get an error message saying "`Unable to print Test Page`." The reason might be that there is not yet a valid device mode set for the driver or that the "printer driver data" set is still incomplete.

You must make sure that a valid *`device mode`* is set for the driver. We now explain what that means.

21.7.2 Setting Device Modes on New Printers

For a printer to be truly usable by a Windows NT/200x/XP client, it must possess:

- A valid *device mode* generated by the driver for the printer (defining things like paper size, orientation and duplex settings).
- A complete set of *printer driver data* generated by the driver.

If either of these is incomplete, the clients can produce less than optimal output at best. In the worst cases, unreadable garbage or nothing at all comes from the printer, or it produces a harvest of error messages when attempting to print. Samba stores the named values and all printing-related information in its internal TDB database files (`ntprinters.tdb`, `ntdrivers.tdb`, `printing.tdb`, and `ntforms.tdb`).

The device mode and the set of printer driver data are basically collections of settings for all print queue properties, initialized in a sensible way. Device modes and printer driver data should initially be set on the print server (the Samba host) to healthy values so the clients can start to use them immediately. How do we set these initial healthy values? This can be achieved by accessing the drivers remotely from an NT (or 200x/XP) client, as discussed in the following paragraphs.

Be aware that a valid device mode can only be initiated by a *printer admin* or root (the reason should be obvious). Device modes can be correctly set only by executing the printer driver program itself. Since Samba cannot execute this Win32 platform driver code, it sets this field initially to NULL (which is not a valid setting for clients to use). Fortunately, most drivers automatically generate the printer driver data that is needed when they are uploaded to the *[print$]* share with the help of the APW or rpcclient.

The generation and setting of a first valid device mode, however, requires some tickling from a client to set it on the Samba server. The easiest means of doing so is to simply change the page orientation on the server's printer. This executes enough of the printer driver program on the client for the desired effect to happen and feeds back the new device mode to our Samba server. You can use the native Windows NT/200x/XP printer properties page from a Window client for this: Procedure to Initialize the Printer Driver Settings

1. Browse the **Network Neighborhood**.
2. Find the Samba server.
3. Open the Samba server's **Printers and Faxes** folder.
4. Highlight the shared printer in question.
5. Right-click on the printer (you may already be here if you followed the last section's description).

6. At the bottom of the context menu select **Properties** (if the menu still offers the **Connect** entry further above, you need to click on that one first to achieve the driver installation, as shown in the last section).
7. Go to the **Advanced** tab; click on **Printing Defaults**.
8. Change the **Portrait** page setting to **Landscape** (and back).
9. Make sure to apply changes between swapping the page orientation to cause the change to actually take effect.
10. While you are at it, you may also want to set the desired printing defaults here, which then apply to all future client driver installations.

This procedure executes the printer driver program on the client platform and feeds back the correct device mode to Samba, which now stores it in its TDB files. Once the driver is installed on the client, you can follow the analogous steps by accessing the *local* **Printers** folder, too, if you are a Samba printer admin user. From now on, printing should work as expected.

Samba includes a service-level parameter name *`default devmode`* for generating a default device mode for a printer. Some drivers function well with Samba's default set of properties. Others may crash the client's spooler service. So use this parameter with caution. It is always better to have the client generate a valid device mode for the printer and store it on the server for you.

21.7.3 Additional Client Driver Installation

Every additional driver may be installed in the same way as just described. Browse **Network Neighborhood**, open the **Printers** folder on Samba server, right-click on **Printer**, and choose **Connect...**. Once this completes (should be not more than a few seconds, but could also take a minute, depending on network conditions), you should find the new printer in your client workstation local **Printers and Faxes** folder.

You can also open your local **Printers and Faxes** folder by using this command on Windows 200x/XP Professional workstations:

```
rundll32 shell32.dll,SHHelpShortcuts_RunDLL PrintersFolder
```

or this command on Windows NT 4.0 workstations:

```
rundll32 shell32.dll,Control_RunDLL MAIN.CPL @2
```

You can enter the commands either inside a **DOS box** window or in the **Run command...** field from the **Start** menu.

21.7.4 Always Make First Client Connection as root or "printer admin"

After you installed the driver on the Samba server (in its *[print$]* share), you should always make sure that your first client installation completes correctly. Make it a habit for yourself to build the very first connection from a client as *printer admin*. This is to make sure that:

- A first valid *device mode* is really initialized (see above Section 21.7.2) for more explanation details).
- The default print settings of your printer for all further client installations are as you want them.

Do this by changing the orientation to landscape, click on **Apply**, and then change it back again. Next, modify the other settings (for example, you do not want the default media size set to **Letter** when you are all using **A4**, right? You may want to set the printer for **duplex** as the default, and so on).

To connect as root to a Samba printer, try this command from a Windows 200x/XP DOS box command prompt:

```
C:\> runas /netonly /user:root "rundll32 printui.dll,PrintUIEntry /p /t3 /n
   \\SAMBA-SERVER\printername"
```

You will be prompted for `root`'s Samba password; type it, wait a few seconds, click on **Printing Defaults**, and proceed to set the job options that should be used as defaults by all clients. Alternatively, instead of root you can name one other member of the *printer admin* from the setting.

Now all the other users downloading and installing the driver the same way (using `Point'n'Print`) will have the same defaults set for them. If you miss

this step, you'll get a lot of help desk calls from your users, but maybe you like to talk to people.

21.8 Other Gotchas

Your driver is installed. It is now ready for Point'n'Print installation by the clients. You may have tried to download and use it on your first client machine, but wait. Let's make sure you are acquainted first with a few tips and tricks you may find useful. For example, suppose you did not set the defaults on the printer, as advised in the preceding paragraphs. Your users complain about various issues (such as, "We need to set the paper size for each job from Letter to A4 and it will not store it").

21.8.1 Setting Default Print Options for Client Drivers

The last sentence might be viewed with mixed feelings by some users and Admins. They have struggled for hours and could not arrive at a point where their settings seemed to be saved. It is not their fault. The confusing thing is that in the multitabbed dialog that pops up when you right-click on the printer name and select **Properties**, you can arrive at two dialogs that appear identical, each claiming that they help you to set printer options in three different ways. Here is the definitive answer to the Samba default driver setting FAQ:

"I can not set and save default print options for all users on Windows 200x/XP. Why not?". How are you doing it? I bet the wrong way. (It is not easy to find out, though.) There are three different ways to bring you to a dialog that seems to set everything. All three dialogs look the same, but only one of them does what you intend. You need to be Administrator or Print Administrator to do this for all users. Here is how I reproduce it in an XP Professional:

A The first "wrong" way:

1 Open the **Printers** folder.

2 Right-click on the printer (*remoteprinter on cupshost*) and select in context menu **Printing Preferences...**.

3 Look at this dialog closely and remember what it looks like.

B The second "wrong" way: .

1 Open the **Printers** folder.

2 Right-click on the printer (*remoteprinter on cupshost*) and select in the context menu **Properties**

3 Click on the **General** tab.

4 Click on the **Printing Preferences...** button.

5 A new dialog opens. Keep this dialog open and go back to the parent dialog.

C The third and correct way (should you do this from the beginning, just carry out steps 1 and 2 from the second method above):

1 Click on the **Advanced** tab. (If everything is "grayed out," then you are not logged in as a user with enough privileges.)

2 Click on the **Printing Defaults** button.

3 On any of the two new tabs, click on the **Advanced** button.

4 A new dialog opens. Compare this one to the other. Are they identical when you compare one from "B.5" and one from A.3?

Do you see any difference in the two settings dialogs? I do not either. However, only the last one, which you arrived at with steps C.1 through C.6 will permanently save any settings which will then become the defaults for new users. If you want all clients to have the same defaults, you need to conduct these steps as administrator (*printer admin*) before a client downloads the driver (the clients can later set their own per-user defaults by following procedures A or B above). Windows 200x/XP allow per-user default settings and the ones the administrator gives them before they set up their own. The parents of the identical-looking dialogs have a slight difference in their window names; one is called `Default Print Values for Printer Foo on Server Bar` (which is the one you need) and the other is called "`Print Settings for Printer Foo on Server Bar`". The last one is the one you arrive at when you right-click on the printer and select **Print Settings...**. This is the one that you were taught to use back in the days of Windows NT, so it is only natural to try the same way with Windows 200x/XP. You would not dream that there is now a different path to arrive at an identical-looking, but functionally different, dialog to set defaults for all users.

TIP

Try (on Windows 200x/XP) to run this command (as a user with the right privileges):

`rundll32 printui.dll,PrintUIEntry /p /t3 /n\\SAMBA-SERVER\printersharename`

To see the tab with the **Printing Defaults** button (the one you need), also run this command:

`rundll32 printui.dll,PrintUIEntry /p /t0 /n\\SAMBA-SERVER\printersharename`

To see the tab with the **Printing Preferences** button (the one that does not set systemwide defaults), you can start the commands from inside a DOS box or from **Start** -> **Run**.

21.8.2 Supporting Large Numbers of Printers

One issue that has arisen during the recent development phase of Samba is the need to support driver downloads for hundreds of printers. Using Windows NT APW for this task is somewhat awkward (to say the least). If you do not want to acquire RSS pains from the printer installation clicking orgy alone, you need to think about a non-interactive script.

If more than one printer is using the same driver, the **rpcclient setdriver** command can be used to set the driver associated with an installed queue. If the driver is uploaded to *[print$]* once and registered with the printing TDBs, it can be used by multiple print queues. In this case, you just need to repeat the **setprinter** subcommand of **rpcclient** for every queue (without the need to conduct the **adddriver** repeatedly). The following is an example of how this can be accomplished:

```
root# rpcclient SAMBA-CUPS -U root%secret -c 'enumdrivers'
 cmd = enumdrivers
```

```
 [Windows NT x86]
 Printer Driver Info 1:
   Driver Name: [infotec  IS 2075 PCL 6]

 Printer Driver Info 1:
   Driver Name: [DANKA InfoStream]

 Printer Driver Info 1:
   Driver Name: [Heidelberg Digimaster 9110 (PS)]

 Printer Driver Info 1:
   Driver Name: [dm9110]

 Printer Driver Info 1:
   Driver Name: [mydrivername]

 [....]
```

```
root# rpcclient SAMBA-CUPS -U root%secret -c 'enumprinters'
 cmd = enumprinters
   flags:[0x800000]
   name:[\\SAMBA-CUPS\dm9110]
   description:[\\SAMBA-CUPS\dm9110,,110ppm HiVolume DANKA Stuttgart]
   comment:[110 ppm HiVolume DANKA Stuttgart]
 [....]
```

```
root# rpcclient SAMBA-CUPS -U root%secret -c \
  'setdriver dm9110 "Heidelberg Digimaster 9110 (PS)"'
 cmd = setdriver dm9110 Heidelberg Digimaster 9110 (PPD)
 Successfully set dm9110 to driver Heidelberg Digimaster 9110 (PS).
```

```
root# rpcclient SAMBA-CUPS -U root%secret -c 'enumprinters'
 cmd = enumprinters
   flags:[0x800000]
   name:[\\SAMBA-CUPS\dm9110]
```

```
   description:[\\SAMBA-CUPS\dm9110,Heidelberg Digimaster 9110 (PS),\
     110ppm HiVolume DANKA Stuttgart]
   comment:[110ppm HiVolume DANKA Stuttgart]
 [....]
```

```
root# rpcclient SAMBA-CUPS -U root%secret -c 'setdriver dm9110 mydrivername'
 cmd = setdriver dm9110 mydrivername
 Successfully set dm9110 to mydrivername.
```

```
root# rpcclient SAMBA-CUPS -U root%secret -c 'enumprinters'
 cmd = enumprinters
   flags:[0x800000]
   name:[\\SAMBA-CUPS\dm9110]
   description:[\\SAMBA-CUPS\dm9110,mydrivername,\
     110ppm HiVolume DANKA Stuttgart]
   comment:[110ppm HiVolume DANKA Stuttgart]
 [....]
```

It may not be easy to recognize that the first call to **enumprinters** showed the "dm9110" printer with an empty string where the driver should have been listed (between the two commas in the description field). After the **setdriver** command succeeds, all is well.

21.8.3 Adding New Printers with the Windows NT APW

By default, Samba exhibits all printer shares defined in `smb.conf` in the **Printers** folder. Also located in this folder is the Windows NT Add Printer Wizard icon. The APW will be shown only if:

- The connected user is able to successfully execute an **OpenPrinterEx(\\server)** with administrative privileges (i.e., root or *printer admin*).

> **Tip**
>
> Try this from a Windows 200x/XP DOS box command prompt:
>
> `runas /netonly /user:root rundll32 printui.dll,PrintUIEntry /p /t0 /n \\SAMBA-SERVER\printersharename`
>
> Click on **Printing Preferences**.

- ... contains the setting *show add printer wizard* = yes (the default).

The APW can do various things:

- Upload a new driver to the Samba *[print$]* share.
- Associate an uploaded driver with an existing (but still driverless) print queue.
- Exchange the currently used driver for an existing print queue with one that has been uploaded before.
- Add an entirely new printer to the Samba host (only in conjunction with a working *add printer command*. A corresponding *delete printer command* for removing entries from the **Printers** folder may also be provided).

The last one (add a new printer) requires more effort than the previous ones. To use the APW to successfully add a printer to a Samba server, the *add printer command* must have a defined value. The program hook must successfully add the printer to the UNIX print system (i.e., to `/etc/printcap`, `/etc/cups/printers.conf` or other appropriate files) and to `smb.conf` if necessary.

When using the APW from a client, if the named printer share does not exist, smbd will execute the *add printer command* and reparse to attempt to locate the new printer share. If the share is still not defined, an error of "Access Denied" is returned to the client. The *add printer command* is executed under the context of the connected user, not necessarily a root account. A *map to guest* = bad user may have connected you unwittingly under the wrong privilege. You should check it by using the **smbstatus** command.

21.8.4 Error Message: "Cannot connect under a different Name"

Once you are connected with the wrong credentials, there is no means to reverse the situation other than to close all Explorer windows, and perhaps reboot.

- The **net use \\SAMBA-SERVER\sharename /user:root** gives you an error message: "Multiple connections to a server or a shared resource by the same user utilizing several user names are not allowed. Disconnect all previous connections to the server, esp. the shared resource, and try again."

- Every attempt to "connect a network drive" to `\\SAMBASERVER\\print$` to `z:` is countered by the pertinacious message: "This network folder is currently connected under different credentials (username and password). Disconnect first any existing connection to this network share in order to connect again under a different username and password".

So you close all connections. You try again. You get the same message. You check from the Samba side, using **smbstatus**. Yes, there are more connections. You kill them all. The client still gives you the same error message. You watch the smbd.log file on a high debug level and try reconnect. Same error message, but not a single line in the log. You start to wonder if there was a connection attempt at all. You run ethereal and tcpdump while you try to connect. Result: not a single byte goes on the wire. Windows still gives the error message. You close all Explorer windows and start it again. You try to connect — and this times it works! Windows seems to cache connection information somewhere and does not keep it up to date (if you are unlucky, you might need to reboot to get rid of the error message).

The easiest way to forcefully terminate all connections from your client to a server is by executing:

```
C:\>  net use * /delete
```

This will also disconnect all mapped drives and will allow you create fresh connection as required.

21.8.5 Take Care When Assembling Driver Files

You need to be extremely careful when you take notes about the files belonging to a particular driver. Don't confuse the files for driver version "0" (for Windows 9x/Me, going into `[print$]/WIN/0/`), driver version 2 (kernel mode driver for Windows NT, going into `[print$]/W32X86/2/`; may be used on Windows 200x/XP also), and driver version "3" (non-kernel mode driver going into `[print$]/W32X86/3/`; cannot be used on Windows NT). Quite often these different driver versions contain files that have the same name but actually are very different. If you look at them from the Windows Explorer (they reside in `%WINDOWS%\system32\spool\drivers\W32X86\`), you will probably see names in capital letters, while an **enumdrivers** command from Samba would show mixed or lowercase letters, so it is easy to confuse them. If you install them manually using **rpcclient** and subcommands, you may even succeed without an error message. Only later, when you try install on a client, you will encounter error messages like `This server has no appropriate driver for the printer`.

Here is an example. You are invited to look closely at the various files, compare their names and their spelling, and discover the differences in the composition of the version 2 and 3 sets. Note: the version 0 set contained 40 *`Dependentfiles`*, so I left it out for space reasons:

```
root# rpcclient -U 'Administrator%secret' -c 'enumdrivers 3' 10.160.50.8

 Printer Driver Info 3:
        Version: [3]
        Driver Name: [Canon iR8500 PS3]
        Architecture: [Windows NT x86]
        Driver Path: [\\10.160.50.8\print$\W32X86\3\cns3g.dll]
        Datafile: [\\10.160.50.8\print$\W32X86\3\iR8500sg.xpd]
        Configfile: [\\10.160.50.8\print$\W32X86\3\cns3gui.dll]
        Helpfile: [\\10.160.50.8\print$\W32X86\3\cns3g.hlp]

        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\aucplmNT.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\ucs32p.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\tnl32.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\aussdrv.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\cnspdc.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\aussapi.dat]
```

```
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\cns3407.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\CnS3G.cnt]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\NBAPI.DLL]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\NBIPC.DLL]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\cpcview.exe]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\cpcdspl.exe]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\cpcedit.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\cpcqm.exe]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\cpcspl.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\cfine32.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\cpcr407.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\Cpcqm407.hlp]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\cpcqm407.cnt]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\3\cns3ggr.dll]

        Monitorname: []
        Defaultdatatype: []

Printer Driver Info 3:
        Version: [2]
        Driver Name: [Canon iR5000-6000 PS3]
        Architecture: [Windows NT x86]
        Driver Path: [\\10.160.50.8\print$\W32X86\2\cns3g.dll]
        Datafile: [\\10.160.50.8\print$\W32X86\2\IR5000sg.xpd]
        Configfile: [\\10.160.50.8\print$\W32X86\2\cns3gui.dll]
        Helpfile: [\\10.160.50.8\print$\W32X86\2\cns3g.hlp]

        Dependentfiles: [\\10.160.50.8\print$\W32X86\2\AUCPLMNT.DLL]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\2\aussdrv.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\2\cnspdc.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\2\aussapi.dat]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\2\cns3407.dll]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\2\CnS3G.cnt]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\2\NBAPI.DLL]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\2\NBIPC.DLL]
        Dependentfiles: [\\10.160.50.8\print$\W32X86\2\cns3gum.dll]

        Monitorname: [CPCA Language Monitor2]
        Defaultdatatype: []
```

If we write the "version 2" files and the "version 3" files into different text files and compare the result, we see this picture:

```
root# sdiff 2-files 3-files


 cns3g.dll                      cns3g.dll
 iR8500sg.xpd                   iR8500sg.xpd
 cns3gui.dll                    cns3gui.dll
 cns3g.hlp                      cns3g.hlp
 AUCPLMNT.DLL                 | aucplmNT.dll
                              > ucs32p.dll
                              > tnl32.dll
 aussdrv.dll                    aussdrv.dll
 cnspdc.dll                     cnspdc.dll
 aussapi.dat                    aussapi.dat
 cns3407.dll                    cns3407.dll
 CnS3G.cnt                      CnS3G.cnt
 NBAPI.DLL                      NBAPI.DLL
 NBIPC.DLL                      NBIPC.DLL
 cns3gum.dll                  | cpcview.exe
                              > cpcdspl.exe
                              > cpcqm.exe
                              > cpcspl.dll
                              > cfine32.dll
                              > cpcr407.dll
                              > Cpcqm407.hlp
                              > cpcqm407.cnt
                              > cns3ggr.dll
```

Do not be fooled! Driver files for each version with identical names may be different in their content, as you can see from this size comparison:

```
root# for i in cns3g.hlp cns3gui.dll cns3g.dll; do                  \
           smbclient //10.160.50.8/print\$ -U 'Administrator%xxxx' \
           -c "cd W32X86/3; dir $i; cd .. ; cd 2; dir $i";      \
```

```
        done

  CNS3G.HLP               A   122981  Thu May 30 02:31:00 2002
  CNS3G.HLP               A    99948  Thu May 30 02:31:00 2002

  CNS3GUI.DLL             A  1805824  Thu May 30 02:31:00 2002
  CNS3GUI.DLL             A  1785344  Thu May 30 02:31:00 2002

  CNS3G.DLL               A  1145088  Thu May 30 02:31:00 2002
  CNS3G.DLL               A    15872  Thu May 30 02:31:00 2002
```

In my example were even more differences than shown here. Conclusion: you must be careful to select the correct driver files for each driver version. Don't rely on the names alone, and don't interchange files belonging to different driver versions.

21.8.6 Samba and Printer Ports

Windows NT/2000 print servers associate a port with each printer. These normally take the form of `LPT1:`, `COM1:`, `FILE:`, and so on. Samba must also support the concept of ports associated with a printer. By default, only one printer port, named "Samba Printer Port", exists on a system. Samba does not really need such a "port" in order to print; rather it is a requirement of Windows clients. They insist on being told about an available port when they request this information; otherwise, they throw an error message at you. So Samba fakes the port information to keep the Windows clients happy.

Samba does not support the concept of `Printer Pooling` internally either. Printer pooling assigns a logical printer to multiple ports as a form of load balancing or failover.

If you require multiple ports to be defined for some reason or another (my users and my boss should not know that they are working with Samba), configure the *enumports command*, which can be used to define an external program that generates a listing of ports on a system.

21.8.7 Avoiding Common Client Driver Misconfiguration

So now the printing works, but there are still problems. Most jobs print well, some do not print at all. Some jobs have problems with fonts, which do not look good. Some jobs print fast and some are dead-slow. We cannot cover it all, but we want to encourage you to read the brief paragraph about "Avoiding the Wrong PostScript Driver Settings" in Chapter 22, "CUPS Printing Support", Section 22.10.16.

21.9 The Imprints Toolset

The Imprints tool set provides a UNIX equivalent of the Windows NT APW. For complete information, please refer to the Imprints[3] Web site as well as the documentation included with the Imprints source distribution. This section provides only a brief introduction to the features of Imprints.

Unfortunately, the Imprints toolset is no longer maintained. As of December 2000, the project is in need of a new maintainer. The most important skill to have is Perl coding and an interest in MS-RPC-based printing used in Samba. If you wish to volunteer, please coordinate your efforts on the Samba technical mailing list. The toolset is still in usable form, but only for a series of older printer models where there are prepared packages to use. Packages for more up-to-date print devices are needed if Imprints should have a future. Information regarding the Imprints toolset can be obtained from the Imprints[4] home page.

21.9.1 What Is Imprints?

Imprints is a collection of tools for supporting these goals:

- Providing a central repository of information regarding Windows NT and 95/98 printer driver packages.
- Providing the tools necessary for creating the Imprints printer driver packages.

[3] <http://imprints.sourceforge.net/>
[4] <http://imprints.sourceforge.net/>

- Providing an installation client that will obtain printer drivers from a central Internet (or intranet) Imprints Server repository and install them on remote Samba and Windows NT4 print servers.

21.9.2 Creating Printer Driver Packages

The process of creating printer driver packages is beyond the scope of this document (refer to Imprints.txt, included with the Samba distribution for more information). In short, an Imprints driver package is a gzipped tarball containing the driver files, related INF files, and a control file needed by the installation client.

21.9.3 The Imprints Server

The Imprints server is really a database server that may be queried via standard HTTP mechanisms. Each printer entry in the database has an associated URL for the actual downloading of the package. Each package is digitally signed via GnuPG, which can be used to verify that the package downloaded is actually the one referred in the Imprints database. It is strongly recommended that this security check not be disabled.

21.9.4 The Installation Client

More information regarding the Imprints installation client is available from the documentation file `Imprints-Client-HOWTO.ps` that is included with the Imprints source package. The Imprints installation client comes in two forms:

- A set of command-line Perl scripts.
- A GTK+-based graphical interface to the command-line Perl scripts.

The installation client (in both forms) provides a means of querying the Imprints database server for a matching list of known printer model names as well as a means to download and install the drivers on remote Samba and Windows NT print servers.

The basic installation process is in four steps, and Perl code is wrapped around smbclient and rpcclient.

- For each supported architecture for a given driver:

1. rpcclient: Get the appropriate upload directory on the remote server.
2. smbclient: Upload the driver files.
3. rpcclient: Issues an AddPrinterDriver() MS-RPC.

- rpcclient: Issues an AddPrinterEx() MS-RPC to actually create the printer.

One of the problems encountered when implementing the Imprints tool set was the namespace issues between various supported client architectures. For example, Windows NT includes a driver named "Apple LaserWriter II NTX v51.8", and Windows 95 calls its version of this driver "Apple LaserWriter II NTX".

The problem is how to know what client drivers have been uploaded for a printer. An astute reader will remember that the Windows NT Printer Properties dialog only includes space for one printer driver name. A quick look in the Windows NT 4.0 system registry at:

```
HKLM\System\CurrentControlSet\Control\Print\Environment
```

will reveal that Windows NT always uses the NT driver name. This is okay because Windows NT always requires that at least the Windows NT version of the printer driver is present. Samba does not have the requirement internally; therefore, "How can you use the NT driver name if it has not already been installed?"

The way of sidestepping this limitation is to require that all Imprints printer driver packages include both the Intel Windows NT and 95/98 printer drivers and that the NT driver is installed first.

21.10 Adding Network Printers without User Interaction

The following MS Knowledge Base article may be of some help if you need to handle Windows 2000 clients: *How to Add Printers with No User Interaction in Windows 2000,* (Microsoft KB 189105[5]). It also applies to Windows XP Professional clients. The ideas sketched out in this section are inspired by this article, which describes a command-line method that can be applied to install network and local printers and their drivers. This is most useful

[5] <http://support.microsoft.com/default.aspx?scid=kb;en-us;189105>

if integrated in Logon Scripts. You can see what options are available by typing in the command prompt (**DOS box**):

```
rundll32 printui.dll,PrintUIEntry /?
```

A window pops up that shows you all of the command-line switches available. An extensive list of examples is also provided. This is only for Windows 200x/XP; it does not work on Windows NT. Windows NT probably has some other tools in the respective Resource Kit. Here is a suggestion about what a client logon script might contain, with a short explanation of what the lines actually do (it works if 200x/XP Windows clients access printers via Samba, and works for Windows-based print servers too):

```
rundll32 printui.dll,PrintUIEntry /dn /n "\\cupsserver\infotec2105-IPDS" /q
rundll32 printui.dll,PrintUIEntry /in /n "\\cupsserver\infotec2105-PS"
rundll32 printui.dll,PrintUIEntry /y /n "\\cupsserver\infotec2105-PS"
```

Here is a list of the used command-line parameters:

/dn deletes a network printer.

/q quiet modus.

/n names a printer.

/in adds a network printer connection.

/y sets printer as default printer.

- Line 1 deletes a possibly existing previous network printer *infotec2105-IPDS* (which had used native Windows drivers with LPRng that were removed from the server that was converted to CUPS). The **/q** at the end prevents confirm or error dialog boxes from popping up. They should not be presented to the user logging on.
- Line 2 adds the new printer *infotec2105-PS* (which actually is the same physical device but is now run by the new CUPS printing system and associated with the CUPS/Adobe PS drivers). The printer and its

driver must have been added to Samba prior to the user logging in (e.g., by a procedure as discussed earlier in this chapter or by running **cupsaddsmb**). The driver is now autodownloaded to the client PC where the user is about to log in.

- Line 3 sets the default printer to this new network printer (there might be several other printers installed with this same method, and some may be local as well, so we decide for a default printer). The default printer selection may, of course, be different for different users.

The second line only works if the printer *infotec2105-PS* has an already working print queue on the `cupsserver` and if the printer drivers have been successfully uploaded (via the **APW**, **smbclient/rpcclient**, or **cupsaddsmb**) into the *[print$]* driver repository of Samba. Some Samba versions prior to version 3.0 required a restart of smbd after the printer install and the driver upload; otherwise the script (or any other client driver download) would fail.

Since there is no easy way to test for the existence of an installed network printer from the logon script, do not bother checking. Just allow the de-installation/re-installation to occur every time a user logs in; it's really quick anyway (1 to 2 seconds).

The additional benefits for this are:

- It puts in place any printer default setup changes automatically at every user logon.
- It allows for "roaming" users' login to the domain from different workstations.

Since network printers are installed per user, this much simplifies the process of keeping the installation up to date. The few extra seconds at logon time will not really be noticeable. Printers can be centrally added, changed, and deleted at will on the server with no user intervention required from the clients (you just need to keep the logon scripts up to date).

21.11 The addprinter Command

The **addprinter** command can be configured to be a shell script or program executed by Samba. It is triggered by running the APW from a client against the Samba print server. The APW asks the user to fill in several fields (such as printer name, driver to be used, comment, port monitor,

and so on). These parameters are passed on to Samba by the APW. If the addprinter command is designed in a way that it can create a new printer (through writing correct printcap entries on legacy systems or by executing the **lpadmin** command on more modern systems) and create the associated share, then the APW will in effect really create a new printer on Samba and the UNIX print subsystem!

21.12 Migration of Classical Printing to Samba

The basic NT-style printer driver management has not changed considerably in 3.0 over the 2.2.x releases (apart from many small improvements). Here migration should be quite easy, especially if you followed previous advice to stop using deprecated parameters in your setup. For migrations from an existing 2.0.x setup, or if you continued Windows 9x/Me-style printing in your Samba 2.2 installations, it is more of an effort. Please read the appropriate release notes and the HOWTO Collection for Samba-2.2.x. You can follow several paths. Here are possible scenarios for migration:

- You need to study and apply the new Windows NT printer and driver support. Previously used parameters *`printer driver file`*, *`printer driver`*, and *`printer driver location`* are no longer supported.
- If you want to take advantage of Windows NT printer driver support, you also need to migrate the Windows 9x/Me drivers to the new setup.
- An existing `printers.def` file (the one specified in the now removed parameter *`printer driver file`*) will no longer work with Samba-3. In 3.0, smbd attempts to locate Windows 9x/Me driver files for the printer in *[print$]* and additional settings in the TDB and only there; if it fails, it will *not* (as 2.2.x used to do) drop down to using a `printers.def` (and all associated parameters). The make_printerdef tool is removed and there is no backward compatibility for this.
- You need to install a Windows 9x/Me driver into the *[print$]* share for a printer on your Samba host. The driver files will be stored in the "WIN40/0" subdirectory of *[print$]*, and some other settings and information go into the printing-related TDBs.
- If you want to migrate an existing `printers.def` file into the new setup, the only current solution is to use the Windows NT APW to install the NT drivers and the 9x/Me drivers. This can be scripted

using smbclient and rpcclient. See the Imprints installation client on the Imprints[6] web site for example. See also the discussion of rpcclient usage in Chapter 22, "CUPS Printing Support".

21.13 Publishing Printer Information in Active Directory or LDAP

This topic has also been addressed in Chapter 13, "Remote and Local Management: The Net Command". If you wish to volunteer your services to help document this further, please contact John H. Terpstra[7].

21.14 Common Errors

21.14.1 I Give My Root Password but I Do Not Get Access

Do not confuse the root password, which is valid for the UNIX system (and in most cases stored in the form of a one-way hash in a file named `/etc/shadow`), with the password used to authenticate against Samba. Samba does not know the UNIX password. Root access to Samba resources requires that a Samba account for root must first be created. This is done with the **smbpasswd** command as follows:

```
root#  smbpasswd -a root
New SMB password: secret
Retype new SMB password: secret
```

21.14.2 My Print Jobs Get Spooled into the Spooling Directory, but Then Get Lost

Do not use the existing UNIX print system spool directory for the Samba spool directory. It may seem convenient and a savings of space, but it only leads to problems. The two must be separate. The UNIX/Linux system print spool directory (e.g., `/var/spool/cups`) is typically owned by a non-privileged user such as `cups` or `lp`. Additionally. the permissions on the

[6] <http://imprints.sourceforge.net/>
[7] <mail://jht@samba.org>

spool directory are typically restrictive to the owner and/or group. On the other hand, the Samba spool directory must be world writable, and should have the 't' bit set to ensure that only a temporary spool file owner can change or delete the file.

Depending on the type of print spooling system in use on the UNIX/Linux host, files that the spool management application finds and that are not currently part of job queue that it is managing can be deleted. This may explain the observation that jobs are spooled (by Samba) into this directory and just disappear.

Chapter 22

CUPS PRINTING SUPPORT

22.1 Introduction

22.1.1 Features and Benefits

The Common UNIX Print System (CUPS[1]) has become quite popular. All major Linux distributions now ship it as their default printing system. To many, it is still a mystical tool. Mostly, it just works. People tend to regard it as a "black box" that they do not want to look into as long as it works. But once there is a little problem, they have trouble finding out where to start debugging it. Refer to Chapter 21, "Classical Printing Support", which contains much information that is also relevant to CUPS.

CUPS sports quite a few unique and powerful features. While its basic functions may be grasped quite easily, they are also new. Because it is different from other, more traditional printing systems, it is best not to try to apply any prior knowledge about printing to this new system. Rather, try to understand CUPS from the beginning. This documentation will lead you to a complete understanding of CUPS. Let's start with the most basic things first.

22.1.2 Overview

CUPS is more than just a print spooling system. It is a complete printer management system that complies with the new Internet Printing Protocol (IPP). IPP is an industry and Internet Engineering Task Force (IETF) standard for network printing. Many of its functions can be managed remotely

[1] <http://www.cups.org/>

(or locally) via a Web browser (giving you platform-independent access to the CUPS print server). Additionally, it has the traditional command line and several more modern GUI interfaces (GUI interfaces developed by third parties, like KDE's overwhelming KDEPrint[2]).

CUPS allows creation of *raw* printers (i.e., no print file format translation) as well as *smart* printers (i.e., CUPS does file format conversion as required for the printer). In many ways, this gives CUPS capabilities similar to the MS Windows print monitoring system. Of course, if you are a CUPS advocate, you would argue that CUPS is better! In any case, let us now explore how to configure CUPS for interfacing with MS Windows print clients via Samba.

22.2 Basic CUPS Support Configuration

Printing with CUPS in the most basic `smb.conf` setup in Samba-3.0 (as was true for 2.2.x) requires just two parameters: *printing* = cups and *printcap* = cups. CUPS does not need a printcap file. However, the `cupsd.conf` configuration file knows of two related directives that control how such a file will be automatically created and maintained by CUPS for the convenience of third-party applications (example: *`Printcap /etc/printcap`* and *`PrintcapFormat BSD`*). Legacy programs often require the existence of a printcap file containing printer names or they will refuse to print. Make sure CUPS is set to generate and maintain a printcap file. For details, see **man cupsd.conf** and other CUPS-related documentation, like the wealth of documents regarding the CUPS server itself available from the CUPS[3] web site.

22.2.1 Linking smbd with libcups.so

Samba has a special relationship to CUPS. Samba can be compiled with CUPS library support. Most recent installations have this support enabled. By default, CUPS linking is compiled into smbd and other Samba binaries. Of course, you can use CUPS even if Samba is not linked against `libcups.so` — but there are some differences in required or supported configuration.

When Samba is compiled and linked with `libcups`, *printcap* = cups uses the CUPS API to list printers, submit jobs, query queues, and so on. Otherwise

[2] <http://printing.kde.org/>
[3] <http://localhost:631/documentation.html>

it maps to the System V commands with an additional **-oraw** option for printing. On a Linux system, you can use the **ldd** utility to find out if smbd has been linked with the libcups library (**ldd** may not be present on other OS platforms, or its function may be embodied by a different command):

```
root# ldd `which smbd`
libssl.so.0.9.6 => /usr/lib/libssl.so.0.9.6 (0x4002d000)
libcrypto.so.0.9.6 => /usr/lib/libcrypto.so.0.9.6 (0x4005a000)
libcups.so.2 => /usr/lib/libcups.so.2 (0x40123000)
[....]
```

The line `libcups.so.2 =>` `/usr/lib/libcups.so.2 (0x40123000)` shows there is CUPS support compiled into this version of Samba. If this is the case, and printing = cups is set, then *any otherwise manually set print command in* `smb.conf` *is ignored*. This is an important point to remember!

TIP

Should it be necessary, for any reason, to set your own print commands, you can do this by setting *printing* = sysv. However, you will lose all the benefits of tight CUPS-Samba integration. When you do this, you must manually configure the printing system commands (most important: *print command*; other commands are *lppause command*, *lpresume command*, *lpq command*, *lprm command*, *queuepause command* and *queue resume command*).

22.2.2 Simple smb.conf Settings for CUPS

To summarize, Example 22.2.1 shows the simplest printing-related setup for `smb.conf` to enable basic CUPS support:

This is all you need for basic printing setup for CUPS. It will print all graphic, text, PDF, and PostScript files submitted from Windows clients. However, most of your Windows users would not know how to send these

Example 22.2.1 Simplest Printing-Related smb.conf

```
[global]
        load printers = yes
        printing = cups
        printcap name = cups
 [printers]
        comment = All Printers
        path = /var/spool/samba
        browseable = no
        guest ok = yes
        writable = no
        printable = yes
        printer admin = root, @ntadmins, @smbprintadm
```

kinds of files to print without opening a GUI application. Windows clients tend to have local printer drivers installed, and the GUI application's print buttons start a printer driver. Your users also rarely send files from the command line. Unlike UNIX clients, they rarely submit graphic, text, or PDF formatted files directly to the spooler. They nearly exclusively print from GUI applications with a "printer driver" hooked between the application's native format and the print data stream. If the backend printer is not a PostScript device, the print data stream is "binary," sensible only for the target printer. Read on to learn what problem this may cause and how to avoid it.

22.2.3 More Complex CUPS smb.conf Settings

Example 22.2.2 is a slightly more complex printing-related setup for `smb.conf`. It enables general CUPS printing support for all printers, but defines one printer share, which is set up differently.

This special share is only for testing purposes. It does not write the print job to a file. It just logs the job parameters known to Samba into the `/tmp/smbprn.log` file and deletes the job-file. Moreover, the *printer admin* of this share is "kurt" (not the "@ntadmins" group), guest access is not allowed, the share isn't published to the Network Neighborhood (so you need to know it is there), and it allows access from only three hosts. To prevent CUPS from kicking in and taking over the print jobs for that share, we need to set *printing* = sysv and *printcap* = lpstat.

Example 22.2.2 Overriding Global CUPS Settings for One Printer

```
[global]
        printing = cups
        printcap name = cups
        load printers = yes
[printers]
        comment = All Printers
        path = /var/spool/samba
        guest ok = yes
        writable = no
        printable = yes
        printer admin = root, @ntadmins, @smbprintadm
[special_printer]
        comment = A special printer with his own settings
        path = /var/spool/samba-special
        printing = sysv
        printcap = lpstat
        print command = echo "NEW: `date`: printfile %f"  ↩
           >> /tmp/smbprn.log ; echo "      `date`: p-%p  ↩
           s-%s f-%f" >> /tmp/smbprn.log ; echo "      `  ↩
           date`: j-%j J-%J z-%z c-%c" >> /tmp/smbprn. ↩
           log ; rm %f
        guest ok = no
        writable = no
        printable = yes
        printer admin = kurt
        hosts deny = 0.0.0.0
        hosts allow = turbo_xp, 10.160.50.23,  ↩
           10.160.51.60
```

22.3 Advanced Configuration

Before we delve into all the configuration options, let us clarify a few points. *Network printing needs to be organized and set up correctly.* This frequently doesn't happen. Legacy systems or small business LAN environments often lack design and good housekeeping.

22.3.1 Central Spooling vs. "Peer-to-Peer" Printing

Many small office or home networks, as well as badly organized larger environments, allow each client a direct access to available network printers. This is generally a bad idea. It often blocks one client's access to the printer when another client's job is printing. It might freeze the first client's application while it is waiting to get rid of the job. Also, there are frequent complaints about various jobs being printed with their pages mixed with each other. A better concept is the use of a print server: it routes all jobs through one central system, which responds immediately, takes jobs from multiple concurrent clients, and transfers them to the printer(s) in the correct order.

22.3.2 Raw Print Serving: Vendor Drivers on Windows Clients

Most traditionally configured UNIX print servers acting on behalf of Samba's Windows clients represented a really simple setup. Their only task was to manage the "raw" spooling of all jobs handed to them by Samba. This approach meant that the Windows clients were expected to prepare the print job file that is ready to be sent to the printing device. In this case, a native (vendor-supplied) Windows printer driver needs to be installed on each and every client for the target device.

It is possible to configure CUPS, Samba, and your Windows clients in the same traditional and simple way. When CUPS printers are configured for raw print-through mode operation, it is the responsibility of the Samba client to fully render the print job (file). The file must be sent in a format that is suitable for direct delivery to the printer. Clients need to run the vendor-provided drivers to do this. In this case, CUPS will not do any print file format conversion work.

The easiest printing configuration possible is raw print-through. This is achieved by installation of the printer as if it were physically attached to the Windows client. You then redirect output to a raw network print queue. This procedure may be followed to achieve this: Configuration Steps for Raw CUPS Printing Support

1. Edit `/etc/cups/mime.types` to uncomment the line near the end of the file that has:

```
#application/octet-...
```

2. Do the same for the file `/etc/cups/mime.convs`.

3. Add a raw printer using the Web interface. Point your browser at `http://localhost:631`. Enter Administration, and add the printer following the prompts. Do not install any drivers for it. Choose Raw. Choose queue name `Raw Queue`.

4. In the `smb.conf` file `[printers]` section add *use client driver* = Yes, and in the `[global]` section add *printing* = CUPS, plus *printcap* = CUPS.

5. Install the printer as if it is a local printer, that is, Printing to `LPT1:`.

6. Edit the configuration under the **Detail** tab and create a `local port` that points to the raw printer queue that you have configured above. Example: `\\server\raw_q`. Here, the name `raw_q` is the name you gave the print queue in the CUPS environment.

22.3.3 Installation of Windows Client Drivers

The printer drivers on the Windows clients may be installed in two functionally different ways:

- Manually install the drivers locally on each client, one by one; this yields the old LanMan style printing and uses a `\\sambaserver\printershare` type of connection.

- Deposit and prepare the drivers (for later download) on the print server (Samba); this enables the clients to use "Point'n'Print" to get drivers semi-automatically installed the first time they access the printer; with this method NT/200x/XP clients use the *SPOOLSS/MS-RPC* type printing calls.

The second method is recommended for use over the first as it reduces the administrative efforts and prevents that different versions of the drivers are used accidentally.

22.3.4 Explicitly Enable "raw" Printing for *application/octet-stream*

If you use the first option (drivers are installed on the client side), there is one setting to take care of: CUPS needs to be told that it should allow "raw" printing of deliberate (binary) file formats. The CUPS files that need to be correctly set for raw mode printers to work are:

- `/etc/cups/mime.types`
- `/etc/cups/mime.convs`

Both contain entries (at the end of the respective files) that must be uncommented to allow RAW mode operation. In `/etc/cups/mime.types`, make sure this line is present:

```
application/octet-stream
```

In `/etc/cups/mime.convs`, have this line:

```
application/octet-stream   application/vnd.cups-raw   0   -
```

If these two files are not set up correctly for raw Windows client printing, you may encounter the dreaded `Unable to convert file 0` in your CUPS `error_log` file.

> NOTE
>
> Editing the `mime.convs` and the `mime.types` file does not *enforce* "raw" printing, it only *allows* it.

Background. That CUPS is a more security-aware printing system than traditional ones does not by default allow a user to send deliberate (possibly binary) data to printing devices. This could be easily abused to launch a "Denial of Service" attack on your printer(s), causing at least the loss of a lot of paper and ink. "Unknown" data are tagged by CUPS as *MIME type: application/octet-stream* and not allowed to go to the printer.

By default, you can only send other (known) MIME types "raw." Sending data "raw" means that CUPS does not try to convert them and passes them to the printer untouched.

This is all you need to know to get the CUPS/Samba combo printing "raw" files prepared by Windows clients, which have vendor drivers locally installed. If you are not interested in background information about more advanced CUPS/Samba printing, simply skip the remaining sections of this chapter.

22.3.5 Driver Upload Methods

This section describes three familiar methods, plus one new one, by which printer drivers may be uploaded.

If you want to use the MS-RPC-type printing, you must upload the drivers onto the Samba server first (*[print$]* share). For a discussion on how to deposit printer drivers on the Samba host (so the Windows clients can download and use them via "Point'n'Print"), please refer to the Chapter 21, "Classical Printing Support" of this book. There you will find a description or reference to three methods of preparing the client drivers on the Samba server:

- The GUI, "Add Printer Wizard" *upload-from-a-Windows-client* method.
- The command line, "smbclient/rpcclient" upload-from-a-UNIX-workstation method.
- The Imprints tool set method.

These three methods apply to CUPS all the same. The **cupsaddsmb** utility is a new and more convenient way to load the Windows drivers into Samba and is provided if you use CUPS.

cupsaddsmb is discussed in much detail later in this chapter. But we first explore the CUPS filtering system and compare the Windows and UNIX printing architectures.

22.4 Advanced Intelligent Printing with PostScript Driver Download

We now know how to set up a "dump" print server, that is, a server that spools print jobs "raw", leaving the print data untouched.

You might need to set up CUPS in a smarter way. The reasons could be manifold:

- Maybe your boss wants to get monthly statistics: Which printer did how many pages? What was the average data size of a job? What was the average print run per day? What are the typical hourly peaks in printing? Which department prints how much?
- Maybe you are asked to set up a print quota system: Users should not be able to print more jobs once they have surpassed a given limit per period.
- Maybe your previous network printing setup is a mess and must be re-organized from a clean beginning.
- Maybe you are experiencing too many "blue screens" originating from poorly debugged printer drivers running in NT "kernel mode"?

These goals cannot be achieved by a raw print server. To build a server meeting these requirements, you'll first need to learn how CUPS works and how you can enable its features.

What follows is the comparison of some fundamental concepts for Windows and UNIX printing, then a description of the CUPS filtering system, how it works, and how you can tweak it.

22.4.1 GDI on Windows, PostScript on UNIX

Network printing is one of the most complicated and error-prone day-to-day tasks any user or administrator may encounter. This is true for all OS platforms, and there are reasons it is so.

You can't expect to throw just any file format at a printer and have it get printed. A file format conversion must take place. The problem is that there is no common standard for print file formats across all manufacturers and printer types. While PostScript (trademark held by Adobe) and, to an extent, PCL (trademark held by Hewlett-Packard) have developed into

semi-official "standards" by being the most widely used page description languages (PDLs), there are still many manufacturers who "roll their own" (their reasons may be unacceptable license fees for using printer-embedded PostScript interpreters, and so on).

22.4.2 Windows Drivers, GDI, and EMF

In Windows OS, the format conversion job is done by the printer drivers. On MS Windows OS platforms all application programmers have at their disposal a built-in API, the graphical device interface (GDI), as part and parcel of the OS itself to base themselves on. This GDI core is used as one common unified ground for all Windows programs to draw pictures, fonts, and documents *on screen* as well as *on paper* (print). Therefore, printer driver developers can standardize on a well-defined GDI output for their own driver input. Achieving WYSIWYG (What You See Is What You Get) is relatively easy, because the on-screen graphic primitives, as well as the on-paper drawn objects, come from one common source. This source, the GDI, often produces a file format called Enhanced MetaFile (EMF). The EMF is processed by the printer driver and converted to the printer-specific file format.

NOTE

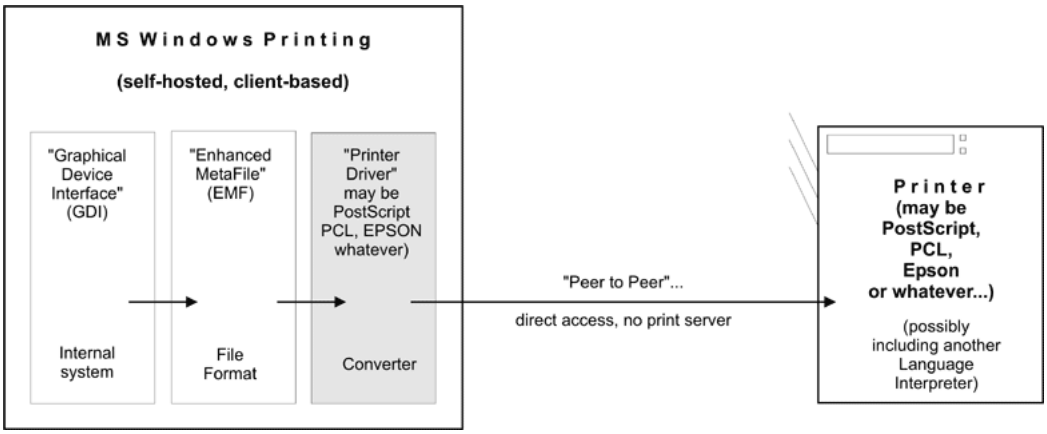
To the GDI foundation in MS Windows, Apple has chosen to put paper and screen output on a common foundation for its (BSD-UNIX-based, did you know?) Mac OS X and Darwin operating systems. Apple's *core graphic engine* uses a *PDF* derivative for all display work.

The example in Figure 22.1 illustrates local Windows printing.

22.4.3 UNIX Printfile Conversion and GUI Basics

In UNIX and Linux, there is no comparable layer built into the OS kernel(s) or the X (screen display) server. Every application is responsible for itself to create its print output. Fortunately, most use PostScript and that at

Figure 22.1 Windows Printing to a Local Printer.



least gives some common ground. Unfortunately, there are many different levels of quality for this PostScript. And worse, there is a huge difference (and no common root) in the way the same document is displayed on screen and how it is presented on paper. WYSIWYG is more difficult to achieve. This goes back to the time, decades ago, when the predecessors of X.org, designing the UNIX foundations and protocols for graphical user interfaces, refused to take responsibility for "paper output", as some had demanded at the time, and restricted itself to "on-screen only." (For some years now, the "Xprint" project has been under development, attempting to build printing support into the X framework, including a PostScript and a PCL driver, but it is not yet ready for prime time.) You can see this unfavorable inheritance up to the present day by looking into the various "font" directories on your system; there are separate ones for fonts used for X display and fonts to be used on paper.

Background. The PostScript programming language is an "invention" by Adobe, but its specifications have been published extensively. Its strength lies in its powerful abilities to describe graphical objects (fonts, shapes, patterns, lines, curves, and dots), their attributes (color, linewidth), and the way to manipulate (scale, distort, rotate, shift) them. Because of its open specification, anybody with the skill can start writing his or her own implementation of a PostScript interpreter and use it to display PostScript files on screen or on paper. Most graphical output devices are based on the concept of "raster images" or "pixels" (one notable exception is pen

plotters). Of course, you can look at a PostScript file in its textual form and you will be reading its PostScript code, the language instructions that need to be interpreted by a rasterizer. Rasterizers produce pixel images, which may be displayed on screen by a viewer program or on paper by a printer.

22.4.4 PostScript and Ghostscript

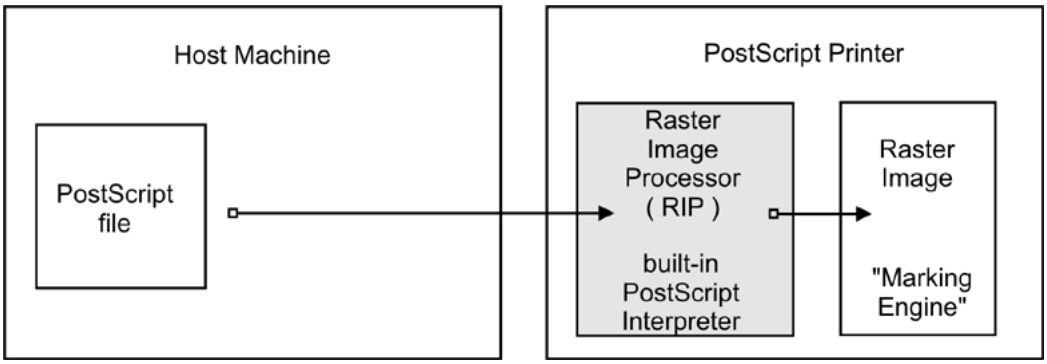
So UNIX is lacking a common ground for printing on paper and displaying on screen. Despite this unfavorable legacy for UNIX, basic printing is fairly easy if you have PostScript printers at your disposal. The reason is that these devices have a built-in PostScript language "interpreter," also called a raster image processor (RIP), (which makes them more expensive than other types of printers; throw PostScript toward them, and they will spit out your printed pages. The RIP does all the hard work of converting the PostScript drawing commands into a bitmap picture as you see it on paper, in a resolution as done by your printer. This is no different than PostScript printing a file from a Windows origin.

NOTE

Traditional UNIX programs and printing systems — while using PostScript — are largely not PPD-aware. PPDs are "PostScript Printer Description" files. They enable you to specify and control all options a printer supports: duplexing, stapling, and punching. Therefore, UNIX users for a long time couldn't choose many of the supported device and job options, unlike Windows or Apple users. But now there is CUPS. as illustrated in Figure 22.2.

However, there are other types of printers out there. These do not know how to print PostScript. They use their own PDL, often proprietary. To print to them is much more demanding. Since your UNIX applications mostly produce PostScript, and since these devices do not understand PostScript, you need to convert the print files to a format suitable for your printer on the host before you can send it away.

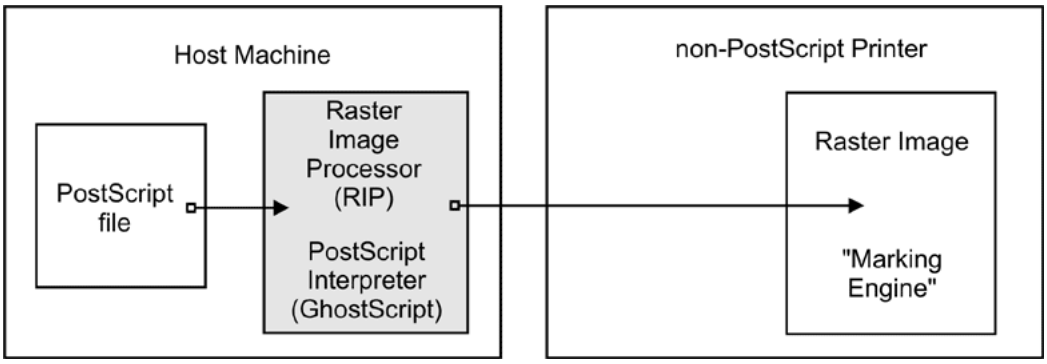
Figure 22.2 Printing to a PostScript Printer.



22.4.5 Ghostscript: The Software RIP for Non-PostScript Printers

Here is where Ghostscript kicks in. Ghostscript is the traditional (and quite powerful) PostScript interpreter used on UNIX platforms. It is a RIP in software, capable of doing a *lot* of file format conversions for a very broad spectrum of hardware devices as well as software file formats. Ghostscript technology and drivers are what enable PostScript printing to non-PostScript hardware. This is shown in Figure 22.3.

Figure 22.3 Ghostscript as a RIP for Non-PostScript Printers.



TIP

Use the "gs -h" command to check for all built-in "devices" on your Ghostscript version. If you specify a parameter of *-sDEVICE=png256* on your Ghostscript command line, you are asking Ghostscript to convert the input into a PNG file. Naming a "device" on the command line is the most important single parameter to tell Ghostscript exactly how it should render the input. New Ghostscript versions are released at fairly regular intervals, now by artofcode LLC. They are initially put under the "AFPL" license, but re-released under the GNU GPL as soon as the next AFPL version appears. GNU Ghostscript is probably the version installed on most Samba systems. But it has some deficiencies. Therefore, ESP Ghostscript was developed as an enhancement over GNU Ghostscript, with lots of bug-fixes, additional devices, and improvements. It is jointly maintained by developers from CUPS, Gutenprint, MandrakeSoft, SuSE, Red Hat, and Debian. It includes the "cups" device (essential to print to non-PS printers from CUPS).

22.4.6 PostScript Printer Description (PPD) Specification

While PostScript in essence is a PDL to represent the page layout in a device-independent way, real-world print jobs are always ending up being output on hardware with device-specific features. To take care of all the differences in hardware and to allow for innovations, Adobe has specified a syntax and file format for PostScript Printer Description (PPD) files. Every PostScript printer ships with one of these files.

PPDs contain all the information about general and special features of the given printer model: Which different resolutions can it handle? Does it have a duplexing unit? How many paper trays are there? What media types and sizes does it take? For each item, it also names the special command string to be sent to the printer (mostly inside the PostScript file) in order to enable it.

Information from these PPDs is meant to be taken into account by the printer drivers. Therefore, installed as part of the Windows PostScript driver for a given printer is the printer's PPD. Where it makes sense, the PPD features are presented in the drivers' UI dialogs to display to the user a choice of print options. In the end, the user selections are somehow written (in the form of special PostScript, PJL, JCL, or vendor-dependent commands) into the PostScript file created by the driver.

WARNING

A PostScript file that was created to contain device-specific commands for achieving a certain print job output (e.g., duplexed, stapled, and punched) on a specific target machine may not print as expected, or may not be printable at all on other models; it also may not be fit for further processing by software (e.g., by a PDF distilling program).

22.4.7 Using Windows-Formatted Vendor PPDs

CUPS can handle all spec-compliant PPDs as supplied by the manufacturers for their PostScript models. Even if a vendor does not mention our favorite OS in his or her manuals and brochures, you can safely trust this: *If you get the Windows NT version of the PPD, you can use it unchanged in CUPS* and thus access the full power of your printer just like a Windows NT user could!

TIP

To check the spec compliance of any PPD online, go to <http://www.cups.org/testppd.php> and upload your PPD. You will see the results displayed immediately. CUPS in all versions after 1.1.19 has a much stricter internal PPD parsing and checking code enabled; in case of printing trouble, this online resource should be one of your first pit stops.

WARNING

For real PostScript printers, *do not* use the *Foomatic* or *cupsomatic* PPDs from Linuxprinting.org. With these devices, the original vendor-provided PPDs are always the first choice.

TIP

If you are looking for an original vendor-provided PPD of a specific device, and you know that an NT4 box (or any other Windows box) on your LAN has the PostScript driver installed, just use **smbclient //NT4-box/print\$ -U username** to access the Windows directory where all printer driver files are stored. First look in the `W32X86/2` subdirectory for the PPD you are seeking.

22.4.8 CUPS Also Uses PPDs for Non-PostScript Printers

CUPS also uses specially crafted PPDs to handle non-PostScript printers. These PPDs are usually not available from the vendors (and no, you can't just take the PPD of a PostScript printer with the same model name and

hope it works for the non-PostScript version too). To understand how these PPDs work for non-PS printers, we first need to dive deeply into the CUPS filtering and file format conversion architecture. Stay tuned.

22.5 The CUPS Filtering Architecture

The core of the CUPS filtering system is based on Ghostscript. In addition to Ghostscript, CUPS uses some other filters of its own. You (or your OS vendor) may have plugged in even more filters. CUPS handles all data file formats under the label of various MIME types. Every incoming print file is subjected to an initial autotyping. The autotyping determines its given MIME type. A given MIME type implies zero or more possible filtering chains relevant to the selected target printer. This section discusses how MIME types recognition and conversion rules interact. They are used by CUPS to automatically set up a working filtering chain for any given input data format.

If CUPS rasterizes a PostScript file natively to a bitmap, this is done in two stages:

- The first stage uses a Ghostscript device named "cups" (this is since version 1.1.15) and produces a generic raster format called "CUPS raster".
- The second stage uses a "raster driver" that converts the generic CUPS raster to a device-specific raster.

Make sure your Ghostscript version has the "cups" device compiled in (check with **gs -h | grep cups**). Otherwise you may encounter the dreaded `Unable to convert file 0` in your CUPS error_log file. To have "cups" as a device in your Ghostscript, you either need to patch GNU Ghostscript and recompile or use ESP Ghostscript[4]. The superior alternative is ESP Ghostscript. It supports not just CUPS, but 300 other devices (while GNU Ghostscript supports only about 180). Because of this broad output device support, ESP Ghostscript is the first choice for non-CUPS spoolers, too. It is now recommended by Linuxprinting.org for all spoolers.

CUPS printers may be set up to use external rendering paths. One of the most common is provided by the Foomatic/cupsomatic concept from

[4] <http://www.cups.org/ghostscript.php>

Linuxprinting.org[5]. This uses the classical Ghostscript approach, doing everything in one step. It does not use the "cups" device, but one of the many others. However, even for Foomatic/cupsomatic usage, best results and broadest printer model support is provided by ESP Ghostscript (more about Foomatic/cupsomatic, particularly the new version called now *foomatic-rip*, follows).

22.5.1 MIME Types and CUPS Filters

CUPS reads the file `/etc/cups/mime.types` (and all other files carrying a `*.types` suffix in the same directory) upon startup. These files contain the MIME type recognition rules that are applied when CUPS runs its autotyping routines. The rule syntax is explained in the man page for `mime.types` and in the comments section of the `mime.types` file itself. A simple rule reads like this:

```
application/pdf         pdf string(0,%PDF)
```

This means if a filename has a `.pdf` suffix or if the magic string *%PDF* is right at the beginning of the file itself (offset 0 from the start), then it is a PDF file (*application/pdf*). Another rule is this:

```
application/postscript  ai eps ps string(0,%!) string(0,<04>%!)
```

If the filename has one of the suffixes `.ai`, `.eps`, `.ps`, or if the file itself starts with one of the strings *%!* or *<04>%!*, it is a generic PostScript file (*application/postscript*).

Warning

Don't confuse the other mime.types files your system might be using with the one in the `/etc/cups/` directory.

[5] <http://www.linuxprinting.org/>

NOTE

There is an important difference between two similar MIME types in CUPS: one is *application/postscript*, the other is *application/vnd.cups-postscript*. While *application/postscript* is meant to be device-independent, job options for the file are still outside the PS file content, embedded in command line or environment variables by CUPS, *application/vnd.cups-postscript* may have the job options inserted into the PostScript data itself (where applicable). The transformation of the generic PostScript (*application/postscript*) to the device-specific version (*application/vnd.cups-postscript*) is the responsibility of the CUPS *pstops* filter. pstops uses information contained in the PPD to do the transformation.

CUPS can handle ASCII text, HP-GL, PDF, PostScript, DVI, and many image formats (GIF, PNG, TIFF, JPEG, Photo-CD, SUN-Raster, PNM, PBM, SGI-RGB, and more) and their associated MIME types with its filters.

22.5.2 MIME Type Conversion Rules

CUPS reads the file `/etc/cups/mime.convs` (and all other files named with a `*.convs` suffix in the same directory) upon startup. These files contain lines naming an input MIME type, an output MIME type, a format conversion filter that can produce the output from the input type, and virtual costs associated with this conversion. One example line reads like this:

```
application/pdf          application/postscript   33   pdftops
```

This means that the *pdftops* filter will take *application/pdf* as input and produce *application/postscript* as output; the virtual cost of this

operation is 33 CUPS-$. The next filter is more expensive, costing 66 CUPS-$:

```
application/vnd.hp-HPGL application/postscript   66   hpgltops
```

This is the *hpgltops*, which processes HP-GL plotter files to PostScript.

```
application/octet-stream
```

Here are two more examples:

```
application/x-shell     application/postscript   33    texttops
text/plain              application/postscript   33    texttops
```

The last two examples name the *texttops* filter to work on *text/plain* as well as on *application/x-shell*. (Hint: This differentiation is needed for the syntax highlighting feature of *texttops*).

22.5.3 Filtering Overview

There are many more combinations named in `mime.convs`. However, you are not limited to use the ones predefined there. You can plug in any filter you like to the CUPS framework. It must meet, or must be made to meet, some minimal requirements. If you find (or write) a cool conversion filter of some kind, make sure it complies with what CUPS needs and put in the right lines in `mime.types` and `mime.convs`; then it will work seamlessly inside CUPS.

22.5.3.1 Filter Requirements

The "CUPS requirements" for filters are simple. Take filenames or `stdin` as input and write to `stdout`. They should take these arguments:

printer The name of the printer queue (normally this is the name of the filter being run).

job The numeric job ID for the job being printed.

user The string from the originating-user-name attribute.

title The string from the job-name attribute.

copies The numeric value from the number-copies attribute.

options The job options.

filename (optionally) The print request file (if missing, filters expect data fed through `stdin`). In most cases, it is easy to write a simple wrapper script around existing filters to make them work with CUPS.

22.5.4 Prefilters

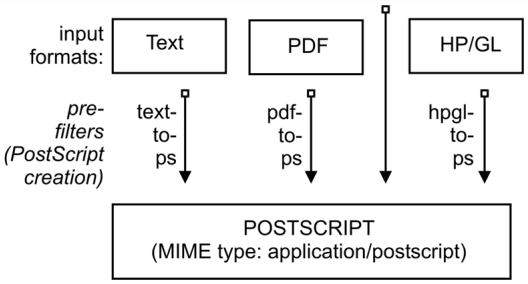
As previously stated, PostScript is the central file format to any UNIX-based printing system. From PostScript, CUPS generates raster data to feed non-PostScript printers.

But what happens if you send one of the supported non-PS formats to print? Then CUPS runs "prefilters" on these input formats to generate PostScript first. There are prefilters to create PostScript from ASCII text, PDF, DVI, or HP-GL. The outcome of these filters is always of MIME type *`application/postscript`* (meaning that any device-specific print options are not yet embedded into the PostScript by CUPS and that the next filter to be called is pstops). Another prefilter is running on all supported image formats, the *`imagetops`* filter. Its outcome is always of MIME type *`application/vnd.cups-postscript`* (not application/postscript), meaning it has the print options already embedded into the file. This is shown in Figure 22.4.

22.5.5 pstops

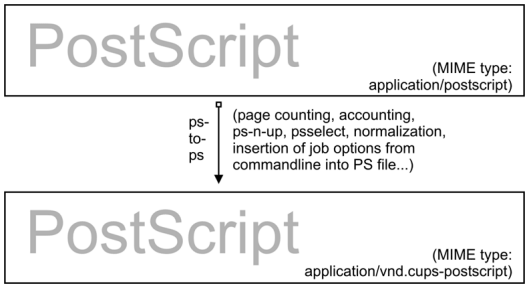
pstops is a filter that is used to convert *`application/postscript`* to *`application/vnd.`*

Figure 22.4 Prefiltering in CUPS to Form PostScript.



cups-postscript. As stated earlier, this filter inserts all device-specific print options (commands to the printer to ask for the duplexing of output, or stapling and punching it, and so on) into the PostScript file. An example is illustrated in Figure 22.5.

Figure 22.5 Adding Device-Specific Print Options.



This is not all. Other tasks performed by it are:

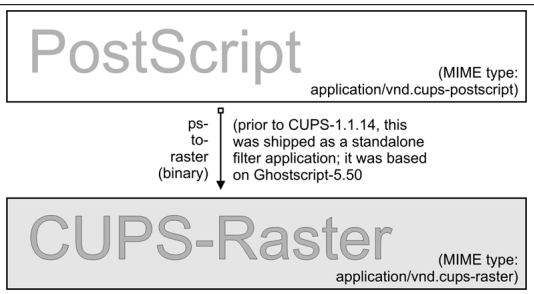
- Selecting the range of pages to be printed (e.g., you can choose to print only pages "3, 6, 8-11, 16, and 19-21", or only odd-numbered pages).
- Putting two or more logical pages on one sheet of paper (the so-called "number-up" function).
- Counting the pages of the job to insert the accounting information into the `/var/log/cups/page_log`.

22.5.6 pstoraster

pstoraster is at the core of the CUPS filtering system. It is responsible for the first stage of the rasterization process. Its input is of MIME type

application/vnd.cups-postscript; its output is application/vnd.cups-raster. This output format is not yet meant to be printable. Its aim is to serve as a general-purpose input format for more specialized *raster drivers* that are able to generate device-specific printer data. This is shown in Figure 22.6.

Figure 22.6 PostScript to Intermediate Raster Format.

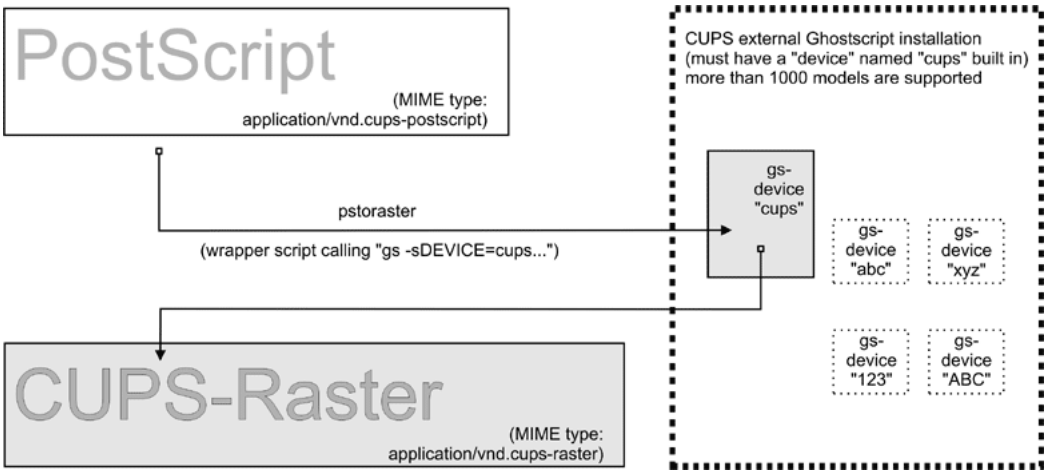


CUPS raster is a generic raster format with powerful features. It is able to include per-page information, color profiles, and more, to be used by the downstream raster drivers. Its MIME type is registered with IANA and its specification is, of course, completely open. It is designed to make it quite easy and inexpensive for manufacturers to develop Linux and UNIX raster drivers for their printer models should they choose to do so. CUPS always takes care of the first stage of rasterization so these vendors do not need to care about Ghostscript complications (in fact, there are currently more than one vendor financing the development of CUPS raster drivers). This is illustrated in Figure 22.7.

CUPS versions before version 1.1.15 shipped a binary (or source code) standalone filter, named *pstoraster*. *pstoraster*, which was derived from GNU Ghostscript 5.50 and could be installed instead of and in addition to any GNU or AFPL Ghostscript package without conflicting.

Since version 1.1.15, this feature has changed. The functions for this filter have been integrated back into Ghostscript (now based on GNU Ghostscript version 7.05). The *pstoraster* filter is now a simple shell script calling **gs** with the **-sDEVICE=cups** parameter. If your Ghostscript fails when this command is executed: **gs -h |grep cups**, you might not be able to print, update your Ghostscript.

Figure 22.7 CUPS-Raster Production Using Ghostscript.



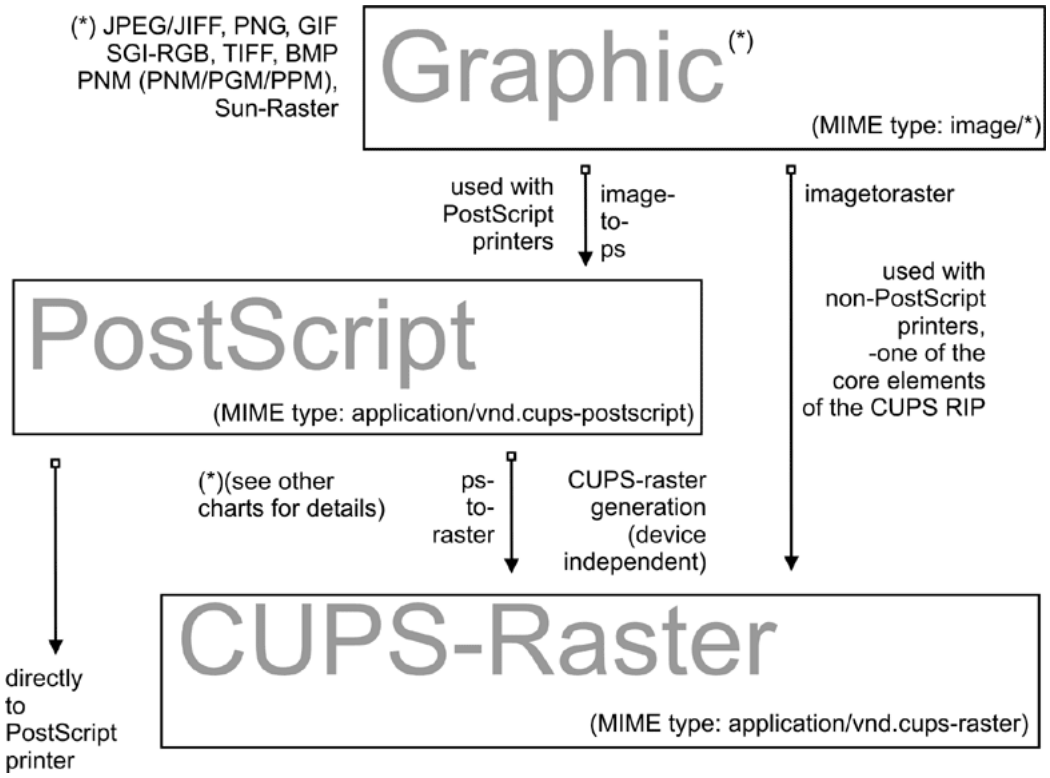
22.5.7 imagetops and imagetoraster

In the section about prefilters, we mentioned the prefilter that generates PostScript from image formats. The *imagetoraster* filter is used to convert directly from image to raster, without the intermediate PostScript stage. It is used more often than the previously mentioned prefilters. We summarize in a flowchart the image file filtering in Figure 22.8.

22.5.8 rasterto [printers specific]

CUPS ships with quite a variety of raster drivers for processing CUPS raster. On my system, I find in /usr/lib/cups/filter/ the following: *rastertoalps*, *rastertobj*, *rastertoepson*, *rastertoescp*, *rastertopcl*, *rastertoturboprint*, *rastertoapdk*, *rastertodymo*, *rastertoescp*, *rastertohp*, and *rastertoprinter*. Don't worry if you have fewer drivers than this; some of these are installed by commercial add-ons to CUPS (like *rastertoturboprint*), and others (like *rastertoprinter*) by third-party driver development projects (such as Gutenprint) wanting to cooperate as closely as possible with CUPS. See Figure 22.9.

Figure 22.8 Image Format to CUPS-Raster Format Conversion.

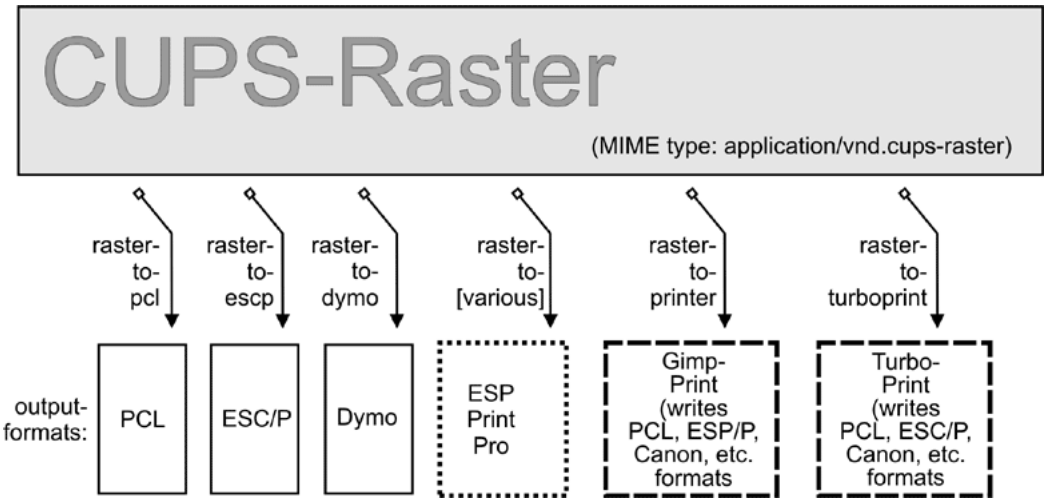


22.5.9 CUPS Backends

The last part of any CUPS filtering chain is a backend. Backends are special programs that send the print-ready file to the final device. There is a separate backend program for any transfer protocol for sending print jobs over the network, and one for every local interface. Every CUPS print queue needs to have a CUPS "device-URI" associated with it. The device URI is the way to encode the backend used to send the job to its destination. Network device-URIs use two slashes in their syntax, local device URIs only one, as you can see from the following list. Keep in mind that local interface names may vary greatly from my examples, if your OS is not Linux:

usb This backend sends print files to USB-connected printers. An example for the CUPS device-URI to use is `usb:/dev/usb/lp0`.

Figure 22.9 Raster to Printer-Specific Formats.



serial This backend sends print files to serially connected printers. An example for the CUPS device-URI to use is `serial:/dev/ttyS0?baud=11500`.

parallel This backend sends print files to printers connected to the parallel port. An example for the CUPS device-URI to use is `parallel:/dev/lp0`.

SCSI This backend sends print files to printers attached to the SCSI interface. An example for the CUPS device-URI to use is `scsi:/dev/sr1`.

lpd This backend sends print files to LPR/LPD-connected network printers. An example for the CUPS device-URI to use is `lpd://remote_host_name/remote_queue_name`.

AppSocket/HP JetDirect This backend sends print files to AppSocket (a.k.a., HP JetDirect) connected network printers. An example for the CUPS device-URI to use is `socket://10.11.12.13:9100`.

ipp This backend sends print files to IPP-connected network printers (or to other CUPS servers). Examples for CUPS device-URIs to use are `ipp:://192.193.194.195/ipp` (for many HP printers) and `ipp://remote_cups_server/printers/remote_printer_name`.

http This backend sends print files to HTTP-connected printers. (The http:// CUPS backend is only a symlink to the ipp:// backend.) Examples for the CUPS device-URIs to use are `http:://192.193.194.195:631/ipp` (for many HP printers) and `http://remote_cups_server:631/printers/remote_printer_name`.

smb This backend sends print files to printers shared by a Windows host. Examples of CUPS device-URIs that may be used includes:

```
smb://workgroup/server/printersharename
smb://server/printersharename
smb://username:password@workgroup/server/printersharename
smb://username:password@server/printersharename
```

The smb:// backend is a symlink to the Samba utility *smbspool* (does not ship with CUPS). If the symlink is not present in your CUPS backend directory, have your root user create it: **ln -s 'which smbspool' /usr/lib/cups/backend/smb**.

It is easy to write your own backends as shell or Perl scripts if you need any modification or extension to the CUPS print system. One reason could be that you want to create "special" printers that send the print jobs as email (through a "mailto:/" backend), convert them to PDF (through a "pdfgen:/" backend) or dump them to "/dev/null". (In fact, I have the systemwide default printer set up to be connected to a devnull:/ backend: there are just too many people sending jobs without specifying a printer, and scripts and programs that do not name a printer. The systemwide default deletes the job and sends a polite email back to the $USER asking him or her to always specify the correct printer name.)

Not all of the mentioned backends may be present on your system or usable (depending on your hardware configuration). One test for all available CUPS backends is provided by the *lpinfo* utility. Used with the `-v` parameter, it lists all available backends:

```
$ lpinfo -v
```

22.5.10 The Role of cupsomatic/foomatic

cupsomatic filters may be the most widely used on CUPS installations. You must be clear that these were not developed by the CUPS people. They are a third-party add-on to CUPS. They utilize the traditional Ghostscript devices to render jobs for CUPS. When troubleshooting, you should know about the difference. Here the whole rendering process is done in one stage, inside Ghostscript, using an appropriate device for the target printer. *cupsomatic* uses PPDs that are generated from the Foomatic Printer & Driver Database at Linuxprinting.org.

You can recognize these PPDs from the line calling the *cupsomatic* filter:

```
*cupsFilter: "application/vnd.cups-postscript  0  cupsomatic"
```

You may find this line among the first 40 or so lines of the PPD file. If you have such a PPD installed, the printer shows up in the CUPS Web interface with a *foomatic* namepart for the driver description. *cupsomatic* is a Perl script that runs Ghostscript with all the complicated command line options autoconstructed from the selected PPD and command line options given to the print job.

However, *cupsomatic* is now deprecated. Its PPDs (especially the first generation of them, still in heavy use out there) are not meeting the Adobe specifications. You might also suffer difficulties when you try to download them with "Point'n'Print" to Windows clients. A better and more powerful successor is now available: it is called *foomatic-rip*. To use *foomatic-rip* as a filter with CUPS, you need the new type of PPDs, which have a similar but different line:

```
*cupsFilter: "application/vnd.cups-postscript  0  foomatic-rip"
```

The PPD-generating engine at Linuxprinting.org has been revamped. The new PPDs comply with the Adobe spec. They also provide a new way to

specify different quality levels (hi-res photo, normal color, grayscale, and draft) with a single click, whereas before you could have required five or more different selections (media type, resolution, inktype, and dithering algorithm). There is support for custom-size media built in. There is support to switch print options from page to page in the middle of a job. And the best thing is that the new `foomatic-rip` works seamlessly with all legacy spoolers too (like LPRng, BSD-LPD, PDQ, PPR, and so on), providing for them access to use PPDs for their printing.

22.5.11 The Complete Picture

If you want to see an overview of all the filters and how they relate to each other, the complete picture of the puzzle is at the end of this chapter.

22.5.12 mime.convs

CUPS autoconstructs all possible filtering chain paths for any given MIME type and every printer installed. But how does it decide in favor of or against a specific alternative? (There may be cases where there is a choice of two or more possible filtering chains for the same target printer.) Simple. You may have noticed the figures in the third column of the mime.convs file. They represent virtual costs assigned to this filter. Every possible filtering chain will sum up to a total "filter cost." CUPS decides for the most "inexpensive" route.

Tip

Setting *`FilterLimit 1000`* in `cupsd.conf` will not allow more filters to run concurrently than will consume a total of 1000 virtual filter cost. This is an efficient way to limit the load of any CUPS server by setting an appropriate "FilterLimit" value. A FilterLimit of 200 allows roughly one job at a time, while a FilterLimit of 1000 allows approximately five jobs maximum at a time.

22.5.13 "Raw" Printing

You can tell CUPS to print (nearly) any file "raw". "Raw" means it will not be filtered. CUPS will send the file to the printer "as is" without bothering if the printer is able to digest it. Users need to take care themselves that they send sensible data formats only. Raw printing can happen on any queue if the "*-o raw*" option is specified on the command line. You can also set up raw-only queues by simply not associating any PPD with it. This command:

```
$ lpadmin -P rawprinter -v socket://11.12.13.14:9100 -E
```

sets up a queue named "rawprinter", connected via the "socket" protocol (a.k.a. "HP JetDirect") to the device at IP address 11.12.1.3.14, using port 9100. (If you had added a PPD with **-P /path/to/PPD** to this command line, you would have installed a "normal" print queue.)

CUPS will automatically treat each job sent to a queue as a "raw" one if it can't find a PPD associated with the queue. However, CUPS will only send known MIME types (as defined in its own mime.types file) and refuse others.

22.5.14 application/octet-stream Printing

Any MIME type with no rule in the `/etc/cups/mime.types` file is regarded as unknown or *application/octet-stream* and will not be sent. Because CUPS refuses to print unknown MIME types by default, you will probably have experienced that print jobs originating from Windows clients were not printed. You may have found an error message in your CUPS logs like:

```
Unable to convert file 0 to printable format for job
```

To enable the printing of *application/octet-stream* files, edit these two files:

- `/etc/cups/mime.convs`
- `/etc/cups/mime.types`

Both contain entries (at the end of the respective files) that must be uncommented to allow raw mode operation for *application/octet-stream*. In `/etc/cups/mime.types` make sure this line is present:

```
application/octet-stream
```

This line (with no specific autotyping rule set) makes all files not otherwise auto-typed a member of *application/octet-stream*. In `/etc/cups/mime.convs`, have this line:

```
application/octet-stream   application/vnd.cups-raw   0   -
```

This line tells CUPS to use the *Null Filter* (denoted as "-", doing nothing at all) on *application/octet-stream*, and tag the result as *application/vnd.cups-raw*. This last one is always a green light to the CUPS scheduler to now hand the file over to the backend connecting to the printer and sending it over.

NOTE

Editing the `mime.convs` and the `mime.types` file does not *enforce* "raw" printing, it only *allows* it.

Background. That CUPS is a more security-aware printing system than traditional ones does not by default allow one to send deliberate (possibly binary) data to printing devices. (This could be easily abused to launch a Denial of Service attack on your printer(s), causing at least the loss of a lot of paper and ink.) "Unknown" data are regarded by CUPS as *MIME type application/octet-stream*. While you *can* send data "raw", the MIME type for these must be one that is known to CUPS and allowed by it. The file `/etc/cups/mime.types` defines the "rules" of how CUPS recognizes MIME types. The file `/etc/cups/mime.convs` decides which file conversion filter(s) may be applied to which MIME types.

22.5.15 PostScript Printer Descriptions for Non-PostScript Printers

Originally PPDs were meant to be used for PostScript printers only. Here, they help to send device-specific commands and settings to the RIP, which processes the job file. CUPS has extended this scope for PPDs to cover non-PostScript printers too. This was not difficult, because it is a standardized file format. In a way it was logical too: CUPS handles PostScript and uses a PostScript RIP (Ghostscript) to process the job files. The only difference is that a PostScript printer has the RIP built-in, for other types of printers the Ghostscript RIP runs on the host computer.

PPDs for a non-PostScript printer have a few lines that are unique to CUPS. The most important one looks similar to this:

```
*cupsFilter: application/vnd.cups-raster  66   rastertoprinter
```

It is the last piece in the CUPS filtering puzzle. This line tells the CUPS daemon to use as a last filter *rastertoprinter*. This filter should be served as input an *application/vnd.cups-raster* MIME type file. Therefore, CUPS should autoconstruct a filtering chain, which delivers as its last output the specified MIME type. This is then taken as input to the specified *rastertoprinter* filter. After the last filter has done its work (*rastertoprinter* is a Gutenprint filter), the file should go to the backend, which sends it to the output device.

CUPS by default ships only a few generic PPDs, but they are good for several hundred printer models. You may not be able to control different paper trays, or you may get larger margins than your specific model supports. See Table 21.1Table 22.1 for summary information.

22.5.16 *cupsomatic/foomatic-rip* Versus *Native CUPS* Printing

Native CUPS rasterization works in two steps:

- First is the *pstoraster* step. It uses the special CUPS device from ESP Ghostscript 7.05.x as its tool.

Table 22.1 PPDs Shipped with CUPS

PPD file	Printer type
deskjet.ppd	older HP inkjet printers and compatible
deskjet2.ppd	newer HP inkjet printers and compatible
dymo.ppd	label printers
epson9.ppd	Epson 24-pin impact printers and compatible
epson24.ppd	Epson 24-pin impact printers and compatible
okidata9.ppd	Okidata 9-pin impact printers and compatible
okidat24.ppd	Okidata 24-pin impact printers and compatible
stcolor.ppd	older Epson Stylus Color printers
stcolor2.ppd	newer Epson Stylus Color printers
stphoto.ppd	older Epson Stylus Photo printers
stphoto2.ppd	newer Epson Stylus Photo printers
laserjet.ppd	all PCL printers

- Second is the *rasterdriver* step. It uses various device-specific filters; there are several vendors who provide good quality filters for this step. Some are free software, some are shareware, and some are proprietary.

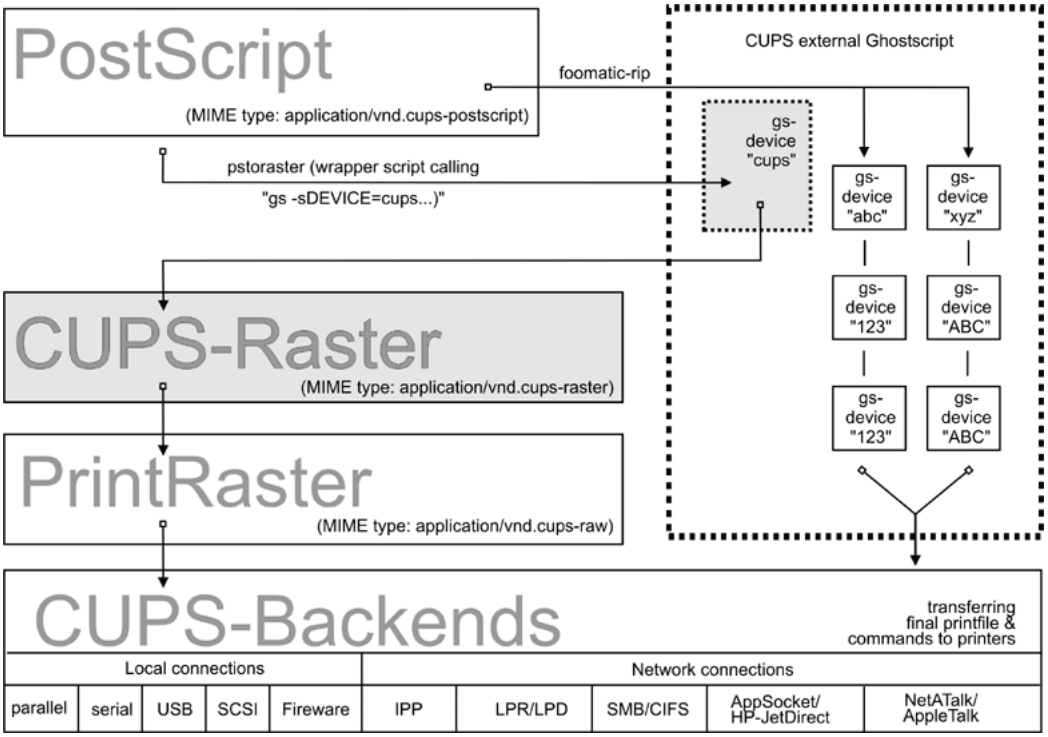
Often this produces better quality (and has several more advantages) than other methods. This is shown in Figure 22.10.

One other method is the *cupsomatic/foomatic-rip* way. Note that *cupsomatic* is *not* made by the CUPS developers. It is an independent contribution to printing development, made by people from Linuxprinting.org.[6] *cupsomatic* is no longer developed, maintained, or supported. It now been replaced by *foomatic-rip*. *foomatic-rip* is a complete rewrite of the old *cupsomatic* idea, but very much improved and generalized to other (non-CUPS) spoolers. An upgrade to *foomatic-rip* is strongly advised, especially if you are upgrading to a recent version of CUPS, too.

Like the old *cupsomatic* method, the *foomatic-rip* (new) method from Linuxprinting.org uses the traditional Ghostscript print file processing, doing everything in a single step. It therefore relies on all the other devices built into Ghostscript. The quality is as good (or bad) as Ghostscript rendering is in other spoolers. The advantage is that this method supports many printer models not supported (yet) by the more modern CUPS method.

[6] See also <http://www.cups.org/cups-help.html>

Figure 22.10 cupsomatic/foomatic Processing Versus Native CUPS.



Of course, you can use both methods side by side on one system (and even for one printer, if you set up different queues) and find out which works best for you.

cupsomatic kidnaps the print file after the *application/vnd.cups-postscript* stage and deviates it through the CUPS-external, systemwide Ghostscript installation. Therefore, the print file bypasses the *pstoraster* filter (and also bypasses the CUPS raster drivers *rastertosomething*). After Ghostscript finished its rasterization, *cupsomatic* hands the rendered file directly to the CUPS backend. Figure 22.10, illustrates the difference between native CUPS rendering and the *Foomatic/cupsomatic* method.

22.5.17 Examples for Filtering Chains

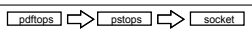
Here are a few examples of commonly occurring filtering chains to illustrate the workings of CUPS.

Assume you want to print a PDF file to an HP JetDirect-connected PostScript printer, but you want to print pages 3-5, 7, and 11-13 only, and you want to print them "two-up" and "duplex":

- Your print options (page selection as required, two-up, duplex) are passed to CUPS on the command line.
- The (complete) PDF file is sent to CUPS and autotyped as *application/pdf*.
- The file therefore must first pass the *pdftops* prefilter, which produces PostScript MIME type *application/postscript* (a preview here would still show all pages of the original PDF).
- The file then passes the *pstops* filter that applies the command line options: it selects pages 2-5, 7, and 11-13, creates the imposed layout "two pages on one sheet", and inserts the correct "duplex" command (as defined in the printer's PPD) into the new PostScript file; the file is now of PostScript MIME type *application/vnd.cups-postscript*.
- The file goes to the *socket* backend, which transfers the job to the printers.

The resulting filter chain, therefore, is as shown in Figure 22.11.

Figure 22.11 PDF to Socket Chain.



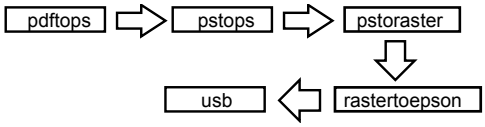
Assume you want to print the same filter to an USB-connected Epson Stylus Photo Printer installed with the CUPS `stphoto2.ppd`. The first few filtering stages are nearly the same:

- Your print options (page selection as required, two-up, duplex) are passed to CUPS on the command line.
- The (complete) PDF file is sent to CUPS and autotyped as *application/pdf*.

- The file must first pass the *pdftops* prefilter, which produces PostScript MIME type *application/postscript* (a preview here would still show all pages of the original PDF).
- The file then passes the "pstops" filter that applies the command line options: it selects the pages 2-5, 7, and 11-13, creates the imposed layout "two pages on one sheet," and inserts the correct "duplex" command (oops — this printer and PPD do not support duplex printing at all, so this option will be ignored) into the new PostScript file; the file is now of PostScript MIME type *application/vnd.cups-postscript*.
- The file then passes the *pstoraster* stage and becomes MIME type *application/cups-raster*.
- Finally, the *rastertoepson* filter does its work (as indicated in the printer's PPD), creating the printer-specific raster data and embedding any user-selected print options into the print data stream.
- The file goes to the *usb* backend, which transfers the job to the printers.

The resulting filter chain therefore is as shown in Figure 22.12.

Figure 22.12 PDF to USB Chain.



22.5.18 Sources of CUPS Drivers/PPDs

On the Internet you can now find many thousands of CUPS-PPD files (with their companion filters), in many national languages supporting more than 1,000 non-PostScript models.

- ESP PrintPro[7] (commercial, non-free) is packaged with more than 3,000 PPDs, ready for successful use "out of the box" on Linux, Mac OS X, IBM-AIX, HP-UX, Sun-Solaris, SGI-IRIX, Compaq Tru64, Digital UNIX, and other commercial Unices (it is written by the CUPS developers themselves and its sales help finance the further development of CUPS, as they feed their creators).

[7] <http://www.easysw.com/printpro/>

- The Gutenprint Project[8] (GPL, free software) provides around 140 PPDs (supporting nearly 400 printers, many driven to photo quality output), to be used alongside the Gutenprint CUPS filters.
- TurboPrint[9] (shareware, non-free) supports roughly the same number of printers in excellent quality.
- OMNI[10] (LPGL, free) is a package made by IBM, now containing support for more than 400 printers, stemming from the inheritance of IBM OS/2 know-how ported over to Linux (CUPS support is in a beta stage at present).
- HPIJS[11] (BSD-style licenses, free) supports approximately 150 of HP's own printers and also provides excellent print quality now (currently available only via the Foomatic path).
- Foomatic/cupsomatic[12] (LPGL, free) from Linuxprinting.org provide PPDs for practically every Ghostscript filter known to the world (including Omni, Gutenprint, and HPIJS).

22.5.19 Printing with Interface Scripts

CUPS also supports the use of "interface scripts" as known from System V AT&T printing systems. These are often used for PCL printers, from applications that generate PCL print jobs. Interface scripts are specific to printer models. They have a role similar to PPDs for PostScript printers. Interface scripts may inject the Escape sequences as required into the print data stream if the user, for example, selects a certain paper tray, or changes paper orientation, or uses A3 paper. Interface scripts are practically unknown in the Linux realm. On HP-UX platforms they are more often used. You can use any working interface script on CUPS too. Just install the printer with the **-i** option:

```
root# lpadmin -p pclprinter -v socket://11.12.13.14:9100 \
          -i /path/to/interface-script
```

[8] <http://gimp-print.sourceforge.net/>
[9] <http://www.turboprint.de/english.html/>
[10] <http://www-124.ibm.com/developerworks/oss/linux/projects/omni/>
[11] <http://hpinkjet.sourceforge.net/>
[12] <http://www.linuxprinting.org/>

Interface scripts might be the "unknown animal" to many. However, with CUPS they provide the easiest way to plug in your own custom-written filtering script or program into one specific print queue (some information about the traditional use of interface scripts is found at http://playground.sun.com/printing/documentation/interfa

22.6 Network Printing (Purely Windows)

Network printing covers a lot of ground. To understand what exactly goes on with Samba when it is printing on behalf of its Windows clients, let's first look at a "purely Windows" setup: Windows clients with a Windows NT print server.

22.6.1 From Windows Clients to an NT Print Server

Windows clients printing to an NT-based print server have two options. They may:

- Execute the driver locally and render the GDI output (EMF) into the printer-specific format on their own.
- Send the GDI output (EMF) to the server, where the driver is executed to render the printer-specific output.

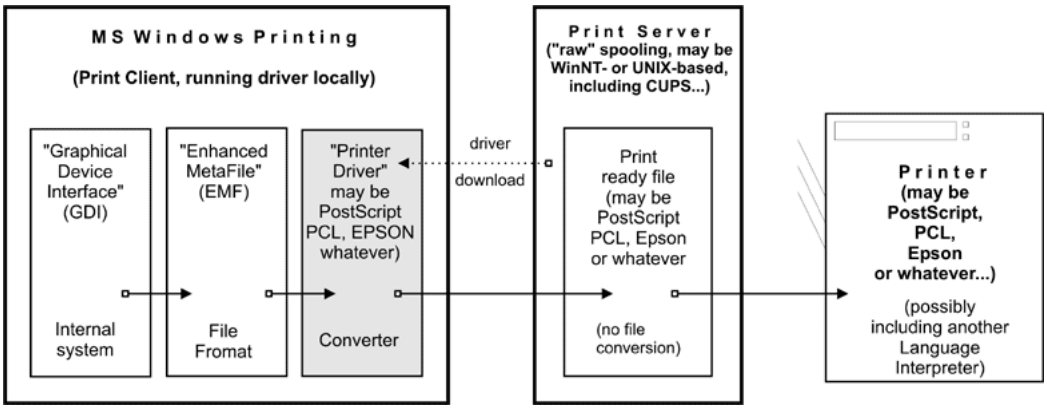
Both print paths are shown in the flowcharts in Figure 22.13, and Figure 22.14.

22.6.2 Driver Execution on the Client

In the first case, the print server must spool the file as raw, meaning it shouldn't touch the job file and try to convert it in any way. This is what a traditional UNIX-based print server can do too, and at a better performance and more reliably than an NT print server. This is what most Samba administrators probably are familiar with. One advantage of this setup is that this "spooling-only" print server may be used even if no driver(s) for UNIX is available. It is sufficient to have the Windows client drivers available and installed on the clients. This is illustrated in Figure 22.13.

[13] <http://playground.sun.com/printing/documentation/interface.html>

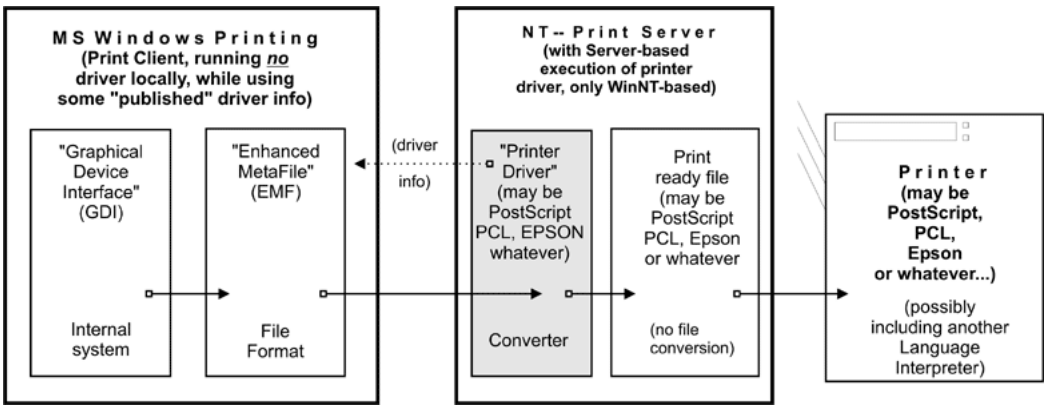
Figure 22.13 Print Driver Execution on the Client.



22.6.3 Driver Execution on the Server

The other path executes the printer driver on the server. The client transfers print files in EMF format to the server. The server uses the PostScript, PCL, ESC/P, or other driver to convert the EMF file into the printer-specific language. It is not possible for UNIX to do the same. Currently, there is no program or method to convert a Windows client's GDI output on a UNIX server into something a printer could understand. This is illustrated in Figure 22.14.

Figure 22.14 Print Driver Execution on the Server.



However, something similar is possible with CUPS, so read on.

22.7 Network Printing (Windows Clients and UNIX/Samba Print Servers)

Since UNIX print servers *cannot* execute the Win32 program code on their platform, the picture is somewhat different. However, this does not limit your options all that much. On the contrary, you may have a way here to implement printing features that are not possible otherwise.

22.7.1 From Windows Clients to a CUPS/Samba Print Server

Here is a simple recipe showing how you can take advantage of CUPS's powerful features for the benefit of your Windows network printing clients:

- Let the Windows clients send PostScript to the CUPS server.
- Let the CUPS server render the PostScript into device-specific raster format.

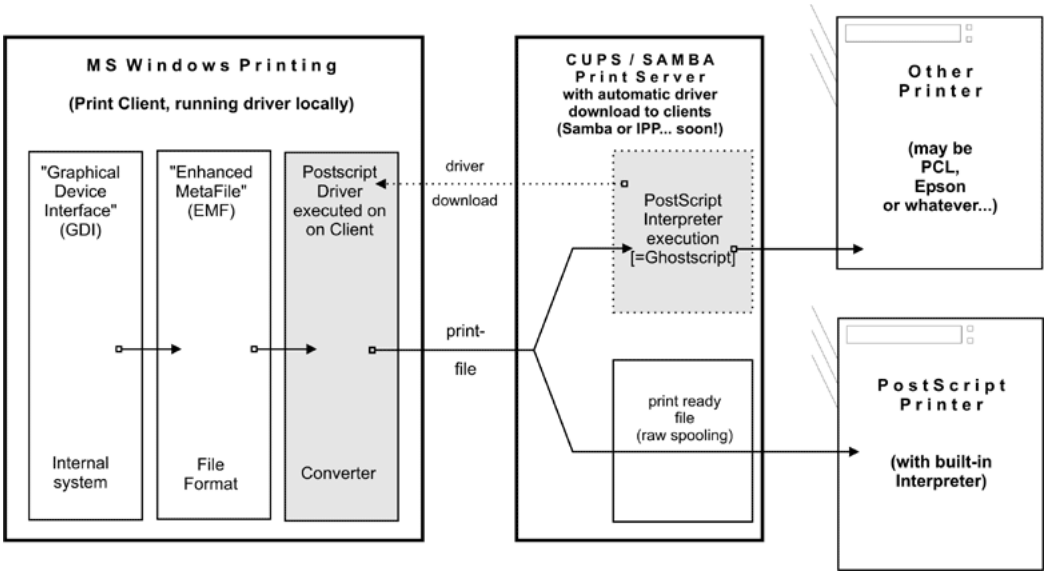
This requires the clients to use a PostScript driver (even if the printer is a non-PostScript model. It also requires that you have a driver on the CUPS server.

First, to enable CUPS-based printing through Samba, the following options should be set in your `smb.conf` file *[global]* section:

```
        printing = cups
        printcap = cups
```

When these parameters are specified, all manually set print directives (like *print command* or *lppause command*) in `smb.conf` (as well as in Samba itself) will be ignored. Instead, Samba will directly interface with CUPS through its application program interface (API), as long as Samba has been compiled with CUPS library (libcups) support. If Samba has not been compiled with CUPS support, and if no other print commands are set up, then printing will use the *System V* AT&T command set, with the -oraw option automatically passing through (if you want your own defined print commands to work with a Samba server that has CUPS support compiled in, simply use *classicalprinting* = sysv). This is illustrated in Figure 22.15.

Figure 22.15 Printing via CUPS/Samba Server.



22.7.2 Samba Receiving Job-Files and Passing Them to CUPS

Samba *must* use its own spool directory (it is set by a line similar to *path* = /var/spool/samba, in the *[printers]* or *[printername]* section of `smb.conf`). Samba receives the job in its own spool space and passes it into the spool directory of CUPS (the CUPS spool directory is set by the *`RequestRoot`* directive in a line that defaults to *`RequestRoot /var/spool/cups`*). CUPS checks the access rights of its spool directory and resets it to healthy values with every restart. We have seen quite a few people who used a common spooling space for Samba and CUPS, and struggled for weeks with this "problem."

A Windows user authenticates only to Samba (by whatever means is configured). If Samba runs on the same host as CUPS, you only need to allow "localhost" to print. If it runs on different machines, you need to make sure the Samba host gets access to printing on CUPS.

22.8 Network PostScript RIP

This section discusses the use of CUPS filters on the server — configuration where clients make use of a PostScript driver with CUPS-PPDs.

PPDs can control all print device options. They are usually provided by the manufacturer — if you own a PostScript printer, that is. PPD files are always a component of PostScript printer drivers on MS Windows or Apple Mac OS systems. They are ASCII files containing user-selectable print options, mapped to appropriate PostScript, PCL, or PJL commands for the target printer. Printer driver GUI dialogs translate these options "on the fly" into buttons and drop-down lists for the user to select.

CUPS can load, without any conversions, the PPD file from any Windows (NT is recommended) PostScript driver and handle the options. There is a Web browser interface to the print options (select <`http://localhost:631/printers/`> and click on one **Configure Printer** button to see it) or a command line interface (see **man lpoptions** or see if you have **lphelp** on your system). There are also some different GUI front-ends on Linux/UNIX, which can present PPD options to users. PPD options are normally meant to be evaluated by the PostScript RIP on the real PostScript printer.

22.8.1 PPDs for Non-PS Printers on UNIX

CUPS does not limit itself to "real" PostScript printers in its use of PPDs. The CUPS developers have extended the scope of the PPD concept to also describe available device and driver options for non-PostScript printers through CUPS-PPDs.

This is logical, because CUPS includes a fully featured PostScript interpreter (RIP). This RIP is based on Ghostscript. It can process all received PostScript (and additionally many other file formats) from clients. All CUPS-PPDs geared to non-PostScript printers contain an additional line, starting with the keyword *`*cupsFilter`*. This line tells the CUPS print system which printer-specific filter to use for the interpretation of the supplied PostScript. Thus CUPS lets all its printers appear as PostScript devices to its clients, because it can act as a PostScript RIP for those printers, processing the received PostScript code into a proper raster print format.

22.8.2 PPDs for Non-PS Printers on Windows

CUPS-PPDs can also be used on Windows clients, on top of a "core" PostScript driver (now recommended is the CUPS PostScript Driver for Windows NT/200x/XP; you can also use the Adobe one, with limitations). This feature enables CUPS to do a few tricks no other spooler can do:

- Act as a networked PostScript RIP handling print files from all client platforms in a uniform way.
- Act as a central accounting and billing server, since all files are passed through the pstops filter and are therefore logged in the CUPS `page_log` file. *Note:* this cannot happen with "raw" print jobs, which always remain unfiltered per definition.
- Enable clients to consolidate on a single PostScript driver, even for many different target printers.

Using CUPS PPDs on Windows clients enables them to control all print job settings just as a UNIX client can do.

22.9 Windows Terminal Servers (WTS) as CUPS Clients

This setup may be of special interest to people experiencing major problems in WTS environments. WTS often need a multitude of non-PostScript drivers installed to run their clients' variety of different printer models. This often imposes the price of much increased instability.

22.9.1 Printer Drivers Running in "Kernel Mode" Cause Many Problems

Windows NT printer drivers, which run in "kernel mode", introduce a high risk for the stability of the system if the driver is not really stable and well-tested. And there are a lot of bad drivers out there! Especially notorious is the example of the PCL printer driver that had an additional sound module running to notify users via soundcard of their finished jobs. Do I need to say that this one was also reliably causing "blue screens of death" on a regular basis?

PostScript drivers are generally well-tested. They are not known to cause any problems, even though they also run in kernel mode. This might be

because until now there have been only two different PostScript drivers: the one from Adobe and the one from Microsoft. Both are well-tested and are as stable as you can imagine on Windows. The CUPS driver is derived from the Microsoft one.

22.9.2 Workarounds Impose Heavy Limitations

In an attempt to work around problems, site administrators have resorted to restricting the allowed drivers installed on their WTS to one generic PCL and one PostScript driver. This, however, restricts the number of printer options available for clients to use. Often they can't get out more than simplex prints from one standard paper tray, while their devices could do much better if driven by a different driver!

22.9.3 CUPS: A "Magical Stone"?

Using a PostScript driver, enabled with a CUPS-PPD, seems to be a very elegant way to overcome all these shortcomings. There are, depending on the version of Windows OS you use, up to three different PostScript drivers now available: Adobe, Microsoft, and CUPS PostScript drivers. None of them is known to cause major stability problems on WTS (even if used with many different PPDs). The clients will be able to (again) choose paper trays, duplex printing, and other settings. However, there is a certain price for this too: a CUPS server acting as a PostScript RIP for its clients requires more CPU and RAM than when just acting as a "raw spooling" device. Plus, this setup is not yet widely tested, although the first feedbacks look very promising.

22.9.4 PostScript Drivers with No Major Problems, Even in Kernel Mode

More recent printer drivers on W200x and XP no longer run in kernel mode (unlike Windows NT). However, both operating systems can still use the NT drivers, running in kernel mode (you can roughly tell which is which as the drivers in subdirectory "2" of "W32X86" are "old" ones). As was said before, the Adobe as well as the Microsoft PostScript drivers are not known to cause any stability problems. The CUPS driver is derived from the Microsoft one. There is a simple reason for this: the MS DDK (Device

Development Kit) for Windows NT (which used to be available at no cost to licensees of Visual Studio) includes the source code of the Microsoft driver, and licensees of Visual Studio are allowed to use and modify it for their own driver development efforts. This is what the CUPS people have done. The license does not allow them to publish the whole of the source code. However, they have released the "diff" under the GPL, and if you are the owner of an "MS DDK for Windows NT," you can check the driver yourself.

22.10 Configuring CUPS for Driver Download

As we have said before, all previously known methods to prepare client printer drivers on the Samba server for download and Point'n'Print convenience of Windows workstations are working with CUPS, too. These methods were described in Chapter 21, "Classical Printing Support". In reality, this is a pure Samba business and relates only to the Samba-Windows client relationship.

22.10.1 *cupsaddsmb*: The Unknown Utility

The ***cupsaddsmb*** utility (shipped with all current CUPS versions) is an alternative method to transfer printer drivers into the Samba *[print$]* share. Remember, this share is where clients expect drivers deposited and set up for download and installation. It makes the sharing of any (or all) installed CUPS printers quite easy. **cupsaddsmb** can use the Adobe PostScript driver as well as the newly developed CUPS PostScript driver for Windows NT/200x/XP. ***cupsaddsmb*** does *not* work with arbitrary vendor printer drivers, but only with the *exact* driver files that are named in its man page.

The CUPS printer driver is available from the CUPS download site. Its package name is `cups-samba-[version].tar.gz`. It is preferred over the Adobe drivers because it has a number of advantages:

- It supports a much more accurate page accounting.
- It supports banner pages and page labels on all printers.
- It supports the setting of a number of job IPP attributes (such as job priority, page label, and job billing).

However, currently only Windows NT, 2000, and XP are supported by the CUPS drivers. You will also need to get the respective part of the Adobe driver if you need to support Windows 95, 98, and Me clients.

22.10.2 Prepare Your smb.conf for cupsaddsmb

Prior to running **cupsaddsmb**, you need the settings in `smb.conf` as shown in Example 22.10.1.

Example 22.10.1 smb.conf for cupsaddsmb Usage

```
[global]
        load printers = yes
        printing = cups
        printcap name = cups
 [printers]
        comment = All Printers
        path = /var/spool/samba
        browseable = no
 # setting depends on your requirements
        guest ok = yes
        writable = no
        printable = yes
        printer admin = root
 [print$]
        comment = Printer Drivers
        path = /etc/samba/drivers
        browseable = yes
        guest ok = no
        read only = yes
        write list = root, @smbprintadm
```

22.10.3 CUPS "PostScript Driver for Windows NT/200x/XP"

CUPS users may get the exact same package from <`http://www.cups.org/software.html`>. It is a separate package from the CUPS-based software files, tagged as CUPS 1.1.x Windows NT/200x/XP Printer Driver for Samba (tar.gz, 192k). The filename to download is `cups-samba-1.1.x.tar.gz`. Upon untar and unzipping, it will reveal these files:

```
root# tar xvzf cups-samba-1.1.19.tar.gz
cups-samba.install
cups-samba.license
cups-samba.readme
cups-samba.remove
cups-samba.ss
```

These have been packaged with the ESP meta-packager software EPM. The `*.install` and `*.remove` files are simple shell scripts, which untar the `*.ss` (the `*.ss` is nothing else but a tar archive, which can be untarred by "tar" too). Then it puts the content into `/usr/share/cups/drivers/`. This content includes three files:

```
root# tar tv cups-samba.ss
cupsdrvr.dll
cupsui.dll
cups.hlp
```

The *`cups-samba.install`* shell scripts are easy to handle:

```
root# ./cups-samba.install
[....]
Installing software...
Updating file permissions...
Running post-install commands...
Installation is complete.
```

The script should automatically put the driver files into the `/usr/share/cups/drivers/` directory:

```
root# cp /usr/share/drivers/cups.hlp /usr/share/cups/drivers/
```

WARNING

Due to a bug, one recent CUPS release puts the `cups.hlp` driver file into`/usr/share/drivers/` instead of `/usr/share/cups/drivers/`. To work around this, copy/move the file (after running the **./cups-samba.install** script) manually to the correct place.

This new CUPS PostScript driver is currently binary only, but free of charge. No complete source code is provided (yet). The reason is that it has been developed with the help of the Microsoft DDK and compiled with Microsoft Visual Studio 6. Driver developers are not allowed to distribute the whole of the source code as free software. However, CUPS developers released the "diff" in source code under the GPL, so anybody with a license for Visual Studio and a DDK will be able to compile for himself or herself.

22.10.4 Recognizing Different Driver Files

The CUPS drivers do not support the older Windows 95/98/Me, but only the Windows NT/2000/XP client.

Windows NT, 2000, and XP are supported by:

- cups.hlp
- cupsdrvr.dll
- cupsui.dll

Adobe drivers are available for the older Windows 95/98/Me as well as for Windows NT/2000/XP clients. The set of files is different from the different platforms.

Windows 95, 98, and ME are supported by:

- ADFONTS.MFM
- ADOBEPS4.DRV
- ADOBEPS4.HLP
- DEFPRTR2.PPD

- ICONLIB.DLL
- PSMON.DLL

Windows NT, 2000, and XP are supported by:

- ADOBEPS5.DLL
- ADOBEPSU.DLL
- ADOBEPSU.HLP

NOTE

If both the Adobe driver files and the CUPS driver files for the support of Windows NT/200x/XP are presently installed on the server, the Adobe files will be ignored and the CUPS files will be used. If you prefer — for whatever reason — to use Adobe-only drivers, move away the three CUPS driver files. The Windows 9x/Me clients use the Adobe drivers in any case.

22.10.5 Acquiring the Adobe Driver Files

Acquiring the Adobe driver files seems to be unexpectedly difficult for many users. They are not available on the Adobe Web site as single files, and the self-extracting and/or self-installing Windows-.exe is not easy to locate either. You probably need to use the included native installer and run the installation process on one client once. This will install the drivers (and one generic PostScript printer) locally on the client. When they are installed, share the generic PostScript printer. After this, the client's *[print$]* share holds the Adobe files, which you can get with smbclient from the CUPS host.

22.10.6 ESP Print Pro PostScript Driver for Windows NT/200x/XP

Users of the ESP Print Pro software are able to install the ESP print drivers package as an alternative to the Adobe PostScript drivers. To do so, retrieve

the driver files from the normal download area of the ESP Print Pro software at Easy Software[14] web site. You need to locate the link labeled "SAMBA" among the **Download Printer Drivers for ESP Print Pro 4.x** area and download the package. Once installed, you can prepare any driver by simply highlighting the printer in the Printer Manager GUI and selecting **Export Driver...** from the menu. Of course, you need to have prepared Samba beforehand to handle the driver files; that is, set up the *[print$]* share, and so on. The ESP Print Pro package includes the CUPS driver files as well as a (licensed) set of Adobe drivers for the Windows 95/98/Me client family.

22.10.7 Caveats to Be Considered

Once you have run the install script (and possibly manually moved the `cups.hlp` file to `/usr/share/cups/drivers/`), the driver is ready to be put into Samba's *[print$]* share (which often maps to `/etc/samba/drivers/` and contains a subdirectory tree with *WIN40* and *W32X86* branches). You do this by running **cupsaddsmb** (see also **man cupsaddsmb** for CUPS since release 1.1.16).

> **Tip**
>
> You may need to put root into the smbpasswd file by running **smbpasswd**; this is especially important if you should run this whole procedure for the first time and are not working in an environment where everything is configured for *single sign-on* to a Windows Domain Controller.

Once the driver files are in the *[print$]* share and are initialized, they are ready to be downloaded and installed by the Windows NT/200x/XP clients.

[14] <http://www.easysw.com/software.html>

NOTE

Win 9x/Me clients will not work with the CUPS PostScript driver. For these you still need to use the `ADOBE*.*` drivers, as previously stated.

NOTE

It is not harmful if you still have the `ADOBE*.*` driver files from previous installations in the `/usr/share/cups/drivers/` directory. The new **cupsaddsmb** (from 1.1.16) will automatically prefer its own drivers if it finds both.

NOTE

Should your Windows clients have had the old `ADOBE*.*` files for the Adobe PostScript driver installed, the download and installation of the new CUPS PostScript driver for Windows NT/200x/XP will fail at first. You need to wipe the old driver from the clients first. It is not enough to "delete" the printer, because the driver files will still be kept by the clients and re-used if you try to re-install the printer. To really get rid of the Adobe driver files on the clients, open the **Printers** folder (possibly via **Start** -> **Settings** -> **Control Panel** -> **Printers**), right-click on the folder background, and select **Server Properties**. When the new dialog opens, select the **Drivers** tab. On the list select the driver you want to delete and click the **Delete** button. This will only work if there is not one single printer left that uses that particular driver. You need to "delete" all printers using this driver in the **Printers** folder first. You will need Administrator privileges to do this.

NOTE

Once you have successfully downloaded the CUPS PostScript driver to a client, you can easily switch all printers to this one by proceeding as described in Chapter 21, "Classical Printing Support". Either change a driver for an existing printer by running the **Printer Properties** dialog, or use **rpcclient** with the **setdriver** subcommand.

22.10.8 Windows CUPS PostScript Driver Versus Adobe Driver

Are you interested in a comparison between the CUPS and the Adobe PostScript drivers? For our purposes, these are the most important items that weigh in favor of CUPS:

- No hassle with the Adobe EULA.
- No hassle with the question, "Where do I get the ADOBE*.* driver files?"
- The Adobe drivers (on request of the printer PPD associated with them) often put a PJL header in front of the main PostScript part of the print file. Thus, the print file starts with *<1B >%-12345X* or *<escape>%-12345X* instead of *%!PS*. This leads to the CUPS daemon autotyping the incoming file as a print-ready file, not initiating a pass through the *pstops* filter (to speak more technically, it is not regarded as the generic MIME-type *application/postscript*, but as the more special MIME type *application/cups.vnd-postscript*), which therefore also leads to the page accounting in */var/log/cups/-page_log* not receiving the exact number of pages; instead the dummy page number of "1" is logged in a standard setup).
- The Adobe driver has more options to misconfigure the PostScript generated by it (like setting it inadvertently to **Optimize for Speed** instead of **Optimize for Portability**, which could lead to CUPS being unable to process it).
- The CUPS PostScript driver output sent by Windows clients to the CUPS server is guaranteed to autotype as the generic MIME type *application/postscript*, thus passing through the CUPS *pstops* filter and logging the correct number of pages in the `page_log` for accounting and quota purposes.
- The CUPS PostScript driver supports the sending of additional standard (IPP) print options by Windows NT/200x/XP clients. Such additional print options are naming the CUPS standard *banner pages* (or the custom ones, should they be installed at the time of driver download), using the CUPS page-label option, setting a job priority, and setting the scheduled time of printing (with the option to support additional useful IPP job attributes in the future).

- The CUPS PostScript driver supports the inclusion of the new *`*cupsJobTicket`* comments at the beginning of the PostScript file (which could be used in the future for all sorts of beneficial extensions on the CUPS side, but which will not disturb any other applications because they will regard it as a comment and simply ignore it).
- The CUPS PostScript driver will be the heart of the fully fledged CUPS IPP client for Windows NT/200x/XP to be released soon (probably alongside the first beta release for CUPS 1.2).

22.10.9 Run cupsaddsmb (Quiet Mode)

The **cupsaddsmb** command copies the needed files into your *[print$]* share. Additionally, the PPD associated with this printer is copied from `/etc/cups/ppd/` to *[print$]*. There the files wait for convenient Windows client installations via Point'n'Print. Before we can run the command successfully, we need to be sure that we can authenticate toward Samba. If you have a small network, you are probably using user-level security (*security* = user).

Here is an example of a successfully run **cupsaddsmb** command:

```
root# cupsaddsmb -U root infotec_IS2027
Password for root required to access localhost via Samba: ['secret']
```

To share *all* printers and drivers, use the `-a` parameter instead of a printer name. Since **cupsaddsmb** "exports" the printer drivers to Samba, it should be obvious that it only works for queues with a CUPS driver associated.

22.10.10 Run cupsaddsmb with Verbose Output

Probably you want to see what's going on. Use the `-v` parameter to get a more verbose output. The output below was edited for better readability: all "\" at the end of a line indicate that I inserted an artificial line break plus some indentation here:

```
root# cupsaddsmb -U root -v infotec_2105
Password for root required to access localhost via GANDALF:
Running command: smbclient //localhost/print\$ -N -U'root%secret' \
```

```
    -c 'mkdir W32X86; \
    put /var/spool/cups/tmp/3e98bf2d333b5 W32X86/infotec_2105.ppd; \
   put /usr/share/cups/drivers/cupsdrvr.dll W32X86/cupsdrvr.dll; \
    put /usr/share/cups/drivers/cupsui.dll W32X86/cupsui.dll; \
    put /usr/share/cups/drivers/cups.hlp W32X86/cups.hlp'
added interface ip=10.160.51.60 bcast=10.160.51.255 nmask=255.255.252.0
Domain=[CUPS-PRINT] OS=[UNIX] Server=[Samba 2.2.7a]
NT_STATUS_OBJECT_NAME_COLLISION making remote directory \W32X86
putting file /var/spool/cups/tmp/3e98bf2d333b5 as \W32X86/infotec_2105.ppd
putting file /usr/share/cups/drivers/cupsdrvr.dll as \W32X86/cupsdrvr.dll
putting file /usr/share/cups/drivers/cupsui.dll as \W32X86/cupsui.dll
putting file /usr/share/cups/drivers/cups.hlp as \W32X86/cups.hlp

Running command: rpcclient localhost -N -U'root%secret'
   -c 'adddriver "Windows NT x86"   \
   "infotec_2105:cupsdrvr.dll:infotec_2105.ppd:cupsui.dll:cups.hlp:NULL: \
    RAW:NULL"'
cmd = adddriver "Windows NT x86" \
   "infotec_2105:cupsdrvr.dll:infotec_2105.ppd:cupsui.dll:cups.hlp:NULL: \
   RAW:NULL"
Printer Driver infotec_2105 successfully installed.

Running command: smbclient //localhost/print\$ -N -U'root%secret' \
-c 'mkdir WIN40; \
    put /var/spool/cups/tmp/3e98bf2d333b5 WIN40/infotec_2105.PPD; \
   put /usr/share/cups/drivers/ADFONTS.MFM WIN40/ADFONTS.MFM;   \
    put /usr/share/cups/drivers/ADOBEPS4.DRV WIN40/ADOBEPS4.DRV; \
    put /usr/share/cups/drivers/ADOBEPS4.HLP WIN40/ADOBEPS4.HLP; \
    put /usr/share/cups/drivers/DEFPRTR2.PPD WIN40/DEFPRTR2.PPD; \
   put /usr/share/cups/drivers/ICONLIB.DLL WIN40/ICONLIB.DLL; \
   put /usr/share/cups/drivers/PSMON.DLL WIN40/PSMON.DLL;'
  added interface ip=10.160.51.60 bcast=10.160.51.255 nmask=255.255.252.0
  Domain=[CUPS-PRINT] OS=[UNIX] Server=[Samba 2.2.7a]
  NT_STATUS_OBJECT_NAME_COLLISION making remote directory \WIN40
  putting file /var/spool/cups/tmp/3e98bf2d333b5 as \WIN40/infotec_2105.PPD
  putting file /usr/share/cups/drivers/ADFONTS.MFM as \WIN40/ADFONTS.MFM
  putting file /usr/share/cups/drivers/ADOBEPS4.DRV as \WIN40/ADOBEPS4.DRV
  putting file /usr/share/cups/drivers/ADOBEPS4.HLP as \WIN40/ADOBEPS4.HLP
  putting file /usr/share/cups/drivers/DEFPRTR2.PPD as \WIN40/DEFPRTR2.PPD
  putting file /usr/share/cups/drivers/ICONLIB.DLL as \WIN40/ICONLIB.DLL
```

```
putting file /usr/share/cups/drivers/PSMON.DLL as \WIN40/PSMON.DLL

Running command: rpcclient localhost -N -U'root%secret' \
 -c 'adddriver "Windows 4.0"      \
 "infotec_2105:ADOBEPS4.DRV:infotec_2105.PPD:NULL:ADOBEPS4.HLP: \
 PSMON.DLL:RAW:ADOBEPS4.DRV,infotec_2105.PPD,ADOBEPS4.HLP,PSMON.DLL, \
  ADFONTS.MFM,DEFPRTR2.PPD,ICONLIB.DLL"'
 cmd = adddriver "Windows 4.0" "infotec_2105:ADOBEPS4.DRV:\
 infotec_2105.PPD:NULL:ADOBEPS4.HLP:PSMON.DLL:RAW:ADOBEPS4.DRV,\
 infotec_2105.PPD,ADOBEPS4.HLP,PSMON.DLL,ADFONTS.MFM,DEFPRTR2.PPD,\
 ICONLIB.DLL"
Printer Driver infotec_2105 successfully installed.

Running command: rpcclient localhost -N -U'root%secret'  \
 -c 'setdriver infotec_2105 infotec_2105'
cmd = setdriver infotec_2105 infotec_2105
Successfully set infotec_2105 to driver infotec_2105.
```

WARNING

You will see the root password for the Samba account printed on screen.

If you look closely, you'll discover your root password was transferred unencrypted over the wire, so beware! Also, if you look further, you may discover error messages like NT_STATUS_OBJECT_NAME_COLLISION in the output. This will occur when the directories WIN40 and W32X86 already existed in the *[print$]* driver download share (from a previous driver installation). These are harmless warning messages.

22.10.11 Understanding cupsaddsmb

What has happened? What did **cupsaddsmb** do? There are five stages of the procedure:

1. Call the CUPS server via IPP and request the driver files and the PPD file for the named printer.
2. Store the files temporarily in the local TEMPDIR (as defined in `cupsd.conf`).
3. Connect via smbclient to the Samba server's *[print$]* share and put the files into the share's WIN40 (for Windows 9x/Me) and W32X86 (for Windows NT/200x/XP) subdirectories.
4. Connect via rpcclient to the Samba server and execute the **adddriver** command with the correct parameters.
5. Connect via rpcclient to the Samba server a second time and execute the **setdriver** command.

NOTE

You can run the **cupsaddsmb** utility with parameters to specify one remote host as Samba host and a second remote host as CUPS host. Especially if you want to get a deeper understanding, it is a good idea to try it and see more clearly what is going on (though in real life most people will have their CUPS and Samba servers run on the same host):

```
root# cupsaddsmb -H sambaserver -h cupsserver -v printer
```

22.10.12 How to Recognize If cupsaddsmb Completed Successfully

You *must* always check if the utility completed successfully in all fields. You need at minimum these three messages among the output:

1. *Printer Driver infotec_2105 successfully installed.* # (for the W32X86 == Windows NT/200x/XP architecture).

2. *Printer Driver infotec_2105 successfully installed.* # (for the WIN40 == Windows 9x/Me architecture).
3. *Successfully set [printerXPZ] to driver [printerXYZ].*

These messages are probably not easily recognized in the general output. If you run **cupsaddsmb** with the `-a` parameter (which tries to prepare *all* active CUPS printer drivers for download), you might miss if individual printer drivers had problems installing properly. A redirection of the output will help you analyze the results in retrospective.

If you get:

```
SetPrinter call failed!
result was WERR_ACCESS_DENIED
```

it means that you might have set *use client driver* = yes for this printer. Setting it to "no" will solve the problem. Refer to the `smb.conf` man page for explanation of the *`use client driver`*.

> **Note**
>
> It is impossible to see any diagnostic output if you do not run **cupsaddsmb** in verbose mode. Therefore, we strongly recommend against use of the default quiet mode. It will hide any problems from you that might occur.

22.10.13 cupsaddsmb with a Samba PDC

Can't get the standard **cupsaddsmb** command to run on a Samba PDC? Are you asked for the password credential again and again, and the command just will not take off at all? Try one of these variations:

```
root# cupsaddsmb -U MIDEARTH\\root -v printername
root# cupsaddsmb -H SAURON -U MIDEARTH\\root -v printername
```

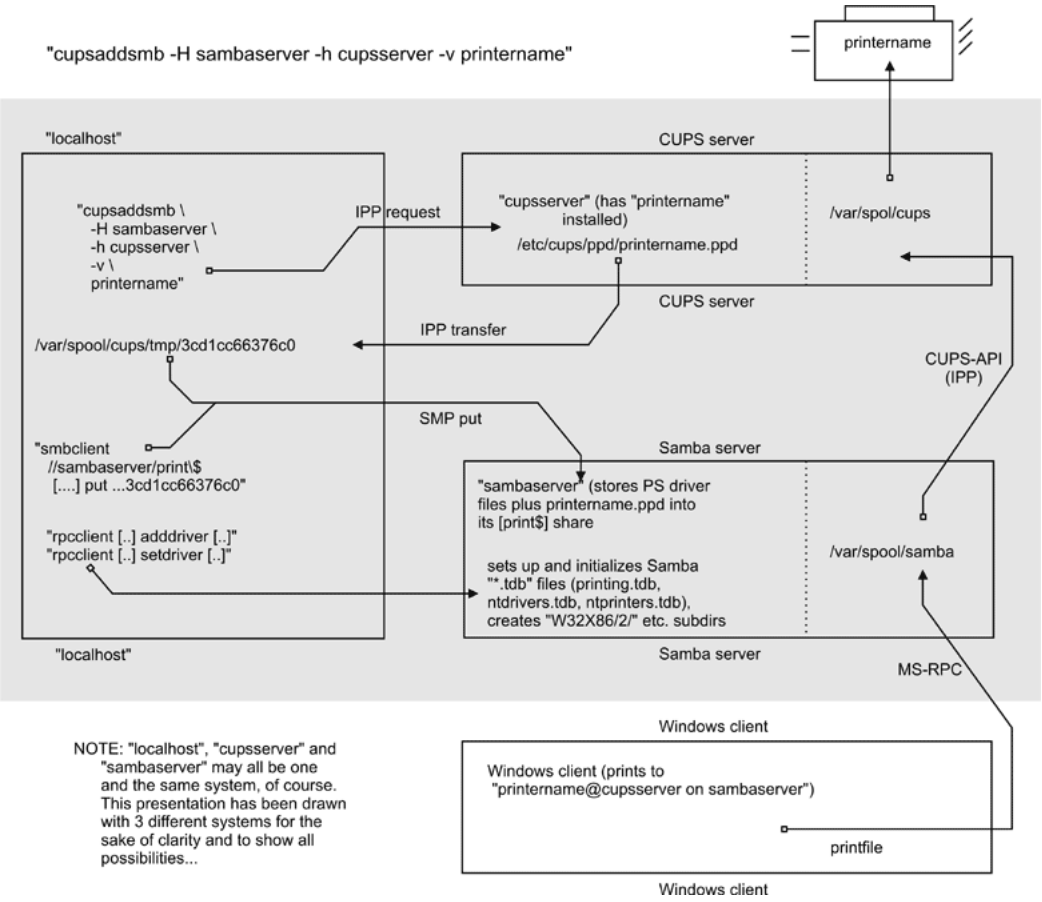
```
root# cupsaddsmb -H SAURON -U MIDEARTH\\root -h cups-server -v printername
```

(Note the two backslashes: the first one is required to "escape" the second one).

22.10.14 cupsaddsmb Flowchart

Figure 22.16 shows a chart about the procedures, command flows, and data flows of the **cupaddsmb** command. Note again: cupsaddsmb is not intended to, and does not work with, raw print queues!

Figure 22.16 cupsaddsmb Flowchart.



22.10.15 Installing the PostScript Driver on a Client

After **cupsaddsmb** is completed, your driver is prepared for the clients to use. Here are the steps you must perform to download and install it via Point'n'Print. From a Windows client, browse to the CUPS/Samba server:

- Open the **Printers** share of Samba in Network Neighborhood.
- Right-click on the printer in question.
- From the opening context menu select **Install...** or **Connect...** (depending on the Windows version you use).

After a few seconds, there should be a new printer in your client's *local* **Printers** folder. On Windows XP it will follow a naming convention of *PrinterName on SambaServer*. (In my current case it is infotec_2105 on kdebitshop). If you want to test it and send your first job from an application like Microsoft Word, the new printer appears in a `\\SambaServer\PrinterName` entry in the drop-down list of available printers.

cupsaddsmb will only reliably work with CUPS version 1.1.15 or higher and with Samba version 2.2.4, or later. If it does not work, or if the automatic printer driver download to the clients does not succeed, you can still manually install the CUPS printer PPD on top of the Adobe PostScript driver on clients. Then point the client's printer queue to the Samba printer share for a UNC type of connection:

```
C:\> net use lpt1: \\sambaserver\printershare /user:ntadmin
```

should you desire to use the CUPS networked PostScript RIP functions. (Note that user "ntadmin" needs to be a valid Samba user with the required privileges to access the printershare.) This sets up the printer connection in the traditional LanMan way (not using MS-RPC).

22.10.16 Avoiding Critical PostScript Driver Settings on the Client

Printing works, but there are still problems. Most jobs print well, some do not print at all. Some jobs have problems with fonts, which do not look very good. Some jobs print fast and some are dead-slow. Many of these problems can be greatly reduced or even completely eliminated if you follow a few guidelines. Remember, if your print device is not PostScript-enabled,

you are treating your Ghostscript installation on your CUPS host with the output your client driver settings produce. Treat it well:

- Avoid the PostScript Output Option: Optimize for Speed setting. Use the Optimize for Portability instead (Adobe PostScript driver).
- Don't use the Page Independence: NO setting. Instead, use Page Independence: YES (CUPS PostScript Driver).
- Recommended is the True Type Font Downloading Option: Native True Type over Automatic and Outline; you should by all means avoid Bitmap (Adobe PostScript Driver).
- Choose True Type Font: Download as Softfont into Printer over the default Replace by Device Font (for exotic fonts, you may need to change it back to get a printout at all; Adobe).
- Sometimes you can choose PostScript Language Level: in case of problems try 2 instead of 3 (the latest ESP Ghostscript package handles Level 3 PostScript very well; Adobe).
- Say Yes to PostScript Error Handler (Adobe).

22.11 Installing PostScript Driver Files Manually Using rpcclient

Of course, you can run all the commands that are embedded into the cupsaddsmb convenience utility yourself, one by one, and upload and prepare the driver files for future client downloads.

1. Prepare Samba (a CUPS print queue with the name of the printer should be there. We are providing the driver now).
2. Copy all files to *[print$]*.
3. Run **rpcclient adddriver** (for each client architecture you want to support).
4. Run **rpcclient setdriver.**

We are going to do this now. First, read the man page on *rpcclient* to get a first idea. Look at all the printing-related subcommands: **enumprinters**, **enumdrivers**, **enumports**, **adddriver**, and **setdriver** are among the

most interesting ones. *rpcclient* implements an important part of the MS-RPC protocol. You can use it to query (and command) a Windows NT (or 200x/XP) PC, too. MS-RPC is used by Windows clients, among other things, to benefit from the Point'n'Print features. Samba can now mimic this as well.

22.11.1 A Check of the rpcclient man Page

First let's check the *rpcclient* man page. Here are two relevant passages:

adddriver <arch> <config> Execute an **AddPrinterDriver()** RPC to install the printer driver information on the server. The driver files should already exist in the directory returned by **getdriverdir**. Possible values for *arch* are the same as those for the **getdriverdir** command. The *config* parameter is defined as follows:

```
Long Printer Name:\
Driver File Name:\
Data File Name:\
Config File Name:\
Help File Name:\
Language Monitor Name:\
Default Data Type:\
Comma Separated list of Files
```

Any empty fields should be entered as the string "NULL".

Samba does not need to support the concept of print monitors, since these only apply to local printers whose drivers can use a bidirectional link for communication. This field should be "NULL". On a remote NT print server, the print monitor for a driver must already be installed before adding the driver or else the RPC will fail.

setdriver <printername> <drivername> Execute a **SetPrinter()** command to update the printer driver associated with an installed printer. The printer driver must already be correctly installed on the print server.

See also the **enumprinters** and **enumdrivers** commands to obtain a list of installed printers and drivers.

22.11.2 Understanding the rpcclient man Page

The *exact* format isn't made too clear by the man page, since you have to deal with some parameters containing spaces. Here is a better description for it. We have line-broken the command and indicated the breaks with "\". Usually you would type the command in one line without the line breaks:

```
adddriver "Architecture" \
   "LongPrinterName:DriverFile:DataFile:ConfigFile:HelpFile:\
   LanguageMonitorFile:DataType:ListOfFiles,Comma-separated"
```

What the man pages denote as a simple <*config*> keyword in reality consists of eight colon-separated fields. The last field may take multiple (in some very insane cases, even 20 different additional) files. This might sound confusing at first. What the man pages call the "LongPrinterName" in reality should be called the "Driver Name". You can name it anything you want, as long as you use this name later in the **rpcclient ... setdriver** command. For practical reasons, many name the driver the same as the printer.

It isn't simple at all. I hear you asking: "How do I know which files are Driver File", "Data File", "Config File", "Help File" and "Language Monitor File in each case?" For an answer, you may want to have a look at how a Windows NT box with a shared printer presents the files to us. Remember that this whole procedure has to be developed by the Samba Team by listening to the traffic caused by Windows computers on the wire. We may as well turn to a Windows box now and access it from a UNIX workstation. We will query it with **rpcclient** to see what it tells us and try to understand the man page more clearly.

22.11.3 Producing an Example by Querying a Windows Box

We could run **rpcclient** with a **getdriver** or a **getprinter** subcommand (in level 3 verbosity) against it. Just sit down at a UNIX or Linux workstation with the Samba utilities installed, then type the following command:

```
root# rpcclient -U'user%secret' NT-SERVER -c 'getdriver printername 3'
```

From the result it should become clear which is which. Here is an example from my installation:

```
root# rpcclient -U'Danka%xxxx' W200xSERVER \
    -c'getdriver "DANKA InfoStream Virtual Printer" 3'
    cmd = getdriver "DANKA InfoStream Virtual Printer" 3

 [Windows NT x86]
 Printer Driver Info 3:
         Version: [2]
         Driver Name: [DANKA InfoStream]
         Architecture: [Windows NT x86]
         Driver Path: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\PSCRIPT.DLL]
         Datafile: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\INFOSTRM.PPD]
         Configfile: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\PSCRPTUI.DLL]
         Helpfile: [C:\WINNT\System32\spool\DRIVERS\W32X86\2\PSCRIPT.HLP]

         Dependentfiles: []
         Dependentfiles: []
         Dependentfiles: []
         Dependentfiles: []
         Dependentfiles: []
         Dependentfiles: []
         Dependentfiles: []

         Monitorname: []
         Defaultdatatype: []
```

Some printer drivers list additional files under the label *Dependentfiles*, and these would go into the last field *ListOfFiles,Comma-separated*. For the CUPS PostScript drivers, we do not need any (nor would we for the Adobe PostScript driver); therefore, the field will get a "NULL" entry.

22.11.4 Requirements for adddriver and setdriver to Succeed

From the man page (and from the quoted output of **cupsaddsmb** above) it becomes clear that you need to have certain conditions in order to make the manual uploading and initializing of the driver files succeed. The two

rpcclient subcommands (**adddriver** and **setdriver**) need to encounter the following preconditions to complete successfully:

- You are connected as *printer admin* or root (this is *not* the "Printer Operators" group in NT, but the *printer admin* group as defined in the *[global]* section of `smb.conf`).
- Copy all required driver files to `\\SAMBA\print$\w32x86` and `\\SAMBA\print$\win40` as appropriate. They will end up in the "0" respective "2" subdirectories later. For now, *do not* put them there; they'll be automatically used by the **adddriver** subcommand. (If you use **smbclient** to put the driver files into the share, note that you need to escape the "$": **smbclient //sambaserver/print\$ -U root.**)
- The user you're connecting as must be able to write to the *[print$]* share and create subdirectories.
- The printer you are going to set up for the Windows clients needs to be installed in CUPS already.
- The CUPS printer must be known to Samba; otherwise the **setdriver** subcommand fails with an NT_STATUS_UNSUCCESSFUL error. To check if the printer is known by Samba, you may use the **enumprinters** subcommand to **rpcclient**. A long-standing bug prevented a proper update of the printer list until every smbd process had received a SIGHUP or was restarted. Remember this in case you've created the CUPS printer just recently and encounter problems: try restarting Samba.

22.11.5 Manual Driver Installation in 15 Steps

We are going to install a printer driver now by manually executing all required commands. Because this may seem a rather complicated process at first, we go through the procedure step by step, explaining every single action item as it comes up. Manual Driver Installation

1. **Install the printer on CUPS.**

```
root# lpadmin -p mysmbtstprn -v socket://10.160.51.131:9100 -E \
        -P canonIR85.ppd
```

This installs a printer with the name *mysmbtstprn* to the CUPS system. The printer is accessed via a socket (a.k.a. JetDirect or Direct TCP/IP) connection. You need to be root for this step.

2. **(Optional.) Check if the printer is recognized by Samba.**

```
root# rpcclient -Uroot%xxxx -c 'enumprinters' localhost \
  | grep -C2 mysmbtstprn
flags:[0x800000]
name:[\\kde-bitshop\mysmbtstprn]
description:[\\kde-bitshop\mysmbtstprn,,mysmbtstprn]
comment:[mysmbtstprn]
```

This should show the printer in the list. If not, stop and restart the Samba daemon (smbd) or send a HUP signal:

```
root# kill -HUP `pidof smbd`
```

Check again. Troubleshoot and repeat until successful. Note the "empty" field between the two commas in the "description" line. The driver name would appear here if there was one already. You need to know root's Samba password (as set by the **smbpasswd** command) for this step and most of the following steps. Alternatively, you can authenticate as one of the users from the "write list" as defined in `smb.conf` for *[print$]*.

3. **(Optional.) Check if Samba knows a driver for the printer.**

```
root# rpcclient -Uroot%xxxx -c 'getprinter mysmbtstprn 2'\
 localhost | grep driver

drivername:[]

root# rpcclient -Uroot%xxxx -c 'getprinter mysmbtstprn 2' \
 localhost | grep -C4 driv

servername:[\\kde-bitshop]
printername:[\\kde-bitshop\mysmbtstprn]
```

```
sharename:[mysmbtstprn]
portname:[Samba Printer Port]
drivername:[]
comment:[mysmbtstprn]
location:[]
sepfile:[]
printprocessor:[winprint]

root# rpcclient -U root%xxxx -c 'getdriver mysmbtstprn' localhost
 result was WERR_UNKNOWN_PRINTER_DRIVER
```

None of the three commands shown above should show a driver. This step was done for the purpose of demonstrating this condition. An attempt to connect to the printer at this stage will prompt a message along the lines of, "The server does not have the required printer driver installed."

4. **Put all required driver files into Samba's [print$].**

```
root# smbclient //localhost/print\$ -U 'root%xxxx' \
   -c 'cd W32X86; \
   put /etc/cups/ppd/mysmbtstprn.ppd mysmbtstprn.PPD; \
   put /usr/share/cups/drivers/cupsui.dll cupsui.dll; \
   put /usr/share/cups/drivers/cupsdrvr.dll cupsdrvr.dll; \
   put /usr/share/cups/drivers/cups.hlp cups.hlp'
```

(This command should be entered in one long single line. Line breaks and the line ends indicated by "\" have been inserted for readability reasons.) This step is *required* for the next one to succeed. It makes the driver files physically present in the *[print$]* share. However, clients would still not be able to install them, because Samba does not yet treat them as driver files. A client asking for the driver would still be presented with a "not installed here" message.

5. **Verify where the driver files are now.**

```
root# ls -l /etc/samba/drivers/W32X86/
total 669
drwxr-sr-x    2 root     ntadmin       532 May 25 23:08 2
```

```
drwxr-sr-x    2 root     ntadmin       670 May 16 03:15 3
-rwxr--r--    1 root     ntadmin     14234 May 25 23:21 cups.hlp
-rwxr--r--    1 root     ntadmin    278380 May 25 23:21 cupsdrvr.dll
-rwxr--r--    1 root     ntadmin    215848 May 25 23:21 cupsui.dll
-rwxr--r--    1 root     ntadmin    169458 May 25 23:21 mysmbtstprn.PPD
```

The driver files now are in the W32X86 architecture "root" of *[print$]*.

6. **Tell Samba that these are driver files (adddriver).**

```
root# rpcclient -Uroot%xxxx -c 'adddriver "Windows NT x86" \
   "mydrivername:cupsdrvr.dll:mysmbtstprn.PPD: \
  cupsui.dll:cups.hlp:NULL:RAW:NULL"' \
  localhost
Printer Driver mydrivername successfully installed.
```

You cannot repeat this step if it fails. It could fail even as a result of a simple typo. It will most likely have moved a part of the driver files into the "2" subdirectory. If this step fails, you need to go back to the fourth step and repeat it before you can try this one again. In this step, you need to choose a name for your driver. It is normally a good idea to use the same name as is used for the printer name; however, in big installations you may use this driver for a number of printers that obviously have different names, so the name of the driver is not fixed.

7. **Verify where the driver files are now.**

```
root# ls -l /etc/samba/drivers/W32X86/
total 1
drwxr-sr-x    2 root     ntadmin       532 May 25 23:22 2
drwxr-sr-x    2 root     ntadmin       670 May 16 03:15 3

root# ls -l /etc/samba/drivers/W32X86/2
total 5039
[....]
-rwxr--r--    1 root     ntadmin     14234 May 25 23:21 cups.hlp
-rwxr--r--    1 root     ntadmin    278380 May 13 13:53 cupsdrvr.dll
-rwxr--r--    1 root     ntadmin    215848 May 13 13:53 cupsui.dll
```

```
-rwxr--r--    1 root     ntadmin    169458 May 25 23:21 mysmbtstprn.PPD
```

Notice how step 6 also moved the driver files to the appropriate subdirectory. Compare this with the situation after step 5.

8. **(Optional.) Verify if Samba now recognizes the driver.**

```
root# rpcclient -Uroot%xxxx -c 'enumdrivers 3' \
   localhost | grep -B2 -A5 mydrivername
Printer Driver Info 3:
Version: [2]
Driver Name: [mydrivername]
Architecture: [Windows NT x86]
Driver Path: [\\kde-bitshop\print$\W32X86\2\cupsdrvr.dll]
Datafile: [\\kde-bitshop\print$\W32X86\2\mysmbtstprn.PPD]
Configfile: [\\kde-bitshop\print$\W32X86\2\cupsui.dll]
Helpfile: [\\kde-bitshop\print$\W32X86\2\cups.hlp]
```

Remember, this command greps for the name you chose for the driver in step 6. This command must succeed before you can proceed.

9. **Tell Samba which printer should use these driver files (setdriver).**

```
root# rpcclient -Uroot%xxxx -c 'setdriver mysmbtstprn mydrivername' \
   localhost
Successfully set mysmbtstprn to driver mydrivername
```

Since you can bind any printer name (print queue) to any driver, this is a convenient way to set up many queues that use the same driver. You do not need to repeat all the previous steps for the setdriver command to succeed. The only preconditions are that **enumdrivers** must find the driver and **enumprinters** must find the printer.

10. **(Optional) Verify if Samba has recognized this association.**

```
root# rpcclient -Uroot%xxxx -c 'getprinter mysmbtstprn 2' localhost \
  | grep driver
```

```
drivername:[mydrivername]

root# rpcclient -Uroot%xxxx -c 'getprinter mysmbtstprn 2' localhost \
  | grep -C4 driv
servername:[\\kde-bitshop]
printername:[\\kde-bitshop\mysmbtstprn]
sharename:[mysmbtstprn]
portname:[Done]
drivername:[mydrivername]
comment:[mysmbtstprn]
location:[]
sepfile:[]
printprocessor:[winprint]

root# rpcclient -U root%xxxx -c 'getdriver mysmbtstprn' localhost
[Windows NT x86]
Printer Driver Info 3:
     Version: [2]
     Driver Name: [mydrivername]
     Architecture: [Windows NT x86]
     Driver Path: [\\kde-bitshop\print$\W32X86\2\cupsdrvr.dll]
     Datafile: [\\kde-bitshop\print$\W32X86\2\mysmbtstprn.PPD]
     Configfile: [\\kde-bitshop\print$\W32X86\2\cupsui.dll]
     Helpfile: [\\kde-bitshop\print$\W32X86\2\cups.hlp]
     Monitorname: []
     Defaultdatatype: [RAW]
     Monitorname: []
     Defaultdatatype: [RAW]

root# rpcclient -Uroot%xxxx -c 'enumprinters' localhost \
   | grep mysmbtstprn
     name:[\\kde-bitshop\mysmbtstprn]
     description:[\\kde-bitshop\mysmbtstprn,mydrivername,mysmbtstprn]
     comment:[mysmbtstprn]
```

Compare these results with the ones from steps 2 and 3. Every one of these commands show the driver is installed. Even the **enumprinters** command now lists the driver on the "description" line.

11. **(Optional.) Tickle the driver into a correct device mode.** You certainly know how to install the driver on the client. In case you are not particularly familiar with Windows, here is a short recipe: Browse the Network Neighborhood, go to the Samba server, and look for the shares. You should see all shared Samba printers. Double-click on the one in question. The driver should get installed and the network connection set up. Another way is to open the **Printers (and Faxes)** folder, right-click on the printer in question, and select **Connect** or **Install**. As a result, a new printer should appear in your client's local **Printers (and Faxes)** folder, named something like **printersharename on Sambahostname**. It is important that you execute this step as a Samba printer admin (as defined in `smb.conf`). Here is another method to do this on Windows XP. It uses a command line, which you may type into the "DOS box" (type root's smbpassword when prompted):

```
C:\> runas /netonly /user:root "rundll32 printui.dll,PrintUIEntry \
   /in /n \\sambaserver\mysmbtstprn"
```

Change any printer setting once (like changing **portrait** *to* **landscape**), click on **Apply**, and change the setting back.

12. **Install the printer on a client (Point'n'Print).**

```
C:\> rundll32 printui.dll,PrintUIEntry /in /n "\\sambaserver\mysmbtstprn"
```

If it does not work, it could be a permissions problem with the *[print$]* share.

13. **(Optional) Print a test page.**

```
C:\> rundll32 printui.dll,PrintUIEntry /p /n "\\sambaserver\mysmbtstprn"
```

Then hit [TAB] five times, [ENTER] twice, [TAB] once, and [ENTER] again, and march to the printer.

14. **(Recommended.) Study the test page.** Hmmm. Just kidding! By now you know everything about printer installations and you do not

need to read a word. Just put it in a frame and bolt it to the wall with the heading "MY FIRST RPCCLIENT-INSTALLED PRINTER" — why not just throw it away!

15. **(Obligatory.) Enjoy. Jump. Celebrate your success.**

```
root# echo "Cheeeeerioooooo! Success..." >> /var/log/samba/log.smbd
```

22.11.6 Troubleshooting Revisited

The setdriver command will fail if in Samba's mind the queue is not already there. A successful installation displys the promising message that the:

```
Printer Driver ABC successfully installed.
```

following the **adddriver** parts of the procedure. But you may also see a disappointing message like this one: `result was NT_STATUS_UNSUCCESSFUL`

It is not good enough that you can see the queue in CUPS, using the **lpstat -p ir85wm** command. A bug in most recent versions of Samba prevents the proper update of the queue list. The recognition of newly installed CUPS printers fails unless you restart Samba or send a HUP to all smbd processes. To verify if this is the reason why Samba does not execute the **setdriver** command successfully, check if Samba "sees" the printer:

```
root# rpcclient transmeta -N -U'root%xxxx' -c 'enumprinters 0'|grep ir85wm
        printername:[ir85wm]
```

An alternate command could be this:

```
root# rpcclient transmeta -N -U'root%secret' -c 'getprinter ir85wm'
        cmd = getprinter ir85wm
        flags:[0x800000]
        name:[\\transmeta\ir85wm]
        description:[\\transmeta\ir85wm,ir85wm,DPD]
```

```
comment:[CUPS PostScript-Treiber for Windows NT/200x/XP]
```

By the way, you can use these commands, plus a few more, of course, to install drivers on remote Windows NT print servers too!

22.12 The Printing *.tdb Files

Some mystery is associated with the series of files with a tdb suffix appearing in every Samba installation. They are `connections.tdb`, `printing.tdb`, `share_info.tdb`, `ntdrivers.tdb`, `unexpected.tdb`, `brlock.tdb`, `locking.tdb`, `ntforms.tdb`, `messages.tdb` , `ntprinters.tdb`, `sessionid.tdb`, and `secrets.tdb`. What is their purpose?

22.12.1 Trivial Database Files

A Windows NT (print) server keeps track of all information needed to serve its duty toward its clients by storing entries in the Windows registry. Client queries are answered by reading from the registry, Administrator or user configuration settings that are saved by writing into the registry. Samba and UNIX obviously do not have such a Registry. Samba instead keeps track of all client-related information in a series of `*.tdb` files. (TDB stands for trivial data base). These are often located in `/var/lib/samba/` or `/var/lock/samba/`. The printing-related files are `ntprinters.tdb`, `printing.tdb`,`ntforms.tdb`, and `ntdrivers.tdb`.

22.12.2 Binary Format

`*.tdb` files are not human readable. They are written in a binary format. "Why not ASCII?", you may ask. "After all, ASCII configuration files are a good and proven tradition on UNIX." The reason for this design decision by the Samba Team is mainly performance. Samba needs to be fast; it runs a separate **smbd** process for each client connection, in some environments many thousands of them. Some of these **smbds** might need to write-access the same `*.tdb` file *at the same time*. The file format of Samba's `*.tdb` files allows for this provision. Many smbd processes may write to the same `*.tdb` file at the same time. This wouldn't be possible with pure ASCII files.

22.12.3 Losing *.tdb Files

It is very important that all `*.tdb` files remain consistent over all write and read accesses. However, it may happen that these files *do* get corrupted. (A **kill -9 'pidof smbd'** while a write access is in progress could do the damage, as could a power interruption, etc.). In cases of trouble, a deletion of the old printing-related `*.tdb` files may be the only option. After that, you need to re-create all print-related setups unless you have made a backup of the `*.tdb` files in time.

22.12.4 Using tdbbackup

Samba ships with a little utility that helps the root user of your system to backup your `*.tdb` files. If you run it with no argument, it prints a usage message:

```
root# tdbbackup
 Usage: tdbbackup [options] <fname...>

 Version:3.0a
   -h            this help message
   -s suffix     set the backup suffix
   -v            verify mode (restore if corrupt)
```

Here is how I backed up my `printing.tdb` file:

```
root# ls
.               browse.dat     locking.tdb     ntdrivers.tdb printing.tdb
..              share_info.tdb connections.tdb messages.tdb  ntforms.tdb
printing.tdbkp unexpected.tdb brlock.tdb      gmon.out      namelist.debug
ntprinters.tdb sessionid.tdb

root# tdbbackup -s .bak printing.tdb
 printing.tdb : 135 records

root# ls -l printing.tdb*
 -rw-------    1 root     root        40960 May  2 03:44 printing.tdb
 -rw-------    1 root     root        40960 May  2 03:44 printing.tdb.bak
```

22.13 CUPS Print Drivers from Linuxprinting.org

CUPS ships with good support for HP LaserJet-type printers. You can install the generic driver as follows:

```
root# lpadmin -p laserjet4plus -v parallel:/dev/lp0 -E -m laserjet.ppd
```

The `-m` switch will retrieve the `laserjet.ppd` from the standard repository for not-yet-installed PPDs, which CUPS typically stores in `/usr/share/cups/model`. Alternatively, you may use `-P /path/to/your.ppd`.

The generic `laserjet.ppd,` however, does not support every special option for every LaserJet-compatible model. It constitutes a sort of "least common denominator" of all the models. If for some reason you must pay for the commercially available ESP Print Pro drivers, your first move should be to consult the database on the Linuxprinting[15] Web site. Linuxprinting.org has excellent recommendations about which driver is best used for each printer. Its database is kept current by the tireless work of Till Kamppeter from Mandrakesoft, who is also the principal author of the **foomatic-rip** utility.

[15] <http://www.linuxprinting.org/printer_list.cgi>

NOTE

The former **cupsomatic** concept is now being replaced by the new successor, a much more powerful **foomatic-rip**. **cupsomatic** is no longer maintained. Here is the new URL to the Foomatic-3.0[a] database. If you upgrade to **foomatic-rip**, remember to also upgrade to the new-style PPDs for your Foomatic-driven printers. foomatic-rip will not work with PPDs generated for the old **cupsomatic**. The new-style PPDs are 100% compliant with the Adobe PPD specification. They are also intended to be used by Samba and the cupsaddsmb utility, to provide the driver files for the Windows clients!

[a] <http://www.linuxprinting.org/driver_list.cgi>

22.13.1 foomatic-rip and Foomatic Explained

Nowadays, most Linux distributions rely on the utilities from the Linuxprinting.org[16] to create their printing-related software (which, by the way, works on all UNIXes and on Mac OS X and Darwin, too). The utilities from this sire have a very end-user-friendly interface that allows for an easy update of drivers and PPDs for all supported models, all spoolers, all operating systems, and all package formats (because there is none). Its history goes back a few years.

Recently, Foomatic has achieved the astonishing milestone of 1,000 listed[17] printer models. Linuxprinting.org keeps all the important facts about printer drivers, supported models, and which options are available for the various driver/printer combinations in its Foomatic[18] database. Currently there are 245 drivers[19] in the database. Many drivers support various models, and many models may be driven by different drivers — its your choice!

[16] <http://www.linuxprinting.org/>
[17] <http://www.linuxprinting.org/printer_list.cgi?make=Anyone>
[18] <http://www.linuxprinting.org/foomatic.html>
[19] <http://www.linuxprinting.org/driver_list.cgi>

22.13.1.1 690 "Perfect" Printers

At present, there are 690 devices dubbed as working perfectly: 181 are *mostly* perfect, 96 are *partially* perfect, and 46 are paperweights. Keeping in mind that most of these are non-PostScript models (PostScript printers are automatically supported by CUPS to perfection by using their own manufacturer-provided Windows PPD), and that a multifunctional device never qualifies as working perfectly if it does not also scan and copy and fax under GNU/Linux — then this is a truly astonishing achievement! Three years ago the number was not more than 500, and Linux or UNIX printing at the time wasn't anywhere near the quality it is today.

22.13.1.2 How the Printing HOWTO Started It All

A few years ago Grant Taylor[20] started it all. The roots of today's Linuxprinting.org are in the first Linux Printing HOWTO[21] that he authored. As a side-project to this document, which served many Linux users and admins to guide their first steps in this complicated and delicate setup (to a scientist, printing is "applying a structured deposition of distinct patterns of ink or toner particles on paper substrates"), he started to build in a little Postgres database with information about the hardware and driver zoo that made up Linux printing of the time. This database became the core component of today's Foomatic collection of tools and data. In the meantime, it has moved to an XML representation of the data.

22.13.1.3 Foomatic's Strange Name

"Why the funny name?" you ask. When it really took off, around spring 2000, CUPS was far less popular than today, and most systems used LPD, LPRng, or even PDQ to print. CUPS shipped with a few generic drivers (good for a few hundred different printer models). These didn't support many device-specific options. CUPS also shipped with its own built-in rasterization filter (*pstoraster*, derived from Ghostscript). On the other hand, CUPS provided brilliant support for *controlling* all printer options through standardized and well-defined PPD files. Plus, CUPS was designed to be easily extensible.

[20] <http://www2.picante.com/>
[21] <http://www.linuxprinting.org/foomatic2.9/howto/>

Taylor already had in his database a respectable compilation of facts about many more printers and the Ghostscript "drivers" they run with. His idea, to generate PPDs from the database information and use them to make standard Ghostscript filters work within CUPS, proved to work very well. It also killed several birds with one stone:

- It made all current and future Ghostscript filter developments available for CUPS.
- It made available a lot of additional printer models to CUPS users (because often the traditional Ghostscript way of printing was the only one available).
- It gave all the advanced CUPS options (Web interface, GUI driver configurations) to users wanting (or needing) to use Ghostscript filters.

22.13.1.4 cupsomatic, pdqomatic, lpdomatic, directomatic

CUPS worked through a quickly hacked-up filter script named cupsomatic[22]. cupsomatic ran the printfile through Ghostscript, constructing automatically the rather complicated command line needed. It just needed to be copied into the CUPS system to make it work. To configure the way cupsomatic controls the Ghostscript rendering process, it needs a CUPS-PPD. This PPD is generated directly from the contents of the database. For CUPS and the respective printer/filter combo, another Perl script named CUPS-O-Matic did the PPD generation. After that was working, Taylor implemented within a few days a similar thing for two other spoolers. Names chosen for the config-generator scripts were PDQ-O-Matic[23] (for PDQ) and LPD-O-Matic[24] (for — you guessed it — LPD); the configuration here didn't use PPDs but other spooler-specific files.

From late summer of that year, Till Kamppeter[25] started to put work into the database. Kamppeter had been newly employed by Mandrakesoft[26] to convert its printing system over to CUPS, after they had seen his FLTK[27]-based XPP[28] (a GUI front-end to the CUPS lp-command). He added a huge

[22] <http://www.linuxprinting.org/download.cgi?filename=cupsomatic&show=0>
[23] <http://www.linuxprinting.org/download.cgi?filename=lpdomatic&show=0>
[24] <http://www.linuxprinting.org/download.cgi?filename=lpdomatic&show=0>
[25] <http://www.linuxprinting.org/till/>
[26] <http://www.mandrakesoft.com/>
[27] <http://www.fltk.org/>
[28] <http://cups.sourceforge.net/xpp/>

amount of new information and new printers. He also developed the support for other spoolers, like PPR[29] (via ppromatic), GNUlpr[30], and LPRng[31] (both via an extended lpdomatic) and spooler-less printing (directomatic[32]).

So, to answer your question, "Foomatic" is the general name for all the overlapping code and data behind the "*omatic" scripts. Foomatic, up to versions 2.0.x, required (ugly) Perl data structures attached to Linuxprinting.org PPDs for CUPS. It had a different "*omatic" script for every spooler, as well as different printer configuration files.

22.13.1.5 The *Grand Unification* Achieved

This has all changed in Foomatic versions 2.9 (beta) and released as "stable" 3.0. It has now achieved the convergence of all *omatic scripts and is called the foomatic-rip[33]. This single script is the unification of the previously different spooler-specific *omatic scripts. foomatic-rip is used by all the different spoolers alike, and because it can read PPDs (both the original PostScript printer PPDs and the Linuxprinting.org-generated ones), all of a sudden all supported spoolers can have the power of PPDs at their disposal. Users only need to plug foomatic-rip into their system. For users there is improved media type and source support — paper sizes and trays are easier to configure.

Also, the new generation of Linuxprinting.org PPDs no longer contains Perl data structures. If you are a distro maintainer and have used the previous version of Foomatic, you may want to give the new one a spin, but remember to generate a new-version set of PPDs via the new foomatic-db-engine![34]. Individual users just need to generate a single new PPD specific to their model by following the steps[35] outlined in the Foomatic tutorial or in this chapter. This new development is truly amazing.

---

[29] <http://ppr.sourceforge.net/>

[30] <http://sourceforge.net/projects/lpr/>

[31] <http://www.lprng.org/>

[32] <http://www.linuxprinting.org/download.cgi?filename=directomatic&show=0>

[33] <http://www.linuxprinting.org/foomatic2.9/download.cgi?filename=foomatic-rip&show=0>

[34] <http://www.linuxprinting.org/download/foomatic/foomatic-db-engine-3.0.0beta1.tar.gz>

[35] <http://www.linuxprinting.org/kpfeifle/LinuxKongress2002/Tutorial/II.Foomatic-User/II.tutorial-handout-foomatic-user.html>

foomatic-rip is a very clever wrapper around the need to run Ghostscript with a different syntax, options, device selections, and/or filters for each different printer or spooler. At the same time, it can read the PPD associated with a print queue and modify the print job according to the user selections. Together with this comes the 100% compliance of the new Foomatic PPDs with the Adobe spec. Some innovative features of the Foomatic concept may surprise users. It will support custom paper sizes for many printers and will support printing on media drawn from different paper trays within the same job (in both cases, even where there is no support for this from Windows-based vendor printer drivers).

22.13.1.6 Driver Development Outside

Most driver development itself does not happen within Linuxprinting.org. Drivers are written by independent maintainers. Linuxprinting.org just pools all the information and stores it in its database. In addition, it also provides the Foomatic glue to integrate the many drivers into any modern (or legacy) printing system known to the world.

Speaking of the different driver development groups, most of the work is currently done in three projects:

- Omni[36] — a free software project by IBM that tries to convert its printer driver knowledge from good-ol' OS/2 times into a modern, modular, universal driver architecture for Linux/UNIX (still beta). This currently supports 437 models.
- HPIJS[37] — a free software project by HP to provide the support for its own range of models (very mature, printing in most cases is perfect and provides true photo quality). This currently supports 369 models.
- Gutenprint[38] — a free software effort, started by Michael Sweet (also lead developer for CUPS), now directed by Robert Krawitz, which has achieved an amazing level of photo print quality (many Epson users swear that its quality is better than the vendor drivers provided by Epson for the Microsoft platforms). This currently supports 522 models.

[36] <http://www-124.ibm.com/developerworks/oss/linux/projects/omni/>
[37] <http://hpinkjet.sf.net/>
[38] <http://gimp-print.sourceforge.net/>

22.13.1.7 Forums, Downloads, Tutorials, Howtos (Also for Mac OS X and Commercial UNIX)

Linuxprinting.org today is the one-stop shop to download printer drivers. Look for printer information and tutorials[39] or solve printing problems in its popular forums[40]. This forum is not just for GNU/Linux users, but admins of commercial UNIX systems[41] are also going there, and the relatively new Mac OS X forum[42] has turned out to be one of the most frequented forums after only a few weeks.

Linuxprinting.org and the Foomatic driver wrappers around Ghostscript are now a standard tool-chain for printing on all the important distros. Most of them also have CUPS underneath. While in recent years most printer data had been added by Kamppeter, many additional contributions came from engineers with SuSE, Red Hat, Conectiva, Debian, and others. Vendor-neutrality is an important goal of the Foomatic project. Mandrake and Conectiva have merged and are now called Mandriva.

> **Note**
>
> Till Kamppeter from Mandrakesoft is doing an excellent job in his spare time to maintain Linuxprinting.org and Foomatic. So if you use it often, please send him a note showing your appreciation.

22.13.1.8 Foomatic Database-Generated PPDs

The Foomatic database is an amazing piece of ingenuity in itself. Not only does it keep the printer and driver information, but it is organized in a way that it can generate PPD files on the fly from its internal XML-based datasets. While these PPDs are modeled to the Adobe specification of PPDs, the Linuxprinting.org/Foomatic-PPDs do not normally drive

[39] <http://www.linuxprinting.org//kpfeifle/LinuxKongress2002/Tutorial/>
[40] <http://www.linuxprinting.org/newsportal/>
[41] <http://www.linuxprinting.org/macosx/>
[42] <http://www.linuxprinting.org/newsportal/thread.php3?name=linuxprinting.macosx.general>

PostScript printers. They are used to describe all the bells and whistles you could ring or blow on an Epson Stylus inkjet, or an HP Photosmart, or what-have-you. The main trick is one little additional line, not envisaged by the PPD specification, starting with the *`*cupsFilter`* keyword. It tells the CUPS daemon how to proceed with the PostScript print file (old-style Foomatic-PPDs named the cupsomatic filter script, while the new-style PPDs are now call foomatic-rip). This filter script calls Ghostscript on the host system (the recommended variant is ESP Ghostscript) to do the rendering work. foomatic-rip knows which filter or internal device setting it should ask from Ghostscript to convert the PostScript print job into a raster format ready for the target device. This usage of PPDs to describe the options of non-PostScript printers was the invention of the CUPS developers. The rest is easy. GUI tools (like KDE's marvelous kprinter[43] or the GNOME gtklp[44] xpp and the CUPS Web interface) read the PPD as well and use this information to present the available settings to the user as an intuitive menu selection.

22.13.2 foomatic-rip and Foomatic PPD Download and Installation

Here are the steps to install a foomatic-rip-driven LaserJet 4 Plus-compatible printer in CUPS (note that recent distributions of SuSE, UnitedLinux and Mandrake may ship with a complete package of Foomatic-PPDs plus the **foomatic-rip** utility. Going directly to Linuxprinting.org ensures that you get the latest driver/PPD files).

- Open your browser at the Linuxprinting.org printer list page.[45]
- Check the complete list of printers in the database.[46].
- Select your model and click on the link.
- You'll arrive at a page listing all drivers working with this model (for all printers, there will always be *one* recommended driver. Try this one first).

[43] <http://printing.kde.org/overview/kprinter.phtml>
[44] <http://gtklp.sourceforge.net/>
[45] <http://www.linuxprinting.org/printer_list.cgi>
[46] <http://www.linuxprinting.org/printer_list.cgi?make=Anyone>

- In our case (HP LaserJet 4 Plus), we'll arrive at the default driver for the HP-LaserJet 4 Plus.[47]
- The recommended driver is ljet4.
- Several links are provided here. You should visit them all if you are not familiar with the Linuxprinting.org database.
- There is a link to the database page for the ljet4[48]. On the driver's page, you'll find important and detailed information about how to use that driver within the various available spoolers.
- Another link may lead you to the home page of the author of the driver.
- Important links are the ones that provide hints with setup instructions for CUPS[49]; PDQ[50]; LPD, LPRng, and GNUlpr[51]); as well as PPR[52] or "spoolerless" printing[53].
- You can view the PPD in your browser through this link: <`http://www.linuxprinting.org/ppd-o-matic.cgi?driver=ljet4&printer=HP-LaserJet_4_Plus&show=1`>
- Most importantly, you can also generate and download the PPD[54].
- The PPD contains all the information needed to use our model and the driver; once installed, this works transparently for the user. Later you'll only need to choose resolution, paper size, and so on, from the Web-based menu, or from the print dialog GUI, or from the command line.
- If you ended up on the drivers page[55], you can choose to use the "PPD-O-Matic" online PPD generator program.

[47] <`http://www.linuxprinting.org/show_printer.cgi?recnum=HP-LaserJet_4_Plus`>
[48] <`http://www.linuxprinting.org/show_driver.cgi?driver=ljet4`>
[49] <`http://www.linuxprinting.org/cups-doc.html`>
[50] <`http://www.linuxprinting.org/pdq-doc.html`>
[51] <`http://www.linuxprinting.org/lpd-doc.html`>
[52] <`http://www.linuxprinting.org/ppr-doc.html`>
[53] <`http://www.linuxprinting.org/direct-doc.html`>
[54] <`http://www.linuxprinting.org/ppd-o-matic.cgi?driver=ljet4&printer=HP-LaserJet_4_Plus&show=0`>
[55] <`http://www.linuxprinting.org/show_driver.cgi?driver=ljet4`>

- Select the exact model and check either **Download** or **Display PPD file** and click **Generate PPD file**.
- If you save the PPD file from the browser view, please do not use cut and paste (since it could possibly damage line endings and tabs, which makes the PPD likely to fail its duty), but use **Save as...** in your browser's menu. (It is best to use the **Download** option directly from the Web page.)
- Another interesting part on each driver page is the **Show execution details** button. If you select your printer model and click on that button, a complete Ghostscript command line will be displayed, enumerating all options available for that combination of driver and printer model. This is a great way to "learn Ghostscript by doing". It is also an excellent cheat sheet for all experienced users who need to reconstruct a good command line for that darned printing script, but can't remember the exact syntax.
- Sometime during your visit to Linuxprinting.org, save the PPD to a suitable place on your hard disk, say `/path/to/my-printer.ppd` (if you prefer to install your printers with the help of the CUPS Web interface, save the PPD to the `/usr/share/cups/model/` path and restart cupsd).
- Then install the printer with a suitable command line, like this:

```
root# lpadmin -p laserjet4plus -v parallel:/dev/lp0 -E \
   -P path/to/my-printer.ppd
```

- For all the new-style "Foomatic-PPDs" from Linuxprinting.org, you also need a special CUPS filter named foomatic-rip.
- The foomatic-rip Perl script itself also makes some interesting reading[56] because it is well documented by Kamppeter's in-line comments (even non-Perl hackers will learn quite a bit about printing by reading it).
- Save foomatic-rip either directly in `/usr/lib/cups/filter/foomatic-rip` or somewhere in your $PATH (and remember to make it world-

[56] <http://www.linuxprinting.org/foomatic2.9/download.cgi?filename=foomatic-rip&show=1>

executable). Again, do not save by copy and paste but use the appropriate link or the **Save as...** menu item in your browser.

- If you save foomatic-rip in your $PATH, create a symlink:

```
root# cd /usr/lib/cups/filter/ ; ln -s `which foomatic-rip'
```

  CUPS will discover this new available filter at startup after restarting cupsd.

Once you print to a print queue set up with the Foomatic PPD, CUPS will insert the appropriate commands and comments into the resulting PostScript job file. foomatic-rip is able to read and act upon these and uses some specially encoded Foomatic comments embedded in the job file. These in turn are used to construct (transparently for you, the user) the complicated Ghostscript command line telling the printer driver exactly how the resulting raster data should look and which printer commands to embed into the data stream. You need:

- A "foomatic+something" PPD — but this is not enough to print with CUPS (it is only *one* important component).
- The *`foomatic-rip`* filter script (Perl) in `/usr/lib/cups/filters/`.
- Perl to make foomatic-rip run.
- Ghostscript (because it is doing the main work, controlled by the PPD/foomatic-rip combo) to produce the raster data fit for your printer model's consumption.
- Ghostscript *must* (depending on the driver/model) contain support for a certain device representing the selected driver for your model (as shown by **gs -h**).
- foomatic-rip needs a new version of PPDs (PPD versions produced for cupsomatic do not work with foomatic-rip).

22.14 Page Accounting with CUPS

Often there are questions regarding print quotas where Samba users (that is, Windows clients) should not be able to print beyond a certain number of pages or data volume per day, week, or month. This feature is dependent

on the real print subsystem you're using. Samba's part is always to receive the job files from the clients (filtered *or* unfiltered) and hand them over to this printing subsystem.

Of course one could hack things with one's own scripts. But then there is CUPS. CUPS supports quotas that can be based on the size of jobs or on the number of pages or both, and can span any time period you want.

22.14.1 Setting Up Quotas

This is an example command of how root would set a print quota in CUPS, assuming an existing printer named "quotaprinter":

```
root# lpadmin -p quotaprinter -o job-quota-period=604800 \
   -o job-k-limit=1024 -o job-page-limit=100
```

This would limit every single user to print no more than 100 pages or 1024 KB of data (whichever comes first) within the last 604,800 seconds (= 1 week).

22.14.2 Correct and Incorrect Accounting

For CUPS to count correctly, the printfile needs to pass the CUPS pstops filter; otherwise it uses a dummy count of "one". Some print files do not pass it (e.g., image files), but then those are mostly one-page jobs anyway. This also means that proprietary drivers for the target printer running on the client computers and CUPS/Samba, which then spool these files as "raw" (i.e., leaving them untouched, not filtering them), will be counted as one-pagers too!

You need to send PostScript from the clients (i.e., run a PostScript driver there) to have the chance to get accounting done. If the printer is a non-PostScript model, you need to let CUPS do the job to convert the file to a print-ready format for the target printer. This is currently working for about a thousand different printer models. Linuxprinting.org has a driver list[57].

[57] <http://www.linuxprinting.org/printer_list.cgi>

22.14.3 Adobe and CUPS PostScript Drivers for Windows Clients

Before CUPS 1.1.16, your only option was to use the Adobe PostScript driver on the Windows clients. The output of this driver was not always passed through the **pstops** filter on the CUPS/Samba side, and therefore was not counted correctly (the reason is that it often, depending on the PPD being used, wrote a PJL-header in front of the real PostScript, which caused CUPS to skip **pstops** and go directly to the **pstoraster** stage).

From CUPS 1.1.16 and later releases, you can use the CUPS PostScript driver for Windows NT/200x/XP clients (which is tagged in the download area of `http://www.cups.org/` as the `cups-samba-1.1.16.tar.gz` package). It does *not* work for Windows 9x/Me clients, but it guarantees:

- To not write a PJL-header.
- To still read and support all PJL-options named in the driver PPD with its own means.
- That the file will pass through the **pstops** filter on the CUPS/Samba server.
- To page-count correctly the print file.

You can read more about the setup of this combination in the man page for **cupsaddsmb** (which is only present with CUPS installed, and only current from CUPS 1.1.16).

22.14.4 The page_log File Syntax

These are the items CUPS logs in the `page_log` for every page of a job:

- Printer name
- User name
- Job ID
- Time of printing
- Page number
- Number of copies
- A billing information string (optional)

- The host that sent the job (included since version 1.1.19)

Here is an extract of my CUPS server's `page_log` file to illustrate the format and included items:

```
tec_IS2027 kurt 401 [22/Apr/2003:10:28:43 +0100] 1 3 #marketing 10.160.50.13
tec_IS2027 kurt 401 [22/Apr/2003:10:28:43 +0100] 2 3 #marketing 10.160.50.13
tec_IS2027 kurt 401 [22/Apr/2003:10:28:43 +0100] 3 3 #marketing 10.160.50.13
tec_IS2027 kurt 401 [22/Apr/2003:10:28:43 +0100] 4 3 #marketing 10.160.50.13
Dig9110 boss 402 [22/Apr/2003:10:33:22 +0100] 1 440 finance-dep 10.160.51.33
```

This was job ID *401*, printed on *tec_IS2027* by user *kurt*, a 64-page job printed in three copies, billed to *#marketing*, and sent from IP address `10.160.50.13`. The next job had ID *402*, was sent by user *boss* from IP address `10.160.51.33`, printed from one page 440 copies, and is set to be billed to *finance-dep*.

22.14.5 Possible Shortcomings

What flaws or shortcomings are there with this quota system?

- The ones named above (wrongly logged job in case of printer hardware failure, and so on).
- In reality, CUPS counts the job pages that are being processed in *software* (that is, going through the RIP) rather than the physical sheets successfully leaving the printing device. Thus, if there is a jam while printing the fifth sheet out of 1,000 and the job is aborted by the printer, the page count will still show the figure of 1,000 for that job.
- All quotas are the same for all users (no flexibility to give the boss a higher quota than the clerk) and no support for groups.
- No means to read out the current balance or the "used-up" number of current quota.
- A user having used up 99 sheets of a 100 quota will still be able to send and print a 1,000 sheet job.

- A user being denied a job because of a filled-up quota does not get a meaningful error message from CUPS other than "client-error-not-possible".

22.14.6 Future Developments

This is the best system currently available, and there are huge improvements under development for CUPS 1.2:

- Page counting will go into the backends (these talk directly to the printer and will increase the count in sync with the actual printing process; thus, a jam at the fifth sheet will lead to a stop in the counting).
- Quotas will be handled more flexibly.
- Probably there will be support for users to inquire about their accounts in advance.
- Probably there will be support for some other tools around this topic.

22.14.7 Other Accounting Tools

Other accounting tools that can be used includes: PrintAnalyzer, pyKota, printbill, LogReport. For more information regarding these tools you can try a Google search.

22.15 Additional Material

A printer queue with *no* PPD associated to it is a "raw" printer, and all files will go directly there as received by the spooler. The exceptions are file types *`application/octet-stream`* that need the pass-through feature enabled. "Raw" queues do not do any filtering at all; they hand the file directly to the CUPS backend. This backend is responsible for sending the data to the device (as in the "device URI" notation: `lpd://`, `socket://`, `smb://`, `ipp://`, `http://`, `parallel:/`, `serial:/`, `usb:/`, and so on).

cupsomatic/Foomatic are *not* native CUPS drivers and they do not ship with CUPS. They are a third-party add-on developed at Linuxprinting.org. As such, they are a brilliant hack to make all models (driven by Ghostscript

drivers/filters in traditional spoolers) also work via CUPS, with the same (good or bad!) quality as in these other spoolers. *cupsomatic* is only a vehicle to execute a Ghostscript command line at that stage in the CUPS filtering chain where normally the native CUPS *pstoraster* filter would kick in. *cupsomatic* bypasses *pstoraster*, kidnaps the print file from CUPS, and redirects it to go through Ghostscript. CUPS accepts this because the associated cupsomatic/foomatic-PPD specifies:

```
*cupsFilter:  "application/vnd.cups-postscript 0 cupsomatic"
```

This line persuades CUPS to hand the file to *cupsomatic* once it has successfully converted it to the MIME type *application/vnd.cups-postscript*. This conversion will not happen for jobs arriving from Windows that are autotyped *application/octet-stream*, with the according changes in `/etc/cups/mime.types` in place.

CUPS is widely configurable and flexible, even regarding its filtering mechanism. Another workaround in some situations would be to have in `/etc/cups/mime.types` entries as follows:

```
application/postscript           application/vnd.cups-raw  0  -
application/vnd.cups-postscript  application/vnd.cups-raw  0  -
```

This would prevent all PostScript files from being filtered (rather, they will through the virtual *nullfilter* denoted with "-"). This could only be useful for PostScript printers. If you want to print PostScript code on non-PostScript printers (provided they support ASCII text printing), an entry as follows could be useful:

```
*/*           application/vnd.cups-raw  0  -
```

and would effectively send *all* files to the backend without further processing.

You could have the following entry:

```
application/vnd.cups-postscript application/vnd.cups-raw 0 \
```

```
my_PJL_stripping_filter
```

You will need to write a *my_PJL_stripping_filter* (which could be a shell script) that parses the PostScript and removes the unwanted PJL. This needs to conform to CUPS filter design (mainly, receive and pass the parameters printername, job-id, username, jobtitle, copies, print options, and possibly the filename). It is installed as world executable into `/usr/lib/cups/filters/` and is called by CUPS if it encounters a MIME type *application/vnd.cups-postscript*.

CUPS can handle *-o job-hold-until=indefinite*. This keeps the job in the queue on hold. It will only be printed upon manual release by the printer operator. This is a requirement in many central reproduction departments, where a few operators manage the jobs of hundreds of users on some big machine, where no user is allowed to have direct access (such as when the operators often need to load the proper paper type before running the 10,000 page job requested by marketing for the mailing, and so on).

22.16 Autodeletion or Preservation of CUPS Spool Files

Samba print files pass through two spool directories. One is the incoming directory managed by Samba (set in the *path* = /var/spool/samba directive in the *[printers]* section of `smb.conf`). The other is the spool directory of your UNIX print subsystem. For CUPS it is normally `/var/spool/cups/`, as set by the `cupsd.conf` directive `RequestRoot /var/spool/cups`.

22.16.1 CUPS Configuration Settings Explained

Some important parameter settings in the CUPS configuration file `cupsd.conf` are:

PreserveJobHistory Yes This keeps some details of jobs in cupsd's mind (well, it keeps the c12345, c12346, and so on, files in the CUPS spool directory, which does a similar job as the old-fashioned BSD-LPD control files). This is set to "Yes" as a default.

PreserveJobFiles Yes This keeps the job files themselves in cupsd's mind (it keeps the d12345, d12346, etc., files in the CUPS spool directory). This is set to "No" as the CUPS default.

"MaxJobs 500" This directive controls the maximum number of jobs that are kept in memory. Once the number of jobs reaches the limit, the oldest completed job is automatically purged from the system to make room for the new one. If all of the known jobs are still pending or active, then the new job will be rejected. Setting the maximum to 0 disables this functionality. The default setting is 0.

(There are also additional settings for *`MaxJobsPerUser`* and *`MaxJobsPerPrinter`*.)

22.16.2 Preconditions

For everything to work as it should, you need to have three things:

- A Samba smbd that is compiled against `libcups` (check on Linux by running `ldd 'which smbd'`).
- A Samba-`smb.conf` setting of *printing* = cups.
- Another Samba `smb.conf` setting of *printcap* = cups.

NOTE

In this case, all other manually set printing-related commands (like *print command*, *lpq command*, *lprm command*, *lppause command*, and *lpresume command*) are ignored, and they should normally have no influence whatsoever on your printing.

22.16.3 Manual Configuration

If you want to do things manually, replace the *printing* = cups by *printing* = bsd. Then your manually set commands may work (I haven't tested this),

and a *print command* = lp -d %P %s; rm %s may do what you need.

22.17 Printing from CUPS to Windows-Attached Printers

From time to time the question arises, how can you print *to* a Windows-attached printer *from* Samba? Normally the local connection from Windows host to printer would be done by USB or parallel cable, but this does not matter to Samba. From here only an SMB connection needs to be opened to the Windows host. Of course, this printer must be shared first. As you have learned by now, CUPS uses *backends* to talk to printers and other servers. To talk to Windows shared printers, you need to use the `smb` (surprise, surprise!) backend. Check if this is in the CUPS backend directory. This usually resides in `/usr/lib/cups/backend/`. You need to find an `smb` file there. It should be a symlink to `smbspool`, and the file must exist and be executable:

```
root# ls -l /usr/lib/cups/backend/
total 253
drwxr-xr-x    3 root   root     720 Apr 30 19:04 .
drwxr-xr-x    6 root   root     125 Dec 19 17:13 ..
-rwxr-xr-x    1 root   root   10692 Feb 16 21:29 canon
-rwxr-xr-x    1 root   root   10692 Feb 16 21:29 epson
lrwxrwxrwx    1 root   root       3 Apr 17 22:50 http -> ipp
-rwxr-xr-x    1 root   root   17316 Apr 17 22:50 ipp
-rwxr-xr-x    1 root   root   15420 Apr 20 17:01 lpd
-rwxr-xr-x    1 root   root    8656 Apr 20 17:01 parallel
-rwxr-xr-x    1 root   root    2162 Mar 31 23:15 pdfdistiller
lrwxrwxrwx    1 root   root      25 Apr 30 19:04 ptal -> /usr/sbin/ptal-cups
-rwxr-xr-x    1 root   root    6284 Apr 20 17:01 scsi
lrwxrwxrwx    1 root   root      17 Apr  2 03:11 smb -> /usr/bin/smbspool
-rwxr-xr-x    1 root   root    7912 Apr 20 17:01 socket
-rwxr-xr-x    1 root   root    9012 Apr 20 17:01 usb

root# ls -l `which smbspool`
-rwxr-xr-x    1 root   root  563245 Dec 28 14:49 /usr/bin/smbspool
```

If this symlink does not exist, create it:

```
root# ln -s `which smbspool` /usr/lib/cups/backend/smb
```

smbspool was written by Mike Sweet from the CUPS folks. It is included and ships with Samba. It may also be used with print subsystems other than CUPS, to spool jobs to Windows printer shares. To set up printer *winprinter* on CUPS, you need to have a driver for it. Essentially this means to convert the print data on the CUPS/Samba host to a format that the printer can digest (the Windows host is unable to convert any files you may send). This also means you should be able to print to the printer if it were hooked directly at your Samba/CUPS host. For troubleshooting purposes, this is what you should do to determine if that part of the process chain is in order. Then proceed to fix the network connection/authentication to the Windows host, and so on.

To install a printer with the *smb* backend on CUPS, use this command:

```
root# lpadmin -p winprinter -v smb://WINDOWSNETBIOSNAME/printersharename \
  -P /path/to/PPD
```

The PPD must be able to direct CUPS to generate the print data for the target model. For PostScript printers, just use the PPD that would be used with the Windows NT PostScript driver. But what can you do if the printer is only accessible with a password? Or if the printer's host is part of another workgroup? This is provided for: You can include the required parameters as part of the `smb://` device-URI like this:

- `smb://WORKGROUP/WINDOWSNETBIOSNAME/printersharename`
- `smb://username:password@WORKGROUP/WINDOWSNETBIOSNAME/printersharename`
- `smb://username:password@WINDOWSNETBIOSNAME/printersharename`

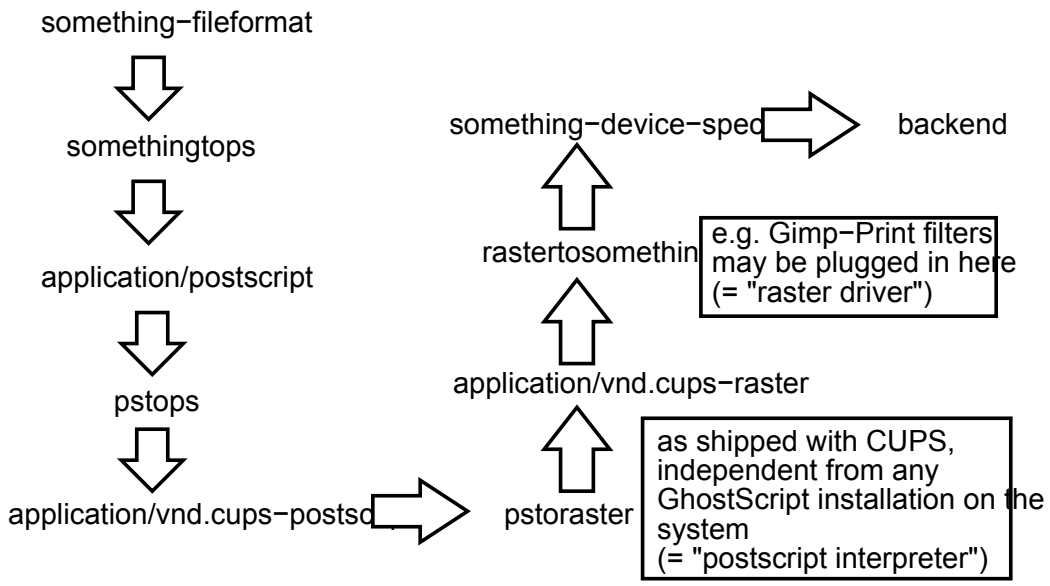
Note that the device URI will be visible in the process list of the Samba server (e.g., when someone uses the **ps -aux** command on Linux), even if the username and passwords are sanitized before they get written into the log files. This is an inherently insecure option; however, it is the only one. Don't use it if you want to protect your passwords. Better share the printer in a way that does not require a password! Printing will only work if you have a working NetBIOS name resolution up and running. Note that this is a feature of CUPS and you do not necessarily need to have smbd running.

22.18 More CUPS Filtering Chains

The diagrams in Figure 22.17 and Figure 22.18 show how CUPS handles print jobs.

Figure 22.17 Filtering Chain 1.

CUPS in and of itself has this (general) filter chain (italic letters are file−formats or MIME types, other are filters (this is true for pre−1.1.15 of pre−4.3 versions of CUPS and ESP PrintPro):



ESP PrintPro has some enhanced "rastertosomething" filters as compared to CUPS, and also a somewhat improved "pstoraster" filter.

NOTE: Gimp−Print and some other 3rd−Party−Filters (like TurboPrint) to CUPS and ESP PrintPro plug−in where rastertosomething is noted.

22.19 Common Errors

22.19.1 Windows 9x/Me Client Can't Install Driver

For Windows 9x/Me, clients require the printer names to be eight characters (or "8 plus 3 chars suffix") max; otherwise, the driver files will not get transferred when you want to download them from Samba.

22.19.2 "cupsaddsmb" Keeps Asking for Root Password in Never-ending Loop

Have you set *security* = user? Have you used **smbpasswd** to give root a Samba account? You can do two things: open another terminal and execute **smbpasswd -a root** to create the account and continue entering the password into the first terminal. Or, break out of the loop by pressing Enter twice (without trying to type a password).

If the error is "Tree connect failed: NT_STATUS_BAD_NETWORK_NAME", you may have forgotten to create the `/etc/samba/drivers` directory.

22.19.3 "cupsaddsmb" or "rpcclient addriver" Emit Error

If **cupsaddsmb**, or **rpcclient addriver** emit the error message WERR_BAD_PASSWORD, refer to Section 22.19.2.

22.19.4 "cupsaddsmb" Errors

The use of "cupsaddsmb" gives "No PPD file for printer..." message while PPD file is present. What might the problem be?

Have you enabled printer sharing on CUPS? This means, do you have a `<Location /printers>....</Location>` section in CUPS server's `cupsd.conf` that does not deny access to the host you run "cupsaddsmb" from? It *could* be an issue if you use cupsaddsmb remotely, or if you use it with a `-h` parameter: `cupsaddsmb -H sambaserver -h cupsserver -v printername`.

Is your *`TempDir`* directive in `cupsd.conf` set to a valid value, and is it writable?

22.19.5 Client Can't Connect to Samba Printer

Use **smbstatus** to check which user you are from Samba's point of view. Do you have the privileges to write into the *[print$]* share?

22.19.6 New Account Reconnection from Windows 200x/XP Troubles

Once you are connected as the wrong user (for example, as `nobody`, which often occurs if you have *map to guest* = bad user), Windows Explorer will not accept an attempt to connect again as a different user. There will not be any bytes transferred on the wire to Samba, but still you'll see a stupid error message that makes you think Samba has denied access. Use **smbstatus** to check for active connections. Kill the PIDs. You still can't re-connect, and you get the dreaded `You can't connect with a second account from the same machine` message as soon as you try. And you do not see a single byte arriving at Samba (see logs; use "ethereal") indicating a renewed connection attempt. Shut all Explorer Windows. This makes Windows forget what it has cached in its memory as established connections. Then reconnect as the right user. The best method is to use a DOS terminal window and *first* do `net use z: \\GANDALF\print$ /user:root`. Check with **smbstatus** that you are connected under a different account. Now open the **Printers** folder (on the Samba server in the **Network Neighborhood**), right-click on the printer in question, and select **Connect...**.

22.19.7 Avoid Being Connected to the Samba Server as the Wrong User

You see per **smbstatus** that you are connected as user nobody, but you want to be root or printer admin. This is probably due to *map to guest* = bad user, which silently connected you under the guest account when you gave (maybe by accident) an incorrect username. Remove *map to guest* if you want to prevent this.

22.19.8 Upgrading to CUPS Drivers from Adobe Drivers

This information came from a mailing list posting regarding problems experienced when upgrading from Adobe drivers to CUPS drivers on Microsoft Windows NT/200x/XP clients.

First delete all old Adobe-using printers. Then delete all old Adobe drivers. (On Windows 200x/XP, right-click in the background of **Printers** folder, select **Server Properties...**, select tab **Drivers**, and delete here).

22.19.9 Can't Use "cupsaddsmb" on Samba Server, Which Is a PDC

Do you use the "naked" root user name? Try to do it this way: `cupsaddsmb -U` *`DOMAINNAME`*`\\root -v` *`printername`*> (note the two backslashes: the first one is required to "escape" the second one).

22.19.10 Deleted Windows 200x Printer Driver Is Still Shown

Deleting a printer on the client will not delete the driver too (to verify, right-click on the white background of the **Printers** folder, select **Server Properties** and click on the **Drivers** tab). These same old drivers will be re-used when you try to install a printer with the same name. If you want to update to a new driver, delete the old ones first. Deletion is only possible if no other printer uses the same driver.

22.19.11 Windows 200x/XP Local Security Policies

Local security policies may not allow the installation of unsigned drivers — "local security policies" may not allow the installation of printer drivers at all.

22.19.12 Administrator Cannot Install Printers for All Local Users

Windows XP handles SMB printers on a "per-user" basis. This means every user needs to install the printer himself or herself. To have a printer available for everybody, you might want to use the built-in IPP client capabilities of Win XP. Add a printer with the print path of *`http://cupsserver:631/printers/printername`*. We're still looking into this one. Maybe a logon script could automatically install printers for all users.

22.19.13 Print Change, Notify Functions on NT Clients

For print change, notify functions on NT++ clients. These need to run the **Server** service first (renamed to **File & Print Sharing for MS Networks** in XP).

22.19.14 Windows XP SP1

Windows XP SP1 introduced a Point and Print Restriction Policy (this restriction does not apply to "Administrator" or "Power User" groups of users). In Group Policy Object Editor, go to **User Configuration** -> **Administrative Templates** -> **Control Panel** -> **Printers**. The policy is automatically set to `Enabled` and the `Users can only Point and Print to machines in their Forest` . You probably need to change it to `Disabled` or `Users can only Point and Print to these servers` to make driver downloads from Samba possible.

22.19.15 Print Options for All Users Can't Be Set on Windows 200x/XP

How are you doing it? I bet the wrong way (it is not easy to find out, though). There are three different ways to bring you to a dialog that *seems* to set everything. All three dialogs *look* the same, yet only one of them does what you intend. You need to be Administrator or Print Administrator to do this for all users. Here is how I do it on XP:

A The first wrong way: .

(a) Open the **Printers** folder.

(b) Right-click on the printer (**remoteprinter on cupshost**) and select in context menu **Printing Preferences...**

(c) Look at this dialog closely and remember what it looks like.

B The second wrong way: .

(a) Open the **Printers** folder.

(b) Right-click on the printer (**remoteprinter on cupshost**) and select the context menu **Properties**.

(c) Click on the **General** tab.

(d) Click on the button **Printing Preferences...**

(e) A new dialog opens. Keep this dialog open and go back to the parent dialog.

C The third and correct way:

(a) Open the **Printers** folder.

(b) Right-click on the printer (**remoteprinter on cupshost**) and select the context menu **Properties**.

(c) Click on the **Advanced** tab. (If everything is "grayed out," then you are not logged in as a user with enough privileges).

(d) Click on the **Printing Defaults...** button.

(e) On any of the two new tabs, click on the **Advanced...** button.

(f) A new dialog opens. Compare this one to the other identical-looking one from step "B.5" or A.3".

Do you see any difference? I don't either. However, only the last one, which you arrived at with steps "C.1. to C.6.", will save any settings permanently and be the defaults for new users. If you want all clients to get the same defaults, you need to conduct these steps *as Administrator* (*printer admin* in `smb.conf`) *before* a client downloads the driver (the clients can later set their own *per-user defaults* by following the procedures *A* or *B*).

22.19.16 Most Common Blunders in Driver Settings on Windows Clients

Don't use *`Optimize for Speed`*, but use *`Optimize for Portability`* instead (Adobe PS Driver). Don't use *`Page Independence: No`*. Always settle with *`Page Independence: Yes`* (Microsoft PS Driver and CUPS PS Driver for Windows NT/200x/XP). If there are problems with fonts, use *`Download as Softfont into printer`* (Adobe PS Driver). For **TrueType Download Options** choose `Outline`. Use PostScript Level 2 if you are having trouble with a non-PS printer and if there is a choice.

22.19.17 cupsaddsmb Does Not Work with Newly Installed Printer

Symptom: The last command of **cupsaddsmb** does not complete successfully. If the **cmd = setdriver printername printername** result was NT_STATUS_UNSUCCESSFUL, then possibly the printer was not yet recognized by Samba. Did it show up in Network Neighborhood? Did it show up in **rpcclient hostname -c 'enumprinters'**? Restart smbd (or send a **kill -HUP** to all processes listed by **smbstatus**, and try again.

22.19.18 Permissions on /var/spool/samba/ Get Reset After Each Reboot

Have you ever by accident set the CUPS spool directory to the same location (*RequestRoot /var/spool/samba/* in `cupsd.conf` or the other way round: `/var/spool/cups/` is set as *path*> in the *[printers]* section)? These *must* be different. Set *RequestRoot /var/spool/cups/* in `cupsd.conf` and *path* = /var/spool/samba in the *[printers]* section of `smb.conf`. Otherwise, cupsd will sanitize permissions to its spool directory with each restart and printing will not work reliably.

22.19.19 Print Queue Called "lp" Mishandles Print Jobs

In this case a print queue called "lp" intermittently swallows jobs and spits out completely different ones from what was sent.

It is a bad idea to name any printer "lp". This is the traditional UNIX name for the default printer. CUPS may be set up to do an automatic creation of Implicit Classes. This means, to group all printers with the same name to a pool of devices and load-balance the jobs across them in a round-robin fashion. Chances are high that someone else has a printer named "lp" too. You may receive that person's jobs and send your own to his or her device unwittingly. To have tight control over the printer names, set *BrowseShortNames No*. It will present any printer as *printername@cupshost*, which gives you better control over what may happen in a large networked environment.

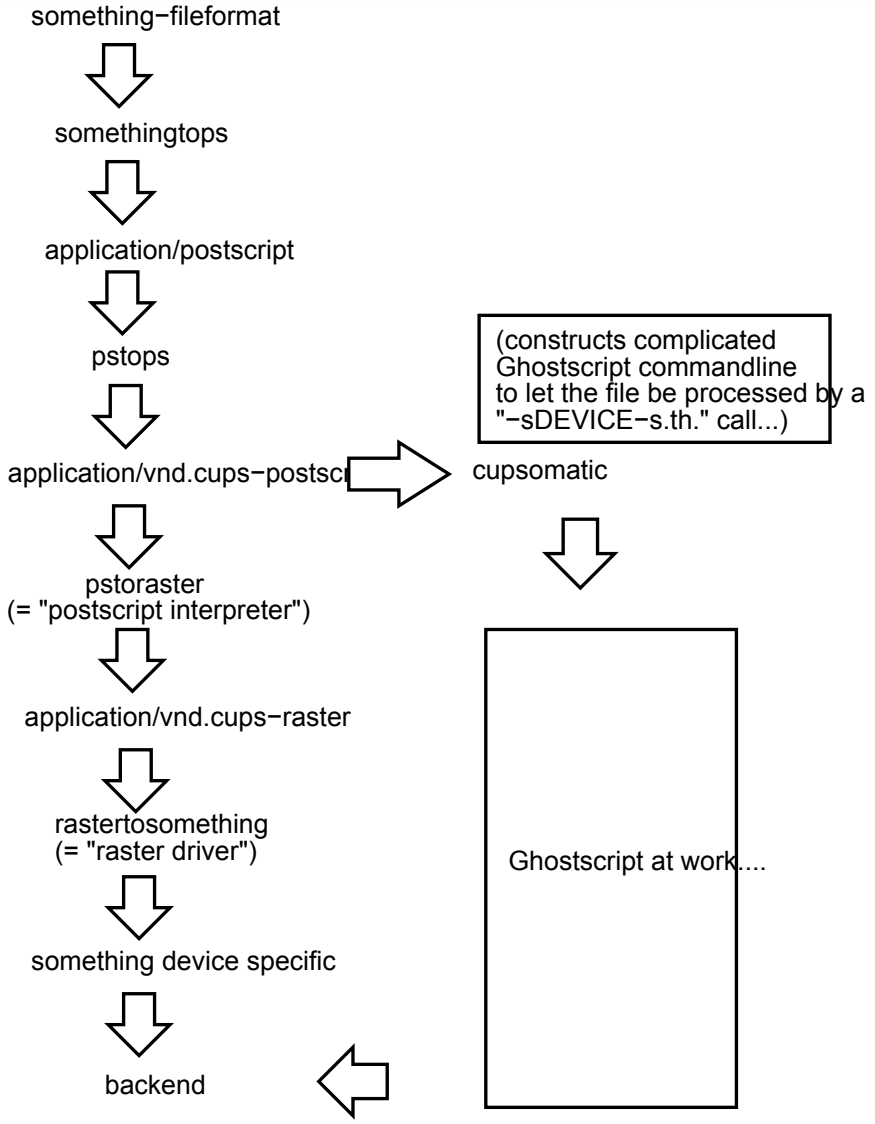
22.19.20 Location of Adobe PostScript Driver Files for "cupsaddsmb"

Use **smbclient** to connect to any Windows box with a shared PostScript printer: **smbclient //windowsbox/print\$ -U guest**. You can navigate to the `W32X86/2` subdir to **mget ADOBE*** and other files or to `WIN40/0` to do the same. Another option is to download the `*.exe` packaged files from the Adobe Web site.

22.20 Overview of the CUPS Printing Processes

A complete overview of the CUPS printing processes can be found in Figure 22.19.

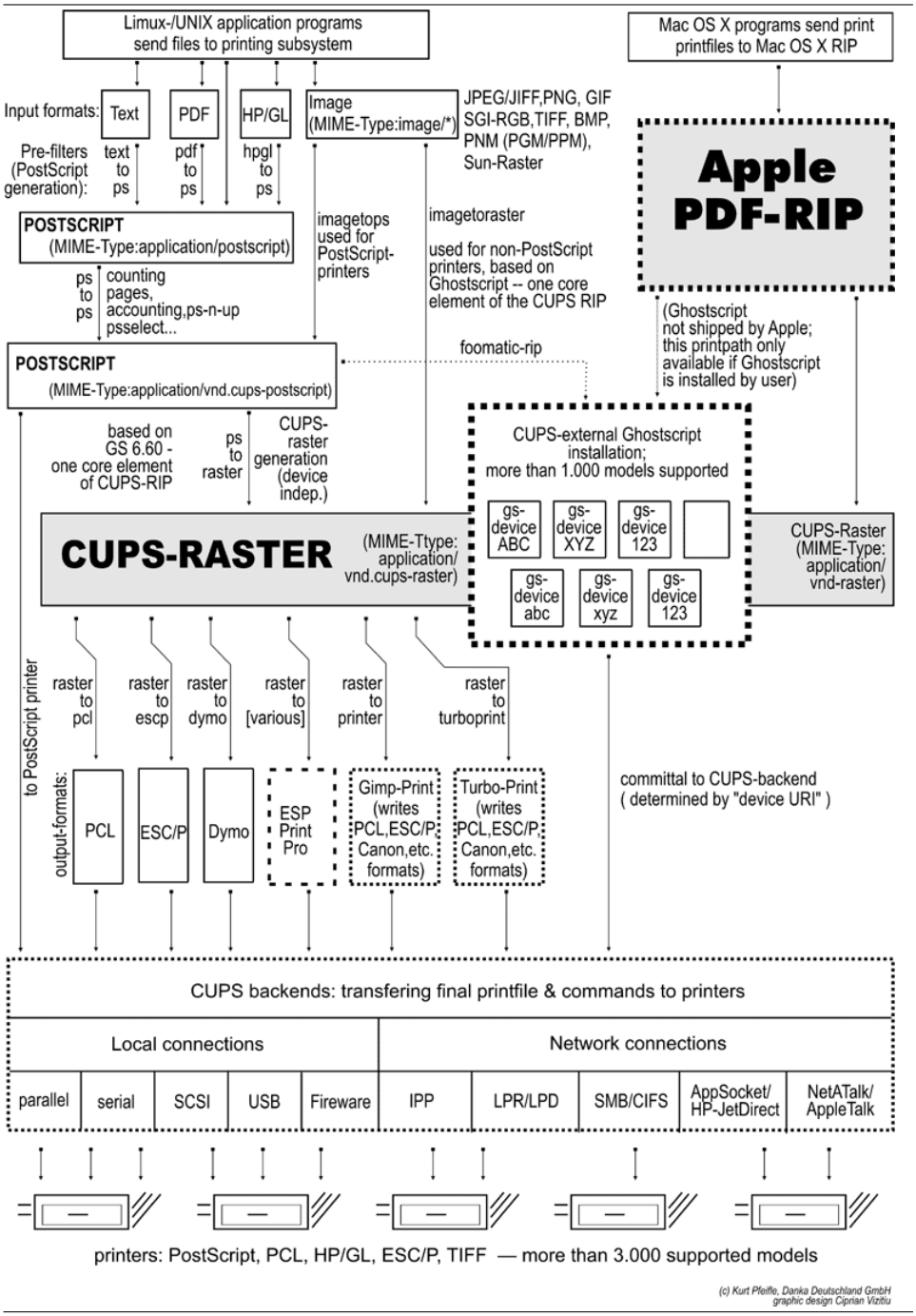
Figure 22.18 Filtering Chain with cupsomatic



Note, that cupsomatic "kidnaps" the printfile after the application/vnd.cups−postscript stage and deviates it gh the CUPS−external, systemwide Ghostscript installation, bypassing the "pstoraster" filter (therefore also bypassing the CUPS−raster−drivers "rastertosomething", and hands the rasterized file directly to the CUPS backend...

cupsomatic is not made by the CUPS developers. It is an independent contribution to printing development, made by people from Linuxprinting.org. (see also http://www.cups.org/cups−help.html)

Figure 22.19 CUPS Printing Overview.



Chapter 23

STACKABLE VFS MODULES

23.1 Features and Benefits

Stackable VFS (Virtual File System) modules support was new to Samba-3 and has proven quite popular. Samba passes each request to access the UNIX file system through the loaded VFS modules. This chapter covers the modules that come with the Samba source and provides references to some external modules.

23.2 Discussion

If not supplied with your platform distribution binary Samba package, you may have problems compiling these modules, as shared libraries are compiled and linked in different ways on different systems. They currently have been tested against GNU/Linux and IRIX.

To use the VFS modules, create a share similar to the one below. The important parameter is the *vfs objects* parameter where you can list one or more VFS modules by name. For example, to log all access to files and put deleted files in a recycle bin, see Example 23.2.1:

The modules are used in the order in which they are specified. Let's say that you want to both have a virus scanner module and a recycle bin module. It is wise to put the virus scanner module as the first one so that it is the first to get run and may detect a virus immediately, before any action is performed on that file. *vfs objects* = vscan-clamav recycle

Example 23.2.1 smb.conf with VFS modules

```
[audit]
        comment = Audited /data directory
        path = /data
        vfs objects = audit recycle
        writeable = yes
        browseable = yes
```

Samba will attempt to load modules from the `/lib` directory in the root directory of the Samba installation (usually `/usr/lib/samba/vfs` or `/usr/local/samba/lib/vfs`).

Some modules can be used twice for the same share. This can be done using a configuration similar to the one shown in Example 23.2.2.

Example 23.2.2 smb.conf with multiple VFS modules

```
[test]
        comment = VFS TEST
        path = /data
        writeable = yes
        browseable = yes
        vfs objects = example:example1 example example: ↩
            test
        example1: parameter = 1
        example:  parameter = 5
        test:     parameter = 7
```

23.3 Included Modules

23.3.1 audit

A simple module to audit file access to the syslog facility. The following operations are logged:

- share
- connect/disconnect
- directory opens/create/remove

- file open/close/rename/unlink/chmod

23.3.2 default_quota

This module allows the default quota values, in the windows explorer GUI, to be stored on a Samba-3 server. The challenge is that linux filesystems only store quotas for users and groups, but no default quotas.

Samba returns NO_LIMIT as the default quotas by default and refuses to update them. With this module you can store the default quotas that are reported to a windows client, in the quota record of a user. By default the root user is taken because quota limits for root are typically not enforced.

This module takes 2 parametric entries in the `smb.conf` file. The default prefix for each is the "default_quota". This can be overwrittem when you load the module in the *vfs modules* parameter like this:

```
vfs objects = default_quota:myprefix
```

The parametric entries that may be specified for the default_quotas module are:

myprefix:uid This parameter takes a integer argument that specifies the uid of the quota record that will be used for storing the default user quotas.

The default value is 0 (for root user). An example of use is:

```
vfs objects = default_quota
default_quota: uid = 65534
```

The above demonstrates the case where the `myprefix` was omitted, thus the default prefix is the name of the module. When a `myprefix` parameter is specified the above can be re-written like this:

```
vfs objects = default_quota:myprefix
myprefix:   uid = 65534
```

myprefix:uid nolimit This parameter takes a boolean argument that specifies if the stored default quota values also be reported for the user record, or if the value `NO_LIMIT` should be reported to the windows client for the user specified by the *`prefix:uid`* parameter.

The default value is `yes` (which means to report NO_LIMIT). An example of use is shown here:

```
vfs objects = default_quota:myprefix
myprefix:   uid nolimit = no
```

myprefix:gid This parameter takes an integer argument, it's just like the *`prefix>:uid`* but for group quotas. NOTE: group quotas are not supported from the windows explorer.

The default value is 0 (for root group). An example of use is shown here:

```
vfs objects = default_quota
default_quota: gid = 65534
```

myprefix:gid nolimit This parameter takes a boolean argument, just like the *`prefix>:uid nolimit`* but for group quotas. NOTE: group quotas are not supported from the windows explorer.

The default value is `yes` (which means to report NO_LIMIT). An example of use is shown here:

```
vfs objects = default_quota
default_quota: uid nolimit = no
```

An example of use of multiple parametric specifications is shown here:

...

```
vfs objects = default_quota:quotasettings
quotasettings: uid nolimit = no
quotasettings: gid = 65534
quotasettings: gid nolimit = no
...
```

23.3.3 extd_audit

This module is identical with the **audit** module above except that it sends audit logs to both syslog as well as the **smbd** log files. The *log level* for this module is set in the `smb.conf` file.

Valid settings and the information that will be recorded are shown in Table 23.1.

Table 23.1 Extended Auditing Log Information

Log Level	Log Details - File and Directory Operations
0	Make Directory, Remove Directory, Unlink
1	Open Directory, Rename File, Change Permissions/ACLs
2	Open & Close File
10	Maximum Debug Level

23.3.3.1 Configuration of Auditing

This auditing tool is more felxible than most people readily will recognize. There are a number of ways by which useful logging information can be recorded.

- Syslog can be used to record all transaction. This can be disabled by setting in the `smb.conf` file *`syslog = 0`*.
- Logging can take place to the default log file (`log.smbd`) for all loaded VFS modules just by setting in the `smb.conf` file *`log level = 0 vfs:x`*, where x is the log level. This will disable general logging while activating all logging of VFS module activity at the log level specified.
- Detailed logging can be obtained per user, per client machine, etc. This requires the above together with the creative use of the *`log file`* settings.

An example of detailed per-user and per-machine logging can be obtained by setting *log file* = /var/log/samba/%U.%m.log.

Auditing information often must be preserved for a long time. So that the log files do not get rotated it is essential that the *max log size* = 0 be set in the `smb.conf` file.

23.3.4 fake_perms

This module was created to allow Roaming Profile files and directories to be set (on the Samba server under UNIX) as read only. This module will, if installed on the Profiles share, report to the client that the Profile files and directories are writeable. This satisfies the client even though the files will never be overwritten as the client logs out or shuts down.

23.3.5 recycle

A Recycle Bin-like module. Where used, unlink calls will be intercepted and files moved to the recycle directory instead of being deleted. This gives the same effect as the **Recycle Bin** on Windows computers.

The **Recycle Bin** will not appear in Windows Explorer views of the network file system (share) nor on any mapped drive. Instead, a directory called `.recycle` will be automatically created when the first file is deleted and *`recycle:repository`* is not configured. If *`recycle:repository`* is configured, the name of the created directory depends on *`recycle:repository`*. Users can recover files from the recycle bin. If the *`recycle:keeptree`* has been specified, deleted files will be found in a path identical with that from which the file was deleted.

Supported options for the **recycle** module are as follow:

recycle:repository Path of the directory where deleted files should be moved.

recycle:directory_mode Set it to the octal mode you want for the recycle directory. With this mode the recycle directory will be created if it not exists and the first file is deleted. If *`recycle:subdir_mode`* is not

set, these mode also apply to sub directories. If *directory_mode* not exists, the default mode 0700 is used.

recycle:subdir_mode Set it to the octal mode you want for the sub directories of the recycle directory. With this mode the sub directories will be created. If *recycle:subdir_mode* is not set, the sub directories will be created with the mode from *directory_mode*.

recycle:keeptree Specifies whether the directory structure should be kept or if the files in the directory that is being deleted should be kept separately in the recycle bin.

recycle:versions If this option is set, two files with the same name that are deleted will both be kept in the recycle bin. Newer deleted versions of a file will be called "Copy #x of *filename*".

recycle:touch Specifies whether a file's access date should be touched when the file is moved to the recycle bin.

recycle:touch_mtime Specifies whether a file's last modify date date should be touched when the file is moved to the recycle bin.

recycle:maxsize Files that are larger than the number of bytes specified by this parameter will not be put into the recycle bin.

recycle:exclude List of files that should not be put into the recycle bin when deleted, but deleted in the regular way.

recycle:exclude_dir Contains a list of directories. When files from these directories are deleted, they are not put into the recycle bin but are deleted in the regular way.

recycle:noversions Specifies a list of paths (wildcards such as * and ? are supported) for which no versioning should be used. Only useful when *recycle:versions* is enabled.

23.3.6 netatalk

A netatalk module will ease co-existence of Samba and netatalk file sharing services.

Advantages compared to the old netatalk module:

- Does not care about creating .AppleDouble forks, just keeps them in sync.
- If a share in `smb.conf` does not contain .AppleDouble item in hide or veto list, it will be added automatically.

23.3.7 shadow_copy

WARNING

THIS IS NOT A BACKUP, ARCHIVAL, OR VERSION CONTROL SOLUTION!

With Samba or Windows servers, shadow_copy is designed to be an end-user tool only. It does not replace or enhance your backup and archival solutions and should in no way be considered as such. Additionally, if you need version control, implement a version control system. You have been warned.

The shadow_copy module allows you to setup functionality that is similar to MS shadow copy services. When setup properly, this module allows Microsoft shadow copy clients to browse "shadow copies" on Samba shares. You will need to install the shadow copy client. You can get the MS shadow copy client here.[1]. Note the additional requirements for pre-Windows XP clients. I did not test this functionality with any pre-Windows XP clients. You should be able to get more information about MS Shadow Copy from the Microsoft's site[2].

[1] <http://www.microsoft.com/windowsserver2003/downloads/shadowcopyclient.mspx>

[2] <http://www.microsoft.com/windowsserver2003/techinfo/overview/scr.mspx>

The shadow_copy VFS module requires some underlying file system setup with some sort of Logical Volume Manager (LVM) such as LVM1, LVM2, or EVMS. Setting up LVM is beyond the scope of this document; however, we will outline the steps we took to test this functionality for *example purposes only.* You need to make sure the LVM implementation you choose to deploy is ready for production. Make sure you do plenty of tests.

Here are some common resources for LVM and EVMS:

- Sistina's LVM1 and LVM2[3]
- Enterprise Volume Management System (EVMS)[4]
- The LVM HOWTO[5]
- See Learning Linux LVM, Part 1[6] and Learning Linux LWM, Part 2[7] for Daniel Robbins' well-written, two part tutorial on Linux and LVM using LVM source code and reiserfs.

23.3.7.1 Shadow Copy Setup

At the time of this writing, not much testing has been done. I tested the shadow copy VFS module with a specific scenario which was not deployed in a production environment, but more as a proof of concept. The scenario involved a Samba-3 file server on Debian Sarge with an XFS file system and LVM1. I do NOT recommend you use this as a solution without doing your own due diligence with regard to all the components presented here. That said, following is an basic outline of how I got things going.

1. **Installed Operating System.** In my tests, I used Debian Sarge[8] (i.e., testing) on an XFS file system. Setting up the OS is a bit beyond the scope of this document. It is assumed that you have a working OS capable of running Samba.
2. **Install & Configure Samba.** See the Part I of this HOWTO for more detail on this. It doesn't matter if it is a Domain Controller or

[3] <http://www.sistina.com/products_lvm_download.htm>
[4] <http://evms.sourceforge.net/>
[5] <http://tldp.org/HOWTO/LVM-HOWTO/>
[6] <http://www-106.ibm.com/developerworks/linux/library/l-lvm/>
[7] <http://www-106.ibm.com/developerworks/library/l-lvm2.html>
[8] <http://www.debian.org/devel/debian-installer/>

Member File Server, but it is assumed that you have a working Samba 3.0.3 or later server running.

3. **Install & Configure LVM.** Before you can make shadow copies available to the client, you have to create the shadow copies. This is done by taking some sort of file system snapshot. Snapshots are a typical feature of Logical Volume Managers such as LVM, so we first need to have that setup.

   The following is provided as an example and will be most helpful for Debian users. Again, this was tested using the "testing" or "Sarge" distribution.

   - Install lvm10 and devfsd packages if you have not done so already. On Debian systems, you are warned of the interaction of devfs and lvm1 which requires the use of devfs filenames. Running **apt-get update && apt-get install lvm10 devfsd xfsprogs** should do the trick for this example.

   - Now you need to create a volume. You will need to create a partition (or partitions) to add to your volume. Use your favorite partitioning tool (e.g., Linux fdisk, cfdisk, etc.). The partition type should be set to 0x8e for "Linux LVM." In this example, we will use /dev/hdb1.

     Once you have the Linux LVM partition (type 0x8e), you can run a series of commands to create the LVM volume. You can use several disks and/or partitions, but we will use only one in this example. You may also need to load the kernel module with something like **modprobe lvm-mod** and set your system up to load it on reboot by adding it to (`/etc/modules`).

   - Create the physical volume with **pvcreate /dev/hdb1**

   - Create the volume group and add /dev/hda1 to it with **vgcreate shadowvol /dev/hdb1**

     You can use **vgdisplay** to review information about the volume group.

   - Now you can create the logical volume with something like **lvcreate -L400M -nsh_test shadowvol**

This creates the logical volume of 400 MBs named "sh_test" in the volume group we created called shadowvol. If everything is working so far, you should see them in `/dev/shadowvol`.

- Now we should be ready to format the logical volume we named sh_test with **mkfs.xfs /dev/shadowvol/sh_test**

  You can format the logical volume with any file system you choose, but make sure to use one that allows you to take advantage of the additional features of LVM such as freezing, resizing, and growing your file systems.

  Now we have an LVM volume where we can play with the shadow_copy VFS module.

- Now we need to prepare the directory with something like

```
root#  mkdir -p /data/shadow_share
```

  or whatever you want to name your shadow copy-enabled Samba share. Make sure you set the permissions so that you can use it. If in doubt, use **chmod 777 /data/shadow_share** and tighten the permissions once you get things working.

- Mount the LVM volume using something like **mount /dev/shadowvol/sh_test /data/shadow_share**

  You may also want to edit your `/etc/fstab` so that this partition mounts during the system boot.

4. **Install & Configure the shadow_copy VFS Module.** Finally we get to the actual shadow_copy VFS module. The shadow_copy VFS module should be available in Samba 3.0.3 and higher. The smb.conf configuration is pretty standard. Here is our example of a share configured with the shadow_copy VFS module:

5. **Create Snapshots and Make Them Available to shadow_copy.so.** Before you can browse the shadow copies, you must create them and mount them. This will most likely be done with a script that runs as a cron job. With this particular solution, the shadow_copy VFS module is used to browse LVM snapshots. Those snapshots are not created by the module. They are not made available by the module either. This

Example 23.3.1 Share With shadow_copy VFS

```
[shadow_share]
        comment = Shadow Copy Enabled Share
        path = /data/shadow_share
        vfs objects = shadow_copy
        writeable = yes
        browseable = yes
```

module allows the shadow copy-enabled client to browse the snapshots you take and make available.

Here is a simple script used to create and mount the snapshots:

```
#!/bin/bash
# This is a test, this is only a test
SNAPNAME=`date +%Y.%m.%d-%H.%M.%S`
xfs_freeze -f /data/shadow_share/
lvcreate -L10M -s -n $SNAPNAME /dev/shadowvol/sh_test
xfs_freeze -u /data/shadow_share/
mkdir /data/shadow_share/@GMT-$SNAPNAME
mount /dev/shadowvol/$SNAPNAME \
       /data/shadow_share/@GMT-$SNAPNAME -onouuid,ro
```

Note that the script does not handle other things like remounting snapshots on reboot.

6. **Test From Client.** To test, you will need to install the shadow copy client which you can obtain from the Microsoft web site.[9] I only tested this with an XP client so your results may vary with other pre-XP clients. Once installed, with your XP client you can right-click on specific files or in the empty space of the shadow_share and view the "properties." If anything has changed, then you will see it on the "Previous Versions" tab of the properties window.

[9] <http://www.microsoft.com/windowsserver2003/downloads/shadowcopyclient.mspx>

23.4 VFS Modules Available Elsewhere

This section contains a listing of various other VFS modules that have been posted but do not currently reside in the Samba CVS tree for one reason or another (e.g., it is easy for the maintainer to have his or her own CVS tree).

No statements about the stability or functionality of any module should be implied due to its presence here.

23.4.1 DatabaseFS

URL: Taylors University DatabaeFS[10]

By Eric Lorimer.[11]

I have created a VFS module that implements a fairly complete read-only filesystem. It presents information from a database as a filesystem in a modular and generic way to allow different databases to be used. (Originally designed for organizing MP3s under directories such as "Artists," "Song Keywords," and so on. I have since easily applied it to a student roster database.) The directory structure is stored in the database itself and the module makes no assumptions about the database structure beyond the table it requires to run.

Any feedback would be appreciated: comments, suggestions, patches, and so on. If nothing else, it might prove useful for someone else who wishes to create a virtual filesystem.

23.4.2 vscan

URL: Open Anti-Virus vscan[12]

samba-vscan is a proof-of-concept module for Samba, which provides on-access anti-virus support for files shared using Samba. samba-vscan supports various virus scanners and is maintained by Rainer Link.

[10] <http://www.css.tayloru.edu/~elorimer/databasefs/index.php>
[11] <mailto:elorimer@css.tayloru.edu>
[12] <http://www.openantivirus.org/projects.php#samba-vscan>

23.4.3 vscan-clamav

Samba users have been using the RPMS from SerNet without a problem. OpenSUSE Linux users have also used the vscan scanner for quite some time with excellent results. It does impact overall write performance though.

The following share stanza is a good guide for those wanting to configure vscan-clamav:

```
[share]
vfs objects = vscan-clamav
vscan-clamav: config-file = /etc/samba/vscan-clamav.conf
```

The following example of the `vscan-clamav.conf` file may help to get this fully operational:

```
VFS: Vscan ClamAV Control File
#
# /etc/samba/vscan-clamav.conf
#

[samba-vscan]
; run-time configuration for vscan-samba using
; clamd
; all options are set to default values

; do not scan files larger than X bytes. If set to 0 (default),
; this feature is disable (i.e. all files are scanned)
max file size = 10485760

; log all file access (yes/no). If set to yes, every access will
; be logged. If set to no (default), only access to infected files
; will be logged
verbose file logging = no

; if set to yes (default), a file will be scanned while opening
scan on open = yes
; if set to yes, a file will be scanned while closing (default is yes)
scan on close = yes
```

```
; if communication to clamd fails, should access to file denied?
; (default: yes)
deny access on error = no

; if daemon failes with a minor error (corruption, etc.),
; should access to file denied?
; (default: yes)
deny access on minor error = no

; send a warning message via Windows Messenger service
; when virus is found?
; (default: yes)
send warning message = yes

; what to do with an infected file
; quarantine: try to move to quantine directory
; delete:     delete infected file
; nothing:    do nothing (default)
infected file action = quarantine

; where to put infected files - you really want to change this!
quarantine directory  = /opt/clamav/quarantine
; prefix for files in quarantine
quarantine prefix = vir-

; as Windows tries to open a file multiple time in a (very) short time
; of period, samba-vscan use a last recently used file mechanism to avoid
; multiple scans of a file. This setting specified the maximum number of
; elements of the last recently used file list. (default: 100)
max lru files entries = 100

; an entry is invalidad after lru file entry lifetime (in seconds).
; (Default: 5)
lru file entry lifetime = 5

; exclude files from being scanned based on the MIME-type! Semi-colon
; seperated list (default: empty list). Use this with care!
exclude file types =
```

```
; socket name of clamd (default: /var/run/clamd). Setting will be ignored if
; libclamav is used
clamd socket name = /tmp/clamd

; limits, if vscan-clamav was build for using the clamav library (libclamav)
; instead of clamd

; maximum number of files in archive (default: 1000)
libclamav max files in archive = 1000

; maximum archived file size, in bytes (default: 10 MB)
libclamav max archived file size = 5242880

; maximum recursion level (default: 5)
libclamav max recursion level = 5
```

Obviously, a running clam daemon is necessary for this to work. This is a working example for me using ClamAV. The ClamAV documentation should provide additional configuration examples. On your system these may be located under the `/usr/share/doc/` directory. Some examples may also target other virus scanners, any of which can be used.

Chapter 24

WINBIND: USE OF DOMAIN ACCOUNTS

24.1 Features and Benefits

Integration of UNIX and Microsoft Windows NT through a unified logon has been considered a "holy grail" in heterogeneous computing environments for a long time.

There is one other facility without which UNIX and Microsoft Windows network interoperability would suffer greatly. It is imperative that there be a mechanism for sharing files across UNIX systems and to be able to assign domain user and group ownerships with integrity.

winbind is a component of the Samba suite of programs that solves the unified logon problem. Winbind uses a UNIX implementation of Microsoft RPC calls, Pluggable Authentication Modules (PAMs), and the name service switch (NSS) to allow Windows NT domain users to appear and operate as UNIX users on a UNIX machine. This chapter describes the Winbind system, the functionality it provides, how it is configured, and how it works internally.

Winbind provides three separate functions:

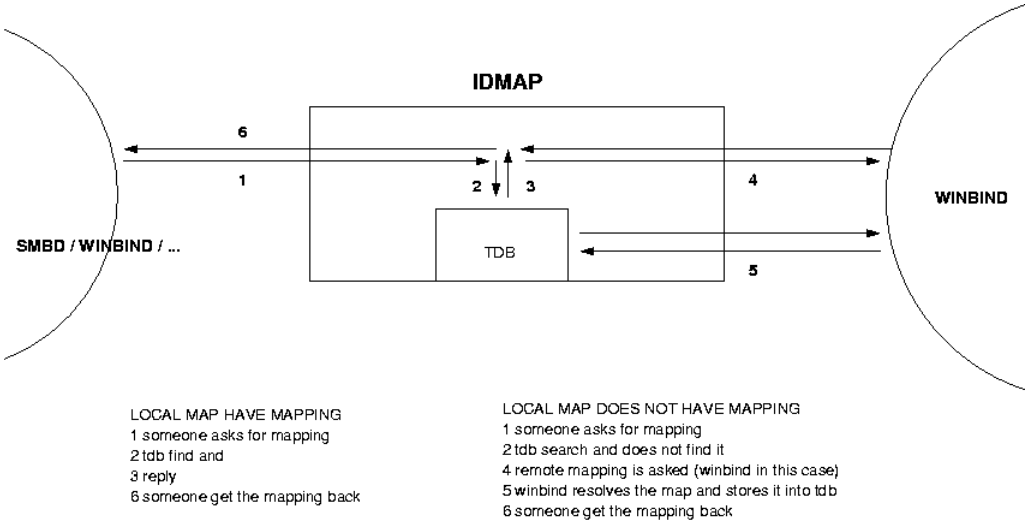
- Authentication of user credentials (via PAM). This makes it possible to log onto a UNIX/Linux system using user and group accounts from a Windows NT4 (including a Samba domain) or an Active Directory domain.
- Identity resolution (via NSS). This is the default when winbind is not used.

- Winbind maintains a database called winbind_idmap.tdb in which it stores mappings between UNIX UIDs, GIDs, and NT SIDs. This mapping is used only for users and groups that do not have a local UID/GID. It stores the UID/GID allocated from the idmap uid/gid range that it has mapped to the NT SID. If *`idmap backend`* has been specified as `ldap:ldap://hostname[:389]`, then instead of using a local mapping, Winbind will obtain this information from the LDAP database.

NOTE

If **winbindd** is not running, smbd (which calls **winbindd**) will fall back to using purely local information from `/etc/passwd` and `/etc/group` and no dynamic mapping will be used. On an operating system that has been enabled with the NSS, the resolution of user and group information will be accomplished via NSS.

Figure 24.1 Winbind Idmap



24.2 Introduction

It is well known that UNIX and Microsoft Windows NT have different models for representing user and group information and use different technologies for implementing them. This fact has made it difficult to integrate the two systems in a satisfactory manner.

One common solution in use today has been to create identically named user accounts on both the UNIX and Windows systems and use the Samba suite of programs to provide file and print services between the two. This solution is far from perfect, however, because adding and deleting users on both sets of machines becomes a chore, and two sets of passwords are required — both of which can lead to synchronization problems between the UNIX and Windows systems and confusion for users.

We divide the unified logon problem for UNIX machines into three smaller problems:

- Obtaining Windows NT user and group information.
- Authenticating Windows NT users.
- Password changing for Windows NT users.

Ideally, a prospective solution to the unified logon problem would satisfy all the above components without duplication of information on the UNIX machines and without creating additional tasks for the system administrator when maintaining users and groups on either system. The Winbind system provides a simple and elegant solution to all three components of the unified logon problem.

24.3 What Winbind Provides

Winbind unifies UNIX and Windows NT account management by allowing a UNIX box to become a full member of an NT domain. Once this is done, the UNIX box will see NT users and groups as if they were "native" UNIX users and groups, allowing the NT domain to be used in much the same manner that NIS+ is used within UNIX-only environments.

The end result is that whenever a program on the UNIX machine asks the operating system to look up a user or group name, the query will be resolved by asking the NT domain controller for the specified domain to do

the lookup. Because Winbind hooks into the operating system at a low level (via the NSS name resolution modules in the C library), this redirection to the NT domain controller is completely transparent.

Users on the UNIX machine can then use NT user and group names as they would "native" UNIX names. They can chown files so they are owned by NT domain users or even login to the UNIX machine and run a UNIX X-Window session as a domain user.

The only obvious indication that Winbind is being used is that user and group names take the form `DOMAIN\user` and `DOMAIN\group`. This is necessary because it allows Winbind to determine that redirection to a domain controller is wanted for a particular lookup and which trusted domain is being referenced.

Additionally, Winbind provides an authentication service that hooks into the PAM system to provide authentication via an NT domain to any PAM-enabled applications. This capability solves the problem of synchronizing passwords between systems, since all passwords are stored in a single location (on the domain controller).

24.3.1 Target Uses

Winbind is targeted at organizations that have an existing NT-based domain infrastructure into which they wish to put UNIX workstations or servers. Winbind will allow these organizations to deploy UNIX workstations without having to maintain a separate account infrastructure. This greatly simplifies the administrative overhead of deploying UNIX workstations into an NT-based organization.

Another interesting way in which we expect Winbind to be used is as a central part of UNIX-based appliances. Appliances that provide file and print services to Microsoft-based networks will be able to use Winbind to provide seamless integration of the appliance into the domain.

24.3.2 Handling of Foreign SIDs

The term *foreign SID* is often met with the reaction that it is not relevant to a particular environment. The following documents an interchange that took place on the Samba mailing list. It is a good example of the confusion often expressed regarding the use of winbind.

Fact: Winbind is needed to handle users who use workstations that are NOT part of the local domain.

Response: "Why? I've used Samba with workstations that are not part of my domains lots of times without using winbind. I thought winbind was for using Samba as a member server in a domain controlled by another Samba/Windows PDC."

If the Samba server will be accessed from a domain other than the local Samba domain, or if there will be access from machines that are not local domain members, winbind will permit the allocation of UIDs and GIDs from the assigned pool that will keep the identity of the foreign user separate from users that are members of the Samba domain.

This means that winbind is eminently useful in cases where a single Samba PDC on a local network is combined with both domain member and domain non-member workstations. If winbind is not used, the user george on a Windows workstation that is not a domain member will be able to access the files of a user called george in the account database of the Samba server that is acting as a PDC. When winbind is used, the default condition is that the local user george will be treated as the account DOMAIN\george and the foreign (non-member of the domain) account will be treated as MACHINE\george because each has a different SID.

24.4 How Winbind Works

The Winbind system is designed around a client/server architecture. A long-running **winbindd** daemon listens on a UNIX domain socket waiting for requests to arrive. These requests are generated by the NSS and PAM clients and are processed sequentially.

The technologies used to implement Winbind are described in detail below.

24.4.1 Microsoft Remote Procedure Calls

Over the last few years, efforts have been underway by various Samba Team members to implement various aspects of the Microsoft Remote Procedure Call (MSRPC) system. This system is used for most network-related operations between Windows NT machines, including remote management, user authentication, and print spooling. Although initially this work was done to

aid the implementation of Primary Domain Controller (PDC) functionality in Samba, it has also yielded a body of code that can be used for other purposes.

Winbind uses various MSRPC calls to enumerate domain users and groups and to obtain detailed information about individual users or groups. Other MSRPC calls can be used to authenticate NT domain users and to change user passwords. By directly querying a Windows PDC for user and group information, Winbind maps the NT account information onto UNIX user and group names.

24.4.2 Microsoft Active Directory Services

Since late 2001, Samba has gained the ability to interact with Microsoft Windows 2000 using its "native mode" protocols rather than the NT4 RPC services. Using LDAP and Kerberos, a domain member running Winbind can enumerate users and groups in exactly the same way as a Windows 200x client would, and in so doing provide a much more efficient and effective Winbind implementation.

24.4.3 Name Service Switch

The NSS is a feature that is present in many UNIX operating systems. It allows system information such as hostnames, mail aliases, and user information to be resolved from different sources. For example, a standalone UNIX workstation may resolve system information from a series of flat files stored on the local file system. A networked workstation may first attempt to resolve system information from local files, and then consult an NIS database for user information or a DNS server for hostname information.

The NSS application programming interface allows Winbind to present itself as a source of system information when resolving UNIX usernames and groups. Winbind uses this interface and information obtained from a Windows NT server using MSRPC calls to provide a new source of account enumeration. Using standard UNIX library calls, you can enumerate the users and groups on a UNIX machine running Winbind and see all users and groups in an NT domain plus any trusted domain as though they were local users and groups.

The primary control file for NSS is `/etc/nsswitch.conf`. When a UNIX application makes a request to do a lookup, the C library looks in `/etc/nsswitch.conf` for a line that matches the service type being requested; for example, the "passwd" service type is used when user or group names are looked up. This config line specifies which implementations of that service should be tried and in what order. If the passwd config line is:

```
passwd: files example
```

then the C library will first load a module called `/lib/libnss_files.so` followed by the module `/lib/libnss_example.so`. The C library will dynamically load each of these modules in turn and call resolver functions within the modules to try to resolve the request. Once the request is resolved, the C library returns the result to the application.

This NSS interface provides an easy way for Winbind to hook into the operating system. All that needs to be done is to put `libnss_winbind.so` in `/lib/` then add "winbind" into `/etc/nsswitch.conf` at the appropriate place. The C library will then call Winbind to resolve user and group names.

24.4.4 Pluggable Authentication Modules

PAMs provide a system for abstracting authentication and authorization technologies. With a PAM module, it is possible to specify different authentication methods for different system applications without having to recompile these applications. PAM is also useful for implementing a particular policy for authorization. For example, a system administrator may only allow console logins from users stored in the local password file but only allow users resolved from an NIS database to log in over the network.

Winbind uses the authentication management and password management PAM interface to integrate Windows NT users into a UNIX system. This allows Windows NT users to log in to a UNIX machine and be authenticated against a suitable PDC. These users can also change their passwords and have this change take effect directly on the PDC.

PAM is configured by providing control files in the directory `/etc/pam.d/` for each of the services that require authentication. When an authentication request is made by an application, the PAM code in the C library looks up this control file to determine what modules to load to do the authentication

check and in what order. This interface makes adding a new authentication service for Winbind very easy: simply copy the `pam_winbind.so` module to `/lib/security/`, and the PAM control files for relevant services are updated to allow authentication via Winbind. See the PAM documentation in Chapter 28, "PAM-Based Distributed Authentication", for more information.

24.4.5 User and Group ID Allocation

When a user or group is created under Windows NT/200x, it is allocated a numerical relative identifier (RID). This is slightly different from UNIX, which has a range of numbers that are used to identify users and the same range used to identify groups. It is Winbind's job to convert RIDs to UNIX ID numbers and vice versa. When Winbind is configured, it is given part of the UNIX user ID space and a part of the UNIX group ID space in which to store Windows NT users and groups. If a Windows NT user is resolved for the first time, it is allocated the next UNIX ID from the range. The same process applies for Windows NT groups. Over time, Winbind will have mapped all Windows NT users and groups to UNIX user IDs and group IDs.

The results of this mapping are stored persistently in an ID mapping database held in a tdb database. This ensures that RIDs are mapped to UNIX IDs in a consistent way.

24.4.6 Result Caching

An active directory system can generate a lot of user and group name lookups. To reduce the network cost of these lookups, Winbind uses a caching scheme based on the SAM sequence number supplied by NT domain controllers. User or group information returned by a PDC is cached by Winbind along with a sequence number also returned by the PDC. This sequence number is incremented by Windows NT whenever any user or group information is modified. If a cached entry has expired, the sequence number is requested from the PDC and compared against the sequence number of the cached entry. If the sequence numbers do not match, then the cached information is discarded and up-to-date information is requested directly from the PDC.

24.5 Installation and Configuration

24.5.1 Introduction

This section describes the procedures used to get Winbind up and running. Winbind is capable of providing access and authentication control for Windows Domain users through an NT or Windows 200x PDC for regular services, such as telnet and ftp, as well for Samba services.

- *Why should I do this?*

  This allows the Samba administrator to rely on the authentication mechanisms on the Windows NT/200x PDC for the authentication of domain members. Windows NT/200x users no longer need to have separate accounts on the Samba server.

- *Who should be reading this document?*

  This document is designed for system administrators. If you are implementing Samba on a file server and wish to (fairly easily) integrate existing Windows NT/200x users from your PDC onto the Samba server, this document is for you.

24.5.2 Requirements

If you have a Samba configuration file that you are currently using, *BACK IT UP!* If your system already uses PAM, *back up the `/etc/pam.d` directory contents!* If you haven't already made a boot disk, *MAKE ONE NOW!*

Messing with the PAM configuration files can make it nearly impossible to log in to your machine. That's why you want to be able to boot back into your machine in single-user mode and restore your `/etc/pam.d` to the original state it was in if you get frustrated with the way things are going.

The latest version of Samba-3 includes a functioning winbindd daemon. Please refer to the main Samba Web page[1], or better yet, your closest Samba mirror site for instructions on downloading the source code.

To allow domain users the ability to access Samba shares and files, as well as potentially other services provided by your Samba machine, PAM must be set up properly on your machine. In order to compile the Winbind modules,

[1] <http://samba.org/>

you should have at least the PAM development libraries installed on your system. Please refer to the PAM Web site <http://www.kernel.org/pub/linux/libs/pam/>.

24.5.3 Testing Things Out

Before starting, it is probably best to kill off all the Samba-related daemons running on your server. Kill off all smbd, nmbd, and winbindd processes that may be running. To use PAM, make sure that you have the standard PAM package that supplies the /etc/pam.d directory structure, including the PAM modules that are used by PAM-aware services, several PAM libraries, and the /usr/doc and /usr/man entries for PAM. Winbind is built better in Samba if the pam-devel package is also installed. This package includes the header files needed to compile PAM-aware applications.

24.5.3.1 Configure nsswitch.conf and the Winbind Libraries on Linux and Solaris

PAM is a standard component of most current generation UNIX/Linux systems. Unfortunately, few systems install the pam-devel libraries that are needed to build PAM-enabled Samba. Additionally, Samba-3 may auto-install the Winbind files into their correct locations on your system, so before you get too far down the track, be sure to check if the following configuration is really necessary. You may only need to configure /etc/nsswitch.conf.

The libraries needed to run the winbindd daemon through nsswitch need to be copied to their proper locations:

```
root# cp ../samba/source/nsswitch/libnss_winbind.so /lib
```

I also found it necessary to make the following symbolic link:

```
root# ln -s /lib/libnss_winbind.so /lib/libnss_winbind.so.2
```

And, in the case of Sun Solaris:

```
root# ln -s /usr/lib/libnss_winbind.so /usr/lib/libnss_winbind.so.1
root# ln -s /usr/lib/libnss_winbind.so /usr/lib/nss_winbind.so.1
```

```
root# ln -s /usr/lib/libnss_winbind.so /usr/lib/nss_winbind.so.2
```

As root, edit `/etc/nsswitch.conf` to allow user and group entries to be visible from the winbindd daemon. My `/etc/nsswitch.conf` file looked like this after editing:

```
passwd:     files winbind
shadow:     files
group:      files winbind
```

The libraries needed by the **winbindd** daemon will be automatically entered into the **ldconfig** cache the next time your system reboots, but it is faster (and you do not need to reboot) if you do it manually:

```
root# /sbin/ldconfig -v | grep winbind
```

This makes `libnss_winbind` available to winbindd and reports the current search path that is used by the dynamic link loader. The use of the **grep** filters the output of the **ldconfig** command so that we may see proof that this library is indeed recognized by the dynamic link loader.

The Sun Solaris dynamic link loader management tool is called **crle**. The use of this tool is necessary to instruct the dynamic link loader to search directories that contain library files that were not supplied as part of the original operating system platform. The following example shows how to use this tool to add the directory `/usr/local/lib` to the dynamic link loader's search path:

```
root#  crle -u -l /usr/lib:/usr/local/lib
```

When executed without arguments, **crle** reports the current dynamic link loader configuration. This is demonstrated here:

```
root#  crle

Configuration file [version 4]: /var/ld/ld.config
```

```
  Default Library Path (ELF):   /lib:/usr/lib:/usr/local/lib
  Trusted Directories (ELF):    /lib/secure:/usr/lib/secure  (system default)

Command line:
  crle -c /var/ld/ld.config -l /lib:/usr/lib:/usr/local/lib
```

From this it is apparent that the `/usr/local/lib` directory is included in the search dynamic link libraries in order to satisfy object module dependencies.

24.5.3.2 NSS Winbind on AIX

(This section is only for those running AIX.)

The Winbind AIX identification module gets built as `libnss_winbind.so` in the nsswitch directory of the Samba source. This file can be copied to `/usr/lib/security`, and the AIX naming convention would indicate that it should be named WINBIND. A stanza like the following:

```
WINBIND:
        program = /usr/lib/security/WINBIND
        options = authonly
```

can then be added to `/usr/lib/security/methods.cfg`. This module only supports identification, but there have been reports of success using the standard Winbind PAM module for authentication. Use caution configuring loadable authentication modules, since misconfiguration can make it impossible to log on to the system. Information regarding the AIX authentication module API can be found in the "Kernel Extensions and Device Support Programming Concepts for AIX" document that describes the Loadable Authentication Module Programming Interface[2] for AIX. Further information on administering the modules can be found in the System Management Guide: Operating System and Devices.[3]

[2] <http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/aixprggd/kernextc/sec_load_mod.htm>

[3] <http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/aixbman/baseadmn/iandaadmin.htm>

24.5.3.3 Configure smb.conf

Several parameters are needed in the `smb.conf` file to control the behavior of winbindd. These are described in more detail in the winbindd(8) man page. My `smb.conf` file, as shown in Example 24.5.1, was modified to include the necessary entries in the [global] section.

Example 24.5.1 smb.conf for Winbind Setup

```
[global]
 # separate domain and username with '\textbackslash ',  ↩
    like DOMAIN\textbackslash username
        winbind separator = \
 # use uids from 10000 to 20000 for domain users
        idmap uid = 10000-20000
 # use gids from 10000 to 20000 for domain groups
        idmap gid = 10000-20000
 # allow enumeration of winbind users and groups
        winbind enum users = yes
        winbind enum groups = yes
 # give winbind users a real shell (only needed if they  ↩
    have telnet access)
        template homedir = /home/winnt/%D/%U
        template shell = /bin/bash
```

24.5.3.4 Join the Samba Server to the PDC Domain

All machines that will participate in domain security should be members of the domain. This applies also to the PDC and all BDCs.

The process of joining a domain requires the use of the **net rpc join** command. This process communicates with the domain controller it will register with (usually the PDC) via MS DCE RPC. This means, of course, that the **smbd** process must be running on the target domain controller. It is therefore necessary to temporarily start Samba on a PDC so that it can join its own domain.

Enter the following command to make the Samba server join the domain, where *PDC* is the name of your PDC and *Administrator* is a domain user who has administrative privileges in the domain.

NOTE

Before attempting to join a machine to the domain, verify that Samba is running on the target domain controller (usually PDC) and that it is capable of being reached via ports 137/udp, 135/tcp, 139/tcp, and 445/tcp (if Samba or Windows Server 2Kx).

The use of the **net rpc join** facility is shown here:

```
root# /usr/local/samba/bin/net rpc join -S PDC -U Administrator
```

The proper response to the command should be "Joined the domain *DOMAIN*" where *DOMAIN* is your domain name.

24.5.3.5 Starting and Testing the winbindd Daemon

Eventually, you will want to modify your Samba startup script to automatically invoke the winbindd daemon when the other parts of Samba start, but it is possible to test out just the Winbind portion first. To start up Winbind services, enter the following command as root:

```
root# /usr/local/samba/sbin/winbindd
```

Use the appropriate path to the location of the **winbindd** executable file.

NOTE

The command to start up Winbind services assumes that Samba has been installed in the `/usr/local/samba` directory tree. You may need to search for the location of Samba files if this is not the location of **winbindd** on your system.

I'm always paranoid and like to make sure the daemon is really running.

```
root# ps -ae | grep winbindd
```

This command should produce output like the following if the daemon is running.

```
3025 ?        00:00:00 winbindd
```

Now, for the real test, try to get some information about the users on your PDC:

```
root# /usr/local/samba/bin/wbinfo -u
```

This should echo back a list of users on your Windows users on your PDC. For example, I get the following response:

```
CEO\Administrator
CEO\burdell
CEO\Guest
CEO\jt-ad
CEO\krbtgt
CEO\TsInternetUser
```

Obviously, I have named my domain "CEO" and my *winbind separator* is "\".

You can do the same sort of thing to get group information from the PDC:

```
root# /usr/local/samba/bin/wbinfo -g
CEO\Domain Admins
CEO\Domain Users
CEO\Domain Guests
CEO\Domain Computers
CEO\Domain Controllers
CEO\Cert Publishers
CEO\Schema Admins
CEO\Enterprise Admins
CEO\Group Policy Creator Owners
```

The function **getent** can now be used to get unified lists of both local and PDC users and groups. Try the following command:

```
root# getent passwd
```

You should get a list that looks like your `/etc/passwd` list followed by the domain users with their new UIDs, GIDs, home directories, and default shells.

The same thing can be done for groups with the command:

```
root# getent group
```

24.5.3.6 Fix the init.d Startup Scripts

Linux The winbindd daemon needs to start up after the smbd and nmbd daemons are running. To accomplish this task, you need to modify the startup scripts of your system. They are located at `/etc/init.d/smb` in Red Hat Linux and in `/etc/init.d/samba` in Debian Linux. Edit your script to add commands to invoke this daemon in the proper sequence. My startup script starts up smbd, nmbd, and winbindd from the `/usr/local/samba/bin` directory directly. The **start** function in the script looks like this:

```
start() {
        KIND="SMB"
        echo -n $"Starting $KIND services: "
        daemon /usr/local/samba/bin/smbd $SMBDOPTIONS
        RETVAL=$?
        echo
        KIND="NMB"
        echo -n $"Starting $KIND services: "
        daemon /usr/local/samba/bin/nmbd $NMBDOPTIONS
        RETVAL2=$?
        echo
        KIND="Winbind"
        echo -n $"Starting $KIND services: "
        daemon /usr/local/samba/sbin/winbindd
        RETVAL3=$?
        echo
        [ $RETVAL -eq 0 -a $RETVAL2 -eq 0 -a $RETVAL3 -eq 0 ] && \
      touch /var/lock/subsys/smb || RETVAL=1
        return $RETVAL
}
```

If you would like to run winbindd in dual daemon mode, replace the line:

```
        daemon /usr/local/samba/sbin/winbindd
```

in the example above with:

```
        daemon /usr/local/samba/sbin/winbindd -D
```

.

The **stop** function has a corresponding entry to shut down the services and looks like this:

```
stop() {
        KIND="SMB"
```

```
        echo -n $"Shutting down $KIND services: "
        killproc smbd
        RETVAL=$?
        echo
        KIND="NMB"
        echo -n $"Shutting down $KIND services: "
        killproc nmbd
        RETVAL2=$?
        echo
        KIND="Winbind"
        echo -n $"Shutting down $KIND services: "
        killproc winbindd
        RETVAL3=$?
        [ $RETVAL -eq 0 -a $RETVAL2 -eq 0 -a $RETVAL3 -eq 0 ] && \
       rm -f /var/lock/subsys/smb
        echo ""
        return $RETVAL
}
```

Solaris Winbind does not work on Solaris 9; see Section 43.6.2 for details.

On Solaris, you need to modify the `/etc/init.d/samba.server` startup script. It usually only starts smbd and nmbd but should now start winbindd, too. If you have Samba installed in `/usr/local/samba/bin`, the file could contains something like this:

```
   ##
   ## samba.server
   ##

   if [ ! -d /usr/bin ]
   then                    # /usr not mounted
      exit
   fi

   killproc() {            # kill the named process(es)
      pid=`/usr/bin/ps -e |
           /usr/bin/grep -w $1 |
           /usr/bin/sed -e 's/^  *//' -e 's/ .*//'`
```

```
   [ "$pid" != "" ] && kill $pid
}

# Start/stop processes required for Samba server

case "$1" in

'start')
#
# Edit these lines to suit your installation (paths, workgroup, host)
#
echo Starting SMBD
   /usr/local/samba/bin/smbd -D -s \
   /usr/local/samba/smb.conf

echo Starting NMBD
   /usr/local/samba/bin/nmbd -D -l \
   /usr/local/samba/var/log -s /usr/local/samba/smb.conf

echo Starting Winbind Daemon
   /usr/local/samba/sbin/winbindd
   ;;

'stop')
   killproc nmbd
   killproc smbd
   killproc winbindd
   ;;

*)
   echo "Usage: /etc/init.d/samba.server { start | stop }"
   ;;
esac
```

Again, if you would like to run Samba in dual daemon mode, replace:

```
/usr/local/samba/sbin/winbindd
```

in the script above with:

```
/usr/local/samba/sbin/winbindd -D
```

Restarting If you restart the smbd, nmbd, and winbindd daemons at this point, you should be able to connect to the Samba server as a domain member just as if you were a local user.

24.5.3.7 Configure Winbind and PAM

If you have made it this far, you know that **winbindd** and Samba are working together. If you want to use Winbind to provide authentication for other services, keep reading. The PAM configuration files need to be altered in this step. (Did you remember to make backups of your original `/etc/pam.d` files? If not, do it now.)

You will need a PAM module to use winbindd with these other services. This module will be compiled in the `../source/nsswitch` directory by invoking the command:

```
root# make nsswitch/pam_winbind.so
```

from the `../source` directory. The `pam_winbind.so` file should be copied to the location of your other PAM security modules. On my Red Hat system, this was the `/lib/security` directory. On Solaris, the PAM security modules reside in `/usr/lib/security`.

```
root# cp ../samba/source/nsswitch/pam_winbind.so /lib/security
```

Linux/FreeBSD-Specific PAM Configuration The `/etc/pam.d/samba` file does not need to be changed. I just left this file as it was:

```
auth    required  /lib/security/pam_stack.so service=system-auth
account required  /lib/security/pam_stack.so service=system-auth
```

The other services that I modified to allow the use of Winbind as an authentication service were the normal login on the console (or a terminal session), telnet logins, and ftp service. In order to enable these services, you may first need to change the entries in `/etc/xinetd.d` (or `/etc/inetd.conf`). Red Hat Linux 7.1 and later uses the new xinetd.d structure, in this case you need to change the lines in `/etc/xinetd.d/telnet` and `/etc/xinetd.d/wu-ftp` from:

```
enable = no
```

to

```
enable = yes
```

For ftp services to work properly, you will also need to either have individual directories for the domain users already present on the server or change the home directory template to a general directory for all domain users. These can be easily set using the `smb.conf` global entry *template homedir*.

> **Note**
>
> The directory in *template homedir* is not created automatically! Use pam_mkhomedir or pre-create the directories of users to make sure users can log in on UNIX with their own home directory.

The `/etc/pam.d/ftp` file can be changed to allow Winbind ftp access in a manner similar to the samba file. My `/etc/pam.d/ftp` file was changed to look like this:

```
auth       required     /lib/security/pam_listfile.so item=user sense=deny \
    file=/etc/ftpusers onerr=succeed
auth       sufficient   /lib/security/pam_winbind.so
auth       required     /lib/security/pam_stack.so service=system-auth
```

```
auth       required     /lib/security/pam_shells.so
account    sufficient   /lib/security/pam_winbind.so
account    required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
```

The `/etc/pam.d/login` file can be changed in nearly the same way. It now looks like this:

```
auth       required     /lib/security/pam_securetty.so
auth       sufficient   /lib/security/pam_winbind.so
auth       sufficient   /lib/security/pam_unix.so use_first_pass
auth       required     /lib/security/pam_stack.so service=system-auth
auth       required     /lib/security/pam_nologin.so
account    sufficient   /lib/security/pam_winbind.so
account    required     /lib/security/pam_stack.so service=system-auth
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
session    optional     /lib/security/pam_console.so
```

In this case, I added the

```
auth sufficient /lib/security/pam_winbind.so
```

lines as before, but also added the

```
required pam_securetty.so
```

above it to disallow root logins over the network. I also added a

```
sufficient /lib/security/pam_unix.so
use_first_pass
```

line after the **winbind.so** line to get rid of annoying double prompts for passwords.

Solaris-Specific Configuration The `/etc/pam.conf` needs to be changed. I changed this file so my Domain users can log on both locally as well as with telnet. The following are the changes that I made. You can customize the `pam.conf` file as per your requirements, but be sure of those changes because in the worst case it will leave your system nearly impossible to boot.

```
#
#ident "@(#)pam.conf 1.14 99/09/16 SMI"
#
# Copyright (c) 1996-1999, Sun Microsystems, Inc.
# All Rights Reserved.
#
# PAM configuration
#
# Authentication management
#
login   auth required   /usr/lib/security/pam_winbind.so
login auth required  /usr/lib/security/$ISA/pam_unix.so.1 try_first_pass
login auth required  /usr/lib/security/$ISA/pam_dial_auth.so.1 try_first_pass
#
rlogin  auth sufficient /usr/lib/security/pam_winbind.so
rlogin  auth sufficient /usr/lib/security/$ISA/pam_rhosts_auth.so.1
rlogin auth required  /usr/lib/security/$ISA/pam_unix.so.1 try_first_pass
#
dtlogin auth sufficient /usr/lib/security/pam_winbind.so
dtlogin auth required  /usr/lib/security/$ISA/pam_unix.so.1 try_first_pass
#
rsh auth required /usr/lib/security/$ISA/pam_rhosts_auth.so.1
other   auth sufficient /usr/lib/security/pam_winbind.so
other auth required /usr/lib/security/$ISA/pam_unix.so.1 try_first_pass
#
# Account management
#
login   account sufficient      /usr/lib/security/pam_winbind.so
login account requisite /usr/lib/security/$ISA/pam_roles.so.1
login account required /usr/lib/security/$ISA/pam_unix.so.1
#
dtlogin account sufficient      /usr/lib/security/pam_winbind.so
dtlogin account requisite /usr/lib/security/$ISA/pam_roles.so.1
dtlogin account required /usr/lib/security/$ISA/pam_unix.so.1
#
other   account sufficient      /usr/lib/security/pam_winbind.so
other account requisite /usr/lib/security/$ISA/pam_roles.so.1
other account required /usr/lib/security/$ISA/pam_unix.so.1
#
```

```
# Session management
#
other session required /usr/lib/security/$ISA/pam_unix.so.1
#
# Password management
#
#other   password sufficient     /usr/lib/security/pam_winbind.so
other password required /usr/lib/security/$ISA/pam_unix.so.1
dtsession auth required /usr/lib/security/$ISA/pam_unix.so.1
#
# Support for Kerberos V5 authentication (uncomment to use Kerberos)
#
#rlogin auth optional /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
#login auth optional /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
#dtlogin auth optional /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
#other auth optional /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
#dtlogin account optional /usr/lib/security/$ISA/pam_krb5.so.1
#other account optional /usr/lib/security/$ISA/pam_krb5.so.1
#other session optional /usr/lib/security/$ISA/pam_krb5.so.1
#other password optional /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
```

I also added a *try_first_pass* line after the `winbind.so` line to get rid of annoying double prompts for passwords.

Now restart your Samba and try connecting through your application that you configured in the pam.conf.

24.6 Conclusion

The Winbind system, through the use of the NSS, PAMs, and appropriate Microsoft RPC calls, have allowed us to provide seamless integration of Microsoft Windows NT domain users on a UNIX system. The result is a great reduction in the administrative cost of running a mixed UNIX and NT network.

24.7 Common Errors

Winbind has a number of limitations in its current released version that we hope to overcome in future releases:

- Winbind is currently only available for the Linux, Solaris, AIX, and IRIX operating systems, although ports to other operating systems are certainly possible. For such ports to be feasible, we require the C library of the target operating system to support the NSS and PAM systems. This is becoming more common as NSS and PAM gain support among UNIX vendors.
- The mappings of Windows NT RIDs to UNIX IDs is not made algorithmically and depends on the order in which unmapped users or groups are seen by Winbind. It may be difficult to recover the mappings of RID to UNIX ID if the file containing this information is corrupted or destroyed.
- Currently the Winbind PAM module does not take into account possible workstation and logon time restrictions that may be set for Windows NT users; this is instead up to the PDC to enforce.

24.7.1 NSCD Problem Warning

WARNING

Do not under any circumstances run **nscd** on any system on which **winbindd** is running.

If **nscd** is running on the UNIX/Linux system, then even though NSSWITCH is correctly configured, it will not be possible to resolve domain users and groups for file and directory controls.

24.7.2 Winbind Is Not Resolving Users and Groups

"My `smb.conf` file is correctly configured. I have specified *idmap uid* = 12000, and *idmap gid* = 3000-3500 and **winbind** is running. When I do the

following, it all works fine."

```
root# wbinfo -u
MIDEARTH\maryo
MIDEARTH\jackb
MIDEARTH\ameds
...
MIDEARTH\root

root# wbinfo -g
MIDEARTH\Domain Users
MIDEARTH\Domain Admins
MIDEARTH\Domain Guests
...
MIDEARTH\Accounts

root# getent passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
...
maryo:x:15000:15003:Mary Orville:/home/MIDEARTH/maryo:/bin/false
```

"But the following command just fails:"

```
root# chown maryo a_file
chown: 'maryo': invalid user
```

"This is driving me nuts! What can be wrong?"

Same problem as the one above. Your system is likely running **nscd**, the name service caching daemon. Shut it down, do not restart it! You will find your problem resolved. Alternately, fix the operation of nscd to resolve the problem.

Chapter 25

ADVANCED NETWORK MANAGEMENT

This section documents peripheral issues that are of great importance to network administrators who want to improve network resource access control, to automate the user environment, and to make their lives a little easier.

25.1 Features and Benefits

Often the difference between a working network environment and a well-appreciated one can best be measured by the *little things* that make everything work more harmoniously. A key part of every network environment solution is the ability to remotely manage MS Windows workstations, remotely access the Samba server, provide customized logon scripts, as well as other housekeeping activities that help to sustain more reliable network operations.

This chapter presents information on each of these areas. They are placed here, and not in other chapters, for ease of reference.

25.2 Remote Server Administration

"How do I get User Manager and Server Manager?"

Since I do not need to buy an NT4 server, how do I get the User Manager for Domains and the Server Manager?

Microsoft distributes a version of these tools called `Nexus.exe` for installation on Windows 9x/Me systems. The tools set includes:

- Server Manager
- User Manager for Domains
- Event Viewer

Download the archived file at the Microsoft Nexus[1] link.

The Windows NT 4.0 version of the User Manager for Domains and Server Manager are available from Microsoft via ftp[2].

25.3 Remote Desktop Management

There are a number of possible remote desktop management solutions that range from free through costly. Do not let that put you off. Sometimes the most costly solution is the most cost effective. In any case, you will need to draw your own conclusions as to which is the best tool in your network environment.

25.3.1 Remote Management from NoMachine.Com

The following information was posted to the Samba mailing list at Apr 3 23:33:50 GMT 2003. It is presented in slightly edited form (with author details omitted for privacy reasons). The entire answer is reproduced below with some comments removed.

" I have a wonderful Linux/Samba server running as PDC for a network. Now I would like to add remote desktop capabilities so users outside could login to the system and get their desktop up from home or another country."

" Is there a way to accomplish this? Do I need a Windows Terminal server? Do I need to configure it so it is a member of the domain or a BDC or PDC? Are there any hacks for MS Windows XP to enable remote login even if the computer is in a domain?"

[1] <`ftp://ftp.microsoft.com/Softlib/MSLFILES/NEXUS.EXE`>
[2] <`ftp://ftp.microsoft.com/Softlib/MSLFILES/SRVTOOLS.EXE`>

Answer provided: Check out the new offer of "NX" software from NoMachine[3].

It implements an easy-to-use interface to the Remote X protocol as well as incorporating VNC/RFB and rdesktop/RDP into it, but at a speed performance much better than anything you may have ever seen.

Remote X is not new at all, but what they did achieve successfully is a new way of compression and caching technologies that makes the thing fast enough to run even over slow modem/ISDN connections.

I test drove their (public) Red Hat machine in Italy, over a loaded Internet connection, with enabled thumbnail previews in KDE konqueror, which popped up immediately on "mouse-over". From inside that (remote X) session I started a rdesktop session on another, a Windows XP machine. To test the performance, I played Pinball. I am proud to announce that my score was 631,750 points at first try.

NX performs better on my local LAN than any of the other "pure" connection methods I use from time to time: TightVNC, rdesktop or Remote X. It is even faster than a direct crosslink connection between two nodes.

I even got sound playing from the Remote X app to my local boxes, and had a working "copy'n'paste" from an NX window (running a KDE session in Italy) to my Mozilla mailing agent. These guys are certainly doing something right!

I recommend test driving NX to anybody with a only a passing interest in remote computing the NX[4] utility.

Just download the free-of-charge client software (available for Red Hat, SuSE, Debian and Windows) and be up and running within 5 minutes (they need to send you your account data, though, because you are assigned a real UNIX account on their testdrive.nomachine.com box).

They plan to get to the point were you can have NX application servers running as a cluster of nodes, and users simply start an NX session locally and can select applications to run transparently (apps may even run on another NX node, but pretend to be on the same as used for initial login, because it displays in the same window. You also can run it full-screen, and after a short time you forget that it is a remote session at all).

[3] <http://www.nomachine.com/>
[4] <http://www.nomachine.com/testdrive.php>

Now the best thing for last: All the core compression and caching technologies are released under the GPL and available as source code to anybody who wants to build on it! These technologies are working, albeit started from the command line only (and very inconvenient to use in order to get a fully running remote X session up and running).

To answer your questions:

- You do not need to install a terminal server; XP has RDP support built in.
- NX is much cheaper than Citrix — and comparable in performance, probably faster.
- You do not need to hack XP — it just works.
- You log into the XP box from remote transparently (and I think there is no need to change anything to get a connection, even if authentication is against a domain).
- The NX core technologies are all Open Source and released under the GPL — you can now use a (very inconvenient) command line at no cost, but you can buy a comfortable (proprietary) NX GUI front end for money.
- NoMachine is encouraging and offering help to OSS/Free Software implementations for such a front-end too, even if it means competition to them (they have written to this effect even to the LTSP, KDE, and GNOME developer mailing lists).

25.3.2 Remote Management with ThinLinc

Another alternative for remote access is *ThinLinc* from Cendio.

ThinLinc is a terminal server solution that is available for Linux and Solaris based on standard protocols such as SSH, TightVNC, NFS and PulseAudio.

ThinLinc an be used both in the LAN environment to implement a Thin Client strategy for an organization, and as secure remote access solution for people working from remote locations, even over smallband connections. ThinLinc is free to use for a single concurrent user.

The product can also be used as a frontend to access Windows Terminal Server or Citrix farms, or even Windows XP machines, securing the connec-

tion via the ssh protocol. The client is available both for Linux (supporting all Linux distributions as well as numerous thin terminals) and for Windows. A Java-based Web client is also available.

ThinLinc may be evaluated by connecting to Cendio's demo system, see Cendio's[5] web site testdrive[6] center.

Cendio is a major contributor to several open source projects including TightVNC[7], PulseAudio[8] , unfsd, Python[9] and rdesktop[10].

25.4 Network Logon Script Magic

There are several opportunities for creating a custom network startup configuration environment.

- No Logon Script.
- Simple universal Logon Script that applies to all users.
- Use of a conditional Logon Script that applies per-user or per-group attributes.
- Use of Samba's preexec and postexec functions on access to the NETLOGON share to create a custom logon script and then execute it.
- User of a tool such as KixStart.

The Samba source code tree includes two logon script generation/execution tools. See `examples` directory `genlogon` and `ntlogon` subdirectories.

The following listings are from the genlogon directory.

This is the `genlogon.pl` file:

```
#!/usr/bin/perl
#
# genlogon.pl
#
```

[5] <http://www.cendio.com>
[6] <http://www.cendio.com/testdrive>
[7] <http://www.tightvnc.com>
[8] <http://pulseaudio.org>
[9] <http://www.python.org>
[10] <http://www.rdesktop.org>

```
# Perl script to generate user logon scripts on the fly, when users
# connect from a Windows client. This script should be called from
# smb.conf with the %U, %G and %L parameters. I.e:
#
#       root preexec = genlogon.pl %U %G %L
#
# The script generated will perform
# the following:
#
# 1. Log the user connection to /var/log/samba/netlogon.log
# 2. Set the PC's time to the Linux server time (which is maintained
#    daily to the National Institute of Standards Atomic clock on the
#    internet.
# 3. Connect the user's home drive to H: (H for Home).
# 4. Connect common drives that everyone uses.
# 5. Connect group-specific drives for certain user groups.
# 6. Connect user-specific drives for certain users.
# 7. Connect network printers.

# Log client connection
#($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time);
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time);
open LOG, ">>/var/log/samba/netlogon.log";
print LOG "$mon/$mday/$year $hour:$min:$sec";
print LOG " - User $ARGV[0] logged into $ARGV[1]\n";
close LOG;

# Start generating logon script
open LOGON, ">/shared/netlogon/$ARGV[0].bat";
print LOGON "\@ECHO OFF\r\n";

# Connect shares just use by Software Development group
if ($ARGV[1] eq "SOFTDEV" || $ARGV[0] eq "softdev")
{
   print LOGON "NET USE M: \\\\$ARGV[2]\\SOURCE\r\n";
}

# Connect shares just use by Technical Support staff
if ($ARGV[1] eq "SUPPORT" || $ARGV[0] eq "support")
{
```

```
   print LOGON "NET USE S: \\\\$ARGV[2]\\SUPPORT\r\n";
}

# Connect shares just used by Administration staff
If ($ARGV[1] eq "ADMIN" || $ARGV[0] eq "admin")
{
   print LOGON "NET USE L: \\\\$ARGV[2]\\ADMIN\r\n";
   print LOGON "NET USE K: \\\\$ARGV[2]\\MKTING\r\n";
}

# Now connect Printers. We handle just two or three users a little
# differently, because they are the exceptions that have desktop
# printers on LPT1: - all other user's go to the LaserJet on the
# server.
if ($ARGV[0] eq 'jim'
    || $ARGV[0] eq 'yvonne')
{
   print LOGON "NET USE LPT2: \\\\$ARGV[2]\\LJET3\r\n";
   print LOGON "NET USE LPT3: \\\\$ARGV[2]\\FAXQ\r\n";
}
else
{
   print LOGON "NET USE LPT1: \\\\$ARGV[2]\\LJET3\r\n";
   print LOGON "NET USE LPT3: \\\\$ARGV[2]\\FAXQ\r\n";
}

# All done! Close the output file.
close LOGON;
```

Those wishing to use a more elaborate or capable logon processing system should check out these sites:

- <http://www.craigelachie.org/rhacer/ntlogon>
- <http://www.kixtart.org>

25.4.1 Adding Printers without User Intervention

Printers may be added automatically during logon script processing through the use of:

```
C:\> rundll32 printui.dll,PrintUIEntry /?
```

See the documentation in the Microsoft Knowledge Base article 189105[11].

25.4.2 Limiting Logon Connections

Sometimes it is necessary to limit the number of concurrent connections to a Samba shared resource. For example, a site may wish to permit only one network logon per user.

The Samba *preexec script* parameter can be used to permit only one connection per user. Though this method is not foolproof and may have side effects, the following contributed method may inspire someone to provide a better solution.

This is not a perfect solution because Windows clients can drop idle connections with an auto-reconnect capability that could result in the appearance that a share is no longer in use, while actually it is. Even so, it demonstrates the principle of use of the *preexec script* parameter.

The following share configuration demonstrates use of the script shown in Example 25.4.1.

```
[myshare]
   ...
   preexec script = /sbin/PermitSingleLogon.sh
   preexec close = Yes
   ...
```

[11] <http://support.microsoft.com/default.asp?scid=kb;en-us;189105>

Example 25.4.1 Script to Enforce Single Resource Logon

```
#!/bin/bash

IFS="-"
RESULT=$(smbstatus -S -u $1 2> /dev/null | awk 'NF \
        > 6 {print $1}' | sort | uniq -d)

if [ "X${RESULT}" == X  ]; then
  exit 0
else
  exit 1
fi
```

Chapter 26

SYSTEM AND ACCOUNT POLICIES

This chapter summarizes the current state of knowledge derived from personal practice and knowledge from Samba mailing list subscribers. Before reproduction of posted information, every effort has been made to validate the information given. Where additional information was uncovered through this validation, it is provided also.

26.1 Features and Benefits

When MS Windows NT 3.5 was introduced, the hot new topic was the ability to implement Group Policies for users and groups. Then along came MS Windows NT4 and a few sites started to adopt this capability. How do we know that? By the number of "boo-boos" (or mistakes) administrators made and then requested help to resolve.

By the time that MS Windows 2000 and Active Directory was released, administrators got the message: Group Policies are a good thing! They can help reduce administrative costs and actually make happier users. But adoption of the true potential of MS Windows 200x Active Directory and Group Policy Objects (GPOs) for users and machines were picked up on rather slowly. This was obvious from the Samba mailing list back in 2000 and 2001 when there were few postings regarding GPOs and how to replicate them in a Samba environment.

Judging by the traffic volume since mid 2002, GPOs have become a standard part of the deployment in many sites. This chapter reviews techniques and

methods that can be used to exploit opportunities for automation of control over user desktops and network client workstations.

26.2 Creating and Managing System Policies

Under MS Windows platforms, particularly those following the release of MS Windows NT4 and MS Windows 95, it is possible to create a type of file that would be placed in the NETLOGON share of a domain controller. As the client logs onto the network, this file is read and the contents initiate changes to the registry of the client machine. This file allows changes to be made to those parts of the registry that affect users, groups of users, or machines.

For MS Windows 9x/Me, this file must be called `Config.POL` and may be generated using a tool called `poledit.exe`, better known as the Policy Editor. The policy editor was provided on the Windows 98 installation CD-ROM, but disappeared again with the introduction of MS Windows Me. From comments of MS Windows network administrators, it would appear that this tool became a part of the MS Windows Me Resource Kit.

MS Windows NT4 server products include the *System Policy Editor* under **Start** -> **Programs** -> **Administrative Tools**. For MS Windows NT4 and later clients, this file must be called `NTConfig.POL`.

New with the introduction of MS Windows 2000 was the Microsoft Management Console or MMC. This tool is the new wave in the ever-changing landscape of Microsoft methods for management of network access and security. Every new Microsoft product or technology seems to make the old rules obsolete and introduces newer and more complex tools and methods. To Microsoft's credit, the MMC does appear to be a step forward, but improved functionality comes at a great price.

Before embarking on the configuration of network and system policies, it is highly advisable to read the documentation available from Microsoft's Web site regarding Implementing Profiles and Policies in Windows NT 4.0[1]. There are a large number of documents in addition to this old one that should also be read and understood. Try searching on the Microsoft Web site for "Group Policies".

[1] <http://www.microsoft.com/ntserver/techresources/management/prof_policies.asp>

What follows is a brief discussion with some helpful notes. The information provided here is incomplete — you are warned.

26.2.1 Windows 9x/ME Policies

You need the Windows 98 Group Policy Editor to set up Group Profiles under Windows 9x/Me. It can be found on the original full-product Windows 98 installation CD-ROM under `tools\reskit\netadmin\poledit`. Install this using the Add/Remove Programs facility, and then click on **Have Disk**.

Use the Group Policy Editor to create a policy file that specifies the location of user profiles and/or `My Documents`, and so on. Then save these settings in a file called `Config.POL` that needs to be placed in the root of the *[NETLOGON]* share. If Windows 98 is configured to log onto the Samba domain, it will automatically read this file and update the Windows 9x/Me registry of the machine as it logs on.

Further details are covered in the Windows 98 Resource Kit documentation.

If you do not take the correct steps, then every so often Windows 9x/Me will check the integrity of the registry and restore its settings from the backup copy of the registry it stores on each Windows 9x/Me machine. So, you will occasionally notice things changing back to the original settings.

Install the Group Policy handler for Windows 9x/Me to pick up Group Policies. Look on the Windows 98 CD-ROM in `\tools\reskit\netadmin\poledit`. Install Group Policies on a Windows 9x/Me client by double-clicking on `grouppol.inf`. Log off and on again a couple of times and see if Windows 98 picks up Group Policies. Unfortunately, this needs to be done on every Windows 9x/Me machine that uses Group Policies.

26.2.2 Windows NT4-Style Policy Files

To create or edit `ntconfig.pol`, you must use the NT Server Policy Editor, **poledit.exe**, which is included with NT4 Server but not with NT workstation. There is a Policy Editor on an NT4 Workstation but it is not suitable for creating domain policies. Furthermore, although the Windows 95 Policy Editor can be installed on an NT4 workstation/server, it will not work with NT clients. However, the files from the NT Server will run happily enough on an NT4 workstation.

You need `poledit.exe`, `common.adm`, and `winnt.adm`. It is convenient to put the two `*.adm` files in the `c:\winnt\inf` directory, which is where the binary will look for them unless told otherwise. This directory is normally "hidden."

The Windows NT Policy Editor is also included with the Service Pack 3 (and later) for Windows NT 4.0. Extract the files using **servicepackname /x** — that's **Nt4sp6ai.exe /x** for Service Pack 6a. The Policy Editor, **poledit.exe**, and the associated template files (*.adm) should be extracted as well. It is also possible to download the policy template files for Office97 and get a copy of the Policy Editor. Another possible location is with the Zero Administration Kit available for download from Microsoft.

26.2.2.1 Registry Spoiling

With NT4-style registry-based policy changes, a large number of settings are not automatically reversed as the user logs off. The settings that were in the `NTConfig.POL` file were applied to the client machine registry and apply to the hive key HKEY_LOCAL_MACHINE are permanent until explicitly reversed. This is known as tattooing. It can have serious consequences downstream, and the administrator must be extremely careful not to lock out the ability to manage the machine at a later date.

26.2.3 MS Windows 200x/XP Professional Policies

Windows NT4 system policies allow the setting of registry parameters specific to users, groups, and computers (client workstations) that are members of the NT4-style domain. Such policy files will work with MS Windows 200x/XP clients also.

New to MS Windows 2000, Microsoft recently introduced a style of Group Policy that confers a superset of capabilities compared with NT4-style policies. Obviously, the tool used to create them is different, and the mechanism for implementing them is much improved.

The older NT4-style registry-based policies are known as *Administrative Templates* in MS Windows 2000/XP GPOs. The latter includes the ability to set various security configurations, enforce Internet Explorer browser settings, change and redirect aspects of the users desktop (including the location of `My Documents` files, as well as intrinsics of where menu items

will appear in the Start menu). An additional new feature is the ability to make available particular software Windows applications to particular users and/or groups.

Remember, NT4 policy files are named `NTConfig.POL` and are stored in the root of the NETLOGON share on the domain controllers. A Windows NT4 user enters a username and password and selects the domain name to which the logon will attempt to take place. During the logon process, the client machine reads the `NTConfig.POL` file from the NETLOGON share on the authenticating server and modifies the local registry values according to the settings in this file.

Windows 200x GPOs are feature-rich. They are not stored in the NETLOGON share, but rather part of a Windows 200x policy file is stored in the Active Directory itself and the other part is stored in a shared (and replicated) volume called the SYSVOL folder. This folder is present on all Active Directory domain controllers. The part that is stored in the Active Directory itself is called the Group Policy Container (GPC), and the part that is stored in the replicated share called SYSVOL is known as the Group Policy Template (GPT).

With NT4 clients, the policy file is read and executed only as each user logs onto the network. MS Windows 200x policies are much more complex — GPOs are processed and applied at client machine startup (machine specific part), and when the user logs onto the network, the user-specific part is applied. In MS Windows 200x-style policy management, each machine and/or user may be subject to any number of concurrently applicable (and applied) policy sets (GPOs). Active Directory allows the administrator to also set filters over the policy settings. No such equivalent capability exists with NT4-style policy files.

26.2.3.1 Administration of Windows 200x/XP Policies

Instead of using the tool called the System Policy Editor, commonly called Poledit (from the executable name **poledit.exe**), GPOs are created and managed using a Microsoft Management Console (MMC) snap-in as follows:

1. Go to the Windows 200x/XP menu **Start->Programs->Administrative Tools** and select the MMC snap-in called **Active Directory Users and Computers**

2. Select the domain or organizational unit (OU) that you wish to manage, then right-click to open the context menu for that object, and select the **Properties**.
3. Left-click on the **Group Policy** tab, then left-click on the New tab. Type a name for the new policy you will create.
4. Left-click on the **Edit** tab to commence the steps needed to create the GPO.

All policy configuration options are controlled through the use of policy administrative templates. These files have an .adm extension, both in NT4 as well as in Windows 200x/XP. Beware, however, the .adm files are not interchangeable across NT4 and Windows 200x. The latter introduces many new features as well as extended definition capabilities. It is well beyond the scope of this documentation to explain how to program .adm files; for that, refer to the Microsoft Windows Resource Kit for your particular version of MS Windows.

NOTE

The MS Windows 2000 Resource Kit contains a tool called **gpolmig.exe**. This tool can be used to migrate an NT4 `NTConfig.POL` file into a Windows 200x style GPO. Be VERY careful how you use this powerful tool. Please refer to the resource kit manuals for specific usage information.

26.2.3.2 Custom System Policy Templates

Over the past year, there has been a bit of talk regarding the creation of customized templates for the Windows Sytem Policy Editor. A recent announcement on the Samba mailing list is worthy of mention.

Mike Petersen has announced the availability of a template file he has created. This custom System Policy Editor Template will allow you to successfully control Microsoft Windows workstations from an SMB server, such as Samba. This template has been tested on a few networks, although if you

find any problems with any of these policies, or have any ideas for additional policies, let me know at mailto:mgpeter@pcc-services.com. This Template includes many policies for Windows XP to allow it to behave better in a professional environment.

For further information please see the Petersen[2] Computer Consulting web site. There is a download link for the template file.

26.3 Managing Account/User Policies

Policies can define a specific user's settings or the settings for a group of users. The resulting policy file contains the registry settings for all users, groups, and computers that will be using the policy file. Separate policy files for each user, group, or computer are not necessary.

If you create a policy that will be automatically downloaded from validating domain controllers, you should name the file `NTConfig.POL`. As system administrator, you have the option of renaming the policy file and, by modifying the Windows NT-based workstation, directing the computer to update the policy from a manual path. You can do this by either manually changing the registry or by using the System Policy Editor. This can even be a local path such that each machine has its own policy file, but if a change is necessary to all machines, it must be made individually to each workstation.

When a Windows NT4/200x/XP machine logs onto the network, the client looks in the NETLOGON share on the authenticating domain controller for the presence of the `NTConfig.POL` file. If one exists, it is downloaded, parsed, and then applied to the user's part of the registry.

MS Windows 200x/XP clients that log onto an MS Windows Active Directory security domain may additionally acquire policy settings through GPOs that are defined and stored in Active Directory itself. The key benefit of using AD GPOs is that they impose no registry *spoiling* effect. This has considerable advantage compared with the use of `NTConfig.POL` (NT4) style policy updates.

In addition to user access controls that may be imposed or applied via system and/or group policies in a manner that works in conjunction with user profiles, the user management environment under MS Windows NT4/200x/XP

[2] <http://www.pcc-services.com/custom_poledit.html>

allows per-domain as well as per-user account restrictions to be applied. Common restrictions that are frequently used include:

- Logon hours
- Password aging
- Permitted logon from certain machines only
- Account type (local or global)
- User rights

Samba-3.0.20 does not yet implement all account controls that are common to MS Windows NT4/200x/XP. While it is possible to set many controls using the Domain User Manager for MS Windows NT4, only password expiry is functional today. Most of the remaining controls at this time have only stub routines that may eventually be completed to provide actual control. Do not be misled by the fact that a parameter can be set using the NT4 Domain User Manager or in the `NTConfig.POL`.

26.4 Management Tools

Anyone who wishes to create or manage Group Policies will need to be familiar with a number of tools. The following sections describe a few key tools that will help you to create a low-maintenance user environment.

26.4.1 Samba Editreg Toolset

A new tool called **editreg** is under development. This tool can be used to edit registry files (called `NTUser.DAT`) that are stored in user and group profiles. `NTConfig.POL` files have the same structure as the `NTUser.DAT` file and can be edited using this tool. **editreg** is being built with the intent to enable `NTConfig.POL` files to be saved in text format and to permit the building of new `NTConfig.POL` files with extended capabilities. It is proving difficult to realize this capability, so do not be surprised if this feature does not materialize. Formal capabilities will be announced at the time that this tool is released for production use.

26.4.2 Windows NT4/200x

The tools that may be used to configure these types of controls from the MS Windows environment are the NT4 User Manager for Domains, the NT4 System and Group Policy Editor, and the Registry Editor (regedt32.exe). Under MS Windows 200x/XP, this is done using the MMC with appropriate "snap-ins," the registry editor, and potentially also the NT4 System and Group Policy Editor.

26.4.3 Samba PDC

With a Samba domain controller, the new tools for managing user account and policy information include: **smbpasswd**, **pdbedit**, **net**, and **rpcclient**. The administrator should read the man pages for these tools and become familiar with their use.

26.5 System Startup and Logon Processing Overview

The following attempts to document the order of processing the system and user policies following a system reboot and as part of the user logon:

1. Network starts, then Remote Procedure Call System Service (RPCSS) and multiple universal naming convention provider (MUP) start.
2. Where Active Directory is involved, an ordered list of GPOs is downloaded and applied. The list may include GPOs that:
   - Apply to the location of machines in a directory.
   - Apply only when settings have changed.
   - Depend on configuration of the scope of applicability: local, site, domain, organizational unit, and so on.

   No desktop user interface is presented until the above have been processed.
3. Execution of startup scripts (hidden and synchronous by default).
4. A keyboard action to effect start of logon (Ctrl-Alt-Del).
5. User credentials are validated, user profile is loaded (depends on policy settings).

6. An ordered list of user GPOs is obtained. The list contents depends on what is configured in respect of:
   - Is the user a domain member, thus subject to particular policies?
   - Loopback enablement, and the state of the loopback policy (merge or replace).
   - Location of the Active Directory itself.
   - Has the list of GPOs changed? No processing is needed if not changed.
7. User policies are applied from Active Directory. Note: There are several types.
8. Logon scripts are run. New to Windows 200x and Active Directory, logon scripts may be obtained based on GPOs (hidden and executed synchronously). NT4-style logon scripts are then run in a normal window.
9. The user interface as determined from the GPOs is presented. Note: In a Samba domain (like an NT4 domain), machine (system) policies are applied at startup; user policies are applied at logon.

26.6 Common Errors

Policy-related problems can be quite difficult to diagnose and even more difficult to rectify. The following collection demonstrates only basic issues.

26.6.1 Policy Does Not Work

"We have created the `Config.POL` file and put it in the *NETLOGON* share. It has made no difference to our Win XP Pro machines, they just do not see it. It worked fine with Win 98 but does not work any longer since we upgraded to Win XP Pro. Any hints?"

Policy files are not portable between Windows 9x/Me and MS Windows NT4/200x/XP-based platforms. You need to use the NT4 Group Policy Editor to create a file called `NTConfig.POL` so it is in the correct format for your MS Windows XP Pro clients.

Chapter 27

DESKTOP PROFILE MANAGEMENT

27.1 Features and Benefits

Roaming profiles are feared by some, hated by a few, loved by many, and a godsend for some administrators.

Roaming profiles allow an administrator to make available a consistent user desktop as the user moves from one machine to another. This chapter provides much information regarding how to configure and manage roaming profiles.

While roaming profiles might sound like nirvana to some, they are a real and tangible problem to others. In particular, users of mobile computing tools, where often there may not be a sustained network connection, are often better served by purely local profiles. This chapter provides information to help the Samba administrator deal with those situations.

27.2 Roaming Profiles

WARNING

Roaming profiles support is different for Windows 9x/Me and Windows NT4/200x.

Before discussing how to configure roaming profiles, it is useful to see how Windows 9x/Me and Windows NT4/200x clients implement these features.

Windows 9x/Me clients send a NetUserGetInfo request to the server to get the user's profiles location. However, the response does not have room for a separate profiles location field, only the user's home share. This means that Windows 9x/Me profiles are restricted to being stored in the user's home directory.

Windows NT4/200x clients send a NetSAMLogon RPC request, which contains many fields including a separate field for the location of the user's profiles.

27.2.1 Samba Configuration for Profile Handling

This section documents how to configure Samba for MS Windows client profile support.

27.2.1.1 NT4/200x User Profiles

For example, to support Windows NT4/200x clients, set the following in the [global] section of the `smb.conf` file:

```
        logon path = \\profileserver\profileshare\ ↩
           profilepath\%U\moreprofilepath
```

This is typically implemented like:

```
        logon path = \\%L\Profiles\%U
```

where "%L" translates to the name of the Samba server and "%U" translates to the username.

The default for this option is `\\%N\%U\profile`, namely, `\\sambaserver\username\profile`. The `\\%N\%U` service is created automatically by the [homes] service. If you are using a Samba server for the profiles, you must make the share that is specified in the logon path browseable. Please refer to the man page for `smb.conf` regarding the different semantics of "%L" and "%N", as well as "%U" and "%u".

NOTE

MS Windows NT/200x clients at times do not disconnect a connection to a server between logons. It is recommended to not use the *homes* metaservice name as part of the profile share path.

27.2.1.2 Windows 9x/Me User Profiles

To support Windows 9x/Me clients, you must use the *logon home* parameter. Samba has been fixed so `net use /home` now works as well and it, too, relies on the *`logon home`* parameter.

By using the *`logon home`* parameter, you are restricted to putting Windows 9x/Me profiles in the user's home directory. But wait! There is a trick you can use. If you set the following in the *[global]* section of your `smb.conf` file:

```
        logon home = \\%L\%U\.profiles
```

then your Windows 9x/Me clients will dutifully put their clients in a subdirectory of your home directory called `.profiles` (making them hidden).

Not only that, but `net use /home` will also work because of a feature in Windows 9x/Me. It removes any directory stuff off the end of the home directory area and only uses the server and share portion. That is, it looks like you specified `\\%L\%U` for *logon home*.

27.2.1.3 Mixed Windows Windows 9x/Me and NT4/200x User Profiles

You can support profiles for Windows 9x and Windows NT clients by setting both the *logon home* and *logon path* parameters. For example,

```
        logon home = \\%L\%U\.profiles
        logon path = \\%L\profiles\%U
```

Windows 9x/Me and NT4 and later profiles should not be stored in the same location because Windows NT4 and later will experience problems with mixed profile environments.

27.2.1.4 Disabling Roaming Profile Support

The question often asked is, "How may I enforce use of local profiles?" or "How do I disable roaming profiles?"

There are three ways of doing this:

In `smb.conf` Affect the following settings and ALL clients will be forced to use a local profile: *logon home* = and *logon path* =

The arguments to these parameters must be left blank. It is necessary to include the = sign to specifically assign the empty value.

MS Windows Registry: Use the Microsoft Management Console (MMC) **gpedit.msc** to instruct your MS Windows XP machine to use only a local profile. This, of course, modifies registry settings. The full path to the option is:

```
Local Computer Policy\
   Computer Configuration\
      Administrative Templates\
         System\
            User Profiles\

Disable: Only Allow Local User Profiles
Disable: Prevent Roaming Profile Change from Propagating to the Server
```

Change of Profile Type: From the start menu right-click on the **My Computer** icon, select **Properties**, click on the **User Profiles** tab, select the profile you wish to change from **Roaming** type to **Local**, and click on **Change Type**.

Consult the MS Windows registry guide for your particular MS Windows version for more information about which registry keys to change to enforce use of only local user profiles.

NOTE

The specifics of how to convert a local profile to a roaming profile, or a roaming profile to a local one, vary according to the version of MS Windows you are running. Consult the Microsoft MS Windows Resource Kit for your version of Windows for specific information.

27.2.2 Windows Client Profile Configuration Information

27.2.2.1 Windows 9x/Me Profile Setup

When a user first logs in on Windows 9x, the file user.DAT is created, as are folders `Start Menu`, `Desktop`, `Programs`, and `Nethood`. These directories and their contents will be merged with the local versions stored in `c:\windows\profiles\username` on subsequent logins, taking the most recent from each. You will need to use the *[global]* options *preserve case* = yes, *short preserve case* = yes, and *case sensitive* = no in order to maintain capital letters in shortcuts in any of the profile folders.

The `user.DAT` file contains all the user's preferences. If you wish to enforce a set of preferences, rename their `user.DAT` file to `user.MAN`, and deny them write access to this file.

1. On the Windows 9x/Me machine, go to **Control Panel** -> **Passwords** and select the **User Profiles** tab. Select the required level of roaming preferences. Press **OK**, but do not allow the computer to reboot.

2. On the Windows 9x/Me machine, go to **Control Panel** -> **Network** -> **Client for Microsoft Networks** -> **Preferences**. Select **Log on to NT Domain**. Then, ensure that the Primary Logon is **Client for Microsoft Networks**. Press **OK**, and this time allow the computer to reboot.

Under Windows 9x/Me, profiles are downloaded from the Primary Logon. If you have the Primary Logon as "Client for Novell Networks", then the profiles and logon script will be downloaded from your Novell server. If you have the Primary Logon as "Windows Logon", then the profiles will be loaded from the local machine — a bit against the concept of roaming profiles, it would seem!

You will now find that the Microsoft Networks Login box contains `[user, password, domain]` instead of just `[user, password]`. Type in the Samba server's domain name (or any other domain known to exist, but bear in mind that the user will be authenticated against this domain and profiles downloaded from it if that domain logon server supports it), user name and user's password.

Once the user has been successfully validated, the Windows 9x/Me machine informs you that `The user has not logged on before` and asks `Do you wish to save the user's preferences?` Select **Yes**.

Once the Windows 9x/Me client comes up with the desktop, you should be able to examine the contents of the directory specified in the *logon path* on the Samba server and verify that the `Desktop`, `Start Menu`, `Programs`, and `Nethood` folders have been created.

These folders will be cached locally on the client and updated when the user logs off (if you haven't made them read-only by then). You will find that if the user creates further folders or shortcuts, the client will merge the profile contents downloaded with the contents of the profile directory already on the local client, taking the newest folders and shortcut from each set.

If you have made the folders/files read-only on the Samba server, then you will get errors from the Windows 9x/Me machine on logon and logout as it attempts to merge the local and remote profile. Basically, if you have any errors reported by the Windows 9x/Me machine, check the UNIX file permissions and ownership rights on the profile directory contents, on the Samba server.

If you have problems creating user profiles, you can reset the user's local desktop cache, as shown below. When this user next logs in, the user will be told that he/she is logging in "for the first time".

1. Instead of logging in under the [user, password, domain] dialog, press **escape**.
2. Run the **regedit.exe** program, and look in:

`HKEY_LOCAL_MACHINE\Windows\CurrentVersion\ProfileList`

You will find an entry for each user of ProfilePath. Note the contents of this key (likely to be `c:\windows\profiles\username`), then delete the key *`ProfilePath`* for the required user.

3. Exit the registry editor.
4. Search for the user's .PWL password-caching file in the `c:\windows` directory, and delete it.
5. Log off the Windows 9x/Me client.
6. Check the contents of the profile path (see *logon path* described above) and delete the `user.DAT` or `user.MAN` file for the user, making a backup if required.

WARNING

Before deleting the contents of the directory listed in the *`ProfilePath`* (this is likely to be `c:\windows\profiles\username`), ask whether the owner has any important files stored on his or her desktop or start menu. Delete the contents of the directory *`ProfilePath`* (making a backup if any of the files are needed).

This will have the effect of removing the local (read-only hidden system file) `user.DAT` in their profile directory, as well as the local "desktop," "nethood," "start menu," and "programs" folders.

If all else fails, increase Samba's debug log levels to between 3 and 10, and/or run a packet sniffer program such as ethereal or **netmon.exe**, and look for error messages.

If you have access to an Windows NT4/200x server, then first set up roaming profiles and/or netlogons on the Windows NT4/200x server. Make a packet trace, or examine the example packet traces provided with Windows

NT4/200x server, and see what the differences are with the equivalent Samba trace.

27.2.2.2 Windows NT4 Workstation

When a user first logs in to a Windows NT workstation, the profile NTuser.DAT is created. The profile location can be now specified through the *logon path* parameter.

There is a parameter that is now available for use with NT Profiles: *logon drive*. This should be set to `H:` or any other drive, and should be used in conjunction with the new *logon home* parameter.

The entry for the NT4 profile is a directory, not a file. The NT help on profiles mentions that a directory is also created with a .PDS extension. The user, while logging in, must have write permission to create the full profile path (and the folder with the .PDS extension for those situations where it might be created).

In the profile directory, Windows NT4 creates more folders than Windows 9x/Me. It creates `Application Data` and others, as well as `Desktop`, `Nethood`, `Start Menu,` and `Programs`. The profile itself is stored in a file `NTuser.DAT`. Nothing appears to be stored in the .PDS directory, and its purpose is currently unknown.

You can use the System Control Panel to copy a local profile onto a Samba server (see NT help on profiles; it is also capable of firing up the correct location in the System Control Panel for you). The NT help file also mentions that renaming `NTuser.DAT` to `NTuser.MAN` turns a profile into a mandatory one.

The case of the profile is significant. The file must be called `NTuser.DAT` or, for a mandatory profile, `NTuser.MAN`.

27.2.2.3 Windows 2000/XP Professional

You must first convert the profile from a local profile to a domain profile on the MS Windows workstation as follows:

1. Log on as the *local* workstation administrator.
2. Right-click on the **My Computer** icon, and select **Properties**.

3. Click on the **User Profiles** tab.
4. Select the profile you wish to convert (click it once).
5. Click on the **Copy To** button.
6. In the **Permitted to use** box, click on the **Change** button.
7. Click on the **Look in** area that lists the machine name. When you click here, it will open up a selection box. Click on the domain to which the profile must be accessible.

> NOTE
>
> You will need to log on if a logon box opens up. For example, connect as *DOMAIN*\root, password: *mypassword*.

8. To make the profile capable of being used by anyone, select "Everyone".
9. Click on **OK** and the Selection box will close.
10. Now click on **OK** to create the profile in the path you nominated.

Done. You now have a profile that can be edited using the Samba **profiles** tool.

> NOTE
>
> Under Windows NT/200x, the use of mandatory profiles forces the use of MS Exchange storage of mail data and keeps it out of the desktop profile. That keeps desktop profiles from becoming unusable.

Windows XP Service Pack 1 There is a security check new to Windows XP (or maybe only Windows XP service pack 1). It can be disabled via a group policy in the Active Directory. The policy is called:

```
Computer Configuration\Administrative Templates\System\User Profiles\
          Do not check for user ownership of Roaming Profile Folders
```

This should be set to `Enabled`.

Does the new version of Samba have an Active Directory analogue? If so, then you may be able to set the policy through this.

If you cannot set group policies in Samba, then you may be able to set the policy locally on each machine. If you want to try this, then do the following:

1. On the XP workstation, log in with an administrative account.
2. Click on **Start** -> **Run**.
3. Type **mmc**.
4. Click on **OK**.
5. A Microsoft Management Console should appear.
6. Click on **File** -> **Add/Remove Snap-in** -> **Add**.
7. Double-click on **Group Policy**.
8. Click on **Finish** -> **Close**.
9. Click on **OK**.
10. In the "Console Root" window expand **Local Computer Policy** -> **Computer Configuration** -> **Administrative Templates** -> **System** -> **User Profiles**.
11. Double-click on **Do not check for user ownership of Roaming Profile Folders**.
12. Select **Enabled**.
13. Click on **OK**.
14. Close the whole console. You do not need to save the settings (this refers to the console settings rather than the policies you have changed).
15. Reboot.

27.2.3 User Profile Hive Cleanup Service

There are certain situations that cause a cached local copy of roaming profile not to be deleted on exit, even if the policy to force such deletion is set. To deal with that situation, a special service was created. The application **UPHClean** (User Profile Hive Cleanup) can be installed as a service on Windows NT4/2000/XP Professional and Windows 2003.

The UPHClean software package can be downloaded from the User Profile Hive Cleanup Service[1] web site.

27.2.4 Sharing Profiles between Windows 9x/Me and NT4/200x/XP Workstations

Sharing of desktop profiles between Windows versions is not recommended. Desktop profiles are an evolving phenomenon, and profiles for later versions of MS Windows clients add features that may interfere with earlier versions of MS Windows clients. Probably the more salient reason to not mix profiles is that when logging off an earlier version of MS Windows, the older format of profile contents may overwrite information that belongs to the newer version, resulting in loss of profile information content when that user logs on again with the newer version of MS Windows.

If you then want to share the same Start Menu and Desktop with Windows 9x/Me, you must specify a common location for the profiles. The `smb.conf` parameters that need to be common are *logon path* and *logon home*.

If you have this set up correctly, you will find separate `user.DAT` and `NTuser.DAT` files in the same profile directory.

27.2.5 Profile Migration from Windows NT4/200x Server to Samba

There is nothing to stop you from specifying any path that you like for the location of users' profiles. Therefore, you could specify that the profile be stored on a Samba server or any other SMB server, as long as that SMB server supports encrypted passwords.

[1] http://www.microsoft.com/downloads/details.aspx?FamilyID=1B286E6D-8912-4E18-B570-42470E2F3582&displaylang=en

27.2.5.1 Windows NT4 Profile Management Tools

Unfortunately, the resource kit information is specific to the version of MS Windows NT4/200x. The correct resource kit is required for each platform.

Here is a quick guide: Profile Migration Procedure

1. On your NT4 domain controller, right-click on **My Computer**, then select **Properties**, then the tab labeled **User Profiles**.
2. Select a user profile you want to migrate and click on it.

> NOTE
>
> I am using the term "migrate" loosely. You can copy a profile to create a group profile. You can give the user *Everyone* rights to the profile you copy this to. That is what you need to do, since your Samba domain is not a member of a trust relationship with your NT4 PDC.

3. Click on the **Copy To** button.
4. In the box labeled **Copy Profile to** add your new path, such as, `c:\temp\foobar`
5. Click on **Change** in the **Permitted to use** box.
6. Click on the group "Everyone", click on **OK**. This closes the "choose user" box.
7. Now click on **OK**.

Follow these steps for every profile you need to migrate.

27.2.5.2 Side Bar Notes

You should obtain the SID of your NT4 domain. You can use the **net rpc info** to do this. See Chapter 13, "Remote and Local Management: The Net Command", Section 13.14 for more information.

27.2.5.3 moveuser.exe

The Windows 200x professional resource kit has **moveuser.exe**. **moveuser.exe** changes the security of a profile from one user to another. This allows the account domain to change and/or the username to change.

This command is like the Samba **profiles** tool.

27.2.5.4 Get SID

You can identify the SID by using **GetSID.exe** from the Windows NT Server 4.0 Resource Kit.

Windows NT 4.0 stores the local profile information in the registry under the following key: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList`

Under the ProfileList key, there will be subkeys named with the SIDs of the users who have logged on to this computer. (To find the profile information for the user whose locally cached profile you want to move, find the SID for the user with the **GetSID.exe** utility.) Inside the appropriate user's subkey, you will see a string value named *`ProfileImagePath`*.

27.3 Mandatory Profiles

A mandatory profile is a profile that the user does not have the ability to overwrite. During the user's session, it may be possible to change the desktop environment; however, as the user logs out, all changes made will be lost. If it is desired to not allow the user any ability to change the desktop environment, then this must be done through policy settings. See Chapter 26, "System and Account Policies".

NOTE

Under NO circumstances should the profile directory (or its contents) be made read-only because this may render the profile unusable. Where it is essential to make a profile read-only within the UNIX file system, this can be done, but then you absolutely must use the **fake-permissions** VFS module to instruct MS Windows NT/200x/XP clients that the Profile has write permission for the user. See Section 23.3.4.

For MS Windows NT4/200x/XP, the procedure shown in Section 27.2.5.1 can also be used to create mandatory profiles. To convert a group profile into a mandatory profile, simply locate the `NTUser.DAT` file in the copied profile and rename it to `NTUser.MAN`.

For MS Windows 9x/Me, it is the `User.DAT` file that must be renamed to `User.MAN` to effect a mandatory profile.

27.4 Creating and Managing Group Profiles

Most organizations are arranged into departments. There is a nice benefit in this fact, since usually most users in a department require the same desktop applications and the same desktop layout. MS Windows NT4/200x/XP will allow the use of group profiles. A group profile is a profile that is created first using a template (example) user. Then using the profile migration tool (see above), the profile is assigned access rights for the user group that needs to be given access to the group profile.

The next step is rather important. Instead of assigning a group profile to users (Using User Manager) on a "per-user" basis, the group itself is assigned the now modified profile.

NOTE

Be careful with group profiles. If the user who is a member of a group also has a personal profile, then the result will be a fusion (merge) of the two.

27.5 Default Profile for Windows Users

MS Windows 9x/Me and NT4/200x/XP will use a default profile for any user for whom a profile does not already exist. Armed with a knowledge of where the default profile is located on the Windows workstation, and knowing which registry keys affect the path from which the default profile is created, it is possible to modify the default profile to one that has been optimized for the site. This has significant administrative advantages.

27.5.1 MS Windows 9x/Me

To enable default per-use profiles in Windows 9x/Me, you can either use the Windows 98 System Policy Editor or change the registry directly.

To enable default per-user profiles in Windows 9x/Me, launch the System Policy Editor, then select **File** -> **Open Registry**. Next click on the **Local Computer** icon, click on **Windows 98 System**, select **User Profiles**, and click on the enable box. Remember to save the registry changes.

To modify the registry directly, launch the Registry Editor (**regedit.exe**) and select the hive `HKEY_LOCAL_MACHINE\Network\Logon`. Now add a DWORD type key with the name "User Profiles." To enable user profiles to set the value to 1; to disable user profiles set it to 0.

27.5.1.1 User Profile Handling with Windows 9x/Me

When a user logs on to a Windows 9x/Me machine, the local profile path, `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ProfileList`, is checked for an existing entry for that user.

If the user has an entry in this registry location, Windows 9x/Me checks for a locally cached version of the user profile. Windows 9x/Me also checks

the user's home directory (or other specified directory if the location has been modified) on the server for the user profile. If a profile exists in both locations, the newer of the two is used. If the user profile exists on the server but does not exist on the local machine, the profile on the server is downloaded and used. If the user profile only exists on the local machine, that copy is used.

If a user profile is not found in either location, the default user profile from the Windows 9x/Me machine is used and copied to a newly created folder for the logged on user. At log off, any changes that the user made are written to the user's local profile. If the user has a roaming profile, the changes are written to the user's profile on the server.

27.5.2 MS Windows NT4 Workstation

On MS Windows NT4, the default user profile is obtained from the location `%SystemRoot%\Profiles`, which in a default installation will translate to `C:\Windows NT\Profiles`. Under this directory on a clean install, there will be three directories: `Administrator`, `All Users,` and `Default User`.

The `All Users` directory contains menu settings that are common across all system users. The `Default User` directory contains menu entries that are customizable per user depending on the profile settings chosen/created.

When a new user first logs onto an MS Windows NT4 machine, a new profile is created from:

- All Users settings.
- Default User settings (contains the default `NTUser.DAT` file).

When a user logs on to an MS Windows NT4 machine that is a member of a Microsoft security domain, the following steps are followed for profile handling:

1. The user's account information that is obtained during the logon process contains the location of the user's desktop profile. The profile path may be local to the machine or it may be located on a network share. If there exists a profile at the location of the path from the user account, then this profile is copied to the location `%SystemRoot%\Profiles\%USERNAME%`. This profile then inherits the settings in the `All Users` profile in the `%SystemRoot%\Profiles` location.

2. If the user account has a profile path, but at its location a profile does not exist, then a new profile is created in the `%SystemRoot%\Profiles\%USERNAME%` directory from reading the `Default User` profile.
3. If the NETLOGON share on the authenticating server (logon server) contains a policy file (`NTConfig.POL`), then its contents are applied to the `NTUser.DAT`, which is applied to the `HKEY_CURRENT_USER` part of the registry.
4. When the user logs out, if the profile is set to be a roaming profile, it will be written out to the location of the profile. The `NTuser.DAT` file is then re-created from the contents of the `HKEY_CURRENT_USER` contents. Thus, should there not exist in the NETLOGON share an `NTConfig.POL` at the next logon, the effect of the previous `NTConfig.POL` will still be held in the profile. The effect of this is known as tattooing.

MS Windows NT4 profiles may be *local* or *roaming*. A local profile is stored in the `%SystemRoot%\Profiles\%USERNAME%` location. A roaming profile will also remain stored in the same way, unless the following registry key is created:

```
HKEY_LOCAL_MACHINE\SYSTEM\Software\Microsoft\Windows NT\CurrentVersion\
winlogon\"DeleteRoamingCache"=dword:0000000
```

In this case, the local copy (in `%SystemRoot%\Profiles\%USERNAME%`) will be deleted on logout.

Under MS Windows NT4, default locations for common resources like `My Documents` may be redirected to a network share by modifying the following registry keys. These changes may be made via use of the System Policy Editor. To do so may require that you create your own template extension for the Policy Editor to allow this to be done through the GUI. Another way to do this is by first creating a default user profile, then while logged in as that user, running **regedt32** to edit the key settings.

The Registry Hive key that affects the behavior of folders that are part of the default user profile are controlled by entries on Windows NT4 is:

```
HKEY_CURRENT_USER
   \Software
```

```
\Microsoft
   \Windows
      \CurrentVersion
         \Explorer
            \User Shell Folders
```

The above hive key contains a list of automatically managed folders. The default entries are shown in Table 27.1.

Table 27.1 User Shell Folder Registry Keys Default Values

Name	**Default Value**
AppData	%USERPROFILE%\Application Data
Desktop	%USERPROFILE%\Desktop
Favorites	%USERPROFILE%\Favorites
NetHood	%USERPROFILE%\NetHood
PrintHood	%USERPROFILE%\PrintHood
Programs	%USERPROFILE%\Start Menu\Programs
Recent	%USERPROFILE%\Recent
SendTo	%USERPROFILE%\SendTo
Start Menu	%USERPROFILE%\Start Menu
Startup	%USERPROFILE%\Start Menu\Programs\Startup

The registry key that contains the location of the default profile settings is:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\
User Shell Folders
```

The default entries are shown in Table 27.2.

Table 27.2 Defaults of Profile Settings Registry Keys

Common Desktop	%SystemRoot%\Profiles\All Users\Desktop
Common Programs	%SystemRoot%\Profiles\All Users\Programs
Common Start Menu	%SystemRoot%\Profiles\All Users\Start Menu
Common Startup	%SystemRoot%\Profiles\All Users\Start Menu\Programs\Startup

27.5.3 MS Windows 200x/XP

NOTE

MS Windows XP Home Edition does use default per-user profiles, but cannot participate in domain security, cannot log onto an NT/ADS-style domain, and thus can obtain the profile only from itself. While there are benefits in doing this, the beauty of those MS Windows clients that can participate in domain logon processes is that they allow the administrator to create a global default profile and enforce it through the use of Group Policy Objects (GPOs).

When a new user first logs onto an MS Windows 200x/XP machine, the default profile is obtained from `C:\Documents and Settings\Default User`. The administrator can modify or change the contents of this location, and MS Windows 200x/XP will gladly use it. This is far from the optimum arrangement, since it will involve copying a new default profile to every MS Windows 200x/XP client workstation.

When MS Windows 200x/XP participates in a domain security context, and if the default user profile is not found, then the client will search for a default profile in the NETLOGON share of the authenticating server. In MS Windows parlance, it is `%LOGONSERVER%\NETLOGON\Default User,` and if one exists there, it will copy this to the workstation in the `C:\Documents and Settings\` under the Windows login name of the use.

NOTE

This path translates, in Samba parlance, to the `smb.conf` *[NETLOGON]* share. The directory should be created at the root of this share and must be called `Default User`.

If a default profile does not exist in this location, then MS Windows 200x/XP will use the local default profile.

On logging out, the user's desktop profile is stored to the location specified in the registry settings that pertain to the user. If no specific policies have been created or passed to the client during the login process (as Samba does automatically), then the user's profile is written to the local machine only under the path `C:\Documents and Settings\%USERNAME%`.

Those wishing to modify the default behavior can do so through these three methods:

- Modify the registry keys on the local machine manually and place the new default profile in the NETLOGON share root. This is not recommended because it is maintenance intensive.
- Create an NT4-style NTConfig.POL file that specifies this behavior and locate this file in the root of the NETLOGON share along with the new default profile.
- Create a GPO that enforces this through Active Directory, and place the new default profile in the NETLOGON share.

The registry hive key that affects the behavior of folders that are part of the default user profile are controlled by entries on Windows 200x/XP is:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders\
```

This hive key contains a list of automatically managed folders. The default entries are shown in Table 27.3

There is also an entry called "Default" that has no value set. The default entry is of type `REG_SZ`; all the others are of type `REG_EXPAND_SZ`.

It makes a huge difference to the speed of handling roaming user profiles if all the folders are stored on a dedicated location on a network server. This means that it will not be necessary to write the Outlook PST file over the network for every login and logout.

To set this to a network location, you could use the following examples:

```
%LOGONSERVER%\%USERNAME%\Default Folders
```

Table 27.3 Defaults of Default User Profile Paths Registry Keys

Name	Default Value
AppData	%USERPROFILE%\Application Data
Cache	%USERPROFILE%\Local Settings\Temporary Internet Files
Cookies	%USERPROFILE%\Cookies
Desktop	%USERPROFILE%\Desktop
Favorites	%USERPROFILE%\Favorites
History	%USERPROFILE%\Local Settings\History
Local AppData	%USERPROFILE%\Local Settings\Application Data
Local Settings	%USERPROFILE%\Local Settings
My Pictures	%USERPROFILE%\My Documents\My Pictures
NetHood	%USERPROFILE%\NetHood
Personal	%USERPROFILE%\My Documents
PrintHood	%USERPROFILE%\PrintHood
Programs	%USERPROFILE%\Start Menu\Programs
Recent	%USERPROFILE%\Recent
SendTo	%USERPROFILE%\SendTo
Start Menu	%USERPROFILE%\Start Menu
Startup	%USERPROFILE%\Start Menu\Programs\Startup
Templates	%USERPROFILE%\Templates

This stores the folders in the user's home directory under a directory called `Default Folders`. You could also use:

```
\\SambaServer\FolderShare\%USERNAME%
```

in which case the default folders are stored in the server named *`SambaServer`* in the share called *`FolderShare`* under a directory that has the name of the MS Windows user as seen by the Linux/UNIX file system.

Please note that once you have created a default profile share, you *must* migrate a user's profile (default or custom) to it.

MS Windows 200x/XP profiles may be *local* or *roaming*. A roaming profile is cached locally unless the following registry key is created:

```
HKEY_LOCAL_MACHINE\SYSTEM\Software\Microsoft\Windows NT\CurrentVersion\
  winlogon\"DeleteRoamingCache"=dword:00000001
```

In this case, the local cache copy is deleted on logout.

27.6 Common Errors

The following are some typical errors, problems, and questions that have been asked on the Samba mailing lists.

27.6.1 Configuring Roaming Profiles for a Few Users or Groups

With Samba-2.2.x, the choice you have is to enable or disable roaming profiles support. It is a global-only setting. The default is to have roaming profiles, and the default path will locate them in the user's home directory.

If disabled globally, then no one will have roaming profile ability. If enabled and you want it to apply only to certain machines, then on those machines on which roaming profile support is not wanted, it is necessary to disable roaming profile handling in the registry of each such machine.

With Samba-3, you can have a global profile setting in `smb.conf`, and you can override this by per-user settings using the Domain User Manager (as with MS Windows NT4/200x).

In any case, you can configure only one profile per user. That profile can be either:

- A profile unique to that user.
- A mandatory profile (one the user cannot change).
- A group profile (really should be mandatory — that is, unchangable).

27.6.2 Cannot Use Roaming Profiles

A user requested the following: "I do not want roaming profiles to be implemented. I want to give users a local profile alone. I am totally lost with this error. For the past two days I tried everything, I googled around but found no useful pointers. Please help me."

The choices are:

Local profiles I know of no registry keys that will allow autodeletion of LOCAL profiles on log out.

Roaming profiles As a user logs onto the network, a centrally stored profile is copied to the workstation to form a local profile. This local profile will persist (remain on the workstation disk) unless a registry key is changed that will cause this profile to be automatically deleted on logout.

The roaming profile choices are:

Personal roaming profiles These are typically stored in a profile share on a central (or conveniently located local) server.

Workstations cache (store) a local copy of the profile. This cached copy is used when the profile cannot be downloaded at next logon.

Group profiles These are loaded from a central profile server.

Mandatory profiles Mandatory profiles can be created for a user as well as for any group that a user is a member of. Mandatory profiles cannot be changed by ordinary users. Only the administrator can change or reconfigure a mandatory profile.

A Windows NT4/200x/XP profile can vary in size from 130KB to very large. Outlook PST files are most often part of the profile and can be many gigabytes in size. On average (in a well controlled environment), roaming profile size of 2MB is a good rule of thumb to use for planning purposes. In an undisciplined environment, I have seen up to 2GB profiles. Users tend to complain when it takes an hour to log onto a workstation, but they harvest the fruits of folly (and ignorance).

The point of this discussion is to show that roaming profiles and good controls of how they can be changed as well as good discipline make for a problem-free site.

Microsoft's answer to the PST problem is to store all email in an MS Exchange Server backend. This removes the need for a PST file.

Local profiles mean:

- If each machine is used by many users, then much local disk storage is needed for local profiles.
- Every workstation the user logs into has its own profile; these can be very different from machine to machine.

On the other hand, use of roaming profiles means:

- The network administrator can control the desktop environment of all users.
- Use of mandatory profiles drastically reduces network management overheads.
- In the long run, users will experience fewer problems.

27.6.3 Changing the Default Profile

"When the client logs onto the domain controller, it searches for a profile to download. Where do I put this default profile?"

First, the Samba server needs to be configured as a domain controller. This can be done by setting in `smb.conf`:

```
security = user
os level = 32 (or more)
domain logons = Yes
```

There must be a *[netlogon]* share that is world readable. It is a good idea to add a logon script to preset printer and drive connections. There is also a facility for automatically synchronizing the workstation time clock with that of the logon server (another good thing to do).

NOTE

To invoke autodeletion of roaming profiles from the local workstation cache (disk storage), use the Group Policy Editor to create a file called `NTConfig.POL` with the appropriate entries. This file needs to be located in the *netlogon* share root directory.

Windows clients need to be members of the domain. Workgroup machines do not use network logons, so they do not interoperate with domain profiles.

For roaming profiles, add to `smb.conf`:

```
        logon path = \\%N\profiles\%U
# Default logon drive is Z:
        logon drive = H:
# This requires a PROFILES share that is world writable.
```

27.6.4 Debugging Roaming Profiles and NT4-style Domain Policies

Roaming profiles and domain policies are implemented via **USERENV.DLL**. Microsoft Knowledge Base articles 221833[2] and 154120[3] describe how to instruct that DLL to debug the login process.

[2] <http://support.microsoft.com/default.aspx?scid=kb;en-us;221833>
[3] <http://support.microsoft.com/default.aspx?scid=kb;en-us;154120>

Chapter 28

PAM-BASED DISTRIBUTED AUTHENTICATION

This chapter should help you to deploy Winbind-based authentication on any PAM-enabled UNIX/Linux system. Winbind can be used to enable user-level application access authentication from any MS Windows NT domain, MS Windows 200x Active Directory-based domain, or any Samba-based domain environment. It will also help you to configure PAM-based local host access controls that are appropriate to your Samba configuration.

In addition to knowing how to configure Winbind into PAM, you will learn generic PAM management possibilities and in particular how to deploy tools like `pam_smbpass.so` to your advantage.

> **Note**
>
> The use of Winbind requires more than PAM configuration alone. Please refer to Chapter 24, "Winbind: Use of Domain Accounts", for further information regarding Winbind.

28.1 Features and Benefits

A number of UNIX systems (e.g., Sun Solaris), as well as the xxxxBSD family and Linux, now utilize the Pluggable Authentication Modules (PAM)

facility to provide all authentication, authorization, and resource control services. Prior to the introduction of PAM, a decision to use an alternative to the system password database (`/etc/passwd`) would require the provision of alternatives for all programs that provide security services. Such a choice would involve provision of alternatives to programs such as **login**, **passwd**, **chown**, and so on.

PAM provides a mechanism that disconnects these security programs from the underlying authentication/authorization infrastructure. PAM is configured by making appropriate modifications to one file, `/etc/pam.conf` (Solaris), or by editing individual control files that are located in `/etc/pam.d`.

On PAM-enabled UNIX/Linux systems, it is an easy matter to configure the system to use any authentication backend so long as the appropriate dynamically loadable library modules are available for it. The backend may be local to the system or may be centralized on a remote server.

PAM support modules are available for:

`/etc/passwd` There are several PAM modules that interact with this standard UNIX user database. The most common are called `pam_unix.so`, `pam_unix2.so`, `pam_pwdb.so` and `pam_userdb.so`.

Kerberos The `pam_krb5.so` module allows the use of any Kerberos-compliant server. This tool is used to access MIT Kerberos, Heimdal Kerberos, and potentially Microsoft Active Directory (if enabled).

LDAP The `pam_ldap.so` module allows the use of any LDAP v2- or v3-compatible backend server. Commonly used LDAP backend servers include OpenLDAP v2.0 and v2.1, Sun ONE iDentity server, Novell eDirectory server, and Microsoft Active Directory.

NetWare Bindery The `pam_ncp_auth.so` module allows authentication off any bindery-enabled NetWare Core Protocol-based server.

SMB Password This module, called `pam_smbpass.so`, allows user authentication of the passdb backend that is configured in the Samba `smb.conf` file.

SMB Server The `pam_smb_auth.so` module is the original MS Windows networking authentication tool. This module has been somewhat outdated by the Winbind module.

Winbind The `pam_winbind.so` module allows Samba to obtain authentication from any MS Windows domain controller. It can just as easily be used to authenticate users for access to any PAM-enabled application.

RADIUS There is a PAM RADIUS (Remote Access Dial-In User Service) authentication module. In most cases, administrators need to locate the source code for this tool and compile and install it themselves. RADIUS protocols are used by many routers and terminal servers.

Of the modules listed, Samba provides the `pam_smbpasswd.so` and the `pam_winbind.so` modules alone.

Once configured, these permit a remarkable level of flexibility in the location and use of distributed Samba domain controllers that can provide wide-area network bandwidth, efficient authentication services for PAM-capable systems. In effect, this allows the deployment of centrally managed and maintained distributed authentication from a single-user account database.

28.2 Technical Discussion

PAM is designed to provide system administrators with a great deal of flexibility in configuration of the privilege-granting applications of their system. The local configuration of system security controlled by PAM is contained in one of two places: either the single system file `/etc/pam.conf` or the `/etc/pam.d/` directory.

28.2.1 PAM Configuration Syntax

In this section we discuss the correct syntax of and generic options respected by entries to these files. PAM-specific tokens in the configuration file are case insensitive. The module paths, however, are case sensitive, since they indicate a file's name and reflect the case dependence of typical file systems.

The case sensitivity of the arguments to any given module is defined for each module in turn.

In addition to the lines described below, there are two special characters provided for the convenience of the system administrator: comments are preceded by a "#" and extend to the next end-of-line; also, module specification lines may be extended with a "\"-escaped newline.

If the PAM authentication module (loadable link library file) is located in the default location, then it is not necessary to specify the path. In the case of Linux, the default location is `/lib/security`. If the module is located outside the default, then the path must be specified as:

```
auth  required  /other_path/pam_strange_module.so
```

28.2.1.1 Anatomy of /etc/pam.d Entries

The remaining information in this subsection was taken from the documentation of the Linux-PAM project. For more information on PAM, see the Official Linux-PAM home page[1].

A general configuration line of the `/etc/pam.conf` file has the following form:

```
service-name   module-type   control-flag   module-path   args
```

We explain the meaning of each of these tokens. The second (and more recently adopted) way of configuring Linux-PAM is via the contents of the `/etc/pam.d/` directory. Once we have explained the meaning of the tokens, we describe this method.

service-name The name of the service associated with this entry. Frequently, the service-name is the conventional name of the given application — for example, **ftpd**, **rlogind** and **su**, and so on.

There is a special service-name reserved for defining a default authentication mechanism. It has the name *OTHER* and may be specified in

[1] <http://ftp.kernel.org/pub/linux/libs/pam/>

either lower- or uppercase characters. Note, when there is a module specified for a named service, the *OTHER* entries are ignored.

module-type One of (currently) four types of module. The four types are as follows:

- *auth:* This module type provides two aspects of authenticating the user. It establishes that the user is who he or she claims to be by instructing the application to prompt the user for a password or other means of identification. Second, the module can grant group membership (independently of the `/etc/groups` file) or other privileges through its credential-granting properties.

- *account:* This module performs non-authentication-based account management. It is typically used to restrict/permit access to a service based on the time of day, currently available system resources (maximum number of users), or perhaps the location of the user login. For example, the "root" login may be permitted only on the console.

- *session:* Primarily, this module is associated with doing things that need to be done for the user before and after he or she can be given service. Such things include logging information concerning the opening and closing of some data exchange with a user, mounting directories, and so on.

- *password:* This last module type is required for updating the authentication token associated with the user. Typically, there is one module for each "challenge/response" authentication *(auth)* module type.

control-flag The control-flag is used to indicate how the PAM library will react to the success or failure of the module it is associated with. Since modules can be stacked (modules of the same type execute in series, one after another), the control-flags determine the relative importance of each module. The application is not made aware of the individual success or failure of modules listed in the `/etc/pam.conf` file. Instead, it receives a summary success or fail response from the Linux-PAM library. The order of execution of these modules is that of the entries in the `/etc/pam.conf` file; earlier entries are executed before later

ones. As of Linux-PAM v0.60, this control-flag can be defined with one of two syntaxes.

The simpler (and historical) syntax for the control-flag is a single keyword defined to indicate the severity of concern associated with the success or failure of a specific module. There are four such keywords: *required*, *requisite*, *sufficient*, and *optional*.

The Linux-PAM library interprets these keywords in the following manner:

- *required:* This indicates that the success of the module is required for the module-type facility to succeed. Failure of this module will not be apparent to the user until all of the remaining modules (of the same module-type) have been executed.
- *requisite:* Like required, except that if such a module returns a failure, control is directly returned to the application. The return value is that associated with the first required or requisite module to fail. This flag can be used to protect against the possibility of a user getting the opportunity to enter a password over an unsafe medium. It is conceivable that such behavior might inform an attacker of valid accounts on a system. This possibility should be weighed against the not insignificant concerns of exposing a sensitive password in a hostile environment.
- *sufficient:* The success of this module is deemed *sufficient* to satisfy the Linux-PAM library that this module-type has succeeded in its purpose. In the event that no previous required module has failed, no more "stacked" modules of this type are invoked. (In this case, subsequent required modules are not invoked). A failure of this module is not deemed as fatal to satisfying the application that this module-type has succeeded.
- *optional:* As its name suggests, this control-flag marks the module as not being critical to the success or failure of the user's application for service. In general, Linux-PAM ignores such a module when determining if the module stack will succeed or fail. However, in the absence of any definite successes or failures of previous or subsequent stacked modules, this module will determine the nature of the response to the application. One example of this latter case is when the other modules return something like PAM_IGNORE.

The more elaborate (newer) syntax is much more specific and gives the administrator a great deal of control over how the user is authenticated. This form of the control-flag is delimited with square brackets and consists of a series of *value=action* tokens:

```
[value1=action1 value2=action2 ...]
```

Here, *value1* is one of the following return values:

```
success; open_err; symbol_err; service_err; system_err; buf_err;
perm_denied; auth_err; cred_insufficient; authinfo_unavail;
user_unknown; maxtries; new_authtok_reqd; acct_expired; session_err;
cred_unavail; cred_expired; cred_err; no_module_data; conv_err;
authtok_err; authtok_recover_err; authtok_lock_busy;
authtok_disable_aging; try_again; ignore; abort; authtok_expired;
module_unknown; bad_item; and default.
```

The last of these (*default*) can be used to set the action for those return values that are not explicitly defined.

The *action1* can be a positive integer or one of the following tokens: *ignore*; *ok*; *done*; *bad*; *die*; and *reset*. A positive integer, J, when specified as the action, can be used to indicate that the next J modules of the current module-type will be skipped. In this way, the administrator can develop a moderately sophisticated stack of modules with a number of different paths of execution. Which path is taken can be determined by the reactions of individual modules.

- *ignore:* When used with a stack of modules, the module's return status will not contribute to the return code the application obtains.
- *bad:* This action indicates that the return code should be thought of as indicative of the module failing. If this module is the first in the stack to fail, its status value will be used for that of the whole stack.
- *die:* Equivalent to bad with the side effect of terminating the module stack and PAM immediately returning to the application.

- *ok:* This tells PAM that the administrator thinks this return code should contribute directly to the return code of the full stack of modules. In other words, if the former state of the stack would lead to a return of PAM_SUCCESS, the module's return code will override this value. Note, if the former state of the stack holds some value that is indicative of a module's failure, this *ok* value will not be used to override that value.

- *done:* Equivalent to *ok* with the side effect of terminating the module stack and PAM immediately returning to the application.

- *reset:* Clears all memory of the state of the module stack and starts again with the next stacked module.

Each of the four keywords, *required*; *requisite*; *sufficient*; and *optional*, have an equivalent expression in terms of the [...] syntax. They are as follows:

- *required* is equivalent to *[success=ok new_authtok_reqd=ok ignore=ignore default=bad]*.

- *requisite* is equivalent to *[success=ok new_authtok_reqd=ok ignore=ignore default=die]*.

- *sufficient* is equivalent to *[success=done new_authtok_reqd=done default=ignore]*.

- *optional* is equivalent to *[success=ok new_authtok_reqd=ok default=ignore]*.

Just to get a feel for the power of this new syntax, here is a taste of what you can do with it. With Linux-PAM-0.63, the notion of client plug-in agents was introduced. This makes it possible for PAM to support machine-machine authentication using the transport protocol inherent to the client/server application. With the *[... value=action ...]* control syntax, it is possible for an application to be configured to support binary prompts with compliant clients, but to gracefully fail over into an alternative authentication mode for legacy applications.

module-path The pathname of the dynamically loadable object file; the pluggable module itself. If the first character of the module path is "/", it is assumed to be a complete path. If this is not the case, the

given module path is appended to the default module path: `/lib/security` (but see the previous notes).

The arguments are a list of tokens that are passed to the module when it is invoked, much like arguments to a typical Linux shell command. Generally, valid arguments are optional and are specific to any given module. Invalid arguments are ignored by a module; however, when encountering an invalid argument, the module is required to write an error to syslog(3). For a list of generic options, see the next section.

If you wish to include spaces in an argument, you should surround that argument with square brackets. For example:

```
squid auth required pam_mysql.so user=passwd_query passwd=mada \
db=eminence [query=select user_name from internet_service where \
user_name=%u and password=PASSWORD(%p) and service=web_proxy]
```

When using this convention, you can include "[" characters inside the string, and if you wish to have a "]" character inside the string that will survive the argument parsing, you should use "\[". In other words,

```
[..[..\]..]    -->   ..[..]..
```

Any line in one of the configuration files that is not formatted correctly will generally tend (erring on the side of caution) to make the authentication process fail. A corresponding error is written to the system log files with a call to syslog(3).

28.2.2 Example System Configurations

The following is an example `/etc/pam.d/login` configuration file. This example had all options uncommented and is probably not usable because it stacks many conditions before allowing successful completion of the login process. Essentially, all conditions can be disabled by commenting them out, except the calls to `pam_pwdb.so`.

28.2.2.1 PAM: Original Login Config

```
#%PAM-1.0
# The PAM configuration file for the login service
#
auth         required    pam_securetty.so
auth         required    pam_nologin.so
# auth       required    pam_dialup.so
# auth       optional    pam_mail.so
auth         required    pam_pwdb.so shadow md5
# account    requisite   pam_time.so
account      required    pam_pwdb.so
session      required    pam_pwdb.so
# session    optional    pam_lastlog.so
# password   required    pam_cracklib.so retry=3
password     required    pam_pwdb.so shadow md5
```

28.2.2.2 PAM: Login Using pam_smbpass

PAM allows use of replaceable modules. Those available on a sample system include:

```
$/bin/ls /lib/security
```

```
pam_access.so    pam_ftp.so          pam_limits.so
pam_ncp_auth.so  pam_rhosts_auth.so  pam_stress.so
pam_cracklib.so  pam_group.so        pam_listfile.so
pam_nologin.so   pam_rootok.so       pam_tally.so
pam_deny.so      pam_issue.so        pam_mail.so
pam_permit.so    pam_securetty.so    pam_time.so
pam_dialup.so    pam_lastlog.so      pam_mkhomedir.so
pam_pwdb.so      pam_shells.so       pam_unix.so
pam_env.so       pam_ldap.so         pam_motd.so
pam_radius.so    pam_smbpass.so      pam_unix_acct.so
pam_wheel.so     pam_unix_auth.so    pam_unix_passwd.so
pam_userdb.so    pam_warn.so         pam_unix_session.so
```

The following example for the login program replaces the use of the pam_pwdb.so module that uses the system password database (/etc/passwd, /etc/shadow, /etc/group) with the module pam_smbpass.so, which uses the Samba database containing the Microsoft MD4 encrypted password hashes. This database is stored either in /usr/local/samba/private/smbpasswd, /etc/samba/smbpasswd or in /etc/samba.d/smbpasswd, depending on the Samba implementation for your UNIX/Linux system. The pam_smbpass.so module is provided by Samba version 2.2.1 or later. It can be compiled by specifying the --with-pam_smbpass options when running Samba's **configure** script. For more information on the pam_smbpass module, see the documentation in the source/pam_smbpass directory of the Samba source distribution.

```
#%PAM-1.0
# The PAM configuration file for the login service
#
auth        required    pam_smbpass.so nodelay
account     required    pam_smbpass.so nodelay
session     required    pam_smbpass.so nodelay
password    required    pam_smbpass.so nodelay
```

The following is the PAM configuration file for a particular Linux system. The default condition uses pam_pwdb.so.

```
#%PAM-1.0
# The PAM configuration file for the samba service
#
auth       required     pam_pwdb.so nullok nodelay shadow audit
account    required     pam_pwdb.so audit nodelay
session    required     pam_pwdb.so nodelay
password   required     pam_pwdb.so shadow md5
```

In the following example, the decision has been made to use the **smbpasswd** database even for basic Samba authentication. Such a decision could also be made for the **passwd** program and would thus allow the **smbpasswd** passwords to be changed using the **passwd** program:

```
#%PAM-1.0
# The PAM configuration file for the samba service
#
auth       required     pam_smbpass.so nodelay
account    required     pam_pwdb.so audit nodelay
session    required     pam_pwdb.so nodelay
password   required     pam_smbpass.so nodelay smbconf=/etc/samba.d/smb.conf
```

NOTE

PAM allows stacking of authentication mechanisms. It is also possible to pass information obtained within one PAM module through to the next module in the PAM stack. Please refer to the documentation for your particular system implementation for details regarding the specific capabilities of PAM in this environment. Some Linux implementations also provide the `pam_stack.so` module that allows all authentication to be configured in a single central file. The `pam_stack.so` method has some devoted followers on the basis that it allows for easier administration. As with all issues in life, though, every decision has trade-offs, so you may want to examine the PAM documentation for further helpful information.

28.2.3 smb.conf PAM Configuration

There is an option in `smb.conf` called *obey pam restrictions*. The following is from the online help for this option in SWAT:

> When Samba is configured to enable PAM support (i.e., `--with--pam`), this parameter will control whether or not Samba should obey PAM's account and session management directives. The default behavior is to use PAM for clear-text authentication only and to ignore any account or session management. Samba always ignores PAM for authentication in the case of *encrypt passwords*

= yes. The reason is that PAM modules cannot support the challenge/response authentication mechanism needed in the presence of SMB password encryption.

Default: *obey pam restrictions* = no

28.2.4 Remote CIFS Authentication Using winbindd.so

All operating systems depend on the provision of user credentials acceptable to the platform. UNIX requires the provision of a user identifier (UID) as well as a group identifier (GID). These are both simple integer numbers that are obtained from a password backend such as `/etc/passwd`.

Users and groups on a Windows NT server are assigned a relative ID (RID) which is unique for the domain when the user or group is created. To convert the Windows NT user or group into a UNIX user or group, a mapping between RIDs and UNIX user and group IDs is required. This is one of the jobs that winbind performs.

As winbind users and groups are resolved from a server, user and group IDs are allocated from a specified range. This is done on a first come, first served basis, although all existing users and groups will be mapped as soon as a client performs a user or group enumeration command. The allocated UNIX IDs are stored in a database file under the Samba lock directory and will be remembered.

The astute administrator will realize from this that the combination of `pam_smbpass.so`, **winbindd**, and a distributed *passdb backend* such as *`ldap`* will allow the establishment of a centrally managed, distributed user/password database that can also be used by all PAM-aware (e.g., Linux) programs and applications. This arrangement can have particularly potent advantages compared with the use of Microsoft Active Directory Service (ADS) insofar as the reduction of wide-area network authentication traffic.

WARNING

The RID to UNIX ID database is the only location where the user and group mappings are stored by **winbindd**. If this file is deleted or corrupted, there is no way for **winbindd** to determine which user and group IDs correspond to Windows NT user and group RIDs.

28.2.5 Password Synchronization Using pam_smbpass.so

`pam_smbpass` is a PAM module that can be used on conforming systems to keep the `smbpasswd` (Samba password) database in sync with the UNIX password file. PAM is an API supported under some UNIX operating systems, such as Solaris, HPUX, and Linux, that provides a generic interface to authentication mechanisms.

This module authenticates a local `smbpasswd` user database. If you require support for authenticating against a remote SMB server, or if you are concerned about the presence of SUID root binaries on your system, it is recommended that you use `pam_winbind` instead.

Options recognized by this module are shown in Table 28.1.

The following are examples of the use of `pam_smbpass.so` in the format of the Linux `/etc/pam.d/` files structure. Those wishing to implement this tool on other platforms will need to adapt this appropriately.

28.2.5.1 Password Synchronization Configuration

The following is a sample PAM configuration that shows the use of pam_smbpass to make sure `private/smbpasswd` is kept in sync when `/etc/passwd (/etc/shadow)` is changed. It is useful when an expired password might be changed by an application (such as **ssh**).

```
#%PAM-1.0
# password-sync
#
auth       requisite    pam_nologin.so
```

Table 28.1 Options recognized by pam_smbpass

debug	Log more debugging info.
audit	Like debug, but also logs unknown usernames.
use_first_pass	Do not prompt the user for passwords; take them from PAM_ items instead.
try_first_pass	Try to get the password from a previous PAM module; fall back to prompting the user.
use_authtok	Like try_first_pass, but *fail* if the new PAM_AUTHTOK has not been previously set (intended for stacking password modules only).
not_set_pass	Do not make passwords used by this module available to other modules.
nodelay	dDo not insert ~1-second delays on authentication failure.
nullok	Null passwords are allowed.
nonull	Null passwords are not allowed. Used to override the Samba configuration.
migrate	Only meaningful in an "auth" context; used to update smbpasswd file with a password used for successful authentication.
smbconf=*file*	Specify an alternate path to the `smb.conf` file.

```
auth       required     pam_unix.so
account    required     pam_unix.so
password   requisite    pam_cracklib.so retry=3
password   requisite    pam_unix.so shadow md5 use_authtok try_first_pass
password   required     pam_smbpass.so nullok use_authtok try_first_pass
session    required     pam_unix.so
```

28.2.5.2 Password Migration Configuration

The following PAM configuration shows the use of `pam_smbpass` to migrate from plaintext to encrypted passwords for Samba. Unlike other methods, this can be used for users who have never connected to Samba shares: password migration takes place when users **ftp** in, login using **ssh**, pop their mail, and so on.

```
#%PAM-1.0
# password-migration
#
auth       requisite   pam_nologin.so
# pam_smbpass is called IF pam_unix succeeds.
auth       requisite   pam_unix.so
auth       optional    pam_smbpass.so migrate
account    required    pam_unix.so
password   requisite   pam_cracklib.so retry=3
password   requisite   pam_unix.so shadow md5 use_authtok try_first_pass
password   optional    pam_smbpass.so nullok use_authtok try_first_pass
session    required    pam_unix.so
```

28.2.5.3 Mature Password Configuration

The following is a sample PAM configuration for a mature `smbpasswd` installation. `private/smbpasswd` is fully populated, and we consider it an error if the SMB password does not exist or does not match the UNIX password.

```
#%PAM-1.0
# password-mature
#
auth       requisite    pam_nologin.so
auth       required     pam_unix.so
account    required     pam_unix.so
password   requisite    pam_cracklib.so retry=3
password   requisite    pam_unix.so shadow md5 use_authtok try_first_pass
password   required     pam_smbpass.so use_authtok use_first_pass
session    required     pam_unix.so
```

28.2.5.4 Kerberos Password Integration Configuration

The following is a sample PAM configuration that shows *pam_smbpass* used together with *pam_krb5*. This could be useful on a Samba PDC that is also a member of a Kerberos realm.

```
#%PAM-1.0
# kdc-pdc
#
auth       requisite   pam_nologin.so
auth       requisite   pam_krb5.so
auth       optional    pam_smbpass.so migrate
account    required    pam_krb5.so
password   requisite   pam_cracklib.so retry=3
password   optional    pam_smbpass.so nullok use_authtok try_first_pass
password   required    pam_krb5.so use_authtok try_first_pass
session    required    pam_krb5.so
```

28.3 Common Errors

PAM can be fickle and sensitive to configuration glitches. Here we look at a few cases from the Samba mailing list.

28.3.1 pam_winbind Problem

A user reported, *I have the following PAM configuration*:

```
auth required /lib/security/pam_securetty.so
auth sufficient /lib/security/pam_winbind.so
auth sufficient /lib/security/pam_unix.so use_first_pass nullok
auth required /lib/security/pam_stack.so service=system-auth
auth required /lib/security/pam_nologin.so
account required /lib/security/pam_stack.so service=system-auth
account required /lib/security/pam_winbind.so
password required /lib/security/pam_stack.so service=system-auth
```

When I open a new console with [ctrl][alt][F1], I can't log in with my user "pitie." I have tried with user "scienceu\pitie" also.

The problem may lie with the inclusion of *pam_stack.so service=system-auth*. That file often contains a lot of stuff that may duplicate what you are already doing. Try commenting out the *pam_stack* lines for *auth* and *account* and

see if things work. If they do, look at `/etc/pam.d/system-auth` and copy only what you need from it into your `/etc/pam.d/login` file. Alternatively, if you want all services to use Winbind, you can put the Winbind-specific stuff in `/etc/pam.d/system-auth`.

28.3.2 Winbind Is Not Resolving Users and Groups

"My `smb.conf` file is correctly configured. I have specified *idmap uid* = 12000 and *idmap gid* = 3000-3500, and **winbind** is running. When I do the following it all works fine."

```
root# wbinfo -u
MIDEARTH\maryo
MIDEARTH\jackb
MIDEARTH\ameds
...
MIDEARTH\root

root# wbinfo -g
MIDEARTH\Domain Users
MIDEARTH\Domain Admins
MIDEARTH\Domain Guests
...
MIDEARTH\Accounts

root# getent passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
...
maryo:x:15000:15003:Mary Orville:/home/MIDEARTH/maryo:/bin/false
```

"But this command fails:"

```
root# chown maryo a_file
chown: 'maryo': invalid user
```

"This is driving me nuts! What can be wrong?"

Your system is likely running **nscd**, the name service caching daemon. Shut it down, do not restart it! You will find your problem resolved.

Chapter 29

INTEGRATING MS WINDOWS NETWORKS WITH SAMBA

This chapter deals with NetBIOS over TCP/IP name to IP address resolution. If your MS Windows clients are not configured to use NetBIOS over TCP/IP, then this section does not apply to your installation. If your installation involves the use of NetBIOS over TCP/IP, then this chapter may help you to resolve networking problems.

NOTE

NetBIOS over TCP/IP has nothing to do with NetBEUI. NetBEUI is NetBIOS over Logical Link Control (LLC). On modern networks it is highly advised to not run NetBEUI at all. Note also that there is no such thing as NetBEUI over TCP/IP — the existence of such a protocol is a complete and utter misapprehension.

29.1 Features and Benefits

Many MS Windows network administrators have never been exposed to basic TCP/IP networking as it is implemented in a UNIX/Linux operating

system. Likewise, many UNIX and Linux administrators have not been exposed to the intricacies of MS Windows TCP/IP-based networking (and may have no desire to be, either).

This chapter gives a short introduction to the basics of how a name can be resolved to its IP address for each operating system environment.

29.2 Background Information

Since the introduction of MS Windows 2000, it is possible to run MS Windows networking without the use of NetBIOS over TCP/IP. NetBIOS over TCP/IP uses UDP port 137 for NetBIOS name resolution and uses TCP port 139 for NetBIOS session services. When NetBIOS over TCP/IP is disabled on MS Windows 2000 and later clients, then only the TCP port 445 is used, and the UDP port 137 and TCP port 139 are not.

NOTE

When using Windows 2000 or later clients, if NetBIOS over TCP/IP is not disabled, then the client will use UDP port 137 (NetBIOS Name Service, also known as the Windows Internet Name Service, or WINS), TCP port 139, and TCP port 445 (for actual file and print traffic).

When NetBIOS over TCP/IP is disabled, the use of DNS is essential. Most installations that disable NetBIOS over TCP/IP today use MS Active Directory Service (ADS). ADS requires dynamic DNS with Service Resource Records (SRV RR) and with Incremental Zone Transfers (IXFR). Use of DHCP with ADS is recommended as a further means of maintaining central control over the client workstation network configuration.

29.3 Name Resolution in a Pure UNIX/Linux World

The key configuration files covered in this section are:

- `/etc/hosts`

- /etc/resolv.conf
- /etc/host.conf
- /etc/nsswitch.conf

29.3.1 /etc/hosts

This file contains a static list of IP addresses and names.

```
127.0.0.1   localhost localhost.localdomain
192.168.1.1 bigbox.quenya.org bigbox   alias4box
```

The purpose of /etc/hosts is to provide a name resolution mechanism so users do not need to remember IP addresses.

Network packets that are sent over the physical network transport layer communicate not via IP addresses but rather using the Media Access Control address, or MAC address. IP addresses are currently 32 bits in length and are typically presented as four decimal numbers that are separated by a dot (or period) — for example, 168.192.1.1.

MAC addresses use 48 bits (or 6 bytes) and are typically represented as two-digit hexadecimal numbers separated by colons: 40:8e:0a:12:34:56.

Every network interface must have a MAC address. Associated with a MAC address may be one or more IP addresses. There is no relationship between an IP address and a MAC address; all such assignments are arbitrary or discretionary in nature. At the most basic level, all network communications take place using MAC addressing. Since MAC addresses must be globally unique and generally remain fixed for any particular interface, the assignment of an IP address makes sense from a network management perspective. More than one IP address can be assigned per MAC address. One address must be the primary IP address — this is the address that will be returned in the Address Resolution Protocol (ARP) reply.

When a user or a process wants to communicate with another machine, the protocol implementation ensures that the "machine name" or "host name" is resolved to an IP address in a manner that is controlled by the TCP/IP configuration control files. The file /etc/hosts is one such file.

When the IP address of the destination interface has been determined, a protocol called ARP/RARP is used to identify the MAC address of the target interface. ARP is a broadcast-oriented method that uses User Datagram Protocol (UDP) to send a request to all interfaces on the local network segment using the all 1s MAC address. Network interfaces are programmed to respond to two MAC addresses only; their own unique address and the address ff:ff:ff:ff:ff:ff. The reply packet from an ARP request will contain the MAC address and the primary IP address for each interface.

The `/etc/hosts` file is foundational to all UNIX/Linux TCP/IP installations and as a minimum will contain the localhost and local network interface IP addresses and the primary names by which they are known within the local machine. This file helps to prime the pump so a basic level of name resolution can exist before any other method of name resolution becomes available.

29.3.2 /etc/resolv.conf

This file tells the name resolution libraries:

- The name of the domain to which the machine belongs.
- The name(s) of any domains that should be automatically searched when trying to resolve unqualified host names to their IP address.
- The name or IP address of available domain name servers that may be asked to perform name-to-address translation lookups.

29.3.3 /etc/host.conf

`/etc/host.conf` is the primary means by which the setting in `/etc/resolv.conf` may be effected. It is a critical configuration file. This file controls the order by which name resolution may proceed. The typical structure is:

```
order hosts,bind
multi on
```

Both addresses should be returned. Please refer to the man page for `host.conf` for further details.

29.3.4 /etc/nsswitch.conf

This file controls the actual name resolution targets. The file typically has resolver object specifications as follows:

```
# /etc/nsswitch.conf
#
# Name Service Switch configuration file.
#

passwd:     compat
# Alternative entries for password authentication are:
# passwd:   compat files nis ldap winbind
shadow:     compat
group:      compat

hosts:      files nis dns
# Alternative entries for host name resolution are:
# hosts: files dns nis nis+ hesiod db compat ldap wins
networks:   nis files dns

ethers:     nis files
protocols:  nis files
rpc:     nis files
services:   nis files
```

Of course, each of these mechanisms requires that the appropriate facilities and/or services are correctly configured.

It should be noted that unless a network request/message must be sent, TCP/IP networks are silent. All TCP/IP communications assume a principal of speaking only when necessary.

Starting with version 2.2.0, Samba has Linux support for extensions to the name service switch infrastructure so Linux clients will be able to obtain resolution of MS Windows NetBIOS names to IP addresses. To gain this functionality, Samba needs to be compiled with appropriate arguments to the make command (i.e., `make nsswitch/libnss_wins.so`). The resulting library should then be installed in the `/lib` directory, and the *wins* parameter needs to be added to the "hosts:" line in the `/etc/nsswitch.conf` file.

At this point, it will be possible to ping any MS Windows machine by its NetBIOS machine name, as long as that machine is within the workgroup to which both the Samba machine and the MS Windows machine belong.

29.4 Name Resolution as Used within MS Windows Networking

MS Windows networking is predicated on the name each machine is given. This name is known variously (and inconsistently) as the "computer name," "machine name," "networking name," "NetBIOS name," or "SMB name." All terms mean the same thing with the exception of "NetBIOS name," which can also apply to the name of the workgroup or the domain name. The terms "workgroup" and "domain" are really just a simple name with which the machine is associated. All NetBIOS names are exactly 16 characters in length. The 16th character is reserved. It is used to store a 1-byte value that indicates service level information for the NetBIOS name that is registered. A NetBIOS machine name is therefore registered for each service type that is provided by the client/server.

Table 29.1 and Table 29.2 tables list typical NetBIOS name/service type registrations.

Table 29.1 Unique NetBIOS Names

MACHINENAME<00>	Server Service is running on MACHINENAME
MACHINENAME<03>	Generic machine name (NetBIOS name)
MACHINENAME<20>	LanMan server service is running on MACHINENAME
WORKGROUP<1b>	Domain master browser

Table 29.2 Group Names

WORKGROUP<03>	Generic name registered by all members of WORKGROUP
WORKGROUP<1c>	Domain cntrollers/netlogon servers
WORKGROUP<1d>	Local master browsers
WORKGROUP<1e>	Browser election service

It should be noted that all NetBIOS machines register their own names as per Table 29.1 and Table 29.2. This is in vast contrast to TCP/IP instal-

lations where the system administrator traditionally determines in the `/etc/hosts` or in the DNS database what names are associated with each IP address.

One further point of clarification should be noted. The `/etc/hosts` file and the DNS records do not provide the NetBIOS name information that MS Windows clients depend on to locate the type of service that may be needed. An example of this is what happens when an MS Windows client wants to locate a domain logon server. It finds this service and the IP address of a server that provides it by performing a lookup (via a NetBIOS broadcast) for enumeration of all machines that have registered the name type *<1C>. A logon request is then sent to each IP address that is returned in the enumerated list of IP addresses. Whichever machine first replies, it then ends up providing the logon services.

The name "workgroup" or "domain" really can be confusing, since these have the added significance of indicating what is the security architecture of the MS Windows network. The term "workgroup" indicates that the primary nature of the network environment is that of a peer-to-peer design. In a workgroup, all machines are responsible for their own security, and generally such security is limited to the use of just a password (known as share-level security). In most situations with peer-to-peer networking, the users who control their own machines will simply opt to have no security at all. It is possible to have user-level security in a workgroup environment, thus requiring the use of a username and a matching password.

MS Windows networking is thus predetermined to use machine names for all local and remote machine message passing. The protocol used is called Server Message Block (SMB), and this is implemented using the NetBIOS protocol (Network Basic Input/Output System). NetBIOS can be encapsulated using LLC (Logical Link Control) protocol — in which case the resulting protocol is called NetBEUI (Network Basic Extended User Interface). NetBIOS can also be run over IPX (Internetworking Packet Exchange) protocol as used by Novell NetWare, and it can be run over TCP/IP protocols — in which case the resulting protocol is called NBT or NetBT, the NetBIOS over TCP/IP.

MS Windows machines use a complex array of name resolution mechanisms. Since we are primarily concerned with TCP/IP, this demonstration is limited to this area.

29.4.1 The NetBIOS Name Cache

All MS Windows machines employ an in-memory buffer in which is stored the NetBIOS names and IP addresses for all external machines that machine has communicated with over the past 10 to 15 minutes. It is more efficient to obtain an IP address for a machine from the local cache than it is to go through all the configured name resolution mechanisms.

If a machine whose name is in the local name cache is shut down before the name is expired and flushed from the cache, then an attempt to exchange a message with that machine will be subject to timeout delays. Its name is in the cache, so a name resolution lookup will succeed, but the machine cannot respond. This can be frustrating for users but is a characteristic of the protocol.

The MS Windows utility that allows examination of the NetBIOS name cache is called "nbtstat." The Samba equivalent is called **nmblookup**.

29.4.2 The LMHOSTS File

This file is usually located in MS Windows NT 4.0 or Windows 200x/XP in the directory `%SystemRoot%\SYSTEM32\DRIVERS\ETC` and contains the IP address and the machine name in matched pairs. The `LMHOSTS` file performs NetBIOS name to IP address mapping.

It typically looks like this:

```
# Copyright (c) 1998 Microsoft Corp.
#
# This is a sample LMHOSTS file used by the Microsoft Wins Client (NetBIOS
# over TCP/IP) stack for Windows98
#
# This file contains the mappings of IP addresses to NT computer names
# (NetBIOS) names. Each entry should be kept on an individual line.
# The IP address should be placed in the first column followed by the
# corresponding computer name. The address and the computer name
# should be separated by at least one space or tab. The "#" character
# is generally used to denote the start of a comment (see the exceptions
# below).
#
```

```
# This file is compatible with Microsoft LAN Manager 2.x TCP/IP lmhosts
# files and offers the following extensions:
#
#      #PRE
#      #DOM:<domain>
#      #INCLUDE <filename>
#      #BEGIN_ALTERNATE
#      #END_ALTERNATE
#      \0xnn (non-printing character support)
#
# Following any entry in the file with the characters "#PRE" will cause
# the entry to be preloaded into the name cache. By default, entries are
# not preloaded, but are parsed only after dynamic name resolution fails.
#
# Following an entry with the "#DOM:<domain>" tag will associate the
# entry with the domain specified by <domain>. This effects how the
# browser and logon services behave in TCP/IP environments. To preload
# the host name associated with #DOM entry, it is necessary to also add a
# #PRE to the line. The <domain> is always pre-loaded although it will not
# be shown when the name cache is viewed.
#
# Specifying "#INCLUDE <filename>" will force the RFC NetBIOS (NBT)
# software to seek the specified <filename> and parse it as if it were
# local. <filename> is generally a UNC-based name, allowing a
# centralized lmhosts file to be maintained on a server.
# It is ALWAYS necessary to provide a mapping for the IP address of the
# server prior to the #INCLUDE. This mapping must use the #PRE directive.
# In addition the share "public" in the example below must be in the
# LanMan Server list of "NullSessionShares" in order for client machines to
# be able to read the lmhosts file successfully. This key is under
# \machine\system\currentcontrolset\services\lanmanserver\
# parameters\nullsessionshares
# in the registry. Simply add "public" to the list found there.
#
# The #BEGIN_ and #END_ALTERNATE keywords allow multiple #INCLUDE
# statements to be grouped together. Any single successful include
# will cause the group to succeed.
#
# Finally, non-printing characters can be embedded in mappings by
# first surrounding the NetBIOS name in quotations, then using the
```

```
# \0xnn notation to specify a hex value for a non-printing character.
#
# The following example illustrates all of these extensions:
#
# 102.54.94.97     rhino     #PRE #DOM:networking  #net group's DC
# 102.54.94.102    "appname  \0x14"       #special app server
# 102.54.94.123    popular   #PRE         #source server
# 102.54.94.117    localsrv  #PRE         #needed for the include
#
# #BEGIN_ALTERNATE
# #INCLUDE \\localsrv\public\lmhosts
# #INCLUDE \\rhino\public\lmhosts
# #END_ALTERNATE
#
# In the above example, the "appname" server contains a special
# character in its name, the "popular" and "localsrv" server names are
# pre-loaded, and the "rhino" server name is specified so it can be used
# to later #INCLUDE a centrally maintained lmhosts file if the "localsrv"
# system is unavailable.
#
# Note that the whole file is parsed including comments on each lookup,
# so keeping the number of comments to a minimum will improve performance.
# Therefore it is not advisable to simply add lmhosts file entries onto the
# end of this file.
```

29.4.3 HOSTS File

This file is usually located in MS Windows NT 4.0 or Windows 200x/XP in the directory `%SystemRoot%\SYSTEM32\DRIVERS\ETC` and contains the IP address and the IP hostname in matched pairs. It can be used by the name resolution infrastructure in MS Windows, depending on how the TCP/IP environment is configured. This file is in every way the equivalent of the UNIX/Linux `/etc/hosts` file.

29.4.4 DNS Lookup

This capability is configured in the TCP/IP setup area in the network configuration facility. If enabled, an elaborate name resolution sequence is fol-

lowed, the precise nature of which is dependent on how the NetBIOS Node Type parameter is configured. A Node Type of 0 means that NetBIOS broadcast (over UDP broadcast) is used if the name that is the subject of a name lookup is not found in the NetBIOS name cache. If that fails, then DNS, HOSTS, and LMHOSTS are checked. If set to Node Type 8, then a NetBIOS Unicast (over UDP Unicast) is sent to the WINS server to obtain a lookup before DNS, HOSTS, LMHOSTS, or broadcast lookup is used.

29.4.5 WINS Lookup

A WINS (Windows Internet Name Server) service is the equivalent of the rfc1001/1002 specified NBNS (NetBIOS Name Server). A WINS server stores the names and IP addresses that are registered by a Windows client if the TCP/IP setup has been given at least one WINS server IP address.

To configure Samba to be a WINS server, the following parameter needs to be added to the `smb.conf` file:

```
        wins support = Yes
```

To configure Samba to use a WINS server, the following parameters are needed in the `smb.conf` file:

```
        wins support = No
        wins server = xxx.xxx.xxx.xxx
```

where *xxx.xxx.xxx.xxx* is the IP address of the WINS server.

For information about setting up Samba as a WINS server, read Chapter 10, “Network Browsing”.

29.5 Common Errors

TCP/IP network configuration problems find every network administrator sooner or later. The cause can be anything from keyboard mishaps to forgetfulness to simple mistakes to carelessness. Of course, no one is ever deliberately careless!

29.5.1 Pinging Works Only One Way

"I can ping my Samba server from Windows, but I cannot ping my Windows machine from the Samba server."

The Windows machine was at IP address 192.168.1.2 with netmask 255.255.255.0, the Samba server (Linux) was at IP address 192.168.1.130 with netmask 255.255.255.128. The machines were on a local network with no external connections.

Due to inconsistent netmasks, the Windows machine was on network 192.168.1.0/24, while the Samba server was on network 192.168.1.128/25 — logically a different network.

29.5.2 Very Slow Network Connections

A common cause of slow network response includes:

- Client is configured to use DNS and the DNS server is down.
- Client is configured to use remote DNS server, but the remote connection is down.
- Client is configured to use a WINS server, but there is no WINS server.
- Client is not configured to use a WINS server, but there is a WINS server.
- Firewall is filtering out DNS or WINS traffic.

29.5.3 Samba Server Name-Change Problem

"The name of the Samba server was changed, Samba was restarted, and now the Samba server cannot be pinged by its new name from an MS Windows NT4 workstation, but it does still respond to pinging using the old name. Why?"

From this description, three things are obvious:

- WINS is not in use; only broadcast-based name resolution is used.
- The Samba server was renamed and restarted within the last 10 or 15 minutes.

- The old Samba server name is still in the NetBIOS name cache on the MS Windows NT4 workstation.

To find what names are present in the NetBIOS name cache on the MS Windows NT4 machine, open a **cmd** shell and then:

```
C:\> nbtstat -n

              NetBIOS Local Name Table

   Name                 Type          Status
-----------------------------------------------
FRODO            <03>  UNIQUE      Registered
ADMINISTRATOR     <03>  UNIQUE      Registered
FRODO            <00>  UNIQUE      Registered
SARDON           <00>  GROUP       Registered
FRODO            <20>  UNIQUE      Registered
FRODO            <1F>  UNIQUE      Registered


C:\> nbtstat -c

             NetBIOS Remote Cache Name Table

   Name                 Type       Host Address     Life [sec]
-------------------------------------------------------------
GANDALF  <20>  UNIQUE      192.168.1.1          240

C:\>
```

In this example, GANDALF is the Samba server and FRODO is the MS Windows NT4 workstation. The first listing shows the contents of the Local Name Table (i.e., identity information on the MS Windows workstation), and the second shows the NetBIOS name in the NetBIOS name cache. The name cache contains the remote machines known to this workstation.

Chapter 30

UNICODE/CHARSETS

30.1 Features and Benefits

Every industry eventually matures. One of the great areas of maturation is in the focus that has been given over the past decade to make it possible for anyone anywhere to use a computer. It has not always been that way. In fact, not so long ago, it was common for software to be written for exclusive use in the country of origin.

Of all the effort that has been brought to bear on providing native language support for all computer users, the efforts of the Openi18n organization[1] is deserving of special mention.

Samba-2.x supported a single locale through a mechanism called *codepages*. Samba-3 is destined to become a truly transglobal file- and printer-sharing platform.

30.2 What Are Charsets and Unicode?

Computers communicate in numbers. In texts, each number is translated to a corresponding letter. The meaning that will be assigned to a certain number depends on the *character set (charset)* that is used.

A charset can be seen as a table that is used to translate numbers to letters. Not all computers use the same charset (there are charsets with German umlauts, Japanese characters, and so on). The American Standard Code for Information Interchange (ASCII) encoding system has been the normative

[1] <http://www.openi18n.org/>

character encoding scheme used by computers to date. This employs a charset that contains 256 characters. Using this mode of encoding, each character takes exactly one byte.

There are also charsets that support extended characters, but those need at least twice as much storage space as does ASCII encoding. Such charsets can contain **256 * 256 = 65536** characters, which is more than all possible characters one could think of. They are called multibyte charsets because they use more then one byte to store one character.

One standardized multibyte charset encoding scheme is known as unicode[2]. A big advantage of using a multibyte charset is that you only need one. There is no need to make sure two computers use the same charset when they are communicating.

Old Windows clients use single-byte charsets, named *codepages*, by Microsoft. However, there is no support for negotiating the charset to be used in the SMB/CIFS protocol. Thus, you have to make sure you are using the same charset when talking to an older client. Newer clients (Windows NT, 200x, XP) talk Unicode over the wire.

30.3 Samba and Charsets

As of Samba-3, Samba can (and will) talk Unicode over the wire. Internally, Samba knows of three kinds of character sets:

unix charset This is the charset used internally by your operating system. The default is `UTF-8`, which is fine for most systems and covers all characters in all languages. The default in previous Samba releases was to save filenames in the encoding of the clients — for example, CP850 for Western European countries.

display charset This is the charset Samba uses to print messages on your screen. It should generally be the same as the *unix charset*.

dos charset This is the charset Samba uses when communicating with DOS and Windows 9x/Me clients. It will talk Unicode to all newer

[2] <http://www.unicode.org/>

clients. The default depends on the charsets you have installed on your system. Run **testparm -v | grep "dos charset"** to see what the default is on your system.

30.4 Conversion from Old Names

Because previous Samba versions did not do any charset conversion, characters in filenames are usually not correct in the UNIX charset but only for the local charset used by the DOS/Windows clients.

Bjoern Jacke has written a utility named convmv[3] that can convert whole directory structures to different charsets with one single command.

30.5 Japanese Charsets

Setting up Japanese charsets is quite difficult. This is mainly because:

- The Windows character set is extended from the original legacy Japanese standard (JIS X 0208) and is not standardized. This means that the strictly standardized implementation cannot support the full Windows character set.
- Mainly for historical reasons, there are several encoding methods in Japanese, which are not fully compatible with each other. There are two major encoding methods. One is the Shift_JIS series used in Windows and some UNIXes. The other is the EUC-JP series used in most UNIXes and Linux. Moreover, Samba previously also offered several unique encoding methods, named CAP and HEX, to keep interoperability with CAP/NetAtalk and UNIXes that can't use Japanese filenames. Some implementations of the EUC-JP series can't support the full Windows character set.
- There are some code conversion tables between Unicode and legacy Japanese character sets. One is compatible with Windows, another one is based on the reference of the Unicode consortium, and others are a mixed implementation. The Unicode consortium does not officially define any conversion tables between Unicode and legacy character sets, so there cannot be standard one.

[3] <http://j3e.de/linux/convmv/>

- The character set and conversion tables available in iconv() depend on the iconv library that is available. Next to that, the Japanese locale names may be different on different systems. This means that the value of the charset parameters depends on the implementation of iconv() you are using.

  Though 2-byte fixed UCS-2 encoding is used in Windows internally, Shift_JIS series encoding is usually used in Japanese environments as ASCII encoding is in English environments.

30.5.1 Basic Parameter Setting

The *dos charset* and *display charset* should be set to the locale compatible with the character set and encoding method used on Windows. This is usually CP932 but sometimes has a different name.

The *unix charset* can be either Shift_JIS series, EUC-JP series, or UTF-8. UTF-8 is always available, but the availability of other locales and the name itself depends on the system.

Additionally, you can consider using the Shift_JIS series as the value of the *unix charset* parameter by using the vfs_cap module, which does the same thing as setting "coding system = CAP" in the Samba 2.2 series.

Where to set *unix charset* to is a difficult question. Here is a list of details, advantages, and disadvantages of using a certain value.

Shift_JIS series Shift_JIS series means a locale that is equivalent to `Shift_JIS`, used as a standard on Japanese Windows. In the case of `Shift_JIS`, for example, if a Japanese filename consists of 0x8ba4 and 0x974c (a 4-bytes Japanese character string meaning "share") and ".txt" is written from Windows on Samba, the filename on UNIX becomes 0x8ba4, 0x974c, ".txt" (an 8-byte BINARY string), same as Windows.

Since Shift_JIS series is usually used on some commercial-based UNIXes; hp-ux and AIX as the Japanese locale (however, it is also possible to use the EUC-JP locale series). To use Shift_JIS series on these platforms, Japanese filenames created from Windows can be referred to also on UNIX.

If your UNIX is already working with Shift_JIS and there is a user who needs to use Japanese filenames written from Windows, the Shift_JIS

series is the best choice. However, broken filenames may be displayed, and some commands that cannot handle non-ASCII filenames may be aborted during parsing filenames. Especially, there may be "\ (0x5c)" in filenames, which need to be handled carefully. It is best to not touch filenames written from Windows on UNIX.

Note that most Japanized free software actually works with EUC-JP only. It is good practice to verify that the Japanized free software can work with Shift_JIS.

EUC-JP series EUC-JP series means a locale that is equivalent to the industry standard called EUC-JP, widely used in Japanese UNIX (although EUC contains specifications for languages other than Japanese, such as EUC-KR). In the case of EUC-JP series, for example, if a Japanese filename consists of 0x8ba4 and 0x974c and ".txt" is written from Windows on Samba, the filename on UNIX becomes 0xb6a6, 0xcdad, ".txt" (an 8-byte BINARY string).

Since EUC-JP is usually used on open source UNIX, Linux, and FreeBSD, and on commercial-based UNIX, Solaris, IRIX, and Tru64 UNIX as Japanese locale (however, it is also possible on Solaris to use Shift_JIS and UTF-8, and on Tru64 UNIX it is possible to use Shift_JIS). To use EUC-JP series, most Japanese filenames created from Windows can be referred to also on UNIX. Also, most Japanized free software works mainly with EUC-JP only.

It is recommended to choose EUC-JP series when using Japanese filenames on UNIX.

Although there is no character that needs to be carefully treated like "\ (0x5c)", broken filenames may be displayed and some commands that cannot handle non-ASCII filenames may be aborted during parsing filenames.

Moreover, if you built Samba using differently installed libiconv, the eucJP-ms locale included in libiconv and EUC-JP series locale included in the operating system may not be compatible. In this case, you may need to avoid using incompatible characters for filenames.

UTF-8 UTF-8 means a locale equivalent to UTF-8, the international standard defined by the Unicode consortium. In UTF-8, a *character* is

expressed using 1 to 3 bytes. In case of the Japanese language, most characters are expressed using 3 bytes. Since on Windows Shift_JIS, where a character is expressed with 1 or 2 bytes is used to express Japanese, basically a byte length of a UTF-8 string the length of the UTF-8 string is 1.5 times that of the original Shift_JIS string. In the case of UTF-8, for example, if a Japanese filename consists of 0x8ba4 and 0x974c, and ".txt" is written from Windows on Samba, the filename on UNIX becomes 0xe585, 0xb1e6, 0x9c89, ".txt" (a 10-byte BINARY string).

For systems where iconv() is not available or where iconv()'s locales are not compatible with Windows, UTF-8 is the only locale available.

There are no systems that use UTF-8 as the default locale for Japanese.

Some broken filenames may be displayed, and some commands that cannot handle non-ASCII filenames may be aborted during parsing filenames. Especially, there may be "\ (0x5c)" in filenames, which must be handled carefully, so you had better not touch filenames written from Windows on UNIX.

In addition, although it is not directly concerned with Samba, since there is a delicate difference between the iconv() function, which is generally used on UNIX, and the functions used on other platforms, such as Windows and Java, so far is concerens the conversion between Shift_JIS and Unicode UTF-8 must be done with care and recognition of the limitations involved in the process.

Although Mac OS X uses UTF-8 as its encoding method for filenames, it uses an extended UTF-8 specification that Samba cannot handle, so UTF-8 locale is not available for Mac OS X.

Shift_JIS series + vfs_cap (CAP encoding) CAP encoding means a specification used in CAP and NetAtalk, file server software for Macintosh. In the case of CAP encoding, for example, if a Japanese filename consists of 0x8ba4 and 0x974c, and ".txt" is written from Windows on Samba, the filename on UNIX becomes ":8b:a4:97L.txt" (a 14 bytes ASCII string).

For CAP encoding, a byte that cannot be expressed as an ASCII character (0x80 or above) is encoded in an ":xx" form. You need to take

care of containing a "\(0x5c)" in a filename, but filenames are not broken in a system that cannot handle non-ASCII filenames.

The greatest merit of CAP encoding is the compatibility of encoding filenames with CAP or NetAtalk. These are respectively the Columbia Appletalk Protocol, and the NetAtalk Open Source software project. Since these software applications write a file name on UNIX with CAP encoding, if a directory is shared with both Samba and NetAtalk, you need to use CAP encoding to avoid non-ASCII filenames from being broken.

However, recently, NetAtalk has been patched on some systems to write filenames with EUC-JP (e.g., Japanese original Vine Linux). In this case, you need to choose EUC-JP series instead of CAP encoding.

vfs_cap itself is available for non-Shift_JIS series locales for systems that cannot handle non-ASCII characters or systems that share files with NetAtalk.

To use CAP encoding on Samba-3, you should use the unix charset parameter and VFS as in Example 30.5.1.

Example 30.5.1 VFS CAP

```
[global]
 # the locale name "CP932" may be different
        dos charset = CP932
        unix charset = CP932
 [cap-share]
        vfs option = cap
```

You should set CP932 if using GNU libiconv for unix charset. With this setting, filenames in the "cap-share" share are written with CAP encoding.

30.5.2 Individual Implementations

Here is some additional information regarding individual implementations:

GNU libiconv To handle Japanese correctly, you should apply the patch libiconv-1.8-cp932-patch.diff.gz[4] to libiconv-1.8.

Using the patched libiconv-1.8, these settings are available:

```
dos charset = CP932
unix charset = CP932 / eucJP-ms / UTF-8
      |       |
      |       +-- EUC-JP series
      +-- Shift_JIS series
display charset = CP932
```

Other Japanese locales (for example, Shift_JIS and EUC-JP) should not be used because of the lack of the compatibility with Windows.

GNU glibc To handle Japanese correctly, you should apply a patch[5] to glibc-2.2.5/2.3.1/2.3.2 or should use the patch-merged versions, glibc-2.3.3 or later.

Using the above glibc, these setting are available:

```
dos charset = CP932
unix charset = CP932 / eucJP-ms / UTF-8
display charset = CP932
```

Other Japanese locales (for example, Shift_JIS and EUC-JP) should not be used because of the lack of the compatibility with Windows.

30.5.3 Migration from Samba-2.2 Series

Prior to Samba-2.2 series, the "coding system" parameter was used. The default codepage in Samba 2.x was code page 850. In the Samba-3 series this has been replaced with the *unix charset* parameter. Table 30.1 shows the mapping table when migrating from the Samba-2.2 series to Samba-3.

[4] <http://www2d.biglobe.ne.jp/~msyk/software/libiconv-patch.html>
[5] <http://www2d.biglobe.ne.jp/~msyk/software/glibc/>

Table 30.1 Japanese Character Sets in Samba-2.2 and Samba-3

Samba-2.2 Coding System	Samba-3 unix charset
SJIS	Shift_JIS series
EUC	EUC-JP series
EUC3[a]	EUC-JP series
CAP	Shift_JIS series + VFS
HEX	currently none
UTF8	UTF-8
UTF8-Mac[b]	currently none
others	none

[a]Only exists in Japanese Samba version
[b]Only exists in Japanese Samba version

30.6 Common Errors

30.6.1 CP850.so Can't Be Found

"Samba is complaining about a missing `CP850.so` file."

CP850 is the default *dos charset*. The *dos charset* is used to convert data to the codepage used by your DOS clients. If you do not have any DOS clients, you can safely ignore this message.

CP850 should be supported by your local iconv implementation. Make sure you have all the required packages installed. If you compiled Samba from source, make sure that the configure process found iconv. This can be confirmed by checking the `config.log` file that is generated when **configure** is executed.

Chapter 31

BACKUP TECHNIQUES

31.1 Features and Benefits

The Samba project is over 10 years old. During the early history of Samba, UNIX administrators were its key implementors. UNIX administrators use UNIX system tools to backup UNIX system files. Over the past 4 years, an increasing number of Microsoft network administrators have taken an interest in Samba. This is reflected in the questions about backup in general on the Samba mailing lists.

31.2 Discussion of Backup Solutions

During discussions at a Microsoft Windows training course, one of the pro-UNIX delegates stunned the class when he pointed out that Windows NT4 is limiting compared with UNIX. He likened UNIX to a Meccano set that has an unlimited number of tools that are simple, efficient, and, in combination, capable of achieving any desired outcome.

One of the Windows networking advocates retorted that if she wanted a Meccano set, she would buy one. She made it clear that a complex single tool that does more than is needed but does it with a clear purpose and intent is preferred by some like her.

Please note that all information here is provided as is and without recommendation of fitness or suitability. The network administrator is strongly encouraged to perform due diligence research before implementing any backup solution, whether free software or commercial.

A useful Web site I recently stumbled across that you might like to refer to is located at www.allmerchants.com[1].

The following three free software projects might also merit consideration.

31.2.1 BackupPC

BackupPC version 2.0.0 has been released on SourceForge[2]. New features include support for **rsync/rsyncd** and internationalization of the CGI interface (including English, French, Spanish, and German).

BackupPC is a high-performance Perl-based package for backing up Linux, UNIX, and Windows PCs and laptops to a server's disk. BackupPC is highly configurable and easy to install and maintain. SMB (via smbclient), **tar** over **rsh/ssh**, or **rsync/rsyncd** are used to extract client data.

Given the ever-decreasing cost of disks and RAID systems, it is now practical and cost effective to backup a large number of machines onto a server's local disk or network storage. This is what BackupPC does.

Key features are pooling of identical files (big savings in server disk space), compression, and a comprehensive CGI interface that allows users to browse backups and restore files.

BackupPC is free software distributed under a GNU GPL license. BackupPC runs on Linux/UNIX/freenix servers and has been tested on Linux, UNIX, Windows 9x/Me, Windows 98, Windows 200x, Windows XP, and Mac OSX clients.

31.2.2 Rsync

rsync is a flexible program for efficiently copying files or directory trees.

rsync has many options to select which files will be copied and how they are to be transferred. It may be used as an alternative to **ftp, http, scp**, or **rcp**.

The rsync remote-update protocol allows rsync to transfer just the differences between two sets of files across the network link, using an efficient

[1] <http://www.allmerchants.com/Software/Backup_Software/>
[2] <http://backuppc.sourceforge.net>

checksum-search algorithm described in the technical report that accompanies the rsync package.

Some of the additional features of rsync are:

- Support for copying links, devices, owners, groups, and permissions.
- Exclude and exclude-from options are similar to GNU tar.
- A CVS exclude mode for ignoring the same files that CVS would ignore.
- Can use any transparent remote shell, including rsh or ssh.
- Does not require root privileges.
- Pipelining of file transfers to minimize latency costs.
- Support for anonymous or authenticated rsync servers (ideal for mirroring).

31.2.3 Amanda

Amanda, the Advanced Maryland Automatic Network Disk Archiver, is a backup system that allows the administrator of a LAN to set up a single master backup server to back up multiple hosts to a single large capacity tape drive. Amanda uses native dump and/or GNU tar facilities and can back up a large number of workstations running multiple versions of UNIX. Recent versions can also use Samba to back up Microsoft Windows hosts.

For more information regarding Amanda, please check the www.amanda.org/site[3].

31.2.4 BOBS: Browseable Online Backup System

Browseable Online Backup System (BOBS) is a complete online backup system. Uses large disks for storing backups and lets users browse the files using a Web browser. Handles some special files like AppleDouble and icon files.

The home page for BOBS is located at bobs.sourceforge.net[4].

[3] <http://www.amanda.org/>
[4] <http://bobs.sourceforge.net/>

Chapter 32

HIGH AVAILABILITY

32.1 Features and Benefits

Network administrators are often concerned about the availability of file and print services. Network users are inclined toward intolerance of the services they depend on to perform vital task responsibilities.

A sign in a computer room served to remind staff of their responsibilities. It read:

> All humans fail, in both great and small ways we fail continually. Machines fail too. Computers are machines that are managed by humans, the fallout from failure can be spectacular. Your responsibility is to deal with failure, to anticipate it and to eliminate it as far as is humanly and economically wise to achieve. Are your actions part of the problem or part of the solution?

If we are to deal with failure in a planned and productive manner, then first we must understand the problem. That is the purpose of this chapter.

Parenthetically, in the following discussion there are seeds of information on how to provision a network infrastructure against failure. Our purpose here is not to provide a lengthy dissertation on the subject of high availability. Additionally, we have made a conscious decision to not provide detailed working examples of high availability solutions; instead we present an overview of the issues in the hope that someone will rise to the challenge of providing a detailed document that is focused purely on presentation of the current state of knowledge and practice in high availability as it applies to the deployment of Samba and other CIFS/SMB technologies.

32.2 Technical Discussion

The following summary was part of a presentation by Jeremy Allison at the SambaXP 2003 conference that was held at Goettingen, Germany, in April 2003. Material has been added from other sources, but it was Jeremy who inspired the structure that follows.

32.2.1 The Ultimate Goal

All clustering technologies aim to achieve one or more of the following:

- Obtain the maximum affordable computational power.
- Obtain faster program execution.
- Deliver unstoppable services.
- Avert points of failure.
- Exact most effective utilization of resources.

A clustered file server ideally has the following properties:

- All clients can connect transparently to any server.
- A server can fail and clients are transparently reconnected to another server.
- All servers serve out the same set of files.
- All file changes are immediately seen on all servers.
  - Requires a distributed file system.
- Infinite ability to scale by adding more servers or disks.

32.2.2 Why Is This So Hard?

In short, the problem is one of *state*.

- All TCP/IP connections are dependent on state information.

  The TCP connection involves a packet sequence number. This sequence number would need to be dynamically updated on all machines in the cluster to effect seamless TCP failover.

- CIFS/SMB (the Windows networking protocols) uses TCP connections.

  This means that from a basic design perspective, failover is not seriously considered.

  - All current SMB clusters are failover solutions — they rely on the clients to reconnect. They provide server failover, but clients can lose information due to a server failure.

- Servers keep state information about client connections.

  - CIFS/SMB involves a lot of state.
  - Every file open must be compared with other open files to check share modes.

32.2.2.1 The Front-End Challenge

To make it possible for a cluster of file servers to appear as a single server that has one name and one IP address, the incoming TCP data streams from clients must be processed by the front-end virtual server. This server must de-multiplex the incoming packets at the SMB protocol layer level and then feed the SMB packet to different servers in the cluster.

One could split all IPC$ connections and RPC calls to one server to handle printing and user lookup requirements. RPC printing handles are shared between different IPC4 sessions — it is hard to split this across clustered servers!

Conceptually speaking, all other servers would then provide only file services. This is a simpler problem to concentrate on.

32.2.2.2 Demultiplexing SMB Requests

De-multiplexing of SMB requests requires knowledge of SMB state information, all of which must be held by the front-end *virtual* server. This is a perplexing and complicated problem to solve.

Windows XP and later have changed semantics so state information (vuid, tid, fid) must match for a successful operation. This makes things simpler than before and is a positive step forward.

SMB requests are sent by vuid to their associated server. No code exists today to effect this solution. This problem is conceptually similar to the problem of correctly handling requests from multiple requests from Windows 2000 Terminal Server in Samba.

One possibility is to start by exposing the server pool to clients directly. This could eliminate the de-multiplexing step.

32.2.2.3 The Distributed File System Challenge

There exists many distributed file systems for UNIX and Linux.

Many could be adopted to backend our cluster, so long as awareness of SMB semantics is kept in mind (share modes, locking, and oplock issues in particular). Common free distributed file systems include:

- NFS
- AFS
- OpenGFS
- Lustre

The server pool (cluster) can use any distributed file system backend if all SMB semantics are performed within this pool.

32.2.2.4 Restrictive Constraints on Distributed File Systems

Where a clustered server provides purely SMB services, oplock handling may be done within the server pool without imposing a need for this to be passed to the backend file system pool.

On the other hand, where the server pool also provides NFS or other file services, it will be essential that the implementation be oplock-aware so it can interoperate with SMB services. This is a significant challenge today. A failure to provide this interoperability will result in a significant loss of performance that will be sorely noted by users of Microsoft Windows clients.

Last, all state information must be shared across the server pool.

32.2.2.5 Server Pool Communications

Most backend file systems support POSIX file semantics. This makes it difficult to push SMB semantics back into the file system. POSIX locks have different properties and semantics from SMB locks.

All **smbd** processes in the server pool must of necessity communicate very quickly. For this, the current *tdb* file structure that Samba uses is not suitable for use across a network. Clustered **smbd**s must use something else.

32.2.2.6 Server Pool Communications Demands

High-speed interserver communications in the server pool is a design prerequisite for a fully functional system. Possibilities for this include:

- Proprietary shared memory bus (example: Myrinet or SCI [scalable coherent interface]). These are high-cost items.
- Gigabit Ethernet (now quite affordable).
- Raw Ethernet framing (to bypass TCP and UDP overheads).

We have yet to identify metrics for performance demands to enable this to happen effectively.

32.2.2.7 Required Modifications to Samba

Samba needs to be significantly modified to work with a high-speed server interconnect system to permit transparent failover clustering.

Particular functions inside Samba that will be affected include:

- The locking database, oplock notifications, and the share mode database.
- Failure semantics need to be defined. Samba behaves the same way as Windows. When oplock messages fail, a file open request is allowed, but this is potentially dangerous in a clustered environment. So how should interserver pool failure semantics function, and how should such functionality be implemented?
- Should this be implemented using a point-to-point lock manager, or can this be done using multicast techniques?

32.2.3 A Simple Solution

Allowing failover servers to handle different functions within the exported file system removes the problem of requiring a distributed locking protocol.

If only one server is active in a pair, the need for high-speed server interconnect is avoided. This allows the use of existing high-availability solutions, instead of inventing a new one. This simpler solution comes at a price — the cost of which is the need to manage a more complex file name space. Since there is now not a single file system, administrators must remember where all services are located — a complexity not easily dealt with.

The *virtual server* is still needed to redirect requests to backend servers. Backend file space integrity is the responsibility of the administrator.

32.2.4 High-Availability Server Products

Failover servers must communicate in order to handle resource failover. This is essential for high-availability services. The use of a dedicated heartbeat is a common technique to introduce some intelligence into the failover process. This is often done over a dedicated link (LAN or serial).

Many failover solutions (like Red Hat Cluster Manager and Microsoft Wolfpack) can use a shared SCSI of Fiber Channel disk storage array for failover communication. Information regarding Red Hat high availability solutions for Samba may be obtained from www.redhat.com[1].

The Linux High Availability project is a resource worthy of consultation if your desire is to build a highly available Samba file server solution. Please consult the home page at www.linux-ha.org/[2].

Front-end server complexity remains a challenge for high availability because it must deal gracefully with backend failures, while at the same time providing continuity of service to all network clients.

[1] <http://www.redhat.com/docs/manuals/enterprise/RHEL-AS-2.1-Manual/cluster-manager/s1-service-samba.html>

[2] <http://www.linux-ha.org/>

32.2.5 MS-DFS: The Poor Man's Cluster

MS-DFS links can be used to redirect clients to disparate backend servers. This pushes complexity back to the network client, something already included by Microsoft. MS-DFS creates the illusion of a simple, continuous file system name space that works even at the file level.

Above all, at the cost of complexity of management, a distributed system (pseudo-cluster) can be created using existing Samba functionality.

32.2.6 Conclusions

- Transparent SMB clustering is hard to do!
- Client failover is the best we can do today.
- Much more work is needed before a practical and manageable high-availability transparent cluster solution will be possible.
- MS-DFS can be used to create the illusion of a single transparent cluster.

Chapter 33

HANDLING LARGE DIRECTORIES

Samba-3.0.12 and later implements a solution for sites that have experienced performance degradation due to the problem of using Samba-3 with applications that need large numbers of files (100,000 or more) per directory.

The key was fixing the directory handling to read only the current list requested instead of the old (up to samba-3.0.11) behavior of reading the entire directory into memory before doling out names. Normally this would have broken OS/2 applications, which have very strange delete semantics, but by stealing logic from Samba4 (thanks, Tridge), the current code in 3.0.12 handles this correctly.

To set up an application that needs large numbers of files per directory in a way that does not damage performance unduly, follow these steps:

First, you need to canonicalize all the files in the directory to have one case, upper or lower — take your pick (I chose upper because all my files were already uppercase names). Then set up a new custom share for the application as follows:

```
[bigshare]
        path = /data/manyfilesdir
        read only = no
        case sensitive = True
        default case = upper
        preserve case = no
        short preserve case = no
```

Of course, use your own path and settings, but set the case options to match the case of all the files in your directory. The path should point at the large directory needed for the application — any new files created in there and in any paths under it will be forced by smbd into uppercase, but smbd will no longer have to scan the directory for names: it knows that if a file does not exist in uppercase, then it doesn't exist at all.

The secret to this is really in the *case sensitive* = True line. This tells smbd never to scan for case-insensitive versions of names. So if an application asks for a file called `FOO`, and it cannot be found by a simple stat call, then smbd will return file not found immediately without scanning the containing directory for a version of a different case. The other `xxx case xxx` lines make this work by forcing a consistent case on all files created by smbd.

Remember, all files and directories under the ***path*** directory must be in uppercase with this `smb.conf` stanza because smbd will not be able to find lowercase filenames with these settings. Also note that this is done on a per-share basis, allowing this parameter to be set only for a share servicing an application with this problematic behavior (using large numbers of entries in a directory) — the rest of your smbd shares don't need to be affected.

This makes smbd much faster when dealing with large directories. My test case has over 100,000 files, and smbd now deals with this very efficiently.

Chapter 34

ADVANCED CONFIGURATION TECHNIQUES

Since the release of the first edition of this book there have been repeated requests to better document configuration techniques that may help a network administrator to get more out of Samba. Some users have asked for documentation regarding the use of the *include* = file-name parameter.

Commencing around mid-2004 there has been increasing interest in the ability to host multiple Samba servers on one machine. There has also been an interest in the hosting of multiple Samba server personalities on one server.

Feedback from technical reviewers made the inclusion of this chapter a necessity. So, here is an answer the questions that have to date not been adequately addressed. Additional user input is welcome as it will help this chapter to mature. What is presented here is just a small beginning.

There are a number of ways in which multiple servers can be hosted on a single Samba server. Multiple server hosting makes it possible to host multiple domain controllers on one machine. Each such machine is independent, and each can be stopped or started without affecting another.

Sometimes it is desirable to host multiple servers, each with its own security mode. For example, a single UNIX/Linux host may be a domain member server (DMS) as well as a generic anonymous print server. In this case, only domain member machines and domain users can access the DMS, but even guest users can access the generic print server. Another example of a

situation where it may be beneficial to host a generic (anonymous) server is to host a CDROM server.

Some environments dictate the need to have separate servers, each with their own resources, each of which are accessible only by certain users or groups. This is one of the simple, but highly effective, ways that Samba can replace many physical Windows servers in one Samba installation.

34.1 Implementation

34.1.1 Multiple Server Hosting

The use of multiple server hosting involves running multiple separate instances of Samba, each with it's own configuration file. This method is complicated by the fact that each instance of nmbd, smbd and winbindd must have write access to entirely separate TDB files. The ability to keep separate the TDB files used by nmbd, smbd and winbindd can be enabled either by recompiling Samba for each server hosted so each has its own default TDB directories, or by configuring these in the `smb.conf` file, in which case each instance of nmbd, smbd and winbindd must be told to start up with its own `smb.conf` configuration file.

Each instance should operate on its own IP address (that independent IP address can be an IP Alias). Each instance of nmbd, smbd and winbindd should listen only on its own IP socket. This can be secured using the *socket address* parameter. Each instance of the Samba server will have its own SID also, this means that the servers are discrete and independent of each other.

The user of multiple server hosting is non-trivial, and requires careful configuration of each aspect of process management and start up. The `smb.conf` parameters that must be carefully configured includes: *private dir*, *pid directory*,*lock directory*, *interfaces*, *bind interfaces only*, *netbios name*, *workgroup*, *socket address*.

Those who elect to create multiple Samba servers should have the ability to read and follow the Samba source code, and to modify it as needed. This mode of deployment is considered beyond the scope of this book. However, if someone will contribute more comprehensive documentation we will gladly review it, and if it is suitable extend this section of this chapter. Until such documentation becomes available the hosting of multiple samba servers on a single host is considered not supported for Samba-3 by the Samba Team.

34.1.2 Multiple Virtual Server Personalities

Samba has the ability to host multiple virtual servers, each of which have their own personality. This is achieved by configuring an `smb.conf` file that is common to all personalities hosted. Each server personality is hosted using its own *netbios alias* name, and each has its own distinct *[global]* section. Each server may have its own stanzas for services and meta-services.

When hosting multiple virtual servers, each with their own personality, each can be in a different workgroup. Only the primary server can be a domain member or a domain controller. The personality is defined by the combination of the *security* mode it is operating in, the *netbios aliases* it has, and the *workgroup* that is defined for it.

This configuration style can be used either with NetBIOS names, or using NetBIOS-less SMB over TCP services. If run using NetBIOS mode (the most common method) it is important that the parameter *smb ports* = 139 should be specified in the primary `smb.conf` file. Failure to do this will result in Samba operating over TCP port 445 and problematic operation at best, and at worst only being able to obtain the functionality that is specified in the primary `smb.conf` file. The use of NetBIOS over TCP/IP using only TCP port 139 means that the use of the `%L` macro is fully enabled. If the *smb ports* = 139 is not specified (the default is *445 139*, or if the value of this parameter is set at *139 445* then the `%L` macro is not serviceable.

It is possible to host multiple servers, each with their own personality, using port 445 (the NetBIOS-less SMB port), in which case the `%i` macro can be used to provide separate server identities (by IP Address). Each can have its own *security* mode. It will be necessary to use the *interfaces*, *bind interfaces only* and IP aliases in addition to the *netbios name* parameters to create the virtual servers. This method is considerably more complex than that using NetBIOS names only using TCP port 139.

Consider an example environment that consists of a standalone, user-mode security Samba server and a read-only Windows 95 file server that has to be replaced. Instead of replacing the Windows 95 machine with a new PC, it is possible to add this server as a read-only anonymous file server that is hosted on the Samba server. Here are some parameters:

The Samba server is called `ELASTIC`, its workgroup name is `ROBINSNEST`. The CDROM server is called `CDSERVER` and its workgroup is `ARTSDEPT`. A possible implementation is shown here:

The `smb.conf` file for the master server is shown in Example 34.1.1. This file is placed in the `/etc/samba` directory. Only the nmbd and the smbd daemons are needed. When started the server will appear in Windows Network Neighborhood as the machine `ELASTIC` under the workgroup `ROBINSNEST`. It is helpful if the Windows clients that must access this server are also in the workgroup `ROBINSNEST` as this will make browsing much more reliable.

Example 34.1.1 Elastic smb.conf File

```
# Global parameters
 [global]
        workgroup = ROBINSNEST
        netbios name = ELASTIC
        netbios aliases = CDSERVER
        smb ports = 139
        printcap name = cups
        disable spoolss = Yes
        show add printer wizard = No
        printing = cups
        include = /etc/samba/smb-%L.conf
 [homes]
        comment = Home Directories
        valid users = %S
        read only = No
        browseable = No
 [office]
        comment = Data
        path = /data
        read only = No
 [printers]
        comment = All Printers
        path = /var/spool/samba
        create mask = 0600
        guest ok = Yes
        printable = Yes
        use client driver = Yes
        browseable = No
```

The configuration file for the CDROM server is listed in Example 34.1.2. This file is called `smb-cdserver.conf` and it should be located in the `/etc/samba` directory. Machines that are in the workgroup `ARTSDEPT` will be able to browse this server freely.

Example 34.1.2 CDROM Server smb-cdserver.conf file

```
# Global parameters
 [global]
        workgroup = ARTSDEPT
        netbios name = CDSERVER
        map to guest = Bad User
        guest ok = Yes
 [carousel]
        comment = CDROM Share
        path = /export/cddata
        read only = Yes
        guest ok = Yes
```

The two servers have different resources and are in separate workgroups. The server `ELASTIC` can only be accessed by uses who have an appropriate account on the host server. All users will be able to access the CDROM data that is stored in the `/export/cddata` directory. File system permissions should set so that the `others` user has read-only access to the directory and its contents. The files can be owned by root (any user other than the nobody account).

34.1.3 Multiple Virtual Server Hosting

In this example, the requirement is for a primary domain controller for the domain called `MIDEARTH`. The PDC will be called `MERLIN`. An extra machine called `SAURON` is required. Each machine will have only its own shares. Both machines belong to the same domain/workgroup.

The master `smb.conf` file is shown in Example 34.1.3. The two files that specify the share information for each server are shown in Example 34.1.4, and Example 34.1.5. All three files are locate in the `/etc/samba` directory.

Example 34.1.3 Master smb.conf File Global Section

```
# Global parameters
 [global]
        workgroup = MIDEARTH
        netbios name = MERLIN
        netbios aliases = SAURON
        passdb backend = tdbsam
        smb ports = 139
        syslog = 0
        printcap name = CUPS
        show add printer wizard = No
        add user script = /usr/sbin/useradd -m '%u'
        delete user script = /usr/sbin/userdel -r '%u'
        add group script = /usr/sbin/groupadd '%g'
        delete group script = /usr/sbin/groupdel '%g'
        add user to group script = /usr/sbin/usermod -G ↩
            '%g' '%u'
        add machine script = /usr/sbin/useradd -s /bin/ ↩
            false -d /var/lib/nobody '%u'
        logon script = scripts\login.bat
        logon path =
        logon drive = X:
        domain logons = Yes
        preferred master = Yes
        wins support = Yes
        printing = CUPS
        include = /etc/samba/smb-%L.conf
```

Example 34.1.4 MERLIN smb-merlin.conf File Share Section

```
# Global parameters
 [global]
        workgroup = MIDEARTH
        netbios name = MERLIN
 [homes]
        comment = Home Directories
        valid users = %S
        read only = No
        browseable = No
 [office]
        comment = Data
        path = /data
        read only = No
 [netlogon]
        comment = NETLOGON
        path = /var/lib/samba/netlogon
        read only = Yes
        browseable = No
 [printers]
        comment = All Printers
        path = /var/spool/samba
        printable = Yes
        use client driver = Yes
        browseable = No
```

Example 34.1.5 SAURON smb-sauron.conf File Share Section

```
# Global parameters
 [global]
        workgroup = MIDEARTH
        netbios name = SAURON
 [www]
        comment = Web Pages
        path = /srv/www/htdocs
        read only = No
```

Part IV

Migration and Updating

Chapter 35

UPDATING AND UPGRADING SAMBA

This chapter provides a detailed record of changes made during the 3.x series releases. At this time this series consists of the 3.0.x series that is under the GNU GPL version 2 license, and the Samba 3.2.x series that is being released under the terms of the GNU GPL version 3 license.

35.1 Key Update Requirements

Samba is a fluid product in which there may be significant changes between releases. Some of these changes are brought about as a result of changes in the protocols that are used by Microsoft Windows network clients as a result of security or functionality updates through official Microsoft patches and updates. Samba must track such changes, particularly where they affect the internal operation of Samba itself.

Please refer to any notes below that make explicit mention of the version of Samba you are using. In general, all changes that apply to a new release will apply to follow-on releases also. For example, changes to Samba 3.0.23 affect all releases up to an including 3.0.25 and later. Samba 3.2.x was originaly cut from Samba 3.0.25 before 3.2.0-specific changes were applied. Unless a 3.0.x series feature is specifically revoked, the behavior of the 3.2.x series can be expected to follow the earlier pattern.

35.1.1 Upgrading from Samba-3.0.x to Samba-3.2.0

35.1.2 Upgrading from Samba-2.x to Samba-3.0.25

This chapter deals exclusively with the differences between Samba-3.0.25 and Samba-2.2.8a. It points out where configuration parameters have changed, and provides a simple guide for the move from 2.2.x to 3.0.25.

35.1.3 Quick Migration Guide

Samba-3.0.25 default behavior should be approximately the same as Samba-2.2.x. The default behavior when the new parameter *passdb backend* is not defined in the `smb.conf` file provides the same default behavior as Samba-2.2.x with *encrypt passwords* = Yes and will use the `smbpasswd` database.

So why say that *behavior should be approximately the same as Samba-2.2.x*? Because Samba-3.0.25 can negotiate new protocols, such as support for native Unicode, that may result in differing protocol code paths being taken. The new behavior under such circumstances is not exactly the same as the old one. The good news is that the domain and machine SIDs will be preserved across the upgrade.

If the Samba-2.2.x system is using an LDAP backend, and there is no time to update the LDAP database, then make sure that *passdb backend* = ldapsam_compat is specified in the `smb.conf` file. For the rest, behavior should remain more or less the same. At a later date, when there is time to implement a new Samba-3-compatible LDAP backend, it is possible to migrate the old LDAP database to the new one through use of the **pdbedit**. See Section 11.3.2.

35.2 New Featuers in Samba-3.x Series

35.2.1 New Features in Samba-3.2.x Series

35.2.2 New Features in Samba-3.0.x

The major new features are:

1 Active Directory support. This release is able to join an ADS realm as a member server and authenticate users using LDAP/Kerberos.

2 Unicode support. Samba will now negotiate Unicode on the wire, and internally there is a much better infrastructure for multibyte and Unicode character sets.

3 New authentication system. The internal authentication system has been almost completely rewritten. Most of the changes are internal, but the new authoring system is also very configurable.

4 New filename mangling system. The filename mangling system has been completely rewritten. An internal database now stores mangling maps persistently.

5 New "net" command. A new "net" command has been added. It is somewhat similar to the "net" command in Windows. Eventually, we plan to replace a bunch of other utilities (such as smbpasswd) with subcommands in "net".

6 Samba now negotiates NT-style status32 codes on the wire. This considerably improves error handling.

7 Better Windows 200x/XP printing support, including publishing printer attributes in Active Directory.

8 New loadable RPC modules for passdb backends and character sets.

9 New default dual-daemon winbindd support for better performance.

10 Support for migrating from a Windows NT 4.0 domain to a Samba domain and maintaining user, group, and domain SIDs.

11 Support for establishing trust relationships with Windows NT 4.0 domain controllers.

12 Initial support for a distributed Winbind architecture using an LDAP directory for storing SID to UID/GID mappings.

13 Major updates to the Samba documentation tree.

14 Full support for client and server SMB signing to ensure compatibility with default Windows 2003 security settings.

Plus lots of other improvements!

35.2.2.1 Configuration Parameter Changes

This section contains a brief listing of changes to `smb.conf` options since the Samba-2.2.x series up to and including Samba-3.0.25.

Please refer to the smb.conf(5) man page for complete descriptions of new or modified parameters.

Whenever a Samba update or upgrade is performed it is highly recommended to read the file called *WHATSNEW.txt* that is part of the Samba distribution tarball. This file may also be obtain on-line from the Samba web site[1], in the right column, under Current Stable Release, by clicking on *Release Notes*.

35.2.2.2 Removed Parameters

In alphabetical order, these are the parameters eliminated from Samba-2.2.x through 3.0.25.

- admin log
- alternate permissions
- character set
- client codepage
- code page directory
- coding system
- domain admin group
- domain guest group
- enable rid algorithm
- enable svcctl
- force unknown acl user
- hosts equiv
- ldap filter
- min password length

[1] <http://www.samba.org/samba/>

- nt smb support
- post script
- printer admin
- printer driver
- printer driver file
- printer driver location
- read size
- source environment
- status
- strip dot
- total print jobs
- unicode
- use rhosts
- valid chars
- vfs options
- winbind enable local accounts
- winbind max idle children
- wins partners

35.2.2.3 New Parameters

The following new parameters have been released up to and including Samba 3.0.25 (grouped by function:)

Remote Management

- abort shutdown script
- shutdown script

User and Group Account Management

- add group script

- add machine script
- add user to group script
- algorithmic rid base
- delete group script
- delete user from group script
- passdb backend
- rename user script
- set primary group script
- username map script

Authentication

- auth methods
- ldap password sync
- passdb expand explicit
- realm

Protocol Options

- add port command
- afs token lifetime
- client lanman auth
- client NTLMv2 auth
- client schannel
- client signing
- client use spnego
- defer sharing violations
- disable netbios
- dmapi support
- enable privileges
- use kerberos keytab

- log nt token command
- ntlm auth
- paranoid server security
- sendfile
- server schannel
- server signing
- smb ports
- svcctl list
- use spnego

File Service

- allocation roundup size
- acl check permissions
- acl group control
- acl map full control
- aio read size
- aio write size
- dfree cache time
- dfree command
- ea support
- enable asu support
- fam change notify
- force unknown acl user
- get quota command
- hide special files
- hide unwriteable files
- inherit owner
- hostname lookups

- kernel change notify
- mangle prefix
- map acl inherit
- map read only
- max stat cache size
- msdfs proxy
- open files database hash size
- set quota command
- store dos attributes
- use sendfile
- usershare allow guests
- usershare max shares
- usershare owner only
- usershare path
- usershare prefix allow list
- usershare prefix deny list
- usershare template share
- vfs objects

Printing

- cups options
- cups server
- force printername
- iprint server
- max reported print jobs
- printcap cache time

Unicode and Character Sets

- display charset

- dos charset
- UNIX charset

SID to UID/GID Mappings

- idmap backend
- idmap gid
- idmap uid
- username map script
- winbind nss info
- winbind offline logon
- winbind refresh tickets
- winbind trusted domains only
- template primary group

LDAP

- ldap delete dn
- ldap group suffix
- ldap idmap suffix
- ldap machine suffix
- ldap passwd sync
- ldap replication sleep
- ldap timeout
- ldap user suffix

General Configuration

- eventlog list
- preload modules
- reset on zero vc
- privatedir

35.2.2.4 Modified Parameters (Changes in Behavior)

- acl group control (new default is No, deprecated parameter)
- change notify timeout (scope changed)
- dos filemode (disabled by default)
- dos filetimes (enabled by default)
- enable asu support (disabled by default)
- enable privileges (enabled by default)
- encrypt passwords (enabled by default)
- host msdfs (enabled by default)
- mangling method (set to hash2 by default)
- map to guest
- only user (deprecated)
- passwd chat
- passwd program
- password server
- restrict anonymous (integer value)
- security (new ads value)
- strict locking (auto by default)
- winbind cache time (increased to 5 minutes)
- winbind enum groups (disabled by default)
- winbind enum users (disabled by default)
- winbind nested groups (enabled by default)
- winbind uid (deprecated in favor of idmap uid)
- winbind gid (deprecated in favor of idmap gid)
- winbindd nss info
- write cache (deprecated)

35.2.3 New Functionality

The major changes in behavior since that Samba-2.2.x series are documented in this section. Please refer to the `WHATSNEW.txt` file that ships with every release of Samba to obtain detailed information regarding the changes that have been made during the life of the current Samba release.

35.2.3.1 TDB Data Files

Refer to Chapter 1, "How to Install and Test SAMBA", Section 1.2.2 for information pertaining to the Samba-3 data files, their location and the information that must be preserved across server migrations, updates and upgrades.

Please remember to back up your existing ${lock directory}/*tdb before upgrading to Samba-3. If necessary, Samba will upgrade databases as they are opened. Downgrading from Samba-3 to 2.2, or reversion to an earlier version of Samba-3 from a later release, is an unsupported path.

The old Samba-2.2.x tdb files are described in Table 35.1.

35.2.3.2 Changes in Behavior

The following issues are known changes in behavior between Samba-2.2 and Samba-3 that may affect certain installations of Samba.

1. When operating as a member of a Windows domain, Samba-2.2 would map any users authenticated by the remote DC to the "guest account" if a UID could not be obtained via the getpwnam() call. Samba-3 rejects the connection with the error message "NT_STATUS_LOGON_FAILURE." There is no current workaround to re-establish the Samba-2.2 behavior.

2. When adding machines to a Samba-2.2 controlled domain, the "add user script" was used to create the UNIX identity of the machine trust account. Samba-3 introduces a new "add machine script" that must be specified for this purpose. Samba-3 will not fall back to using the "add user script" in the absence of an "add machine script".

Table 35.1 Samba-2.2.x TDB File Descriptions

Name	Description	Backup?
account_policy	User policy settings	yes
brlock	Byte-range file locking information.	no
connections	Client connection information	no
locking	Temporary file locking data.	no
messages	Temporary storage of messages being processed by smbd.	no
ntdrivers	Stores per-printer driver information.	yes
ntforms	Stores per-printer forms information.	yes
ntprinters	Stores the per-printer devmode configuration settings.	yes
printing/*.tdb	Cached output from lpq command created on a per-print-service basis.	no
registry	Read-only Samba registry skeleton that provides support for exporting various database tables via the winreg RPCs.	no
sessionid	Temporary cache for miscellaneous session information.	no
share_info	Share ACL settings.	yes
unexpected	Packets received for which no process was listening.	no
winbindd_cache	Cache of identity information received from an NT4 or an ADS domain.	yes
winbindd_idmap	New ID map table from SIDS to UNIX UIDs/GIDs.	yes

35.2.3.3 Passdb Backends and Authentication

There have been a few new changes that Samba administrators should be aware of when moving to Samba-3.

1. Encrypted passwords have been enabled by default in order to interoperate better with out-of-the-box Windows client installations. This does mean that either (a) a Samba account must be created for each user, or (b) "encrypt passwords = no" must be explicitly defined in `smb.conf`.

2. Inclusion of new *security* = ads option for integration with an Active

Directory domain using the native Windows Kerberos 5 and LDAP protocols.

Samba-3 also includes the possibility of setting up chains of authentication methods (*auth methods*) and account storage backends (*passdb backend*). Please refer to the `smb.conf` man page and Chapter 11, "Account Information Databases", for details. While both parameters assume sane default values, it is likely that you will need to understand what the values actually mean in order to ensure Samba operates correctly.

Certain functions of the **smbpasswd** tool have been split between the new **smbpasswd** utility, the **net** tool, and the new **pdbedit** utility. See the respective man pages for details.

35.2.3.4 LDAP

This section outlines the new features effecting Samba/LDAP integration.

New Schema A new object class (sambaSamAccount) has been introduced to replace the old sambaAccount. This change aids in the renaming of attributes to prevent clashes with attributes from other vendors. There is a conversion script (examples/LDAP/convertSambaAccount) to modify an LDIF file to the new schema.

Example:

```
$ ldapsearch .... -LLL -b "ou=people,dc=..." > old.ldif
$ convertSambaAccount --sid <DOM SID> --input old.ldif --output new.ldif
```

The <DOM SID> can be obtained by running

```
$ net getlocalsid <DOMAINNAME>
```

on the Samba PDC as root.

Under Samba-2.x the domain SID can be obtained by executing:

```
$ smbpasswd -S <DOMAINNAME>
```

The old `sambaAccount` schema may still be used by specifying the *ldapsam_compat* passdb backend. However, the sambaAccount and associated attributes have been moved to the historical section of the schema file and must be uncommented before use if needed. The Samba-2.2 object class declaration for a `sambaAccount` has not changed in the Samba-3 `samba.schema` file.

Other new object classes and their uses include:

- `sambaDomain` — domain information used to allocate RIDs for users and groups as necessary. The attributes are added in "ldap suffix" directory entry automatically if an idmap UID/GID range has been set and the "ldapsam" passdb backend has been selected.
- sambaGroupMapping — an object representing the relationship between a posixGroup and a Windows group/SID. These entries are stored in the "ldap group suffix" and managed by the "net groupmap" command.
- `sambaUNIXIdPool` — created in the "ldap idmap suffix" entry automatically and contains the next available "idmap UID" and "idmap GID".
- `sambaIdmapEntry` — object storing a mapping between a SID and a UNIX UID/GID. These objects are created by the idmap_ldap module as needed.

New Suffix for Searching The following new `smb.conf` parameters have been added to aid in directing certain LDAP queries when *passdb backend = ldapsam://...* has been specified.

- ldap suffix — used to search for user and computer accounts.
- ldap user suffix — used to store user accounts.
- ldap machine suffix — used to store machine trust accounts.
- ldap group suffix — location of posixGroup/sambaGroupMapping entries.
- ldap idmap suffix — location of sambaIdmapEntry objects.

If an *ldap suffix* is defined, it will be appended to all of the remaining subsuffix parameters. In this case, the order of the suffix listings in `smb.conf` is important. Always place the *ldap suffix* first in the list.

Due to a limitation in Samba's `smb.conf` parsing, you should not surround the domain names with quotation marks.

IdMap LDAP Support Samba-3 supports an LDAP backend for the idmap subsystem. The following options inform Samba that the idmap table should be stored on the directory server *onterose* in the ou=Idmap,dc=quenya,dc=org partition.

```
[global]
 ...     idmap backend = ldap:ldap://onterose/
         ldap idmap suffix = ou=Idmap
         idmap uid = 40000-50000
         idmap gid = 40000-50000
```

This configuration allows Winbind installations on multiple servers to share a UID/GID number space, thus avoiding the interoperability problems with NFS that were present in Samba-2.2.

Chapter 36

MIGRATION FROM NT4 PDC TO SAMBA-3 PDC

This is a rough guide to assist those wishing to migrate from NT4 domain control to Samba-3-based domain control.

36.1 Planning and Getting Started

In the IT world there is often a saying that all problems are encountered because of poor planning. The corollary to this saying is that not all problems can be anticipated and planned for. Then again, good planning will anticipate most show-stopper-type situations.

Those wishing to migrate from MS Windows NT4 domain control to a Samba-3 domain control environment would do well to develop a detailed migration plan. So here are a few pointers to help migration get underway.

36.1.1 Objectives

The key objective for most organizations is to make the migration from MS Windows NT4 to Samba-3 domain control as painless as possible. One of the challenges you may experience in your migration process may well be convincing management that the new environment should remain in place. Many who have introduced open source technologies have experienced pressure to return to a Microsoft-based platform solution at the first sign of trouble.

Before attempting a migration to a Samba-3-controlled network, make every possible effort to gain all-round commitment to the change. Know precisely *why* the change is important for the organization. Possible motivations to make a change include:

- Improve network manageability.
- Obtain better user-level functionality.
- Reduce network operating costs.
- Reduce exposure caused by Microsoft withdrawal of NT4 support.
- Avoid MS License 6 implications.
- Reduce organization's dependency on Microsoft.

Make sure everyone knows that Samba-3 is not MS Windows NT4. Samba-3 offers an alternative solution that is both different from MS Windows NT4 and offers advantages compared with it. Gain recognition that Samba-3 lacks many of the features that Microsoft has promoted as core values in migration from MS Windows NT4 to MS Windows 2000 and beyond (with or without Active Directory services).

What are the features that Samba-3 cannot provide?

- Active Directory Server.
- Group Policy Objects (in Active Directory).
- Machine Policy Objects.
- Logon Scripts in Active Directory.
- Software Application and Access Controls in Active Directory.

The features that Samba-3 does provide and that may be of compelling interest to your site include:

- Lower cost of ownership.
- Global availability of support with no strings attached.
- Dynamic SMB servers (can run more than one SMB/CIFS server per UNIX/Linux system).
- Creation of on-the-fly logon scripts.
- Creation of on-the-fly policy files.

- Greater stability, reliability, performance, and availability.
- Manageability via an SSH connection.
- Flexible choices of backend authentication technologies (tdbsam, ldapsam).
- Ability to implement a full single-sign-on architecture.
- Ability to distribute authentication systems for absolute minimum wide-area network bandwidth demand.

Before migrating a network from MS Windows NT4 to Samba-3, consider all necessary factors. Users should be educated about changes they may experience so the change will be a welcome one and not become an obstacle to the work they need to do. The following sections explain factors that will help ensure a successful migration.

36.1.1.1 Domain Layout

Samba-3 can be configured as a domain controller, a backup domain controller (probably best called a secondary controller), a domain member, or a standalone server. The Windows network security domain context should be sized and scoped before implementation. Particular attention needs to be paid to the location of the Primary Domain Controller (PDC) as well as backup controllers (BDCs). One way in which Samba-3 differs from Microsoft technology is that if one chooses to use an LDAP authentication backend, then the same database can be used by several different domains. In a complex organization, there can be a single LDAP database, which itself can be distributed (have a master server and multiple slave servers) that can simultaneously serve multiple domains.

From a design perspective, the number of users per server as well as the number of servers per domain should be scaled taking into consideration server capacity and network bandwidth.

A physical network segment may house several domains. Each may span multiple network segments. Where domains span routed network segments, consider and test the performance implications of the design and layout of a network. A centrally located domain controller that is designed to serve multiple routed network segments may result in severe performance problems. Check the response time (ping timing) between the remote segment

and the PDC. If it's long (more than 100 ms), locate a BDC on the remote segment to serve as the local authentication and access control server.

36.1.1.2 Server Share and Directory Layout

There are cardinal rules to effective network design that cannot be broken with impunity. The most important rule: Simplicity is king in every well-controlled network. Every part of the infrastructure must be managed; the more complex it is, the greater will be the demand of keeping systems secure and functional.

Keep in mind the nature of how data must be shared. Physical disk space layout should be considered carefully. Some data must be backed up. The simpler the disk layout, the easier it will be to keep track of backup needs. Identify what backup media will meet your needs; consider backup to tape, CD-ROM or DVD-ROM, or other offline storage medium. Plan and implement for minimum maintenance. Leave nothing to chance in your design; above all, do not leave backups to chance: backup, test, and validate every backup; create a disaster recovery plan and prove that it works.

Users should be grouped according to data access control needs. File and directory access is best controlled via group permissions, and the use of the "sticky bit" on group-controlled directories may substantially avoid file access complaints from Samba share users.

Inexperienced network administrators often attempt elaborate techniques to set access controls on files, directories, shares, as well as in share definitions. Keep your design and implementation simple and document your design extensively. Have others audit your documentation. Do not create a complex mess that your successor will not understand. Remember, job security through complex design and implementation may cause loss of operations and downtime to users as the new administrator learns to untangle your knots. Keep access controls simple and effective, and make sure that users will never be interrupted by obtuse complexity.

36.1.1.3 Logon Scripts

Logon scripts can help to ensure that all users gain the share and printer connections they need.

Logon scripts can be created on the fly so all commands executed are specific to the rights and privileges granted to the user. The preferred controls should be effected through group membership so group information can be used to create a custom logon script using the *root preexec* parameters to the *NETLOGON* share.

Some sites prefer to use a tool such as **kixstart** to establish a controlled user environment. In any case, you may wish to do a Google search for logon script process controls. In particular, you may wish to explore the use of the Microsoft Knowledge Base article KB189105 that deals with how to add printers without user intervention via the logon script process.

36.1.1.4 Profile Migration/Creation

User and group profiles may be migrated using the tools described in the section titled Desktop Profile Management.

Profiles may also be managed using the Samba-3 tool **profiles**. This tool allows the MS Windows NT-style security identifiers (SIDs) that are stored inside the profile `NTuser.DAT` file to be changed to the SID of the Samba-3 domain.

36.1.1.5 User and Group Accounts

It is possible to migrate all account settings from an MS Windows NT4 domain to Samba-3. Before attempting to migrate user and group accounts, you are STRONGLY advised to create in Samba-3 the groups that are present on the MS Windows NT4 domain *AND* to map them to suitable UNIX/Linux groups. By following this simple advice, all user and group attributes should migrate painlessly.

36.1.2 Steps in Migration Process

The approximate migration process is described below.

- You have an NT4 PDC that has the users, groups, policies, and profiles to be migrated.
- Samba-3 is set up as a domain controller with netlogon share, profile share, and so on. Configure the `smb.conf` file to function as a BDC: *`domain master = No`*.

The Account Migration Process

1. Create a BDC account in the old NT4 domain for the Samba server using NT Server Manager. *Samba must not be running.*

2. `net rpc join -S` *NT4PDC* `-w` *DOMNAME* `-U Administrator%`*passwd*

3. `net rpc vampire -S` *NT4PDC* `-U administrator%`*passwd*

4. `pdbedit -L` Note: Did the users migrate?

5. Now assign each of the UNIX groups to NT groups: (It may be useful to copy this text to a script called `initGroups.sh`)

```
#!/bin/bash
#### Keep this as a shell script for future re-use

# First assign well known domain global groups
net groupmap add ntgroup="Domain Admins" unixgroup=root rid=512 type=d
net groupmap add ntgroup="Domain Users"  unixgroup=users rid=513 type=d
net groupmap add ntgroup="Domain Guests" unixgroup=nobody rid=514 type=d

# Now for our added domain global groups
net groupmap add ntgroup="Designers" unixgroup=designers type=d
net groupmap add ntgroup="Engineers" unixgroup=engineers type=d
net groupmap add ntgroup="QA Team"   unixgroup=qateam    type=d
```

6. `net groupmap list` Check that all groups are recognized.

Migrate all the profiles, then migrate all policy files.

36.2 Migration Options

Sites that wish to migrate from MS Windows NT4 domain control to a Samba-based solution generally fit into three basic categories. Table 36.1 shows the possibilities.

Table 36.1 The Three Major Site Types

Number of Users	Description
< 50	Want simple conversion with no pain.
50 - 250	Want new features; can manage some inhouse complexity.
> 250	Solution/implementation must scale well; complex needs. Cross-departmental decision process. Local expertise in most areas.

36.2.1 Planning for Success

There are three basic choices for sites that intend to migrate from MS Windows NT4 to Samba-3:

- Simple conversion (total replacement).
- Upgraded conversion (could be one of integration).
- Complete redesign (completely new solution).

Minimize downstream problems by:

- Taking sufficient time.
- Avoiding panic.
- Testing all assumptions.
- Testing the full roll-out program, including workstation deployment.

Table 36.2 lists the conversion choices given the type of migration being contemplated.

36.2.2 Samba-3 Implementation Choices

Authentication Database/Backend Samba-3 can use an external authentication backend:

- Winbind (external Samba or NT4/200x server).
- External server could use Active Directory or NT4 domain.
- Can use pam_mkhomedir.so to autocreate home directories.

Table 36.2 Nature of the Conversion Choices

Simple Install	Upgrade Decisions	Redesign Decisions
Make use of minimal OS-specific features	Translate NT4 features to new host OS features	Improve on NT4 functionality, enhance management capabilities
Move all accounts from NT4 into Samba-3	Copy and improve	Authentication regime (database location and access)
Make least number of operational changes	Make progressive improvements	Desktop management methods
Take least amount of time to migrate	Minimize user impact	Better control of Desktops/Users
Live versus isolated conversion	Maximize functionality	Identify Needs for: *Manageability, Scalability, Security, Availability*
Integrate Samba-3, then migrate while users are active, then change of control (swap out)	Take advantage of lower maintenance opportunity	

- Samba-3 can use a local authentication backend: *smbpasswd*, *tdbsam*, *ldapsam*

Access Control Points Samba permits Access Control points to be set:

- On the share itself — using share ACLs.
- On the file system — using UNIX permissions on files and directories.

  Note: Can enable Posix ACLs in file system also.
- Through Samba share parameters — not recommended except as last resort.

Policies (migrate or create new ones) Exercise great caution when making registry changes; use the right tool and be aware that changes made

through NT4-style `NTConfig.POL` files can leave permanent changes.

- Using Group Policy Editor (NT4).
- Watch out for tattoo effect.

User and Group Profiles Platform-specific, so use platform tool to change from a local to a roaming profile. Can use new profiles tool to change SIDs (`NTUser.DAT`).

Logon Scripts Know how they work.

User and Group Mapping to UNIX/Linux User and group mapping code is new. Many problems have been experienced as network administrators who are familiar with Samba-2.2.x migrate to Samba-3. Carefully study the chapters that document the new password backend behavior and the new group mapping functionality.

- The *`username map`* facility may be needed.
- Use **net groupmap** to connect NT4 groups to UNIX groups.
- Use **pdbedit** to set/change user configuration.

  When migrating to LDAP backend, it may be easier to dump the initial LDAP database to LDIF, edit, then reload into LDAP.

OS-Specific Scripts/Programs May be Needed Every operating system has its peculiarities. These are the result of engineering decisions that were based on the experience of the designer and may have side effects that were not anticipated. Limitations that may bite the Windows network administrator include:

- Add/Delete Users: Note OS limits on size of name (Linux 8 chars, NT4 up to 254 chars).
- Add/Delete Machines: Applied only to domain members (Note: machine names may be limited to 16 characters).
- Use **net groupmap** to connect NT4 groups to UNIX groups.

- Add/Delete Groups: Note OS limits on size and nature. Linux limit is 16 char, no spaces, and no uppercase chars (**groupadd**).

Migration Tools Domain Control (NT4-Style) Profiles, Policies, Access Controls, Security

- Samba: **net, rpcclient, smbpasswd, pdbedit, profiles**
- Windows: **NT4 Domain User Manager, Server Manager (NEXUS)**

Chapter 37

SWAT: THE SAMBA WEB ADMINISTRATION TOOL

There are many and varied opinions regarding the usefulness of SWAT. No matter how hard one tries to produce the perfect configuration tool, it remains an object of personal taste. SWAT is a tool that allows Web-based configuration of Samba. It has a wizard that may help to get Samba configured quickly, it has context-sensitive help on each `smb.conf` parameter, it provides for monitoring of current state of connection information, and it allows networkwide MS Windows network password management.

37.1 Features and Benefits

SWAT is a facility that is part of the Samba suite. The main executable is called **swat** and is invoked by the internetworking super daemon. See Section 37.2.2 for details.

SWAT uses integral Samba components to locate parameters supported by the particular version of Samba. Unlike tools and utilities that are external to Samba, SWAT is always up to date as known Samba parameters change. SWAT provides context-sensitive help for each configuration parameter, directly from **man** page entries.

Some network administrators believe that it is a good idea to write systems documentation inside configuration files, and for them SWAT will always be a nasty tool. SWAT does not store the configuration file in any intermediate form; rather, it stores only the parameter settings, so when SWAT writes the `smb.conf` file to disk, it writes only those parameters that are at other

than the default settings. The result is that all comments, as well as parameters that are no longer supported, will be lost from the `smb.conf` file. Additionally, the parameters will be written back in internal ordering.

NOTE

Before using SWAT, please be warned — SWAT will completely replace your `smb.conf` with a fully optimized file that has been stripped of all comments you might have placed there and only nondefault settings will be written to the file.

37.2 Guidelines and Technical Tips

This section aims to unlock the dark secrets behind how SWAT may be made to work, how it can be made more secure, and how to solve internationalization support problems.

37.2.1 Validate SWAT Installation

The very first step that should be taken before attempting to configure a host system for SWAT operation is to check that it is installed. This may seem a trivial point to some, but several Linux distributions do not install SWAT by default, even though they do ship an installable binary support package containing SWAT on the distribution media.

When you have confirmed that SWAT is installed, it is necessary to validate that the installation includes the binary **swat** file as well as all the supporting text and Web files. A number of operating system distributions in the past have failed to include the necessary support files, even though the **swat** binary executable file was installed.

Finally, when you are sure that SWAT has been fully installed, please check that SWAT is enabled in the control file for the internetworking superdaemon (inetd or xinetd) that is used on your operating system platform.

37.2.1.1 Locating the SWAT File

To validate that SWAT is installed, first locate the **swat** binary file on the system. It may be found under the following directories:

`/usr/local/samba/bin` — the default Samba location
`/usr/sbin` — the default location on most Linux systems
`/opt/samba/bin`

The actual location is much dependent on the choice of the operating system vendor or as determined by the administrator who compiled and installed Samba.

There are a number of methods that may be used to locate the **swat** binary file. The following methods may be helpful.

If **swat** is in your current operating system search path, it will be easy to find it. You can ask what are the command-line options for **swat** as shown here:

```
frodo:~ # swat -?
Usage: swat [OPTION...]
  -a, --disable-authentication         Disable authentication (demo mode)

Help options:
  -?, --help                           Show this help message
  --usage                              Display brief usage message

Common samba options:
  -d, --debuglevel=DEBUGLEVEL          Set debug level
  -s, --configfile=CONFIGFILE          Use alternative configuration file
  -l, --log-basename=LOGFILEBASE       Basename for log/debug files
  -V, --version                        Print version
```

37.2.1.2 Locating the SWAT Support Files

Now that you have found that **swat** is in the search path, it is easy to identify where the file is located. Here is another simple way this may be done:

```
frodo:~ # whereis swat
swat: /usr/sbin/swat /usr/share/man/man8/swat.8.gz
```

If the above measures fail to locate the **swat** binary, another approach is needed. The following may be used:

```
frodo:/ # find / -name swat -print
/etc/xinetd.d/swat
/usr/sbin/swat
/usr/share/samba/swat
frodo:/ #
```

This list shows that there is a control file for **xinetd**, the internetwork superdaemon that is installed on this server. The location of the SWAT binary file is `/usr/sbin/swat`, and the support files for it are located under the directory `/usr/share/samba/swat`.

We must now check where **swat** expects to find its support files. This can be done as follows:

```
frodo:/ # strings /usr/sbin/swat | grep "/swat"
/swat/
...
/usr/share/samba/swat
frodo:/ #
```

The `/usr/share/samba/swat/` entry shown in this listing is the location of the support files. You should verify that the support files exist under this directory. A sample list is as shown:

```
jht@frodo:/> find /usr/share/samba/swat -print
/usr/share/samba/swat
/usr/share/samba/swat/help
/usr/share/samba/swat/lang
/usr/share/samba/swat/lang/ja
/usr/share/samba/swat/lang/ja/help
```

```
/usr/share/samba/swat/lang/ja/help/welcome.html
/usr/share/samba/swat/lang/ja/images
/usr/share/samba/swat/lang/ja/images/home.gif
...
/usr/share/samba/swat/lang/ja/include
/usr/share/samba/swat/lang/ja/include/header.nocss.html
...
/usr/share/samba/swat/lang/tr
/usr/share/samba/swat/lang/tr/help
/usr/share/samba/swat/lang/tr/help/welcome.html
/usr/share/samba/swat/lang/tr/images
/usr/share/samba/swat/lang/tr/images/home.gif
...
/usr/share/samba/swat/lang/tr/include
/usr/share/samba/swat/lang/tr/include/header.html
/usr/share/samba/swat/using_samba
...
/usr/share/samba/swat/images
/usr/share/samba/swat/images/home.gif
...
/usr/share/samba/swat/include
/usr/share/samba/swat/include/footer.html
/usr/share/samba/swat/include/header.html
jht@frodo:/>
```

If the files needed are not available, it is necessary to obtain and install them before SWAT can be used.

37.2.2 Enabling SWAT for Use

SWAT should be installed to run via the network super-daemon. Depending on which system your UNIX/Linux system has, you will have either an **inetd**- or **xinetd**-based system.

The nature and location of the network super-daemon varies with the operating system implementation. The control file (or files) can be located in the file `/etc/inetd.conf` or in the directory `/etc/[x]inet[d].d` or in a similar location.

The control entry for the older style file might be:

```
   # swat is the Samba Web Administration Tool
   swat stream tcp nowait.400 root /usr/sbin/swat swat
```

A control file for the newer style xinetd could be:

```
# default: off
# description: SWAT is the Samba Web Admin Tool. Use swat \
#              to configure your Samba server. To use SWAT, \
#              connect to port 901 with your favorite web browser.
service swat
{
   port    = 901
   socket_type     = stream
   wait    = no
   only_from = localhost
   user    = root
   server  = /usr/sbin/swat
   log_on_failure  += USERID
   disable = no
}
```

In the above, the default setting for *disable* is `yes`. This means that SWAT is disabled. To enable use of SWAT, set this parameter to `no` as shown.

Both of the previous examples assume that the **swat** binary has been located in the `/usr/sbin` directory. In addition to the above, SWAT will use a directory access point from which it will load its Help files as well as other control information. The default location for this on most Linux systems is in the directory `/usr/share/samba/swat`. The default location using Samba defaults will be `/usr/local/samba/swat`.

Access to SWAT will prompt for a logon. If you log onto SWAT as any non-root user, the only permission allowed is to view certain aspects of configuration as well as access to the password change facility. The buttons that will be exposed to the non-root user are **HOME**, **STATUS**, **VIEW**, and **PASSWORD**. The only page that allows change capability in this case is **PASSWORD**.

As long as you log onto SWAT as the user *root*, you should obtain full change and commit ability. The buttons that will be exposed include **HOME**, **GLOBALS**, **SHARES**, **PRINTERS**, **WIZARD**, **STATUS**, **VIEW**, and **PASSWORD**.

37.2.3 Securing SWAT through SSL

Many people have asked about how to set up SWAT with SSL to allow for secure remote administration of Samba. Here is a method that works, courtesy of Markus Krieger.

Modifications to the SWAT setup are as follows:

1. Install OpenSSL.
2. Generate certificate and private key.

```
root# /usr/bin/openssl req -new -x509 -days 365 -nodes -config \
   /usr/share/doc/packages/stunnel/stunnel.cnf \
   -out /etc/stunnel/stunnel.pem -keyout /etc/stunnel/stunnel.pem
```

3. Remove SWAT entry from [x]inetd.
4. Start **stunnel**.

```
root# stunnel -p /etc/stunnel/stunnel.pem -d 901 \
    -l /usr/local/samba/bin/swat swat
```

Afterward, simply connect to SWAT by using the URL <`https://myhost:901`>, accept the certificate, and the SSL connection is up.

37.2.4 Enabling SWAT Internationalization Support

SWAT can be configured to display its messages to match the settings of the language configurations of your Web browser. It will be passed to SWAT in the Accept-Language header of the HTTP request.

To enable this feature:

- Install the proper **msg** files from the Samba `source/po` directory into $LIBDIR.
- Set your browsers language setting.

The name of the **msg** file is the same as the language ID sent by the browser. For example, *en* means English, *ja* means Japanese, *fr* means French.

If you do not like some of messages, or there are no **msg** files for your locale, you can create them simply by copying the **en.msg** files to the directory for "your language ID.msg" and filling in proper strings to each "msgstr". For example, in `it.msg`, the **msg** file for the Italian locale, just set:

```
msgid "Set Default"
msgstr "Imposta Default"
```

and so on. If you find a mistake or create a new **msg** file, please email it to us so we will consider it in the next release of Samba. The **msg** file should be encoded in UTF-8.

Note that if you enable this feature and the *display charset* is not matched to your browser's setting, the SWAT display may be corrupted. In a future version of Samba, SWAT will always display messages with UTF-8 encoding. You will then not need to set this `smb.conf` file parameter.

37.3 Overview and Quick Tour

SWAT is a tool that may be used to configure Samba or just to obtain useful links to important reference materials such as the contents of this book as well as other documents that have been found useful for solving Windows networking problems.

37.3.1 The SWAT Home Page

The SWAT title page provides access to the latest Samba documentation. The manual page for each Samba component is accessible from this page, as are the Samba3-HOWTO (this document) as well as the O'Reilly book "Using Samba."

Administrators who wish to validate their Samba configuration may obtain useful information from the man pages for the diagnostic utilities. These are available from the SWAT home page also. One diagnostic tool that is not mentioned on this page but that is particularly useful is **ethereal**[1].

WARNING

SWAT can be configured to run in *demo* mode. This is not recommended because it runs SWAT without authentication and with full administrative ability. It allows changes to `smb.conf` as well as general operation with root privileges. The option that creates this ability is the `-a` flag to SWAT. *Do not use this in a production environment.*

37.3.2 Global Settings

The **GLOBALS** button exposes a page that allows configuration of the global parameters in `smb.conf`. There are two levels of exposure of the parameters:

- **Basic** — exposes common configuration options.
- **Advanced** — exposes configuration options needed in more complex environments.

To switch to other than **Basic** editing ability, click on **Advanced**. You may also do this by clicking on the radio button, then click on the **Commit Changes** button.

After making any changes to configuration parameters, make sure that you click on the **Commit Changes** button before moving to another area; otherwise, your changes will be lost.

[1] <http://www.ethereal.com/>

NOTE

SWAT has context-sensitive help. To find out what each parameter is for, simply click on the **Help** link to the left of the configuration parameter.

37.3.3 Share Settings

To affect a currently configured share, simply click on the pull-down button between the **Choose Share** and the **Delete Share** buttons and select the share you wish to operate on. To edit the settings, click on the **Choose Share** button. To delete the share, simply press the **Delete Share** button.

To create a new share, next to the button labeled **Create Share**, enter into the text field the name of the share to be created, then click on the **Create Share** button.

37.3.4 Printers Settings

To affect a currently configured printer, simply click on the pull-down button between the **Choose Printer** and the **Delete Printer** buttons and select the printer you wish to operate on. To edit the settings, click on the **Choose Printer** button. To delete the share, simply press the **Delete Printer** button.

To create a new printer, next to the button labeled **Create Printer**, enter into the text field the name of the share to be created, then click on the **Create Printer** button.

37.3.5 The SWAT Wizard

The purpose of the SWAT Wizard is to help the Microsoft-knowledgeable network administrator to configure Samba with a minimum of effort.

The Wizard page provides a tool for rewriting the `smb.conf` file in fully optimized format. This will also happen if you press the **Commit** button. The two differ because the **Rewrite** button ignores any changes that may have been made, while the **Commit** button causes all changes to be affected.

The **Edit** button permits the editing (setting) of the minimal set of options that may be necessary to create a working Samba server.

Finally, there are a limited set of options that determine what type of server Samba will be configured for, whether it will be a WINS server, participate as a WINS client, or operate with no WINS support. By clicking one button, you can elect to expose (or not) user home directories.

37.3.6 The Status Page

The status page serves a limited purpose. First, it allows control of the Samba daemons. The key daemons that create the Samba server environment are smbd, nmbd, and winbindd.

The daemons may be controlled individually or as a total group. Additionally, you may set an automatic screen refresh timing. As MS Windows clients interact with Samba, new smbd processes are continually spawned. The auto-refresh facility allows you to track the changing conditions with minimal effort.

Finally, the status page may be used to terminate specific smbd client connections in order to free files that may be locked.

37.3.7 The View Page

The view page allows you to view the optimized `smb.conf` file and, if you are particularly masochistic, permits you also to see all possible global configuration parameters and their settings.

37.3.8 The Password Change Page

The password change page is a popular tool that allows the creation, deletion, deactivation, and reactivation of MS Windows networking users on the local machine. You can also use this tool to change a local password for a user account.

When logged in as a non-root account, the user must provide the old password as well as the new password (twice). When logged in as *root*, only the new password is required.

One popular use for this tool is to change user passwords across a range of remote MS Windows servers.

Part V

Troubleshooting

Chapter 38

THE SAMBA CHECKLIST

38.1 Introduction

This file contains a list of tests you can perform to validate your Samba server. It also tells you what the likely cause of the problem is if it fails any one of these steps. If it passes all these tests, then it is probably working fine.

You should do all the tests in the order shown. We have tried to carefully choose them so later tests only use capabilities verified in the earlier tests. However, do not stop at the first error: there have been some instances when continuing with the tests has helped to solve a problem.

If you send one of the Samba mailing lists an email saying, "It does not work," and you have not followed this test procedure, you should not be surprised if your email is ignored.

38.2 Assumptions

In all of the tests, it is assumed you have a Samba server called BIGSERVER and a PC called ACLIENT, both in workgroup TESTGROUP.

The procedure is similar for other types of clients.

It is also assumed you know the name of an available share in your `smb.conf`. I for our examples this share is called *tmp*. You can add a *tmp* share like this by adding the lines shown in Example 38.2.1.

Example 38.2.1 smb.conf with [tmp] Share

```
[tmp]
        comment = temporary files
        path = /tmp
        read only = yes
```

NOTE

These tests assume version 3.0.0 or later of the Samba suite. Some commands shown did not exist in earlier versions.

Please pay attention to the error messages you receive. If any error message reports that your server is being unfriendly, you should first check that your IP name resolution is correctly set up. Make sure your `/etc/resolv.conf` file points to name servers that really do exist.

Also, if you do not have DNS server access for name resolution, please check that the settings for your `smb.conf` file results in *`dns proxy = no`*. The best way to check this is with **testparm smb.conf**.

It is helpful to monitor the log files during testing by using the **tail -F log_file_name** in a separate terminal console (use ctrl-alt-F1 through F6 or multiple terminals in X). Relevant log files can be found (for default installations) in `/usr/local/samba/var`. Also, connection logs from machines can be found here or possibly in `/var/log/samba`, depending on how or if you specified logging in your `smb.conf` file.

If you make changes to your `smb.conf` file while going through these test, remember to restart smbd and nmbd.

38.3 The Tests

Diagnosing Your Samba Server

1. In the directory in which you store your `smb.conf` file, run the command **testparm smb.conf**. If it reports any errors, then your `smb.conf` configuration file is faulty.

> NOTE
>
> Your `smb.conf` file may be located in `/etc/samba` or in `/usr/local/samba/lib`.

2. Run the command **ping BIGSERVER** from the PC and **ping ACLIENT** from the UNIX box. If you do not get a valid response, then your TCP/IP software is not correctly installed. You will need to start a "DOS prompt" window on the PC to run ping. If you get a message saying "host not found" or a similar message, then your DNS software or `/etc/hosts` file is not correctly set up. If using DNS, check that the `/etc/resolv.conf` has correct, current, entries in it. It is possible to run Samba without DNS entries for the server and client, but it is assumed you do have correct entries for the remainder of these tests. Another reason why ping might fail is if your host is running firewall software. You will need to relax the rules to let in the workstation in question, perhaps by allowing access from another subnet (on Linux this is done via the appropriate firewall maintenance commands **ipchains** or **iptables**).

> NOTE
>
> Modern Linux distributions install ipchains/iptables by default. This is a common problem that is often overlooked.

If you wish to check what firewall rules may be present in a system under test, simply run **iptables -L -v**, or if *ipchains*-based firewall rules are in use, **ipchains -L -v**. Here is a sample listing from a system that has an external Ethernet interface (eth1) on which Samba is not active and an internal (private network) interface (eth0) on which Samba is active:

```
frodo:~ # iptables -L -v
Chain INPUT (policy DROP 98496 packets, 12M bytes)
 pkts bytes target     prot opt in     out     source     destination
 187K  109M ACCEPT     all  --  lo     any     anywhere   anywhere
 892K  125M ACCEPT     all  --  eth0   any     anywhere   anywhere
1399K 1380M ACCEPT     all  --  eth1   any     anywhere   anywhere  \
               state RELATED,ESTABLISHED

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source     destination
 978K 1177M ACCEPT     all  --  eth1   eth0    anywhere   anywhere \
               state RELATED,ESTABLISHED
 658K   40M ACCEPT     all  --  eth0   eth1    anywhere   anywhere
    0     0 LOG        all  --  any    any     anywhere   anywhere \
               LOG level warning

Chain OUTPUT (policy ACCEPT 2875K packets, 1508M bytes)
 pkts bytes target     prot opt in     out     source     destination

Chain reject_func (0 references)
 pkts bytes target     prot opt in     out     source     destination
```

3. Run the command **smbclient -L BIGSERVER** on the UNIX box. You should get back a list of available shares. If you get an error message containing the string "bad password", then you probably have either an incorrect *`hosts allow`*, *`hosts deny`*, or *`valid users`* line in your `smb.conf`, or your guest account is not valid. Check what your guest account is using testparm and temporarily remove any *`hosts allow`*, *`hosts deny`*, *`valid users`*, or *`invalid users`* lines. If you get a message `connection refused` response, then the **smbd** server may not be running. If you installed it in `inetd.conf`, then you probably edited that file incorrectly. If you installed it as a daemon, then check that it is running and check that the netbios-ssn port is in a LISTEN state using **netstat -a**.

NOTE

Some UNIX/Linux systems use **xinetd** in place of **inetd**. Check your system documentation for the location of the control files for your particular system implementation of the network super daemon.

If you get a message saying `session request failed,` the server refused the connection. If it says "Your server software is being unfriendly," then it's probably because you have invalid command line parameters to smbd, or a similar fatal problem with the initial startup of smbd. Also check your config file (`smb.conf`) for syntax errors with testparm and that the various directories where Samba keeps its log and lock files exist. There are a number of reasons for which smbd may refuse or decline a session request. The most common of these involve one or more of the `smb.conf` file entries as shown in Example 38.3.1.

Example 38.3.1 Configuration for Allowing Connections Only from a Certain Subnet

```
[globals]
        hosts deny = ALL
        hosts allow = xxx.xxx.xxx.xxx/yy
        interfaces = eth0
        bind interfaces only = Yes
```

In Example 38.3.1, no allowance has been made for any session requests that will automatically translate to the loopback adapter address 127.0.0.1. To solve this problem, change these lines as shown in Example 38.3.2.

Another common cause of these two errors is having something already running on port `139`, such as Samba (smbd is running from inetd already) or Digital's Pathworks. Check your `inetd.conf` file before trying to start smbd as a daemon — it can avoid a lot of frustration!

And yet another possible cause for failure of this test is when the subnet mask and/or broadcast address settings are incorrect. Please check that the network interface IP address/broadcast address/subnet

Example 38.3.2 Configuration for Allowing Connections from a Certain Subnet and localhost

```
[globals]
        hosts deny = ALL
        hosts allow = xxx.xxx.xxx.xxx/yy 127.
        interfaces = eth0 lo
```

mask settings are correct and that Samba has correctly noted these in the `log.nmbd` file.

4. Run the command **nmblookup -B BIGSERVER __SAMBA__**. You should get back the IP address of your Samba server. If you do not, then nmbd is incorrectly installed. Check your `inetd.conf` if you run it from there, or that the daemon is running and listening to UDP port 137. One common problem is that many inetd implementations can't take many parameters on the command line. If this is the case, then create a one-line script that contains the right parameters and run that from inetd.

5. Run the command **nmblookup -B ACLIENT '*'**. You should get the PC's IP address back. If you do not, then the client software on the PC isn't installed correctly, or isn't started, or you got the name of the PC wrong. If ACLIENT does not resolve via DNS, then use the IP address of the client in the above test.

6. Run the command **nmblookup -d 2 '*'**. This time we are trying the same as the previous test but are trying it via a broadcast to the default broadcast address. A number of NetBIOS/TCP/IP hosts on the network should respond, although Samba may not catch all of the responses in the short time it listens. You should see the `got a positive name query response` messages from several hosts. If this does not give a result similar to the previous test, then nmblookup isn't correctly getting your broadcast address through its automatic mechanism. In this case you should experiment with the *interfaces* option in `smb.conf` to manually configure your IP address, broadcast, and netmask. If your PC and server aren't on the same subnet, then you will need to use the `-B` option to set the broadcast address to that of the PC's subnet. This test will probably fail if your subnet mask and broadcast address are not correct. (Refer to test 3 notes above).

7. Run the command **smbclient //BIGSERVER/TMP**. You should then be prompted for a password. You should use the password of the account with which you are logged into the UNIX box. If you want to test with another account, then add the `-U accountname` option to the end of the command line — for example, **smbclient //bigserver/tmp -Ujohndoe**.

> NOTE
>
> It is possible to specify the password along with the username as follows: **smbclient //bigserver/tmp -Ujohndoe%secret**.

Once you enter the password, you should get the `smb>` prompt. If you do not, then look at the error message. If it says "invalid network name," then the service *tmp* is not correctly set up in your `smb.conf`. If it says "bad password," then the likely causes are:

(a) You have shadow passwords (or some other password system) but didn't compile in support for them in smbd.

(b) Your *valid users* configuration is incorrect.

(c) You have a mixed-case password and you haven't enabled the *password level* option at a high enough level.

(d) The *path* line in `smb.conf` is incorrect. Check it with testparm.

(e) You enabled password encryption but didn't map UNIX to Samba users. Run **smbpasswd -a username**

Once connected, you should be able to use the commands **dir**, **get**, **put**, and so on. Type **help command** for instructions. You should especially check that the amount of free disk space shown is correct when you type **dir**.

8. On the PC, type the command **net view \\BIGSERVER**. You will need to do this from within a DOS prompt window. You should get back a list of shares available on the server. If you get a message `network name not found` or similar error, then NetBIOS name resolution is not working. This is usually caused by a problem in **nmbd**.

To overcome it, you could do one of the following (you only need to choose one of them):

(a) Fix the nmbd installation.

(b) Add the IP address of BIGSERVER to the **wins server** box in the advanced TCP/IP setup on the PC.

(c) Enable Windows name resolution via DNS in the advanced section of the TCP/IP setup.

(d) Add BIGSERVER to your lmhosts file on the PC.

If you get a message "invalid network name" or "bad password error," then apply the same fixes as for the **smbclient -L** test. In particular, make sure your **hosts allow** line is correct (see the man pages). Also, do not overlook that fact that when the workstation requests the connection to the Samba server, it will attempt to connect using the name with which you logged onto your Windows machine. You need to make sure that an account exists on your Samba server with that exact same name and password. If you get a message "specified computer is not receiving requests" or similar error, it probably means that the host is not contactable via TCP services. Check to see if the host is running TCP wrappers, and if so, add an entry in the `hosts.allow` file for your client (or subnet, and so on.)

9. Run the command **net use x: \\BIGSERVER\TMP**. You should be prompted for a password, then you should get a `command completed successfully` message. If not, then your PC software is incorrectly installed or your `smb.conf` is incorrect. Make sure your *`hosts allow`* and other config lines in `smb.conf` are correct. It's also possible that the server can't work out what username to connect you as. To see if this is the problem, add the line *user* = username to the *[tmp]* section of `smb.conf` where *`username`* is the username corresponding to the password you typed. If you find this fixes things, you may need the username mapping option. It might also be the case that your client only sends encrypted passwords and you have *encrypt passwords* = no in `smb.conf`. Change this setting to 'yes' to fix this.

10. Run the command **nmblookup -M** *`testgroup`* where *`testgroup`* is the name of the workgroup that your Samba server and Windows PCs belong to. You should get back the IP address of the master browser for that workgroup. If you do not, then the election process has failed.

Wait a minute to see if it is just being slow, then try again. If it still fails after that, then look at the browsing options you have set in `smb.conf`. Make sure you have *preferred master* = yes to ensure that an election is held at startup.

11. From file manager, try to browse the server. Your Samba server should appear in the browse list of your local workgroup (or the one you specified in `smb.conf`). You should be able to double-click on the name of the server and get a list of shares. If you get the error message "invalid password," you are probably running Windows NT and it is refusing to browse a server that has no encrypted password capability and is in user-level security mode. In this case, either set *security* = server and *password server* = Windows_NT_Machine in your `smb.conf` file or make sure *encrypt passwords* is set to "yes".

Chapter 39

ANALYZING AND SOLVING SAMBA PROBLEMS

There are many sources of information available in the form of mailing lists, RFCs, and documentation. The documentation that comes with the Samba distribution contains good explanations of general SMB topics such as browsing.

39.1 Diagnostics Tools

With SMB networking, it is often not immediately clear what the cause is of a certain problem. Samba itself provides rather useful information, but in some cases you might have to fall back to using a *sniffer*. A sniffer is a program that listens on your LAN, analyzes the data sent on it, and displays it on the screen.

39.1.1 Debugging with Samba Itself

One of the best diagnostic tools for debugging problems is Samba itself. You can use the `-d option` for both smbd and nmbd to specify the *debug level* at which to run. See the man pages for **smbd, nmbd**, and `smb.conf` for more information regarding debugging options. The debug level (log level) can range from 1 (the default) to 10 (100 for debugging passwords).

Another helpful method of debugging is to compile Samba using the **gcc -g** flag. This will include debug information in the binaries and allow you to attach **gdb** to the running **smbd/nmbd** process. To attach **gdb** to an **smbd** process for an NT workstation, first get the workstation to make

the connection. Pressing ctrl-alt-delete and going down to the domain box is sufficient (at least, the first time you join the domain) to generate a *LsaEnumTrustedDomains*. Thereafter, the workstation maintains an open connection and there will be an smbd process running (assuming that you haven't set a really short smbd idle timeout). So, in between pressing **ctrl-alt-delete** and actually typing in your password, you can attach **gdb** and continue.

Some useful Samba commands worth investigating are:

```
$ testparm | more
$ smbclient -L //{netbios name of server}
```

39.1.2 Tcpdump

Tcpdump[1] was the first UNIX sniffer with SMB support. It is a command-line utility and now, its SMB support is somewhat lagging that of **ethereal** and **tethereal**.

39.1.3 Ethereal

Ethereal[2] is a graphical sniffer, available for both UNIX (Gtk) and Windows. Ethereal's SMB support is quite good. For details on the use of **ethereal**, read the well-written Ethereal User Guide.

Listen for data on ports 137, 138, 139, and 445. For example, use the filter `port 137, port 138, port 139, or port 445` as seen in Figure 39.1 snapshot.

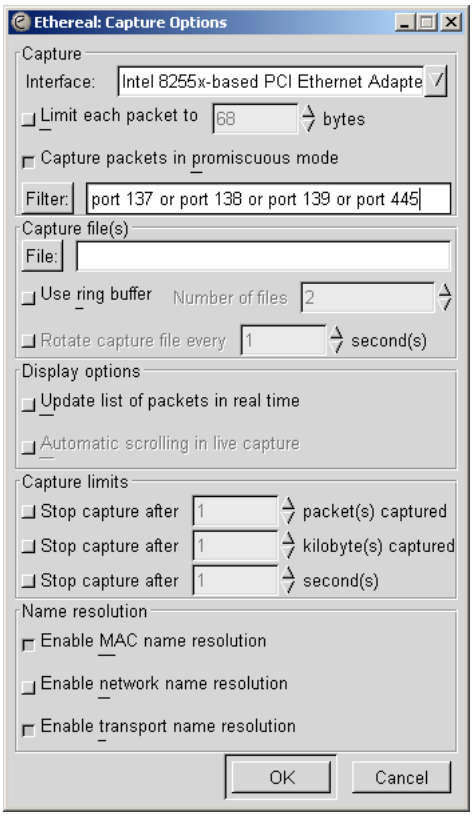
A console version of ethereal is available as well and is called **tethereal**.

39.1.4 The Windows Network Monitor

For tracing things on Microsoft Windows NT, Network Monitor (aka Netmon) is available on Microsoft Developer Network CDs, the Windows NT Server install CD, and the SMS CDs. The version of Netmon that ships with

[1] <http://www.tcpdump.org/>
[2] <http://www.ethereal.com/>

Figure 39.1 Starting a Capture.

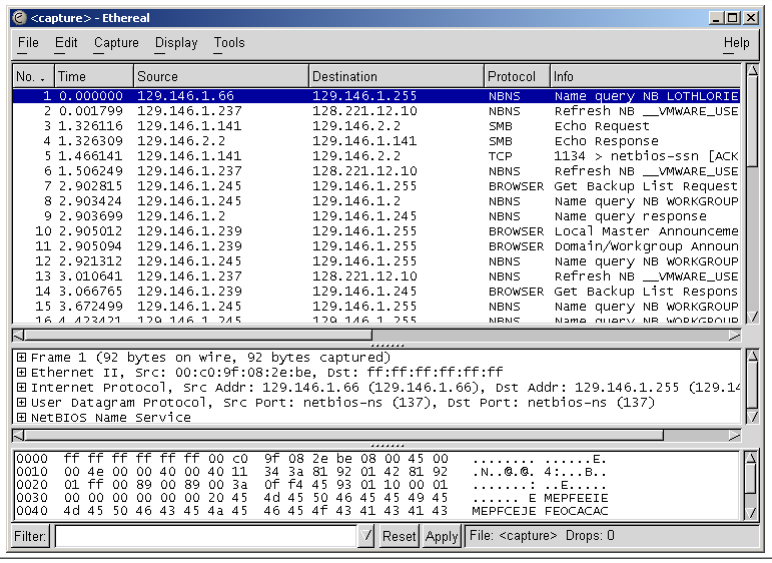


SMS allows for dumping packets between any two computers (i.e., placing the network interface in promiscuous mode). The version on the NT Server install CD will only allow monitoring of network traffic directed to the local NT box and broadcasts on the local subnet. Be aware that Ethereal can read and write Netmon formatted files.

39.1.4.1 Installing Network Monitor on an NT Workstation

Installing Netmon on an NT workstation requires a couple of steps. The following are instructions for installing Netmon V4.00.349, which comes with Microsoft Windows NT Server 4.0, on Microsoft Windows NT Workstation 4.0. The process should be similar for other versions of Windows NT version of Netmon. You will need both the Microsoft Windows NT Server 4.0 Install CD and the Workstation 4.0 Install CD.

Figure 39.2 Main Ethereal Data Window.



Initially you will need to install Network Monitor Tools and Agent on the NT Server to do this:

- Go to **Start** -> **Settings** -> **Control Panel** -> **Network** -> **Services** -> **Add**.
- Select the **Network Monitor Tools and Agent** and click on **OK**.
- Click on **OK** on the Network Control Panel.
- Insert the Windows NT Server 4.0 install CD when prompted.

At this point, the Netmon files should exist in `%SYSTEMROOT%\System32\netmon\*.*`. Two subdirectories exist as well: `parsers\`, which contains the necessary DLLs for parsing the Netmon packet dump, and `captures\`.

To install the Netmon tools on an NT Workstation, you will first need to install the Network Monitor Agent from the Workstation install CD.

- Go to **Start** -> **Settings** -> **Control Panel** -> **Network** -> **Services** -> **Add**.
- Select the **Network Monitor Agent**, click on **OK**.
- Click on **OK** in the Network Control Panel.

- Insert the Windows NT Workstation 4.0 install CD when prompted.

Now copy the files from the NT Server in `%SYSTEMROOT%\System32\netmon` to `%SYSTEMROOT%\System32\netmon` on the workstation and set permissions as you deem appropriate for your site. You will need administrative rights on the NT box to run Netmon.

39.1.4.2 Installing Network Monitor on Windows 9x/Me

To install Netmon on Windows 9x/Me, install the Network Monitor Agent from the Windows 9x/Me CD (`\admin\nettools\netmon`). There is a readme file included with the Netmon driver files on the CD if you need information on how to do this. Copy the files from a working Netmon installation.

39.2 Useful URLs

.

- See how Scott Merrill simulates a BDC behavior at http://www.skippy.net/linux/smb-howto.html[3].
- FTP site for older SMB specs, ftp://ftp.microsoft.com/developr/drg/CIFS/[4]

39.3 Getting Mailing List Help

There are a number of Samba-related mailing lists. Go to <`http://samba.org`>, click on your nearest mirror, and then click on **Support**. Next, click on **Samba-related mailing lists**.

For questions relating to Samba TNG, go to <`http://www.samba-tng.org/`>. It has been requested that you do not post questions about Samba-TNG to the mainstream Samba lists.

If you do post a message to one of the lists, please observe the following guidelines:

[3] <`http://www.skippy.net/linux/smb-howto.html`>
[4] <`ftp://ftp.microsoft.com/developr/drg/CIFS/`>

- Always remember that the developers are volunteers; they are not paid and they never guarantee to produce a particular feature at a particular time. Any timelines are "best guess," and nothing more.

- Always mention what version of Samba you are using and what operating system it's running under. You should list the relevant sections of your `smb.conf` file, at least the options in *[global]* that affect PDC support.

- In addition to the version, if you obtained Samba via CVS, mention the date when you last checked it out.

- Try to make your questions clear and brief. Lots of long, convoluted questions get deleted before they are completely read! Do not post HTML-encoded messages. Most people on mailing lists simply delete them.

- If you run one of those nifty "I'm on holiday" things when you are away, make sure its configured to not answer mailing list traffic. Autoresponses to mailing lists really irritate the thousands of people who end up having to deal with such bad netiquet bahavior.

- Don't cross post. Work out which is the best list to post to and see what happens. Do not post to both samba-ntdom and samba-technical. Many people active on the lists subscribe to more than one list and get annoyed to see the same message two or more times. Often someone who thinks a message would be better dealt with on another list will forward it on for you.

- You might include *partial* log files written at a log level set to as much as 20. Please do not send the entire log but just enough to give the context of the error messages.

- If you have a complete Netmon trace (from the opening of the pipe to the error), you can send the *.CAP file as well.

- Please think carefully before attaching a document to an email. Consider pasting the relevant parts into the body of the message. The Samba mailing lists go to a huge number of people. Do they all need a copy of your `smb.conf` in their attach directory?

39.4 How to Get Off the Mailing Lists

To have your name removed from a Samba mailing list, go to the same place where you went to subscribe to it, go to http://lists.samba.org[5], click on your nearest mirror, click on **Support**, and then click on **Samba-related mailing lists**.

Please do not post messages to the list asking to be removed. You will only be referred to the above address (unless that process failed in some way).

[5] <http://lists.samba.org/>

Chapter 40

REPORTING BUGS

40.1 Introduction

Please report bugs using Samba's Bugzilla[1] facilities and take the time to read this file before you submit a bug report. Also, check to see if it has changed between releases, as we may be changing the bug reporting mechanism at some point.

Please do as much as you can yourself to help track down the bug. Samba is maintained by a dedicated group of people who volunteer their time, skills, and efforts. We receive far more mail than we can possibly answer, so you have a much higher chance of a response and a fix if you send us a "developer-friendly" bug report that lets us fix it fast.

If you post the bug to the comp.protocols.smb newsgroup or the mailing list, do not assume that we will read it. If you suspect that your problem is not a bug but a configuration problem, it is better to send it to the Samba mailing list, as there are thousands of other users on that list who may be able to help you.

You may also like to look though the recent mailing list archives, which are conveniently accessible on the Samba Web pages at <`http://samba.org/samba/`>.

40.2 General Information

Before submitting a bug report, check your config for silly errors. Look in your log files for obvious messages that tell you've misconfigured something.

[1] <`https://bugzilla.samba.org/`>

Run testparm to check your config file for correct syntax.

Have you looked through Chapter 38, "The Samba Checklist"? This is extremely important.

If you include part of a log file with your bug report, then be sure to annotate it with exactly what you were doing on the client at the time and exactly what the results were.

40.3 Debug Levels

If the bug has anything to do with Samba behaving incorrectly as a server (like refusing to open a file), then the log files will probably be quite useful. Depending on the problem, a log level of between 3 and 10 showing the problem may be appropriate. A higher level gives more detail but may use too much disk space.

To set the debug level, use the *log level* in your `smb.conf`. You may also find it useful to set the log level higher for just one machine and keep separate logs for each machine. To do this, add the following lines to your main `smb.conf` file:

```
        log level = 10
        log file = /usr/local/samba/lib/log.%m
        include = /usr/local/samba/lib/smb.conf.%m
```

and create a file `/usr/local/samba/lib/smb.conf.`*`machine`* where *`machine`* is the name of the client you wish to debug. In that file put any `smb.conf` commands you want; for example, *log level* may be useful. This also allows you to experiment with different security systems, protocol levels, and so on, on just one machine.

The `smb.conf` entry *log level* is synonymous with the parameter *debuglevel* that has been used in older versions of Samba and is being retained for backward compatibility of `smb.conf` files.

As the *log level* value is increased, you will record a significantly greater level of debugging information. For most debugging operations, you may not need a setting higher than `3`. Nearly all bugs can be tracked at a setting of `10`, but be prepared for a large volume of log data.

40.3.1 Debugging-Specific Operations

Samba-3.x permits debugging (logging) of specific functional components without unnecessarily cluttering the log files with detailed logs for all operations. An example configuration to achieve this is shown in:

```
        log level = 0 tdb:3 passdb:5 auth:4 vfs:2
        max log size = 0
        log file = /var/log/samba/%U.%m.log
```

This will cause the level of detail to be expanded to the debug class (log level) passed to each functional area per the value shown above. The first value passed to the *log level* of 0 means turn off all unnecessary debugging except the debug classes set for the functional areas as specified. The table shown in Table 40.1 may be used to attain very precise analysis of each SMB operation Samba is conducting.

Table 40.1 Debuggable Functions

Function Name	Function Name
all	passdb
tdb	sam
printdrivers	auth
lanman	winbind
smb	vfs
rpc_parse	idmap
rpc_srv	quota
rpc_cli	acls

40.4 Internal Errors

If you get the message "INTERNAL ERROR" in your log files, it means that Samba got an unexpected signal while running. It is probably a segmentation fault and almost certainly means a bug in Samba (unless you have faulty hardware or system software).

If the message came from smbd, it will probably be accompanied by a message that details the last SMB message received by smbd. This information is often useful in tracking down the problem, so please include it in your bug report.

You should also detail how to reproduce the problem, if possible. Please make this reasonably detailed.

You may also find that a core file appeared in a `corefiles` subdirectory of the directory where you keep your Samba log files. This file is the most useful tool for tracking down the bug. To use it, you do this:

```
$ gdb smbd core
```

adding appropriate paths to smbd and core so gdb can find them. If you do not have gdb, try `dbx`. Then within the debugger, use the command **where** to give a stack trace of where the problem occurred. Include this in your report.

If you know any assembly language, do a **disass** of the routine where the problem occurred (if it's in a library routine, then disassemble the routine that called it) and try to work out exactly where the problem is by looking at the surrounding code. Even if you do not know assembly, including this information in the bug report can be useful.

40.5 Attaching to a Running Process

Unfortunately, some UNIXes (in particular some recent Linux kernels) refuse to dump a core file if the task has changed UID (which smbd does often). To debug with this sort of system, you could try to attach to the running process using `gdb smbd` *`PID`*, where you get *`PID`* from smbstatus. Then use **c** to continue and try to cause the core dump using the client. The debugger should catch the fault and tell you where it occurred.

Sometimes it is necessary to build Samba binary files that have debugging symbols so as to make it possible to capture enough information from a crashed operation to permit the Samba Team to fix the problem.

Compile with `-g` to ensure you have symbols in place. Add the following line to the `smb.conf` file global section:

```
panic action = "/bin/sleep 90000"
```

to catch any panics. If **smbd** seems to be frozen, look for any sleep processes. If it is not, and appears to be spinning, find the PID of the spinning process and type:

```
root#  gdb /usr/local/samba/sbin/smbd
```

then "attach 'pid'" (of the spinning process), then type "bt" to get a backtrace to see where the smbd is in the call path.

40.6 Patches

The best sort of bug report is one that includes a fix! If you send us patches, please use `diff -u` format if your version of diff supports it; otherwise, use `diff -c4`. Make sure you do the diff against a clean version of the source and let me know exactly what version you used.

Chapter 41

MANAGING TDB FILES

41.1 Features and Benefits

Samba uses a lightweight database called Trivial Database (tdb) in which it stores persistent and transient data. Some tdb files can be disposed of before restarting Samba, but others are used to store information that is vital to Samba configuration and behavior. The following information is provided to help administrators who are seeking to better manage their Samba installations.

Those who package Samba for commercial distribution with operating systems and appliances would do well to take note that tdb files can get corrupted, and for this reason ought to be backed up regularly. An appropriate time is at system shutdown (backup) and startup (restore from backup).

41.2 Managing TDB Files

The **tdbbackup** utility is a tool that may be used to backup samba tdb files. This tool may also be used to verify the integrity of the tdb files prior to Samba startup or during normal operation. If it finds file damage it will search for a prior backup the backup file from which the damaged tdb file will be restored. The **tdbbackup** utility can safely be run at any time. It was designed so that it can be used at any time to validate the integrity of tdb files, even during Samba operation.

It is recommended to backup all tdb files as part of the Samba start-up scripts on a Samba server. The following command syntax can be used:

```
myserver# > cd /var/lib/samba
myserver@ > tdbbackup *.tdb
```

The default extension is `.bak`. Any alternate extension can be specified by executing `tdbbackup -s 'new_extension' *.tdb` as part of your startup script.

Table 41.1 Samba's Trivial Database Files

Ca
Unexpected pack
WINS databa

Part VI

Reference Section

Chapter 42

HOW TO COMPILE SAMBA

You can obtain the Samba source file from the Samba Web site[1]. To obtain a development version, you can download Samba from Subversion or using **rsync**.

42.1 Access Samba Source Code via Subversion

42.1.1 Introduction

Samba is developed in an open environment. Developers use a Subversion to "checkin" (also known as "commit") new source code. Samba's various Subversion branches can be accessed via anonymous Subversion using the instructions detailed in this chapter.

This chapter is a modified version of the instructions found at the Samba[2] Web site.

42.1.2 Subversion Access to samba.org

The machine samba.org runs a publicly accessible Subversion repository for access to the source code of several packages, including Samba, rsync, distcc, ccache, and jitterbug. There are two main ways of accessing the Subversion server on this host.

[1] <http://samba.org/>
[2] <http://samba.org/samba/subversion.html>

42.1.2.1 Access via ViewCVS

You can access the source code via your favorite WWW browser. This allows you to access the contents of individual files in the repository and also to look at the revision history and commit logs of individual files. You can also ask for a diff listing between any two versions on the repository.

Use the URL <`http://viewcvs.samba.org/`>.

42.1.2.2 Access via Subversion

You can also access the source code via a normal Subversion client. This gives you much more control over what you can do with the repository and allows you to check out whole source trees and keep them up to date via normal Subversion commands. This is the preferred method of access if you are a developer and not just a casual browser.

In order to be able to download the Samba sources off Subversion, you need a Subversion client. Your distribution might include one, or you can download the sources from <`http://subversion.tigris.org/`>.

To gain access via anonymous Subversion, use the following steps. Retrieving Samba using Subversion

1. Install a recent copy of Subversion. All you really need is a copy of the Subversion client binary.

2. Run the command

   ```
   svn co svn://svnanon.samba.org/samba/trunk samba.
   ```

   This will create a directory called `samba` containing the latest Samba source code (usually the branch that is going to be the next major release). This currently corresponds to the 3.1 development tree. Subversion branches other then trunk can be obtained by adding branches/BRANCH_NAME to the URL you check out. A list of branch names can be found on the "Development" page of the Samba Web site. A common request is to obtain the latest 3.0 release code. This could be done by using the following command:

```
svn co svn://svnanon.samba.org/samba/branches/SAMBA_3_0 samba_3.
```

3. Whenever you want to merge in the latest code changes, use the following command from within the Samba directory:

```
svn update
```

42.2 Accessing the Samba Sources via rsync and ftp

pserver.samba.org also exports unpacked copies of most parts of the Subversion tree at the Samba pserver[3] location and also via anonymous rsync at the Samba rsync[4] server location. I recommend using rsync rather than ftp, because rsync is capable of compressing data streams, but it is also more useful than FTP because during a partial update it will transfer only the data that is missing plus a small overhead. See the rsync home page[5] for more info on rsync.

The disadvantage of the unpacked trees is that they do not support automatic merging of local changes as Subversion does. **rsync** access is most convenient for an initial install.

42.3 Verifying Samba's PGP Signature

It is strongly recommended that you verify the PGP signature for any source file before installing it. Even if you're not downloading from a mirror site, verifying PGP signatures should be a standard reflex. Many people today use the GNU GPG tool set in place of PGP. GPG can substitute for PGP.

With that said, go ahead and download the following files:

```
$ wget http://us1.samba.org/samba/ftp/samba-3.0.20.tar.asc
```

[3] <ftp://pserver.samba.org/pub/unpacked>
[4] <rsync://pserver.samba.org/ftp/unpacked/>
[5] <http://rsync.samba.org/>

```
$ wget http://us1.samba.org/samba/ftp/samba-pubkey.asc
```

The first file is the PGP signature for the Samba source file; the other is the Samba public PGP key itself. Import the public PGP key with:

```
$ gpg --import samba-pubkey.asc
```

and verify the Samba source code integrity with:

```
$ gzip -d samba-3.0.20.tar.gz
$ gpg --verify samba-3.0.20.tar.asc
```

If you receive a message like, "Good signature from Samba Distribution Verification Key...," then all is well. The warnings about trust relationships can be ignored. An example of what you would not want to see would be:

```
gpg: BAD signature from Samba Distribution Verification Key
```

42.4 Building the Binaries

After the source tarball has been unpacked, the next step involves configuration to match Samba to your operating system platform. If your source directory does not contain the **configure** script, it is necessary to build it before you can continue. Building of the configure script requires the correct version of the autoconf tool kit. Where the necessary version of autoconf is present, the configure script can be generated by executing the following:

```
root#  cd samba-3.0.20/source
root#  ./autogen.sh
```

To build the binaries, run the program `./configure` in the source directory. This should automatically configure Samba for your operating system. If you have unusual needs, then you may wish to first run:

```
root# ./configure --help
```

This will help you to see what special options can be enabled. Now execute `./configure` with any arguments it might need:

```
root# ./configure [... arguments ...]
```

Execute the following create the binaries:

```
root#  make
```

Once it is successfully compiled, you can execute the command shown here to install the binaries and manual pages:

```
root#  make install
```

Some people prefer to install binary files and man pages separately. If this is your wish, the binary files can be installed by executing:

```
root#  make installbin
```

The man pages can be installed using this command:

```
root#  make installman
```

Note that if you are upgrading from a previous version of Samba the old versions of the binaries will be renamed with an ".old" extension. You can go back to the previous version by executing:

```
root#  make revert
```

As you can see from this, building and installing Samba does not need to result in disaster!

42.4.1 Compiling Samba with Active Directory Support

In order to compile Samba with ADS support, you need to have installed on your system:

- The MIT or Heimdal Kerberos development libraries (either install from the sources or use a package).
- The OpenLDAP development libraries.

If your Kerberos libraries are in a nonstandard location, then remember to add the configure option `--with-krb5=`*`DIR`*.

After you run configure, make sure that the `include/config.h` it generates contain lines like this:

```
#define HAVE_KRB5 1
#define HAVE_LDAP 1
```

If it does not, configure did not find your KRB5 libraries or your LDAP libraries. Look in `config.log` to figure out why and fix it.

42.4.1.1 Installing the Required Packages for Debian

On Debian, you need to install the following packages:

- libkrb5-dev
- krb5-user

42.4.1.2 Installing the Required Packages for Red Hat Linux

On Red Hat Linux, this means you should have at least:

- krb5-workstation (for kinit)
- krb5-libs (for linking with)
- krb5-devel (because you are compiling from source)

in addition to the standard development environment.

If these files are not installed on your system, you should check the installation CDs to find which has them and install the files using your tool of choice. If in doubt about what tool to use, refer to the Red Hat Linux documentation.

42.4.1.3 SuSE Linux Package Requirements

SuSE Linux installs Heimdal packages that may be required to allow you to build binary packages. You should verify that the development libraries have been installed on your system.

SuSE Linux Samba RPMs support Kerberos. Please refer to the documentation for your SuSE Linux system for information regarding SuSE Linux specific configuration. Additionally, SuSE is very active in the maintenance of Samba packages that provide the maximum capabilities that are available. You should consider using SuSE-provided packages where they are available.

42.5 Starting the smbd nmbd and winbindd

You must choose to start smbd, winbindd and nmbd either as daemons or from inetd. Don't try to do both! Either you can put them in `inetd.conf` and have them started on demand by inetd or xinetd, or you can start them as daemons either from the command-line or in `/etc/rc.local`. See the man pages for details on the command line options. Take particular care to read the bit about what user you need to have to start Samba. In many cases, you must be root.

The main advantage of starting smbd and nmbd using the recommended daemon method is that they will respond slightly more quickly to an initial connection request.

42.5.1 Starting from inetd.conf

NOTE

The following will be different if you use NIS, NIS+, or LDAP to distribute services maps.

Look at your `/etc/services`. What is defined at port 139/tcp? If nothing is defined, then add a line like this:

```
netbios-ssn     139/tcp
```

Similarly for 137/udp, you should have an entry like:

```
netbios-ns  137/udp
```

Next, edit your `/etc/inetd.conf` and add two lines like this:

```
netbios-ssn stream tcp nowait root /usr/local/samba/sbin/smbd smbd
netbios-ns dgram udp wait root /usr/local/samba/sbin/nmbd nmbd
```

The exact syntax of `/etc/inetd.conf` varies between UNIXes. Look at the other entries in inetd.conf for a guide.

Some distributions use xinetd instead of inetd. Consult the xinetd manual for configuration information.

NOTE

Some UNIXes already have entries like netbios_ns (note the underscore) in `/etc/services`. You must edit `/etc/services` or `/etc/inetd.conf` to make them consistent.

NOTE

On many systems you may need to use the *interfaces* option in `smb.conf` to specify the IP address and netmask of your interfaces. Run ifconfig as root if you do not know what the broadcast is for your net. nmbd tries to determine it at runtime, but fails on some UNIXes.

WARNING

Many UNIXes only accept around five parameters on the command line in `inetd.conf`. This means you shouldn't use spaces between the options and arguments, or you should use a script and start the script from **inetd**.

Restart inetd, perhaps just send it a HUP, like this:

```
root# killall -HUP inetd
```

42.5.2 Alternative: Starting smbd as a Daemon

To start the server as a daemon, you should create a script something like this one, perhaps calling it `startsmb`.

```
#!/bin/sh
/usr/local/samba/sbin/smbd -D
/usr/local/samba/sbin/winbindd -D
/usr/local/samba/sbin/nmbd -D
```

Make it executable with **chmod +x startsmb**.

You can then run **startsmb** by hand or execute it from `/etc/rc.local`.

To kill it, send a kill signal to the processes nmbd and smbd.

NOTE

If you use the SVR4-style init system, you may like to look at the `examples/svr4-startup` script to make Samba fit into that system.

42.5.2.1 Starting Samba for Red Hat Linux

Red Hat Linux has not always included all Samba components in the standard installation. So versions of Red Hat Linux do not install the winbind utility, even though it is present on the installation CDROM media. Check to see if the **winbindd** is present on the system:

```
root#  ls /usr/sbin/winbindd
/usr/sbin/winbindd
```

This means that the appropriate RPM package was installed. The following response means that it is not installed:

```
/bin/ls: /usr/sbin/winbind: No such file or directory
```

In this case, it should be installed if you intend to use **winbindd**. Search the CDROM installation media for the samba-winbind RPM and install it following Red Hat guidelines.

The process for starting Samba will now be outlined. Be sure to configure Samba's `smb.conf` file before starting Samba. When configured, start Samba by executing:

```
root#  service smb start
root#  service winbind start
```

These steps will start nmbd, smbd and winbindd.

To ensure that these services will be automatically restarted when the system is rebooted execute:

```
root#  chkconfig smb on
root#  chkconfig winbind on
```

Samba will be started automatically at every system reboot.

42.5.2.2 Starting Samba for Novell SUSE Linux

Novell SUSE Linux products automatically install all essential Samba components in a default installation. Configure your `smb.conf` file, then execute the following to start Samba:

```
root#  rcnmb start
root#  rcsmb start
root#  rcwinbind start
```

Now execute these commands so that Samba will be started automatically following a system reboot:

```
root#  chkconfig nmb on
root#  chkconfig smb on
root#  chkconfig winbind on
```

The Samba services will now be started automatically following a system reboot.

Chapter 43

PORTABILITY

Samba works on a wide range of platforms, but the interface all the platforms provide is not always compatible. This chapter contains platform-specific information about compiling and using Samba.

43.1 HPUX

Hewlett-Packard's implementation of supplementary groups is nonstandard (for historical reasons). There are two group files, `/etc/group` and `/etc/logingroup`; the system maps UIDs to numbers using the former, but initgroups() reads the latter. Most system admins who know the ropes symlink `/etc/group` to `/etc/logingroup` (hard-link does not work for reasons too obtuse to go into here). initgroups() will complain if one of the groups you're in, in `/etc/logingroup`, has what it considers to be an invalid ID, which means outside the range `[0..UID_MAX]`, where `UID_MAX` is 60000 currently on HP-UX. This precludes -2 and 65534, the usual `nobody` GIDs.

If you encounter this problem, make sure the programs that are failing to initgroups() are run as users, not in any groups with GIDs outside the allowed range.

This is documented in the HP manual pages under setgroups(2) and passwd(4).

On HP-UX you must use gcc or the HP ANSI compiler. The free compiler that comes with HP-UX is not ANSI compliant and cannot compile Samba.

43.2 SCO UNIX

If you run an old version of SCO UNIX, you may need to get important TCP/IP patches for Samba to work correctly. Without the patch, you may encounter corrupt data transfers using Samba.

The patch you need is UOD385 Connection Drivers SLS. It is available from SCO ftp.sco.com[1], directory SLS, files uod385a.Z and uod385a.ltr.Z).

The information provided here refers to an old version of SCO UNIX. If you require binaries for more recent SCO UNIX products, please contact SCO to obtain packages that are ready to install. You should also verify with SCO that your platform is up to date for the binary packages you will install. This is important if you wish to avoid data corruption problems with your installation. To build Samba for SCO UNIX products may require significant patching of Samba source code. It is much easier to obtain binary packages directly from SCO.

43.3 DNIX

DNIX has a problem with seteuid() and setegid(). These routines are needed for Samba to work correctly, but they were left out of the DNIX C library for some reason.

For this reason Samba by default defines the macro NO_EID in the DNIX section of includes.h. This works around the problem in a limited way, but it is far from ideal, and some things still will not work right.

To fix the problem properly, you need to assemble the following two functions and then either add them to your C library or link them into Samba. Put the following in the file `setegid.s`:

```
        .globl  _setegid
_setegid:
        moveq   #47,d0
        movl    #100,a0
        moveq   #1,d1
        movl    4(sp),a1
        trap    #9
```

[1] <ftp://ftp.sco.com/>

```
        bccs    1$
        jmp     cerror
1$:
        clrl    d0
        rts
```

Put this in the file `seteuid.s`:

```
        .globl  _seteuid
_seteuid:
        moveq   #47,d0
        movl    #100,a0
        moveq   #0,d1
        movl    4(sp),a1
        trap    #9
        bccs    1$
        jmp     cerror
1$:
        clrl    d0
        rts
```

After creating the files, you then assemble them using

```
$ as seteuid.s
$ as setegid.s
```

which should produce the files `seteuid.o` and `setegid.o`.

Next you need to add these to the LIBSM line in the DNIX section of the Samba Makefile. Your LIBSM line will look something like this:

```
LIBSM = setegid.o seteuid.o -ln
```

You should then remove the line:

```
#define NO_EID
```

from the DNIX section of `includes.h`.

43.4 Red Hat Linux

By default during installation, some versions of Red Hat Linux add an entry to `/etc/hosts` as follows:

```
127.0.0.1 loopback "hostname"."domainname"
```

This causes Samba to loop back onto the loopback interface. The result is that Samba fails to communicate correctly with the world and therefore may fail to correctly negotiate who is the master browse list holder and who is the master browser.

Corrective action: Delete the entry after the word "loopback" in the line starting 127.0.0.1.

43.5 AIX: Sequential Read Ahead

Disabling sequential read ahead can improve Samba performance significantly when there is a relatively high level of multiprogramming (many smbd processes or mixed with another workload), not an abundance of physical memory or slower disk technology. These can cause AIX to have a higher WAIT values. Disabling sequential read-ahead can also have an adverse affect on other workloads in the system so you will need to evaluate other applications for impact.

It is recommended to use the defaults provided by IBM, but if you experience a high amount of wait time, try disabling read-ahead with the following commands:

For AIX 5.1 and earlier: `vmtune -r 0`

For AIX 5.2 and later jfs filesystems: `ioo -o minpgahead=0`

For AIX 5.2 and later jfs2 filesystems: `ioo -o j2_minPageReadAhead=0`

If you have a mix of jfs and jfs2 filesystems on the same host, simply use both ioo commands.

43.6 Solaris

43.6.1 Locking Improvements

Some people have been experiencing problems with F_SETLKW64/fcntl when running Samba on Solaris. The built-in file-locking mechanism was not scalable. Performance would degrade to the point where processes would get into loops of trying to lock a file. It would try a lock, then fail, then try again. The lock attempt was failing before the grant was occurring. The visible manifestation of this was a handful of processes stealing all of the CPU, and when they were trussed, they would be stuck in F_SETLKW64 loops.

Please check with Sun support for current patches needed to fix this bug. The patch revision for 2.6 is 105181-34, for 8 is 108528-19, and for 9 is 112233-04. After the installation of these patches, it is recommended to reconfigure and rebuild Samba.

Thanks to Joe Meslovich for reporting this.

43.6.2 Winbind on Solaris 9

Nsswitch on Solaris 9 refuses to use the Winbind NSS module. This behavior is fixed by Sun in patch 112960-14[2].

[2] <http://sunsolve.sun.com/search/advsearch.do?collection=PATCH&type=collections&max=50&language=en&queryKey5=112960;rev=14&toDocument=yes>

Chapter 44

SAMBA AND OTHER CIFS CLIENTS

This chapter contains client-specific information.

44.1 Macintosh Clients

Yes. Thursby[1] has a CIFS client/server called DAVE[2]. They test it against Windows 95, Windows NT/200x/XP, and Samba for compatibility issues. At the time of this writing, DAVE was at version 5.1. Please refer to Thursby's Web site for more information regarding this product.

Alternatives include two free implementations of AppleTalk for several kinds of UNIX machines and several more commercial ones. These products allow you to run file services and print services natively to Macintosh users, with no additional support required on the Macintosh. The two free implementations are Netatalk[3] and CAP[4]. What Samba offers MS Windows users, these packages offer to Macs. For more info on these packages, Samba, and Linux (and other UNIX-based systems), see http://www.eats.com/linux_mac_win.html.[5]

Newer versions of the Macintosh (Mac OS X) include Samba.

[1] <http://www.thursby.com/>
[2] <http://www.thursby.com/products/dave.html>
[3] <http://www.umich.edu/~rsug/netatalk/>
[4] <http://www.cs.mu.oz.au/appletalk/atalk.html>
[5] <http://www.eats.com/linux_mac_win.html>

44.2 OS2 Client

44.2.1 Configuring OS/2 Warp Connect or OS/2 Warp 4

Basically, you need three components:

- The File and Print Client (IBM peer)
- TCP/IP (Internet support)
- The "NetBIOS over TCP/IP" driver (TCPBEUI)

Installing the first two together with the base operating system on a blank system is explained in the Warp manual. If Warp has already been installed, but you now want to install the networking support, use the "Selective Install for Networking" object in the "System Setup" folder.

Adding the "NetBIOS over TCP/IP" driver is not described in the manual and just barely in the online documentation. Start **MPTS.EXE**, click on **OK**, click on **Configure LAPS**, and click on **IBM OS/2 NETBIOS OVER TCP/IP** in **Protocols**. This line is then moved to **Current Configuration**. Select that line, click on **Change number**, and increase it from 0 to 1. Save this configuration.

If the Samba server is not on your local subnet, you can optionally add IP names and addresses of these servers to the **Names List** or specify a WINS server (NetBIOS Nameserver in IBM and RFC terminology). For Warp Connect, you may need to download an update for `IBM Peer` to bring it on the same level as Warp 4. See the IBM OS/2 Warp Web page

44.2.2 Configuring Other Versions of OS/2

This sections deals with configuring OS/2 Warp 3 (not Connect), OS/2 1.2, 1.3 or 2.x.

You can use the free Microsoft LAN Manager 2.2c Client for OS/2 that is available from ftp://ftp.microsoft.com/BusSys/Clients/LANMAN.OS2/[6]. In a nutshell, edit the file `\OS2VER` in the root directory of the OS/2 boot partition and add the lines:

```
20=setup.exe
```

[6] <ftp://ftp.microsoft.com/BusSys/Clients/LANMAN.OS2/>

```
20=netwksta.sys
20=netvdd.sys
```

before you install the client. Also, do not use the included NE2000 driver because it is buggy. Try the NE2000 or NS2000 driver from ftp://ftp.cdrom.com/pub/os2/network/ndis/[7] instead.

44.2.3 Printer Driver Download for OS/2 Clients

Create a share called *[PRINTDRV]* that is world-readable. Copy your OS/2 driver files there. The `.EA_` files must still be separate, so you will need to use the original install files and not copy an installed driver from an OS/2 system.

Install the NT driver first for that printer. Then, add to your `smb.conf` a parameter, *os2 driver map*. Next, in the file specified by *`filename`*, map the name of the NT driver name to the OS/2 driver name as follows:

`nt driver name = os2 driver name.device name`, e.g.,

`HP LaserJet 5L = LASERJET.HP LaserJet 5L`

You can have multiple drivers mapped in this file.

If you only specify the OS/2 driver name, and not the device name, the first attempt to download the driver will actually download the files, but the OS/2 client will tell you the driver is not available. On the second attempt, it will work. This is fixed simply by adding the device name to the mapping, after which it will work on the first attempt.

44.3 Windows for Workgroups

44.3.1 Latest TCP/IP Stack from Microsoft

Use the latest TCP/IP stack from Microsoft if you use Windows for Workgroups. The early TCP/IP stacks had lots of bugs.

Microsoft has released an incremental upgrade to its TCP/IP 32-bit VxD drivers. The latest release can be found at ftp.microsoft.com, located in

[7] <ftp://ftp.cdrom.com/pub/os2/network/ndis/>

/Softlib/MSLFILES/TCP32B.EXE. There is an update.txt file there that describes the problems that were fixed. New files include WINSOCK.DLL, TELNET.EXE, WSOCK.386, VNBT.386, WSTCP.386, TRACERT.EXE, NETSTAT.EXE, and NBTSTAT.EXE.

More information about this patch is available in Knowledge Base article 99891[8].

44.3.2 Delete .pwl Files After Password Change

Windows for Workgroups does a lousy job with passwords. When you change passwords on either the UNIX box or the PC, the safest thing to do is delete the .pwl files in the Windows directory. The PC will complain about not finding the files, but will soon get over it, allowing you to enter the new password.

If you do not do this, you may find that Windows for Workgroups remembers and uses the old password, even if you told it a new one.

Often Windows for Workgroups will totally ignore a password you give it in a dialog box.

44.3.3 Configuring Windows for Workgroups Password Handling

There is a program call admincfg.exe on the last disk (disk 8) of the WFW 3.11 disk set. To install it, type EXPAND A:\ADMINCFG.EX_ C:\WINDOWS\ADMINCFG.EXE. Then add an icon for it via the Program Manager **New** menu. This program allows you to control how WFW handles passwords, Disable Password Caching and so on, for use with *security* = user.

44.3.4 Password Case Sensitivity

Windows for Workgroups uppercases the password before sending it to the server. UNIX passwords can be case-sensitive though. Check the smb.conf information on *password level* to specify what characters Samba should try to uppercase when checking.

[8] <http://support.microsoft.com/kb/q99891/>

44.3.5 Use TCP/IP as Default Protocol

To support print queue reporting, you may find that you have to use TCP/IP as the default protocol under Windows for Workgroups. For some reason, if you leave NetBEUI as the default, it may break the print queue reporting on some systems. It is presumably a Windows for Workgroups bug.

44.3.6 Speed Improvement

Note that some people have found that setting *`DefaultRcvWindow`* in the *[MSTCP]* section of the `SYSTEM.INI` file under Windows for Workgroups to 3072 gives a big improvement.

My own experience with DefaultRcvWindow is that I get a much better performance with a large value (16384 or larger). Other people have reported that anything over 3072 slows things down enormously. One person even reported a speed drop of a factor of 30 when he went from 3072 to 8192.

44.4 Windows 95/98

When using Windows 95 OEM SR2, the following updates are recommended where Samba is being used. Please note that the changes documented in Section 44.3.6 will affect you once these updates have been installed.

There are more updates than the ones mentioned here. Refer to the Microsoft Web site for all currently available updates to your specific version of Windows 95.

Kernel Update: KRNLUPD.EXE
Ping Fix: PINGUPD.EXE
RPC Update: RPCRTUPD.EXE
TCP/IP Update: VIPUPD.EXE
Redirector Update: VRDRUPD.EXE

Also, if using MS Outlook, it is desirable to install the **OLEUPD.EXE** fix. This fix may stop your machine from hanging for an extended period when exiting Outlook, and you may notice a significant speedup when accessing network neighborhood services.

44.4.1 Speed Improvement

Configure the Windows 95 TCP/IP registry settings to give better performance. I use a program called **MTUSPEED.exe** that I got off the Internet. There are various other utilities of this type freely available.

44.5 Windows 2000 Service Pack 2

There are several annoyances with Windows 2000 SP2, one of which only appears when using a Samba server to host user profiles to Windows 2000 SP2 clients in a Windows domain. This assumes that Samba is a member of the domain, but the problem will most likely occur if it is not.

In order to serve profiles successfully to Windows 2000 SP2 clients (when not operating as a PDC), Samba must have *nt acl support* = no added to the file share that houses the roaming profiles. If this is not done, then the Windows 2000 SP2 client will complain about not being able to access the profile (Access Denied) and create multiple copies of it on disk (DOMAIN.user.001, DOMAIN.user.002, and so on). See the `smb.conf` man page for more details on this option. Also note that the *nt acl support* parameter was formally a global parameter in releases prior to Samba 2.2.2.

Example 44.5.1 provides a minimal profile share.

Example 44.5.1 Minimal Profile Share

```
[profile]
        path = /export/profile
        create mask = 0600
        directory mask = 0700
        nt acl support = no
        read only = no
```

The reason for this bug is that the Windows 200x SP2 client copies the security descriptor for the profile that contains the Samba server's SID, and not the domain SID. The client compares the SID for SAMBA\user and realizes it is different from the one assigned to DOMAIN\user; hence, access denied message.

When the *nt acl support* parameter is disabled, Samba will send the Windows 200x client a response to the QuerySecurityDescriptor trans2 call,

which causes the client to set a default ACL for the profile. This default ACL includes:

DOMAIN\user "Full Control">

NOTE

This bug does not occur when using Winbind to create accounts on the Samba host for Domain users.

44.6 Windows NT 3.1

If you have problems communicating across routers with Windows NT 3.1 workstations, read this Microsoft Knowledge Base article:[9].

[9] <http://support.microsoft.com/default.aspx?scid=kb;Q103765>

Chapter 45

SAMBA PERFORMANCE TUNING

45.1 Comparisons

The Samba server uses TCP to talk to the client, so if you are trying to see if it performs well, you should really compare it to programs that use the same protocol. The most readily available programs for file transfer that use TCP are ftp or another TCP-based SMB server.

If you want to test against something like an NT or Windows for Workgroups server, then you will have to disable all but TCP on either the client or server. Otherwise, you may well be using a totally different protocol (such as NetBEUI) and comparisons may not be valid.

Generally, you should find that Samba performs similarly to ftp at raw transfer speed. It should perform quite a bit faster than NFS, although this depends on your system.

Several people have done comparisons between Samba and Novell, NFS, or Windows NT. In some cases Samba performed the best, in others the worst. I suspect the biggest factor is not Samba versus some other system, but the hardware and drivers used on the various systems. Given similar hardware, Samba should certainly be competitive in speed with other systems.

45.2 Socket Options

There are a number of socket options that can greatly affect the performance of a TCP-based server like Samba.

The socket options that Samba uses are settable both on the command line with the `-O` option and in the `smb.conf` file.

The *socket options* section of the `smb.conf` manual page describes how to set these and gives recommendations.

Getting the socket options correct can make a big difference to your performance, but getting them wrong can degrade it by just as much. The correct settings are very dependent on your local network.

The socket option TCP_NODELAY is the one that seems to make the biggest single difference for most networks. Many people report that adding *socket options* = TCP_NODELAY doubles the read performance of a Samba drive. The best explanation I have seen for this is that the Microsoft TCP/IP stack is slow in sending TCP ACKs.

There have been reports that setting *`socket options = SO_RCVBUF=8192`* in smb.conf can seriously degrade Samba performance on the loopback adaptor (IP Address 127.0.0.1). It is strongly recommended that before specifying any settings for *`socket options`*, the effect first be quantitatively measured on the server being configured.

45.3 Read Size

The option *read size* affects the overlap of disk reads/writes with network reads/writes. If the amount of data being transferred in several of the SMB commands (currently SMBwrite, SMBwriteX, and SMBreadbraw) is larger than this value, then the server begins writing the data before it has received the whole packet from the network, or in the case of SMBreadbraw, it begins writing to the network before all the data has been read from disk.

This overlapping works best when the speeds of disk and network access are similar, having little effect when the speed of one is much greater than the other.

The default value is 16384, but little experimentation has been done as yet to determine the optimal value, and it is likely that the best value will vary greatly between systems anyway. A value over 65536 is pointless and will cause you to allocate memory unnecessarily.

45.4 Max Xmit

At startup the client and server negotiate a *maximum transmit* size, which limits the size of nearly all SMB commands. You can set the maximum size that Samba will negotiate using the *max xmit* option in `smb.conf`. Note that this is the maximum size of SMB requests that Samba will accept, but not the maximum size that the client will accept. The client maximum receive size is sent to Samba by the client, and Samba honors this limit.

It defaults to 65536 bytes (the maximum), but it is possible that some clients may perform better with a smaller transmit unit. Trying values of less than 2048 is likely to cause severe problems. In most cases the default is the best option.

45.5 Log Level

If you set the log level (also known as *debug level*) higher than 2, then you may suffer a large drop in performance. This is because the server flushes the log file after each operation, which can be quite expensive.

45.6 Read Raw

The *read raw* operation is designed to be an optimized, low-latency file read operation. A server may choose to not support it, however, and Samba makes support for *read raw* optional, with it being enabled by default.

In some cases clients do not handle *read raw* very well and actually get lower performance using it than they get using the conventional read operations, so you might like to try *read raw* = no and see what happens on your network. It might lower, raise, or not affect your performance. Only testing can really tell.

45.7 Write Raw

The *write raw* operation is designed to be an optimized, low-latency file write operation. A server may choose to not support it, however, and Samba makes support for *write raw* optional, with it being enabled by default.

Some machines may find *write raw* slower than normal write, in which case you may wish to change this option.

45.8 Slow Logins

Slow logins are almost always due to the password checking time. Using the lowest practical *password level* will improve things.

45.9 Client Tuning

Often a speed problem can be traced to the client. The client (for example Windows for Workgroups) can often be tuned for better TCP performance. Check the sections on the various clients in Chapter 44, "Samba and Other CIFS Clients".

45.10 Samba Performance Problem Due to Changing Linux Kernel

A user wrote the following to the mailing list:

> I am running Gentoo on my server and Samba 2.2.8a. Recently I changed kernel versions from `linux-2.4.19-gentoo-r10` to `linux-2.4.20-wolk4.0s`. Now I have a performance issue with Samba. Many of you will probably say, "Move to vanilla sources!" Well, I tried that and it didn't work. I have a 100MB LAN and two computers (Linux and Windows 2000). The Linux server shares directories with DivX files, the client (Windows 2000) plays them via LAN. Before, when I was running the 2.4.19 kernel, everything was fine, but now movies freeze and stop. I tried moving files between the server and Windows, and it is terribly slow.

The answer he was given is:

> Grab the mii-tool and check the duplex settings on the NIC. My guess is that it is a link layer issue, not an application layer problem. Also run ifconfig and verify that the framing error, collisions, and so on, look normal for ethernet.

45.11 Corrupt tdb Files

Our Samba PDC server has been hosting three TB of data to our 500+ users [Windows NT/XP] for the last three years using Samba without a problem. Today all shares went very slow. Also, the main smbd kept spawning new processes, so we had 1600+ running SMDB's (normally we average 250). It crashed the SUN E3500 cluster twice. After a lot of searching, I decided to **rm /var/locks/*.tdb**. Happy again.

Question: Is there any method of keeping the *.tdb files in top condition, or how can I detect early corruption?

Answer: Yes, run **tdbbackup** each time after stopping nmbd and before starting nmbd.

Question: What I also would like to mention is that the service latency seems a lot lower than before the locks cleanup. Any ideas on keeping it top notch?

Answer: Yes. Same answer as for previous question!

45.12 Samba Performance is Very Slow

A site reported experiencing very baffling symptoms with MYOB Premier opening and accessing its data files. Some operations on the file would take between 40 and 45 seconds.

It turned out that the printer monitor program running on the Windows clients was causing the problems. From the logs, we saw activity coming through with pauses of about 1 second.

Stopping the monitor software resulted in the networks access at normal (quick) speed. Restarting the program caused the speed to slow down again. The printer was a Canon LBP-810 and the relevant task was something like CAPON (not sure on spelling). The monitor software displayed a "printing now" dialog on the client during printing.

We discovered this by starting with a clean install of Windows and trying the application at every step of the installation of other software process (we had to do this many times).

Moral of the story: Check everything (other software included)!

Chapter 46

LDAP AND TRANSPORT LAYER SECURITY

46.1 Introduction

Up until now, we have discussed the straightforward configuration of OpenLDAP™, with some advanced features such as ACLs. This does not however, deal with the fact that the network transmissions are still in plain text. This is where *Transport Layer Security (TLS)* comes in.

OpenLDAP™ clients and servers are capable of using the Transport Layer Security (TLS) framework to provide integrity and confidentiality protections in accordance with RFC 2830[1]; *Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security.*

TLS uses X.509 certificates. All servers are required to have valid certificates, whereas client certificates are optional. We will only be discussing server certificates.

[1] <http://rfc.net/rfc2830.html>

TIP

The DN of a server certificate must use the CN attribute to name the server, and the CN must carry the server's fully qualified domain name (FQDN). Additional alias names and wildcards may be present in the `subjectAltName` certificate extension. More details on server certificate names are in RFC2830[a].

[a] <http://rfc.net/rfc2830.html>

We will discuss this more in the next sections.

46.2 Configuring

Now on to the good bit.

46.2.1 Generating the Certificate Authority

In order to create the relevant certificates, we need to become our own Certificate Authority (CA). [2] This is necessary, so we can sign the server certificate.

We will be using the OpenSSL[3] [4] software for this, which is included with every great Linux™ distribution.

TLS is used for many types of servers, but the instructions[5] presented here, are tailored for OpenLDAP.

[2] We could however, get our generated server certificate signed by proper CAs, like Thawte <http://www.thawte.com/> and VeriSign <http://www.verisign.com/>, which you pay for, or the free ones, via CAcert <http://www.cacert.org/>

[3] <http://www.openssl.org>

[4] The downside to making our own CA, is that the certificate is not automatically recognized by clients, like the commercial ones are.

[5] For information straight from the horse's mouth, please visit <http://www.openssl.org/docs/HOWTO/>; the main OpenSSL site.

NOTE

The *Common Name (CN)*, in the following example, *MUST* be the fully qualified domain name (FQDN) of your ldap server.

First we need to generate the CA:

```
root#  mkdir myCA
```

Move into that directory:

```
root#  cd myCA
```

Now generate the CA:[6]

```
root#  /usr/share/ssl/misc/CA.pl -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 1024 bit RSA private key
.......................++++++
.............................++++++
writing new private key to './demoCA/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

[6]Your `CA.pl` or `CA.sh` might not be in the same location as mine is, you can find it by using the **locate** command, i.e., **locate CA.pl**. If the command complains about the database being too old, run **updatedb** as *root* to update it.

```
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AU
State or Province Name (full name) [Some-State]:NSW
Locality Name (eg, city) []:Sydney
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Abmas
Organizational Unit Name (eg, section) []:IT
Common Name (eg, YOUR name) []:ldap.abmas.biz
Email Address []:support@abmas.biz
```

There are some things to note here.

1. You *MUST* remember the password, as we will need it to sign the server certificate..
2. The *Common Name (CN)*, *MUST* be the fully qualified domain name (FQDN) of your ldap server.

46.2.2 Generating the Server Certificate

Now we need to generate the server certificate:

```
root#  openssl req -new -nodes -keyout newreq.pem -out newreq.pem
Generating a 1024 bit RSA private key
.............++++++
.........................................................++++++
writing new private key to 'newreq.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
```

```
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AU
State or Province Name (full name) [Some-State]:NSW
Locality Name (eg, city) []:Sydney
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Abmas
Organizational Unit Name (eg, section) []:IT
Common Name (eg, YOUR name) []:ldap.abmas.biz
Email Address []:support@abmas.biz

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Again, there are some things to note here.

1. You should *NOT* enter a password.
2. The *Common Name (CN)*, *MUST* be the fully qualified domain name (FQDN) of your ldap server.

Now we sign the certificate with the new CA:

```
root#  /usr/share/ssl/misc/CA.pl -sign
Using configuration from /etc/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
Serial Number: 1 (0x1)
Validity
   Not Before: Mar  6 18:22:26 2005 EDT
   Not After : Mar  6 18:22:26 2006 EDT
Subject:
   countryName               = AU
```

```
   stateOrProvinceName       = NSW
   localityName              = Sydney
   organizationName          = Abmas
   organizationalUnitName    = IT
   commonName                = ldap.abmas.biz
   emailAddress              = support@abmas.biz
X509v3 extensions:
   X509v3 Basic Constraints:
       CA:FALSE
   Netscape Comment:
       OpenSSL Generated Certificate
   X509v3 Subject Key Identifier:
       F7:84:87:25:C4:E8:46:6D:0F:47:27:91:F0:16:E0:86:6A:EE:A3:CE
   X509v3 Authority Key Identifier:
       keyid:27:44:63:3A:CB:09:DC:B1:FF:32:CC:93:23:A4:F1:B4:D5:F0:7E:CC
       DirName:/C=AU/ST=NSW/L=Sydney/O=Abmas/OU=IT/
                  CN=ldap.abmas.biz/emailAddress=support@abmas.biz
       serial:00

Certificate is to be certified until Mar  6 18:22:26 2006 EDT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Signed certificate is in newcert.pem
```

That completes the server certificate generation.

46.2.3 Installing the Certificates

Now we need to copy the certificates to the right configuration directories, rename them at the same time (for convenience), change the ownership and finally the permissions:

```
root#  cp demoCA/cacert.pem /etc/openldap/
root#  cp newcert.pem /etc/openldap/servercrt.pem
root#  cp newreq.pem /etc/openldap/serverkey.pem
root#  chown ldap.ldap /etc/openldap/*.pem
root#  chmod 640 /etc/openldap/cacert.pem;
root#  chmod 600 /etc/openldap/serverkey.pem
```

Now we just need to add these locations to `slapd.conf`, anywhere before the `database` declaration as shown here:

```
TLSCertificateFile /etc/openldap/servercrt.pem
TLSCertificateKeyFile /etc/openldap/serverkey.pem
TLSCACertificateFile /etc/openldap/cacert.pem
```

Here is the declaration and `ldap.conf`: `ldap.conf`

```
TLS_CACERT /etc/openldap/cacert.pem
```

That's all there is to it. Now on to Section 46.3

46.3 Testing

This is the easy part. Restart the server:

```
root#  /etc/init.d/ldap restart
Stopping slapd:                                            [  OK  ]
Checking configuration files for slapd: config file testing succeeded
Starting slapd:                                            [  OK  ]
```

Then, using **ldapsearch**, test an anonymous search with the -ZZ[7] option:

```
root#  ldapsearch -x -b "dc=ldap,dc=abmas,dc=biz" \
        -H 'ldap://ldap.abmas.biz:389' -ZZ
```

Your results should be the same as before you restarted the server, for example:

```
root#  ldapsearch -x -b "dc=ldap,dc=abmas,dc=biz" \
    -H 'ldap://ldap.abmas.biz:389' -ZZ

# extended LDIF
#
# LDAPv3
# base <> with scope sub
# filter: (objectclass=*)
# requesting: ALL
#

# abmas.biz
dn: dc=ldap,dc=abmas,dc=biz
objectClass: dcObject
objectClass: organization
o: Abmas
dc: abmas

# Manager, ldap.abmas.biz
dn: cn=Manager,dc=ldap,dc=abmas,dc=biz
objectClass: organizationalRole
cn: Manager

# ABMAS, abmas.biz
dn: sambaDomainName=ABMAS,dc=ldap,dc=abmas,dc=biz
sambaDomainName: ABMAS
```

[7] See **man ldapsearch**

```
sambaSID: S-1-5-21-238355452-1056757430-1592208922
sambaAlgorithmicRidBase: 1000
objectClass: sambaDomain
sambaNextUserRid: 67109862
sambaNextGroupRid: 67109863
```

If you have any problems, please read Section 46.4

46.4 Troubleshooting

The most common error when configuring TLS, as I have already mentioned numerous times, is that the *Common Name (CN)* you entered in Section 46.2.2 is *NOT* the Fully Qualified Domain Name (FQDN) of your ldap server.

Other errors could be that you have a typo somewhere in your **ldapsearch** command, or that your have the wrong permissions on the `servercrt.pem` and `cacert.pem` files. They should be set with **chmod 640**, as per Section 46.2.3.

For anything else, it's best to read through your ldap logfile or join the OpenLDAP mailing list.

Chapter 47

SAMBA SUPPORT

One of the most difficult to answer questions in the information technology industry is, "What is support?". That question irritates some folks, as much as common answers may annoy others.

The most aggravating situation pertaining to support is typified when, as a Linux user, a call is made to an Internet service provider who, instead of listening to the problem to find a solution, blandly replies: "Oh, Linux? We do not support Linux!". It has happened to me, and similar situations happen through-out the IT industry. Answers like that are designed to inform us that there are some customers that a business just does not want to deal with, and well may we feel the anguish of the rejection that is dished out.

One way to consider support is to view it as consisting of the right answer, in the right place, at the right time, no matter the situation. Support is all that it takes to take away pain, disruption, inconvenience, loss of productivity, disorientation, uncertainty, and real or perceived risk.

One of the forces that has become a driving force for the adoption of open source software is the fact that many IT businesses have provided services that have perhaps failed to deliver what the customer expected, or that have been found wanting for other reasons.

In recognition of the need for needs satisfaction as the primary experience an information technology user or consumer expects, the information provided in this chapter may help someone to avoid an unpleasant experience in respect of problem resolution.

In the open source software arena there are two support options: free support and paid-for (commercial) support.

47.1 Free Support

Free support may be obtained from friends, colleagues, user groups, mailing lists, and interactive help facilities. An example of an interactive dacility is the Internet relay chat (IRC) channels that host user supported mutual assistance.

The Samba project maintains a mailing list that is commonly used to discuss solutions to Samba deployments. Information regarding subscription to the Samba mailing list can be found on the Samba web[1] site. The public mailing list that can be used to obtain free, user contributed, support is called the `samba` list. The email address for this list is at `mail:samba@samba.org`. Information regarding the Samba IRC channels may be found on the Samba IRC[2] web page.

As a general rule, it is considered poor net behavior to contact a Samba Team member directly for free support. Most active members of the Samba Team work exceptionally long hours to assist users who have demonstrated a qualified problem. Some team members may respond to direct email or telephone contact, with requests for assistance, by requesting payment. A few of the Samba Team members actually provide professional paid-for Samba support and it is therefore wise to show appropriate discretion and reservation in all direct contact.

When you stumble across a Samba bug, often the quickest way to get it resolved is by posting a bug report[3]. All such reports are mailed to the responsible code maintainer for action. The better the report, and the more serious it is, the sooner it will be dealt with. On the other hand, if the responsible person can not duplicate the reported bug it is likely to be rejected. It is up to you to provide sufficient information that will permit the problem to be reproduced.

We all recognize that sometimes free support does not provide the answer that is sought within the time-frame required. At other times the problem is elusive and you may lack the experience necessary to isolate the problem and thus to resolve it. This is a situation where is may be prudent to purchase paid-for support.

[1] <https://lists.samba.org/mailman/>
[2] <http://www.samba.org/samba.irc.html>
[3] <https://bugzilla.samba.org/>

47.2 Commercial Support

There are six basic support oriented services that are most commonly sought by Samba sites:

- Assistance with network design
- Staff Training
- Assistance with Samba network deployment and installation
- Priority telephone or email Samba configuration assistance
- Trouble-shooting and diagnostic assistance
- Provision of quality assured ready-to-install Samba binary packages

Information regarding companies that provide professional Samba support can be obtained by performing a Google search, as well as by reference to the Samba Support[4] web page. Companies who notify the Samba Team that they provide commercial support are given a free listing that is sorted by the country of origin. Multiple listings are permitted, however no guarantee is offered. It is left to you to qualify a support provider and to satisfy yourself that both the company and its staff are able to deliver what is required of them.

The policy within the Samba Team is to treat all commercial support providers equally and to show no preference. As a result, Samba Team members who provide commercial support are lumped in with everyone else. You are encouraged to obtain the services needed from a company in your local area. The open source movement is pro-community; so do what you can to help a local business to prosper.

Open source software support can be found in any quality, at any price and in any place you can to obtain it. Over 180 companies around the world provide Samba support, there is no excuse for suffering in the mistaken belief that Samba is unsupported software — it is supported.

[4] <http://www.samba.org/samba/support.html>

Chapter 48

DNS AND DHCP CONFIGURATION GUIDE

48.1 Features and Benefits

There are few subjects in the UNIX world that might raise as much contention as Domain Name System (DNS) and Dynamic Host Configuration Protocol (DHCP). Not all opinions held for or against particular implementations of DNS and DHCP are valid.

We live in a modern age where many information technology users demand mobility and freedom. Microsoft Windows users in particular expect to be able to plug their notebook computer into a network port and have things "just work."

UNIX administrators have a point. Many of the normative practices in the Microsoft Windows world at best border on bad practice from a security perspective. Microsoft Windows networking protocols allow workstations to arbitrarily register themselves on a network. Windows 2000 Active Directory registers entries in the DNS namespace that are equally perplexing to UNIX administrators. Welcome to the new world!

The purpose of this chapter is to demonstrate the configuration of the Internet Software Consortium (ISC) DNS and DHCP servers to provide dynamic services that are compatible with their equivalents in the Microsoft Windows 2000 Server products.

This chapter provides no more than a working example of configuration files for both DNS and DHCP servers. The examples used match configuration examples used elsewhere in this document.

This chapter explicitly does not provide a tutorial, nor does it pretend to be a reference guide on DNS and DHCP, as this is well beyond the scope and intent of this document as a whole. Anyone who wants more detailed reference materials on DNS or DHCP should visit the ISC Web site at http://www.isc.org[1]. Those wanting a written text might also be interested in the O'Reilly publications on DNS, see the O'Reilly[2] web site, and the BIND9.NET[3] web site for details. The books are:

1. DNS and BIND, By Cricket Liu, Paul Albitz, ISBN: 1-56592-010-4
2. DNS & Bind Cookbook, By Cricket Liu, ISBN: 0-596-00410-9
3. The DHCP Handbook (2nd Edition), By: Ralph Droms, Ted Lemon, ISBN 0-672-32327-3

48.2 Example Configuration

The DNS is to the Internet what water is to life. Nearly all information resources (host names) are resolved to their Internet protocol (IP) addresses through DNS. Windows networking tried hard to avoid the complexities of DNS, but alas, DNS won. The alternative to DNS, the Windows Internet Name Service (WINS) — an artifact of NetBIOS networking over the TCP/IP protocols — has demonstrated scalability problems as well as a flat, nonhierarchical namespace that became unmanageable as the size and complexity of information technology networks grew.

WINS is a Microsoft implementation of the RFC1001/1002 NetBIOS Name Service (NBNS). It allows NetBIOS clients (like Microsoft Windows machines) to register an arbitrary machine name that the administrator or user has chosen together with the IP address that the machine has been given. Through the use of WINS, network client machines could resolve machine names to their IP address.

The demand for an alternative to the limitations of NetBIOS networking finally drove Microsoft to use DNS and Active Directory. Microsoft's new implementation attempts to use DNS in a manner similar to the way that WINS is used for NetBIOS networking. Both WINS and Microsoft DNS rely on dynamic name registration.

[1] <http://www.isc.org>
[2] <http://www.oreilly.com/catalog/dns/index.htm>
[3] <http://www.bind9.net/books-dhcp>

Microsoft Windows clients can perform dynamic name registration to the DNS server on startup. Alternatively, where DHCP is used to assign workstation IP addresses, it is possible to register hostnames and their IP address by the DHCP server as soon as a client acknowledges an IP address lease. Finally, Microsoft DNS can resolve hostnames via Microsoft WINS.

The following configurations demonstrate a simple, insecure dynamic DNS server and a simple DHCP server that matches the DNS configuration.

48.2.1 Dynamic DNS

The example DNS configuration is for a private network in the IP address space for network 192.168.1.0/24. The private class network address space is set forth in RFC1918.

It is assumed that this network will be situated behind a secure firewall. The files that follow work with ISC BIND version 9. BIND is the Berkeley Internet Name Daemon.

The master configuration file `/etc/named.conf` determines the location of all further configuration files used. The location and name of this file is specified in the startup script that is part of the operating system.

```
# Quenya.Org configuration file

acl mynet {
   192.168.1.0/24;
   127.0.0.1;
};

options {

   directory "/var/named";
   listen-on-v6 { any; };
   notify no;
   forward first;
   forwarders {
      192.168.1.1;
      };
   auth-nxdomain yes;
```

```
   multiple-cnames yes;
   listen-on {
      mynet;
      };
};

# The following three zone definitions do not need any modification.
# The first one defines localhost while the second defines the
# reverse lookup for localhost. The last zone "." is the
# definition of the root name servers.

zone "localhost" in {
   type master;
   file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
   type master;
   file "127.0.0.zone";
};

zone "." in {
   type hint;
   file "root.hint";
};

# You can insert further zone records for your own domains below.

zone "quenya.org" {
   type master;
   file "/var/named/quenya.org.hosts";
   allow-query {
      mynet;
      };
   allow-transfer {
      mynet;
      };
   allow-update {
      mynet;
      };
```

```
   };

zone "1.168.192.in-addr.arpa" {
   type master;
   file "/var/named/192.168.1.0.rev";
   allow-query {
      mynet;
   };
   allow-transfer {
      mynet;
   };
   allow-update {
      mynet;
   };
};
```

The following files are all located in the directory `/var/named`. This is the `/var/named/localhost.zone` file:

```
$TTL 1W
@               IN SOA  @   root (
            42              ; serial (d. adams)
            2D              ; refresh
            4H              ; retry
            6W              ; expiry
            1W )            ; minimum

      IN NS           @
      IN A            127.0.0.1
```

The `/var/named/127.0.0.zone` file:

```
$TTL 1W
@               IN SOA          localhost.  root.localhost. (
            42              ; serial (d. adams)
            2D              ; refresh
            4H              ; retry
            6W              ; expiry
```

```
            1W )            ; minimum

            IN NS           localhost.
1               IN PTR          localhost.
```

The /var/named/quenya.org.host file:

```
$ORIGIN .
$TTL 38400      ; 10 hours 40 minutes
quenya.org      IN SOA  marvel.quenya.org. root.quenya.org. (
            2003021832 ; serial
            10800      ; refresh (3 hours)
            3600       ; retry (1 hour)
            604800     ; expire (1 week)
            38400      ; minimum (10 hours 40 minutes)
            )
         NS      marvel.quenya.org.
         MX      10 mail.quenya.org.
$ORIGIN quenya.org.
frodo                   A       192.168.1.1
marvel                  A       192.168.1.2
;
mail                    CNAME   marvel
www                     CNAME   marvel
```

The /var/named/192.168.1.0.rev file:

```
$ORIGIN .
$TTL 38400      ; 10 hours 40 minutes
1.168.192.in-addr.arpa  IN SOA  marvel.quenya.org. root.quenya.org. (
            2003021824 ; serial
            10800      ; refresh (3 hours)
            3600       ; retry (1 hour)
            604800     ; expire (1 week)
            38400      ; minimum (10 hours 40 minutes)
            )
         NS      marvel.quenya.org.
$ORIGIN 1.168.192.in-addr.arpa.
```

```
1                       PTR     frodo.quenya.org.
2                       PTR     marvel.quenya.org.
```

The configuration files shown here were copied from a fully working system. All dynamically registered entries have been removed. In addition to these files, BIND version 9 will create for each of the dynamic registration files a file that has a `.jnl` extension. Do not edit or tamper with the configuration files or with the `.jnl` files that are created.

48.2.2 DHCP Server

The following file is used with the ISC DHCP Server version 3. The file is located in `/etc/dhcpd.conf`:

```
ddns-updates on;
ddns-domainname "quenya.org";
option ntp-servers 192.168.1.2;
ddns-update-style ad-hoc;
allow unknown-clients;
default-lease-time 86400;
max-lease-time 172800;

option domain-name "quenya.org";
option domain-name-servers 192.168.1.2;
option netbios-name-servers 192.168.1.2;
option netbios-dd-server 192.168.1.2;
option netbios-node-type 8;

subnet 192.168.1.0 netmask 255.255.255.0 {
   range dynamic-bootp 192.168.1.60 192.168.1.254;
   option subnet-mask 255.255.255.0;
   option routers 192.168.1.2;
   allow unknown-clients;
}
```

In this example, IP addresses between 192.168.1.1 and 192.168.1.59 are reserved for fixed-address (commonly called `hard-wired`) IP addresses. The

addresses between 192.168.1.60 and 192.168.1.254 are allocated for dynamic use.

Appendix A

GNU GENERAL PUBLIC LICENSE VERSION 3

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a

work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered

by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices

stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a The work must carry prominent notices stating that you modified it, and giving a relevant date.

b The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the

work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the

operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d Limiting the use for publicity purposes of names of licensors or authors of the material; or

e Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3)

arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty

for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it does.
Copyright (C) year name of author

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see http://www.gnu.org/licenses/.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
program Copyright (C) year name of author
This program comes with ABSOLUTELY NO WARRANTY; for details type '{\verb show w}
This is free software, and you are welcome to redistribute it
under certain conditions; type '{\verb show c}' for details.
```

The hypothetical commands '`show w`' and '`show c`' should show the ap-

propriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

GLOSSARY

Access Control List (`ACL`)

A detailed list of permissions granted to users or groups with respect to file and network resource access. See Chapter 16, "File, Directory, and Share Access Controls", for details.

Active Directory Service (`ADS`)

A service unique to Microsoft Windows 200x servers that provides a centrally managed directory for management of user identities and computer objects, as well as the permissions each user or computer may be granted to access distributed network resources. ADS uses Kerberos-based authentication and LDAP over Kerberos for directory access.

Common Internet File System (`CIFS`)

The new name for SMB. Microsoft renamed the SMB protocol to CIFS during the Internet hype in the nineties. At about the time that the SMB protocol was renamed to CIFS, an additional dialect of the SMB protocol was in development. The need for the deployment of the NetBIOS layer was also removed, thus paving the way for use of the SMB protocol natively over TCP/IP (known as NetBIOS-less SMB or "naked" TCP transport).

Common UNIX Printing System (`CUPS`)

A recent implementation of a high capability printing system for UNIX developed by <`http://www.easysw.com/`>. The design objective of CUPS was to provide a rich print processing system that has built-in intelligence capable of correctly rendering (processing) a file that is submitted for printing even if it was formatted for an entirely different printer.

Domain Master Browser (`DMB`)

The domain master browser maintains a list of all the servers that have announced their services within a given workgroup or NT domain. See Section 10.4.1 for details.

Domain Name Service (`DNS`)

A protocol by which computer hostnames may be resolved to the matching IP address/es. DNS is implemented by the Berkeley Internet Name Daemon. There exists a recent version of DNS that allows dynamic name registration by network clients or by a DHCP server. This recent protocol is known as dynamic DNS (DDNS).

Dynamic Host Configuration Protocol (`DHCP`)

A protocol that was based on the BOOTP protocol that may be used to dynamically assign an IP address, from a reserved pool of addresses, to a network client or device. Additionally, DHCP may assign all network configuration settings and may be used to register a computer name and its address with a dynamic DNS server.

Extended Meta-file Format (`EMF`)

An intermediate file format used by Microsoft Windows-based servers and clients. EMF files may be rendered into a page description language by a print processor.

Graphical Device Interface (`GDI`)

Device-independent format for printing used by Microsoft Windows. It is quite similar to what PostScript is for UNIX. Printing jobs are first generated in GDI and then converted to a device-specific format. See Section 22.4.1 for details.

Group IDentifier (`GID`)

The UNIX system group identifier; on older systems, a 32-bit unsigned integer, and on newer systems an unsigned 64-bit integer. The GID is used in UNIX-like operating systems for all group-level access control.

Internet Print Protocol (`IPP`)

An IETF standard for network printing. CUPS implements IPP.

Key Distribution Center (`KDC`)

The Kerberos authentication protocol makes use of security keys (also called a ticket) by which access to network resources is controlled. The issuing of Kerberos tickets is effected by a KDC.

NetBIOS Extended User Interface (`NetBEUI`)

Very simple network protocol invented by IBM and Microsoft. It is used to do NetBIOS over Ethernet with low overhead. NetBEUI is a nonroutable protocol.

Network Basic Input/Output System (`NetBIOS`)

NetBIOS is a simple application programming interface (API) invented in the 1980s that allows programs to send data to certain network names. NetBIOS is always run over another network protocol such as IPX/SPX, TCP/IP, or Logical Link Control (LLC). NetBIOS run over LLC is best known as NetBEUI (NetBIOS Extended User Interface — a complete misnomer!).

NetBT (`NBT`)

Protocol for transporting NetBIOS frames over TCP/IP. Uses ports 137, 138, and 139. NetBT is a fully routable protocol.

Local Master Browser (`LMB`)

The local master browser maintains a list of all servers that have announced themselves within a given workgroup or NT domain on a particular broadcast-isolated subnet. See Section 10.4.1 for details.

Printer Command Language (`PCL`)

A printer page description language that was developed by Hewlett-Packard and is in common use today.

Portable Document Format (`PDF`)

A highly compressed document format, based on PostScript, used as a document distribution format that is supported by Web browsers as well as many applications. Adobe also distributes an application called "Acrobat," which is a PDF reader.

Page Description Language (`PDL`)

A language for describing the layout and contents of a printed page. The best-known PDLs are Adobe PostScript and Hewlett-Packard

PCL (Printer Control Language), both of which are used to control laser printers.

PostScript Printer Description (`PPD`)

PPDs specify and control options supported by PostScript printers, such as duplexing, stapling, and DPI. See also Section 22.4.4. PPD files can be read by printing applications to enable correct PostScript page layout for a particular PostScript printer.

Remote Procedure Call (`RPC`)

RPCs are a means for executing network operations. The RPC protocol is independent of transport protocols. RPC does not try to implement any kind of reliability and the application that uses RPCs must be aware of the type of transport protocol underneath RPC. An RPC is like a programmatic jump subroutine over a network. RPCs used in the UNIX environment are specified in RFC 1050. RPC is a powerful technique for constructing distributed, client-server based applications. It is based on extending the notion of conventional, or local procedure calling, so that the called procedure need not exist in the same address space as the calling procedure. The two processes may be on the same system, or they may be on different systems with a network connecting them. By using RPC, programmers of distributed applications avoid the details of the interface with the network. The transport independence of RPC isolates the application from the physical and logical elements of the data communications mechanism and allows the application to use a variety of transports.

Server Message Block (`SMB`)

SMB was the original name of the protocol 'spoken' by Samba. It was invented in the 1980s by IBM and adopted and extended further by Microsoft. Microsoft renamed the protocol to CIFS during the Internet hype in the 1990s.

User IDentifier (`UID`)

The UNIX system user identifier; on older systems a 32-bit unsigned integer, and on newer systems, an unsigned 64-bit integer. The UID is used in UNIX-like operating systems for all user-level access control.

Universal Naming Convention (`UNC`)

A syntax for specifying the location of network resources (such as file shares). The UNC syntax was developed in the early days of MS DOS 3.x and is used internally by the SMB protocol.

SUBJECT INDEX

"Printers" folder, 499, 507, 518
../source/nsswitch, 586
.AppleDouble, 558
.PDS extension, 620
.ai, 465
.eps, 465
.pdf, 465
.profiles, 615
.ps, 465
.recycle, 556
/bin/false, 50, 303
/dev/null, 303
/dev/shadowvol, 560
/etc/cups/, 465
/etc/cups/mime.convs, 453, 454, 466, 478
/etc/cups/mime.types, 452, 454, 477, 478
/etc/fstab, 561
/etc/group, 48, 234, 237, 241, 287, 396, 568, 783
/etc/groups, 643
/etc/host.conf, 660, 662
/etc/hosts, 114, 163, 660, 662, 665, 745
/etc/hosts>, 661
/etc/inetd.conf, 587, 778
/etc/init.d/samba, 110, 582
/etc/init.d/samba.server, 584
/etc/init.d/smb, 582
/etc/krb5.conf, 112, 116, 293, 297
/etc/ldap.conf, 299, 302
/etc/logingroup, 783
/etc/mime.conv, 125
/etc/mime.types, 125
/etc/nsswitch.conf, 288, 291, 295, 301, 572, 573, 576, 577, 660, 663
/etc/openldap/slapd.conf, 35
/etc/openldap/sldap.conf, 223
/etc/pam.conf, 588, 640–642
/etc/pam.d, 575, 576, 586, 640
/etc/pam.d/, 573, 641
/etc/pam.d/ftp, 587
/etc/pam.d/login, 588
/etc/pam.d/samba, 586
/etc/passwd, 47, 48, 50, 81, 101–103, 106, 119, 122, 124, 190, 214, 218, 239, 287, 376, 568, 582, 639, 640
/etc/printcap, 448
/etc/resolv.conf, 660, 744, 745
/etc/samba, 699, 701, 745
/etc/samba/scripts, 257
/etc/samba/secrets.tdb, 109
/etc/samba/smb.conf, 3
/etc/samba/smbpasswd, 214
/etc/samba/smbusers, 260
/etc/shadow, 122, 190
/etc/smbpasswd, 214
/etc/ssl/certs/slapd.pem, 90
/etc/xinetd.d, 587
/etc/xinetd.d/telnet, 587
/export, 122
/lib/libnss_example.so, 573
/lib/libnss_files.so, 573
/lib/security, 586, 642
/lib/security/, 573
/opt/samba/bin, 733
/tmp, 319
/usr/bin/openssl, 737

/usr/lib/samba/vfs, 551
/usr/lib/security, 578, 586
/usr/lib/security/methods.cfg, 578
/usr/local/lib, 577
/usr/local/samba, 581
/usr/local/samba/bin, 582, 584, 733
/usr/local/samba/lib, 745
/usr/local/samba/lib/vfs, 551
/usr/local/samba/private/secrets.tdb, 109
/usr/local/samba/swat, 736
/usr/local/samba/var, 325, 744
/usr/local/samba/var/locks, 173
/usr/sbin, 733, 736
/usr/share/samba/swat, 736
/var/locks/*.tdb, 801
/var/log/samba, 744
/var/run/samba, 173
/var/spool/cups/, 538
/var/spool/samba, 124, 538
[global], 699
$, 103
%L, 699
%PDF, 465
%SystemRoot%\System32\config, 88
%i macro, 699
_kerberos.REALM.NAME, 112
_kerberos._udp, 117
_ldap._tcp, 117
_ldap._tcp.pdc._msdcs.quenya.org, 93
\\SERVER, 178
\\%L\%U\.profiles, 615
>Domain User Manager, 373
250-user limit, 215
3.0.11, 310
4,500 user accounts, 216
8.3 file names, 316

4294967295, 371

abbreviated keystrokes, 128
aborting shutdown, 307
accept connections, 365
Access, 306
access, 151
access authentication, 197
Access Control, 138
access control, 61, 65, 122, 324, 593
Access Control Entries, *see* ACE 364
Access Control List, 313
access control needs, 724
Access Controls, 722
access controls, 100, 198, 203, 250, 314, 320
access denied, 367
access policies, 212
access rights, 100, 246, 372
account, 7, 60, 106, 643
  backend, 62, 67
  database, 48
    backends, 44
account access controls, 191
account attributes, 290
account backends, 189
account containers, 220
account control block, *see* ACB 209
account control flags, 209, 210
Account Controls, 610
account controls, 101
account database, 214
account deleted, 207
account encode_bits, 209
account flag order, 210
Account Flags, 206
account flags, 209
account import/export, 203, 213
account information, 87, 101, 218, 258

account information database, 196
account management, 290
account name, 287, 303, 376
account policies, 72
account policy, 202
account restrictions, 609
account security, 203
account storage backends, 717
account storage mechanisms, 189
account storage system, 189
Account Unknown, 236
accountability, 203
accounts, 575
ACL, 150, 225, 236, 307, 364, 403, 803
ACLs, 313, 405
  File System, 318
  POSIX, 313, 315
  share, 314
  Windows, 315
ACLs on share, 325
ACLs on shares, 314
across network segments, 158
Active Directory, 91, 111, 247, 286, 295, 371
active directory, 44, 62–64, 72, 75
Active Directory Server, 722
AD4UNIX, 289
ADAM, 296
add a user account, 206
add client machines, 305
add domain users and groups to a local group, 311
add drivers, 396
add group script, 243, 249
add machine script, 83, 91, 104, 105, 118, 119, 139, 306, 715
add printer command, 432
Add Printer Wizard, 385, 395, 403
add printer wizard, 455
add share command, 272
add user script, 196, 203, 715
add/delete/change share, 307
adddriver, 415, 417, 420, 421, 509, 519
additional driver, 425
additional privileges, 307
addmem, 237
AddPrinterDriver(), 509
admin users, 323, 339
admincfg.exe, 792
administrative actions, 305
administrative duties, 246
administrative privileges, 236, 579
administrative responsibilities, 372
administrative rights, 307, 310
administrative rights and privileges, 311
Administrative Templates, 606
Administrator, 234, 238, 579
Administrator account, 115
administrator account, 106
administrator password, 117
Administrator%password, 109
Adobe, 456, 491, 526
Adobe driver, 500
Adobe driver files, 496
Adobe PostScript, 499, 534
Adobe PostScript driver, 507
Adobe PPD, 523
Adobe specifications, 475
ADS, 51, 66, 69, 99, 100, 107, 109, 111, *see* Active Directory 111, 112, 114–116, 154, 159, 160, 179, 189, 191, 198, 223, 246, 287, 288, 290, 296, 303, 371–373, 378, 403, 567, 574, 603, 607, 609,

611, 631, 639, 640, 660, 706, 716, 722, 817
ADS DC, 111
ADS domain, 287, 293
ADS domain members, 285
ADS manager, 115
ADS schema, 290
Advanced TCP/IP configuration, 129
advantages, 722
affect users, 604
affordable power, 688
AFPL, 461
AFPL Ghostscript, 470
AFS, 690
AIX, 197, 394, 578
algorithmic mapping, 290
alias group, 237
allow access, 364
allow trusted domains, 294
already exists, 118
alternative solution, 722
Amanda, 685
analyzes data, 753
anonymous, 124
  print server, 20
  read-write server, 20
anonymous access, 178
anonymous file server, 699
anonymous server, 697
ANSI compiler, 783
anticipate failure, 687
API, 215
Appliances, 570
application servers, 100
application/cups.vnd-postscript, 500
application/octet-stream, 454, 467, 477
application/pdf, 465, 466
application/postscript, 465, 466, 468, 500
application/vnd.cups-postscript, 468
application/vnd.cups-raster, 479
application/vnd.cups-raw, 454
application/x-shell, 467
apt-get, 560
ARCFOUR-HMAC-MD5, 116
architecture, 198
ARP/RARP, 661
ASCII, 466, 673, 676
ASCII text, 468
assign rights, 305
assigned RID, 235
assistance, 814
associations, 231
attach gdb, 753
attribute, 219
attributes, 717
audit file access, 552
audit module, 555
auth, 643
auth methods, 228, 717
authenticate, 100, 108
authenticate users, 108
authenticated, 112
authenticating server, 629
authentication, 44, 48, 64, 68, 77, 110, 195, 198, 213, 246, 586, 641
  backend, 107
authentication agents, 66
authentication architecture, 66
authentication backend, 723
authentication control, 575
authentication database, 372
authentication management, 573
authentication mechanisms, 575
authentication methods, 573
authentication module API, 578

authentication regime, 107
authentication reply, 111
authentication server, 87
authentication service, 587
authentication system, 64, 707
authenticatior, 67
authoritative, 181
authoritive, 163
authorization, 573
auto-reconnect, 195
autogen.sh, 774
autogenerated printcap, 400
automatic account creation, 104
automatic mapping, 288
automatic reconnects, 196
automatic redundancy, 167
autopoweruser.sh, 257
autotyping, 465
AUXILIARY, 218
auxiliary members, 237
availability, 687, 722
available, 124
available port, 437
available printerd, 396
available rights, 305
average print run, 456

b-node, 157
back up, 575
backed up, 724
backend, 690
backend authentication, 722
backend database, 68, 101, 119
backend failures, 692
backend file system pool, 690
backends, 152, 540
backup, 683, 724, 767
backup domain controller, 723
backup solution, 683
BackupPC, 684
bad hardware, 185
bad logon attempts, 208
Bad networking hardware, 185
bad password, 746
banner pages, 500, 501
barriers, 363
Batch Oplock, 346
BDC, 48, 50, 67, 68, 85, 86, 88–91, 93–97, 108, 110, 191, 196, 197, 215, 246, 286, 290, 579, 594, 723, 725
BDCs, 723
behavior approximately same, 706
between domains, 375
bias, 166
binary format TDB, 191
BIND, 819, 823
bind interfaces only, 169, 698, 699
BIND9, 160
BIND9.NET, 817
bindery-enabled, 640
block device, 319
block incoming packets, 366
BOBS, 685
bogus, 53
boot disk‘, 575
bridge, 313
bridges networks, 163
brlock.tdb, *see also* TDB 520
broadcast, 92, 157
broadcast address, 178, 748
broadcast isolated subnet, 167
broadcast messages, 158
broadcast messaging, 92
Broadcast node, 174
broadcast request, 77
broadcast traffic, 179
broadcast-based, 157
broadcast-based name resolution, 108

broadcasts, 166, 180
browsable, 389
browse across subnet, 179
browse list, 72, 156, 163, 170, 176, 180
browse list handling, 153
browse list maintainers, 163
browse list management, 80, 155
browse lists, 167, 179, 181, 182
browse resources, 178
browse server resources, 178
browse shares, 367
browse.dat, 178
browseable, 4, 389, 397, 399, 406
browser election, 165
browser elections, 166, 167
BrowseShortNames, 548
browsing, 77, 155, 167, 177
browsing across subnets, 153, 179
browsing another subnet, 177
browsing intrinsics, 163
browsing problems, 174, 183, 184
BSD, 81, 102, 385
BSD Printing, 388
BSD-style printing, 393
bug report, 814
bug reports, 761
Bugzilla, 761
built-in commands, 401
bypasses privilege, 306
byte ranges, 344
byte-range lock, 344
byte-range locking, 344, 345

c:\winnt\inf, 605
C:\WinNT\System32\config, 88
cached
    password, 54
cached encrypted password, 195
cached in memory, 196
cached local file, 345
cached locally, 618
cached references, 186
caching, 345, 346
caching reads, 347
caching scheme, 574
caching writes, 347
called name, 365
cannot join domain, 139
canonicalize files, 695
CAP, 675, 678, 789
cap-share, 679
CAP_LINUX_IMMUTABLE, 320
capability to delete, 320
case options, 695
case sensitive, 341, 617, 696
case sensitivity, 641
case-insensitive, 46, 388, 696
case-preserving, 46
central environment, 198
centralized
    authentication, 65
centralized identity management, 65
centrally managed, 641
certificate, 737
Certificate Authority, *see* CA 804
cfdisk, 560
challenge/response mechanis, 195
change capabilities, 202
change motivations, 721
change password, 117
change passwords, 201
change share command, 272
changed parameters, 706
changes password, 104
character device, 319
character set, 673
character sets, 674, 707
charset, 673

charset conversion, 675
chattr, 320
check for locks, 344
check logs, 118
checksum-search, 684
chmod, 124, 561
chown, 124, 328, 639
chpass, 102
CIFS, 117
CIFS function calls, 304
CIFS/SMB, 687, 689
Citrix, 596
classicalprinting, 487
clear purpose preferred, 683
clear-text, 55, 195, 225
clear-text passwords, 192
client client instructions, 127
Client for Microsoft Networks, 137
Client for Novell Networks, 618
client use spnego, 119
client-server mode, 201
client-side caching, 346
client-side data caching, 346, 349
clock skew, 114
cluster servers, 689
clustered file server, 688
Clustered smbds, 691
clustering technologies, 688
cluttering, 763
cmd, 185, 311
cmd shell, 311
CN, 90, 804
code maintainer, 814
codepages, 673
collating, 167
collisions, 800
color, 458
COM1:, 437
command-line, 245
command-line utility, 305
comment, 397, 399, 406
commenting out setting, 391
commercial Linux products, 313
commercial support, 813, 815
commit the settings, 131
Common Internet Filesystem, *see* CIFS 45
Common restrictions, 609
Common UNIX Printing System, *see* CUPS 385
common.adm, 605
comp.protocols.smb, 761
compatible, 193, 783
compile, 3
compile-time options, 392
complex file name space, 692
complex organization, 723
complexity, 122
compliance, 203
complicated, 179
complicated problem, 689
comprehensive documentation, 698
Computer Account, 106
computer account, 116
computer accounts, 189, 199, 304
Computer Management, 324, 326
Computer Name, 135
computer name, 138, 664
concurrent access, 345
Conectiva, 528
config.cache, 116
CONFIG.POL, 78
Config.POL, 604, 605
configuration
documentation, 8
configuration files, 731
configuration problem, 761
configuration syntax, 388
configuration techniques, 697
configuration too complex, 126

configuration tool, 731
configuration wizard, 135
configure, 774
configuring a firewall, 366
confirm address, 365
confirm the password, 377
confirm the trust, 376
connect transparently, 688
connection resources, 110
connections, 7
connections.tdb, *see also* TDB 520
consistent case, 696
console, 587
consumer expects, 813
container, 115
continuity of service, 692
contribute, 698
Control Panel, 135
controls, 364
convert
    domain member server, 69
converted, 192
copy'n'paste, 595
core files, 764
core graphic engine, 457
core values, 722
corrupted file, 288
cosine.schema, 219
country of origin, 815
CP850, 674
CP932, 676, 679
cracker, 365
create, 317
Create a Computer Account, 106
create a domain machine account, 107
create domain member, 106
create machine trust account, 109
create mask, 324, 332
create partition, 560
Create the Computer Account, 114
create user accounts, 122
create volume, 560
credentials, 46, 65, 90, 114, 135, 261, 304
credentials validation, 92
critical aspects of configuration, 127
crle, 577
cron, 94
cross post, 758
cross-segment browsing, 157
cross-subnet browsing, 163, 164, 172, 179
csc policy, 341
CUPS, 385, 386, 393, 394, 396, 447, 448, 462
    Page Accounting, 532
    quotas, 533
CUPS API, 4, 400
CUPS backends, 474
CUPS filtering, 463, 464
CUPS filtering chain, 472
CUPS libarary API, 125
CUPS PostScript, 499
CUPS PostScript driver, 500
CUPS print filters, 125
CUPS raster, 464, 470
CUPS-PPD, 525
cups.hlp, 497
cupsaddsmb, 455, 492, 497, 501, 503, 505–507, 511
cupsd.conf, 400, 448, 476, 538
cupsomatic, 463, 464, 475, 479–481, 523, 525
currupted, 767
custom scripts, 215
customer expected, 813
customers, 813
customized print commands, 401

daemon, 5, 371, 575, 779
daemon running, 581
daemons, 586
damaged data, 209
data caching, 346
data corruption, 185, 348
data interchange, 313
data stream, 387
database, 101, 706
DatabaseFS, 563
DAVE, 789
dbx, 764
DCE RPC, 579
DDK, 491, 495
DDNS, 159, 160, 660
de-multiplex, 689
de-multiplexing, 690
Debian, 560
Debian Sarge, 559
debug, 764
debug level, 753, 762, 799
debugging, 753, 763
debugging passwords, 753
debugging problems, 753
debuglevel, 762
dedicated heartbeat, 692
dedicated print server, 385
default accounts, 73
default aliases, 239
default behavior, 285
default case, 341
default devmode, 425
default DNS setup, 117
default gateways, 129
default groups, 239
default mapping, 152
default mappings, 250
default print command, 400
default print commands, 394
default printer, 402
default printing, 447
default profile, 627, 636
default settings, 211
default shells, 582
Default User, 631
default users, 239
defective hardware, 185
deferred open, 345
defined shares, 368
delegate administrative privileges, 261
delegated, 238
delegation, 372
delete, 317
delete a file, 320
delete printer command, 432
delete roaming profiles, 633
delete share command, 272
delete user script, 207
deleted files, 556
deleted parameters, 708
delmem, 237
demote, 68
demoted, 89
denial of service, 365
deny, 366
deny access, 366
deny modes, 344
deny-none, 345
DENY_ALL, 344
DENY_DOS, 344
DENY_FCB, 344
DENY_NONE, 344
DENY_READ, 344
DENY_WRITE, 344
deployment, 814
deployment guidelines, 199
DES-CBC-CRC, 112
DES-CBC-MD5, 112, 116
desirable solution, 311

desktop cache, 618
desktop profile, 74, 87
desktop profiles, 100, 270
deterents, 363
development libraries, 575
devfsd package, 560
device mode, 423
device-specific commands, 479
DFS, 381, *see* MS-DFS, Distributed File Systems 693
DFS junction, 381
DFS links, 381
DFS root, 382
DFS server, 381
DFS tree, 381, 382
DFS-aware, 381
DFS-aware clients, 382
DHCP, 128, 129, 131, 133, 179, 660, 817
DHCP servers, 198
DHCP-enabled, 131
DHCP-enabled operation, 128
diagnostic, 300
diagnostic tools, 753
diff, 765
differences, 684
different resources, 700
differently encrypted passwords, 192
differing protocol, 706
dir, 749
direct internet access, 363
directory, 91, 124, 290
directory access control, 236
directory access permissions, 313
directory controls, 313
Directory Information Tree, *see* DIT 198
directory mask, 324
directory permissions, 313, 320
directory schema, 290
directory security mask, 332, 333
Directory Separators, 316
directory server, 216
directory_mode, 556
disable LMB, 165
disable locking, 344
disable roaming profiles, 616
disabling oplocks, 349
disass, 764
disaster recovery, 724
disconnect a connection, 615
disk, 195
disk space, 724
disparate information systems, 65
display charset, 674, 676, 738
display PostScript, 458
displayName, 219
distort, 458
distribute authentication systems, 722
distributed, 62, 73
distributed account, 191
Distributed Computing Environment, *see* DCE 246
distributed directory, 107
distributed file system, *see* DFS 381, 688
Distributed File Systems, 690
distributed file systems, 690
distributed locking protocol, 692
distribution, 7, 110
dithering algorithm, 475
DMB, 72, 73, 79, 92, 163–165, 167, 168, 170, 174, 180–182
DMB for a workgroup, 177
DMC, 291
DMS, 49, 246, 291, 697
DN, 90, 804
DNS, 72, 92, 93, 111, 114, 128, 129, 131, 133, 154, 155,

158, 159, 162, 163, 179, 220, 572, 660, 668, 745, 817, 818
Active Directory, 160
Dynamic, 660, 819
SRV records, 160
DNS Configuration, 185
DNS lookup, 113
DNS name resolution, 108
dns proxy, 156, 744
DNS server, 183
DNS server access, 744
DNS server settings, 129, 131
DNS servers, 198
DNS zon, 112
DNS/LDAP/ADS, 176
document design, 724
documentation, 200, 731, 753
domain, 77, 206, 665
control, 45
role, 69
controller, 44, 48, 59, 62
convert, 68
hierarchy, 67
controllers, 48
groups, 247
master
browser, 72
member, 45, 62
server, 67, 86
member server, 62
security, 48
protocols, 60
trust account, 62
domain access, 286
domain account access policies, 212
domain admin group, 231
domain Administrator, 309
Domain Admins, 152, 234, 236, 238, 250, 306
Domain Admins group, 235
domain authentication, 246
domain context, 723
domain control, 67, 80, 96, 99, 287, 721
backup, 45
primary, 45
domain control database, *see* SAM 87
Domain Controller, 497
domain controller, 79, 86, 87, 91, 92, 100, 304, 569, 570, 579, 580, 604, 641, 723, 725
domain controllers, 75, 108, 607, 697, 707
domain environment, 193
domain global, 310
domain global group, 246, 311
domain global groups, 236
domain global user, 311
domain global users, 236
domain group, 567
domain group settings, 236
Domain Groups, 223
domain groups, 152, 231, 239
Domain Guests, 250
domain information, 718
domain join, 293, 579
domain joining, 134
domain logon, 74, 76, 89, 136
domain logon server, 618
domain logons, 72, 77, 177, 195
domain management tools, 104
domain master, 73, 76, 95, 156, 166, 167, 177
Domain Member, 289
joining, 49
domain member, 48, 71, 99, 100, 106, 118, 134, 195, 234, 246, 571, 723

Domain Member Client, *see* DMC 291
domain member client, 238
Domain Member Server, *see* DMS 291
domain member server, 96, 107, 159, 236, 287, 385
domain member servers, 96, 197, 236, 304
domain member workstations, 236
domain members, 100, 166, 575
domain membership, 70, 74, 99
domain name, 137
Domain Name System, *see* DNS 817
domain non-member, 571
domain policies, 605
domain radio button, 135
domain security, 60, 89, 99, 107, 110, 134, 193, 236, 290, 372, 579, 631
domain security account, 101
Domain Server Manager, 239
domain SID, 93, 270
domain trust, 86, 373
domain user, 138, 567, 570
domain user accounts, 247
Domain User Manager, 237, 239, 610
domain user manager, 203
Domain Users, 237, 250
domain users, 575, 587, 590
Domain Users group, 243
domain-level, 110
domain-level security, 110
domain-wide browse list, 168
DOMAIN<1B>, 79
DOMAIN<1C>, 78, 79
DOMAIN<1D>, 80
dont descend, 341
dos charset, 674, 676, 681
dos filemode, 324
dos filetime resolution, 341
dos filetimes, 341
draft, 475
Drive Identification, 316
driver, 389
driver CDROM, 411
driver download, 405
Driver File, 412
driver files, 411
Driver Path, 412
dual-daemon winbindd, 707
due diligence, 683
duplex, 482
duplex printing, 483
duplicate, 90
duplication of information, 569
DVI, 466, 468
Dynamic DNS, *see* DDNS 159, *see* DDNS 817
Dynamic Host Configuration Protocol, *see* DHCP 817
dynamic link loader, 577
dynamic registration files, 823
Dynamic SMB servers, 722
dynamically loadable library modules, 640

e-Directory, 67, 107
EAs, 318
economically wise, 687
eDirectory, 198
editreg, 610
efficient authentication, 641
election, 79, 163, 166
election criteria, 163
election packet, 166
election process, 166
EMF, 457, 485, 486

enable privileges, 304
enables clients to print, 388
enables NetBIOS over TCP/IP, 157
encapsulating, 157
encoding, 116
encryped password, 73
encrypt passwords, 108, 214, 650, 706, 750, 751
encrypted, 44, 54, 195
encrypted password, 196
encrypted passwords, 54, 190, 192, 193, 195, 196, 623, 716
encrypted session, 225
encryption, 52
encryption key, 106
encryption types, 112, 117
enforcing, 102
English, 676, 738
enhanced browsing, 156
Enhanced MetaFile, *see* EMF 457
enterprise, 215
enumdrivers, 411, 509
enumerate domain groups, 572
enumerate domain users, 572
EnumJobs(), 403
enumports command, 437
enumprinters, 509
environment variables, 402
EPM, *see* ESP meta packager 494
Epson Stylus, 482
Epson Stylus inkjet, 528
equivalence, 309
equivalent rights and privileges, 310
error message, 294, 417
error messages, 744
errors that can afflict, 138
ESC/P, 486
ESP, 461
    Ghostscript, 464, 465, 479
    meta packager, 494
    Print Pro, 483, 496
ESP Ghostscript, 464
established, 376
ethereal, 619, 754
Ethernet adapters, 365
EUC-JP, 675–677
eucJP-ms locale, 677
Event Viewer, 593
Everyone - Full Control, 323
Everyone group, 403
EVMS, 558
examples, 7
examples/LDAP, 191
execute, 319
existing LDAP DIT, 199
expands control abilities, 191
expired password, 208
explicit trust, 373
explicitly set, 391
exploit opportunities, 603
exploitation, 365
exported file system, 692
exposed, 366
extd_audit module, 555
Extended Attributes, 313
extended attributes, 320
Extended BSD Printing, 393
extended characters, 674
extended protocol, 175
extended SAM, 191
extra machine, 701

fail, 687
failed join, 292, 296
failed logins, 200
failover communication, 692
failover process, 692
failover servers, 692
fails, 118
failure, 294

failure semantics, 691
fake oplocks, 341
fake-permissions module, 626
fake_permissions, 74
fake_perms, 556, 626
fdisk, 560
Federated Identity Management, *see* FIM 65
federated organizations, 65
federated-identity, 65
Fiber Channel, 692
fickle, 154
fid, 689
file access permissions, 313
File Naming Conventions, 317
file ownership, 100
file serving, 385
File System, 315
    case sensitivity, 316
    feature comparison, 315
    UNIX, 315
    Windows, 315
file system capabilities, 320
FILE:, 437
filemanager, 178
filename mangling, 707
filter, 466
Filter Oplock, 346
FilterLimit, 476
filters, 466
FIM, 66, 198
firewall, 363, 365, 745
firewall active, 366
firewall setups, 366
fixed IP address, 129, 131
fixed IP addresses, 128
flush local locks, 345
flush name cache, 183
foomatic, 463, 464, 475, 480, 523, 524
Foomatic database, 528
Foomatic Printer, 475
Foomatic tutorial, 526
foomatic-rip, 464, 475, 479, 523, 526
Foomatic/cupsomatic, 481
force an election, 167
force create mode, 324, 332
force directory mode, 324, 333
force directory security mode, 324, 332
force election, 167
force group, 322, 323
force security mode, 324, 332
force unknown acl user, 277
force user, 322, 323, 339, 350
forced synchronization, 163
foreign domain, 196, 238
foreign SID, 570
foreign user, 571
FQDN, 804
framing error, 800
free support, 813, 814
FreeBSD, 81, 677
freezing, 561
French, 738
front-end virtual server, 689
frustrating experience, 199
FTP, 196
ftp, 684, 773
ftp access, 587
ftp service, 587
ftp services, 587
ftpd, 642
full rights, 236
functional components, 763
functionality, 722

gateway address, 129
gcc, 753, 783

gdb, 753, 764
GDI, 456, 457, 485, 486
general security service application programming interface, *see* GSSAPI 66
generic PostScript, 465
generic raster, 470
generic raster format, 464
genlogon.pl, 597
Gentoo, 800
Germany, 688
get, 749
getdriver, 411, 414
getdriverdir, 509
getent, 249, 295, 582
getent group demo, 238
gethostbyname() function call, 176
getpwnam, 218, 287
getpwnam() call, 715
GetSID.exe, 625
GhostScript, *see also* PostScript 459, 460
Ghostscript, 464, 479
  ESP, *see* ESP GhostScript 461
GID, 95, 102, 110, 117, 150, 152, 197, 232, 233, 246, 250, 285, 286, 288–290, 568, 571, 582
GID numbers, 288
GID range, 371
GIF, 466
global print command, 402
global right, 307
global section, 394
Global support, 722
global-level, 387
GNOME, 596
GNU Ghostscript, 464, 470
GNU GPL, 684
GNU tar, 685
GNU/Linux, 551
GPG, 773
GPL, 595
gpolmig.exe, 608
GPOs, 603, 606, 607, 609, 611, 631
grace time, 208
grant rights, 305
graphical objects, 458
graphically illustrated client configuration, 127
grayscale, 475
greater scalability, 191
greatest mistake, 126
grep, 577
group, 95, 150, 200, 319
  account, 73
  mapping, 63
group account, 238, 290
group accounts, 138, 199, 232, 235, 287
group management, 246, 247
group mapping, 150, 151, 231
group mappings, 152, 232
group membership, 236
group ownership, 567
group permissions, 151, 724
Group Policies, 603
group policies, 603
Group Policy, 605
group policy, 74
Group Policy Container, *see* GPC 607
Group Policy Editor, 605, 611, 729
Group Policy Objects, *see* GPO 603, 722
group policy objects, *see* GPOs 603
Group Policy Template, *see* GPT 607
group privileges, 234
group profiles, 626

group SID, 270
groupadd, 232, 241, 243
groupadd limitations, 241
groupdel, 232
groupmap, 231
groupmod, 232
grouppol.inf, 605
groups, 247, 603
    domain, 235
    mapping, 231
    nested, 236
groups of users, 261
growing, 561
GSSAPI, 66
gtklp, 528
guest, 73, 123
guest account, 124, 178, 184, 397, 402, 746
guest ok, 323, 397–399, 406
GUI, 447
Gutenprint, 471, 527

h-node, 157
harvesting password hashes, 225
hashed password equivalent, 192
headers files, 116
Heimdal, 112–114, 293, 640
Heimdal kerberos, 293, 297
help, 814
help command, 749
heterogeneous computing, 567
HEX, 675
hi-res photo, 475
hide dot files, 341
hide files, 341
hide unreadable, 324
hide unwriteable files, 324
high availability, 687
high order ports, 366
high-availability, 347
high-availability services, 692
high-speed server interconnect, 692
higher availability, 381
HKEY_CURRENT_USER, 629
HKEY_LOCAL_MACHINE, 606
holy grail, 567
home directories, 191, 582
home directory, 103, 215
home directory template, 587
home drive, 89
host msdfs, 381
host multiple servers, 699
host security, 363
host-based protection, 364
hostname, 114
hosts allow, 364, 399, 746
hosts deny, 364, 399, 746
house-keeping, 307
HOWTO documents, 199
HP JetDirect, 482
HP Photosmart, 528
HP-GL, 466
HP-GL., 468
hpgltops, 467
HPIJS, 527
HPUX, 394
http, 684
hybrid, 157
Hybrid node, 174

IANA, 470
ID mapping, 96, 707
ID mapping database, 574
ID range, 232
IDEALX, 217
Identification, 138
identify, 294
identity, 286
identity information, 66
identity management, 64, 107

centralized, 65
identity resolution, 567
IDMAP, 152, 232, 285, 286, 288, 294
idmap, 718
IDMAP backend, 197
idmap backend, 95, 96, 197, 288, 289, 568, 719
idmap GID, 718
idmap gid, 196, 232, 288, 289, 294, 568, 591, 656
IDMAP infrastructure, 285
idmap UID, 718
idmap uid, 196, 232, 288, 289, 294, 568, 591, 656
idmap_ad, 197
idmap_ldap module, 718
idmap_rid, 288, 294
IETF, 447
ifconfig, 779, 800
ignore connection, 365
imagetoraster, 471
immutible, 320
impersonate, 225
implementing oplocks, 350
Implicit Classes, 548
important announcements, 368
Imprints, 438
imprints, 455
include, 697
independent, 122, 698
individual domain user, 236
individual section, 394
inetd, 732, 747, 777
inetd.conf, 746, 748
inetorgperson.schema, 219
inf file, 411
infrastructure, 198, 570
inheritance, 320
inherits rights, 234
initdb.ldif, 36
initGroups.sh, 33, 242, 726
inktype, 475
insecure, 121, 364
inspire simplicity, 122
inspired structure, 688
install drivers, 385, 404
interactive help, 814
interdomain
trust
account, 62
trustrs, 44
interdomain connection, 377
interdomain trust, 375, 378
interdomain trust accounts, 189, 199
Interdomain Trusts, 371
Completing, 374
creating, 373
Facilities, 374
interdomain trusts, 246, 372
interface, 128
interface scripts, 203
interface-based exclusion, 364
interfaces, 168, 179, 365, 698, 699, 748, 779
intermediate information, 199
intermediate tools, 198
internal ordering, 731
internationalization support, 732
Internet, 364, 365
Internet Engineering Task Force, *see* IETF 447
Internet Printing Protocol, *see* IPP 447
Internet Protocol TCP/IP, 132
Internetworking Packet Exchange, *see* IPX 665
internetworking super daemon, 731

interoperability, 44, 62, 66, 285, 313, 567, 690
intolerance, 687
invalid shell, 50
invalid users, 322, 323, 746
IP address, 748
IP address automatically, 128, 131
IP addresses, 661
IP aliases, 129
IPC$, 78, 178, 366, 367
IPC$ connections, 689
ipchains, 745
ipconfig, 159
iPlanet, 107
IPP, 503
IPP client, 545
iptables, 745
IPX, 175
IRC, 814
IRIX, 551, 677
ISC
    DHCP, 817
    DNS, 817
ISC DHCP server, 128, 133
isolated workgroup, 164
IXFR, 660

Japanese, 675, 738
Japanese locale, 677
Japanese UNIX, 677
Java, 596, 678
JIS X 0208, 675
join, 292
join client, 307
join domain, 81
join the ADS domain, 109
join the domain, 107, 109
join the machine, 106
joined client, 102
Joined domain, 109
joining domain, 579
joining the domain, 109
JPEG, 466

KB 129202, 361
KB 224992, 361
KB 296264, 361
KB 811492, 360
KB 812937, 360
KDC, 111, 112, 114
KDE, 596
KDE konqueror, 595
KDE session, 595
KDEPrint, 447
Kerberos, 111, 114, 116, 117, 372, 572, 640, 716
    /etc/krb5.conf, 112
kerberos, 63, 112, 293
Kerberos authentication, 117
killall, 779
kinit, 112, 114–116
kixstart, 725
kprinter, 528
KRB, 293
KRB5, 113
krb5.conf, 113

LAN, 167, 596, 692, 753
LanMan, 63, 86, 101, 192, 402
LanMan logon service, 72
LanMan passwords, 190
LanManager, 46, 136, 137
LanManager-compatible, 170
LanManger password, 206
laptops, 684
large directory, 695
large domain, 295
large numbers of files, 695
large organizations, 372
last change time, 206

latency, 349
laws, 203
LCT, *see* last change time 206
LDAP, 62, 66, 67, 90, 94, 97, 98, 100, 107, 117, 189, 191, 192, 197–200, 216–218, 222, 239, 285, 287–290, 371, 372, 403, 568, 572, 640, 716, 723
  directories, 198
  master, 90
  server, 90
  slave, 85, 90
ldap admin dn, 93, 117
LDAP administration password, 93
LDAP administrative password, 117
LDAP backend, 122, 290, 706
LDAP backends, 197
LDAP database, 95, 220, 723
LDAP deployment, 198
LDAP directory, 198, 209, 216, 707
ldap group suffix, 718
LDAP idmap Backend, 197
ldap idmap suffix, 93, 117, 718
ldap machine suffix, 718
ldap passwd sync, 227
LDAP queries, 718
LDAP redirects, 289
ldap replication sleep, 91
LDAP schema, 152
LDAP server, 289
ldap ssl, 225
ldap suffix, 93, 718
ldap user suffix, 718
LDAP-based, 197, 371
LDAP., 199
LDAP/Kerberos, 706
LDAPS, 225
ldapsam, 73, 101, 189, 215–217, 239, 290, 718, 722
ldapsam_compat, 190, 718
ldapsearch, 717
LDAPv3, 225
ldconfig, 577
ldd, 448
LDIF, 222, 717
LDIF file, 220
legacy systems, 65
legal UNIX system account name, 119
Level1 Oplock, 345
Level1 oplock, 346
Level2 Oplock, 346
LGPL, 216
libcups, 400, 448
libcups.so, 448
libcups.so.2, 449
Liberty Alliance, 65
libiconv, 679
libnss_winbind, 237, 577
libnss_winbind.so, 573, 576
libnss_wins.so, 663
libraries, 113
licensing, 722
limitations, 372
linewidth, 458
link loader configuration, 577
Links
  hard, 317
  soft, 317
Linux, 197, 596, 639, 677
Linux High Availability project, 692
Linux LVM, 560
Linux LVM partition, 560
LinuxKongress2002, 526
Linuxprinting.org, 475, 522, 527
list of domain controllers, 108
listen for connections, 365
listen own socket, 698
LLC, 659

lm announce, 156
lm interval, 156
LM/NT password hashes, 214, 225
LMB, 72, *see* Local Master Browser 157, 163–168, *see* Local Master Browser 169, 170, 174, 175, 177, 180–182
LMHOSTS, 162, 163, 666
lmhosts, 170
load balancing, 381
load printers, 390, 391, 395
loaded modules, 551
loading printer drivers, 398
local
  groups, 247
  master
    browser, 72
local access permissions, 238
local accounts, 287
local administrative privileges, 236
Local Area Connection, 128
Local Area Connection Properties, 130
local authentication, 122
local authentication database, 122
local cache, 666
local disk, 684
local domain, 570
local group, 310
local groups, 236, 237, 287, 572
Local Machine Trust Account, 96
local master, 156, 165, 166
Local Master Browser, 157, 165, 169
local master browser, *see* LMB 163
local names, 158
local print driver, 405
local profile, 616, 618
local profiles, 613
local registry values, 607
Local security policies, 545
local smbpasswd file, 122
local spool area, 387
local subnet, 167
local system printing, 387
local UNIX groups, 246
local user, 286, 586
local user account, 196
local users, 287, 572
locale, 738
localhost, 365
locally known UID, 250
locate domain controller, 92
Lock caching, 345
lock directory, 698
lock password, 124
lock the account, 208
locking, 343, 344, 690
locking protocol, 343
locking semantics, 343, 344
locking.tdb, *see also* TDB 520
lockout, 53
log file, 556
log files, 744
  monitoring, 744
log level, 119, 178, 294, 555, 619, 753, 762
log.nmbd, 178, 748
logging, 555, 763
logical directories, 381
Logical Link Control, *see* LLC 665
logical volume, 561
Logical Volume Manager, *see* LVM 558
Login, 196
login, 587, 639
login id, 206
login name, 7
login shells, 198
LoginID, 287

logon, 48
logon authentication, 93
logon drive, 73, 620
logon home, 73, 226, 615, 616, 620, 623
logon name, 260
logon path, 73, 229, 615, 616, 618–620, 623
logon processing, 74
logon requests, 86, 92, 97
logon script, 73, 100, 229
Logon Scripts, 722
Logon scripts, 724
logon server, 78, 629
logons, 615
lookups, 215
loopback adapter, 747
loopback interface, 365, 786
lower-case, 46
lowercase filenames, 696
lp, 389, 548
lpadmin, 477, 484, 522, 533
LPD, 394
lpinfo, 474
lppause command, 449, 487, 539
lpq cache time, 396
lpq command, 396, 449, 539
lpresume command, 449, 539
lprm command, 449, 539
LPRNG, 394
lpstat, 4, 519
LPT1:, 437
LsaEnumTrustedDomains, 753
LTSP, 596
Lustre, 690
lvcreate, 560
LVM, 558, 561
LVM snapshots, 561
LVM volume, 560, 561
lvm10 package, 560
m-node, 157
MAC address, 661
MAC Addresses, 661
Mac OS X, 678
machine, 200
  account, 48
machine account, 62, 64, 67, 90, 209, 215, 303
machine account password
  change protocol, 109
machine accounts, 102, 199, 200, 303
machine accounts database, 87
machine authentication, 107
machine name, 661, 664
Machine Policy Objects, 722
machine SID, 270
Machine Trust Account, 100, 102–104, 106
  creation, 105
  password, 103
  UNIX account, 105
machine trust account, 64, 70, 79, 99, 106, 115, 118
  create privilege, 106
  creation, 102
  password, 70, 100
Machine Trust Accounts, 96, 100
  creating, 102
machine trust accounts, 97, 104, 118, 189, 199
machine_name, 103
machine_nickname, 103
Macintosh, 678
macros, 401
mail, 198
mailing list, 814
mailing lists, 814
maintaining ids, 707
major changes, 715

make, 663, 775
man, 731
man page, 579
man pages, 246
man-in-the-middle, 303
manage accounts, 203
manage drivers, 385
manage groups, 261
manage printers, 261
manage privileges, 304
manage roaming profiles, 613
manage share permissions, 325
manage share-level ACL, 239
manage shares, 261
manage users, 261
Manageability, 722
manageability, 722
managed by humans, 687
management bottleneck, 349
management costs, 198
management overheads, 64
management procedures, 65
management tools, 200
managing rights, 305
mandatory profiles, 625
Mandrake, 528
Mandriva, 528
manual UNIX account creation, 102
manual WINS server entries, 129, 132
manually configured, 134
manually configured DNS settings, 129
map, 106, 725
map to guest, 407, 432, 544
mapped, 238, 246, 250
mapping, 196, 250
mapping home directory, 368
mapping printer driver, 421
mappings, 117
maps UNIX users and groups, 117
master browser, 166
master browsers, 180
master server, 723
master smb.conf, 701
MasterAnnouncement, 181
match case, 695
max log size, 556
max print jobs, 396
max xmit, 799
maximum value, 371
mbd kept spawning, 801
Meccano set, 683
mechanism, 108
media type, 475
member, 70, 306
member machine, 236
memory, 195
messages.tdb, *see also* TDB 520
messaging systems, 198
Meta node, 174
meta-directory, 65
meta-service, 4
meta-services, 699
Microsoft Active Directory, 640
Microsoft Developer Network CDs, 754
Microsoft driver, 491
Microsoft management console, *see* MMC 63
Microsoft Remote Procedure Call, *see* MSRPC 571
Microsoft Windows 9x/Me, 104
Microsoft Wolfpack, 692
middle-ware, 198
migrate, 43, 721
migrate account settings, 725
migrate group, 725
migrate user, 725
migrating, 707

migration, 722
migration plan, 721
migration process, 721
MIME, 465–467, 478
    filters, 465
    raw, 23, 125, 454
MIME conversion rules, 464
MIME recognition, 464
MIME type, 454, 464, 468, 478
mime.types, 465
minimal
    configuration, 4
minimal configuration, 4
minimum security control, 121
misconfigurations, 8
misconfigured settings, 388
misinformation, 99
mission-critical, 347, 385
MIT, 112, 113, 293
MIT Kerberos, 640
MIT kerberos, 293, 297
mixed mode, 51, 378
mixed profile, 615
mkdir, 124, 561
mkfs.xfs, 561
MMC, 63, 232, 289, 290, 324, 326, 604, 611, 616
MMC snap-in, 607
modem/ISDN, 595
moderately secure, 363
modprobe, 560
module, 561
modules, 551, 552
more than one protocol, 174
mount, 47, 561
mouse-over, 595
moveuser.exe, 625
MS DCE RPC, 579
MS Windows 2000, 91
MS Windows NT4/200x, 191
MS Windows SID, 286
MS WINS, 154
MS-DFS, 693
MS-RPC, 402, 403
MS-WINS replication, 158
msdfs links, 382
msdfs root, 381
msg, 738
msg file, 738
MSRPC, 572
multibyte character sets, 707
multibyte charsets, 674
multiple backends, 214
multiple domains, 723
multiple hosting, 697
multiple modules, 552
multiple network interfaces, 168
multiple network segments, 723
multiple personality, 699
multiple server hosting, 698
multiple server personalities, 697
multiple servers, 697, 698
multiple universal naming convention provider, *see* MUP 611
multiple VFS, 552
multiple virtual servers, 699
multiple Windows workgroups or domains, 134
multiple WINS servers, 158
Multiuser databases, 349
mutual assistance, 814
mutually exclusive options, 156
My Network Places, 132, 178
Myrinet, 691

n security context, 163
n-memory buffer, 666
name conflict, 398
name lookup, 68, 666
name lookups, 159, 162

name registration, 92
name resolution, 155, 157, 163, 176, 183, 661, 744
name resolution across routed networks, 162
name resolve order, 156, 176
name service switch, *see* NSS 47
name-to-address, 171
name_type, 170, 175
nameserv.h, 174
native ACLs, 315
native dump, 685
native member, 69, 99
native mode, 51, 572
NBT, 665
nbtstat, 118, 666
necessary rights, 306
negotiate, 195
negotiating the charset, 674
nested group, 236
Nested Group Support, 311
nested groups, 236, 237
NET, 611
net, 200, 231, 245–247, 311
  ads, 247
    join, 109, 115, 266, 293
    leave, 267
    printer info, 281
    printer publish, 282
    printer remove, 282
    printer search, 282
    status, 267
    testjoin, 265
  getlocalsid, 150, 270, 717
  groupmap, 33, 150, 232, 234, 240, 726
    add, 250
    delete, 250
    list, 240, 250
    modify, 250
  localgroup, 311
  rap, 247
    session, 281
  rpc, 28, 49, 86, 247
    getsid, 93, 271
    group, 237, 248
    group add, 249
    group addmem, 254, 257
    group delete, 252
    group delmem, 254
    group list, 248
    group members, 255
    group rename, 253
    info, 283, 624
    join, 50, 109, 265, 579, 580, 726
    join bdc, 266
    join member, 266
    list, 305
    printer migrate drivers, 280
    printer migrate forms, 280
    printer migrate printers, 280
    printer migrate security, 280
    printer migrate settings, 280
    right list accounts, 275
    rights grant, 263, 306
    rights list, 262
    rights list accounts, 263
    share add, 272
    share delete, 273
    share migrate, 275
    share migrate all, 278
    share migrate files, 277
    share migrate security, 278
    testjoin, 265
    trustdom add, 268
    trustdom establish, 269, 377
    trustdom list, 267, 268
    trustdom revoke, 270
    user add, 259

user delete, 260, 267
user info, 260
user password, 259
user rename, 260
vampire, 150, 273, 726
setlocalsid, 271
time, 284
set, 284
system, 284
zone, 284
use, 116
net command, 707
net getlocalsid, 309
net groupmap, 718
net rpc user add, 307
net tool, 717
net use, 433
net use /home, 615
net use lpt1:, 507
net view, 395, 749
NetAtalk, 678
Netatalk, 789
netatalk, 558
NetBEUI, 659
NetBIOS, 48, 62, 92, 95, 154, 156, 159, 176, 659, 664–666
brooadcast, 72
name, 48
netbios alias, 699
netbios aliases, 699
NetBIOS broadcast, 109
NetBIOS disabled, 155
NetBIOS flags, 174
NetBIOS name, 79, 103, 104, 109, 664, 665, 699
netbios name, 698, 699
NetBIOS name cache, 118, 183
NetBIOS name length, 170
NetBIOS name resolution, 183
NetBIOS Name Server, *see* NBNS 669
NetBIOS name type, 163
NetBIOS names, 175, 663
NetBIOS network interface, 175
NetBIOS networking, 154
NetBIOS over TCP/IP, 154, 155, 159, 176, 179, 660
NetBIOS over TCP/IP disabled, 185
NetBIOS-less, 159, 699
NetBIOS-less SMB, 699
NetBIOSless SMB over TCP/IP, 158
NetBT, 665
NETLOGON, 72, 74, 604, 607, 609, 629, 631
Netlogon, 86
netlogon, 68
NetLogon service, 170
netlogon share, 94, 725
Netmon, 754
Netmon., 755
netmon.exe, 619
NetSAMLogon, 614
Netscape's Directory Server, 217
NetServerEnum2, 181
NetUserGetInfo, 78, 614
NetWare, 665
NetWare Bindery, 640
NetWare Core Protocol-based server, 640
NetWkstaUserLogon, 78
network
browsing, 62
logon, 73
service, 80, 86
performance, 67
wide-area, 88
network access controls, 313

network access profile, 87
network administrator, 313
network administrator's toolbox, 245
network administrators, 724
network analyzer, 753
network bandwidth, 167, 723
Network Basic Extended User Interface, *see* NetBEUI 665
Network Basic Input/Output System, *see* NetBIOS 154, *see* NetBIOS 665
Network Bridge, 128
Network Bridge Configuration, 128
network browsing problems, 167, 185
network client, 127, 285
network clients, 131
network configuration problems, 128
network difficulty, 127
network environment, 594
Network ID, 135
network interface, 365, 748
network logon, 77, 100, 137
network logon services, 79
network membership, 127
Network Monitor, 754
Network Monitor Tools and Agent, 755
Network Neighborhood, 155, 178, 181, 183, 395, 419
network neighborhood, 181
network policies, 604
network security, 723
network segment, 157, 163, 723
Network settings, 179
network sniffer, 195
network storage, 684
network traffic, 287
networked workstation, 572
networking advocates, 683
networking environment, 199
networking systems, 138
networks access, 801
Networks Properties, 137
new account, 376
new parameters, 709
newsgroup, 761
Nexus toolkit, 102
Nexus.exe, 63, 104, 593
NFS, 117, 197, 596, 690, 719
NFS clients, 348
NIS, 48, 93, 218, 287, 572
NIS database, 573
nmbd, 6, 8, 25, 29, 155–157, 177, 183, 292, 576, 582, 584, 698, 699, 748, 749, 753, 801
nmblookup, 666, 748
No NetBIOS layer, 159
no network logon service, 122
no printcap file, 125
nobody, 124
nobody account, 402, 700
node-type, 157
NoMachine, 596
NoMachine.Com, 594
non-authentication-based account management, 643
non-authoritative, 181
non-LDAP
backend, 86
non-member Windows client, 196
non-PostScript, 463, 479
non-PostScript printers, 468, 528
nonhierarchical, 372
nontransitive, 373
normal color, 475
normal user, 261
not domain member, 122
not domain members, 121

not part of domain, 172
not stored anywhere, 195
not transitive, 378
Novell, 107, 618
Novell eDirectory server, 640
NSS, 122, 197, 200, 216, 218, 223, 237, 285, 287, 295, 567–569, 571–573, 586, 590
nss_ldap, 95, 197, 200, 285, 289, 301, 302
nss_winbind.so.1, 576
nsswitch.conf, 47
nt acl support, 324, 328–330, 794
NT domain, 569
NT groups, 111, 239
NT migration scripts, 217
NT password, 206
NT Server Manager, 325
NT-controlled domain, 377
NT-encrypted password, 101
NT-encrypted passwords, 190
NT4, 286, 287
NT4 Domain, 286
NT4 domain, 287, 567
NT4 domain members, 285
NT4 style policy updates, 609
NT4 User Manager for Domains, 305
NT4-style, 378
NT4-style domain, 371
NT4-style domains, 373
Nt4sp6ai.exe, 606
NT_STATUS_LOGON_FAILURE, 715
NT_STATUS_UNSUCCESSFUL, 417
NTConfig.POL, 74, 100, 605–610, 628, 629, 728
ntconfig.pol, 605
ntdrivers.tdb, 424, *see also* TDB 520
ntforms.tdb, 424, *see also* TDB 520
NTFS, 150, 151, 315
ntlm_auth, 66
NTLMv2, 367
ntprinters.tdb, 424, *see also* TDB 520
NTUser.DAT, 610, 626, 729
NTuser.DAT, 620, 623, 725
NTUser.MAN, 626
NTuser.MAN, 620
null shell, 103
NX, 595

obey pam restrictions, 650, 651
object class, 717
object class declaration, 718
object module dependencies, 577
ObjectClass, 218
ObjectClasses, 218, 219
obtuse complexity, 724
office server, 23
OID, 218
old sambaAccount, 718
Omni, 527
on the fly, 106
on-the-fly, 290
on-the-fly logon scripts, 722
on-the-fly policy files, 722
one direction, 372
one domain, 286
one-way trust, 374
only one WINS server, 172
only user, 323, 369
OpenGFS, 690
OpenLDAP, 66, 90, 107, 152, 191, 217, 218, 640
OpenLDAP backend, 190
OpenSSL, 737, 804
operating costs, 722
operating system search path, 733

oplock, 690
oplock break, 345–347, 350
oplock break wait time, 350, 354
oplock contention limit, 350
oplock handling, 690
oplock mechanism, 350
oplock messages, 691
oplock parameters, 350
oplocks, 345–347
oplocks disabled, 349
oplocks management, 349
Opportunistic locking, 346
opportunistic locking, 343, 345
optional, 644
ordinary connection, 377
Organization for the Advancement of Structured Information Standards, *see* OASIS 65
organizational directory, 115
organizational unit, 115, *see* OU 607
os level, 73, 155, 165, 166, 168
os2 driver map, 791
OSS/Free Software, 596
other, 319
output duplexing, 468
outside threat, 364
own home directory, 368
ownership, 328
ownership cost, 722
ownership rights, 618

p-node, 157
package, 7
packages, 3
packet sniffer, 619
packet trace, 619
PADL, 197, 200, 289, 299
PADL Software, 197
page description languages, *see* PDL 456
page_log, 534
pager program, 388
paid-for support, 813
PAM, 122, 190, 197, 214, 216, 571, 573, 575, 576, 586, 590, 640, 641
PAM authentication module, 642
PAM configuration, 575
PAM management, 639
PAM module, 578
PAM modules, 640
PAM-capable, 641
pam-devel, 576
PAM-enabled, 570, 639, 640
PAM-specific tokens, 641
pam_krb5.so, 640
pam_ldap, 197
pam_ldap.so, 640
pam_mkhomedir, 587
pam_ncp_auth.so, 640
pam_pwdb.so, 640
pam_securetty.so, 588
pam_smb_auth.so, 641
pam_smbpass.so, 639, 640
pam_smbpasswd.so, 641
pam_unix.so, 588, 640
pam_unix2.so, 640
pam_userdb.so, 640
pam_winbind.so, 573, 586, 588, 641
parameters, 390
paranoid, 581
passdb, 96
passdb backend, 23, 73, 101, 152, 189, 192, 198, 201, 203, 207, 216, 228, 239, 287, 290, 309, 371, 640, 651, 706, 717, 718
passdb backends, 215, 707

passed across the network, 195
passwd, 124, 200–202, 572, 639
password, 87, 376, 377, 643
    plaintext, 79
password aging, 200
password assigned, 374
password backend, 123, 205
password backends, 189
password change facility, 736
password database, 94, 376
password encryption, 214
password expiration, 215
password expired, 208
password history, 64
password level, 55, 749, 792, 800
password management, 573
password prompt, 196
password scheme, 193
password server, 52, 53, 80, 108, 111, 112, 751
password uniqueness, 64
passwords, 569
patch, 765
path, 397, 399, 400, 405, 406, 408, 488, 538, 548, 749
path specified, 119
pauses, 801
PBM, 466
PCL, 456, 457, 484, 486, 489
pdb_ldap, 98
pdbedit, 32, 200, 202–204, 206, 207, 209, 210, 213, 214, 310, 611, 706, 717, 726, 729, 730
PDC, 48, 50, 67, 68, 79, 85, 86, 88–90, 92–95, 97, 100, 104, 108–111, 118, 164, 165, 191, 196, 215, 225, 226, 234, 246, 286, 287, 290, 376, 377, 505, 571, 573–575, 579–582, 594, 717, 723, 758, 801
PDF, 449, 457, 462, 466, 468, 483
pdf, 467
PDF distilling, 462
PDF filter, 125
pdftops, 466, 483
pdftosocket, 482
PDL, 456, 459, 461
PDM, 349
peer domain, 375
Peer node, 174
per-share access control, 323
performance, 695, 722
performance advantage, 343
performance degradation, 695
performance enhancement, 345
performance improvement, 349
performance-based, 215
performed as root, 306
perimeter firewall, 363
permanent changes, 729
Permanent name, 174
Permissions, 325
permissions, 368, 369, 561
    file/directory ACLs, 327
    share, 322
    share ACLs, 323
    UNIX file and directory, 314
permissions and controls, 314
PGP, 773, 774
phasing out NetBIOS, 156
Photo-CD, 466
physical locations, 381
physical network transport layer, 661
PID, 764
pid directory, 698
ping, 723, 745
pipe device, 319

PJL, 489, 500, 534
PJL-header, 534
plague network users, 128
plain-text
    passwords, 54
plaintext, 190
plaintext authentication, 190
plaintext password, 79, 97
plaintext passwords, 192, 193, 195
platforms, 783
PLP, 394
Pluggable Authentication Modules, *see* PAM 567, *see* PAM 639
PNG, 461, 466
PNM, 466
point 'n' print, 453, 501, 518
Point'n'Print, 385, 403, 404, 416
point'n'print, 455, 475, 507
Poledit, 607
poledit.exe, 604–607
Policies, 604, 609
policies, 728
Policy Editor, 605, 606
policy editor, 604
policy file, 609
policy files, 100
policy settings, 203
port 135, 169
Port 135/TCP, 366
port 137, 169, 748
Port 137/UDP, 366
port 138, 169
Port 138/UDP, 366
port 139, 169
Port 139/TCP, 366
port 445, 169
Port 445/TCP, 366
ports, 389, 754
POSIX, 94, 199, 223, 249
POSIX account, 204, 258
POSIX ACLS, 728
POSIX ACLs, 318, 320
POSIX identity, 199
POSIX locks, 691
POSIX semantics, 691
POSIX user accounts, 371
posixAccount, 218, 219
posixGroup, 219, 223
PostScript, 449, 456, *see also* Ghostscript 456, 457–459, 461, 462, 466, 468, 479, 482, 486, 489, 491, 493
    RIP, 459
PostScript driver, 415
PostScript interpreter, 459
PostScript Printer Description, *see* PPD 459
PostScript printers, 541
potential master browsers, 167
potential printer, 405
Power Users, 311
powerful, 67
PPD, 415, 459, 461, 463, 466, 477, 479, 489–491, 507, 534, 541
    CUPS, *see* CUPS-PPD 525
PPD-aware, 459
PPDs, 462, 475, 526
PPP, 365
precedence, 166
preferred master, 73, 156, 165–168, 751
prefilter, 471
prefilters, 468
preserve case, 617
primary domain controller, 701
primary group, 102
Primary Logon, 618
Primary WINS Server, 172
print, 389

queue, 4
spooler, 4
print accounting, 385
print command, 395, 399, 400, 402, 449, 487, 539, 540
print commands, 402
print configuration, 387, 389
print environment, 388
print filtering, 387
print job, 400, 402
print jobs, 395
print processing, 387
print queue, 404, 416, 421, 472
print quota, 456
print server, 124, 385
print service, 385
print spooling, 571
print spooling system, 447
print statistics, 456
print subsystem, 386, 399
print test page, 423
printable, 397–399
Printcap, 448
printcap, 4, 395, 396, 400, 448, 450, 453, 539
printcap name, 125, 396
PrintcapFormat, 448
printer admin, 307, 396, 399, 407, 410, 422, 424, 426, 428, 431, 450, 512, 547
printer attributes publishing, 707
printer default permissions, 403
printer driver, 404, 405, 449
printer driver data, 423
printer driver file, 405
printer driver files, 416
printer drivers, 404, 526
printer icon, 419
printer management, 246
printer management system, 447
printer migration, 246
printer monitor, 801
printer objects, 403
Printer Pooling, 437
printer queue, 403
printer share, 396
printer shares, 389, 395
printer$ share, 404
Printers, 395
printers, 4, 121
printers admin, 307
Printers and Faxes, 419
printers available, 155
printers section, 396
printing, 394, 395, 400–402, 448–450, 453, 539
printing behavior, 387
printing calls, 403
printing now, 801
printing support, 385, 386
printing system, 386
printing systems, 198
printing-related settings, 389
printing.tdb, 424, *see also* TDB 520
PrintPro, *see* ESP Print Pro 483
private dir, 698
private groups, 235
private key, 737
private network, 363
private networks, 365
private/MACHINE.SID, 93
private/secrets.tdb, 93
privilege, 238, 307
privilege management, 238, 261
privilege model, 304
privilege-granting applications, 641
privileged accounts, 305
privileges, 65, 106, 238, 239, 304, 307, 372, 403
privileges assigned, 305

problem report, 814
problem resolution, 813
problematic print, 387
Process data management, 349
professional support, 814
profile, 74, 78, 87, 191, 192
profile access rights, 626
profile contents, 623
profile directory, 618
profile migration tool, 626
profile path, 89, 618, 620
profile sharing, 623
ProfilePath, 619
Profiles, 604
profiles, 78
project, 814
promiscuous mode, 754
promote, 68, 69
promoted, 89
propagate, 86
Properties, 132, 137
protect directories, 320
protect files, 320
protection against attackers, 367
protocol stack settings, 131
provided services, 813
provisioned, 65
pstops, 468, 483, 534
pstoraster, 469, 470, 479, 481, 534
public, 398
publish printers, 403
publishing printers, 391
PulseAudio, 596
punching, 468
purchase support, 814
put, 749
pvcreate, 560

QNX, 394
qualified problem, 814
queue control, 394
queue resume command, 449
queuepause command, 449
quota controls, 198

RAID, 684
random machine account password, 109
range, 258
range of hosts, 364
RAP, 247
raster, 468, 528
raster driver, 464
raster drivers, 469, 470
raster image processor, *see* RIP 459
raster images, 458
rasterization, 469, 481
rastertoalps, 471
rastertobj, 471
rastertoepson, 471, 483
rastertoescp, 471
rastertohp, 471
rastertopcl, 471
rastertoprinter, 471
rastertosomething, 481
rastertoturboprint, 471
raw mode, 477
raw print, 506
raw printers, 448
raw printing, 23, 125, 452, 454
raw SMB, 62
raw SMB over TCP/IP, 159
rawprinter, 477
rcp, 684
rdesktop, 595
rdesktop/RDP, 595
read, 319
read directory into memory, 695
read list, 323
read only, 341, 398, 407, 556

server, 17
read raw, 799
read size, 798
Read-ahead, 345
read-only, 121, 122
read-only access, 290, 700
read-only files, 121
read-write access, 405
realm, 51, 93, 111, 114, 294, 297
rebooted, 136, 164
rebooting server, 307
recompiling, 698
reconfiguration, 89
record locking, 344
recycle, 556
recycle bin, 551
recycle directory, 556
recycle:exclude, 557
recycle:exclude_dir, 557
recycle:keeptree, 556, 557
recycle:maxsize, 557
recycle:noversions, 557
recycle:repository, 556
recycle:subdir_mode, 557
recycle:touch, 557
recycle:versions, 557
Red Hat Cluster Manager, 692
Red Hat Linux, 90, 105, 235
redirect, 95
redirection, 569
redirector, 345
redundancy, 158
reference documents, 122
refusing connection, 365
regedit.exe, 627
regedt32, 629
regedt32.exe, 611
register driver files, 417
register NetBIOS names, 162
registered, 170, 419
registers, 165
registry, 68, 192, 343, 604–606, 627
registry change, 195
registry keys, 627
registry settings, 609
regulations, 203
rejoin, 270
relationship password, 376
Relative Identifier, *see* RID 239
relative identifier, *see* RID 61, 215, *see* RID 215
Relative Identifiers, *see* RID 150
reliability, 62, 722
Remote Access Dial-In User Service, *see* RADIUS 641
remote announce, 157, 158, 162, 169, 178, 179
remote browse sync, 157, 158, 163, 170, 179
remote desktop capabilities, 594
remote desktop management, 594
remote domain, 373, 374, 376
remote login, 594
remote management, 245, 571
Remote Procedure Call, *see* RPC 246
Remote Procedure Call System Service, *see* RPCSS 611
remote profile, 618
remote segment, 170, 723
Remote X, 595
Remote X protocol, 595
remote-update protocol, 684
rename, 317
render, 452
rendering, 481
repeated intervals, 158
replicate, 94, 215
replicated, 44, 62, 91, 94, 607
replicated SYSVOL, 607

replication, 63, 90
  browse lists, 179
  SAM, 69, 86, 88, 93, 97
  WINS, 158, 172, 173
replication protocols, 172
repository, 287
requesting payment, 814
required, 644
requisite, 644
research, 683
resizing, 561
resolution, 475
resolution of NetBIOS names, 153
resolve NetBIOS names, 168
resolver functions, 573
resource failover, 692
resource kit, 608, 624
resource-based exclusion, 364
response, 295
restore, 767
restrict DNS, 176
reviewers, 697
revoke privileges, 306
RFC 1001, 818
RFC 1002, 818
RFC 1179, 394
RFC 2307, 197
RFC 2307., 218
RFC 2830, 803
rfc2307bis, 301
RFC2830, 90
RFCs, 753
rich database backend, 191
rich directory backend, 191
RID, 61, 103, 150, 235, 239, 288, 290, 294, 310, 574, 718
RID 500, 310
RID base, 290
right to join domain, 307
rights, 65, 77, 116, 304
rights and privilege, 261
rights and privileges, 238, 310
rights assigned, 304, 305
RIP, 479
rlogind, 642
Roaming Profile, 556
roaming profiles, 72, 613, 616, 619
rogue machine, 183
rogue user, 100
root, 106, 136, 304
root account, 304, 310
root preexec, 725
root user, 306
rotate, 458
RPC, 110, 567, 579, 614
RPC calls, 590, 689
RPC modules, 707
rpc.lockd, 344
rpcclient, 245, 411, 421, 519, 611
  adddriver, 501, 504, 508, 510, 511, 515
  enumdrivers, 508, 516
  enumports, 508
  enumprinters, 508, 512, 513, 516, 517, 519
  getdriver, 510, 511, 513, 516
  getprinter, 510, 513, 516, 519
  setdriver, 499, 501, 504, 508, 512, 516
rsh, 684
rsync, 94, 97, 197, 215, 684, 773
rsyncd, 684
runas, 426
rundll32, 425, 429, 518, 599

SAM, 63, 67, 87–89, 96, 97, 100, 150, 190, 196, 574
  delta file, 88
  replication, 69, 88
SAM backend, 197, 198

LDAP, 85
ldapsam, 86, 191, 197, 216
ldapsam_compat, 190
non-LDAP, 86
smbpasswd, 190, 214
tdbsam, 86, 191, 215
Samba 1.9.17, 171
Samba account, 103
Samba administrator, 575
Samba backend database, 119
Samba daemons, 110
Samba differences, 706
Samba mailing lists, 683
Samba private directory, 115
Samba SAM, 196
Samba SAM account, 119
Samba SAM account flags, 209
Samba schema, 191
Samba security, 363
Samba-2.2.x LDAP schema, 190
Samba-3-compatible LDAP backend, 706
Samba-PDC-LDAP-HOWTO, 217
samba-to-samba trusts, 371
samba-vscan, 563
samba.schema, 218, 219, 718
sambaDomain, 718
sambaGroupMapping, 718
sambaHomeDrive, 226
sambaHomePath, 226
sambaIdmapEntry, 718
sambaLogonScript, 226
SambaNTPassword, 225
sambaProfilePath, 226
SambaSAMAccount, 94, 200, 206, 207, 209, 216
sambaSAMAccount, 225
sambaSamAccount, 199, 218, 219, 223, 225, 226, 717
sambaSID, 152
sambaUNIXIdPool, 718
SambaXP conference, 688
samdb interface, 215
same domain/workgroup, 701
Sarbanes-Oxley, 202
scalability, 62, 85, 189, 215, 372
scalable, 198
scalable backend, 372
scalable coherent interface, *see* SCI 691
scale, 458
scanner module, 551
schannel, 84
schema, 301
schema file, 191
scp, 684
script, 119
scripted control, 245
scripts, 177, 199
SCSI, 692
SeAddUsersPrivilege, 262, 305, 307
SeAssignPrimaryTokenPrivilege, 308
SeAuditPrivilege, 308
SeBackupPrivilege, 262, 308
SeChangeNotifyPrivilege, 308
Seclib, 328
secondary controller, 723
SeCreateGlobalPrivilege, 308
SeCreatePagefilePrivilege, 308
SeCreatePermanentPrivilege, 308
SeCreateTokenPrivilege, 308
secret, 192
secrets.tdb, 93, 117, 222, *see also* TDB 520
section name, 3
secure, 121
secure access, 65
secure authentication, 304
secure communications, 225
secured networks, 363

security, 45, 48, 52, 56, 70, 71, 80, 107, 110–112, 363, 501, 543, 699, 716, 751, 792
  controllers, 48
  modes, 44
  settings, 8
security = user, 107
security account, 246
Security Account Manager, *see* SAM 67, *see* SAM 87
Security Assertion Markup Language, *see* SAML 65
security context, 107
security contexts, 373
security credentials, 290, 373
security domain, 373
security domains, 372, 373
security flaw, 368
security hole, 366
security identifier, *see* SID 61, 270
security level, 52
security levels, 45
security mask, 324, 332
Security Mode, 45
security mode, 43, 79
security modes, 45
security name-space, 285
security policies, 369
security settings, 707
security structure, 372
security vulnerability, 368
security-aware, 478
SeDebugPrivilege, 308
SeDiskOperatorPrivilege, 262, 305, 307
SeEnableDelegationPrivilege, 308
SeImpersonatePrivilege, 308
SeIncreaseBasePriorityPrivilege, 308
SeIncreaseQuotaPrivilege, 308
SeLoadDriverPrivilege, 308
SeLockMemoryPrivilege, 308
SeMachineAccountPrivilege, 262, 305, 307, 308
SeManageVolumePrivilege, 308
separate instances, 698
separate servers, 698
separate shares, 395
separate workgroups, 700
SePrintOperatorPrivilege, 262, 305, 307
SeProfileSingleProcessPrivilege, 308
SeRemoteShutdownPrivilege, 262, 305, 307, 308
SeRestorePrivilege, 262, 308
server failure, 689
Server Manager, 102, 104, 593, 594
Server Manager for Domains, 104
Server Message Block, *see* SMB 45
server pool, 690
Server Type, 44
  Domain Controller, 31
  Domain Member, 27, 96, 99
  Stand-alone, 17
server type, 246
  domain member, 49
Server Types, 286
server-mode, 56
service name, 7
service-level, 387, 395
services provided, 813
SeSecurityPrivilege, 308
SeShutdownPrivilege, 308
session, 643
session services, 62
session setup, 46, 52
sessionid.tdb, *see also* TDB 520
SessionSetupAndX, 287
SeSyncAgentPrivilege, 308
SeSystemEnvironmentPrivilege, 308
SeSystemProfilePrivilege, 308

SeSystemtimePrivilege, 308
set a password, 124
set group id, *see* SGID 319
set printer properties, 396
set user id, *see* SUID 319
SeTakeOwnershipPrivilege, 262, 305, 307, 308
SeTcbPrivilege, 308
setdriver, 509, 511
SetPrinter(), 509
setting up directories, 319
SeUndockPrivilege, 308
severely impaired, 159
SFU, 302
SFU 3.5, 290
SGI-RGB, 466
SGID, 319
shadow, 200
shadow copies, 560
shadow password file, 110
shadow utilities, 232
shadow_copy, 558, 561
shadow_copy module, 558
share, 4, 313, 394
share access, 323
share ACLs, 728
share management, 246
share modes, 690
Share Permissions, 326
share permissions, 325
share settings, 314
share stanza controls, 728
share-level, 45, 47, 381
share-level ACLs, 239
share-mode, 121
share-mode security, 79
share-mode server, 121
share_info.tdb, 325, *see also* TDB 520
shared secret, 100
shares, 155
shares and files, 575
Sharing, 325
shell scripts, 399
shift, 458
Shift_JIS, 675–677
short preserve case, 341, 617
Shortcuts, 317
shortcuts, 128, 618
show add printer wizard, 395, 432
show-stopper-type, 721
SID, 61, 82, 93, 95, 111, 117, 150–152, 192, 196, 200, 231, 233, 270, 285, 286, 288–290, 294, 304, 309, 568, 571, 624, 625, 698, 707, 725
SID management, 246
SID-to-GID, 232
SIDs, 729
signing, 84
simple access controls, 724
simple configuration, 7
simple guide, 706
Simple Object Access Protocol, *see* SOAP 65
simple operation, 191
simple print server, 123
simple printing, 388
simplest configuration, 7
simplicity, 121
Simplicity is king, 724
single DHCP server, 134
single repository, 189
single server, 689
Single Sign-On, 497
single sign-on, *see* SSO 60, *see* SSO 64, 99
single-byte charsets, 674

single-logon, 77
single-sign-on, 722
single-user mode, 575
slapadd, 222
slapd, 218
slapd.conf, 152, 219, 225
slapd.pem, 90
slapindex, 152
slappasswd, 222
slave servers, 723
slow browsing, 184
slow network, 800
slow network browsing, 186
slow performance, 801
smart printers, 448
SMB, 52, 119, 122, 154, 157, 176, 365, 403, 665, 684, 689, 691, 753
SMB encryption, 196
SMB locks, 691
SMB name, 664
SMB networking, 753
SMB Password, 640
SMB password, 202
SMB password encryption, 192
smb ports, 699
SMB printers, 545
SMB requests, 689
SMB semantics, 690
SMB Server, 641
SMB server, 195
SMB services, 690
SMB signing, 119, 707
SMB state information, 689
SMB-based messaging, 156
smb-cdserver.conf, 700
smb.conf, 699
SMB/CIFS, 92, 119, 195, 674
SMB/CIFS server, 214
smbclient, 117, 414, 415, 684, 747, 748, 754
smbd, 6–8, 25, 29, 215, 218, 222, 287, 292, 307, 389, 391, 555, 568, 576, 579, 582, 584, 691, 696, 698, 699, 753
smbgroupedit, 245
smbgrpadd.sh, 241
smbHome, 226
smbldap-groupadd, 249
smbldap-tools, 217
smbpasswd, 50, 73, 93, 94, 97, 101, 112, 117, 152, 189, 190, 200–203, 214–216, 218, 222, 287, 376, 611, 717
smbpasswd format, 205
smbpasswd plaintext database, 215
SMBsessetupX, 78
smbspool, 540, 541
smbstatus, 544, 764
SMBtconX, 78
smbusers, 365
SMS, 754
Snapshots, 560
sniffer, 79, 753
socket, 698
socket address, 698
socket options, 798
SOFTQ printing system, 395
Solaris, 197, 586, 596, 640, 677
Solaris 9, 584
source code, 7
space character, 243
special account, 303, 376
special section, 405
special sections, 394
special stanza, 405
specific restrictions, 323
Specify an IP address, 133

spinning process, 765
spool, 389
    directory, 4
spool files, 402
spooled file, 387
spooler., 4
spooling, 400, 452
    central, 452
    peer-to-peer, 452
spooling path, 389
spooling-only, 452
SPOOLSS, 402
SQL, 152
SQUID, 66
SRV records, 112, 113, 160
SRV RR, 660
SrvMgr.exe, 104
srvmgr.exe, 104
SRVTOOLS.EXE, 104, 594
SSH, 415, 596
ssh, 94, 97, 215, 684
SSL, 737
SSO, 64, 99, 198
stability, 722
stack trace, 764
stale network links, 186
stand-alone server, 286
standalone, 45, 70, 246, 287
standalone filter, 470
standalone server, 107, 121, 122, 206, 385, 723
standard confirmation, 373
stanza, 4, 696
stapling, 468
StartDocPrinter, 403
starting samba
    nmbd, 6, 25, 29
    smbd, 6, 25, 29
    winbindd, 6, 29, 568
startsmb, 779
StartTLS, 225
startup
    process, 7
startup script, 580
state, 689
state information, 688
state of knowledge, 687
static WINS entries, 173
status32 codes, 707
sticky bit, 319, 724
storage mechanism, 200
storage methods, 201
stphoto2.ppd, 482
strange delete semantics, 695
strict locking, 344
stripped of comments, 732
strptime, 209
stunnel, 737
su, 642
subnet mask, 129, 133, 748
subnets, 158, 164
subscription, 814
subsuffix parameters, 718
Subversion, 771, 772
successful join, 116
successful migration, 723
sufficient, 644
suffixes, 465
SUID, 319
Sun, 107
Sun ONE iDentity server, 640
Sun Solaris, 639
SUN-Raster, 466
support, 813
support exposure, 722
SVN
    web, 772
SVRTOOLS.EXE, 63
SWAT, 3, 731
swat, 9, 732, 733, 736

enable, 735
security, 737
SWAT binary support, 732
swat command-line options, 733
SWAT permission allowed, 736
symbolic links, 382
synchronization, 67, 80, 170, 181
synchronization problems, 569
synchronize, 94, 114, 170, 181
synchronized, 93
syntax tolerates spelling errors, 388
system access controls, 191
system accounts, 203
system administrator, 304
system groups, 250
system interface scripts, 304
system policies, 604
System Policy Editor, 604, 607, 627
system security, 238
system tools, 683
SYSV, 394
SYSVOL, 607

tail, 744
Take Ownership, 328
take ownership, 307
tape, 724
tar, 684
tarball, 7
tattoo effect, 729
TCP, 169, 689
TCP data streams, 689
TCP failover, 688
TCP port, 62
TCP port 139, 660, 699
TCP port 445, 660, 699
tcp ports, 580
TCP/IP, 128, 133, 154, 175
TCP/IP configuration, 129, 132
TCP/IP configuration panel, 130
TCP/IP protocol configuration, 127
TCP/IP protocol settings, 128, 130
TCP/IP protocol stack, 171
TCP/IP-only, 175
tcpdump, 754
TDB, 191, 421, 520, 698
backing up, *see* tdbbackup 521
tdb, 574, 691, 767
tdb data files, 715
TDB database, 417
TDB database files, 424
tdb file backup, 715
tdb file descriptions, 5, 715
tdb file locations, 5
tdb files, 325
tdbbackup, 521, 801
tdbdump, 325
tdbsam, 73, 152, 189, 192, 205, 215, 216, 239, 287, 722
tdbsam databases, 214
technical reviewers, 697
Telnet, 196
telnet logins, 587
template, 626
template homedir, 587
temporary location, 399
Terminal Server, 689
terminal server, 596
Testing Server Setup, 114
testparm, 8, 124, 388–391, 393, 744, 754
tethereal, 754
text/plain, 467
texttops, 467
thin client, 596
ThinLinc, 596
tid, 689
TIFF, 466
TightVNC, 595, 596
time difference, 114

time format, 209
time-to-live, *see* TTL 173
tool, 326
tools, 123, 199
tools\reskit\netadmin\poledit, 605
traditional printing, 401
training course, 683
transfer differences, 684
transformation, 466
transitive, 373
transparent access, 100
transparently reconnected, 688
transport connection loss, 347
Transport Layer Seccurity, TLS
    Configuring, 804
    Introduction, 803
transport layer security, *see* TLS 90
Transport Layer Security, TLS
    Testing, 809
    Troubleshooting, 811
trigger, 70, 88
Trivial Database, 767
trivial database, 191, *see* TDB 215
troubleshoot, 390
troubleshooting, 541
Tru64 UNIX, 677
trust, 60, 200
    account, 48
trust account, 48, 209, 378
    interdomain, 62
    machine, 64
trust account password, 86
trust accounts, 199, 246
trust established, 374
trust relationship, 373–375, 378
trust relationships, 371–373, 707
trusted, 181, 303
trusted domain, 238, 372, 374, 377, 572
trusted domain name, 376
trusted party, 376
trusting domain, 372, 374
trusting party, 376
trusts, 371, 372
TTL, 173
turn oplocks off, 350
turnkey solution, 199
two-up, 482
two-way
    propagation, 86
two-way trust, 373, 374

UCS-2, 676
UDP, 72, 157, 162, 167, 169, 179
UDP port 137, 660
udp ports, 580
UDP unicast, 162
UID, 95, 101, 103, 110, 117, 192, 196, 197, 200, 206, 232, 246, 250, 258, 285, 286, 288–290, 304, 568, 571, 582
uid, 219
UID numbers, 288
UID range, 371
unauthorized, 100
unauthorized access, 313
UNC notation, 412
unexpected.tdb, *see also* TDB 520
unicast, 157
Unicode, 674, 707
unicode, 674
Unicode UTF-8, 678
unified logon, 569
UNIX, 677
    server, 44
UNIX account, 101, 103, 105
unix charset, 674, 676, 679, 680
UNIX Domain Socket, 319
UNIX domain socket, 571

UNIX file system access controls, 314
UNIX group, 250
UNIX groups, 231, 569
UNIX home directories, 368
UNIX host system, 304
UNIX ID, 574
UNIX locking, 344
UNIX login ID, 101
UNIX permissions, 728
UNIX printer, 395
UNIX printing, 386
UNIX system account, 119
UNIX system accounts, 304
UNIX system files, 683
UNIX user identifier, *see* UID 101
UNIX users, 110, 569
UNIX-style encrypted passwords, 192
UNIX-user database, 122
UNIX/Linux group, 235
UNIX/Linux user account, 258
unlink calls, 556
unlinked, 319
unmapped groups, 150
unmapped users, 150
unprivileged account names, 123
unsigned drivers, 545
unstoppable services, 688
unsupported encryption, 116
unsupported software, 815
updates, 368
upload drivers, 385
uploaded driver, 395
uploaded drivers, 404
uploading, 404
upper-case, 46
uppercase, 114, 119, 695, 696
uppercase character, 243
USB, 482
use client driver, 396, 453, 505
use computer anywhere, 673
user, 47, 150, 200, 319, 750
user access management, 100
user account, 199, 203, 209, 214, 258
    Adding/Deleting, 201
user account database, 88
User Accounts
    Adding/Deleting, 202, 223
user accounts, 199, 287, 303
user and group, 570
user and trust accounts, 189
user attributes, 215
user authentication, 571
user database, 93, 214
user encoded, 270
user groups, 814
user logons, 303
User Management, 202, 223
user management, 201, 246, 247
User Manager, 376, 377, 593, 626
User Manager for Domains, 594
user or group, 305
user profiles, 618
User Rights and Privileges, 309
user-level, 45, 46
User-level access control, 138
user-level security, 196
user-mode security, 79
user.DAT, 617, 623
User.MAN, 626
user.MAN, 617
useradd, 102, 105
username, 87, 323
username and password, 135
username map, 106, 259, 260
username-level, 55
userPassword, 222
users, 369, 603

UsrMgr.exe, 104
UTF-8, 674, 676, 677
UTF-8 encoding, 738

valid username/password, 367
valid users, 322, 323, 746, 749
validate, 8, 743
validate every backup, 724
validation, 64, 603
vendor-provided drivers, 452
verifiable, 181
verify, 390
version control, 558
veto files, 341
VFS, 74, 552
VFS module, 558, 626
VFS modules, 551, 563
vfs objects, 551
vgcreate, 560
vgdisplay, 560
vipw, 81, 102
Virtual File System, *see* VFS 551
virtual server, 689, 692
virus scanner, 551
Visual Studio, 491
vital task, 687
VNC/RFB, 595
volume group, 560
volunteers, 758
vscan, 563
vuid, 689

W32X86, 411, 412, 491, 497
W32X86/2, 463
WAN, 167, 349
wbinfo, 581
Web-based configuration, 731
WebClient, 185
Welcome, 136
well known RID, 310
well-controlled network, 724
well-known RID, 239
wide-area network bandwidth, 641
win election, 167
Win32 printing API, 403
WIN40, 412, 414, 497
Winbind, 122, 570, 572–576, 578, 581, 587, 590, 639, 641
winbind, 110, 237, 287–289, 291, 292, 371, 372, 567, 579
Winbind architecture, 707
Winbind hooks, 569
winbind separator, 581
Winbind services, 580
Winbind-based authentication, 639
winbind.so, 590
winbindd, 6, 8, 29, 95, 96, 152, 200, 232, 237, 258, 286, 287, 371, 568, 571, 575–577, 579–581, 584, 586, 698
winbindd daemon, 582
Windows, 285, 678
Windows 2000, 112, 116, 154, 373
Windows 2000 Professional TCP/IP, 130
Windows 2000 server, 378
Windows 2003, 114, 119
Windows 200x/XP, 158, 386
Windows 9x/Me, 136, 172, 175, 593
Windows 9x/Me/XP Home, 100
Windows account management, 569
Windows client, 310
Windows client failover, 347
Windows domain, 715
Windows Explorer, 178, 412
Windows group, 231, 235, 250, 303
Windows group account, 310
Windows groups, 250
Windows Internet Name Server, *see* WINS 669

Windows Logon, 618
Windows Me TCP/IP, 132
Windows Millennium, 132
Windows Millennium edition (Me) TCP/IP, 132
Windows network clients, 154
Windows NT domain name, 137
Windows NT PostScript driver, 541
Windows NT Server, 376
Windows NT/2000/XP, 419
Windows NT/200x, 172, 575
Windows NT/200x/XP, 396
Windows NT/200x/XP Professional, 100, 134, 139
Windows NT3.10, 86
Windows NT4, 325, 386
Windows NT4 domains, 374
Windows NT4 Server, 375
Windows NT4/200X, 199
Windows NT4/200x, 234
Windows NT4/200x/XP, 92, 239, 325
Windows NT4/2kX/XPPro, 303
Windows PPD, 524
Windows privilege model, 304
Windows Registry, 100
windows registry settings, 618
  default profile locations, 630, 632
  profile path, 618
  roaming profiles, 616
Windows Resource Kit, 617
Windows Security Identifiers, *see* SID 285
Windows Terminal Server, 596
Windows Terminal server, 594
Windows user, 303
Windows user accounts, 258
Windows workstation., 311
Windows XP Home, 195
Windows XP Home Edition, 631
Windows XP Home edition, 63, 76, 137
Windows XP Professional, 128, 386
Windows XP Professional TCP/IP, 130
Windows XP TCP/IP, 128
Windows95/98/ME, 419
winnt.adm, 605
WINS, 62, 68, 72, 92, 108, 123, 129, 131–134, 153–159, 162, 163, 165, 168, 170, 171, 176, 177, 179–181, 183, 669, 818
wins, 663
WINS Configuration, 185
wins hook, 156
WINS lookup, 109
wins proxy, 156
WINS replication, 173, 174
WINS Server, 156
WINS server, 162–164, 168, 172, 177, 185
wins server, 156, 171, 172
WINS server address, 162
WINS server settings, 133
WINS servers, 171
WINS service, 172
WINS Support, 156
wins support, 156, 171, 172
wins.dat, 173
without Administrator account, 310
without ADS, 722
work-flow protocol, 65
workgroup, 53, 70, 77, 80, 108, 138, 164, 167, 177, 665, 698, 699
  membership, 70
workstations, 192
world-writable, 319

writable, 398, 399
write, 319
write access, 320
Write caching, 345
write changes, 290
write list, 323, 407
write permission, 115
write raw, 799, 800
writeable, 556
WYSIWYG, 457

X Window System, 457
X.509 certificates, 803
XFS file system, 559
xfsprogs, 560
xinetd, 732, *see* inetd 747, 778
XML, 152
XML-based datasets, 528
xpp, 528
Xprint, 457
xxxxBSD, 639

yppasswd, 201, 202

Zero Administration Kit, 606
zero-based broadcast, 168