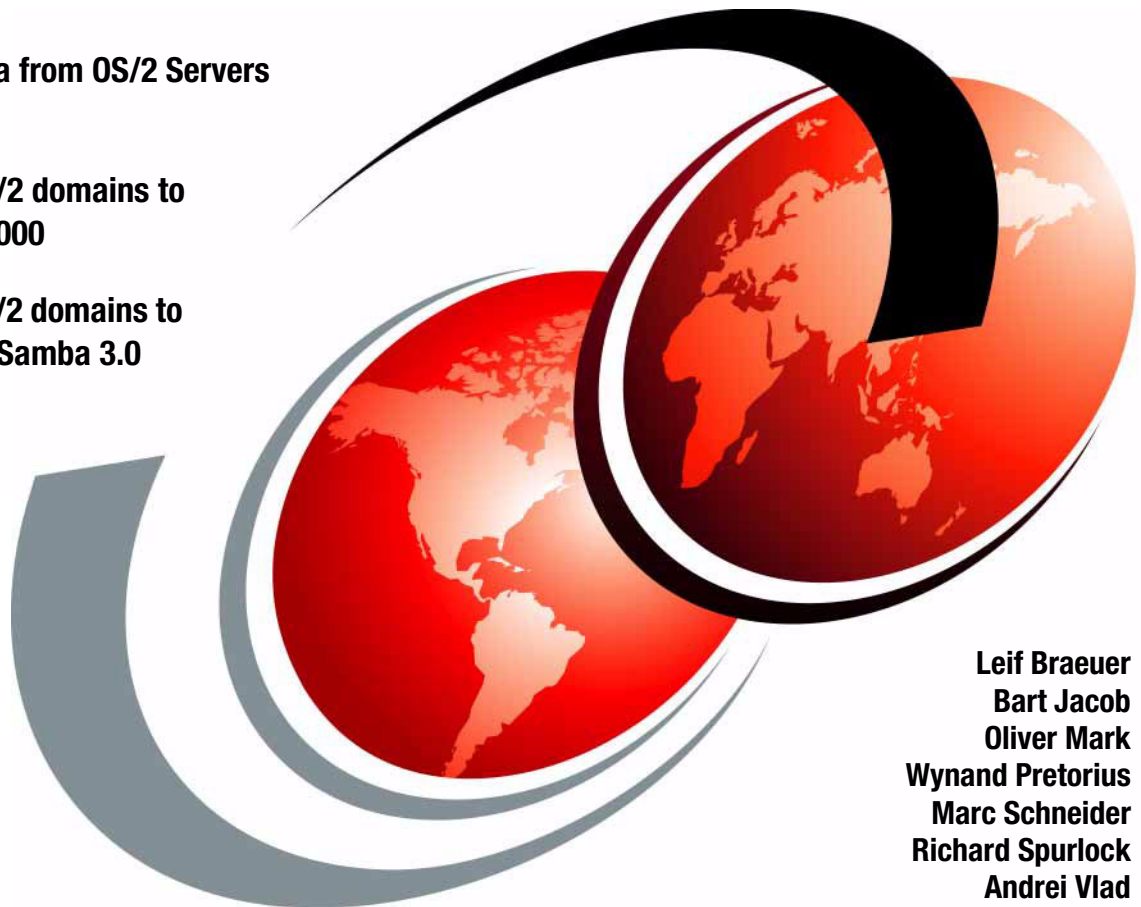


OS/2 Server Transition

Extract data from OS/2 Servers

Migrate OS/2 domains to
Windows 2000

Migrate OS/2 domains to
Linux with Samba 3.0



Leif Braeuer
Bart Jacob
Oliver Mark
Wynand Pretorius
Marc Schneider
Richard Spurlock
Andrei Vlad



International Technical Support Organization

OS/2 Server Transition

October 2003

Note: Before using this information and the product it supports, read the information in “Notices” on page xvii.

First Edition (October 2003)

© Copyright International Business Machines Corporation 2003. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	xv
Notices	xvii
Trademarks	xviii
Preface	xix
The team that wrote this redbook	xxi
Become a published author	xxiv
Comments welcome	xxiv
Part 1. Introduction and preparation	1
Chapter 1. OS/2 Server environment	3
1.1 IBM OS/2 Warp Server for e-business base installation	4
1.2 Sample domain	5
1.3 Configured TCP/IP-based services	6
1.4 Product stack on OS/2	7
1.4.1 IBM Universal Database	7
1.4.2 IBM e-Network Communications Server	7
1.4.3 Lotus Domino Server	8
1.4.4 IBM HTTP Server	8
1.4.5 IBM Tivoli Storage Manager Client	9
1.4.6 IBM LAN Distributed Platform	9
1.4.7 IBM WebSphere MQ	10
1.4.8 IBM Netfinity® Manager™	11
1.5 Recommended steps prior to migration	11
1.5.1 General architectural thoughts	11
1.5.2 Security	12
1.5.3 Virus protection	12
1.5.4 Printer migration	13
1.5.5 Transport protocol migration	13
1.6 Summary	17
Chapter 2. Target platforms	19
2.1 Windows 2000 as a target platform	20
2.1.1 Base installation	20
2.1.2 FTP server	22
2.1.3 DHCP server	23
2.1.4 WINS server	23

2.1.5	DNS server	23
2.1.6	Active Directory services	24
2.1.7	Certificate service	25
2.2	Software stack on Windows 2000	26
2.2.1	IBM Universal Database	26
2.2.2	IBM e-Network Communications Server	27
2.2.3	Lotus Domino	27
2.2.4	IBM HTTP Server	28
2.2.5	Tivoli Storage Manager Client	28
2.3	Red Hat Linux as a target	28
2.3.1	Base installation	29
2.3.2	FTP server	33
2.3.3	NFS server	34
2.3.4	DNS server	34
2.3.5	DHCP server	35
2.3.6	Samba server	35
2.4	SuSE Linux as a target	36
2.4.1	Base installation	36
2.4.2	FTP server	41
2.4.3	NFS server	41
2.4.4	DNS server	42
2.4.5	DHCP server	43
2.4.6	Samba server	43
2.5	Software stack on Linux	43
2.5.1	IBM Universal Database	44
2.5.2	IBM Communication Server	44
2.5.3	Lotus Domino	49
2.5.4	IBM HTTP Server	50
2.5.5	Tivoli Storage Manager (TSM) Client	51
2.6	Samba and OpenLDAP	51
2.6.1	Environment overview	51
2.6.2	Downloading products	52
2.6.3	Decompressing and extracting products	53
2.6.4	Configuring and compiling products	53
2.6.5	Configuring products	56
2.7	Summary	61
Chapter 3. Starting the OS/2 Server migration		63
3.1	Introduction	64
3.2	LSMT package	64
3.2.1	Install LSMT package	65
3.2.2	REXX and LAN server functions	66
3.2.3	LSMT INI files	66

3.2.4	LSMT ASCII file	67
3.3	Collecting data using LSMT	68
3.3.1	Domain	68
3.3.2	Servers	69
3.3.3	Groups	72
3.3.4	Users	74
3.3.5	Access	77
3.3.6	File and printer shares	78
3.3.7	Serial devices	79
3.3.8	Applications	80
3.4	Considerations and limitations	81
3.5	Cross references	82
3.6	Summary	83
Part 2.	Migration to Windows 2000	85
Chapter 4.	Migrating OS/2 Servers to Windows 2000	87
4.1	Overview of Windows 2000 migration	88
4.1.1	Considering the order of migration steps	88
4.1.2	Design of Active Directory	89
4.1.3	Tools used during migration	94
4.2	Migrating the domain	100
4.2.1	Preparing Active Directory prior to first migration	100
4.2.2	Steps for each domain	101
4.3	Migrating server definitions	103
4.3.1	Domain controller	104
4.3.2	Member servers	107
4.3.3	Common issues	107
4.4	Migrating groups	108
4.4.1	Before you start	109
4.4.2	Steps to follow	113
4.5	Migrating users	113
4.5.1	Where to start	114
4.5.2	Basic user object	115
4.5.3	Group membership	127
4.5.4	Passwords	129
4.5.5	Logon assignments	132
4.5.6	Steps to follow	140
4.6	Migrating directories	141
4.6.1	Migrating access control	142
4.6.2	Migrating share definitions	148
4.6.3	Migrating the data	150
4.6.4	Migrating DASD limits	152

4.6.5	Steps to follow	153
4.7	Migrating printers	154
4.7.1	Client printing considerations	155
4.7.2	Print queue options	156
4.7.3	Define print queue shares	156
4.7.4	Steps to follow	157
4.8	Migrating serial devices	158
4.9	Migrating applications	158
4.10	NFS migration	159
4.10.1	Software requirement	159
4.10.2	Source platform configuration	159
4.10.3	Migration scenario	160
4.10.4	Installation on the target platform	160
4.10.5	AD4UNIX installation	160
4.10.6	Hummingbird Maestro™ NFS server installation	161
4.10.7	Hummingbird Maestro NFS server configuration	161
4.10.8	Hummingbird Maestro NFS server configuration	162
4.10.9	Windows services for UNIX installation	162
4.10.10	Windows services for UNIX configuration	163
4.11	Migrating OS/2 FTP server to Windows 2000	163
4.11.1	Software requirements	164
4.11.2	Migration scenario	164
4.11.3	Source platform configuration	164
4.11.4	Target platform	165
4.11.5	Hummingbird FTP server installation	165
4.11.6	Hummingbird InetD configuration	165
4.11.7	Microsoft IIS Server installation (FTP server)	166
4.12	DHCP server migration	167
4.12.1	Software requirements	168
4.12.2	Migration scenario	168
4.12.3	Source platform	168
4.12.4	DHCP server installation	170
4.12.5	DHCP server configuration	170
4.13	DDNS server migration	171
4.13.1	Software requirements	171
4.13.2	Migration scenario	171
4.13.3	Source platform	172
4.13.4	Target platform	175
4.13.5	DDNS server installation	175
4.13.6	DDNS server configuration	175
4.14	Summary	176
Chapter 5. Migrating the software stack to Windows 2000		177

5.1 Migrating IBM Universal Database	178
5.1.1 Migration scenario	178
5.1.2 Exporting and importing the data	178
5.2 Migrating IBM e-Network Communications Server	180
5.2.1 Source platform configuration	180
5.2.2 Migration scenario	181
5.2.3 Communications Server installation	181
5.2.4 Migrating the configuration	181
5.3 Migrating Lotus Domino	183
5.3.1 Migration scenario	183
5.3.2 Migrating the configuration	184
5.4 Migrating IBM HTTP Server	184
5.4.1 Software requirements	184
5.4.2 Migration scenario	184
5.4.3 Installing IBM HTTP Server	184
5.4.4 Migrating the IBM HTTP Server	185
5.5 Migrating TSM Client	185
5.5.1 Software requirements	185
5.5.2 Migration scenario	186
5.5.3 Migrating the configuration	186
5.6 Summary	186
Part 3. Migration to Linux	187
Chapter 6. Migrating OS/2 Servers to Linux and Samba	189
6.1 LDAP directory organization	190
6.1.1 Directory structure	190
6.1.2 Enterprise objects	191
6.1.3 Importing basic directory elements and objects	194
6.1.4 LDAP directory maintenance	197
6.2 Migrating the OS/2 domain	198
6.2.1 Organizational Units for each branch	198
6.2.2 Overview of OS/2 domain mapping to Samba	199
6.2.3 Samba domain configuration	199
6.3 Migrating server definitions	199
6.3.1 Overview of OS/2 Server name mapping to Samba	199
6.3.2 Additional OS/2 Server services	200
6.3.3 Configuring Samba server name	200
6.4 Migrating groups	201
6.4.1 Overview of OS/2 group mappings to Samba	201
6.4.2 Preparation for migration	202
6.4.3 Steps to follow for groups	206
6.5 Migrating users and passwords	206

6.5.1	Overview of OS/2 user mapping to Samba	206
6.5.2	Preparation for migration	209
6.5.3	Group membership	213
6.5.4	Logon assignments	217
6.5.5	Steps to follow	218
6.6	Migrating directories and access controls	219
6.6.1	Overview of access controls with Samba	219
6.6.2	Overview of Samba directory shares	220
6.6.3	Create the share point directories	222
6.6.4	Define shares and access controls	222
6.6.5	Copy the data from the OS/2 aliases to the Samba shares	226
6.6.6	Migrating DASD limits	226
6.6.7	Steps to follow	227
6.7	Migrating printers	227
6.7.1	Client printing considerations	227
6.7.2	Print queue options	228
6.7.3	Overview of Samba printer shares	229
6.7.4	Define print queue shares	230
6.7.5	Steps to follow	231
6.8	Migrating serial devices	232
6.9	Migrating applications	232
6.10	NFS migration	232
6.10.1	Software requirement	233
6.10.2	Migration scenario	233
6.10.3	Configuration file for OS/2 Server	234
6.10.4	Configuration file for Linux server	235
6.10.5	Advanced configuration	235
6.11	FTP migration	235
6.11.1	Software requirements	235
6.11.2	The migration scenario	236
6.11.3	SuSE FTP configuration	236
6.11.4	Red Hat FTP configuration	238
6.11.5	Creating users on Red Hat	240
6.11.6	Creating users on SuSE	241
6.11.7	Transfer the data from OS/2 to Linux	242
6.12	DHCP migration	243
6.12.1	Software requirements	243
6.12.2	Migration scenario	243
6.12.3	Configuration file for OS/2	244
6.12.4	Using a script to migrate the DHCP configuration	245
6.12.5	DHCP configuration file for Linux	250
6.12.6	Advanced configuration	250
6.13	DNS migration	251

6.13.1	Software requirements	251
6.13.2	Migration scenario	251
6.13.3	Creating a secondary DNS	253
6.13.4	DNS configuration files for OS/2	254
6.13.5	DNS configuration files for Linux	255
6.13.6	Advanced configuration	257
6.14	DDNS migration	257
6.14.1	Software requirements	257
6.14.2	Migration scenario	258
6.14.3	Configure the Linux DDNS server	258
6.15	Summary	263
Part 4. Tools and scenarios		265
Chapter 7. Migrating the software stack to Linux		267
7.1	Migrating IBM Universal Database	268
7.1.1	Migration scenario	268
7.1.2	Exporting and importing the data	268
7.2	Migrating IBM e-Network Communications Server	270
7.3	Migrating Lotus Notes® server	270
7.3.1	Migration scenario	271
7.3.2	Migrating the configuration	272
7.4	Migrating IBM HTTP Server	272
7.4.1	Software requirements	272
7.4.2	Migration scenario	272
7.5	Migration of ADSM Client	273
7.5.1	Software requirements	273
7.5.2	Migration scenario	273
7.5.3	Migration of the dsm.opt file	274
7.5.4	Migrating the configuration	275
7.6	Summary	275
Chapter 8. Additional migration tools		277
8.1	Various IBM tools	278
8.1.1	IBM Networks UAM for Microsoft Windows 2000	278
8.1.2	IBM Networks Password Synchronization Tool	281
8.2	Starfire Titan	283
8.2.1	Configuration	284
8.2.2	Features and functions	285
8.2.3	OS/2 LAN Server migration scenario	286
8.2.4	Transformation customization	288
8.2.5	Extraction from OS/2	290
8.2.6	Migrating an OS/2 domain	292
8.2.7	Starfire Titan during and after migration	300

8.3	6PAC Network administrative tools	301
8.3.1	Logon Script Manager offering	302
8.3.2	OS/2 to Windows migration tools	308
8.3.3	Network application tools	309
8.3.4	Unattended Installation Manager	310
8.4	Lieberman & Associates	312
8.4.1	Migration procedures	313
8.4.2	Installing the Migration Wizard and preparation	313
8.4.3	Step 1: Setting Windows 2000 domain definitions	316
8.4.4	Step 2: Import the OS/2 LAN Server data	318
8.4.5	Step 3: Resolve	320
8.4.6	File ACLs/Directory ACLs	334
8.4.7	Printer aliases	336
8.4.8	Logon scripts	337
8.4.9	Step 4: Export to the Windows 2000 Domain	343
8.5	Servolution	345
8.5.1	Overview of Servolution	345
8.5.2	How Servolution works	347
8.5.3	Migration scenario using Servolution	350
8.5.4	Installation of the Servolution SyncPacket on the target server	355
8.5.5	Install and configure the Servolution SyncProxy	356
8.5.6	Preparation of the Linux target platform	357
8.5.7	Installing the SyncAgent for Linux	359
8.5.8	Resource migration to Samba	361
8.5.9	User management migration to LDAP	366
8.6	Summary	372
	Chapter 9. Linux for OS/2 administrators	373
9.1	Linux security	374
9.1.1	Physical security	374
9.1.2	System security	374
9.1.3	Network security	381
9.1.4	Backup security	387
9.2	Linux administration	388
9.2.1	Partitions	388
9.2.2	File systems	391
9.2.3	Scripts	392
9.2.4	Crontab	394
9.2.5	Network status	395
9.2.6	System logs	401
9.2.7	Remote administration	403
9.3	Summary	407

Part 5. Appendixes	409
Appendix A. Windows 2000 migration related scripts	411
CID installation of Windows 2000	412
Windows installation related scripts	413
SERVICE.CMD	413
W2K.CMD	414
POST1.CMD	414
POST2.CMD	415
SYSOCRMGR.CMD	415
DCPROMO.CMD	416
Windows installation related response files	417
WINDC.TXT	417
PART.TXT	420
W2K_INST.REG	420
WINDCP.TXT	420
WINMEMP.TXT	423
INSTDHCP.TXT	426
INSTWINS.TXT	426
INSTDNS.TXT	426
INSTFTP.TXT	426
INSTWWW.TXT	427
DCPROMO1.TXT	427
DCPROMO2.TXT	428
INSCERTSRV.TXT	428
CS.ISS	429
NOTES.ISS	430
DB2.RSP	435
Domain migration related files	449
BASEOU.LDIF	449
BRANCHOU.LDIF	451
Migrating groups	452
GROUPS.LDIF	452
GROUP-DB.CSV	453
Migrating users	454
CreateUser.vbs	454
USERS.LDIF	459
GRPMEM.LDIF	463
LOGON.CMD	465
CHECKOS.CMD	467
OS2ENV.CMD	468
SETWINUSERASN.CMD	468
WYNAND.CMD	470

Migrating directories	471
GETWINACL.CMD	471
SETWINACL.CMD	472
SETWINSHARE.CMD	473
SETWINCOPY.CMD	475
Appendix B. REXX source code	477
Global source code and input files	478
RGUTIL.CMD	478
RGUTILS.CMD	479
RGLSRXUT.CMD	480
LSMT.RSC	483
TRANSFORM.USER	488
TRANSFORM.GROUP	488
Source code for retrieving server information	489
GETSRVR.CMD	489
SERVER.INI	494
Source code for groups	496
GETGRPS1.CMD	496
GROUPS.INI	501
SETGROUPS.CMD	501
GETGRPS2.CMD	504
SETGRPMEM.CMD	510
Source code for USER	512
GETUSERS.CMD	512
USERS.INI	518
SETUSERS.CMD	519
Source code for passwords	524
GETPWD.CMD	524
Source code for access control lists	529
GETSMBACL.CMD	529
GETWINACL.CMD	537
Source code for aliases	539
GETALIAS.CMD	539
ALIAS.INI	545
SETSMBDIRALIAS.CMD (Samba Only)	545
SETSMBPRNALIAS.CMD (Samba Only)	549
SETWINSHARE.CMD (Windows only)	551
SETWINCOPY.CMD (Windows Only)	553
Appendix C. Additional material	557
Locating the Web material	557
Using the Web material	557

How to use the Web material	558
Abbreviations and acronyms	559
Related publications	561
IBM Redbooks	561
Online resources	561
How to get IBM Redbooks	564
Help from IBM	564
Index	565

Figures

1-1	Sample OS/2 domain product overview	6
1-2	Communication server sample configuration	8
3-1	Transition process	64
4-1	General workflow of domain migration to WIndows 2000 domain	89
4-2	Active Directory design for transition of OS/2 branch domain	90
4-3	Organizational units in somedomain.local	93
4-4	Migration workflow for domain	100
4-5	Migration workflow for additional servers	103
4-6	Migration workflow for group part	109
4-7	Migration workflow for user part	114
4-8	Migration workflow for directory part	141
4-9	Migration workflow for printer migration	154
6-1	LDAP Directory design for transition of OS/2 branch domains	191
6-2	LDAP Directory Organizational Unit layout for somedomain.local	193
6-3	Migration scenario	234
6-4	Creating users using redhat-config-users tool	241
6-5	Creating users using the yast2 tool	242
6-6	Migration scenario using DHCP	253
8-1	Starfire Titan deployment overview	284
8-2	Titan Control Center view with migration activities	293
8-3	Select the OS/2 domain to migrate	294
8-4	Select the target platform	295
8-5	Selecting the target Linux OrgUnit	295
8-6	Selecting the OS/2 domain object types to migrate	296
8-7	Selecting which objects of each type to migrate	297
8-8	Select the user IDs for migration	297
8-9	Select group IDs - not all groups are selected for migration	298
8-10	Select the directory aliases for migration	298
8-11	Select the printer aliases for migration	299
8-12	Completed migration Activity Launch Summary	299
8-13	Administration of logon assignments with LSM	306
8-14	Status windows of LSMUSE.EXE	307
8-15	Summary of failed connections from LSMUSE	308
8-16	Main window of UIM	311
8-17	Edit domains	316
8-18	Add Windows 2000 domain	317
8-19	LAN Server Domain Configuration	318
8-20	Migration Wizard main screen	319

8-21	Import from Local File to Internal Database	320
8-22	Resolve Importation Issues	322
8-23	Resolve User Accounts	322
8-24	Path Mapping	324
8-25	Set Password Policy	326
8-26	LAN Server Group Enable/Mapping	329
8-27	File Alias Mapping	330
8-28	Export Data	333
8-29	File ACLs mapping.	335
8-30	Directory ACLs Mapping	336
8-31	Mapping Printer Aliases.	337
8-32	Logon script creation	338
8-33	Add script location	339
8-34	BAT file settings under logon scripts	340
8-35	CMD file settings for OS/2 and Windows 2000 Clients	342
8-36	Set up home directories and shares	343
8-37	Export to Windows 2000 Domain.	344
8-38	Servolution solution overview	349
8-39	Servolution migration step 1	352
8-40	Servolution migration step 2	353
8-41	Servolution migration step 4	355
8-42	LDAP Configuration step 1	368
8-43	LDAP Configuration step 2	371
9-1	Security settings in SuSE.	377
9-2	Password settings	378
9-3	Authentication Configuration for Red Hat 7.2	381
9-4	SysV Init Editor	383
9-5	Properties for a service	384
9-6	Start/Stop a service	385
9-7	netstat -a grep "LISTEN " command output.	386
9-8	netstat output	395
9-9	netstat -a output.	397
9-10	netstat -p output.	397
9-11	netstat statistic output	398
9-12	IPTraf utility	400
9-13	Red Hat system services	401
9-14	Syslog configuration file	402
9-15	tail -f /var/log/messages.	403
9-16	Webmin server configuration page	404
9-17	Webmin interface.	405
9-18	Starting VNC Server on Linux	406
9-19	VNC viewer	406
A-1	Core tree of CID structure for Windows 2000 unattended installation	412

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law. INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.


This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

IBM eServer™	IMS™	Redbooks™
Application System/400®	Lotus Notes®	RACF®
AIX®	Lotus®	RS/6000®
AS/400®	LANDP®	S/390®
Domino®	Maestro™	Sequent®
DB2 Connect™	Monday™	ServeRAID™
DB2 Universal Database™	Netfinity Manager™	SOM®
DB2®	Netfinity®	Tivoli®
DFS™	Notes®	Visual Warehouse™
DRDA®	Operating System/2®	WebSphere®
Home Director™	OS/2®	iRedbooks(logo)  ™
Hummingbird®	OS/400®	
IBM®	Perform™	

The following terms are trademarks of other companies:

Intel, Intel Inside (logos) are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Preface

IBM® has endorsed the strengths and benefits of Internet technologies and platform independence for several years, and has encouraged customers worldwide to make the transition to network computing.

To facilitate this transition, IBM has enhanced the OS/2® operating system to become an excellent platform for the deployment of e-business applications, while at the same time helping preserve investments in legacy applications. IBM has created a transformation plan that includes information customers can use to help transform their current client-and-server solutions into e-business solutions.

Industry standards, Internet technologies, and platform independence are IBM's strategic recommendations for coping with the rapid pace of software and hardware technology changes. Exploitation of industry standards and Internet technologies hedges information technology investments, and platform independence preserves choices and options. Customers who have already made the transition to network computing have discovered that Internet technologies and platform independence can create a competitive advantage: this helps reduce costs and facilitates the rapid deployment of new applications and services. The transformation to e-business is a critical factor in a company's growth and prosperity, or a defensive strategy to protect a business from competitors. IBM has formalized its vision of e-business with the WebSphere® Software Platform as its cornerstone.

With all this in mind, it clearly makes sense to explore the possibilities to transform the underlying platform to another operating system with its given advantages or limitations. While the operating platform is less important for the business applications itself, it still provides critical functions such as user authentication, file and print services, as well as the base platform for the middleware stack.

With IBM OS/2 Warp Server for e-business as the starting point, this book will focus on the migration of Intel® based server environments from the latest version of IBM OS/2 Warp Server for e-business, namely Convenience Pack 2, to Windows® and Linux with the version available during the time this book was written. Namely, Microsoft® Windows 2000 Advanced Server, SuSE Linux Enterprise Server 8 (based on UnitedLinux 1.0), and Red Hat Enterprise Server 2.1.

This IBM Redbook covers a detailed step-by-step migration towards the target platforms, advice on tools or scripts to support the migration, and also how to

automate and simplify these tasks as well as covering an end to end approach within a given environment:

- ▶ Part 1 of the book focuses on the scenario for a migration from OS/2 to Microsoft Windows or Linux, and the major pre-requisites.
- ▶ Chapter 1 describes the technical context where a migration would typically start from. That is, an OS/2 Warp Server for e-business based environment.
- ▶ Chapter 2 discusses both the Windows environment, and the two major Linux distributions, Red Hat and SuSE, to outline the typical target environments of a migration. This chapter includes a detailed discussion on how to set up the proper environment using LDAP and Samba, which is common for both Linux distributions.
- ▶ Chapter 3 outlines a number of activities to extract data from the OS/2 based environment. This data will then be used to create a similar environment on the target platforms later. At this stage, the focus is clearly on administrative information within the OS/2 LAN Server context.
- ▶ Part 2 of the book focuses on the migration from OS/2 to Windows.
- ▶ Chapter 4 discusses the migration of administrative information from an OS/2 domain to a Windows 2000 infrastructure based on Active Directory as outlined in Chapter 2.
- ▶ Chapter 5 provides a brief overview of recommendations and activities to migrate the major IBM products currently implemented and used by a majority of customers on OS/2 to their equivalent product versions on Windows.
- ▶ Part 3 of the book focuses on the migration from OS/2 to Linux.
- ▶ Chapter 5 discusses the migration of the administrative information from an OS/2 domain to a Linux infrastructure based on LDAP and Samba 3.0 as it is outlined in Chapter 2. Most steps will be common for both distributions covered in the book. Differences will be outlined as appropriate.
- ▶ Chapter 6 provides a brief overview of recommendations and activities to migrate the major IBM products currently implemented and used by a majority of customers on OS/2 over to their equivalent product versions on Linux.
- ▶ Part 4 of the book covers some additional tools that can be used to assist with the migration, or to go beyond a pure and native migration.
- ▶ Chapter 8 covers a number of tools used during and after a migration for management of a heterogeneous environment.
- ▶ Chapter 9 gives a brief introduction to some key administration tasks on Linux for the OS/2 administrator.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

The team from left to right: Oliver Mark, Richard Spurlock, Marc Schneider, Wynand Pretorius, Leif Braeuer, Andrei Vlad, Bart Jacob.



Bart Jacob is a Senior Consulting IT Specialist at IBM's International Technical Support Organization, Austin Center. He has 23 years of experience providing technical support across a variety of IBM products and technologies, including communications, object-oriented software development, and systems management. He has over 10 years of experience at the ITSO, where he has been writing IBM Redbooks™ and creating and teaching workshops around the world on a variety of topics. He holds an MS degree in Numerical Analysis from Syracuse University.

Oliver Mark is a Certified Consulting IT Architect at IBM Global Services Integrated Technology Services Germany. He has worked as a temporary Project Leader for this redbook at the International Technical Support Organization, Austin Center. He has participated in a number of residencies including Warp,

Linux, and other topics and has published two external books. He has over 12 years of experience in the client/server area in various sectors, with a strong focus on OS/2-based end-to-end architectures. He is responsible for all IBM Global Services service offerings for OS/2 and its customers in the EMEA Central Region. He holds a degree in economics and computer technologies. His areas of expertise include OS/2, Warp Server, IBM communication products, Windows, and Linux, as well as infrastructure and systems management architectures. As a member of the IBM Technical Expert Council, he has the opportunity to work on emerging topics, and gives IBM management advice on future directions.

Richard Spurlock is the Vice President of Starfire Engineering and Technologies, Inc. and the Chief Architect and Developer of the Starfire product called Titan. He is a graduate of the University of Kansas and holds a Bachelor's of Science in Computer and Electrical Engineering. He has 14 years of experience in the OS/2 field and has been designing, building, and deploying OS/2 solutions since the release of OS/2 1.0. During the past five years, Starfire has been building and delivering products for the administration, integration, and migration of OS/2, Windows, Linux, and AIX® platforms. He has extensive system and development experience with OS/2, OS/2 Warp Server, Workspace On-Demand, and related products. He currently leads Starfire teams designing and deploying cross-platform customer solutions.

Marc Schneider is a Project Manager at Sinius GmbH in Germany. He has over 10 years of experience in OS/2. He has worked for OS/2 related projects at Deutsche Bank for over five years. He has experiences in managing large OS/2 domains and migrating them to NetBIOS over TCP/IP. His areas of expertise include OS/2, Warp Server, IBM communication products, virtualization, Windows, and Linux.

Andrei Vlad is an IT specialist in IBM Global Services Romania. He has two years of experience in the OS/2 field and is an AIX Certified Engineer, and a Linux expert. He has five years of UNIX® experience, and has been involved in the design and implementation of several large UNIX-based projects, including support and special customizations for a variety of applications. His areas of expertise include OS/2, Linux, AIX, TCP/IP, firewalls, and clustering.

Leif Braeuer is an IT Consultant and member of the executive board at 6PAC Consulting AG in Hamburg, Germany. He has 14 years of experience in the OS/2 field starting with release 1.0. Having a focus on the Warp Server area, he has expanded his areas of expertise to Windows Server products and Linux, as well as network infrastructure, virtualization, and terminal services. At 6PAC, he is responsible for managing and implementing large IT projects and consulting with executives. In addition, during the last 10 years he has brought several software

products for the administration and migration of OS/2 and Windows domains to the marketplace.

Wynand Pretorius is a Software Support and Development Manager at Standard Bank of South Africa. He has 10 years of experience in the OS/2 and six years in Workspace on Demand. He develops, maintains, and supports the Workspace on Demand and OS/2 environment at Standard Bank. His areas of expertise include OS/2, WSOD, IBM Communication, Network, and Architect design.

Thanks to the following people for their contributions to this project:

Vasfi Gucer, Chris Blatchley
International Technical Support Organization, Austin Center

David Fritz, Jason Kersten, Steven French, Jim McDonough, Larry Sullenger,
Milind M Bade, Uwe Zimmermann
IBM Austin

Alain Rykaert
IBM Belgium

Arthur Cannon
IBM Dallas

Paul Landay, Jeff L. Smith
IBM Raleigh

Doug Foulds
IBM Toronto

David Bennin
IBM Poughkeepsie

Bill Marshall
IBM Rochester

Jason Edmeades
IBM UK

Michael Wagner
Comtarsia IT Services GmbH

Vicki Trufasiu
Hummingbird® Ltd.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:

ibm.com/redbooks

- ▶ Send your comments in an Internet note to:

redbook@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. JN9B Building 003 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493



Part 1

Introduction and preparation

The first part of this book covers areas related to the preparation for a migration from an OS/2 Server to another platform. It describes various items to consider prior to a migration in the context of a client/server based environment.

The first chapter introduces the audience for a typical OS/2 installation, and the implementation of OS/2 Warp Server for e-business by providing an overview of the products and tools commonly used.

Chapter 2 describes both the Windows and Linux environments that will be the target of the migration.

Chapter 3 discusses the tools and utilities used to extract data from an OS/2 environment so the data can be re-used to set up and configure the target environment on the appropriate platform.



OS/2 Server environment

This chapter provides an overview of a typical and current IBM OS/2 Warp Server for e-business based implementation. It discusses the products and features installed and used, common configurations, and the product stack used on top of the base IBM OS/2 Warp Server for the e-business installation.

The sample scenario on which this book is based is described in detail, so that the reader receives a solid understanding of the initial environment.

In this chapter, the following topics are described:

- ▶ The IBM OS/2 Warp Server for e-business base installation
- ▶ The sample domain structure
- ▶ Configured TCP/IP based services
- ▶ Product stack on OS/2:
 - IBM Universal Database
 - IBM e-Network Communications Server
 - IBM Lotus® Domino® Server
 - IBM HTTP Server
 - IBM Tivoli® Storage Manager Client

1.1 IBM OS/2 Warp Server for e-business base installation

The IBM OS/2 Warp Server for e-business base installation is based on the Convenience Package 2. This package has been installed on two IBM 300GL machines with one network card installed in each system.

The partition table has been set up as follows.

Table 1-1 Partition table

Drive Letter	Filesystem	Type	Name	Size
C:	HPFS386	Compatibility	OS/2 System	800 MB
D:	HPFS386	Compatibility	Maintenance	400 MB
E:	HPFS386	LVM	Data 1	1500 MB
F:	JFS	LVM	Data 2	remaining space
I:	DumpFS	Compatibility	SADUMP	512 MB

During the base installation, the following features were selected in addition to the default features:

- ▶ Security enabling services (SES)
- ▶ LAN Server file and print
- ▶ Tivoli Management Agent

The following network protocols have been bound to the network cards on each server:

- ▶ IBM IEEE 802.2 for Communication Server support
- ▶ IBM NetBIOS for native LAN Server support
- ▶ IBM TCP/IP for IP-based services such as DHCP and NFS
- ▶ IBM NetBIOS over TCP/IP towards the enablement of migration

On top of the base installation, a number of services and products were installed on the two servers as outlined in the following pages. On the Primary Domain Controller (PDC) the TCP/IP services, the DHCP server, and DDNS server have also been installed. On the Backup Domain Controller (BDC) machine, the LPRPORTD server services were installed to act as a TCP/IP based print server as well.

For the configuration of these, refer to 4.12, “DHCP server migration” on page 167, and 4.13, “DDNS server migration” on page 171.

1.2 Sample domain

Our sample domain consists Primary Domain Controller (PDC) and the Backup Domain Controller (BDC). Both of them are acting as file servers, while the PDC is also a DHCP and DDNS server, and the BDC is a TCP/IP based print server.

The domain name is SOMEDOMAIN and there are only two aliases, called BOOK and LANSHARE. In this domain, four user groups of interest exist: PRINTER, TRANSITION, BOOKWRITE, and BOOKREAD.

The table below shows the users associated with the groups.

Table 1-2 User accounts and groups

User ID	PRINTER	TRANSITION	BOOKREAD	BOOKWRITE
ANDREI		X	X	
LEIF	X	X	X	
MARC	X	X	X	
OLIVER	X	X		X
RICHARD		X	X	
WYNAND		X	X	

The user account of Wynand is restricted to logon only weekdays (Monday™ to Friday) from 07:00 to 19:00, and only from workstations PC1 and PC2.

Table 1-3 Resources

ALIAS	DASD Limit	GROUP	RIGHTS	LOCATION
LANSHARE	500 MB	TRANSITION	RWCDA	BDC on E: Drive
BOOK	50 MB	BOOKREAD BOOKWRITE	R\ RWCDA	PDC on F: Drive
HOMEDIR	100 MB	%USER%	RWCXDAP	PDC on E Drive
PRINT_Q		PRINTER	CP	BDC IBMNULL

A DOS application is defined as a Public application within the domain. The application name is DOS_PRG and it is located on the LANSHARE share in the sub-directory DOSAPP. No access restrictions apply.

Several services and IBM products have been installed on the two servers in the domain as can be seen in Figure 1-1.

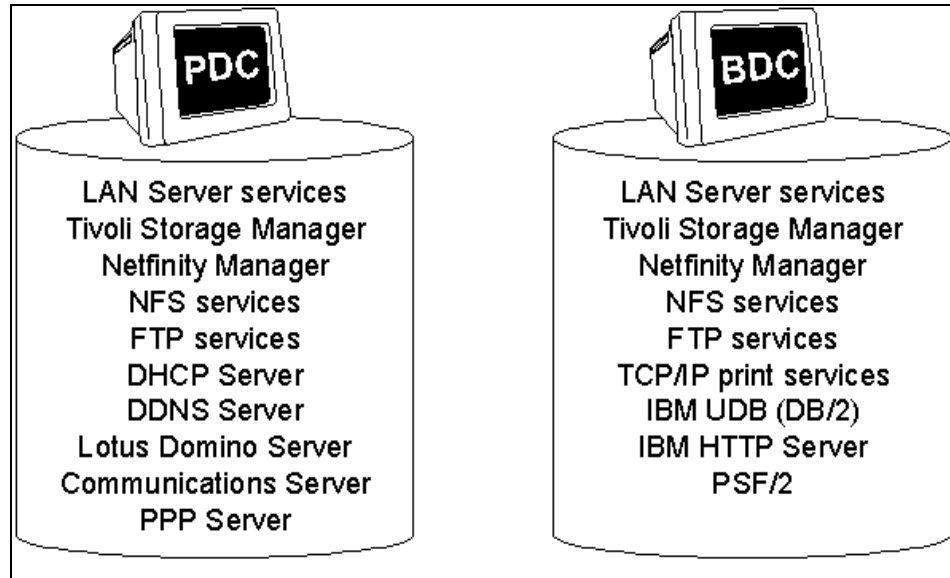


Figure 1-1 Sample OS/2 domain product overview

1.3 Configured TCP/IP-based services

There are some basic TCP/IP-based services that come along with OS/2 Warp Server for e-business. In our environment, we installed:

- ▶ FTPD

All users were configured as they are within LAN Server with full access to their own directories on each machine (on drive F:\FTP\HOME\%USER%).

- ▶ NFSD

Configured in a basic way, with access to Drive F:\NFS on each machine. No access restrictions applied.

- ▶ DHCP

Configured to provide one subnet 192.168.25.0 to requesting clients. Assigned range for dynamic configuration was from 192.168.25.10 to 192.168.25.200.

- ▶ DDNS

The base domain name is `somedomain.local`. The DDNS is configured such that every client can modify its own host name. The two servers are added statically to avoid IP address conflicts.

1.4 Product stack on OS/2

In the following, several IBM middleware products are covered. Several products were installed and migrated during the creation of this book. If a product is not listed below, please review the product documentation or any redbook on this product to verify its ability to be migrated to a target platform.

1.4.1 IBM Universal Database

The DB2® Universal Database™ Enterprise Edition version 7.2 was installed on the BDC.

The default components were selected for the UDB installation. The DB2 system name was configured as BDC with the **Auto start DB2 instance at boot time** selected. The user ID and password that are used to administer the server are *userid* and *password*. Note the user ID must have special account privileges. For more information on special accounts, review the UDB help documentation.

A UDB sample database with sample information was created.

1.4.2 IBM e-Network Communications Server

The e-Network Communications Server Version 6.1 without any Fixpacks was installed on the PDC. You may want to add Fixpacks to the system if you are operating it in a production environment. The Fixpacks should not have any impact to the migration scenario.

As an example, e-Network Communications Server 6.1 was installed with Enterprise Extender, acting as a Gateway between a number of clients and a mainframe. Connections to the clients in this profile are based on the SNA protocol, while the uplink to the host uses TCP/IP communication. This is a pretty common scenario for applications using LUA communication locally, but which are required to use TCP/IP on a wide area network.

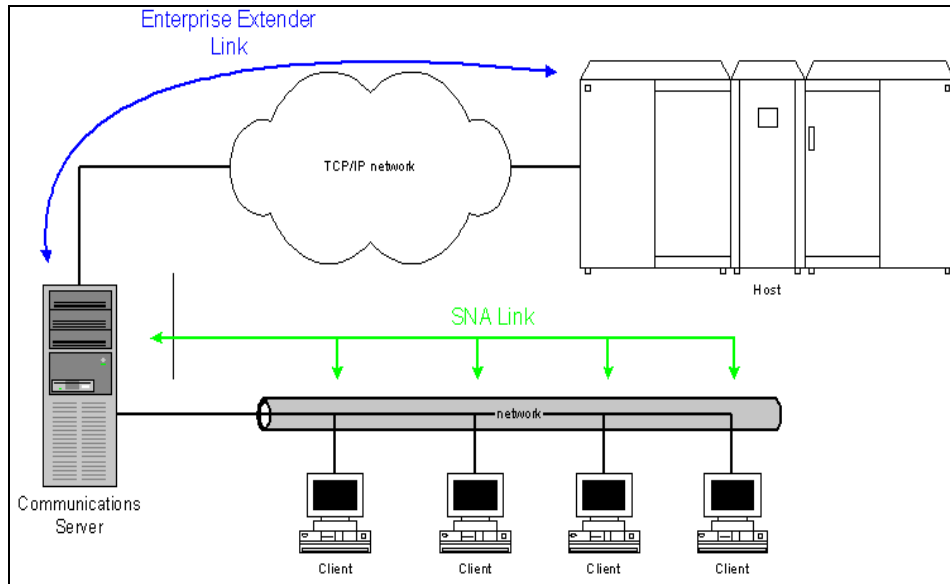


Figure 1-2 Communication server sample configuration

1.4.3 Lotus Domino Server

Lotus Domino Server was installed using Version 5.0.11 on OS/2, which was the current level at the time of the writing of this book. Cross-platform migration of Lotus Domino within the same version is simple and straight forward, while upgrading to a newer version is generally a bigger task regardless of which platform this is done on. Therefore, this redbook will cover the general tasks to migrate a Domino Server to another platform, but not a version to version migration, which is beyond the scope of this book, and can be found within the documentation provided by Lotus.

1.4.4 IBM HTTP Server

The IBM HTTP Server powered by Apache, as it is available on the Software Choice download Web site:

<http://www.software.ibm.com/os/warp/swchoice>

This was installed and a number of sample Web pages were used to validate the migration.

1.4.5 IBM Tivoli Storage Manager Client

While the Tivoli Storage Manager Server product has never been available on OS/2, Tivoli Storage Manager (TSM), or, to be more specific, Adstar Distributed Storage Manager (ADSM) has a client version available on OS/2, which is widely used.

The following TSM client for OS/2 components were selected for the reference installation:

- ▶ Administrative client command line interface
- ▶ Backup-Archive client command line interface
- ▶ Backup-Archive client graphical user interface
- ▶ ADSM Getting Started, Requires OS/2 Multi-Media
- ▶ Application Programming Interface

The following components were *not* selected for the reference installation:

- ▶ WEB Backup-Archive client
- ▶ WEB client National Language Support (NLS) files

The TSM client for OS/2 v3.1.0.8 was configured to use TCP/IP for backup and archive functions. The target server was an RS/6000® with AIX 4.3.3 and TSM 5.1.5 Server installed. The following was the configuration file used with the ADSM client for OS/2:

```
NODENAME OS2MEM
TCPSEVERADDRESS 9.3.5.36
TCPPOINT 1500
COMPRESSION YES
COMPRESSALWAYS NO
```

1.4.6 IBM LAN Distributed Platform

LANDP® is a distributed platform solution that includes support for financial devices, host communications and data management. LANDP can be used to create the solution that best fits a customer's needs, or it can be integrated with other products such as WebSphere Business Component Composer in order to build a customized solution.

A LANDP solution is built on a group of workstations, which provide services to other workstations within the workgroup. The platforms of these workstations can be a mixture of the supported LANDP platforms. LANDP support DOS, OS/2, Windows, and Linux, and it allows your solution to be built in either a distributed or centralized environment.

LANDP can provide an effective solution for a centralized environment. Using Java™ and an application server, your applications can communicate with

LANDP services stored centrally. LANDP can also drive financial devices attached to a client machine with a small footprint on the client machine.

LANDP enables existing LANDP solutions to migrate to other platforms. The application programming interface (API) is common across all platforms and the most common services offered by LANDP are available on all platforms.

For LANDP solutions that use DOS based applications, Virtual DOS machine relay support is used. This allows LANDP (running on Windows or OS/2) to process requests from LANDP for DOS clients that are running in a Windows or OS/2 virtual DOS machine. This will help reduce the effort to migrate to Windows, because you do not need to port your DOS applications. LANDP for Linux requires a third party virtual DOS software called DOSEMU.

Please see the LandP whitepaper on migrating to an e-business infrastructure at:

<http://www-3.ibm.com/software/network/landp/library/whitepapers.html>

1.4.7 IBM WebSphere MQ

IBM WebSphere MQ is market-leading business integration software. It connects business software together to form one efficient enterprise by providing an open, scalable, industrial-strength messaging backbone.

WebSphere MQ minimizes the time taken to integrate key resources and applications held in different systems, so a company can respond to the changing demands of e-business. By connecting business information with people and other applications, more value can be extracted from existing investment, and quickly integrated into new systems to support new market strategies.

Generally, the majority of the migration issues are related to getting the users applications rewritten, while the MQ portion is minimal. MQ passes messages from one place to another, so they are generated or received elsewhere, and the configuration should be a very small part of the migration issue. The information that would need to be migrated for WebSphere MQ are the definitions such as queue managers, queues, and so on. IBM does not provide any tools to migrate WebSphere MQ configurations from OS/2. However, there is a support pack available at:

<http://www-3.ibm.com/software/integration/support/supportpacs/individual1/ms03.html>

This Web site contains a program that can be compiled on OS/2. The makefiles are included, but the source code is shipped “as-is” and is unsupported. This

would export the definitions in the form of a script, which can be executed to produce the same configuration on another platform.

Similarly, the instructions to generate the definitions and the programming interface used is common between OS/2, Linux, and Windows, although with Microsoft Windows a GUI is available.

1.4.8 IBM Netfinity® Manager™

The Netfinity Manager is installed on the system disk with Remote Workstation Control, and all available network protocols are bound to the manager.

Unfortunately, no migration path or tool is available to transform any settings or configurations to IBM Director.

1.5 Recommended steps prior to migration

As with all other migrations of a current system or platform to a new environment, a good portion of the work should be considering some redesign to support current standards, products and future directions. A plain migration of functions from one platform to another will not provide any benefits in the long term, while incorporating newer technologies or implementing better approaches might be a little more complex or expensive, but will usually be easier to manage and provide more flexibility in the future.

On the downside, this most likely requires a lot of re-work, and only a limited amount of the current configurations and experiences can be used. This kind of migration is approached in a staged fashion, starting with the application stack first, migrating clients, and finally moving the server infrastructure.

1.5.1 General architectural thoughts

The infrastructure of the current OS/2 Warp Server based domain might have grown over the last ten or more years. It might have been state-of-the-art at the time it was introduced. Some of the limitations from prior LAN Server versions might have been carried over the years and never been changed.

Nevertheless, when migrating from one platform to another, the general design of the infrastructure might be worth reconsidering. With technologies such as LDAP and Active Directory being available, the structure of a domain can be seen in a different light. New devices might be introduced as well since storage can be moved from servers to network attached storage, or SAN devices. Software deployment needs to be re-evaluated along with many other considerations.

Covering all the concerns and areas of discussion would be well beyond the scope of this book and can be found with other sources. Prior to any migration, and prior to modifying any of the machines in a network, be sure that a well thought out design is in place, and that the infrastructure is ready for the future.

1.5.2 Security

Security has become one of the major issues in current IT infrastructures including physical security for machines, logical security for files and content, or protection against viruses or intrusions from the outside.

Network implementations on OS/2 have tended to be more secure and less vulnerable against viruses, hoaxes, and trojan horses. The lack of Visual Basic has also eliminated a common environment for viruses.

Products known for their bugs or security exposures are often implemented differently on OS/2, or not available at all.

From a security point-of-view, migrating from OS/2 to Windows should include a lot of considerations, while migrating to Linux might be a little less painful. Nevertheless, the security considerations when choosing a new platform would require an extensive discussion, and is beyond the scope of this book.

1.5.3 Virus protection

Antivirus software protects your data from viruses, scans e-mail for viruses, hoaxes and trojan horses, tells you when you have a virus, and rids your system of viruses. The virus protection of a network, its servers, and clients is very important.

Important: Be sure to have appropriate virus protection installed on your servers and clients in your network.

There are two types of antivirus software:

- ▶ *Operating system level* antivirus software, which scans the files on the computer for known viruses.
- ▶ *Application level* antivirus software, which is written for a specific application, such as the Domino server, or mail server and file share servers such as Samba.

An operating system level antivirus product can be scheduled to run daily or weekly at certain times (such as at midnight, at night, at the end of the work

hours, or the end of the week). It is recommended that the antivirus software is scheduled to run outside work hours because it is a heavy task, consuming a lot of CPU power, memory, and intensive disk access.

Generally, there is no migration path for virus scanners. If you are installing a virus scanner, you will get a new virus definition file, and you have to update it frequently. The virus scanner configuration depends on the machine itself and there is no need to migrate it.

Note: The risk of being infected on OS/2 or Linux is currently much lower than on Windows.

1.5.4 Printer migration

Because networks are getting more complex and becoming more heterogeneous, it is highly recommended to reconsider network services such as printing from local or server attached to the TCP/IP protocol. Most existing and traditional products, which are in use nowadays in the network may use different and proprietary communications between the print server and the printing device. These products may not be available on all target and future platforms. So, a migration to an open and standard-based implementation to ensure readiness for future requirements and hardware should be considered. If it is impossible to change the printing devices themselves to TCP/IP, you should consider implementing the TCP/IP protocol for printing on the target server.

Note: If possible, it is recommended to change your printer environment to the TCP/IP based protocol. This might have an impact to the clients in the network.

1.5.5 Transport protocol migration

Many clients and servers still communicating through NetBIOS with each other. NetBIOS is a non routable protocol, which means it is not possible to use it through segmented networks, if these are not prepared in a special manner.

Note: Change the communications protocol to NetBIOS over TCP/IP.

The protocol migration itself is very easy, just add the TCPBEUI protocol to the adapters within MPTS. You have to take care about the adapter number, which must not be the same as the NetBIOS adapter. If this is done you can add the additional adapter to the LAN-Server / LAN-Requester configuration stored in IBMLAN.INI.

For the configuration, it is important that the adapter numbers in the protocol.ini file and in the ibmlan.ini file are matching.

Example 1-1 Example sections of PROTOCOL.INI

```
[NETBIOS]

    DRIVERNAME = NETBIOS$
    ADAPTER0 = NETBEUI$,0
    ADAPTER1 = TCPBEUI$,1

[tcpbeui_nif]

    DriverName = tcpbeui$
    Bindings = ,IBMFEEO2.NIF
    NODETYPE = "H-Node"
    OS2TRACEMASK = 0x0
    SESSIONS = 254
    NCBS = 255
    NAMES = 21
    SELECTORS = 16
    USEMAXDATAGRAM = "NO"
    NETBIOSTIMEOUT = 800
    NETBIOSRETRIES = 2
    NAMECACHE = 100
    PURGECACHE = 0
    PRELOADCACHE = "YES"
    NAMESFILE = 100
    DATAGRAMPACKETS = 22
    PACKETS = 80
    ENABLEDNS = 2
    INTERFACERATE = 360
```

Example 1-2 Example sections of IBMLAN.INI

```
[networks]

    net1 = NETBEUI$,0,LM10,102,222,14
    net2 = TCPBEUI$,1,LM10,102,222,14

[requester]
...
    wrknets = NET1, NET2
[server]
...
    srvnets = NET1, NET2
```

In the IBMLAN.INI netx statement, the first number after the protocol qualifier NETBEUI or TCPBEUI and followed by the LM10 keyword corresponds with the adapter number as it is configured in the PROTOCOL.INI file. The next three parameters are the number of NetBIOS sessions, NCBS, and NetBIOS names the requester will use for the its communications.

By default the TCPBEUI protocol on OS/2 will use the Broadcast mode to resolve names in a NetBIOS over TCP/IP Network. For small networks this will work very well but increases the network traffic. For large networks this will not be very pleasing, increase network traffic and is very time-consuming. A mechanism for improved name resolving solution should be considered.

There are various methods for NetBIOS Names resolution

- ▶ Simply broadcasting and hoping the computer you are trying to reach responds prior to application or communication time-outs occurring.
This can be very time-consuming as well as the fact that broadcasts increase network traffic.
- ▶ Keep the NetBIOS computer name and TCP/IP host name identical, so that you can use DNS (Domain Name Services).
This can be complex in existing networks, where different NetBIOS and host names were introduced over time.
- ▶ The recommended method is a dedicated server that provides NetBIOS name to IP address resolution.
 - NBNS (NetBIOS Name Server), such as Shadow IPserver or the NBNS provided with SAMBA. Microsoft WINS cannot be used in this case, since it only supports name resolution for Windows-based clients and since Microsoft's implementation does not always comply with the public standards.
 - Using the different name resolution products or approaches, the client can be configured in various modes:
 - M-Node (Mixed Mode) Clients (name query through broadcast first, and if that fails, it uses NBNS)
 - H-Mode (Hybrid Mode) Clients (Uses NBNS first, and if that fails, it uses broadcast)
 - P-Mode (Point-to-Point Mode)

Using NetBIOS Name resolution through a DNS Server, the *ENABLEDNS* Setting in the TCPBEUI section within the PROTOCOL.INI file has to be modified.

- ▶ ENABLEDNS = 0 No DNS Name resolution

- ▶ ENABLEDNS = 1 First RFC coded names, then DNS names
- ▶ ENABLEDNS = 2 First DNS names, then RFC coded names

In this context, RFC encoded means that all NetBIOS names are converted to a 32-byte word. Using this conversion every byte of the name is known. This is necessary to get the 16th NetBIOS name byte, which describes the services running on this machine (for example, 0x20 means Server, 0x03 means Messenger name, and so on). DNS does not provide this information with unconverted names. To convert names to the RFC coded names, you can use the MAPNAME utility, which can be found in the utility package that comes along with the MPTS Stack.

For more Information about RFC Coded names and NetBIOS over TCP/IP, please see RFC1001/1002.

The plain DNS names are simple names without any service identifier. The client will check the destination machine after it was resolved to an IP address, and returns the local name table from this machine, which includes all the service records.

The DOMAINSCOPE parameter specifies the suffix that will be appended to the query. For example, assume ENABLEDNS is set to 2 and DOMAINSCOPE is set to somedomain.local and your machine is looking for the name PDC. In this case, the DNS query will search for *pdc.somedomain.local*.

As described earlier, a NetBIOS name server (NBNS) is a good choice for large enterprise networks. An NBNS works very similar to a DDNS Server in TCP/IP networks. With NetBIOS over TCP/IP, clients are able to register themselves in the NBNS database.

The Microsoft Windows IP Name Services (WINS) Server handles Microsoft clients, but it is not RFC1001/1002 compliant. An OS/2 domain, for example, would not be resolved by a WINS Server because Microsoft uses 0x1C as the Domain identifier instead of 0x00, which should be correct according to the RFC's for the 16th NetBIOS byte of the name. Using the WINS Server, it will not be possible to add abnormal names with the GUI, instead the **netsh** command should be used.

NBNS settings can be easily distributed using DHCP options. Unfortunately, the current OS/2 Stack does not update these settings dynamically. You have to use a script, which will put the option value coming from DHCP to the protocol.ini before the clients will be rebooted. Windows clients can use the DHCP setting directly.

For more details on NetBIOS over TCP/IP and DHCP please refer to the IBM redbook, *Beyond DHCP*, SG24-5280-01 at:
<http://www.redbooks.ibm.com/>

1.6 Summary

This chapter describes a typical OS/2 Server environment and the starting point for our migration scenarios. Though each OS/2 Server deployment is unique and will vary somewhat from the environment described here, we have chosen a common configuration and product set that should be familiar to most readers.

In the next chapter, we describe the target environments for our migration scenarios.



Target platforms

This chapter provides a detailed overview of the setup for the given target platforms used in the book. This includes a Windows 2000 Server, Red Hat, and SuSE Linux server distributions. For the Linux platforms, it also includes a discussion on the configuration of Samba 3.0.

In this chapter, the following topics are discussed and described:

- ▶ Setup of the Windows 2000 Server with Active Directory
- ▶ Setup of Red Hat Enterprise Server and SuSE Linux Enterprise Server along with Samba 3 and OpenLDAP
- ▶ Installation of IBM middleware products on both platforms

2.1 Windows 2000 as a target platform

The following section will describe the installation of a Windows 2000 Server implementation as the target platform of the migration. It will cover the major steps during installation, while configuration and importing settings and definitions will be done in Part 2 of this redbook.

As part of the migration scenario, a native Windows 2000 Active Directory Services (ADS) domain is created with having one's eye on a state-of-the-art domain concept, but keeping most of the services untouched to describe a migration rather than a redesign or consolidation scenario.

Note: At the time of the writing this book, Microsoft released the Windows 2003 Server. Because of the lack of experience in customer scenarios, we still focus on Windows 2000 Server, and give annotations to the reader for changes or enhancements in the new server release of which we are aware.

2.1.1 Base installation

The base operating system installation is described using Service Pack 3 without any additional hot fixes. Both servers are deployed through an unattended installation, providing only the base services necessary for any type of server. All additional services for a specific server are installed in distinct steps afterwards. To keep things simple, software deployment or distribution system is not used. Rather, commands together with response files are utilized. They can be easily embedded into an existing deployment mechanism. The response files that are used can be found in Appendix A, "Windows 2000 migration related scripts" on page 411.

The base level for both systems is as follows:

- ▶ Windows 2000 Server SP3
- ▶ Drive C: Provides a maintenance system installation on a 2 GB NTFS partition
- ▶ Drive D: Provides the production system installation on a 4 GB NTFS partition
- ▶ Drive E: Contains applications and data on at least a 4 GB NTFS partition
- ▶ All base operating system components are installed on drive D:, the application stack (IBM Universal Database, IBM Communication Server, Lotus Domino, and so on) is installed on drive E:.

Unattended installation

The installation for the base services is outlined as follows. All files used can be found in Appendix A, “Windows 2000 migration related scripts” on page 411, and used as a template for individual CID installations if desired:

1. Boot the network enabled DOS boot diskette.
2. Partition a 2 GB C: drive with FAT on the local hard drive. This FAT partition will be converted automatically to NTFS by the Windows 2000 installation.
3. Start the DOS-based installation with the following commands, where `xfer1` is the name of the CID server:

```
net use r: \\xfer1\rsp
net use t: \\xfer1\w2ksrv
t:\i386\winnt /s:t:\i386 /u:r:\%SrvName%\%SrvName%.txt
```

This step creates a maintenance partition.

4. After installation, the script is started as a **run once command** performing the following steps:
 - a. Create partitions D: and E: on local hard drive.
 - b. Format these partitions with NTFS.
 - c. Start the installation of the production environment with the command:
`\\xfer1\w2ksrv\WINNT32.EXE /s:\\xfer1\w2ksrv\ /tempdrive:d:\ /unattend5:\\xfer1\rsp\%computername%p.txt`
 - d. Install additional services depending on the role of the server.

Functions on the domain controller WINDC

After the installation of base services, the following services are added step-by-step to the domain controller, matching the definitions of the original OS/2 domain:

- ▶ File and print services as provided by Windows 2000 Server
- ▶ Logon services using Windows 2000 Active Directory services
- ▶ Replication services using DFS™ (distributed file services) and NTFRs (NT File replication services)
- ▶ IP services:
 - Windows 2000 DHCP server
 - Windows 2000 DNS server
 - Windows 2000 FTP server as a part of the IIS
 - Windows 2000 RRAS server (dial in only)
 - Windows Services for UNIX (to provide NFS server services)
 - Software stack
 - IBM Tivoli Storage Manager Client for Windows Version 5, Release 1, Level 5

- Lotus Domino Server 5.0.12
- IBM eNetwork Communication Server 6.1
- Tivoli Endpoint
- IBM Director 4.1

File, print, and replication services are part of the base operating system.

Functions on the member server WINMEM

The member server provides the following services, consistent with the definitions of the original OS/2 domain. Deviating from the original OS/2 domain, the role of a member server is used since the authorization aspect differs from domain to member servers in Windows 2000 domains:

- ▶ File and print services as provided by Windows 2000 Server
- ▶ Replication services using DFS (distributed file services) and NTFrs (NT File replication services)
- ▶ IP services:
 - Windows 2000 FTP server as a part of the IIS
 - Windows 2000 Web publishing services as part of the IIS
 - Windows Services for UNIX (including NFS)
- ▶ Software stack
 - IBM Tivoli Storage Manager Client for Windows Version 5 Release 1, Level 5
 - Tivoli Endpoint
 - IBM Director 4.1
 - IBM DB2 Universal Database 7.2

2.1.2 FTP server

FTP services can also be installed using an unattended installation method. This will only install the service itself on the server, but gives no option to configure it. Therefore, after installation it is necessary to carry out additional steps for creating the virtual directories and the user accounts the server should provide. This is discussed in more detail in 4.11, “Migrating OS/2 FTP server to Windows 2000” on page 163.

The sample response file provided below lets the Windows installer install the core components of Internet Information Server (IIS), the FTP components, and the management console (MMC). Additionally, it defines the directory E:\FTP as the root directory for FTP sites. The following line starts the installation of the FTP server:


```
sysocmgr /i:%WINDIR%\INF\SYSOC.INF /u:instftp.txt
```

Example 2-1 Response file for FTP server service (instftp.txt)

```
[Components]
iis_common = on
iis_ftp = on
iis_inetmgr = on

[InternetServer]
pathFTPRoot = "E:\FTP"
```

2.1.3 DHCP server

This optional service is also installed through unattended installation with the following command given the response file shown below. The parameters are discussed later as part of the migration process. This command starts the installation:

```
sysocmgr /i:%WINDIR%\INF\SYSOC.INF /u:instdhcp.txt
```

Example 2-2 Response file for DHCP Server service (instdhcp.txt)

```
[NetOptionalComponents]
DHCP=1
```

2.1.4 WINS server

For WINS Services, the same method is used for installation. Parameters are covered in detail later as part of the migration process. This command starts the installation:

```
sysocmgr /i:%WINDIR%\INF\SYSOC.INF /u:instwins.txt
```

Example 2-3 Response file for WINS Server service (instwins.txt)

```
[NetOptionalComponents]
WINS = 1
```

2.1.5 DNS server

Before installing Active Directory services, DNS server services are required to be installed. Although the Active Directory wizard *dcpromo* optionally installs DNS Server services when not already installed, still it is good practice to do it intentionally, since that way the proper parameters can be specified. Again, this

service is installed through unattended installation with the following command given the response file shown in Example 2-4.

```
sysocmgr /i:%WINDIR%\INF\SYSOC.INF /u:instdns.txt
```

Example 2-4 Response file for DNS Server service (instdns.txt)

```
[NetOptionalComponents]
DNS = 1
```

2.1.6 Active Directory services

The promotion of the WINDC server is done after the initial installation of the operating system through unattended execution of **dcpromo**. This program can be scripted with the response file listed in Example 2-5.

```
dcpromo /answer:dcpromo.txt
```

Example 2-5 Response file for dcpromo (dcpromo.txt)

```
[DCINSTALL]
ReplicaOrNewDomain=Domain
TreeOrChild=Tree
CreateOrJoin=Create
NewDomainDNSName=somedomain.local
DNSOnNetwork=yes
DomainNetbiosName=SOMEDOMAIN
AutoConfigDNS=yes
SiteName=CENTRAL
AllowAnonymousAccess=no
DatabasePath=e:\ntds
LogPath=e:\ntds
SYSVOLPath=e:\sysvol
; *****
; Password entry will be deleted after executing DCPROMO
; *****
SafeModeAdminPassword=password
CriticalReplicationOnly=No
RebootOnSuccess=Yes
```

This response file accomplishes the following results:

- ▶ A new domain is created with the DNS name `somedomain.local`.
- ▶ NetBIOS name is set to `SOMEDOMAIN`.
- ▶ It is the first domain in the first tree.
- ▶ DNS services will not be installed, but necessary entries are added.
- ▶ The server is joined to the site `CENTRAL`.

- ▶ Pre-Windows 2000 Compatible Access is granted (line AllowAnonymousAccess=yes) to allow OS/2 clients to successfully log on.
- ▶ The installation does not use the default directories, but installs all databases on drive E:
- ▶ For safe mode restore, the password is set to password
- ▶ After successful promotion, the system automatically reboots.

After successfully promoting the domain controller, set the domain to native mode since there is no need to join any backup domain controllers running Windows NT® 4.0 into this domain.

For further details please read the Microsoft Knowledge Base articles, especially the article *Unattended Promotion and Demotion of Windows 2000 Domain Controllers, Q223757*.

Important: For security reasons every time **dcpromo** processes the response file, it deletes password entries. Please review the content of the file before starting **dcpromo** especially for the key SaveModeAdminPassword, because the program also allows blank passwords.

2.1.7 Certificate service

To enable secure access to LDAP, install the Certificate Service. Again, we used the feature for unattended installation provided by Microsoft using the following command:

```
sysocmgr /i:%WINDIR%\INF\SYSOC.INF /u:instdhcp.txt
```

Example 2-6 Response file for Certificate Service (instcertsrv.txt)

```
[Components]
certsrv = on
certsrv_client = on
certsrv_server = on

[Certificatesrv_client]
CAmachine = windc.somedomain.local
CAName = WINDC

[Certificatesrv_server]
CAType = EnterpriseRoot
Country = US
CSPPProvider = "Microsoft Base Cryptographic Provider v1.0"
Description = "Certificate server for Somedomain"
HashAlgorithm = "SHA1"
Locality = "Austin"
```

```
Name = WINDC
Organization = Some Company
OrganizationUnit = IT
PreserveDB = No
SharedFolder=E:\CAConfig
State = Tx
UseExistingCert = No
ValidityPeriod = 2
ValidityPeriodUnits = Years
```

As listed in the response file instcertsrv.txt shown in Example 2-6, a root certificate server using Active Directory Services (CAType=EnterpriseRoot) is installed and configured with these given parameters. To enable the LDAP server listening on SSL-port 634, a group policy to enable Automatic Certificate Requests for domain controllers is created. To do this within the Group policy object *Default domain controllers policy*, use the option **Computer Configuration -> Windows Settings -> Security settings -> Public key Policies -> Automatic Certificate Request Settings**. Within this container select **New -> Automatic Certificate Request**. Follow the wizard and select **Domain Controller** as the template and **WINDC** as the only available certificate authority. After about 5 minutes the domain controllers will apply the new policy and listen to port 634. To speed this up, use the following command:

```
secedit /refreshpolicy machine_policy /enforce
```

2.2 Software stack on Windows 2000

The following sections describe setting up various additional software on Windows 2000 Servers.

2.2.1 IBM Universal Database

The source OS/2 platform in our scenario uses IBM Universal Database or DB2/2 v7.2. To provide the intended one-to-one mapping, the current version 7.2 of IBM DB2 for Windows was installed. The basic installation consists of the following steps:

1. Create the user ID db2admin. This might include additional permissions for this user depending on your security policy and the services this account should run:

```
net user db2admin password /add /comment:"System account for IBM DB2"
```

2. Connect to installation sources with the NET USE command, since **setup.exe** currently does not support UNC path names:

```
NET USE X: \\xfer1\img /persistent:no
```

```
NET USE R: \\xfer1\rsp /persistent:no
```

3. Within the distribution package, IBM delivers templates for unattended installation. The one modified in this sample is located in the subdirectory db2\common named db2udbee.rsp. Copy this file into the response file directory, name it dsp.rsp, and modify it to match your requirements.
4. Run the installation using the provided response file db2.rsp (see Appendix , “DB2.RSP” on page 435):

```
x:\db2\setup /u r:\db2.rsp /l %systemdrive%\db2.log /I en
```

2.2.2 IBM e-Network Communications Server

The source OS/2 platform has IBM Communication Server installed as part of the migration scenario. To provide the intended one-to-one mapping, IBM Communication Serer 6.11 was installed to the target domain controller. The basic installation is done by providing a response file for an unattended installation. One parameter in the response file specifies an administrative group that will be granted the privilege to manage the Communication Server. A group called CSADMINS is created for that purpose before starting the installation:

```
NET LOCALGROUP CSAdmins /add /comment:"Administrators for IBM Communications Server"
```

Note: As the Active Directory is not completely configured, we use the Windows NT backward command **net localgroup** to add this group. After finishing the installation, it is recommended to check users and groups in the default container, and move them to an appropriate location.

The installation program for Communication Server does not yet allow the usage of UNC path names for installation, so the necessary resources have to be attached with the **NET USE** command:

```
NET USE X: \\xfer1\img /persistent:no  
NET USE R: \\xfer1\rsp /persistent:no
```

The last step is to start the setup program to run the installation using the provided response file cs.iss in Appendix , “CS.ISS” on page 429:

```
x:\cs\setup /s /f1r:\cs.iss /f2%SYSTEMDRIVE%\cs.log
```

2.2.3 Lotus Domino

For running Lotus Domino Server on the target platform, release 5.0.12 was current, and therefore used in this book. The installation of this product is once again using **setup.exe** and a response file with the necessary options:

```
%img%\notes\501\setup /s /f1%rsp%\notes.iss /f2%SYSTEMDRIVE%\Notes.log
```

2.2.4 IBM HTTP Server

To install the IBM HTTP Server on Windows, you first have to obtain the Java Developer Kit 1.3 from IBM, which is available at the IBM Developers Web site. Be sure to install all parts of the JDK before installing the HTTP Server. In our example, we are using the IBM HTTP Server version 1.3.26.1, which is available at <http://www-3.ibm.com/software/webservers/httpservers/> This version is very similar to version 1.3.20 on the OS/2 source platform. Once IBM Java is installed you can proceed with installing the HTTP Server for Windows.

Note: Install the Java JDK before you install the Web server.

To install this version, open a command prompt and change to the directory where the install package exists. Now you can type `java -jar setup.jar` and you will be guided through the installation process by the Java installer.

2.2.5 Tivoli Storage Manager Client

We installed Tivoli Storage Manager Client in its current release 5.1.5.0. for the functions that the IBM Adstar Storage Manager provides on the OS/2 platform. The client is delivered as a Microsoft Installer Package (MSI) that can be scripted for unattended installation. The file readme.1st can be found in the distribution package and gives you more details for silent installation commands. We used the following command:

```
msiexec /i "\\xfer1\img\tsm\Tivoli Storage Manager Client.msi"  
RebootYesNo="No" REBOOT="Suppress" ALLUSERS=1  
INSTALLDIR="%PROGRAMFILES%\Tivoli\TSM"  
ADDLOCAL="BackupArchiveGUI,ApiRuntime,BackupArchiveGuiDeu,AdministrativeCmd"  
TRANSFORMS=1033.mst /qn /!v "%SYSTEMDRIVE%\tsm.log"
```

After this you can copy a pre-configured option file (dsm.opt) or the migrated configuration from 5.5, "Migrating TSM Client" on page 185 to configure the client.

```
COPY \\xfer1\rsp\dsm.opt "%ProgramFiles%\Tivoli\TSM\baclient"
```

2.3 Red Hat Linux as a target

The following section covers the installation of Linux Red Hat Enterprise Server and how to install the additional options and software that is required. There are many ways to install and run programs on Linux, and only one of them is covered here as an example. Depending on a company's software installation policy and procedures, a different mechanism might be used.

2.3.1 Base installation

For Red Hat Enterprise Linux ES, there are two types of installation: attended and unattended. In the following, the features of the unattended OS installation are described.

The program that is used when installing Red Hat is called *anaconda*. Also, there is a program called *kickstart* that uses the script language of anaconda to produce an easy and unattended installation mode. It is very useful in large Linux environments or deployments when an attended installation is time consuming and inefficient.

The Linux operating system can be installed from any of the following sources:

- ▶ CD-ROM media
- ▶ Hard disk
- ▶ FTP server
- ▶ NFS server
- ▶ HTTP server

All the installation modes support kickstart installation (unattended) and normal (attended) installation.

There are two approaches to a kickstart installation, one is to simply copy your kickstart configuration file to a Red Hat boot floppy diskette. The other is to use a regular boot floppy and get your kickstart configuration file from the network. A sample of a kickstart configuration file is shown in Example 2-7.

Note: The kickstart configuration file can be easily modified to install several servers with the same base kickstart configuration. At times only the IP address differs from one server to another.

The following partitions were created.

Partition Name	Description
/boot	The boot partition where the kernel and initrd files are. It is recommended that you create a separate boot partition in case your root file system is built on a software RAID or LVM subsystem. In some cases, you may not be able to boot the system if you do not create a separate boot partition.
/(root)	All the OS data are in the root partition. Red Hat by default supports only ext2 and ext3 file systems on root partitions.

Partition Name	Description
/opt	All of the non-built in software is installed here , such as IBM DB2, Lotus Domino, IBM HTTP, and so on.

Note: It might be required, or at least convenient to create different partitions based on your experience, knowledge, and your applications. In fact, for a production server it is recommended to create the partitions and file system based on the application documentation.

For more information and tips regarding the kickstart installation please read:
<http://www.tldp.org/HOWTO/KickStart-HOWTO.html#toc6>

Example 2-7 Kickstart configuration file

```
install
lang en_US
langsupport --default en_US en_US
keyboard us
mouse generic3ps/2 --device psaux --emulthree
xconfig --card "S3 Trio3D" --videoram 4096 --hsync 30.0-85.0 --vsync 50.0-150.0
--resolution 1024x768 --depth 16
network --device eth0 --bootproto static --ip 9.3.4.15 --netmask 255.255.254.0
--gateway 9.3.4.41 --hostname lnxrh
rootpw --iscrypted $1$ÊkIÎÇæT0$IdYJhrTSI1TEZWy0Up0.j0
firewall --disabled
authconfig --enablesshadow --enablemd5
timezone America/Monterrey
bootloader
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
#clearpart --linux
#part /boot --fstype ext2 --onpart hda1
#part /opt --fstype ext3 --onpart hda4
#part / --fstype ext3 --onpart hda2
#part swap --onpart hda3

%packages
@ Printing Support
@ Classic X Window System
@ X Window System
@ GNOME
@ KDE
@ Sound and Multimedia Support
```


@ Network Support
@ Dialup Support
@ Messaging and Web Tools
@ Graphics and Image Manipulation
@ News Server
@ NFS File Server
@ Windows File Server
@ Anonymous FTP Server
@ SQL Database Server
@ Web Server
@ Router / Firewall
@ DNS Name Server
@ Network Managed Workstation
@ Authoring and Publishing
@ Emacs
@ Utilities
@ Software Development
@ Kernel Development
@ Server
balsa
kdenetwork
esound-devel
kdemultimedia
compat-libstdc++
arpwatch
mozilla-mail
VFlib2-devel
gaim
qt-devel
firewall-config
shapcfcg
glade
libesmtplib
ddd
rsync
IBMJava2-SDK
magicdev
pdksh
libgtop-devel
w3c-libwww
libpcap
gnome-pim-devel
SDL
libghttp-devel
gdk-pixbuf-devel
gnome-lokkit
asp2php-gtk
pan
psgml

sane-backends-devel
mtr-gtk
freetype-devel
libogg-devel
gnome-core-devel
rhn_register-gnome
lesstif-devel
XFree86-devel
kdesdk
doxygen
libunicode-devel
timidity++
xawtv
openmotif-devel
libglade-devel
apache-devel
libvorbis-devel
kdeaddons-noatun
nmap-frontend
unixODBC-devel
oaf-devel
up2date-gnome
gsm-devel
tetex-xdvi
GConf-devel
qt-designer
kdoc
vnc
libxml-devel
xisdnload
bonobo-devel
kdenetwork-ppp
kdelibs-devel
autorun
Xaw3d-devel
ORBit-devel
fam-devel
exmh
kdevelop
kdegraphics
gnome-media
vnc-server
dhcp
librep-devel
control-center-devel
htmlview
php-imap
pvm-gui
openssh-askpass-gnome

```
IBMJava2-JRE
redhat-config-network
libmng-devel
netscape-communicator
XFree86-SVGA
libungif-devel
xmms-gnome
memprof
bindconf
apacheconf
gcc-objc
glib-devel
kdelibs-sound-devel
pilot-link-devel
emacs-X11
kpppload
Mesa-devel
kdesdk-devel
netpbm-devel
xpdf
gimp-devel
libao-devel
XFree86-compat-modules
imlib-devel
xmms
kdbg
openssh-askpass
bind-devel
ical
gnome-libs-devel
audiofile-devel
usbview
netscape-common
cdrecord-devel
php-pgsql
galeon
gq
gtk+-devel

%pos
```

2.3.2 FTP server

The Red Hat ES v.2.1 has a built in FTP server called *wu-ftpd* (optional package). It is easy to configure and is quite flexible. The configuration files are:

- ▶ `/etc/ftppass`: The main configuration file

- ▶ `/etc/ftpusers`: The `ftpusers` file is deprecated. Use `deny-uid/deny-gid` in `ftppass`
- ▶ `/etc/hosts.allow`: This file describes the names of the hosts that are allowed to use the local INET services, as decided by the `/usr/sbin/tcpd` server.
- ▶ `/etc/hosts.deny`: This file describes the names of the hosts which are *not* allowed to use the local INET services, as decided by the `/usr/sbin/tcpd` server.

To start or stop the FTP daemon, use the script `/etc/init.d/xinetd` with the parameters **start** or **stop**.

For more information please read the Linux man pages for the `/etc/ftppass` file.

2.3.3 NFS server

Red Hat has a built-in NFS server called *nfs* server (optional package). It is easy to use, fast, and reliable. The configuration files are:

- ▶ `/etc/exports`
- ▶ `/etc/hosts.allow`: This file describes the names of the hosts that are allowed to use the local INET services, as decided by the `/usr/sbin/tcpd` server.
- ▶ `/etc/hosts.deny`: This file describes the names of the hosts which are *not* allowed to use the local INET services, as decided by the `/usr/sbin/tcpd` server.

To start or stop the `nfs` daemon, use the script `/etc/init.d/nfs` with parameters **start** or **stop**.

For more information please read:

<http://www.ibiblio.org/pub/Linux/docs/HOWTO/NFS-HOWTO>

2.3.4 DNS server

Red Hat has a built in DNS server (optional package), the package is called *bind-9.2.x*. Since version 9.2.x, the DNS server supports dynamic DNS functions. The configuration files are:

- ▶ `/etc/named.conf`: The main configuration file
- ▶ `/var/named/`: The directory where the zone files are kept.

To start or stop the DNS daemon, use the script `/etc/init.d/named` with parameters **start** or **stop**.

For more information please read:

<http://www.ibiblio.org/pub/Linux/docs/HOWTO/DNS-HOWTO>

2.3.5 DHCP server

The Red Hat ES v.2.1 has a built in DHCP version 2.0p15-8, called *dhcp.x*.

Note: On your Linux server you may find a package called `dhcpcd-x`. This is the DHCP client daemon package.

This version of the DHCP server does not have the function to update the DNS dynamically. If you want the dynamic DNS function to work on the Linux server, you have to upgrade the DHCP package at least to the version 3.0.1rc7, which can be found at:

<ftp://ftp.rpmfind.net/linux/redhat/8.0/en/os/i386/RedHat/RPMS/dhcp-3.0p11-9.i386.rpm>

After you download the package, you can update the DHCP package by running the command:

```
rpm -Uvh <path>/dhcp-3.0p11-9.i386.rpm
```

You can start or stop the DHCP daemon by using the script `/etc/init.d/dhcpd` with the parameters `start` or `stop`.

Note: The DHCP server has a daemon called the DHCP relay agent. It is used when the Linux server acts as a router between two or more networks segments, and relays the DHCP packets. For more information about DHCP relay please read:

<http://download.freeswan.ca/x509patches/dhcprelay/ipsec-dhcp-howto-4.html>

For more information about DHCP please read:

<http://www.tldp.org/HOWTO/mini/DHCP/>

2.3.6 Samba server

The Red Hat server has a built-in Samba server (optional package). The built in version is 2.2.7. At the time of writing this book, the Samba 3.0.0b1 package became available. Because the Samba 3.x has many features specially relating to Windows Active Directory and authentication with LDAP, we chose to use the Samba 3.0.0b1. By the time you read this book there should be a stable Samba 3.x version.

Please see 2.6, “Samba and OpenLDAP” on page 51 for more details on Samba 3.

2.4 SuSE Linux as a target

The following section covers the installation of Linux with the SuSE Linux Enterprise Server (SLES) on your server, and information on how to install the programs that you need. There are many ways to install and run programs on Linux SuSE, and one of them is presented as an example.

2.4.1 Base installation

For SuSE Linux Enterprise Server (SLES) RC5, there are two types of installation: attended and unattended. In the following, the features of the unattended OS installation are described.

The SuSE distribution has a tool for unattended installation called *yast2*. To automate tasks, the *yast2* tool can run with the parameter *autoyast*. This tool can be used to configure the unattended installation. The configuration is saved in a file. To install SuSE in this way, follow these steps:

1. Copy the configuration file that was created with *yast2* as above from the repository directory on the local hard disk to a floppy disk with the name *autoinst.xml*.
2. Put the floppy disk with the configuration file in the target machine.
3. Turn on the target machine, ensure that the CD drive is listed in the boot list of your BIOS, and insert the SLES CD1. The normal boot menu of the SuSE installation program is displayed. As an alternative to booting from CD, Linux provides the options to boot from floppy images, from the network, or using whichever method is normally used to boot the installation program.
4. At the boot menu, leave the default line as Linux to do the standard boot, but add the following parameters in order to read your configuration file from the floppy disk:

```
linux autoyast=floppy
```
5. Your server should now boot the installation program, and it will try to load appropriate modules and install the system with the information that you provided in the configuration file.

A configuration file is shown in Example 2-8.

The following partitions were created during installation.

Partition name	Description
/boot	The boot partition where the kernel and initrd files are. It is recommended to create a separate boot partition in case your root file system is built on a software RAID or LVM subsystem. In some cases, you may not be able to boot the system if you do not create a separate boot partition.
/(root)	All the OS data are in root partition. SuSE, by default, supports only ext2 and ext3 file systems on the root partition.
/opt	All of the non-built in software is installed here, such as IBM DB2, Lotus Domino, IBM HTTP, etc.

Note: It might be required or at least convenient to create different partitions based on your experience, knowledge, and your application. In fact, for a production server it is recommended to create the partitions and file system based on the application documentation.

For more information and tips regarding the unattended installation please read: <http://www.tldp.org/HOWTO/Network-Install-HOWTO-5.html>

Example 2-8 Autoyast2 configuration file

```
<?xml version="1.0"?>
<!DOCTYPE profile SYSTEM "/usr/share/YaST2/include/autoinstall/profile.dtd">
<profile xmlns="http://www.suse.com/1.0/yast2ns"
xmlns:config="http://www.suse.com/1.0/configns">
  <configure>
    <inetd>
      <inetd_services config:type="list">
        <inetd_service>
          <service_name></service_name>
          <service_status>enable</service_status>
        </inetd_service>
        <inetd_service>
          <service_name>telnet</service_name>
          <service_status>enable</service_status>
        </inetd_service>
      </inetd_services>
      <start_inetd config:type="boolean">true</start_inetd>
    </inetd>
    <networking>
      <dns>
        <dhcp_hostname config:type="boolean">false</dhcp_hostname>
        <dhcp_resolv config:type="boolean">false</dhcp_resolv>
        <domain></domain>
      </dns>
    </networking>
  </configure>
</profile>
```

```

    <hostname></hostname>
    <nameservers config:type="list"/>
    <searchlist config:type="list"/>
</dns>
<interfaces config:type="list"/>
<routing>
    <ip_forward config:type="boolean">false</ip_forward>
    <routes config:type="list"/>
</routing>
</networking>
<scripts>
    <chroot-scripts config:type="list"/>
    <post-scripts config:type="list"/>
    <pre-scripts config:type="list"/>
</scripts>
<security>
    <console_shutdown>ignore</console_shutdown>
    <cwd_in_root_path>no</cwd_in_root_path>
    <displaymanager_remote_access>no</displaymanager_remote_access>
    <encryption>des</encryption>
    <fail_delay>3</fail_delay>
    <faillog_enab>yes</faillog_enab>
    <gid_max>60000</gid_max>
    <gid_min>101</gid_min>
    <kdm_shutdown>root</kdm_shutdown>
    <lastlog_enab>yes</lastlog_enab>
    <obscure_checks_enab>no</obscure_checks_enab>
    <pass_max_days>99999</pass_max_days>
    <pass_max_len>8</pass_max_len>
    <pass_min_days>1</pass_min_days>
    <pass_min_len>6</pass_min_len>
    <pass_warn_age>14</pass_warn_age>
    <passwd_use_cracklib>yes</passwd_use_cracklib>
    <permission_security>secure</permission_security>
    <run_updatedb_as>nobody</run_updatedb_as>
    <uid_max>60000</uid_max>
    <uid_min>500</uid_min>
</security>
<users config:type="list">
    <user>
        <encrypted config:type="boolean">>true</encrypted>
        <user_password>password</user_password>
        <username>root</username>
    </user>
</users>
<x11>
    <color_depth config:type="integer">16</color_depth>
    <configure_x11 config:type="boolean">false</configure_x11>
    <display_manager>kdm</display_manager>

```



```

<enable_3d config:type="boolean">false</enable_3d>
<monitor>
  <display>
    <max_hsync config:type="integer">107</max_hsync>
    <max_vsync config:type="integer">160</max_vsync>
    <min_hsync config:type="integer">30</min_hsync>
    <min_vsync config:type="integer">50</min_vsync>
  </display>
  <monitor_device>6558 P202</monitor_device>
  <monitor_vendor>IBM</monitor_vendor>
</monitor>
<resolution>1024x768</resolution>
<start_x11 config:type="boolean">false</start_x11>
</x11>
</configure>
<install>
  <bootloader>
    <activate config:type="boolean">false</activate>
    <kernel_parameters></kernel_parameters>
    <lba_support config:type="boolean">false</lba_support>
    <linear config:type="boolean">false</linear>
  </bootloader>
  <general>
    <clock>
      <hwclock></hwclock>
      <timezone>US/Central</timezone>
    </clock>
    <keyboard>
      <keymap>english-us</keymap>
    </keyboard>
    <language>en_US</language>
    <mode>
      <confirm config:type="boolean">true</confirm>
      <forceboot config:type="boolean">false</forceboot>
      <interactive_boot config:type="boolean">true</interactive_boot>
      <reboot config:type="boolean">true</reboot>
    </mode>
    <mouse>
      <id>00_ps2</id>
    </mouse>
  </general>
  <init>
    <domain></domain>
    <gateway></gateway>
    <info_file>
<![CDATA[
#
# Don't remove the following line:
# start_linuxrc_conf

```

```

#
instmode: cd
Usedhcp: 1

# end_linuxrc_conf
# Do not remove the above comment
#]]>
  </info_file>
  <instmode>cd</instmode>
  <ip></ip>
  <keytable></keytable>
  <language></language>
  <nameserver></nameserver>
  <netmask></netmask>
  <partition></partition>
  <port></port>
  <profile_location></profile_location>
  <profile_port></profile_port>
  <profile_protocol>file</profile_protocol>
  <profile_server></profile_server>
  <server></server>
  <serverdir></serverdir>
  <textmode config:type="boolean">>false</textmode>
  <usedhcp config:type="boolean">>true</usedhcp>
  <workdomain></workdomain>
</init>
<partitioning config:type="list">
  <drive>
    <device>/dev/hda</device>
    <initialize config:type="boolean">>false</initialize>
    <partitions config:type="list"/>
    <use>all</use>
  </drive>
</partitioning>
<software>
  <addons config:type="list">
    <addon>YaST2_modules</addon>
    <addon>Basis-Sound</addon>
    <addon>auth</addon>
    <addon>mail_news</addon>
    <addon>LAMP</addon>
    <addon>X11</addon>
    <addon>file_print</addon>
    <addon>sles_admin</addon>
    <addon>SuSE-Documentation</addon>
    <addon>analyze</addon>
    <addon>Kde-Desktop</addon>
    <addon>dhcp_dns</addon>
    <addon>Gnome</addon>
  </addons>
</software>

```

```
</addons>
<base></base>
<packages config:type="list"/>
</software>
</install>
</profile>
```

2.4.2 FTP server

The SuSE SLES has a built in FTP server called *vs-ftp* (optional package). It is easy to configure and flexible in configuration. The configuration files are:

- ▶ `/etc/vsftpd.conf`: The main configuration file
- ▶ `/etc/ftpusers`: The ftpusers file is deprecated. Use `deny-uid/deny-gid` in `ftppass`.
- ▶ `/etc/hosts.allow`: See `man tcpd` and `man 5 hosts_access` for a detailed description of `/etc/hosts.allow`.
- ▶ `/etc/hosts.deny`: See `man tcpd` and `man 5 hosts_access` for a detailed description of `/etc/hosts.deny`.

To start or stop the FTP daemon, use the script `/etc/init.d/inetd` with the parameters `start` or `stop`.

2.4.3 NFS server

SuSE has a built-in NFS server (optional package). It is easy to use, fast, and reliable. The configuration files are:

- ▶ `/etc/exports`.
- ▶ `/etc/hosts.allow`, see `man tcpd` and `man 5 hosts_access` for a detailed description of `/etc/hosts.allow`.
- ▶ `/etc/hosts.deny`, see `man tcpd` and `man 5 hosts_access` for a detailed description of `/etc/hosts.deny`.

You can start or stop the NFS daemon by using the script `/etc/init.d/nfs` with the parameters `start` or `stop`.

For more information please read:

<http://www.ibiblio.org/pub/Linux/docs/HOWTO/NFS-HOWTO>

2.4.4 DNS server

SuSE has a built-in DNS server (optional package), the package is called *bind-9.1.x*. The configuration files are:

- ▶ `/etc/named.conf`: The main configuration file
- ▶ `/var/named/`: The directory where the zone files are kept.

You can start or stop the DNS daemon by using the script `/etc/init.d/named` with parameters **start** or **stop**.

If you want to use dynamic DNS service with SuSE, it has to be upgraded to a version 9.2.0 or newer. In the following, the steps to compile the latest DNS version 9.2.2 available at the time of writing of this the book are called `bind-9.2.2.tar.gz` and downloaded from:

<http://www.isc.org/products/BIND/bind9.html>

To compile the code follow these steps:

1. `tar zxvf bind-9.2.2.tar.gz`
2. `cd bind-9.2.2.`
3. `./configure --prefix=/opt/bind-9.2.2`
4. `make`
5. `make install`
6. `mkdir -p /opt/bind-9.2.2/var/run`
7. `cd /opt/bind-9.2.2/sbin`
8. `./named -c <path>/named.conf`

Tip: To successfully compile and configure the development environment provided on the SLES, CDs is a required installation.

To start or stop the DNS you have the following options:

- ▶ Start by running `./named -c <path>/named.conf` and stop it by `kill -2 <binds_pid>`
- ▶ Modify the `/etc/inet.d/named` script to suit the new binary path.
- ▶ Build your own script.

For more information please read:

<http://www.ibiblio.org/pub/Linux/docs/HOWTO/DNS-HOWTO>

2.4.5 DHCP server

SuSE has a built in DHCP version 3.0.1rc9-43, called *dhcp.x*.

Note: On your Linux server you may find a package called `dhcpcd-x`. This is the DHCP client daemon package.

This version of the DHCP server updates the DNS dynamically with no known problems.

The configuration file is: */etc/dhcpd.conf*

You can start or stop the DHCP daemon by using the script `/etc/init.d/dhcpd` with parameters `start` or `stop`.

Note: The DHCP server has a daemon for called DHCP relay agent. It is used when your Linux server acts as a router between two or more networks segments, and relays the DHCP packets. For more information about DHCP relay please read:

<http://download.freeswan.ca/x509patches/dhcrelay/ipsec-dhcp-howto-4.html>

For more information about DHCP please read:

<http://www.tldp.org/HOWTO/mini/DHCP/>

2.4.6 Samba server

The SuSE Server has a built in Samba server (optional package). The built in version is 2.2.5-107. At the time of writing this book, the Samba 3.0.0b1 package became available. Because Samba 3.x has many features especially relating to Windows Active Directory and authentication with LDAP, we chose to use Samba 3.0.0b1. At the time you will read this book there should be a stable Samba 3.x version available.

Please see 2.6, “Samba and OpenLDAP” on page 51 for more details on Samba 3.

2.5 Software stack on Linux

In the following sections, we describe how to install various software applications on Linux. Most of the applications are installed in a similar way on both the Red Hat distribution and SuSE distribution. If there are differences in installation procedures between Red Hat and SuSE, we highlight the difference.

2.5.1 IBM Universal Database

In the following we describe the installation process for IBM Universal Database v7.2 on Red Hat ES v.2.1 and SuSE SLES RC5.

Software requirements

To properly install IBM DB2 you need:

- ▶ pdksh rpm package
- ▶ Java installed
- ▶ 500 MB free space in /usr

Software installation

IBM DB2 v7.2 uses a text-based installation method. It is easy to install DB2 on remote sites.

To install IBM DB2 log in as root and follow these steps:

1. Change to the directory with `cd <path>/<db2 directory>`
2. Run `./db2setup`
3. Select the packages that you wish to install by selecting the installation type.
4. Select the creation of the sample database if you wish.
5. Create the users that will own the database or accept the default users.
6. Install the software.

Post installation tasks

On Red Hat ES and SuSE SLES, the IBM Java package is installed by default. IBM DB2 uses Java for all its graphical tools. If one of those tools is started, it searches for the Java environment. If this cannot be found, a Java Runtime Environment (JRE) on a remote machine can be used. To do so, a `jre` command file has to be created to start Java remotely as follows: `ln -s <java_path>/java <java_path>/jre`

Also, to run `db2cc` (db2 command center), the file needs to be edited to remove the option `nojit` from the `JRE_OPTIONS` variable.

For more information about DB2 under Linux please see:

<http://www.redbooks.ibm.com> and search for DB2

2.5.2 IBM Communication Server

CS/Linux provides SNA connectivity for 32-bit Intel based Linux systems, allowing it to connect to IBM mainframe and AS/400® computers, as well as other workstations.

Software requirements

This version of CS/Linux has been tested with the following operating system versions:

- ▶ Red Hat 7.2
- ▶ Red Hat 7.2, Feb 2002 update
- ▶ Red Hat 7.2, Jun 2002 update
- ▶ Red Hat 7.2, Mar 2003 update
- ▶ Red Hat 7.3
- ▶ Red Hat 7.3, Jun 2002 update
- ▶ Red Hat 7.3, Mar 2003 update
- ▶ Red Hat 8.0
- ▶ Red Hat 8.0, Mar 2003 update
- ▶ Red Hat 9
- ▶ Red Hat 9, Apr 2003 update
- ▶ Red Hat Advanced Server 2.1
- ▶ Red Hat Advanced Server 2.1, Feb 2003 update
- ▶ Red Hat Advanced Server 2.1, Mar 2003 update
- ▶ Red Hat Enterprise Linux AS/ES/WS 2.1
- ▶ Red Hat Enterprise Linux AS/ES/WS 2.1, Mar 2003 update
- ▶ SuSE 8.0
- ▶ SuSE 8.1 Pro
- ▶ SuSE 8.1 Pro, Mar 2003 update
- ▶ SuSE 8.2 Pro
- ▶ UnitedLinux 1.0
- ▶ UnitedLinux 1.0, Service Pack 1 or Service Pack 2
- ▶ SLES 8, Mar 2003 update

Red Hat ES v 2.1

- ▶ Required:
 - kernel-headers-2.4.9-e.12
 - kernel-source-2.4.9-e.12
 - gcc-2.96-116.7.2
 - make-3.79.1-8
 - XFree86-libs-4.1.0-44
- ▶ Optional, needed for xsnaadmin:
 - XFree86-4.1.0-44
 - openmotif-2.2.2 (download)
- ▶ Optional, needed for SSL:
 - libstdc++-3-3.0.4-1 (download)
 - libgcc-3.0.4-1 (download)
- ▶ Optional, needed for JavaCPI-C and snakeyman:

- IBMJava2-SDK-1.3.1-5

SuSE SLES 8

- ▶ kernel-source-2.4.19.SuSE-175 (download)
- ▶ One of:
 - k_deflt-2.4.19-274 (download)
 - k_smp-2.4.19-257 (download)
 - k_psmpp-2.4.19-263 (download)

Linux Streams (LiS) 2.16.0

CS/Linux uses the LiS streams implementation. The 2.16.0 level of LiS is required. This package can be obtained from the following URL:

<ftp://ftp.gcom.com/pub/linux/src/LiS/LiS-2.16.0.tgz>

Copy the file LiS-2.16.0.tgz to your Linux machine in binary mode. Copy it to /usr/src.

Execute the following set of commands to install and build LiS:

```
PATH=$PATH:/sbin
cd /usr/src
tar -xzf LiS-2.16.0.tgz
cd /usr/src/LiS-2.16
make
```

Select the default answers to all of the questions:

```
#make install
```

Open Motif

CS/Linux uses the Motif implementation from the Open group at the 2.2 level.

Red Hat

For Red Hat download the Open Motif version 2.2 from:

http://ftp.motifzone.net/2.2/linux-rpm/x86/openmotif-2.2.2-3_IC.S.i386.rpm

Once you have copied the rpm file to your Linux system, issue a command such as the following to install it:

```
rpm -i --force openmotif-2.2.2-3_IC.S.i386.rpm
```

or

```
rpm -U openmotif-2.2.2-3_IC.S.i386.rpm
```


The **--force** flag would be used if you already had the **lesstif** package from Red Hat 7.2 installed, or an **openmotif21** package which was used by previous levels of CS/Linux.

The **-U** flag would be used if you already had an **openmotif-2.1** rpm installed, which was used by previous levels of CS/Linux.

SuSE

Note: The SuSE SLES 8 has Open Motif Version 2.2 already installed.

SSL

If you plan on using SSL with the CS/Linux tn3270 server, you will first need to install the optional RPMs.

Red Hat

You need the packages:

- ▶ libstdc++3-3.0.4-1
- ▶ libgcc-3.0.4-1

These packages are available at:

<ftp://rpmfind.net/linux/redhat/updates/7.2/en/os/i386/libstdc++3-3.0.4-1.i386.rpm>

<ftp://rpmfind.net/linux/redhat/updates/7.2/en/os/i386/libgcc-3.0.4-1.i386.rpm>

SuSE

You need the packages:

- ▶ libgcc-3.2-45
- ▶ libstdc++3-3.0.4-1

The libstdc++ package is available at:

<ftp://rpmfind.net/linux/redhat/updates/7.2/en/os/i386/libstdc++3-3.0.4-1.i386.rpm>

Note: If the prerequisite RPMs are already installed when CS/Linux is installed, then the gskit rpm (gsk6bas-6.0-5.27.i386.rpm) will be automatically installed at that time. In addition, various necessary updates to the Java configuration and file locations are made.

Installing and starting CS/Linux

To install CS/Linux you will need the base rpm, CS-LINUX-6.0.1.0-1.i386.rpm

To install this rpm follow these instructions:

- ▶ Log into the machine as root.
- ▶ Mount the CD and issue the following command to install CS/Linux
- ▶ For Red Hat:
 - mount /dev/cdrom
 - cd /mnt/cdrom
 - ./installcslinux
- ▶ For SuSE:
 - mount /dev/cdrom
 - cd /media/cdrom
 - ./installcslinux

You will be prompted to read and accept the license agreement, then the **installcslinux** tool will install the RPMs.

For machines with limited memory, for example, 64 MB, a reboot is required. For larger systems this may not be needed. If the CS/Linux node fails to start, check the /var/log/messages file for an entry like:

- ▶ Kernel: SNA Trace Driver can only get X blocks of memory - please reboot.

Note: If these messages persist even after rebooting you need more memory.

- ▶ Add the CS/Linux binary directories to your PATH. You may wish to change your profile to do this automatically:
 - export PATH="\$PATH:/opt/sna/bin:/opt/sna/bin/X11"
 - export LD_LIBRARY_PATH=/usr/lib:/opt/sna/lib
 - export LD_RUN_PATH=/usr/lib:/opt/sna/lib
- ▶ For Java CPI-C applications you should also set the environment variables:
 - export CLASSPATH=\$CLASSPATH:/opt/sna/java/cpic.jar
 - export LD_PRELOAD=/usr/lib/libpLiS.so
- ▶ Start CS/Linux. Note that after installation this will happen automatically when the machine is rebooted. Make sure you are not still in the CD's directories when this is done:
 - #cd /
 - #sna start
- ▶ Run the CS/Linux MOTIF administration tool. We strongly recommend you use the Motif administration program until you are familiar with CS/Linux operation. Simply follow the instructions you are given:

– #xsnaadmin &

Note: For more information, please read the README file from the installation CD.

2.5.3 Lotus Domino

Lotus Domino provides a multi-platform foundation for collaboration and e-business, driving solutions from corporate messaging to Web-based transactions, and everything in between.

Note: The Lotus Domino installation is the same for both Red Hat and SuSE.

Software requirements

The user and the group that will own Lotus Domino have to be created before the installation. To do that follow Example 2-9.

Example 2-9 Creating the Lotus Domino user

```
#groupadd notes
#useradd -g notes -d /home/notes notes
#passwd notes
```

Lotus Domino installation

Lotus Domino Server version 5.x has a command line installation process. This type of installation is useful when Lotus Domino is installed remotely through a terminal. Follow Example 2-10.

Example 2-10 Installation procedure

```
cd <path_to_installtion_dir>
./install
```

During the installation steps, you have to choose the type of server, the installation path, and the user that will own Domino. Make sure the user is created before the installation.

After the installation, you can run the server to configure it as shown in Example 2-11.

Example 2-11 Configuring the server

```
cd /opt/notesdata
/opt/lotus/bin/server
```

Note: In our lab we installed the Domino server code in /opt/lotus and the Domino server database in /opt/notesdata.

2.5.4 IBM HTTP Server

IBM HTTP Server is based on the apache server. It has the following features:

- ▶ Easy installation
- ▶ Support for SSL secure connections
- ▶ Fast Response Cache Accelerator
- ▶ IBM support as part of the WebSphere bundle
- ▶ Hardware crypto support
- ▶ Administration Server that helps to administer and configure IHS servers.
- ▶ Help information that uses the easy-to-navigate design that is common to all WebSphere products

IBM HTTP Server installation

To install the IBM HTTP Server, log in as root and follow these steps:

```
cd <path_to_http_installation_kit>
rpm -ivh gsk5bas-5.x.rpm
rpm -ivh IBM_MSG_EN-1.3.x.rpm
rpm -ivh IBM_MAN_ENU-1.3.x.rpm
rpm -ivh IBM_HTTP_Server-1.3.x.rpm
rpm -ivh IBM_ADMIN_Server-1.3.x.rpm
rpm -ivh IBM_ADMIN_EN-1.3.x.rpm
rpm -ivh IBM_FastCGI-1.3.x.rpm
rpm -ivh IBM_Machine_Translation-1.3.x.pm
rpm -ivh IBM_SSL_Base-1.3.x.rpm
rpm -ivh IBM_SSL_128-1.3.x.rpm
rpm -ivh IBM_SSL_EN-1.3.x.rpm
rpm -ivh IBM_SNMP-1.3.x.rpm
```

Post installation tasks

To be able to use the administration Web interface, a user must be created using the following commands:

```
/opt/IBMHTTP/bin/htpasswd -c /opt/IBMHTTP/conf/admin.passwd
```

For more information please visit:

<http://www-3.ibm.com/software/webservers/httpservers/>

2.5.5 Tivoli Storage Manager (TSM) Client

In the following we describe the installation of the TSM client for Linux. We will use the latest version, 5.1.5, available at the time of writing.

TSM Client installation

To install the TSM, client log in as root and follow the steps:

```
cd <path_tsmclient>/tsmcli/linux86
rpm -ivh TIVguid-1.1.0-0.i386.rpm
rpm -ivh TIVsm-API-5.1.5-0.i386.rpm
rpm -ivh TIVsm-BA-5.1.5-0.i386.rpm
```

For more installation please visit:

<http://www-3.ibm.com/software/tivoli/products/storage-mgr/>

2.6 Samba and OpenLDAP

As of the writing of this redbook, Samba's current release was Beta 1, released June 7, 2003. This section takes the reader through the process of downloading, installing, and configuring Samba, OpenLDAP, and related tools.

2.6.1 Environment overview

This overview is based on Red Hat ES. The system configuration starts with a complete (everything) install of Red Hat ES. The intent here is to identify any co-existence problems with other standard Red Hat ES services.

To begin, remove the distribution installed Samba and OpenLDAP packages. Query to determine the level installed through the following two commands:

```
# rpm -qa | grep samba
# rpm -qa | grep openldap
```

These two commands will display the package names that are currently installed on the system. The displayed names will be used to directly reference the installed package for un-installation. To uninstall the packages through the following two command:

```
rpm -ev {packageName}
```

For a complete install of Red Hat ES, the following packages need to be uninstalled, stated in the order to be removed:

- ▶ samba-swat-2.2.7-1.21as
- ▶ samba-client-2.2.7-1.21as
- ▶ samba-2.2.7-1.21as

- ▶ samba-common-2.2.27-1.21as
- ▶ openldap-servers-2.0.27-2.7.3
- ▶ openldap-clients-2.0.27-2.7.3
- ▶ openldap-devel-2.0.27-2.7.3
- ▶ openldap-2.0.27-2.7.3 (this requires an additional parameter of --nodeps to uninstall)

Additionally, ObjectREXX from IBM was downloaded and installed on the target system. Download the product from IBM at the following URL:

<http://www-3.ibm.com/software/awdtools/obj-rexx/>

2.6.2 Downloading products

For illustration purposes, all of the required products are downloaded into the /source directory and will be extracted and built in this structure. The following products are referenced here:

- ▶ OpenLDAP: Used for centralized credential and object storage.
- ▶ Berkeley DB: Used for OpenLDAP's back-end database
- ▶ Samba: Use for file and print services
- ▶ SMLDAP Tools: Used for Samba to LDAP integration enhancements

OpenLDAP

Download the source code package(s) from:

<http://www.openldap.org/software/download/>

The file openldap-2.1.21.tgz is required.

Berkeley DB

Download the source code package(s) from:

<http://www.sleepycat.com/download/index.shtml>

The file db-4.1.25.tar.gz is required.

Samba

Download the source code package(s) from:

<http://us2.samba.org/samba/ftp/beta>

The file samba-3.0.0beta1.tar.gz is required.

SMLDAP tools

Optionally, download the source code package(s) from:

<http://samba.idealx.org>

The file is `smbldap-tools-0.7.tar` and is optional.

2.6.3 Decompressing and extracting products

The products are distributed in compressed form. In the `/source` directory, the following files are now downloaded:

- ▶ `db-4.1.25.tar.gz`
- ▶ `openldap-2.1.21.tgz`
- ▶ `samba-3.0.0beta1.tar.gz`
- ▶ `smbldap-tools-0.7.tgz`

The files can be decompressed by issuing the following command:

```
# gzip -d *gz
```

This results in the following files in the `/source` directory:

- ▶ `db-4.1.25.tar`
- ▶ `openldap-2.1.21.tar`
- ▶ `samba-3.0.0beta1.tar`
- ▶ `smbldap-tools-0.7.tar`

The next step is to extract the files. In the `/source` directory, issue the following commands to extract the source trees:

- ▶ `tar -xvf db-4.1.25.tar`
- ▶ `tar -xvf openldap-2.1.21.tar`
- ▶ `tar -xvf samba-3.0.0beta1.tar`
- ▶ `tar -xvf smbldap-tools-0.7.tar`

This will result in each file creating a directory containing the source tree for each product. In each of these directories is where the configuration and compiles will occur.

2.6.4 Configuring and compiling products

At a minimum, the following products must be configured and compiled for our configuration:

- ▶ Berkeley DB
- ▶ OpenLDAP
- ▶ Samba

Building Berkeley

Berkeley DB is a prerequisite for OpenLDAP as it is used as the back-end database for OpenLDAP's object and attribute storage. To complete the build of

Berkeley DB, the first step is to configure the build environment and the second step is to build the code. From the `/source/db-4.1.25/dist` directory, the following command is issued:

```
./configure
```

The following completes the build:

```
make
```

Then finally to install the build into the correct directories, issue:

```
make install
```

The Berkeley DB should now be installed in the `/usr/local/BerkeleyDB.4.1` directory.

Building OpenLDAP

Building OpenLDAP is the next step. To complete the build of the 2.1.21 release, a number of steps are to be completed. The following steps are completed from the `/source/openldap-2.1.21` directory.

The first step is setting the required environment variables to locate the Berkeley DB libraries and include files as follows:

```
export LDFLAGS="-L/usr/local/BerkeleyDB.4.1/lib"  
export CPPFLAGS="-I/usr/local/BerkeleyDB.4.1/include"
```

The second step is to configure the build environment issuing the following command:

```
./configure {options}
```

The following options were used as parameters to the configure command:

```
--with-threads  
--enable-syslog  
--enable-crypt  
--enable-ldap  
--enable-ldbm
```

The third step is to build the dependencies by issuing the following command:

```
make depend
```

The fourth step is to build the product by issuing the following command:

```
make
```

The fifth step is to test the build by issuing the following command:

```
make test
```


Finally, it is time to install the product by issuing the following command:

```
su root -c 'make install'
```

For proper integration with other products, add the following line to the `/etc/ld.so.conf`:

```
/usr/local/lib
```

Following this, issue the following command:

```
ldconfig
```

The OpenLDAP product should now be installed and ready for use by Samba and other products.

Building Samba

Building Samba is the next step. To complete the build of the 3.0.0 Beta 1 release, a number of steps are to be completed. All of the following steps will be completed from the `/source/samba-3.0.0beta1/source` directory.

The first step is to configure the build environment by issuing the following command:

```
./configure {options}
```

The following options were used as parameters to the configure command:

```
--prefix=/opt/samba-3.0  
--enable-cups  
--with-ads  
--with-ldap  
--with-ldapsam  
--with-syslog  
--with-quotas  
--with-acl-support  
--with-winbind  
--with-sendfile-support
```

The second step is to build the product. From the `/samba/samba-3.0.0beta` directory, the following command is issued:

```
make
```

Finally, install the product by issuing the following command:

```
make install
```

If needed and recommended for integration into the man system for help information, add the following line to `/etc/man.config`:

```
MANPATH /usr/local/samba/man
```

2.6.5 Configuring products

The integration of the products is required for operation. Note that a simple, non-SSL configuration is presented here. Refer to the product documentation for a complete overview of the configuration and integration options. Note that localhost or 127.0.0.1 is used as the LDAP server address in the following configurations. This should be changed to the IP address of the LDAP server as appropriate.

Configure OpenLDAP

It is likely that if the system has had OpenLDAP completely removed, the `/etc/slapd.conf` file will need to be created. The following is the contents of the configured `/etc/slapd.conf` file for OpenLDAP's configuration.

Example 2-12 Example /etc/slapd.conf file

```
include /usr/local/etc/openldap/schema/core.schema
include /usr/local/etc/openldap/schema/cosine.schema
include /usr/local/etc/openldap/schema/inetorgperson.schema
include /usr/local/etc/openldap/schema/nis.schema
include /usr/local/etc/openldap/schema/samba.schema
pidfile /usr/local/var/slapd.pid
argsfile /usr/local/var/slapd.args
access to dn=".*dc=somedomain,dc=local"
    by self write
    by * read
    by anonymous auth
database bdb
suffix "dc=somedomain,dc=local"
rootdn "cn=root,dc=somedomain,dc=local"
rootpw password
directory /usr/local/var/openldap-data
index objectClass eq
index default sub
```

A few items to note for this configuration:

1. The `samba.schema` is included following the `cosine`, `inetorgperson`, and `nis` schema definition files. This is required due to dependencies.
2. The access controls specified here are basic and likely insufficient for a complete deployment of the LDAP server.
3. The password is presented in free text providing a likely security exposure. The password can be encrypted by replacing the clear-text password with the output of the `slappasswd` program.
4. Minimal indexing is configured. Additional indexing will be required for an enterprise deployment for proper performance.

The OpenLDAP base configuration file `/etc/openldap/ldap.conf` needs to contain:

Example 2-13 Example /etc/openldap/ldap.conf entries

```
HOST 127.0.0.1
BASE dc=somedomain,dc=local
```

Next, the OpenLDAP clients configuration must be set up for correct for access. To accomplish this, edit the `/etc/ldap.conf` file with the following entries.

Example 2-14 Example /etc/ldap.conf file

```
host 127.0.0.1
base dc=somedomain,dc=local
uri ldap://127.0.0.1/
ssl no
binddn cn=root,dc=somedomain,dc=local
bindpw password
rootbinddn cn=root,dc=somedomain,dc=local
port 389
scope sub
pam_filter objectclass=posixaccount
pam_login_attribute uid
pam_member_attribute gid
pam_template_login_attribute uid
pam_password md5
nss_base_passwd dc=somedomain,dc=local?sub
nss_base_shadow dc=somedomain,dc=local?sub
nss_base_group dc=somedomain,dc=local?sub
nss_base_hosts dc=somedomain,dc=local?sub
```

For proper name server switch (NSS) operation with LDAP, change the following lines in the `/etc/nsswitch.conf` file:

```
passwd: files nisplus ldap
shadow: files nisplus ldap
group : files nisplus ldap
hosts : files nisplus dns ldap
```

The resulting `/etc/nsswitch.conf` file for our configuration is:

```
passwd: files nisplus ldap
shadow: files nisplus ldap
group: files nisplus ldap
hosts: files nisplus dns ldap
bootparams: nisplus [NOTFOUND=return] files
ethers: files
netmasks: files
networks: files
```

```
protocols: files nisplus
rpc:       files
services:  files nisplus
netgroup:  files nisplus
publickey: nisplus
automount: files nisplus
aliases:   files nisplus
```

Note: The ldap entry on the passwd, shadow, group, and hosts entries should be stated immediately after files for optimal configuration.

Following the above configurations, the NSS client daemon must be running. If the NSS client daemon is not running, the OpenLDAP daemon (slapd) hangs when starting. To autostart nscd, issue the following commands:

```
# chkconfig --add nscd
# chkconfig nscd on
```

Note: It may be required to specify the IP address (direct and reverse) in the hosts file for rapid resolution of the LDAP server's address to eliminate login delays.

If it is desirable to restrict users to specific hosts, the following statement needs to be added to the /etc/ldap.conf file:

```
pam_check_host_attr yes
```

Also, the host attribute must be defined with host names on each user ID to be restricted.

To start the OpenLDAP services, verify that /var/lib/ldap exists, and is owned by the user ID that the slapd process is running as. If it does not exist, create the directory and assign ownership and complete rights to the owner for this directory. This is the directory where OpenLDAP and Berkeley DB will store the LDAP object and attribute data.

The final step in configuring the OpenLDAP server is to import the base organizational and object information. This will be covered later in this book for our migration scenario.

Configure Samba for OpenLDAP

Configuring Samba for use with LDAP requires the modification of the /etc/samba/smb.conf file adding the following lines.

Example 2-15 LDAP Entries for Samba's configuration in global section

```
ldap admin dn = "cn=root,dc=somedomain3,dc=local"
```

```
ldap server = 127.0.0.1
ldap suffix = dc=somedomain3,dc=local
ldap port = 389
ldap ssl = off
passdb backend ldapsam:ldap://127.0.0.1
ldap delete dn = no
```

Note: TLS/SSL is not configured in the example configuration. It is recommended that TLS/SSL be set up and used for all communications between LDAP clients and servers.

After configuring the `/etc/smb.conf` file, the password needs to be set up for Samba's access of the LDAP server. This is accomplished with the following command:

```
# smbpasswd -w {password}
```

Note: As the above command specifies the password on the command line, it is now recorded in the shell command history. Be sure to clear the command history of this entry for security purposes.

Configure Samba

The base configuration of Samba for this redbook's migration scenario uses the following `/etc/samba/smb.conf` file:

Example 2-16 Example configuration file for base Samba configuration

```
[global]
netbios name = lnrxhas
workgroup = somedomain
server string = The Server Description
passdb backend = ldapsam, guest
oslevel = 65
preferred master = yes
domain master = yes
local master = yes
security = user
encrypt passwords = yes
domain logons = yes
logon path = \\%N\profiles\%u
logon drive = H:
logon home = \\homeserver\%u
logon script = logon.cmd
idmap uid = 10000-15000
idmap gid = 10000-15000
add machine script = /usr/sbin/useradd -d /dev/null -g 100 -s /bin/false
-M %u
```

```

passdb backend = ldapsam:ldap://127.0.0.1/ guest
ldap admin dn = "cn=root,dc=somedomain,dc=local"
ldap server = 127.0.0.1
ldap suffix = dc=somedomain,dc=local
ldap port = 389
ldap ssl = off
ldap delete dn = no

printing = bsd
load printers = yes
show add printer wizard = yes
printcap name = /etc/printcap
printer admin = root
lpq cache time = 30
use client driver = no

[homes]
comment = home directories
browsable = no
writable = yes

[netlogon]
path = /shares/netlogon
guest ok = yes
writable = no
share modes = no

[profiles]
path = /shares/profile
writable = yes
browseable = no
guest ok = yes
create mask = 0600
directory mask = 0700

[printers]
comment = all printers
path = /shares/spooler
browseable = no
guest ok = yes
public = yes
read only = yes
writable = no
printable = yes

```

Be aware of the following observations of the `/etc/samba/smb.conf` configuration used as the migration base:

1. It asserts that the server is a primary domain controller for the specified domain.
2. Encrypted passwords are required for a client to access the server resources.
3. The user add script specified does not provide full integration into the LDAP capabilities.
4. SSL communications with the LDAP server are disabled.
5. Standard bsd-style printing is used.
6. The `load printers = yes` statement makes the printers defined on the Linux server available to Samba connected workstations.

2.7 Summary

This chapter has described typical Windows 2000 and Linux server environments that could be the target environments for a migration from OS/2. In any particular environment, special requirements may require a slightly different configuration or product mix, but the environments described here provide a basis for the migration scenarios described in this redbook.

In the next chapter, we describe techniques for extracting the current configuration information from an OS/2 Server environment. This information will be used in Parts 2 and 3 of this redbook to recreate a similar server environment on other platforms.



Starting the OS/2 Server migration

This chapter provides an overview of a package called LSMT used to extract information from IBM OS/2 Warp Server. The extracted information will be used in later chapters for the migration to Windows and Linux.

LSMT primarily consists of a set of REXX procedures and supporting files. For information on obtaining LSMT, please see 3.2, “LSMT package” on page 64.

In this chapter, the following LAN server objects are extracted for manipulation:

- ▶ Domain
- ▶ Servers
- ▶ Groups
- ▶ Users
- ▶ Access
- ▶ Directories
- ▶ Printers
- ▶ Serial
- ▶ Application

3.1 Introduction

This chapter will explain the starting point of the OS/2 transition process. There are a few steps needed for the migration from OS/2 to Linux or Windows. Basically, they can be divided into the following phases:

- ▶ Extracting data
- ▶ Manipulating data
- ▶ Importing data

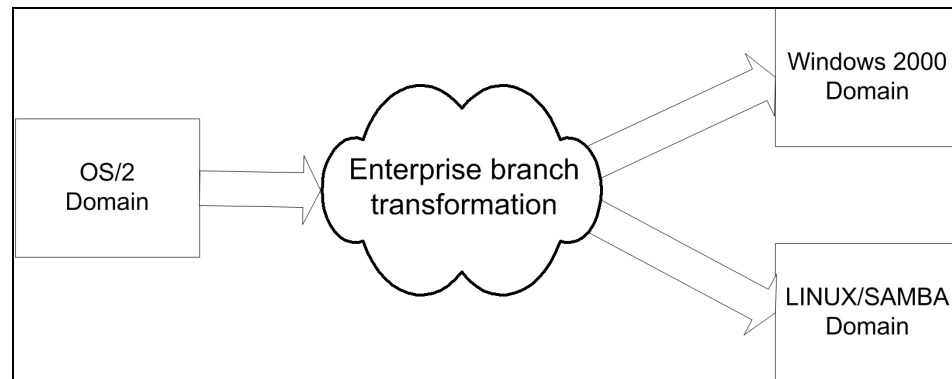


Figure 3-1 Transition process

The easy part of the process is to extract the information currently on the OS/2 domain. There are a number of tools available to extract data from the OS/2 domain, but for the purposes of this book we have to choose the LAN Server Management Tools (LSMT) package. The LSMT package can be downloaded from:

<http://hobbes.nmsu.edu/cgi-bin/h-browse?sh=1&dir=//pub/os2/util/network/lansrv>

LSMT is also available for download along with other source files from this redbook. Please see Appendix C, "Additional material" on page 557.

A list of the other tools can be found in Chapter 8, "Additional migration tools" on page 277.

3.2 LSMT package

The LSMT package was written in REXX by IBMers and is made available on an "as-is" basis. It allows an administrator to extract information from a current OS/2

Warp Server environment into ASCII files, change it, and use it to apply these changes to a newly installed environment.

Attention: Extracting the OS/2 Warp Server environment information to an ASCII file results in lines that easily exceed 254 characters. Depending on the definitions, they might even be much larger. Therefore, the editor or spreadsheet program to use for viewing and manipulating these ASCII files has to be able to handle files with large lines, and it should at least warn you if any truncation occurs. (TEDIT.EXE, E.EXE and IBMWORKS support long lines. EPM.EXE does not, but it warns in case of truncation.)

Most of the base functions have been split into different programs for better reading and selective use of the resources. To retrieve information about the existing LAN environment and extract it to an ASCII file, the GETxxx.COM program is used. If there are any changes required, simply edit these files directly with an editor, spreadsheet, or with a batch program. More information about LSMT can be found in the LSMT.TXT file within the package.

3.2.1 Install LSMT package

To install LSMT, please have these files available:

- ▶ LSMT.ZIP: Compressed file, contains all necessary product files. See 3.1, "Introduction" on page 64 for download instructions.
- ▶ PKUNZIP2.EXE: Available after an OS/2 LAN requester and server installation. Any other program to unpack zip files will work as well.

All files must be installed in a common directory on the same disk; the example assumes the drive letter is D:

- ▶ Download the ZIP file.
- ▶ Go to the drive:
D:
- ▶ Create a directory:
MD \LSMT
- ▶ Go to this directory:
CD \LSMT
- ▶ Unzip the packed file:
PKUNZIP2 LSMT.ZIP
- ▶ Erase LSMT.ZIP from this directory:
DEL D:\LSMT.ZIP

Now, there are two ways to use the LSMT procedures:

- ▶ To get details about the defined users, execute **GETUSERS** with the computer name of the Primary Domain Controller.

```
GETUSERS /SRV:PDC
```

- ▶ To get all the DCs data chained together such as Users, Aliases, Logon Assignments, and so on, execute **GETALL** with the computer name of the Primary Domain Controller.

```
GETALL /SRV:PDC
```

3.2.2 REXX and LAN server functions

The power of REXX allows users to use REXX functions that are not delivered with the original REXX code just by adding and loading additional DLLs that contain the required functions. One of the major enhancements of LS 4.0 in contrast to earlier versions is that all LAN server APIs can be accessed through REXX by external functions in a new DLL (LSRXUT.DLL). There are two DLLs included in LAN Server 4.0: LSRXUT3.DLL and LSRXUT4.DLL. LSRXUT3.DLL is essentially the same as LSRXUT4.DLL with some restrictions originating in the LAN Server 3.0 code. One of the few restrictions is that when using a LAN Server 3.0 environment, the apply API cannot be used to copy the actual ACL of the directory to its subdirectories. If the directory resides on a LAN Server 4.0 server, even if your domain controller has a lower release, the apply will work.

The LSRXUT.DLL does not have to be installed at the server itself, but on the machine where the programs will be executed. If you are using LAN Requester 3.0, then you should use LSRXUT3.DLL. In order to provide the latest information, the most recent versions of LSRXUT3.DLL and LSRXUT4.DLL are included, which will automatically install the correct version by using the registration program **RGLSRXUT.CMD**.

3.2.3 LSMT INI files

For some of the sample programs, an xxx.INI is provided. These are files to enable the user to set several values externally without touching the CMD files. These files allow you to specify which columns will be included in the output file and which columns they should occupy. Some programs will not provide this flexibility in making the decision about which columns to see and which columns to use, because the content of the columns can change dynamically. In these cases, an INI file can always be just a snapshot of your actual environment. Therefore, it needs to be rebuilt anytime you restart the programs because your environment might have changed and some of the objects represented by the columns no longer exist or others have been added.

Attention: Do not change any of the column names, these are predefined names used by the API.

With the INI file, you have following options:

- ▶ Changing the order of the lines will change the order of the corresponding columns in the output file.
- ▶ Deleting lines or typing an asterisk in the first column of the line will have the effect that this column will not be displayed in the output file.
- ▶ Changing the numbers will change the corresponding column width.

3.2.4 LSMT ASCII file

All ASCII files produced by the REXX command files have a common format. In order to use advanced methods in editing these files, a format that is easy to edit and read with an ordinary file editor or a spreadsheet program should be used.

Most spreadsheet programs on the market are able to import comma-separated value (CSV) files. Within these files, each row of the spreadsheet is represented by a corresponding line in the CSV file, and vice versa. To decide which lines belong to which column of the spreadsheet, a special character is used as delimiter of each column. In most countries the comma , is used as a list separator, others use the semicolon ;

By default, the semicolon is used in LSMT as a delimiter because some of the values in OS/2 Warp Server (for example, Comments) may contain commas. If importing the ASCII file into a spreadsheet and data is not separated correctly, try to figure out whether your default list separator is the one used in the CSV file. Most of the spreadsheet programs save the files by default to their own proprietary format. Using a spreadsheet program to change values, the data has again to be exported into a CSV file using the appropriate delimiter.

The top section of each CSV file starts with a line starting with OPT. This line contains the header description of all columns used in this file. Each column is separated by ; (not seen if in a spreadsheet program). With respect to the meaning of the semicolon, do not delete semicolons within a row because deleting one semicolon has the effect of shifting all entries one column to the left (and only for this line!).

Conversely, if you are adding an additional semicolon, all entries shift one column to the right. For this reason, do not use semicolons in your entries.

Example 3-1 LSMT sample ASCII output

```
* List of all logon assignments,allowed Options U=update D=delete
OPT;USERID ;REXX;WARPAPPL;DOSAPPL;WINAPPL;PUBLIC;IBM4039;OPTRA;MODEM;
;ODIER ; ; V ; ; ; P ; LPT1 ; ;COM4 ;
;PAULI ; R ; W ; ; ; P ; ;LPT3 ; ;
;RYKAERTA; R ; W ; X ; ; P ; LPT3 ; ; ;
;SHIMIZU ; ; W ; ; ; P ; ; ; ;
;TESTINI ; ; ; X ; Y ; P ; LPT1 ; ; ;
;VERNON ; ; W ; ; ; P ; ; ;COM3 ;
```

The first column is always your option column. The following entries are allowed for the option column:

- * Indicates that this line is a comment line, it will be ignored.
- OPT This line will contain the import information about the columns used in this file
- A ADD. Indicates that the selected line will be used for the import procedure to Windows/Linux

If there is no entry or just blanks in the option column, the line will not be processed and will be ignored. Therefore, to apply any changes using this file, be sure to set the proper option in the option column; otherwise, the changes will not be processed.

3.3 Collecting data using LSMT

A selected list of REXX scripts, DLLs, and input files from the LSMT package will be used for the OS/2 migration. The selected REXX scripts will extract all the LAN server objects and attributes from the OS/2 Server. Each of the scripts will be explained in the order of migrating the OS/2 Server. For more information and a list of files, please turn to Appendix B, “REXX source code” on page 477.

Important: For almost all of the following sections, it is required to be logged on to the OS/2 Server or Domain with administrative rights.

3.3.1 Domain

There are several ways of retrieving the domain information from your OS/2 domain. Below are two ways to retrieve the domain name:

- ▶ Start the IBM LAN Server Administrator GUI and look at the name of the castle.

- ▶ Run the following command on any of the servers in the domain:

```
FIND /I "DOMAIN =" \\[PDC]\IBMLAN$\IBMLAN.INI
```

3.3.2 Servers

To retrieve all information from the servers within the OS/2 domain, run the **GETSRVR.CMD** located in LSMT directory. When running the command below on *Somedomain*, the result will include the information in Table 3-1.

```
C:\OS2MIG\GETSRVR.CMD /SRV:PDC /OUT:C:\OS2MIG\GETSRVR.LOG /T /M
```

Table 3-1 GETSRVS.CMD

severInfo	Description	PDC	BDC
NAME	The server computer name	PDC	BDC
VERSION_MAJOR	The major version number (Version)	5	5
VERSION_MINOR	The minor version number (Release)	20	20
TYPE	The server type. This information is a hexadecimal value and is not interpreted	0000002B	13
Comment	The server comment	-none-	BDC
ULIST_MTIME	The last time the users list was modified	Never modified or unknown	Never modified or unknown
GLIST_MTIME	The last time the group list was modified	Never modified or unknown	Never modified or unknown
ALIST_MTIME	The last time the access control list was modified	Wed Jun 11 16:41	Thu Jun 12 12:41
USERS	The maximum of users on the server	101	101
DISC	The auto-disconnect value	120	120
ALERTS	The server alerts receiver table. The table can be empty	-none-	-none-

severInfo	Description	PDC	BDC
SECURITY	The security type of the server	User-level	User-level
AUDITING	The auditing setting	Enabled	Enabled
NUMADMIN	The maximum number of administrators	65535	65535
LANMASK	The order in which the network device driver are served. The value is uninterpreted	3	3
HIDDEN	The server hidden attribute setting	Visible	Visible
ANNOUNCE	The network announce delta (in seconds), which determines how often the server will be announced to other computers on the network	180	180
ANNDELTA	The random announce rate (in milliseconds)	3000	3000
GUESTACCT	The guest account name	GUEST	GUEST
USERPATH	The path name to user directories	-none-	-none-
CHDEVS	The number of serial devices that can be shared on the server	16	16
CHDEVQ	The number of serial device queues that can coexist on the server	2	2
CHDEVJOBS	The number of serial jobs that can be pending on the server	48	48
CONNECTIONS	The maximum number of connections to netnames that are allowed	16384	16384

serverInfo	Description	PDC	BDC
SHARES	The maximum number of netnames a server can accommodate	204	204
OPENFILES	The num of files (file handles to for example files or pipes) tat can be opened at once	4500	4500
SESSOPENS	The number files that can be open in one session	256	256
SESSVCS	The maximum number of virtual circuits per client	1	1
SESSREQS	The number of simultaneous requests that a client can make on any virtual circuit	50	50
OPENSEARCH	The number of searches that can be opened at once	50	50
ACTIVELOCKS	The number of file locks that can be active	450	450
NUMREQBUF	The number of server buffer that are provided	202	202
SIZREQBUF	The size (in bytes) of each server buffer	4096	4096
NUMBIGBUF	Number of 64KB server buffers that are provided	46	46
NUMFILETAKS	Number of processes that can access the operating system at one time	2	2
ALERTSCHED	The alert interval for notifying an administrator of a network event	5	5
ERRORALERT	The number of entries that can be written to the error log file during a interval before notifying an administrator	5	5

severInfo	Description	PDC	BDC
LOGONALERT	The number of failed logon attempts to allow a user before notifying an administrator	5	5
ACCESSALERT	The number of failed file accesses to allow before issuing an administrative alert	5	5
DISKALERT	The number of kilobytes of free space, at which, an administrator must be notified that the free space is low	5000	5000
NETIOALERT	The Network I/O error ratio in one tenth of a percent to allow before the administrator is notified	5	5
MAXAUDITSZ	The maximum audit file size	100	100
SRVHEURISTICS	The server heuristics setting	11110141111 3130000000	11110141111 3130000000
AUDITEDEVENT	The audit event setting. The value is unformatted and is presented hexadecimal	8000	8000
AUTOPROFILE	The server auto profile setting	Unknown	Unknown
AUTOPATH	The server autopath location	-none-	-none-

3.3.3 Groups

To retrieve all information about groups within the OS/2 domain, run **GETGRPS1.CMD** and **GETGRPS2.CMD** located in the LSMT directory. When running the commands below on SOMEDOMAIN, the information is provided as seen in Table 3-2 and Table 3-3:

```
C:\OS2MIG\GETGRPS1.CMD /SRV:PDC /OUT:C:\OS2MIG\GETGRPS1.LOG /T /M
```

Table 3-2 GETGRPS1.CMD

groupInfo.NAME	groupInfo.COMMENTS
ADMINS	
BOOKREAD	
BOOKWRITE	
GROUPID	Default Group ID
GUESTS	
LOCAL	
PRINTER	Printer Group
SERVERS	System ID - Server
TRANSITION	
USERS	

C:\OS2MIG\GETGRPS2.CMD /SRV:PDC /OUT:C:\OS2MIG\GETGRPS2.LOG /T /M

Table 3-3 GETGRPS2.CMD

	BDC	GUEST	OLIVER	PDC	USERID	WYNA ND
ADMINS					X	
BOOKREAD						X
BOOKWRITE			X			
GROUPID					X	
GUESTS		X				
LOCAL						
PRINTER			X			
SERVER	X			X		
TRANSITION			X			X
USERS	X		X	X		X

3.3.4 Users

To retrieve all information about the users within the OS/2 domain, run the **GETUSERS.CMD** located in the LSMT directory. When running the command below on SOMEDOMAIN, it results in the information in Table 3-4.

The user ID Wynand, which is listed in Table 3-4, has certain restrictions set like the logon hours and allowed workstations that one may sign on to.

```
C:\OS2MIG\GETUSERS.CMD /SRV:PDC /OUT:C:\OS2MIG\GETUSERS.LOG /T /M
```

Included in this section, you can also extract the user password hash with **GETPWD.CMD** as shown in Table 3-5.

Table 3-4 *GETUSERS.CMD*

userInfo	Description	User ID 1	User ID 2
NAME	The user accounts name	USERID	WYNAND
PASSWORD_AGE	The password age in seconds	425957911	180002
PRIV	The user account privilege level	Administrator	User
HOME_DIR	The user home directory, if one specifies	-none-	U:\PDC\E\$\LANHOMES\WYNAND
COMMENT	The user account comment	-none-	Wynand_Pretorius
FLAGS	User account flags	-none-	-none-
SCRIPT_PATH	The name of the logon script together with the path specification relative to the NETLOGON SCRIPT parameter	-none-	-none-
AUTH_FLAGS	The operator privileges granted to the user		PCSA
FULL_NAME	The full name of the user	-none-	-none-

userInfo	Description	User ID 1	User ID 2
USR_COMMENT	The user comments that are user-settable comments	Default User Id	Standard Bank User
PARMS	The user accounts parameters	-none-	-none-
WORKSTATION	The workstations restrictions for the user	No Restriction	PC1 PC2
LAST_LOGON	The last logon time	Thu Jun 12 12:45:12 2003	Thu Jun 12 12:40:072003
LAST_LOGOFF	The last logoff time	Thu Jun 12 13:03:36 2003	Thu Jun 12 12:40:17 2003
ACCT_EXPIRES	The time the user accounts expires	(null)	(null)
MAX_STORAGE	The maximum storage allowed for the home directory	No Limit	No Limit
RESTRICTED_HOURS	Logon restriction on certain hours	Restrictions provided	Restrictions provided
1.LOGON_HOURS	The logon hours allowed. 0 means 0 to 0:59, 1 1:00 to 1:59. The logon_hours are only valid if userInfo.restricted_hours differs from the value '-none-'	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	
2.LOGON_HOURS	Review description of 1.LOGON_HOURS	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	7 8 9 10 11 12 13 14 15 16 17 18

userInfo	Description	User ID 1	User ID 2
3.LOGON_HOURS	Review description of 1.LOGON_HOURS	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	7 8 9 10 11 12 13 14 15 16 17 18
4.LOGON_HOURS	Review description of 1.LOGON_HOURS	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	7 8 9 10 11 12 13 14 15 16 17 18
5.LOGON_HOURS	Review description of 1.LOGON_HOURS	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	7 8 9 10 11 12 13 14 15 16 17 18
6.LOGON_HOURS	Review description of 1.LOGON_HOURS	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	7 8 9 10 11 12 13 14 15 16 17 18
7.LOGON_HOURS	Review description of 1.LOGON_HOURS	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	
BAD_PW_COUNT	The number of attempts to validate a bad password	0	4
NUM_LOGONS	The number of successful logons	30	476
LOGON_SERVER	The computer to handle logon requests for a user account	*	*
COUNTRY_CODE	The country code of the user	0	0
CODE_PAGE	The country code page for the user	437	0

Table 3-5 GETPWD.CMD

User ID	Password
ANDREI	CD017457761C8B05AAD3B435B51404 EE
BDC	8E2D4DD5EF19A9B0AAD3B435B51404 EE
GUEST	AAD3B435B51404EEAAD3B435B51404 EE
LEIF	32DD5DAB4DC507A4AAD3B435B51404 EE
MARC	2FD076F9E0306FFFEAAD3B435B51404 EE
OLIVER	617093781CC21A60AAD3B435B51404E E
PDC	AAD3B435B51404EEAAD3B435B51404 EE
RICHARD	E4301A7CD8FDD1ECAAD3B435B5140 4EE
USERID	E52CAC67419A9A224A3B108F3FA6CB 6D
WYNAND	D851BE004D8658DFAAD3B435B51404 EE

3.3.5 Access

To retrieve information from the Access Control Lists (ACLs) within the OS/2 domain, run the **GETACL.CMD** located in the LSMT directory. When running the command below on SOMEDOMAIN, the results are shown in Table 3-6:

```
C:\OS2MIG\GETACL.CMD /SRV:PDC /OUT:C:\OS2MIG\GETACL.LOG /T /M
```

Table 3-6 GETACL.CMD

		BOOK	LANSHARE
ALIAS		BOOK	LANSHARE
AUDIT		-none-	-none-
ADMINS			
BOOKREAD		RG	

		BOOK	LANSHARE
BOOKWRITE		RWCDAG	
GROUPID			
GUESTS			
LOCAL			
PRINTER			
SERVER			
TRANSITION			RWCXDAPG
USER			
ANDREI			
BDC			
GUEST			
LEIF			
MARC			
OLIVER			
PDC			
RICHARD			
USERID			
WYNAND			

3.3.6 File and printer shares

To retrieve information from the file and printer shares within the OS/2 domain, run the **GETALIAS.COM** located in LSMT directory. When running the command below on SOMEDOMAIN, the results are seen in Table 3-7.

```
C:\OS2MIG\GETALIAS.COM /SRV:PDC /OUT:C:\OS2MIG\GETALIAS.LOG /T /M
```

Table 3-7 GETALIAS.COM

AliasInfo	Description	PRINT	BOOK	LANSHARE
NAME	The alias name	PRINT_Q	BOOK	LANSHARE
REMARK	The alias remark	Network Printer Queue		

AliasInfo	Description	PRINT	BOOK	LANSHARE
SERVER	The computer name of the server where the resource describes by this alias resides	BDC	PDC	BDC
NETNAME	The alias name for the file alias	IBMNULLP	BOOK	LANSHARE
LOCATION	The alias location	Within Domain	Within Domain	Within Domain
MODE	When the alias is shared	At server startup	At server startup	At server startup
MAXUSES	The maximum number of users who can have redirection to the resource identified by this alias	65535	65535	65535
TYPE	The alias type	Printer	Files	Files
QUEUE	The queue name for serial or printer alias only	IBMNULLP	Unknown	Unknown
PATH	The path for files alias only	Unknown	F:\BOOK	E:\LANSHAR E
PRIORITY	The serial device priority	Unknown	Unknown	Unknown
DEVICE_POOL	The serial device pool	Unknown	Unknown	Unknown

3.3.7 Serial devices

The OS/2 Warp Server services include the sharing of serial devices. Using that feature, an administrator has been able to allow sharing of bidirectional serial devices such as modems within the domain. Currently, there is no one-to-one mapping of this capability to either Windows or Linux. Please review the Windows Chapter 4.8, "Migrating serial devices" on page 158, and Linux Chapter 6.8, "Migrating serial devices" on page 232 for additional information about serial device migration.

3.3.8 Applications

To retrieve all information for the applications within the OS/2 domain, run the **GETAPPL.COMD** located in the LSMT directory. When running the command below on SOMEDOMAIN, you will retrieve the information in Table 3-8.

```
C:\OS2MIG\GETAPPL.COMD /SRV:PDC /OUT:C:\OS2MIG\GETAPPL.LOG /T /M
```

Table 3-8 GETAPPL.COMD

ApplInfo	Description	DOS_PRG
NAME	The application name	DOS_PRG
REMARK	The application remark	Public DOS Application
COMMAND	The command that starts the application	qbasic
COMMAND_PARMS	The application start parameters	
APP_ALIAS_OR_DRV	The alias or drive where the application resides. It specifies a drive letter, followed by a colon(:). if the application resides on the user's local machine or it specifies an existing alias if the application resides on a server.	LANSHARE
APP_DRIVE	Applies to DOS public applications only. It is used to specify the drive that is current when the application runs. A value of * indicates that the system choose a drive letter.	
APP_PATH_TO_DIR	The remaining path to the application	\DOSAPP

ApplInfo	Description	DOS_PRG
WRKDIR_ALIAS_OR_DRIVE	Specifies the directory that is made current when the application runs. If the working directory is on the local machine, it specifies the drive, where the directory is located. If the working directory is remote, it specifies an existing alias where the directory is located	LANSHARE
WRKDIR_DRIVE	Specifies the drive that the working directory is to be assigned to when the application is started. A value of * indicates that the system should choose a drive when the application is started	*
WRKDIR_PATH_TO_DIR	The remaining path to the working directory	\DOSAPP
PROMPT	Prompt for parameters	Prompt user for parameters
INTERFACE	The interface type	Unknown
APPTYPE	The application type	Public DOS application
RES_COUNT	The number of application resource list that follows. A value of zero indicates that the application does not require any redirected devices when it runs	0

3.4 Considerations and limitations

With all of the LSMT information extracted above from the OS/2 domain, there are some considerations to be taken into account:

- ▶ Printers

For the migration of printers, please review the general recommendations made in 1.5.4, “Printer migration” on page 13. Refer to 4.7, “Migrating

printers” on page 154 for Windows, and 6.7, “Migrating printers” on page 227 for Linux.

▶ **DASD limits**

There is no direct migration path of OS/2 LAN Server DASD limits to Windows or Samba. Some third party applications can be considered. Refer to 4.6.4, “Migrating DASD limits” on page 152 for Windows, and 6.6.6, “Migrating DASD limits” on page 226 for Linux.

▶ **Serial devices**

OS/2 LAN Server services included sharing serial devices. Using that feature, an administrator is able to share bidirectional serial devices like modems within the domain. Windows and Samba do not include a comparable feature. Refer to 4.8, “Migrating serial devices” on page 158 for Windows, and 6.8, “Migrating serial devices” on page 232 for Linux.

▶ **Public applications**

There is no direct migration path for OS/2 LAN Server public applications to Windows and Samba. There are some third party products or concepts available that fill this gap. Refer to 4.9, “Migrating applications” on page 158 for Windows, and 6.9, “Migrating applications” on page 232 for Linux.

▶ **Access control list for directories and files**

ACL is another challenging aspect of the migration. Refer to the following chapter for the migration path: 4.6.1, “Migrating access control” on page 142 for Windows, and 6.6, “Migrating directories and access controls” on page 219 for Linux.

3.5 Cross references

Although some of the LSMT code to extract the information from an OS/2 domain was used, some simplified REXX code was created to run on the OS/2 Server for manipulation of the log files, and to create new “usable” files for the migration process on the target platform.

Below you will find a table that cross references the OS/2 object to Windows or Linux object. The complete source code for LSMT is documented in Appendix B, “REXX source code” on page 477.

Table 3-9 Cross reference from OS/2 to Windows and Linux

OS2 Server	Windows	Linux
Domain	Section 4.2, “Migrating the domain” on page 100	Section 6.2, “Migrating the OS/2 domain” on page 198

OS/2 Server	Windows	Linux
Servers	Section 4.3, "Migrating server definitions" on page 103	Section 6.3, "Migrating server definitions" on page 199
Groups	Section 4.4, "Migrating groups" on page 108	Section 6.4, "Migrating groups" on page 201
Users	Section 4.5, "Migrating users" on page 113	Section 6.4, "Migrating groups" on page 201
Access	Section 4.6, "Migrating directories" on page 141	Section 6.5, "Migrating users and passwords" on page 206
File and Printer Shares	Section 4.6, "Migrating directories" on page 141 and Section 4.7, "Migrating printers" on page 154	Section 6.6, "Migrating directories and access controls" on page 219 and Section 6.7, "Migrating printers" on page 227
Serials	Section 4.8, "Migrating serial devices" on page 158	Section 6.8, "Migrating serial devices" on page 232
Applications	Section 4.9, "Migrating applications" on page 158	Section 6.9, "Migrating applications" on page 232

3.6 Summary

LSMT is a set of REXX procedures provided on an as-is basis that extracts various configuration information from OS/2 Servers and places the data into ASCII files. These files can be edited and otherwise manipulated before being used to import the data into a target environment for duplicating or migrating to a new server.

Part 2 of this redbook will address the migration to a Windows 2000 environment. The files generated by LSMT can be used to help move OS/2 Server configuration information to the Windows 2000 environment.



Part 2

Migration to Windows 2000

The chapters in this part of the book describe a step by step migration to a Windows 2000 environment. Data gathered from the OS/2 domain as described in Chapter 3, “Starting the OS/2 Server migration” on page 63, is used and imported to the Windows 2000 and Active Directory Services environment.

Chapter 4, “Migrating OS/2 Servers to Windows 2000” on page 87, addresses the steps to fully migrate the OS/2 domain and LAN servers, providing the basic infrastructure.

Chapter 5, “Migrating the software stack to Windows 2000” on page 177, briefly describes the migration considerations for the most common middleware that often exists in OS/2 Server environments.



Migrating OS/2 Servers to Windows 2000

This chapter describes the migration of the core functions and features from an IBM OS/2 Warp Server Domain to Windows 2000 as the target platform.

Before performing the steps in this chapter, the migration preparation should be completed, including data extraction, and retrieving and modifying the domain definition of your OS/2 domain as discussed in Chapter 3, “Starting the OS/2 Server migration” on page 63.

4.1 Overview of Windows 2000 migration

With the exception of a few areas, the migration to Windows 2000 is straightforward and relatively simple. Going into this chapter, our assumption is that a basic Windows 2000 domain has been installed and is running as described in 2.1, “Windows 2000 as a target platform” on page 20.

We also assume that data has been extracted from the OS/2 domain using the LSMT tools as described in Chapter 3, “Starting the OS/2 Server migration” on page 63. Please refer to that chapter before beginning the tasks in this chapter.

This chapter will cover:

- ▶ Active Directory structure setup
- ▶ Helpful tools for the migration
- ▶ OS/2 domain objects migration
- ▶ Discussion of the limitations or options for the migration scenarios from OS/2 to Windows 2000, including DASH limit, public applications, and serial devices
- ▶ Logon assignment considerations
- ▶ Client printing considerations

4.1.1 Considering the order of migration steps

When planning the migration, there are some dependencies that force a distinct sequence of steps to transfer the OS/2 Server objects to Windows 2000:

- ▶ The domain has to be created first to receive all other domain objects.
- ▶ To operate the domain, at least one server needs to be operational.
- ▶ User objects include group membership, so groups need to be defined before users.
- ▶ Servers file and print resources are protected by ACLs, which are based on group and user objects, and for that reason they need to be the last.

Keeping these considerations in mind, the following order of migration steps is recommended:

1. Define the domain.
2. Install the servers.
3. Migrate the group objects.
4. Migrate the basic user objects:
 - a. Migrate group membership.
 - b. Migrate passwords.

- c. Migrate logon assignments.
5. Migrate file resources:
 - a. Migrate the ACLs.
 - b. Migrate the aliases.
 - c. Migrate the data.
6. Migrate the print resources.

Figure 4-1 outlines the order of our migration process.

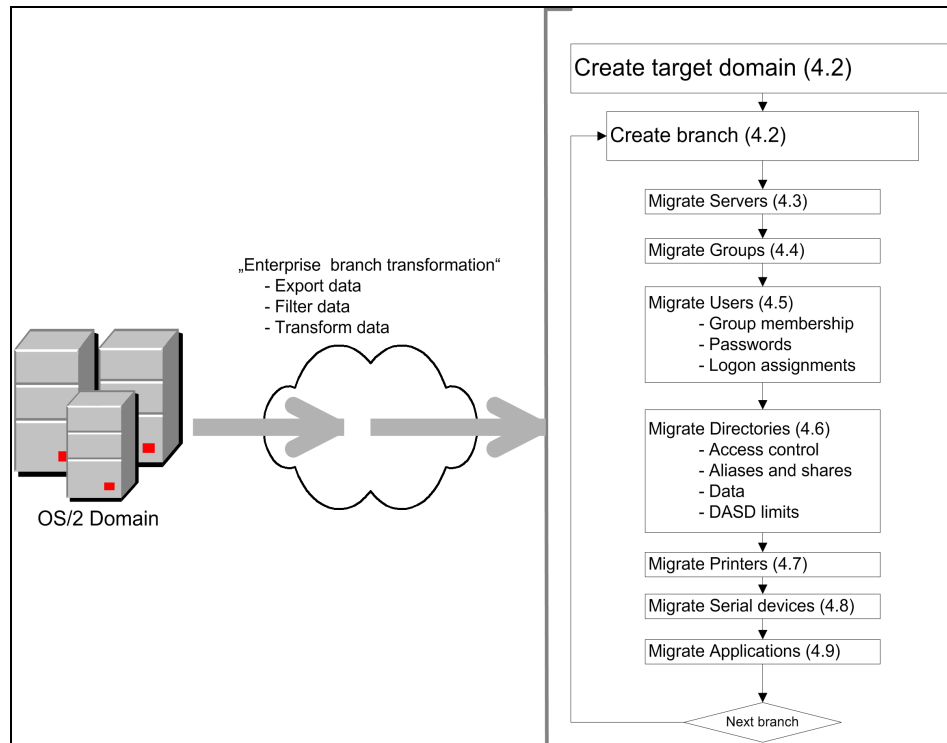


Figure 4-1 General workflow of domain migration to Windows 2000 domain

4.1.2 Design of Active Directory

For Windows 2000 (or higher) being the target platform of the migration, an Active Directory Services (ADS) structure is recommended to be defined first. Microsoft provides extensive documentation on defining and designing Active Directory Services. For this redbook example, a simple implementation is chosen to give a general idea. We do not claim completeness nor best practice design. This section discusses some of the issues, and provides the description of steps to create a target domain. This includes:

- ▶ Overview of our design of domains for Active Directory
- ▶ Creation of general structures (organizational units) done once for the whole enterprise
- ▶ Creation of branch specific structures done one for each branch or source domain
- ▶ Short discussion on designing sites

Domain structure

As part of the Active Directory design, it was decided to create a simple approach that differs from the usual design guidelines. It is good practice to define a root domain that only works as the starting point of your Active Directory domain tree. All domains containing user definitions or resources are defined as sub-domains in this namespace. As this book does not intend to replace Active Directory design guidelines, the domain structure was set up in the most simple way to hold all branches in one global domain without any root or subordinate domains. For that reason, only one domain name space was created, creating organizational units as containers for each branch domain. In the following chapters, for simplification reasons, the domain root `.local` is omitted and the target domain is named `somedomain.local`. Figure 4-2 illustrates this.

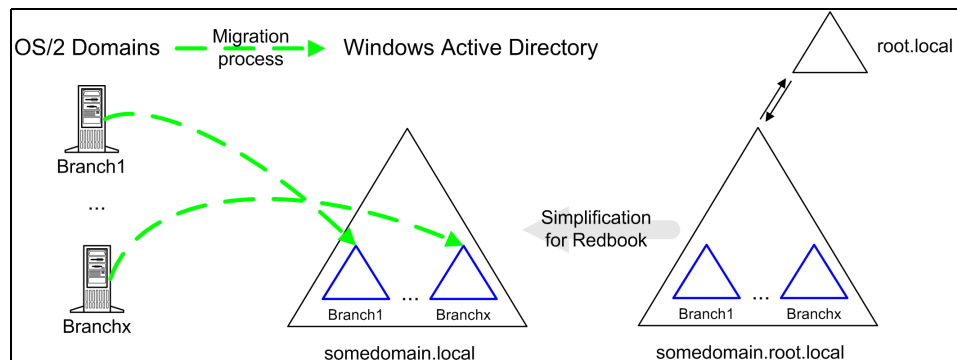


Figure 4-2 Active Directory design for transition of OS/2 branch domain

Organizational units

Considering our design of an Active Directory, we define a very basic LDAP tree consisting of some Organizational Units (OU), which should give you an idea of how to migrate all OS/2 domain definitions rather than claiming to be complete or adequate for any enterprise. With Figure 4-3 you get the general idea of this structure:

Central

This OU is the base container for user and group definitions used in a centralized way. Here groups or

users, which are specific for a service or that have been defined in all source domains (for example, administrator accounts, FTP users), are found.

Systems

Windows 2000 stores server and computer objects in an Active Directory to put them into an organizational, geographical, or other context. Defining these OUs, one can benefit by using group policies to centrally define rules or options for these objects. These are proprietary objects, separated from the user and group tree to simplify synchronization to other LDAP or metadirectory servers. The subsidiary OUs are defined for the different types of workstations (notebooks, standard desktops, specialized workstations) and servers (file, print, domain controllers, application servers, and so on).

GPO

Container for group policy objects. This container holds all GPO of the domain. Because GPOs are often used in different OUs, they are defined here and linked to an OU when needed.

Branch

The branch OU is the base object for our migration scenario. All migrated branches are transferred to that context. The structure was created with the OS/2 domain name as an organizational principle. In larger environments, it may be good practice to add a geographic structure like West or East as seen in Figure 4-3 on page 93. The scripts omitted this for simplification.

Each branch consists of the following OUs:

Groups

Group definitions from OS/2 are transferred here. In the migration process, we will describe concepts to allow a separation depending on their purpose:

Access

We will migrate groups used to define ACL on resources into this OU.

Organization

These groups usually specify membership according to organizational principals, project groups, or a distribution list for e-mail.

Application

Application services like Citrix Metaframe or IBM Workspace on Demand often use group memberships to assign applications to certain users. These application groups will be found here after the migration.

Print

These print groups assign shared printer queues to users.

Users

All user accounts selected for the migration will be found in this OU after migration.

All other containers and organizational groups provided by the promotion process (DCPROMO) were left empty and untouched.

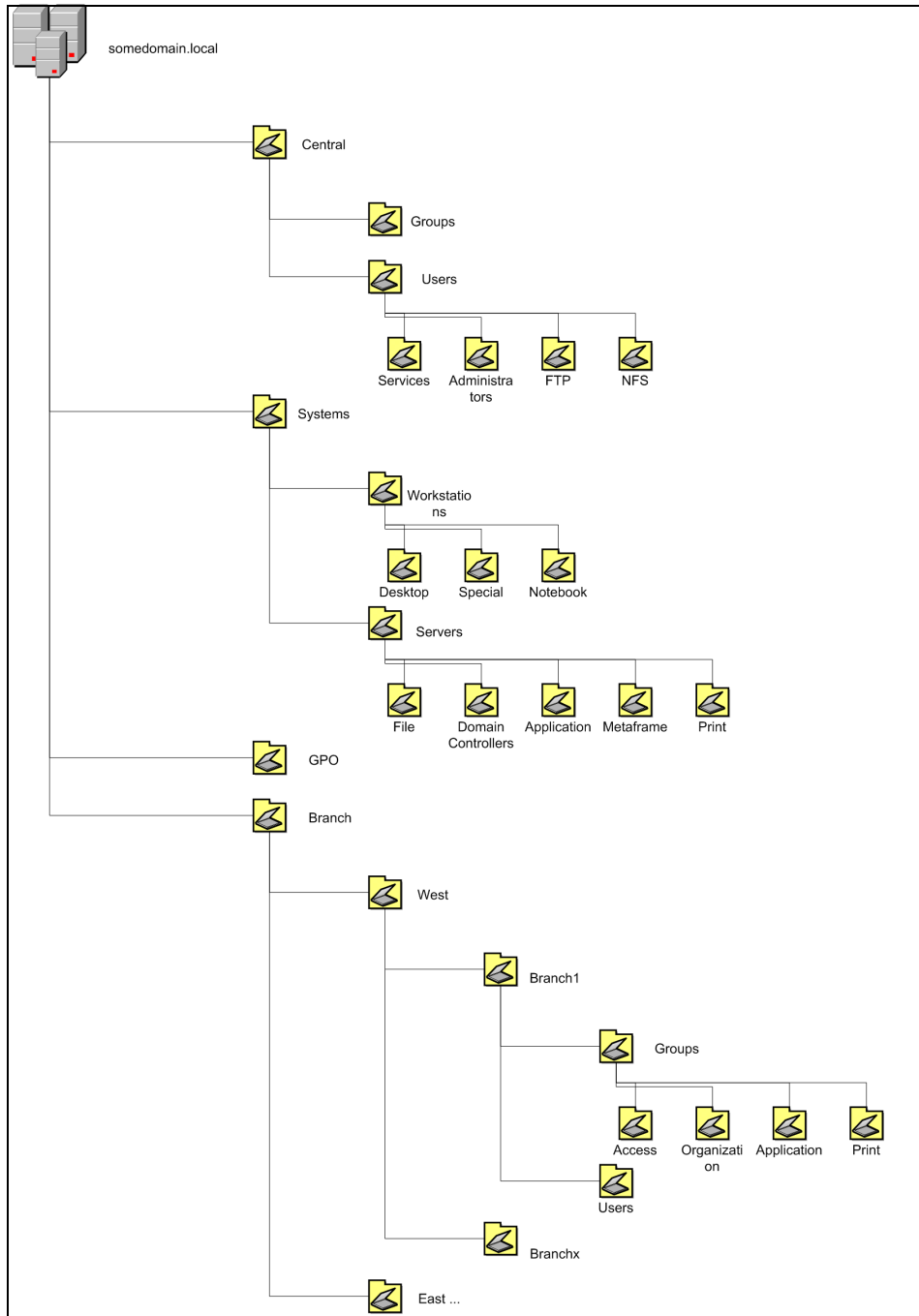


Figure 4-3 Organizational units in `somedomain.local`

Sites

Beside the logical structure of the Active Directory, the physical network structure is based on a unit known as a site. To keep it simple, each branch is represented by one site object, assuming the branch has at least one Internet Protocol (IP) subnet, with all branches connected together with high-speed and reliable connections.

We will create new site objects only while promoting domain controllers by specifying a new site name in the response file. For further information about sites, especially in branch environments, the following publication from Microsoft is a good start: *Active Directory Branch Office Guide Series*, found at:

<http://www.microsoft.com/technet/prodtechnol/ad/windows2000/deploym/adguide/default.asp>

4.1.3 Tools used during migration

Besides the scripts such as those provided with LSMT and those already described, some additional tools were used, and are briefly introduced here.

The LDAP Data Interchange format (LDIF)

The LDIF format, as described in RFC2849, is used to convey directory information, or a description of a set of changes made to directory entries. An LDIF file consists of a series of records separated by line separators. A record consists of a sequence of lines describing a directory entry, or a sequence of lines describing a set of changes to a directory entry. There is a one-to-one correlation between LDAP operations that modify the directory (**add**, **delete**, and **modify**), and the types of LDIF change records operations (add, delete, and modify). This correspondence is intentional, and permits a straightforward translation from LDIF change records to protocol operations. For more information please refer to the following publication:

The LDAP Data Interchange Format (LDIF) - Technical Specification, found at:

<http://www.ietf.org/rfc/rfc2849.txt>

LDIFDE

This tool is part of the Windows 2000 Server and can be found in the System32 directory. This program helps to automatically import directory objects like organizational units (OU), or even user objects through a scriptable command line interface using LDIF files.

Also, it has a simple search and replace capability to replace a string with any given context. It will be used to create base structures within the Active Directory

to prepare the migration of a domain, and to migrate domain definitions, users, and groups.

For additional information about the `Ldifde` utility, visit the Microsoft Knowledge Base and search for the article *Using LDIFDE to Import and Export Directory Objects to Active Directory*, Q237677, or type the following command at a command prompt on a computer that is running Windows 2000 Server:

```
Ldifde /?
```

Active Directory Services Interface

ADSI gives developers access to the Active Directory services and other LDAP directory services through an open set of interfaces. Administrators and developers can use ADSI to manage the resources in a directory service, regardless of which network environment contains the resource. ADSI enables administrators to automate common tasks such as adding users and groups, managing printers, and setting permissions on network resources. Applications can be developed in multiple languages including Visual Scripting Host, Visual Basic, and C++. For more information please refer to the following publication:

Microsoft Platform SDK: Active Directory Service Interfaces, found at:
http://msdn.microsoft.com/library/en-us/netdir/adsi/active_directory_service_interfaces_adsi.asp

This SDK is used as a reference and provides a simple script for user migration to show the possibilities of that interface.

Microsoft Windows 2000 Resource Kit

The Windows 2000 Resource Kits is a collection of tools to help deploy, manage, and support Windows 2000 operating systems. More information can be found at the following Web site:

<http://www.microsoft.com/windows2000/techinfo/reskit/default.asp>

RMTSHARE

This tool is part of the Resource Kit and provides extended support for managing file and printer shares from the command line. This tool provides functions such as the ability to:

- ▶ Run commands remotely
- ▶ Create, delete, and modify shares for print queues and file resources
- ▶ Manage access privilege for shares

Executing the command without any parameters gives you the following help:

```
RMTSHARE  \\server
          \\server\sharename
```

```

\\server\sharename=drive:path [/USERS:number | /UNLIMITED]
    [/REMARK:"text"]
    [/GRANT [user[:perm] [ /GRANT user[:perm]]]]
    [/REMOVE user]
\\server\sharename=printername /PRINTER [/USERS:number | /UNLIMITED]
    [/REMARK:"text"]
    [/GRANT [user[:perm] [ /GRANT user[:perm]]]]
    [/REMOVE user]
\\server\sharename [/USERS:number | /UNLIMITED]
    [/REMARK:"text"]
    [/GRANT [user[:perm] [ /GRANT user[:perm]]]]
    [/REMOVE user]
\\server\sharename /DELETE

```

NOTE: if a sharename or path contains spaces, it should be enclosed in quotes:
 \\server\"with space"="c:\with space"

This utility is used to migrate the alias definitions and create file and printer shares in 4.6, "Migrating directories" on page 141, and 4.7, "Migrating printers" on page 154.

Robocopy

Robocopy is a command-line tool used for file migration and replication. Specifically, it helps maintain identical copies of a directory structure on a single computer, or in separate network locations. **robocopy** is included in the Microsoft Windows 2000 Resource Kit. If a file exists in both the source and destination locations, by default **robocopy** copies the file only if the two versions have different time stamps or different sizes. This saves time if the source and destination are connected by a slow network link. You can also specify that copies are restarted in the event of a failure, which saves even more time when your network links are unreliable.

The help screen is printed when **robocopy** is started with the parameter /??? (For more information, please read the product documentation or the Resource Kit Web site we mentioned above.)

Example 4-1 Robocopy help

```

ROBOCOPY v 1.96 : Robust File Copy for Windows NT
-----
Started : Tue Jul 01 17:58:21 2003

Usage : ROBOCOPY source destination [file [file]...] [options]

source : Source Directory (drive:\path or \\server\share\path).
destination : Destination Dir (drive:\path or \\server\share\path).
file : File(s) to copy (names/wildcards: default is "*.*").

```

Copy options: /S : copy Subdirectories, but not empty ones.
 /E : copy subdirectories, including Empty ones.
 /LEV:n : only copy the top n LEVels of the source directory tree.

/Z : copy files in restartable mode.

/SEC : copy SECurity info (both source and dest must be NTFS).
 /SECFIX : FIX SECurity info on existing files and dirs.
 /TIMFIX : FIX TIMestamps on existing destination files.

/MOV : MOVE files (delete from source after copying).
 /MOVE : MOVE files AND dirs (delete from source after copying).

/PURGE : delete dest files/dirs that no longer exist in source.
 /MIR : MIRror a directory tree (equivalent to /E plus /PURGE).

/A+:[R] [A] [S] [H] : add the given Attributes to copied files.
 /A-:[R] [A] [S] [H] : remove the given Attributes from copied files.

/CREATE : CREATE directory tree structure + zero-length files only.
 /FAT : create destination files using 8.3 FAT file names only.

File Selection: /A : copy only files with the Archive attribute set
 /M : like /A, but remove Archive attribute from source files.
 /IA:[R] [A] [S] [H] : Include only files with some of the given Attributes set.
 /XA:[R] [A] [S] [H] : eXclude files with any of the given Attributes set.

/XF file [file]... : eXclude Files matching given names/paths/wildcards.
 /XD dirs [dirs]... : eXclude Directories matching given names/paths.

/XC | /XN | /XO : eXclude Changed | Newer | Older files.
 /XX | /XL : eXclude eXtra | Lonely files and dirs.
 /IS : Include Same files.

/MAX:n : MAXimum file size - exclude files bigger than n bytes.
 /MIN:n : MINimum file size - exclude files smaller than n bytes.

/MAXAGE:n : MAXimum file AGE - exclude files older than n days/date.
 /MINAGE:n : MINimum file AGE - exclude files newer than n days/date.
 (If n < 1900 then n = n days, else n = YYYYMMDD date).

Retry Options: /R:n : number of Retries on failed copies: default is 1 million.
 /W:n : Wait time between retries: default is 30 seconds.

/REG : Save /R:n and /W:n in the Registry as default settings.

/TBD : wait for sharenames To Be Defined (retry error 67).

Logging Options: /L : List only - don't copy, timestamp or delete any files.

/X : report all extra files, not just those selected.
/V : produce Verbose output, showing skipped files.

/NP : No Progress - don't display % copied.
/ETA : show Estimated Time of Arrival of copied files.

/LOG:file : output status to LOG file (overwrite existing log).
/LOG+:file : output status to LOG file (append to existing log).

This utility is used to migrate the data to Windows 2000 in 4.6.3, "Migrating the data" on page 150.

CACLS

The **cacls** command is used to edit and display file permissions on NTFS partitions. It is part of the Windows 2000 installation, and can be found in %WINDIR%\SYSTEM32.

Here is a list of the options.

Example 4-2 CACLS options

CACLS file [/T][/E][/C] [/G user:perm] [/R user [...]] [/P user:perm [...]] [/D user [...]]
filename Displays ACLs.
/T Changes ACLs of specified files in the current directory and all subdirectories.
/E Edit ACL instead of replacing it.
/C Continue on access denied errors.
/G user:perm Grant specified user access rights.
 Perm can be: R Read
 C Change (write)
 F Full control
/R user Revoke specified user's access rights (only valid with /E).
/P user:perm Replace specified user's access rights.
Perm can be: N None
 R Read
 C Change (write)
 F Full control
/D user Deny specified user access.
You can specify more than one user in a command.
Wildcards can be used to specify more than one file in a command.

This tool is used to migrate access controls in 4.6.1, "Migrating access control" on page 142.

NetDom

This tool enables administrators to manage Windows 2000 domains and trust relationships from the command line, and is part of the Windows 2000 Support Tools.

Use NetDom to:

- ▶ Join a Windows 2000 computer to a Windows NT 4.0 or Windows 2000 domain, and provide an option to specify the organizational unit for the computer account
- ▶ Manage computer accounts for domain member workstations and member servers:
 - Add, remove, query
 - Provide an option to specify the organizational unit for the computer account
 - Provide an option to move an existing computer account for a member workstation from one domain to another, and maintain the security descriptor on the computer account
- ▶ Establish, view, and enumerate trust relationships between domains running Windows 2000 or Windows NT
- ▶ Verify and reset the secure channel for the following configurations:
 - Member workstations and servers
 - BDCs in a Windows NT 4.0 domain
 - Specific Windows 2000 replicas
 - Manage trust relationships between domains

For more information, please refer to the *Microsoft Knowledge Base Article*, Q266651, titled “Using Netdom 2.0 to Create Computer Accounts on Admin-Specified Domain Controllers” found at:

<http://support.microsoft.com/support/kb/articles/q266/6/51.asp>

This tool is used to migrate server information in 4.3, “Migrating server definitions” on page 103.

IBM Networks Password Synchronization Tool

This tool is used in 4.5.4, “Passwords” on page 129 to migrate the user passwords from OS/2 to Windows 2000. A detailed description of this tool can be found in 8.1.2, “IBM Networks Password Synchronization Tool” on page 281.

4.2 Migrating the domain

Migrating the domain from OS/2 to Windows Active Directory is as simple as creating some organizational units. A convenient way to do this automatically is creating a LDIF file that specifies the required information.

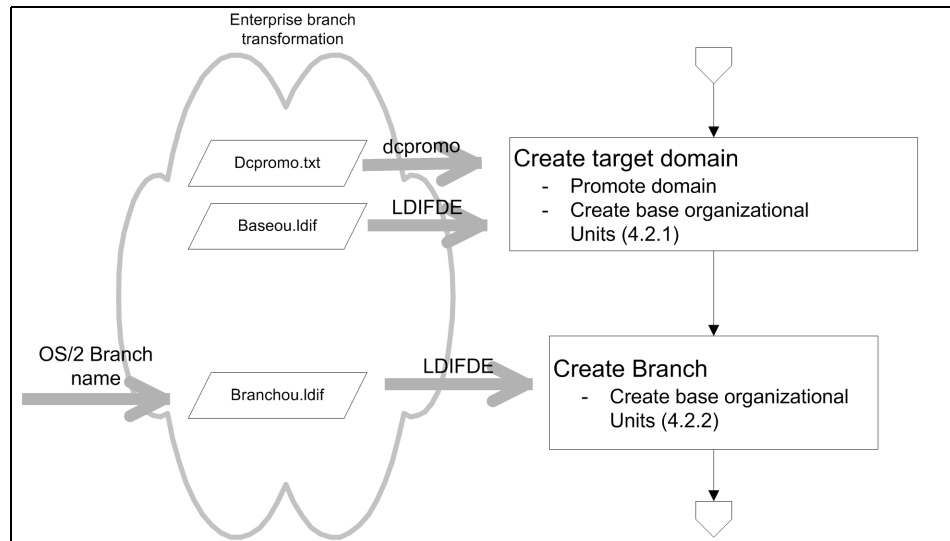


Figure 4-4 Migration workflow for domain

4.2.1 Preparing Active Directory prior to first migration

The creation of the necessary OU is split into two steps. The first has to be executed only once for each target domain since it contains only the general, not branch-specific, OU. The second step will be executed for each domain.

At one domain controller in your Active Directory domain, you execute the following command (an excerpt of the input file is listed in Example 4-4, the full input file is in Appendix , "BASEOU.LDIF" on page 449:

```
ldifde -i -f baseou.ldif -v
```

Example 4-3 Sample ldifde output

```
Connecting to "windc.somedomain.local"  
Logging in as current user using SSPI  
Importing directory from file "baseou.ldif"  
Loading entries  
  
: OU=Branch,DC=somedomain,DC=local  
Entry DN: OU=Branch,DC=somedomain,DC=local
```

```
change: add
Attribute 0) description:Container for all branches
Attribute 1) objectClass:organizationalUnit
Attribute 2) ou:Branch
```

Entry modified successfully.

...

The command has completed successfully

This command imports LDAP objects using the file `baseou.ldif` in verbose mode. The `ldifde` command creates two files (`ldif.err` and `ldif.log`) which report all procedures and errors. They can be found in the current directory where `ldifde` was executed. Example 4-4 shows the result of the import. Each definition consists of five lines specifying the attributes written to LDAP. The example adds an object of the class `organizationalUnit` named `Branch` in the root of the LDAP tree describing the object as a container for all branches.

Example 4-4 Excerpt of LDIF import script to create basic OU once (baseou.ldif)

```
dn: OU=Branch,DC=somedomain,DC=local
changetype: add
description: Container for all branches
objectClass: organizationalUnit
ou: Branch
```

4.2.2 Steps for each domain

This step creates the source domain specific organizational units in the branch context. `ldifde` has a smart feature to do a search and replace before transmitting the script to the LDAP server, so we create a template script to do the work for all branches. We define a variable `{DomainName}` that should be replaced by the current OS/2 domain name that we migrate. In the example, this domain is named `BRANCH1`. The command line to create the OU is as follows:

```
ldifde -i -f branchou.ldif -v -c {DomainName} Branch1
```

This brings the program to import LDAP objects as defined in the file `branchou.ldif` using verbose mode. Before transmitting the commands to the Active Directory server, it changes every occurrence of `{DomainName}` to `Branch1`. The input file and an excerpt of the corresponding protocol are listed in the following two examples.

Example 4-5 LDIF import script to create branch specific OU (branchou.ldif)

```
dn: OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: {DomainName}

dn: OU=Users,OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Users

dn: OU=Groups,OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Groups

dn: OU=Application,OU=Groups,OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
description: Container for groups assigning applications to users
objectClass: organizationalUnit
ou: Application

dn: OU=Access,OU=Groups,OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
description: Container for groups granting access to resources
objectClass: organizationalUnit
ou: Access

dn: OU=Print,OU=Groups,OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
description: Groups for granting access to printer queues
objectClass: organizationalUnit
ou: Print

dn: OU=Organization,OU=Groups,OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
description: Groups defining organisational membership of users (usable as DL)
objectClass: organizationalUnit
ou: Organization
```

Example 4-6 Excerpt of resulting protocol file for creating branches (ldif-branch.log)

```
Connecting to "windc.somedomain.local"
Logging in as current user using SSPI
Importing directory from file "branchou.ldif"
Loading entries
1: OU=Branch1,OU=Branch,DC=somedomain,DC=local
Entry DN: OU=Branch1,OU=Branch,DC=somedomain,DC=local
```



```
change: add
Attribute 0) objectClass:organizationalUnit
Attribute 1) ou:Branch1
```

Entry modified successfully.

```
: OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
Entry DN: OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
change: add
Attribute 0) objectClass:organizationalUnit
Attribute 1) ou:Users
```

Entry modified successfully.

```
[...]
The command has completed successfully
```

4.3 Migrating server definitions

This section discusses the considerations and actions necessary to migrate the additional OS/2 Servers to the new domain. It assumes that the first domain controller is already installed as described in 2.1.1, “Base installation” on page 20, and the Active Directory is set up properly. The migration of servers focuses on two major groups: domain controllers and member servers.

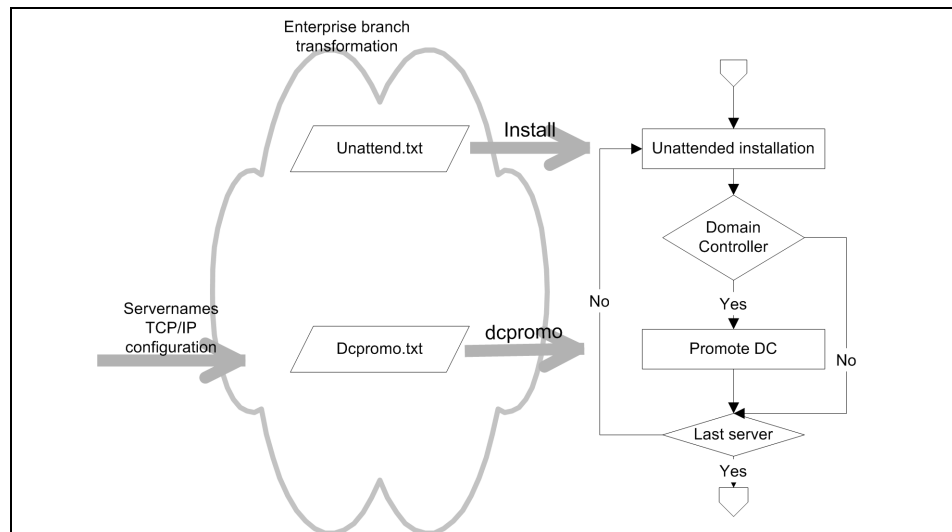


Figure 4-5 Migration workflow for additional servers

4.3.1 Domain controller

The domain controller provides logon services for clients. As in OS/2 domains, Windows domain controllers hold the complete user account database. For Windows 2000 this is part of the Active Directory. To map the functions of the OS/2 domain controller, the following services are mapped to Windows 2000:

- ▶ User and password authentication
- ▶ Logon assignments through the Domain Controller Database (DCDB)

User authentication

Windows 2000 Servers provide backward compatibility for OS/2 clients using LAN Manager authentication. So, after migrating all user accounts and groups, OS/2 clients should see little or no difference.

Note: Windows 2000 domain controllers are also used by Windows 2000 member servers for pass-through authentication. Each member server of a Windows domain keeps its own user and group database providing local accounts that are not replicated within the domain. If an authentication should be done through a domain account (global account), the client has to do this at session setup. The first net use command below uses a local account of a member server to connect, the second command uses the domain account:

```
net use x: \\server\share /user:localuser *  
net use x: \\server\share /user:domain\domainuser *
```

OS/2 clients do not support this distinction, and cannot access resources on member servers protected by a domain based ACL.

Tip: Use Windows 2000 domain controllers to provide resources for legacy OS/2 clients and demote them after finishing your client transition. For non-sensitive resources like some printers, it is also possible to activate the guest account on this machine to grant access.

Logon assignments

IBM OS/2 LAN Server domains use the features of a domain controller database (DCDB) to store alias and logon assignment information. Taking a closer look at this database reveals a directory tree shared by every domain controller running the DCDB-replicator. Clients are able to access this database through the share IBMLAN\$. This approach is not used in OS/2 LAN Manager, nor in Windows NT, or Windows 2000 domains. There are several approaches to do this in a Windows environment:

- ▶ Copy the DCDB subdirectory to each Windows 2000 domain controller to provide a “read-only” backward compatibility for OS/2 clients.
- ▶ Migrate from drive letters and use UNC path names only, and let the user connect his drives using the Windows Explorer and persistent connection.
- ▶ Provide all resources in a distributed file system protecting the branches with discreet group-based ACLs, preventing users from seeing forbidden resources.
- ▶ Use the Active Directory group policy objects to define logon scripts for organizational units (OU) that will be executed depending on the OU in which a user is defined.
- ▶ Use a general logon script that branches out for a user specific routine that creates the assignments.

Tip: Several third party developers provide solutions for this problem. See also Chapter 8, “Additional migration tools” on page 277 for more information.

Steps to follow

After the base installation of the operating system, follow these steps:

1. Promote the server with **DCPROMO** (see Appendix , “DCPROMO2.TXT” on page 428 for an example).
2. Move the server to the correct organizational unit (using the MMC for Active Directory users and computers, or a script).
3. Move the server to the appropriate site (using the MMC for Active Directory sites and services, or a script).

Providing logon services for OS/2 clients

When logging on to a Windows 2000 Active Directory domain, an OS/2 client requires a certain configuration to avoid receiving error messages. The items to consider are:

1. The name of the primary domain controller for the domain.
2. A home directory with a specific syntax that the OS/2 clients can interpret.
3. Access to the DCDB to process the logon assignments and the optional PROFILE.CMD.

The first requirement is usually provided by an Active Directory with one domain controller running the Flexible Single Master Operation (FSMO) role PDC-Emulator.

Note: There is often confusion in the understanding of where the *Native Mode* Active Directory can run. The Native Mode is only necessary if you have the need to run Windows NT 4 Backup Domain Controllers (BDC) in an Active Directory domain. Native Mode still supports Windows NT four-member servers, Windows NT workstations, and all other legacy clients using LAN Manager compatible protocols. The PDC-Emulator continues operating after switching to Native mode.

The second requirement cannot be fulfilled for both environments. OS/2 and Windows NT use a different syntax defining the home directory of a user. When OS/2 users log on to a Windows 2000 domain with a user account having a home directory defined, they will probably receive the following error message:

```
NET8191: Your home directory could not be set up
```

You should consider moving the assignment of a user's home directory to a logon script.

In some cases, the administrator may want the OS/2 clients to still have access to the DCDB. To provide access to these files, the following commands can be included in the PROFILE.CMD for a user:

1. Create a directory on a domain controller:

```
md E:\IBMLAN
```

2. Share this directory as IBMLAN\$ giving all domain users read permissions:

```
rmtshare \\windc\IBMLAN$=E:\IBMLAN/remark:"DCDB for OS/2 clients" /grant  
"SOMEDOMAIN\Domain Users:r"
```

3. Copy the directory x:\IBMLAN\DCDB of the OS/2 primary domain controller into this directory:

```
xcopy \\pdc\ibmlan$\dcdb \\windc\ibmlan$\dcdb /e /i /h /r /k
```

4. If more than one domain controller exists in the domain, configure the Distributed File System (DFS) to replicate this directory providing the same functionality as the DCDB-replicator service. "Replicator service" on page 107 discusses the replicator services in more details.

Attention: It is required to configure your Windows 2000 Active Directory domain in a certain way to allow password changes from an OS/2 client. According to the *Microsoft Knowledge Base Article*, Q135060, titled “Access denied Attempting to Change Client Domain Password,” Microsoft does not support password expiration for down level clients using the LAN Manager protocol (XACT-SMB). This is because of a change in the authentication of a pipe used for the password change protocol. Although it is not documented or recommended due to security reasons, you may add this pipe to the list of null-session shares, and run the domain in pre-Windows 2000 compatible access mode.

4.3.2 Member servers

Member servers provide several different services in a Windows 2000 environment. To identify them easily, they are put into separate organizational units. These target OUs can be defined in either of these ways:

- ▶ Define the OU within the installation process by specifying the parameter `machineObjectOU` in the Identification section of the `unattended.txt` for Windows:

```
[Identification]
MachineObjectOU = "OU=File,OU=Servers,OU=Systems,DC=somedomain,DC=local"
```

- ▶ Use the `netdom.exe` utility from Microsoft Windows 2000 support to join the domain:

```
netdom add /domain:somedomain
/ou:"OU=File,OU=Servers,OU=Systems,DC=somedomain,DC=local"
```

- ▶ Write a script using ADSI to perform this action.

4.3.3 Common issues

In general, the transition of servers to the Windows 2000 environment works without major drawbacks. However, the following should be kept in mind when migrating to Windows 2000.

Replicator service

IBM OS/2 Warp Servers can be configured to synchronize the content of a list of files on each server in a domain with another server. This functionality is called REPLICATOR. Windows 2000 is not backwards compatible with this functionality. It has been replaced with the File Replicator Service (FRS). FRS can also replicate data for Distributed File Systems (DFS), synchronizing the content of each member in a replica set defined by DFS. FRS can copy and maintain shared files and folders on multiple servers simultaneously. Refer to the *Microsoft*

Knowledge Base Article, Q161431, titled “Windows 2000 Does Not Support Windows NT 4.0 Directory Replication (LMRepl)” for more information on this topic, which can be found at:

<http://support.microsoft.com/?kbid=248358>

Another source is Chapter 18., “File Replication Service” from the publication *Windows 2000 Server Distributed System Guide*, found at:

<http://www.microsoft.com/TechNet/prodtechnol/windows2000serv/reskit/distsys/part/dsgch18.asp>

Additional server names (othsrvnames)

The IBM LAN Server is able to use multiple NetBIOS names, while Windows 2000 cannot. There is no direct mapping of this feature within Windows 2000, so name resolution using DNS might be an approach. The *Microsoft Knowledge Base Article, Q161431*, titled “Connecting to NetBIOS Resources Using DNS Names or IP Addresses” gives some information regarding this topic. It can be found at:

<http://support.microsoft.com/?kbid=161431>

Time source

As in OS/2, in Windows 2000, there is a urgent need for precise, domain-wide synchronized time, especially authentication protocols such as Kerberos, which depend on synchronized clocks. In OS/2 domains, a server can act as the time server to signal that it is a reliable time source.

Windows 2000 supports the Simple Network Time Protocol (SNTP) using the following commands to specify and query the name of a server that delivers the correct time:

```
net time /setsntp:<timesource.somedomain.local>
net time /querysnTP
```

Tip: There are several time servers available providing a reliable time signal through the Internet. Consider using one of these external services.

4.4 Migrating groups

The following section describes the migration of all group accounts. This step is quite easy because there is a direct way to map each OS/2 group attribute to an appropriate Windows 2000 attribute. In our example, global groups were used to keep things simple. One could modify this example to use local groups.

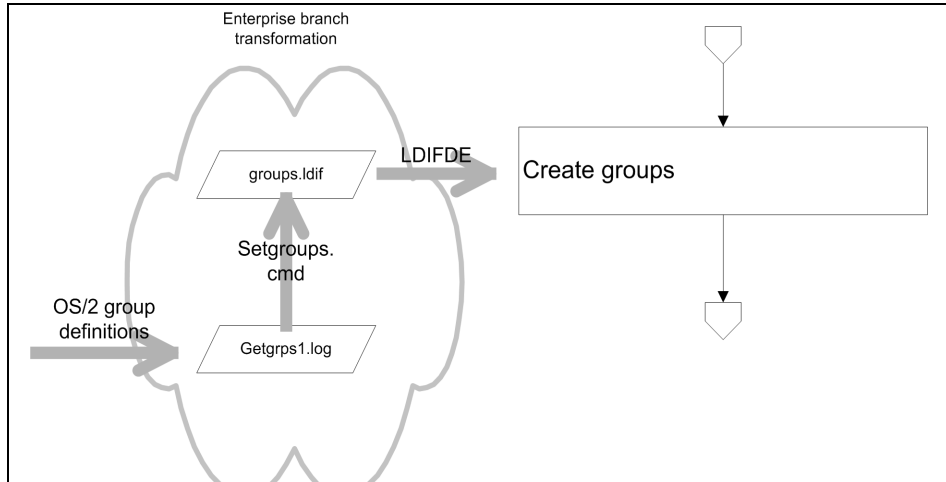


Figure 4-6 Migration workflow for group part

4.4.1 Before you start

In the given example, LSMT writes the group definitions to a file named `getgrps1.log`.

Example 4-7 OS/2 group definitions from example OS/2 domain (getgrps1.log)

```

OPT;NAME           ;COMMENT
;ADMINS            ;
;BOOKREAD         ;
;BOOKWRITE        ;
;GROUPID          ;Default Group ID
;GUESTS           ;
;LOCAL            ;
;PRINTER          ;Printer Group
;SERVERS          ;System ID - Server
;TRANSITION       ;
;USERS            ;

```

Tip: It is a good idea to consider a redesign of groups in your domain. You may change the naming conventions, therefore, helping you to more easily identify the purpose of each group. You can use groups more extensively because the OS/2 LAN Server restriction of 254 groups does not exist for Windows 2000 domains. Because LSMT provides the data in an ASCII format, you can easily modify and add new groups before importing them to Windows 2000.

The basic idea of the concept of migrating groups to Windows 2000 Active Directory is parsing this output file and creating an LDIF file that Active Directory services is able to process. To create a new group object, you only need to specify where the object should be created, provide an optional comment, and give a unique name for the group. This group name should be identical to the back level Windows NT4 group name that has to be specified by the attribute sAMAccountName. The following example shows the format of a group object definition for a global group object:

```
dn: CN=PRINTER,OU=Print,OU=Groups,OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: PRINTER
description: Printer Group
distinguishedName: CN=PRINTER,CN=Users,DC=somedomain,DC=local
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: group
name: PRINTER
sAMAccountName: PRINTER
```

Section 4.1.2, “Design of Active Directory” on page 89 describes the design of the Active Directory. This includes four organizational Units (OU) for a group object:

- OU=Access Defined as the container holding group objects that grant access to resources directories and files
- OU=Print This OU holds all defined group objects that specify access rules to printer objects.
- OU=Application Groups that grant access to published applications (e.g. Citrix Metaframe)
- OU=Organization Groups defined here specify the membership to a particularly group of persons in an enterprise view. These include distribution lists, project teams, or workgroups.

To map the groups from Example 4-7, the REXX script uses the first column (OPT) to map them into the specific context. The following table describes this mapping.

Table 4-1 Mapping group definitions using the OPT column

OPT	Action
<blank>	This line will be ignored in the transformation process. With this option you do not have to remove unwanted groups from the export file.
A	This group definition is treated as an access group. This group is migrated to the OU=Access

OPT	Action
O	This group definition describes an organizational group. It is migrated to the OU=Organization
P	This group definition describes a group granting access to print queues. It is migrated to the OU=Print
X	This group definition is treated as an application group. This group is migrated to the OU=Application

Taking the given example, the file was modified and one new group (BRANCH1) was added resulting in the file shown in Example 4-8.

Example 4-8 Modified OPT file

```

OPT;NAME           ;COMMENT           ;
  ;ADMINS           ;                   ;
A  ;BOOKREAD       ;                   ;
A  ;BOOKWRITE      ;                   ;
  ;GROUPID         ;Default Group ID  ;
  ;GUESTS          ;                   ;
  ;LOCAL           ;                   ;
P  ;PRINTER        ;Printer Group     ;
  ;SERVERS         ;System ID - Server ;
A  ;TRANSITION     ;                   ;
  ;USERS           ;                   ;
O  ;BRANCH1        ;All users of branch 1 ;

```

The group definitions are transformed to an LDIF specific format with the **setgroups** command.

```
setgroups win getgrps1.log groups.ldif Branch1
```

The first parameter specifies the target platform, in this case it is Windows. The next two parameters provide the filenames for the input and output file. The last parameter contains the name of the target branch in the Active Directory tree.

The resulting file is shown in Example 4-9.

Example 4-9 Created LDIF file from setgroups.cmd

```

dn: CN=BOOKREAD,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: BOOKREAD
distinguishedName: CN=BOOKREAD,CN=Users,DC=somedomain,DC=local
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: group
name: BOOKREAD

```

sAMAccountName: BOOKREAD

dn: CN=BOOKWRITE,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: BOOKWRITE
distinguishedName: CN=BOOKWRITE,CN=Users,DC=somedomain,DC=local
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: group
name: BOOKWRITE
sAMAccountName: BOOKWRITE

dn: CN=PRINTER,OU=Print,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: PRINTER
description: Printer Group
distinguishedName: CN=PRINTER,CN=Users,DC=somedomain,DC=local
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: group
name: PRINTER
sAMAccountName: PRINTER

dn: CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: TRANSITION
distinguishedName: CN=TRANSITION,CN=Users,DC=somedomain,DC=local
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: group
name: TRANSITION
sAMAccountName: TRANSITION

dn: CN=BRANCH1,OU=Organization,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: BRANCH1
description: All users of branch 1
distinguishedName: CN=BRANCH1,CN=Users,DC=somedomain,DC=local
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: group
name: BRANCH1
sAMAccountName: BRANCH1

In the REXX script listed in Appendix , “SETGROUPS.COM” on page 501, it can be seen that the script produces another output file named group-db.csv. This file is created as a lookup database that translates the non-hierarchical OS/2 group names to the matching LDAP path names. This database will be used in 4.5.3, “Group membership” on page 127, where users will be assigned as members of given groups:

Example 4-10 Lookup database group-db.csv

```
BOOKREAD;CN=BOOKREAD,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
BOOKWRITE;CN=BOOKWRITE,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
PRINTER;CN=PRINTER,OU=Print,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
TRANSITION;CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
BRANCH1;CN=BRANCH1,OU=Organization,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
```

4.4.2 Steps to follow

To perform the migration of group definitions from OS/2 to Windows 2000, Active Directory follows these steps:

1. Get the export file `getgrps1.log` using the LSMT as described in 3.3.3, “Groups” on page 72.
2. Modify the entries and add an A, O, P, or X in the column OPT for the groups you want to transfer to the target domain.
3. Change descriptions, group names, or add additional groups you need in the Windows 2000 domain for your branch.
4. Run the command `setgroups.cmd` with the following parameters:

```
setgroups win getgrps1.log groups.ldif Branch1
```
5. Save the file `group-db.csv` for use in 4.5.3, “Group membership” on page 127.
6. Import the group definitions to Active Directory with the following command:

```
ldifde -v -i -f groups.ldif
```
7. Save the log files `ldif.err` and `ldif.log`.

4.5 Migrating users

The following section describes the migration of all user account related objects and definitions. Several possible scenarios are discussed, but only one is described in depth.

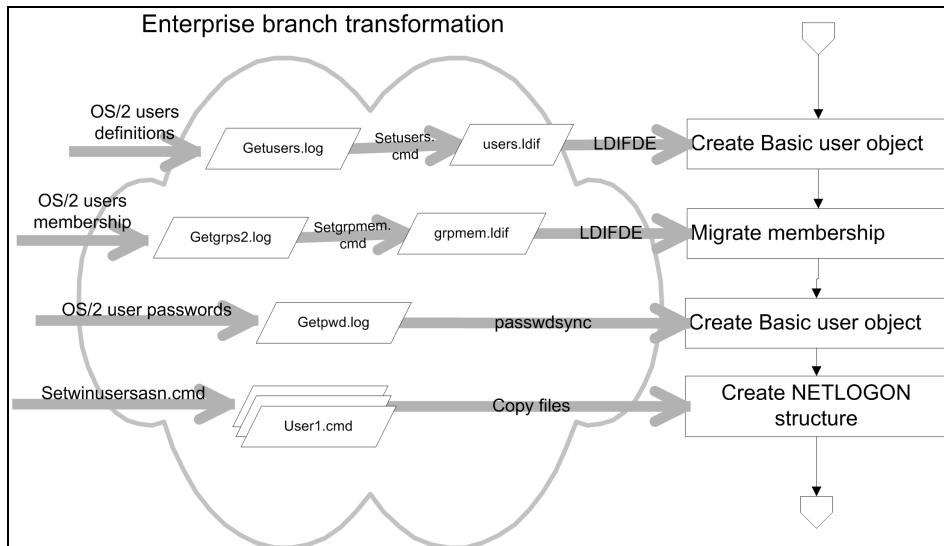


Figure 4-7 Migration workflow for user part

4.5.1 Where to start

Having a closer look at a user object in OS/2 using the **net user** command, several attributes can be discovered, which one can group as follows:

Authentication	Username, Password, Group membership, Privileges, Operator rights, Account disabled, Account expires, Workstations, Logon hours, Password expires, User can change password, Password required
Identification	Fullname, Comment, User comment
Environment	Parameter, Country code, Maxstorage, Logon Server, Logon Script, Homedirectory, logon assignments
Statistics	Password age, Last logon, last logoff, Bad password count.

In the following sections only the first three groups of attributes are mapped, because statistical information is not mandatory for the migration. These three groups of attributes are processed in three distinct steps:

1. Basic user object and group membership
2. Passwords
3. Logon assignments

This distinction is necessary because of the tools and procedures that we use for the migration process.

4.5.2 Basic user object

There are several ways to create new user objects in Active Directory, amongst them are:

1. Microsoft Management Console Active Directory Users and Computer to create users through a graphical interface.
2. The command line program **net user** already known from OS/2
3. Visual Scripting Host using Microsoft Active Directory Services Interface SDK (ADSI SDK).
4. Third-party products or employee written software usually written in C++ or Visual Basic using ADSI SDK.
5. CSV formatted files processed by **csvde** and, respectively, LDIF files process by **ldifde**.

While option 1 is unsuitable to automatically create large numbers of user objects, the second option is helpful only to a limited extent in Active Directory environments because it creates user objects in the default container CN=Users and provides backward compatibility for Windows NT domains. We will take a look at third party products in Chapter 8, “Additional migration tools” on page 277, so this leaves the remaining options 3 and 5 to discuss.

The basic user object in Windows 2000 Active Directory is an instance of the object class *user*. To start discussing the migration path to Windows 2000 it is a good idea to create a simple user account and check the attributes that are necessary for your environment:

1. Create a user within the MMC in the default container (cn=users) and set all attributes you use in your OS/2 environment. In our example, we called the user John.
2. Export this user definition with **ldifde** using the following command:

```
ldifde -v -f john.ldif -d "cn=john,cn=users,dc=somedomain,dc=local"
```
3. View the output file john.ldif to see the results. In the following example, the bold lines contain attributes we need to set for the migration.

Example 4-11 LDIF definition of sample user John

```
dn: CN=JOHN,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local  
changetype: add  
memberOf:  
CN=BOOKREAD,OU=Access,OU=Groups,OU=Branch1,OU=Branch,DC=somedomain,DC=local
```

```

memberOf: CN=Account Operators,CN=Builtin,DC=somedomain,DC=local
accountExpires: 127037556000000000
badPasswordTime: 0
badPwdCount: 0
codePage: 0
cn: JOHN
countryCode: 0
description: Sample user
displayName: John Doe
givenName: John
homeDirectory: \\windc\john
homeDrive: Y:
instanceType: 4
lastLogoff: 0
lastLogon: 0
logonCount: 0
logonHours:: AAAAApj/A/j/A/j/A/j/A/j/AwAA
distinguishedName:
  CN=JOHN,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: user
objectGUID:: PkB318ruX029LVyw6ipSCg==
objectSid:: AQUAAAAAAAAUAAAA9TZFSZp81jaoN9Z1baQAAA==
primaryGroupID: 513
pwdLastSet: 127011211464459152
name: JOHN
sAMAccountName: john
sAMAccountType: 805306368
scriptPath: logon.cmd
sn: Doe
userAccountControl: 512
userPrincipalName: john@somedomain.local
userWorkstations: JOHNSPC
uSNChanged: 3861
uSNCreated: 3844
whenChanged: 20030626171854.0Z
whenCreated: 20030626171225.0Z

```

Most of the attributes are in a self-explanatory form, so only a subset are explained below. For more details, please refer to the documentation of Microsoft Active Directory SDK (see “Active Directory Services Interface” on page 95 to get more information about obtaining this from Microsoft).

accountExpires This property specifies when the account will expire. Microsoft describes this value as the number of seconds elapsed since 00:00:00, January 1, 1970. This does not match with the values the API or `ldifde` returns. Referring to the documentation, Microsoft uses the file time format

for all date attributes, and is described as a value representing the number of 100-nanosecond intervals that have elapsed since 12:00 am, January 1, 1601 (UTC).

logonHours	The logonHours attribute is a string of “0” and “1” specifying which hours in a week a user is allowed to log on. Starting at Sunday 12 am, the string consists of 168 digits having “1” if the user is allowed, “0” if not. In the LDIF export file you see a BASE64 encoded representation of this bitmap.
primaryGroupld	Each user and group in Windows 2000 has an ID called <i>security identifier</i> (SID) that is absolutely unique. While the first five tokens specify the domain SID, the last token is unique for each account, and is called a relative identifier (RID). Predefined accounts like “Domain Admins” have identical RIDs. The primaryGroupld property contains this RID. In our test domain, the SID for Domain Users is: S-1-5-21-1229272821-920026266-1708537768-513, so primaryGroupID contains the value 513. Microsoft defines the following values for predefined accounts: 512 Domain Admins 513 Domain Users 514 Domain Guests
userAccountControl	Similar to the encoding in an OS/2 LAN Server domain, Windows 2000 encodes user attributes like disabled account or the account type. Because there is no direct mapping of the other OS/2 account flags, please refer to the Microsoft ADSI SDK for a list of the options.

Tip: With Windows 2003 Server, Microsoft has extended the Active Directory schema to support a new user class INetOrgPerson, which provides more compatibility than the provided standard user class. For heterogeneous environments this may be a reason to migrate directly to Windows 2003 Server.

Table 4-2 shows the generated and required Active Directory attributes from given OS/2 user account properties. The first column contains the name of the attribute in Active Directory, the second the name of the OS/2 attribute and the last column the necessary transition steps.

Table 4-2 Transformation matrix for Active Directory user objects

AD attribute	SourceOS/2 attribute	transition steps
dn	NAME	The OS/2 attribute is formatted in an LDAP style distinguished name including the complete path.
unicodePwd		The password is left empty in our approach. In a second step we will directly write the LAN server hash into the user object attribute.
givenName	COMMENT	There is no one-to-one correspondence for that attribute. If the COMMENT attribute was used to specify the full name of a user, it can be parsed like using the second word of the COMMENT attribute.
sn	COMMENT	There is no one-to-one correspondence for that attribute. If the COMMENT attribute was used to specify the full name of a user, it can be parsed like using the first word of the COMMENT attribute.
displayName	NAME	There is no one-to-one correspondence for this attribute. If the COMMENT attribute was used to specify the full name of a user, it can be parsed to get the full name. We use the NAME attribute instead.
description	USR_COMMENT	Some additional description may be available in the USR_COMMENT field, so we use this as best match.
userPrincipalName	NAME	The principal name is a combination of an unique user ID and a valid DNS domain name (like an e-mail address).
pwdLastSet		Can be set to 0 (zero) to expire password. Because we use a not well documented way to migrate passwords, it is reasonable to set all passwords to expired to reset the Active Directory attributes with new values.

AD attribute	SourceOS/2 attribute	transition steps
sAMAccountName	NAME	As long the user does not use his principal name (username@somedomain.local) the sAMAccountName is used for logon especially from legacy clients such as OS/2
maxStorage	MAX_STORAGE	This attribute is still available, but not used by Active Directory. We transfer it one-to-one.
codePage	CODE_PAGE	This attribute will be migrated directly
countryCode	COUNTRY_CODE	This attribute will be migrated directly
logonHours	1.LOGON_HOURS 2.LOGON_HOURS 3.LOGON_HOURS 4.LOGON_HOURS 5.LOGON_HOURS 6.LOGON_HOURS 7.LOGON_HOURS	The logonHours attribute is a string of "0" and "1" specifying on which hours in a week a user is allowed to log on. Starting at Sunday 12 am, the string consists of 168 digits having "1" if the user is allowed during that period, "0" if not. The script creates this bitmap, and encode the resulting 21 bytes that contain non-ASCII characters into an BASE64 format. There is not a direct way to move this attribute to Active Directory using Visual Scripting Host, because the ADSI interface does not support the given variable type.
userAccountControl	FLAGS	Though the user account control attribute is still supported in Active Directory, it is used in a very different manner. Except from the ACCOUNT_DISABLED (2) there is no one-to-one matching at all. We only migrate this attribute, adding an hexadecimal value of 0x200 (512) to this, which means for Active Directory that this is a normal account.
userWorkstations	WORKSTATIONS	This field can be mapped directly to an array of computer names in Active Directory. Because LSMT uses the space as a separator instead of ADSI using the comma, we translate each space character to comma.

AD attribute	SourceOS/2 attribute	transition steps
scriptPath	SCRIPT_PATH	Although OS/2 LAN Server provides this attribute it was not used because of the mechanism of DCDB and PROFILE.COMD. We set this value to a static value (logon.cmd) to provide a similar functionality at logon time.
homeDrive	HOME_DIR	This attribute defines the drive letter assigned to the home directory for WIndows 32-bit Clients. We can map it directly to the first character of the OS/2 HOME_DIR attribute
homeDirectory	HOME_DIR	The home directory has a very different format comparing OS/2 and Windows. In OS/2 it is more the server view, defining which server should share which local drive dynamically using the users accounts as share name. In Windows, it is a users view, defining the UNC for the net use command. For that reason, we separate the second word using the back slash as a separator as the servers name sharing the home directory and the users name to create an UNC path like \\server\username
accountExpires	ACCT_EXPIRES	This value is also supported on both systems. Active Directory specifies this point of time in 100-nanoseconds since Jan. 1th 1601 using GMT. So, the REXX script needs to convert this date to a proper format. Because of time functions only available in OBJREXX, you need to active Object REXX first with the switchrx command.

Important: If it is required to run a mixed environment having Windows and OS/2 clients, it is recommended to leave the home directory attribute empty and assign the home directory through the logon script as discussed in “User specific logon script (users\

It should be noted that not all attributes LSMT retrieves were used. Table 4-3 describes these and the additional steps necessary to map them to proper Windows 2000 attributes.

Table 4-3 OS/2 user attributes without direct mappings to Windows 2000

OS/2 Attribute	Transition steps
PRIV	Because this value is not available with Active Directory, it was not migrated.
AUTH_FLAGS	Active Directory maps this functionality with built-in local groups found in the container CN=Built in. For this reason out script needs to map this attributes to a membership in the following groups: P maps to CN=Print Operator,CN=Builtin, A maps to CN=Account Operators,CN=Builtin and S maps to CN=Server Operators,CN=Builtin C cannot be mapped, because Windows 2000 does not support Serial Device operators.
PARMS	Because this value is not available by Active Directory, it as not migrated.
LOGON_SERVER	Because this value is not available by Active Directory, it was not migrated.
FULL_NAME	The OS/2 LAN Server uses this attribute to provide a description of server objects, since these are not transferred in this step of the migration, we do not use this attribute.

Having this completed the transition table, the next step is to consider the tools that can be used to do this transition. As already described in this chapter, there are two alternatives:

1. Visual Scripting Host using Microsoft Active Directory Services Interface SDK (ADSI SDK).
2. CSV formatted files processed by `csvde`, and LDIF files process by `ldifde`.

Migrating users using Visual Scripting Host

With Windows 2000 the Visual Scripting Host became the standard scripting language with a very powerful feature set using COM objects to extend the language with new objects. For Active Directory, you can use the `GetObject` function to retrieve any object using the LDAP distinguished name. The following sample retrieves the Organizational Unit users from our example where branch users will be created.

```
Set objOU =  
GetObject("LDAP://OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local")
```

This object can be used to create new objects within this container. The following sample will create a user within this container.

```
objUsr = objOU.Create("user", "CN=John")
```

After creation of the user object, each attribute of the object can be accessed using one of the two methods:

```
objUsr.Put "userPrincipalName", 'john@somedomain.local'  
objUsr.userPrincipalName = 'john@somedomain.local'
```

After setting the attributes, the changes have to be committed with the setInfo method:

```
objUsr.setInfo
```

The basic mandatory instructions to create a user are in the following code snippet.

Example 4-12 Code snippet for creating a user using Visual Scripting Host

```
Set objOU = GetObject("LDAP://cn=users,dc=somedomain,dc=local")  
if Err.Number Then  
    wscript.echo "Error in opening organizationalUnit."  
    Exit Sub  
End If  
  
Set objUsr = objOU.Create("user", "cn=John")  
If Err.Number>0 Then  
    wscript.echo "Error creating user."  
    Exit Sub  
Else  
    objUsr.Put "sn", 'Doe'  
    objUsr.Put "givenName", 'John'  
    objUsr.Put "displayName", 'John'  
    objUsr.Put "userPrincipalName", 'john@somedomain.local'  
    objUsr.Put "samAccountName", 'JOHN'  
    objUsr.SetInfo  
end if
```

With this knowledge, the script **createuser.vbs** was created, which can be found in Appendix , "CreateUser.vbs" on page 454.

All other migration steps can also be done using this method with existing REXX scripts that can be easily adapted to produce LDIF formatted files. However, we recommend the other method using **ldifde**, as described in the following section.

Migrating users using LDIFDE

Microsoft made available two tools for bulk import and export from Active Directory with Windows 2000 Server. Providing similar functions, the file formats differ. The command `csvde` uses the commonly known CSV format (comma separated value), while `ldifde` uses the LDAP data interchange format (LDIF). The latter was used to simplify the exchange of data between Windows 2000 and LINUX environments.

The preparation of the input files is described in more detail in 3.3.4, “Users” on page 74. Six users are marked for input into the Windows 2000 domain, as seen in Example 4-13.

Example 4-13 Excerpt of the input files for setusers.cmd

```
OPT;NAME      ;PASSWORD;PASSWORD_AGE;PRIV      ;HOME_DIR      ;
...
A ;ANDREI     ;****      ;870047      ;User           ;U:\PDC\E$\LANHOMES\ANDREI ;
...
      ;BDC       ;****      ;162218      ;User           ;-none-           ;
...
      ;GUEST     ;****      ;1375390     ;Guest          ;-none-           ;
...
A ;LEIF       ;****      ;1372736     ;User           ;U:\PDC\E$\LANHOMES\LEIF   ;
...
A ;MARC       ;****      ;1372735     ;User           ;U:\PDC\E$\LANHOMES\MARC   ;
...
      ;MICHAEL   ;****      ;8652        ;User           ;H:\LNXSLES\MICHAEL       ;
...
      ;MIKE      ;****      ;150749      ;User           ;R:\PDC\C$\HOME\MIKE       ;
...
A ;OLIVER     ;****      ;1372735     ;User           ;U:\PDC\E$\LANHOMES\OLIVER ;
...
      ;PDC       ;****      ;1375391     ;User           ;-none-           ;
...
A ;RICHARD    ;****      ;1372735     ;User           ;U:\PDC\E$\LANHOMES\RICHARD ;
...
      ;USERID    ;****      ;426648862   ;Administrator ;-none-           ;
...
A ;WYNAND     ;****      ;242169      ;User           ;U:\PDC\E$\LANHOMES\WYNAND ;
...

```

Taking a closer look at the user WYNAND, all of his attributes are shown in Table 4-4.

Table 4-4 OS/2 user attributes for user WYNAND

Attribute	Value
OPT	A
NAME	WYNAND
PASSWORD	****
PASSWORD_AGE	242169
PRIV	User
HOME_DIR	U:\PDC\E\$\LANHOMES\WYNAND
COMMENT	Wynand_Pretorius
FLAGS	S
SCRIPT_PATH	-none-
AUTH_FLAGS	PCSA
FULL_NAME	-none-
USR_COMMENT	Standard Bank user
PARAMS	-none-
WORKSTATIONS	PC1 PC2
LAST_LOGON	Thu Jun 12 12:40:07 2003
LAST_LOGOFF	Thu Jun 12 12:40:17 2003
ACCT_EXPIRES	(null)
MAX_STORAGE	No limit
RESTRICTED_HOURS	Restrictions provided
1.LOGON_HOURS	
2.LOGON_HOURS	7 8 9 10 11 12 13 14 15 16 17 18
3.LOGON_HOURS	7 8 9 10 11 12 13 14 15 16 17 18
4.LOGON_HOURS	7 8 9 10 11 12 13 14 15 16 17 18
5.LOGON_HOURS	7 8 9 10 11 12 13 14 15 16 17 18
6.LOGON_HOURS	7 8 9 10 11 12 13 14 15 16 17 18
7.LOGON_HOURS	

Attribute	Value
BAD_PW_COUNT	4
NUM_LOGONS	476
LOGON_SERVER	*
COUNTRY_CODE	000
CODE_PAGE	0

The LDIF file is created by executing the following command, specifying Windows as the target platform, the LSMT input file, the output file, and the name of the branch where the users will be created:

```
setusers win getusers.log users.ldif Branch1
```

Passing this single user through the **setusers.cmd**, generates the following LDIF file.

Example 4-14 Sample LDIF file for a single user

```
dn: CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: WYNAND
distinguishedName:
CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: user
givenName: Wynand
sn: Pretorius
displayName: WYNAND
name: WYNAND
userPrincipalName: WYNAND@somedomain.local
description: Standard Bank User
pwdLastSet: 0
sAMAccountName: WYNAND
codePage: 0
countryCode: 0
logonHours:: AAAAAf/gAf/gAf/gAf/gAf/gAAAA
userAccountControl: 512
userWorkstations: PC1,PC2
scriptPath: logon.cmd
homeDrive: U
homeDirectory: \\PDC\WYNAND

dn: CN=Print Operators,CN=Builtin,DC=somedomain,DC=local
changetype: modify
add: member
```

```

member: CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
-

dn: CN=Account Operators,CN=Builtin,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
-

dn: CN=Server Operators,CN=Builtin,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
-

dn: CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: primaryGroupID
primaryGroupID: 513

```

The assignment of group memberships for a user in Active Directory requires a special procedure. For security reasons, the Security Accounts Manager (SAM) prohibits access to the attribute `memberOf` using the LDIF interface the `add` syntax. Trying to access this attribute will result in an error message such as this:

```

Add error on line 1: Unwilling To Perform
The server side error is "Access to the attribute is not permitted because the
attribute is owned by the Security Accounts Manager (SAM)."
```

Related to this, the `primaryGroupID` also cannot be set since the needed membership in the appropriate group is not set at the time the `primaryGroupID` attribute is processed. The correct procedure to import these attributes is as follow:

1. Create user object using the `add` syntax.
2. Add the user to the defined groups by using the `modify` syntax for the group object and changing the attribute `member` instead of the user object and its the `memberOf` attribute.
3. Modify the user object in a second step to change the `primaryGroupID` attribute.

You can see the resulting syntax in example 4-14, which can be generated by using the SAM logic when exporting Active Directory attributes using the following command:

```
ldifde -f users.ldif -m -r "(objectClass=user)"
```


4.5.3 Group membership

The mechanism to define the group membership for a user was discussed in 4.5.2, “Basic user object” on page 115. Because of limited access to the SAM, values cannot be added to the attribute memberOf but rather requires the addition of members to the specific group. A typical LDIF record to do this looks like:

```
dn: CN=Domain Admins,CN=Users,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
```

The transition step is similar to the other steps described in this part of the redbook. Generate an export file using LSMT, modify, add, or delete entries, and use it as the input file for the transition script `setgrpmem.cmd`

Important: LSMT adds three columns for the groups users, guests, and admins to the export file. These groups are special as users cannot be added to these groups. They are auxiliary groups to represent the privilege level of the user, which we already migrated in 4.5.2, “Basic user object” on page 115. Any changes to these columns are ignored within the migration.

To migrate the membership to the Windows 2000 Domain an ‘A’ option has to be set in the first column and optionally the associated column modified. In case additional groups were added as seen in 4.4, “Migrating groups” on page 108, additional columns need to be added to the file and the membership marked as required.

Tip: Remove the columns for the groups ADMINS, GUESTS, USERS, and all groups not required for migration. Otherwise, the resulting LDIF file generates an error because a group cannot be found.

In contrast to OS/2, Windows 2000 Active Directory needs the distinguished name for the group. OS/2 only supplies the common name. This is the reason for creating the group lookup database group-d.csv that was created in 4.4, “Migrating groups” on page 108. Having the modified LSMT file and this database ready, creation of the LDIF file for group membership can be started using the following command:

```
setgrpmem win getgrps2.log grpmem.ldif Branchname
```

The input file used and parts of the generated output files are shown in the following examples. The full LDIF file can be found at Appendix , “GRPMEM.LDIF” on page 463.

Example 4-15 Modified getgrps2.log ready to import

```
* Do not modify a user from the ADMINS, GUEST, SERVERS or USERS groups *
OPT;USERS
;BOOKREAD;BOOKWRITE;GROUPID;LOCAL;PRINTER;SERVERS;TRANSITION;BRANCH1;
A ;ANDREI ; X ; ; ; ; ; ; X ; X
;
; ;BDC ; ; ; ; ; ; X ; ;
;
; ;GUEST ; ; ; ; ; ; ; ; ;
;
A ;LEIF ; X ; ; ; ; X ; ; X ; X
;
A ;MARC ; X ; ; ; ; X ; ; X ; X
;
A ;OLIVER ; ; X ; ; ; X ; ; X ; X
;
; ;PDC ; ; ; ; ; ; X ; ;
;
A ;RICHARD ; X ; ; ; ; ; ; X ; X
;
; ;USERID ; ; ; X ; ; ; ; ;
;
A ;WYNAND ; X ; ; ; ; ; ; X ; X
;
```

Example 4-16 Group lookup database group-db.csv

```
BOOKREAD;CN=BOOKREAD,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
BOOKWRITE;CN=BOOKWRITE,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
PRINTER;CN=PRINTER,OU=Print,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
TRANSITION;CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
BRANCH1;CN=BRANCH1,OU=Organization,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
```

Example 4-17 Excerpt of resulting grpmem.ldif

```
dn: CN=BOOKREAD,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=ANDREI,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=ANDREI,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
```

This LDIF file can be imported to add the users to the desired groups using the following command:

```
ldifde -v -i -f grpmem.ldif
```

4.5.4 Passwords

IBM OS2/ LAN Server uses an encryption that generates a hashed value of a user's password. This is created by taking the user's plaintext password, capitalizing it, and either truncating to 14 bytes or padding to 14 bytes with null bytes. This 14 byte value is used as two 56-bit DES keys to encrypt a "magic" 8 byte value, forming a 16 byte value, which is stored by the server and client. This hashed password is part of the user object and stored in the accounts database NET.ACC. Windows NT encryption consists of doing an MD4 hash on a Unicode version of the user's password. This also produces a 16 byte hash value that is non-reversible.

Windows 2000 differs from NT by using Kerberos password authentication. Kerberos works by considering the password as a private key, and then gets data from the server, which is encrypted with the key and returned to the server. The server then checks the encrypted information, and if it can decrypt it with the password, the user is authenticated. This works only with other Windows 2000 systems and within a Windows 2000-only environment by which Windows 2000 still maintains a Windows NT password for backward compatibility.

Although there is no documented and supported way to transfer passwords from the OS/2 domain to Windows 2000, there are several approaches to solve this problem to enable a smooth migration for the users:

1. Replace the logon client before starting the migration. The client is the only point where the plain text password is available. In this approach the documented API is used to set a password on the target systems. The drawback of this approach is the huge impact to the client systems, especially if there is a heterogeneous client infrastructure in the branch environments. In some cases there are only Windows 2000 logon clients available, which means you have to complete your client migration prior starting the server migration.
2. Install a password synchronization tool before starting the migration. This keeps the accounts on the source and the target always in sync.
3. Transfer the password hashes once during migration with available utilities.
4. Living with this missing feature, setting each account to an initial value and sending each user a letter with the new password.

For the migration scenario, method three from the list above was chosen, for which there is a tool available from IBM named IBM Networks Password

Synchronization Tool, and an undocumented command line utility **pwdexp** supplied with LAN server.

PWDEXP

As a part of the LSMT tools, you can find two small utilities named **pwdexp.exe** and **pwdimp.exe**. These two undocumented IBM utilities allow you to modify the password hash of a user directly. While **pwdimp** imports a given user-hash combination, **pwdexp** exports this hash. Both utilities only run on the local machine. The **getpwd.cmd** interface was used to get the password hash for each user of the domain. **Pwdexp** prints the result to the console in the following format:

```
<userid>:<16-digit-password-hash>
```

Example 4-18 Password export file generated by LSMT using PWDEXP

```
ANDREI:CD017457761C8B05AAD3B435B51404EE  
LEIF:32DD5DAB4DC507A4AAD3B435B51404EE  
MARC:2FD076F9E0306FFEAAD3B435B51404EE  
OLIVER:617093781CC21A60AAD3B435B51404EE  
RICHARD:E4301A7CD8FDD1ECAAD3B435B51404EE  
WYNAND:D851BE004D8658DFAAD3B435B51404EE
```

IBM Networks Password Synchronization Tool

The IBM Networks Password Synchronization Tool facilitates the synchronization of passwords between OS/2 and Windows machines. This is a command line tool and does not add much overhead to the system. You can find it as a part of the supplemental files for this redbook.

This tool can be used on both Windows NT as well as Windows 2000 Servers. This is not meant for workstations. At least Service Pack 6 should be installed on Windows NT and Service Pack 3 should be installed on Windows 2000. Administrator privileges are required to run this tool.

To use this tool in our scenario, only the second step is necessary because the export of the password hashes is done in the same way as LSMT. The password file created is identical.

The command line syntax for importing the password hashed with **passwdsync** is:

```
passwdsync -i <input file> [-v] [ -l [log file] ]
```

where:

- i Specifies the <input file> to be used. In our scenario this file is created by LSMT.
- v Enables verbose mode. The relevant output is echoed to the screen.

- l Enables logging. The relevant output is logged to a file. If no log file is specified the default is PasswdSync.log.

Verbose and Logging mode are disabled by default.

For the migration example, we executed the following the command on the domain controller using the exported password hash file from the previous chapter:

```
passwdsync -i getpwd.log -v -l passwdsync.log
```

You can see the result in the log file.

Example 4-19 Password synchronization log passwdsync.log

```
14:52:48.515: 2744.2572> Trace: The currently running OS is Win2K
14:52:48.525: 2744.2572> Trace:
*****
14:52:48.525: 2744.2572> Trace: ENTER: UserHasAdminPrivilege
14:52:48.525: 2744.2572> Trace:
14:52:48.525: 2744.2572> Trace:
14:52:48.525: 2744.2572> Trace: EXIT: UserHasAdminPrivilege
14:52:48.525: 2744.2572> Trace:
*****

14:52:48.525: 2744.2572> Trace: buf is ANDREI:CD017457761C8B05AAD3B435B51404EE

14:52:48.525: 2744.2572> Trace: Userid is ANDREI
14:52:48.525: 2744.2572> Trace: Password is CD017457761C8B05AAD3B435B51404EE
14:52:48.686: 2744.2572> Trace: Password change for User ANDREI returned 0
14:52:48.686: 2744.2572> Trace: buf is LEIF:32DD5DAB4DC507A4AAD3B435B51404EE

14:52:48.686: 2744.2572> Trace: Userid is LEIF
14:52:48.686: 2744.2572> Trace: Password is 32DD5DAB4DC507A4AAD3B435B51404EE
14:52:48.816: 2744.2572> Trace: Password change for User LEIF returned 0
14:52:48.826: 2744.2572> Trace: buf is MARC:2FD076F9E0306FFEAD3B435B51404EE

14:52:48.826: 2744.2572> Trace: Userid is MARC
14:52:48.826: 2744.2572> Trace: Password is 2FD076F9E0306FFEAD3B435B51404EE
14:52:48.956: 2744.2572> Trace: Password change for User MARC returned 0
14:52:48.956: 2744.2572> Trace: buf is OLIVER:617093781CC21A60AAD3B435B51404EE

14:52:48.956: 2744.2572> Trace: Userid is OLIVER
14:52:48.956: 2744.2572> Trace: Password is 617093781CC21A60AAD3B435B51404EE
14:52:49.076: 2744.2572> Trace: Password change for User OLIVER returned 0
14:52:49.076: 2744.2572> Trace: buf is RICHARD:E4301A7CD8FDD1ECAAD3B435B51404EE

14:52:49.076: 2744.2572> Trace: Userid is RICHARD
14:52:49.076: 2744.2572> Trace: Password is E4301A7CD8FDD1ECAAD3B435B51404EE
14:52:49.196: 2744.2572> Trace: Password change for User RICHARD returned 0
```

```
14:52:49.196: 2744.2572> Trace: buf is WYNAND:D851BE004D8658DFAAD3B435B51404EE
14:52:49.196: 2744.2572> Trace: Userid is WYNAND
14:52:49.206: 2744.2572> Trace: Password is D851BE004D8658DFAAD3B435B51404EE
14:52:49.317: 2744.2572> Trace: Password change for User WYNAND returned 0
14:52:49.327: 2744.2572> Error: File read returned an error
```

Important: Because this transition is not supported, running evaluation tests before starting the migration is highly recommended to ensure that all of the applications are able to use the transferred password hash against Windows 2000 Domains. Also, setting the password to expired is recommended, so that users will be prompted to change their passwords right after the migration using a supported API.

Tip: Microsoft published an Knowledge Base article about the problem of having LAN Manager hashed passwords in the Active Directory. You can find this article published under the title *How to prevent Windows from Storing a LAN Manager Hash of your Password in Active Directory and Local SAM Databases*, Q299656.

4.5.5 Logon assignments

Windows 2000 does not support the domain database approach IBM provides with its LAN server family. Instead of this, Microsoft recommends that customers change their policy and use UNC names to add the Distributed File System services in the domain. It is required to have a consistent naming space and fault tolerance through replica sets within the domain.

Tip: There are several related documents published on the Microsoft Web site. A good start may be the following URL for a step-by-step guide to Distributed File System (DFS):

<http://www.microsoft.com/technet/prodtechnol/windows2000serv/howto/dfsguide.asp>

The DFS file system is not available for OS/2 and for this reason not applicable in this migration. Since the creation of Windows NT Server domains, customer, and third-party developers developed a number of solutions for this problem:

- ▶ Customizing the Client Desktop using persistent mapping of network drives
- ▶ Define logon scripts for each user maintaining these with interfaces or editors
- ▶ Develop applications based on utilities like Kixstart and Visual Scripting Host
- ▶ Replace the logon client with a proprietary product that supplies this feature

Most of these tools do not have a solution for logging on OS/2, Windows, and LINUX clients simultaneous. To keep things simple, the technology of a product called Logon Script Manager from 6PAC Consulting was used to migrate the logon assignments of users into simple text files supporting Windows and OS2 client logon. Please refer to 8.3.1, “Logon Script Manager offering” on page 302 to get more information about this product.

Logon script (logon.cmd)

The general idea of this approach is to create a platform independent logon script to enable OS/2 and Windows clients to run the same script. This is possible because both systems support the same basic command language using the file extension CMD. At logon time, the client processes the file specified in the user account as the logon script. This script is searched for in a share that every domain controller provides, named NETLOGON. For developing a universal script supporting OS/2 and Windows clients, the following issues must be covered:

- ▶ Determine the client’s operating system to allow operating specific commands in a conditional section of the logon script
- ▶ Set OS/2 environment variables equivalent to those available in Windows environments (for example, %LOGONSERVER%, %USERNAME%)
- ▶ Use a basic common command set in the script, deferring all special tasks in delimited sections or external scripts

The result of this approach is shown in the following examples. It consists of the main logon script logon.cmd (Example 4-20), the script to detect the client operating system checkos.cmd (Example 4-21), and a script used for OS/2 clients to add helpful environment variables os2env.cmd (Example 4-22).

When a client logs on, the following steps are processed:

1. The subroutine (checkos.cmd) for detecting the operating system is called:
 - a. Use the variable %SIXPAC.OS% for the result, and set it to the initial value to unknown (“UNK”).
 - b. Check if there is an environment variable %OS% available. In this case, this is a Windows operating system, otherwise go to step h.
 - c. Use the version command to retrieve the OS version.
 - d. If the version equals 5.1 it is Windows XP. Set the variable to the value “W2k” (Since WindowsXP and Windows2000 can be treated the same, no distinction was made here).
 - e. If the version equals 5.0 it is Windows 2000, set the variable to the value “W2k”.

- f. If the version equals 4.0 it is Windows NT Version 4, set the variable to the value "NT4".
 - g. Otherwise, exit the script, because it cannot detect a supported Windows version.
 - h. If the version information contains the string "System/2" assume an OS/2 operating system and set the variable to the value "OS2".
2. If the client uses OS2, call the REXX script `os2env.cmd` passing the path and file name of the global script (this is held in the environment variable %0):
 - a. Use the path name to extract the logon server and set it to the variable %LOGONSERVER%. In the example, the complete file name of the global logon script is `\\windc\netlogon\logon.cmd`, which implicitly gives you the name of the logon server.
 - b. Use the command `NET CONFIG REQ` to fetch additional variables.
 - c. Get the value for "MACHINE ID" and set the variable %COMPUTERNAME%
 - d. Get the value for "USER ID" and set the variable %USERNAME%
 - e. Get the value for "LOGON DOMAIN" and set the variable %USERDOMAIN%
3. As an example for global commands, synchronize the time using the command `net time`
4. Call the user specific logon script in the subdirectory USERS using the %USERNAME% variable to define its file name.
5. Jump to an operating system specific routine with the command `goto %SIXPAC.OS%`. For this there are the following labels defined:

W2k:	Execute everything necessary for Windows XP and Windows 2000 clients.
NT4	Execute or call commands necessary for Windows NT Version 4 clients.
OS2:	Execute or call steps necessary for OS/2 clients.
UNK:	Perform steps necessary if the operating system cannot be detected.
6. Finish processing the logon script.

Tip: If the environment contains Windows9x clients that do not process CMD batch files, you might consider modifying the approach as follows:

1. Set the logon script attribute to logon instead of logon.cmd
In this case, the client has to add the extension that is appropriate.
2. Create a logon.bat for Windows 9x, Windows NT, and Windows 2000 clients.

Create a logon.cmd for OS/2 clients.

Example 4-20 Excerpt of global logon script LOGON.CMD

```
@ECHO OFF
ECHO Please wait while logon script is executed...

CALL %0\..\CHECKOS.CMD

IF "%SIXPAC.OS%"=="OS2" CALL %0\..\OS2\OS2ENV.CMD %0

NET TIME %LOGONSERVR% /SET /Y 1>NUL 2>NUL

IF NOT "%SIXPAC.OS%"=="OS2" NET USE /persistent:no >NUL
IF EXIST %0\..\USERS\%USERNAME%.CMD CALL %0\..\USERS\%USERNAME%.CMD

GOTO %SIXPAC.OS%
GOTO END

:w2k
    ECHO Windows 2000 or Windows XP detected...
GOTO END

:NT4
    ECHO Windows NT 4.0 detected...
GOTO END

:OS2
    ECHO IBM OS/2 detected...
GOTO END

:UNK
    ECHO.
    ECHO Cannot detect operating system. Please call your local support.
    ECHO.
    PAUSE
GOTO END
```

:END

Example 4-21 Excerpt of script to detect the client operating system CHECKOS.CMD

```
@ECHO OFF

SET SIXPAC.OS=UNK
IF %OS%==_ GOTO NO_WINDOWS
VER | FIND /i "5.1" >NUL
IF %ERRORLEVEL%==1 GOTO NOT_XP
SET SIXPAC.OS=W2K
GOTO END_OSCHECK
:NO_XP
VER | FIND /i "5.0" >NUL
IF %ERRORLEVEL%==1 GOTO NOT_W2K
SET SIXPAC.OS=W2K
GOTO END_OSCHECK
:NO_W2K
VER | FIND /i "4.0" >NUL
IF %ERRORLEVEL%==1 GOTO END_OSCHECK
SET SIXPAC.OS=NT4
GOTO END_OSCHECK
:NO_NT4
:NO_WINDOWS
VER | FIND /i "System/2" >NUL
IF ERRORLEVEL==1 GOTO END_OSCHECK
SET SIXPAC.OS=OS2
:NO
:END_OSCHECK
```

Example 4-22 Excerpt of REXX script for adding environment variables OS2ENV.CMD

```
'@ECHO OFF'
PARSE UPPER ARG "\\ " LogonServer "\ "
CALL VALUE 'LOGONSERVER', "\\ " || LogonServer, 'OS2ENVIRONMENT'
'NET CONFIG REQ | RXQUEUE'
DO QUEUED()
  PARSE UPPER PULL line
  IF POS('MACHINE ID',line)>0 THEN CALL VALUE 'COMPUTERNAME',
SUBSTR(WORD(line,3),3), 'OS2ENVIRONMENT'
  IF POS('USER ID',line)>0 THEN CALL VALUE 'USERNAME', WORD(line,3),
'OS2ENVIRONMENT'
  IF POS('LOGON DOMAIN',line)>0 THEN CALL VALUE 'USERDOMAIN', WORD(line,3),
'OS2ENVIRONMENT'
END
```

User specific logon script (users\

With the general logon script described above, a user still does not get his logon assignments. For that reason a subdirectory containing all user specific logon scripts is defined. In this simplified environment, these scripts only assign some network drives and printers. This script is called during logon from the main logon script **Logon.cmd**, which is discussed in the previous section. A sample is shown in Example 4-23.

Example 4-23 Example user specific logon script LEIF.CMD

```
@ECHO OFF
REM *****
REM File      : LEIF.CMD
REM Version   : 2.0
REM Date      : 29 Jun 2003
REM Author    : Leif Braeuer (6PAC Consulting AG)
REM
REM Description:
REM User specific logon script of logon assignments
REM
REM *****

ECHO Assigning network connections...

:START_FILENETUSE
  NET USE L: \\BDC\LANSHARE
  NET USE U: \\PDC\LEIF
:END_FILENETUSE

:START_PRINTNETUSE
:END_PRINTNETUSE
```

The complete source code of these scripts can be found in Appendix , “Migrating users” on page 454.

Restriction: Because of the missing support for sharing serial devices in Windows 2000 Servers, the migration of these assignments is excluded.

Note: The logon assignments were sorted by type to make it easier to add some third-party tools or scripts that supply a convenient way to process the assignments. You may consider writing some programs or scripts using a GUI to alert users if there are errors while processing the assignments.

For the transition of the assignments for all users of the OS/2 domain, a REXX script was developed using the LSRXUTIL.DLL to enumerate all users, retrieve existing logon assignments, and use these to generate distinct logon scripts. The following command generates the files for the migration scenario and Example 4-24 lists the REXX function that generates a user logon script. The complete REXX script setuserasn.cmd can be found in Appendix , "SETWINUSERASN.CMD" on page 468.

```
setwinuserasn.cmd \\windc\netlogon\users
```

The script performs the following operations:

1. Enumerates the users in the domain
2. Checks for each user if there are logon assignments available
3. Creates a new file for this user named %USERNAME%.cmd
4. Writes some static lines like version, comment in this file
5. Iterates the assignments for file aliases:
 - a. For each, translate the alias name into a UNC path, and add an appropriate line specifying the net use command:


```
NET USE L: \\BDC\LANSHARE
```
6. Adds the home directory share to the logon script
7. Iterates the assignments for printer aliases:
 - a. For each, translate the alias name into a UNC path, and add an appropriate line specifying the net use command:


```
NET USE LPT1: \\BDC\PRINTQ
```
8. Add some trailing lines.

Example 4-24 Excerpt of REXX script setwinuserasn.cmd for logon assignments

```
/* -----*/
GenerateBatch:

NETUSER      = 280
myRc = NetGetInfo(NETUSER, 'userInfo', '', UserId)

NETLOGONASN = 52
myRc = NetGetInfo(NETLOGONASN, 'logonAsnInfo', '', UserId)

if myRc=0 then do
  CmdFile= TargetPath || '\|||UserId|||.CMD'
  'DEL ' || CmdFile || ' 1>NUL 2>NUL'
  CALL LineOut CmdFile, "@ECHO OFF"
```

```

CALL LineOut CmdFile, "REM
*****"
CALL LineOut CmdFile, "REM File   : " || userId || ".CMD"
CALL LineOut CmdFile, "REM Version : 2.0"
CALL LineOut CmdFile, "REM Date   : " || Date()
CALL LineOut CmdFile, "REM Author  : Leif Braeuer (6PAC Consulting AG)"
CALL LineOut CmdFile, "REM"
CALL LineOut CmdFile, "REM Description:"
CALL LineOut CmdFile, "REM User specific logon script of logon assignments"
CALL LineOut CmdFile, "REM"
CALL LineOut CmdFile, "REM
*****"
CALL LineOut CmdFile, ""
CALL LineOut CmdFile, ":START_FILENETUSE"
/* Get the user logon assignments information */
DO i=1 TO logonAsnInfo.count
  IF logonAsnInfo.i.type="Files alias" THEN DO
    CALL Lineout cmdFile, " NET USE " || logonAsnInfo.i.device || ": " ||
Alias2UNC()
    END
  END
  call Lineout CmdFile, " NET USE " || LEFT(userInfo.HOME_DIR,2) || " \\" ||
WORD(TRANSLATE(userInfo.HOME_DIR, " ", "\"),2) || "\" || userId

  CALL LineOut CmdFile, ":END_FILENETUSE"
  CALL LineOut CmdFile, ""
  CALL LineOut CmdFile, ":START_PRINTNETUSE"
  DO i=1 TO logonAsnInfo.count
    IF logonAsnInfo.i.type="Printer alias" THEN DO
      CALL Lineout cmdFile, " NET USE " || logonAsnInfo.i.device || ": " ||
Alias2UNC()
      END
    END
    CALL LineOut CmdFile, ":END_PRINTNETUSE"
    CALL LineOut CmdFile, ""
    Rc = Stream(CmdFile, 'c', 'close')
  end
RETURN

/* -----*/
Alias2UNC:
  NETALIAS = 20
  MyRc = NetGetInfo(NETALIAS, 'AliasInfo', '', logonAsnInfo.i.alias)
RETURN "\\" || aliasInfo.server || "\" || aliasInfo.netname

```

4.5.6 Steps to follow

To perform the migration of user definitions from OS/2 to Windows 2000 Active Directory, follow these steps on the domain controller:

1. Create the export file `getusers.log` using the LSMT as described in 3.3.4, "Users" on page 74.
2. Modify the entries and add an A in the column OPT for the users you want to transfer to the target domain.
3. Change descriptions, names, privilege, or other attributes as you need them in Windows 2000 for you branch.
4. Run the command `setusers.cmd` with the following parameters:

```
setusers win getusers.log users.ldif Branch1
```
5. Import the user definitions to Active Directory with the following command:

```
ldifde -v -i -f users.ldif
```
6. Save the log files `ldif.err` and `ldif.log` of this step.
At this step your basic user objects are migrated to the target domain without any group membership, password, or logon assignments.
7. Get the export file `getgrps2.log` using the LSMT as described in 3.3.3, "Groups" on page 72.
8. Modify the entries and add an A in the column OPT for the users' group memberships you want to transfer to the target domain.
9. Change memberships or add new groups as you need them in Windows 2000 for your branch.
10. Create the `group-db.csv` database that the scripts need to translate OS/2 group names to Windows 2000 LDAP names.
11. Run the command `setgrpmem.cmd` with the following parameters:

```
setgrpmem win getgrps2.log grpmem.ldif Branch1
```
12. Import the user definitions to Active Directory with the following command:

```
ldifde -v -i -f grpmem.ldif
```
13. Save the log files `ldif.err` and `ldif.log` of this step.
14. Create the LSMT export file `getpwd.log` containing the password hashes.
15. Remove all lines for users that you do not want to migrate.
16. *Do not modify any password hash!*
17. Run the following command:

```
passwdsync -i getpwd.log -v -l passwdsync.log
```

18. Save the log file `passwdsync.log` of the step.

At this milestone users can already logon to the new domain, but will miss their logon assignments.

19. Create the basic directory structure within the netlogon share to provide logon scripts:

```
md \\windc\netlogon\NT4
md \\windc\netlogon\OS2
md \\windc\netlogon\Users
md \\windc\netlogon\W2k
```

20. Copy the file `logon.cmd` and the file for checking the client OS into the netlogon share:

```
copy logon.cmd \\windc\netlogon
copy checkos.cmd \\windc\netlogon
```

21. Run the migration script for the logon assignments from an OS/2 domain Controller:

```
setwinuserasn.cmd \\windc\netlogon\users
```

4.6 Migrating directories

After the basic user and group accounts are migrated, it is possible to start transferring the resources of the OS/2 domain to Windows 2000.

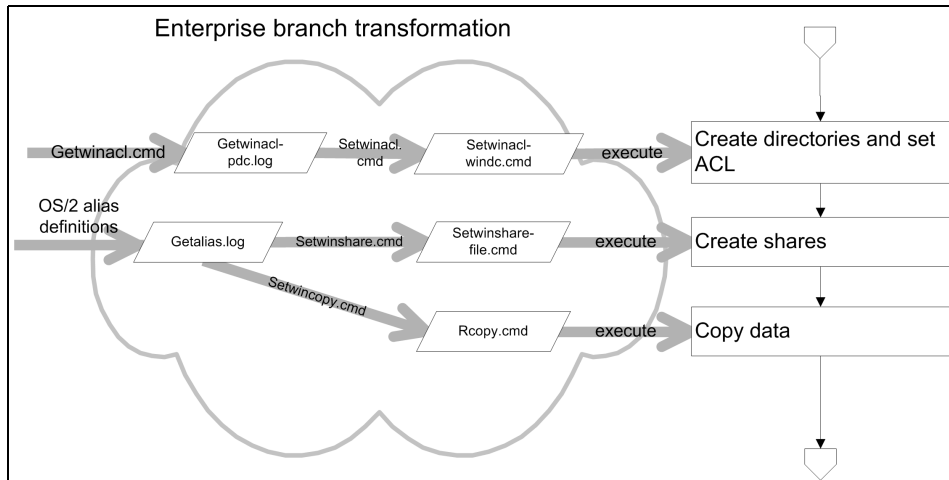


Figure 4-8 Migration workflow for directory part

This section focuses on the migration of the file resources. The printer and serial resources are discussed in 4.7, "Migrating printers" on page 154, and 4.8, "Migrating serial devices" on page 158.

The process of migrating data from OS/2 Servers to Windows 2000 consists of the following steps:

1. Create the empty directories and define the ACL for this directory.
2. Define shares for each directory.
3. Copy the content from the OS/2 UNC name to the Windows 2000 server.

These steps are described in the following sections.

4.6.1 Migrating access control

IBM LAN Server and Windows 2000 use an access control concept on directories and files. Although Windows 2000 extends the features, the basics are still the same. For each resource an Access Control List (ACL) is defined, which consists of auditing flags and a number of Access Control Entries (ACE). Each ACE consists of the name of a user or a group and the granted access rights. In the following is an example of a command that shows this information:

```
[C:\]net access e:\lanshare
```

Resource	Permissions	Permissions

e:\lanshare	*TRANSITION:RWCXDAP	LEIF:RWX

The command completed successfully.

Although each subdirectory is equipped with an ACL, only a few might vary from the top level directory ACLs. Instead of retrieving a database with all of these ACLs, only ACL changes are exported. That means if an ACL does not change in the whole subdirectory only this one entry is saved, because of the inheritance of ACLs for newly created directories. As an example, for the server containing the home directories for all users our script will save the ACL for the root directory instead of all subdirectories, because the subdirectories have the same ACL.

For this, a REXX script processes a directory and exports the ACL in a comma separated format. This script is provided here, instead of Chapter 3, "Starting the OS/2 Server migration" on page 63, because it only supports the Windows 2000 migration. The script can be found in Example 4-25 and performs the following actions:

1. Gets command line arguments for the export file name and the base directory starting the query
2. Retrieves the ACL for this base directory with the NetGetInfo function

3. Processes the returned ACL stem in a manner that the ACEs are sorted by group or user name and are displayed in the form <name>:<ACE>. The function FormatACL() in the script performs this formatting. An example looks like the following:

```
LEIF:RWX;TRANSITION:RWCXDAP
```

4. Writes the ACL to the output file adding the complete path name:

```
E:\LANSHARE;LEIF:RWX;TRANSITION:RWCXDAP
```

5. Calls the function RecurseDir to process all subdirectories. For this function, the base path and its ACL is passed. This function processes the following steps:
 - a. Call SysFileTree to enumerate all subdirectories within the base directory.
 - b. Get the ACL for each using the NetGetInfo function.
 - c. Format the ACL as described in step 3.
 - d. Compare the ACL with the one passed (that of the parent directory).
 - e. If there is any difference write it to the export file.
 - f. Recursively call RecurseDir again with the directory name and its ACL.
 - g. Continue with the next directory.

Example 4-25 Script to export ACL for a directory tree (getwinacl.cmd)

```
/* Get a access control profile for a drive tree */

call RxFuncAdd 'LoadLsRxutFuncs', 'LSRXUT', 'LoadLsRxutFuncs'
call LoadLsRxutFuncs
call RxFuncAdd 'SysLoadFuncs', 'REXXUTIL', 'SysLoadFuncs'
call SysLoadFuncs

Parse Arg outFile basePath

basePath = Strip(basePath)
outFile = Strip(outFile)

'@del 'outfile' 1>NUL 2>NUL'

if LENGTH(basePath)<3 Then basePath=basePath"\
rc = NetGetInfo( 10, 'AccPerm', '', basePath)
if rc <> 0 Then strAc1 = ""
else strAc1 = FormatACL()
call Lineout outFile, basePath || ";" || strAc1

Call RecurseDir basePath, strAc1

call DropLsRxutFuncs
call RxFuncDrop 'LoadLsRxutFuncs'
exit
```

```

RecurseDir: procedure expose outFile
  PARSE ARG baseDir, strACL
  Say baseDir
  baseDir = STRIP(baseDir,"T","\")
  CALL SysFileTree baseDir || '\*', 'dir.', 'DO'
  DO i = 1 TO dir.0
    rc = NetGetInfo( 10, 'AccPerm', '', dir.i)
    if rc <> 0 Then subAc1 = ""
    else subAc1 = FormatACL()
    if subAc1 <> strAc1 Then call Lineout outFile, dir.i || ";" || subAc1
    CALL RecurseDir dir.i, subAc1
  END
RETURN

```

```

FormatACL:
  ac1 = ""
  do fi=1 to AccPerm.count-1
    do fj=fj to AccPerm.count-1
      fk=fj+1
      if AccPerm.fj.ugname > AccPerm.fk.ugname then do
        tempVar = AccPerm.fk.ugname
        AccPerm.fk.ugname = AccPerm.fj.ugname
        AccPerm.fj.ugname = tempvar
        tempVar = AccPerm.fk.access
        AccPerm.fk.access = AccPerm.fj.access
        AccPerm.fj.access = tempvar
      end
    end
  end
  do k=1 to AccPerm.count
    ac1 = ac1 || AccPerm.k.ugname || ":" || AccPerm.k.access || ";"
  end
return ac1

```

Execute this command on each member server sharing file resources at each entry point of a share. To get all ACLs of the E:\ drive, use the following command:

```
getwinac1 getwinac1-bdc.log E:\lanhomes
```

Attention: Unlike the scripts described before, this appends new records to the existing file. Be aware of duplicate entries when executing the script more than once.

Restriction: In the above example, there is no support of the migration of ACLs at the file level, device level, or auditing flags. The script is meant as a template and can be expanded to support these features.

In this example, the ACLs of all home directories are exported. It can be seen that a special group OPERATING has access to all home directories (to restore data or create new users) and each home directory has explicit permissions for the appropriate user. As the ACL does not change for subdirectories within the user's home directory, all directories a user may have are not listed here:

Example 4-26 Example output file of getwinacl.cmd (getwinacl-bdc.log)

```
e:\LANHOMES;OPERATING:RWCXDAG;  
e:\LANHOMES\ANDREI;ANDREI:RWCXDAP;OPERATING:RWCXDAG;  
e:\LANHOMES\LEIF;LEIF:RWCXDAP;OPERATING:RWCXDAG;  
e:\LANHOMES\MARC;MARC:RWCXDAP;OPERATING:RWCXDAG;  
e:\LANHOMES\OLIVER;OLIVER:RWCXDAP;OPERATING:RWCXDAG;  
e:\LANHOMES\RICHARD;OPERATING:RWCXDAG;RICHARD:RWCXDAP;  
e:\LANHOMES\WYNAND;OPERATING:RWCXDAG;WYNAND:RWCXDAP;
```

Having the extracted information of the OS/2 domain, it needs to be transferred to a Windows 2000 readable format. As described for the other migration steps, the data may be changed before import.

For the target platform, the utility **cacl**s that is part of Windows 2000 is used. It provides a similar feature set like the **NET ACCESS** command and is sufficient for our example. This program was introduced in “CACLS” on page 98.

As the format of an ACL is different in the Windows 2000 environment, and the **cacl**s command does not support input files, there is a transition step necessary which we perform with REXX:

1. Parse the two command parameters specifying the name of the import and output files.
1. Read the import file.
2. For each line transfer the ACE in a format the Windows 2000 command does understand (FormatACL):
 - a. If the granted group is ADMINS, change it to Domain Admins
 - b. If the granted group is USERS, change it to Domain Users
 - c. If the granted group is GUESTS, change it to Domain Guests
 - d. Add the environment variable %USERDOMAIN% to what domain the member server uses for authentication.

e. Translate the ACE from OS/2 to Windows 2000:

RWCXDAP to F

RX to R

RWCXDA to C

3. Add static ACEs that each directory should have (using the variable `defaultAcl`).
4. Compose two commands that the target server should execute.
 - a. Create the target directory (e.g. `md e:\lanhomes`).
 - b. Create a `cacls` command using the following syntax:

```
cacls <directory> /g <list-of-ace>
```
 - c. Prefix "echo y|" because `cacls` otherwise waits for the confirmation of the command. Do not add spaces around the `y` character, because this changes the answer for `cacls`.
5. Write these two commands into the output file.

The script doing this is named `setacl.cmd` and is executed as follows:

```
setwinacl getwinacl-bdc.log setwinacl-bdc.cmd
```

Example 4-27 Transition script for ACL (setwinacl.cmd)

```
/* */
Parse Arg inFile outFile
defaultAcl = "Administrators:F SYSTEM:F"

inFile = Strip(inFile)
outFile = Strip(outFile)

'@del 'outFile' 1>NUL 2>NUL'

Do While Lines(inFile)
  curLine = LineIN(inFile)
  if curLine = '' | Left(Strip(Opt),1) = '*' Then Iterate
  else do
    Parse value curLine With strPath ';' curLine
    i = 0
    strAcl = defaultAcl || " "
    Do While curLine <> ''
      i = i + 1
      Parse value curLine With actValue ';' curLine
      strAcl = strAcl || FormatNTAcl( actValue )
    End
    CALL LineOut outFile, "md " || strPath
    CALL LineOut outFile, "echo y|cacls " || strPath || " /g " || strAcl
```

```

        End
    End
    Exit
Return

/* -----*/
FormatNTAcl:
    PARSE ARG userid:"ace
    ace = Strip(ace,"T","G")
    select
        when userid = "USERS" Then userid = "Domain Users"
        when userid = "ADMINS" Then userid = "Domain Admins"
        when userid = "GUESTS" Then userid = "Domain Guests"
        otherwise nop
    end
    select
        when ace = "RWCXDAP" Then ace = "F"
        when ace = "R" Then ace = "R"
        when ace = "RX" Then ace = "R"
        when ace = "RWCXDA" Then ace = "C"
        otherwise nop
    end
    ace = '%"USERDOMAIN%\ ' || userId || ':' || ace || ' '
Return ace

```

After this has completed, a command file can run on the Windows 2000 Server to create the empty directory structure to receive data. Example 4-28 shows the result using the described import file. Execute the following command on the target file server (part of the file name is the name of the target server to identify which file is meant for which server):

setwinac1-bdc.cmd

Example 4-28 Windows 2000 formatted ACL script setwinac1-bdc.cmd

```

md e:\LANHOMES
echo y|cac1s e:\LANHOMES /g Administrators:F SYSTEM:F
"%USERDOMAIN%\OPERATING:C"
md e:\LANHOMES\ANDREI
echo y|cac1s e:\LANHOMES\ANDREI /g Administrators:F SYSTEM:F
"%USERDOMAIN%\ANDREI:F" "%USERDOMAIN%\OPERATING:C"
md e:\LANHOMES\LEIF
echo y|cac1s e:\LANHOMES\LEIF /g Administrators:F SYSTEM:F
"%USERDOMAIN%\LEIF:F" "%USERDOMAIN%\OPERATING:C"
md e:\LANHOMES\MARC
echo y|cac1s e:\LANHOMES\MARC /g Administrators:F SYSTEM:F
"%USERDOMAIN%\MARC:F" "%USERDOMAIN%\OPERATING:C"
md e:\LANHOMES\OLIVER

```

```
echo y|cac1s e:\LANHOMES\OLIVER /g Administrators:F SYSTEM:F
"%USERDOMAIN%\OLIVER:F" "%USERDOMAIN%\OPERATING:C"
md e:\LANHOMES\RICHARD
echo y|cac1s e:\LANHOMES\RICHARD /g Administrators:F SYSTEM:F
"%USERDOMAIN%\OPERATING:C" "%USERDOMAIN%\RICHARD:F"
md e:\LANHOMES\WYNAND
echo y|cac1s e:\LANHOMES\WYNAND /g Administrators:F SYSTEM:F
"%USERDOMAIN%\OPERATING:C" "%USERDOMAIN%\WYNAND:F"
```

4.6.2 Migrating share definitions

The primary data used for migrating the file shares is the alias database of the OS/2 Warp Server. How the LSMT tools retrieve this database and the format of the export file can be found in Section 3.3.6, "File and printer shares" on page 78.

Aliases are not known in the Windows 2000 environment. Instead of this, all shares you define are stored in the server's registry and shared at startup time. Therefore, only the following attributes for file shares are used during the migration:

- ▶ Remark (the optional comment for the share)
- ▶ Server (the name of the server providing the resource)
- ▶ Netname (instead of the alias it is better to use the share name, because they may differ)
- ▶ Maxuses (if the value is not 65535 the usage is limited)
- ▶ Type (only lines with the value "Files" are transferred as file shares)
- ▶ Path

All other values are not needed or specify parameters for serial or printer shares. As in the previous cases, you may need to modify server names, the path, or netnames.

Using these file you can provide all necessary values for the **rmtshare** command to create a share using the following command:

```
rmtshare \\<server>\<netname>=<path> /remark:"<remark>" /users:<maxuses>
```

The script **setwinshares.cmd** in the Example 4-29 (completely listed in Appendix , "SETWINSHARE.CMD" on page 473) does this transition automatically for all shares of the OS/2 domain. Because the program logic and the utility used are the same, the printer migration is also done in this step. For that reason the name of an output file for printer definitions must be specified as well. This will be discussed in more depth in 4.7, "Migrating printers" on page 154. Start the migration by calling the migration script passing the name of

the LSMT definition file and two output files, one for the file share the other for printers:

```
setwinshare getalias.log setwinshare-file.cmd setwinshare-print.cmd
```

As outlined in 4.8, "Migrating serial devices" on page 158, there is no direct migration path for serial devices. For that reason these definition are omitted with an error message:

```
COMM_Q skipped. Target does not support type serial
```

Example 4-29 Exporting aliases to Windows 2000 with setwinshares.cmd

```
/* -----*/
CreateCmd:
select
  when share.TYPE = 'Files' Then Do
    optional = ""
    if share.REMARK <> "" Then optional = optional || "/remark:" ||
share.REMARK || " "
    if share.MAXUSES <> 65535 Then optional = optional || "/users:" ||
share.MAXUSES || " "
    CALL LineOut outFileDir, "rmtshare \" || share.SERVER || "\" ||
share.NETNAME || "=" || share.PATH || optional
    end
  when share.TYPE = 'Printer' Then Do
    if share.REMARK <> "" Then optional = optional || '/remark:" ||
share.REMARK || "' ' '
    if share.MAXUSES <> 65535 Then optional = optional || "/users:" ||
share.MAXUSES || " "
    CALL LineOut outFilePrt, "rmtshare \" || share.SERVER || "\" ||
share.NETNAME || "=" || share.QUEUE || " " || optional
    end
  otherwise SAY share.NAME || ' skipped. Target does not support type ' ||
share.TYPE
end
Return
```

Example 4-30 Excerpt of alias definitions from LSMT (getalias.log)

OPT;NAME	;REMARK	;SERVER	;NETNAME	;MAXUSES	;TYPE	;PATH
A ;BOOK	;	;PDC	;BOOK	;65535	;Files	
;F:\BOOK						
A ;LANSHARE;		;BDC	;LANSHARE;	65535	;Files	
;E:\LANSHARE						

After this completes, import these definitions for file share with the command **setwinshare-file.cmd**

4.6.3 Migrating the data

The next step of migrating the file resource consists of copying the user data. Because both platforms support the same SMB protocol, the migration can be done directly. Several options are available to get the data from OS/2 to Windows 2000. Amongst them are:

- ▶ XCOPY on OS/2 or Windows 2000. Both products offer similar functionality, the OS/2 version cannot be used with UNC-names. For that reason automatic file migration becomes more difficult than using Windows 2000 or other tools.
- ▶ Backup and restore programs such as Tivoli Storage Manager. It may be a good idea to prepare the server that way since due to bandwidth limitations you can prepare the server off site.
- ▶ Robocopy is part of the Microsoft Resource Kit. We described it in “Robocopy” on page 96 and use it for the following reasons:
 - It has rich logging capabilities.
 - It allows retries for open files.
 - Is not aware of HPFS extended attributes, and for that reason keeps the target partition free of them.
 - It allows mirroring. This means you can run the script multiple times, and after the first run, only changes are synchronized.

The script for performing this step is similar to the script developed in 4.6.2, “Migrating share definitions” on page 148, but **setwincopy.cmd** generates the file `rcopy.cmd` as output containing the list of robocopy statements for each alias. The problem in generating this file is that the script is not aware of the naming conventions to keep both servers running. At least one of the servers will have to have another name at the time the process is initiated. To enable this, “nicknames” were defined. Use the replace feature of an editor to adjust this. The source server name of each alias was changed to `OS.servername` the target name to `WIN.servername`. Generate the `rcopy.cmd` with the following command:

setwincopy getalias.log

Example 4-31 Script to generate to robocopy calls (setwincopy.cmd)

```
/* */
Parse Arg inFile
inFile = Strip(inFile)
outFileDir = "rcopy.cmd"

'@del 'outFileDir' 1>NUL 2>NUL'

Do While Lines(inFile)
    curLine = LineIN(inFile)
```



```

orgLine = curLine
Parse Value curLine With Opt ';' curLine
Select
  When Opt = '' | curLine = '' | Left(Strip(Opt),1) = '*' Then Iterate
  When Translate(Opt) = 'OPT' Then Call GetColumns
  When Translate(Opt) = 'A' Then Call AddShare
  Otherwise Iterate
End
End
Exit ExitCode
Return
/* -----*/
AddShare:
  i = 0
  Do While curLine <> ''
    i = i + 1
    columnName = Strip(columnNames.i)
    Parse value curLine With actValue ';' curLine
    share.columnName = Strip(actValue)
    If (share.columnName = "Unknown") | (share.columnName = "(null)") Then
share.columnName = ''
    End
  Call CreateCMD
Return
/* -----*/
GetColumns:
  i = 0
  Do While curLine <> ''
    i = i + 1
    Parse value curLine With columnNames.i ';' curLine
  End
  numColumn = i
Return
/* -----*/
CreateCmd:
  if share.TYPE = 'Files' Then Do
    CALL LineOut outFileDir, "robocopy \\OS2." || share.SERVER || "\" ||
share.NETNAME || " \\WIN." || share.SERVER || "\" || share.NETNAME || " /mir /z
/r:3 /w:30 /np /log+:rcopy.log"
  end
Return

```

The result, based on the input file in Example 4-30 on page 149 may look like this:

```

robocopy \\OS2.PDC\BOOK \\WIN.PDC\BOOK /mir /z /r:3 /w:30 /np /log+:rcopy.log
robocopy \\OS2.BDC\LANSHARE \\WIN.BDC\LANSHARE /mir /z /r:3 /w:30 /np
/log+:rcopy.log

```

Replace the string OS2.PDC with the name of the source OS/2 Server during the migration and WIN.PDC with the name of the target system. The same applies for the other server names. **Robocopy** is called with the following additional parameters:

/mir	Enable mirroring. Only synchronizes changes, add new files and delete removed files on the target system.
/z	Files are copied in a restartable mode.
/r:3 /w:30	If a file cannot be opened, robocopy at most makes three retries within 30 seconds and pauses until it continues with the next file.
/np	No progress in copying a file is written to the log file.
/log:rcopy.log	Append all messages to the log file rcopy.log.

Migrating the file resources of a server consists of calling the appropriate commands in the rcopy.cmd.

Restriction: Keep in mind that Windows 2000 does not support Extended Attributes (EA) within NTFS. So, during migration all data stored in EAs will be lost.

4.6.4 Migrating DASD limits

There is no direct migration path for OS/2 LAN Server DASD limits to Windows 2000. The LAN Server DASD limits are defined on a directory level. Limiting a directory means that the amount of data stored in this directory tree may not exceed the defined amount. This is defined independent of the owner of the file.

Windows 2000 on the other hand handles its quotas on a volume level. The amount of storage available for users may not exceed a certain value regardless of where it will be stored. Using Microsoft quota services sounds in some way only reasonable for home directories, where the owner of a file is mostly the owner of the directory. Because there is no direct mapping for the LAN Server DASD Limits, we only provide you some useful reading for this subject:

- ▶ *Microsoft Technet Best practices Disk Quotas*, found at:
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/entserver/sag_DQbest_practices.asp
- ▶ *Understanding Windows 2000 Disk Quotas*, found at:
<http://www.techsupportalert.com/pdf/t1729.pdf>

Additionally, there are some third party products available:

- ▶ “Precise/StorageCentral SRM” from Precise Software Solutions offers a comparable disk quota solution including several storage reporting capabilities and Active Directory integration. More information can be obtained at:
<http://www.precise.com/Products/StorageCentralSRM>
- ▶ “Quota and File Sentinel” from NTP Software. Product information is available at:
<http://www.ntpsoftware.com/products/qfs>

4.6.5 Steps to follow

In summary, the following steps are necessary to migrate all file resources from OS/2 to Windows 2000 file servers. The sample simplifies the steps and assumes that all required servers of the OS/2 and Windows 2000 environment are online. Adapt these steps to reflect a real migration workflow depending on the requirements:

1. Generate the getalias.log using the LSMT procedures.
2. Generate the access profiles using getwinacl.cmd on each server on each partition:


```
On the PDC: getwinacl getwinacl-pdc.log D:\
On the PDC: getwinacl getwinacl-dc.log E:\

On the BDC: getwinacl getwinacl-bdc.log D:\
On the BDC: getwinacl getwinacl-bdc.log E:\
```
3. Optionally, modify the path and access.
4. Generate the **cacls** command for both servers:


```
setwinacl getwinacl-pdc.log setwinacl-pdc.cmd
setwinacl getwinacl-bdc.log setwinacl-bdc.cmd
```
5. Execute the two scripts on the Domain Controller and on the member server.
6. Run the transition script for file and print aliases:


```
setwinshare getalias.log setwinshare-file.cmd setwinshare-print.cmd
```
7. Save the setwinshare-print.cmd for 4.7, “Migrating printers” on page 154
8. Run the command to generate all shares on all migrated servers within your domain:


```
setwinshare-file
```
9. Prepare for data migration. Remove obsolete backups.
10. Run the script to generate the data migration batch for **robocopy**:


```
setwincopy getalias.log
```

11. Edit the resulting `rcopy.cmd` and replace the source server name and the target server name with the ones you will use during migration. In our example, we will replace “OS2.PDC” with “PDC”, “WIN.PDC” with “WINDC”, “OS2.BDC” with BDC and “WIN.BDC” with “WINMEM”.
12. Run the script once during the days before migration to speed up the day of migration, where only the changes will be synchronized.
13. Complete your migration by running `rcopy.cmd`

4.7 Migrating printers

Migrating the printer resources is logically the next step.

Tip: Because each client type, be it OS/2, Windows 2000, or other brings a unique set of printing concerns that must be resolved for a migration to be complete and successful, it is recommended that the printing environment is evaluated and possibly modified prior to migration. It is recommended to consider the printing infrastructure as it was outlined in 1.5.4, “Printer migration” on page 13.

The process of migrating the printer resources consists of the following steps:

1. Define the operating system print queues.
2. Define the printer shares.

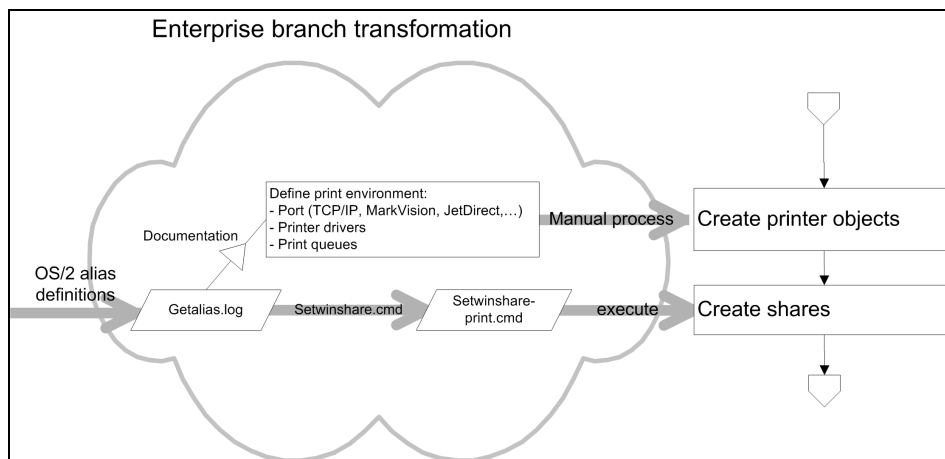


Figure 4-9 Migration workflow for printer migration

4.7.1 Client printing considerations

The general concept of printing services for clients is pretty much the same for OS/2 and Windows 2000. Both platforms support two different queue types to connect a device, and supply a queue the client can use for printing:

- Local printer** The printer is connected to a locally defined port. This port may be a parallel or serial interface (LPTx and COMx), a TCP/IP port (using LPD) or a third party product like Lexmark Markvision. As a rule, print queues on OS/2 or Windows 2000 print servers are defined as local printers.
- Network printer** Network printer directly maps to a local printer defined on a Server. Clients can share the printer driver providing a uniform environment.

There are a number of considerations for workstation printing configuration. An example of a configuration that will be problematic moving from OS/2 to Windows is the configuration and operation of network printers. OS/2 clients can get a list of all available printers in a Windows 2000 domain, but cannot use Active Directory. For that, they still can use both local and network printers. Unfortunately, the OS/2 operating system needs its own printer drivers that are not provided by Windows 2000 using the additional driver support. Within Windows this is only provided for the Windows operating system family. While OS/2 clients search for a share name PRINTDRV, Windows clients need the share PRINT\$ offering different directories for the distinct operating systems.

One way OS/2 clients can print to a Windows 2000 print server is by installing a local printer and redirecting the local port to the network printer share on the Windows 2000-based server. For example, if a local printer is installed on LPT3, and you type the following on the command line:

```
net use lpt3: \\winmem\printq
```

the output to the LPT3 port is redirected by the client network redirector. Additionally, applications can be configured to use UNC names to deliver print output to a print server share.

Tip: Defining a PRINTDRV share on the Windows 2000 print servers and populating the share with current printer driver resources will continue to effectively serve the OS/2 workstations for printer driver setups and updates.

4.7.2 Print queue options

With Windows 2000 there is no standard path to automatically migrate or create print queues. There are some third party tools and scripts available. Most of the necessary Windows Management Instrumentation API (WMI) are only available with Windows 2003.

Here is a list of Web resources that may help you in planning your print queue migration:

- ▶ Microsoft Technet Script Center found at:
<http://www.microsoft.com/technet/scriptcenter>
- ▶ Microsoft Technet “Print Server Upgrade, Migration, and Interoperability” found at:
<http://www.microsoft.com/technet/prodtechnol/windowsserver2003/plan/psumio.asp>

Tip: With Windows 2003 Server, Microsoft supplies features and scripts to manage your printing environment:

- ▶ prncnfg.vbs that configures or displays configuration information about a printer
- ▶ prndrvr.vbs that adds, deletes, and lists printer drivers
- ▶ prnmngr.vbs that adds, deletes, and lists printers or printer connections, in addition to setting and displaying the default printer
- ▶ prnport.vbs that creates, deletes, and lists standard TCP/IP printer ports, in addition to displaying and changing port configuration

Further information about these scripts can be found in the command line reference at:

<http://www.microsoft.com/technet/prodtechnol/winxpro/proddocs/ntcmds.asp>

There is no general solution other than a true network printing environment, so migrating printing must be done prior to this step.

4.7.3 Define print queue shares

Migration of printers is central to the function of the target system. The OS/2 print aliases have already been converted to Windows 2000 shares in 4.6.2, “Migrating share definitions” on page 148. A command file, called setwinshare-print.cmd is generated using the LSMT output file GETALIAS.LOG,

extracting all printer alias definitions to the target file. For further information about this process please refer to 4.6.2, “Migrating share definitions” on page 148. The definition of the share is done by the RMTSHARE command introduced in “RMTSHARE” on page 95, and used for the file share migration in 4.6.2, “Migrating share definitions” on page 148.

For the example migration, the following LSMT output file, GETALIAS.LOG, is used. Please be aware that the file contains wrapped lines.

Example 4-32 Example of the LSMT output for the aliases

```

OPT;NAME      ;TYPE   ;SERVER   ;PATH
;REMARK                                     ;LOCATION   ;MODE
;MAXUSES;QUEUE   ;PRIORITY;DEVICE_POOL ;
;BOOK      ;Files ;PDC       ;F:\BOOK
;
startup;65535 ;Unknown ;Unknown ;Unknown ;
;LANSHARE;Files ;BDC       ;E:\LANSHARE
;
startup;65535 ;Unknown ;Unknown ;Unknown ;
;Within Domain;At server
;Within Domain;At server

```

The following command, already used during the file share migration, produces the necessary printer share definitions:

```
setwinshare getalias.log setwinshare-file.cmd setwinshare-print.cmd
```

Example 4-33 Example setwinshare-print.cmd

```
rmtshare \\BDC\IBMNULLP=IBMNULLP /printer /remark:"Network Printer Queue"
```

This command file is to be executed on the target domain.

Restriction: In the sample environment, no ACLs were defined on printer shares. Therefore, it is not covered in the script. Nevertheless, RMTSHARE utility does support this feature, so it can be added to the script with minimal effort.

4.7.4 Steps to follow

The following steps are necessary to migrate the print resources from the OS/2 Servers to the Windows 2000 server:

1. Generate the getalias.log using the LSMT procedures.
2. Adapt the file to your real migration workflow.
3. Create all necessary print queues on your target servers.

4. Run the transition script for print aliases (as described in chapter 4.6.2, “Migrating share definitions” on page 148):

```
setwinshare getalias.log setwinshare-file.cmd setwinshare-print.cmd
```

5. Execute the **setwinshare-printer.cmd** command.

4.8 Migrating serial devices

OS/2 Warp Server services includes sharing of serial devices. Using that feature, administrators have been able to share bidirectional serial devices such as modems within the domain. Windows 2000 does not include a comparable feature. Some manufacturers, such as those listed below, provide a hardware-based solution connecting serial devices over TCP/IP:

- ▶ Equinox Super Serial Ethernet Serial Provider from Alloy Computer Products, found at:
<http://www.alloy.com.au>
- ▶ THINQ Serial Device Server from Quatech INC, found at:
<http://www.quatech.com>

There are drivers for Windows and LINUX available.

4.9 Migrating applications

There is no direct migration path for OS/2 LAN Server public applications to Windows 2000. The administrator can use the public applications to define a folder containing the applications a user should use. Microsoft tends to encourage a more user centric way by providing a FAT client with all software required by a user installed. In many cases, applications cannot even be installed on a shared drive. There are some third party products or concepts available, that fill this gap:

- ▶ Citrix Metaframe to enable support of Windows applications on the clients desktop. More information can be found at:
<http://www.citrix.com>
- ▶ NetApp suite from 6PAC Consulting, provides network applications within a folder. These tools provide different approaches to provide network defined applications for OS/2 and Windows clients, storing configuration in plain INI files or Active Directory. More information can be found at:
<http://www.6pac-ag.com>
- ▶ Servolution Logon Client from Comtarsia. By replacing the Windows 2000 logon interface, these clients can use features of an extended Active Directory

schema including network applications. More information can be found in 8.5, “Servolution” on page 345.

4.10 NFS migration

The Network File System (NFS) was developed to allow machines to mount a disk partition on a remote machine as if it were on a local hard drive. This allows for fast, seamless sharing of files across a network.

Advantages of NFS today include that it is a mature standard, well understood, and supported robustly across a variety of platforms.

On the OS/2 platform, NFS is used to share data between different OS platforms, especially UNIX. In the following sections, we describe a way to move or translate the NFS server configuration from OS/2 to Windows.

The NFS server that comes along with Warp Server for e-business is a selectable feature during the installation of the TCP/IP stack, was completely rewritten in 32-bit code, and its implementation is very close to the NFS servers implemented on UNIX platforms.

4.10.1 Software requirement

In order to migrate the configuration, the following requirements apply to the OS/2 Server:

- ▶ The OS/2 Server is up and running
- ▶ The OS/2 NFS server is installed
- ▶ The NFS server is configured properly and running

For the Windows server, the following requirement applies:

- ▶ The Windows server is up and running

4.10.2 Source platform configuration

The **exports** file on OS/2 is very similar to that used in UNIX environments. It is located in the ETC Directory.

Example 4-34 OS/2 NFS configuration file

```
f:\nfs -alias nfs -rw # NFS on PDC
f:\nfs
```

In this example, the Directory F:\NFS is exported as an alias named `nfs` and there is read-write access for everybody. Everything after the pound sign `#` is

treated as a comment. The NFS user access is configured in the general TCP/IP Configuration TCPNBK.LST. In this general configuration file, you can see on a “per user basis,” if NFS access is enabled.

4.10.3 Migration scenario

The migration scenario is:

1. Create the users on Windows, if they are not created yet.
2. Install and configure the Windows NFS server.
3. Change the client configuration if necessary.
4. Copy over the data files to Windows.
5. Shut down the OS/2 NFS server.

4.10.4 Installation on the target platform

Windows 2000 does not include an implementation of an NFS server. Either the NFS server from Microsoft's *Windows Services for UNIX*, the *Hummingbird Maestro NFS Server 8.0*, or any other third party NFS server has to be used. In this example, the Hummingbird NFS server was used, because it fits not only into Active Directory Services from Microsoft, but is also able to authenticate against common directory servers like NIS, NIS+, and LDAP.

4.10.5 AD4UNIX installation

To use the Hummingbird NFS server with the Microsoft Active Directory, install *AD4UNIX*, which is available from:

<http://www.css-solutions.ca>

To install this plug-in, Enterprise Administrator rights are required in the Windows environment. The plug-in should be installed on the Schema master. In the example, Version 1.54 of the plug-in was used and it was required to be installed from a local drive with English as the assumed system language. Once the plug-in is installed, the language can be switched back to the preferred one. All configuration options regarding the plug-in were left at default values.

Attention: AD4UNIX could not be installed from a network drive. It must be installed using a local drive as the source, such as a CD-ROM, or local hard drive.

4.10.6 Hummingbird Maestro™ NFS server installation

The installation of the NFS server is rather simple. Either start the self executable program, or do a master installation on a machine, pack the files together, and export the registry path: HKEY_LOCAL_MACHINE\SOFTWARE\Hummingbird*

Important: When installing on a master machine, be sure to have the NFS server configured before backing up the configuration.

4.10.7 Hummingbird Maestro NFS server configuration

The main configuration is stored in the Windows registry. The user mapping (if needed) is stored in the file `mapping.map` and the exported file systems are stored in the file `exports`. These two files are located in the path `%Installpath%\NFS Maestro Server\Program Files\Hummingbird\Connectivity\8.00\NFSServer\`

In the example, in the Hummingbird Directory Services Properties, the Directory service was changed to LDAP, and Directory Services and DNS was selected as the preferred Hostname lookup method to check the DNS first. The LDAP properties are shown in Table 4-5.

Table 4-5 Directory services properties

Key	Value
LDAP Server Name	Full Qualified Host Name
Search base	dc=somedomain,dc=local
username	CN=ADSRead,OU=Services,OU=Users,OU=Central,DC=somedomain,DC=local
password	password
Search timeout	2 minutes
Maximum matches	0
Schema Style	AD4UNIX
Automount	Automount and Automountmap

On the name mappings page, **Directory services** was selected as the UIDs/GIDs source.

4.10.8 Hummingbird Maestro NFS server configuration

To configure the NFS server itself, settings are in the Windows control panel.

For an automatic configuration the preconfigured registry settings were used. If you want to duplicate an installation across a number of servers, you will need to copy the registry settings and then modify them appropriately on each machine. Unfortunately, this might be the only way to automate the configuration.

To take advantage of centralized management through the Active Directory or LDAP, the NFS server was configured to use Directory Services as the authentication method. This setting should be made in the NFS server configuration on the Name Mappings page. This NFS server gives you a very wide spectrum of features to configure. In this book only a few basic settings are covered.

Example 4-35 Hummingbird exports file

```
E:\NFS -sec=sys
```

This configures the NFS server to export the Directory E:\NFS. Clients can see this export as /E/NFS. It is simple to copy the exports file from one machine to another, as long as the path names do not differ. The security settings in this example differ from the one in our source, such as all users in the domain will have access to this NFS share, and they have to mount with user name and password instead of the UID and GID. For more information about security and setting up the Hummingbird NFS server, consult the Hummingbird documentation, which is shipped with the product.

4.10.9 Windows services for UNIX installation

To use the Microsoft NFS server, *Windows Services for UNIX* (SFU) must be installed. The current release, while writing this book, is 3.0 (SFU 3.0).

Attention: Windows services for UNIX cannot be installed from a network drive. It must be installed using local drives as the source, such as a CD-ROM or the local hard drive.

The Microsoft NFS server has an extension for Active Directory Schema as well, so an NIS domain can be maintained in Active Directory. Unfortunately, OS/2 NFS clients do not support NIS. Instead, use the PCNFS server and configure a user mapping.

Note: When using Windows services for UNIX along with OS/2 clients, maintaining an additional user database (mapping file) in the PCNFSD is required.

Windows services for UNIX was installed with these features:

- ▶ All utilities
- ▶ Interix GNU utilities
- ▶ NFS client
- ▶ NFS server
- ▶ Server for NIS
- ▶ Password synchronization
- ▶ All authentication tools for NIS

4.10.10 Windows services for UNIX configuration

The way to export folders using the SFU is not similar to the OS/2, UNIX, or the Hummingbird way. There is no exports file on the disk. Instead, the NFS sharing is seamless integrated in the Windows GUI. To export (share) a folder, right-click on a folder in the Windows explorer and select the **NFS Sharing** tab. When sharing the folder, an NFS share name is required. In the example, the folder E:\NFS was shared as NFS. The anonymous setting was used to allow access and permissions for all machines with read-write access. Also, a sample user and group was added to gain access to the NFS share from an OS/2 client.

It is also possible to share an NFS folder using the command line interface. Many UNIX and or OS/2 administrators often prefer this way of administering shares. The syntax for this command is:

```
nfsshare.exe <sharename>=<drive:\path>
```

To see what the server exports, type **showmount -e <hostname>** in a windows command prompt or **showexp <hostname>** in an OS/2 command prompt.

For further information on Windows services for UNIX, refer to the Microsoft documentation and white papers.

4.11 Migrating OS/2 FTP server to Windows 2000

File Transfer Protocol (FTP) is a simple and highly standard way to exchange files over the Internet.

4.11.1 Software requirements

In order to migrate the configuration, the following requirements apply to the OS/2 Server:

- ▶ The OS/2 Server is up and running.
- ▶ The FTP server is installed, configured properly, and running

For the Windows server, the following requirements applies:

- ▶ The Windows server is up and running

4.11.2 Migration scenario

The migration scenario is:

1. Create the users on the Windows server, if they are not created yet.
2. Configure the Windows FTP server.
3. Copy the data for each user.
4. Start the Windows FTP server.
5. Shutdown the OS/2 FTP server.

4.11.3 Source platform configuration

The OS/2 FTP Server configuration data is stored in the ETC directory and is named TCPNBK.LST. In this file, all relevant information about the users, and the resources in which they have access, is stored. The password is stored in an irreversible hash (SHA-1) format, which is good from the security perspective, but not good from the migration point-of-view. Since the algorithm used for storing passwords is irreversible, either a new password must be defined, or the ones stored within the ADS-tree defined during the user-migration have to be used. If users are defined on the OS/2 FTP Server, and they are not defined in the LAN server domain, they have to be added to the ADS or user database on Windows as well.

Example 4-36 Sample user from tcpnbk.lst

```
SRVRUSR=(
  USERID=marc
  password=5E1B89FA57B93CB4CEC8F2DE5EDA95E14E77E6A3
  comment=Marc Schneider
  uid=
  gid=
  shell= telnetd.cmd
  homedir=f:\ftp\home\marc
  FTPD=(
    active=1
    read=f:\ftp\home\marc
```

```
        canread=1
        write=f:\ftp\home\marc
        canwrite=1
        log=
        idletimeout=900
    )
    TELNETD=(
        active=0
    )
    rexecd=(
        active=0
    )
    nfsd=(
        active=0
    )
)
```

4.11.4 Target platform

To have the FTP server on the Windows side installed, the Internet Information Server (IIS) package must be installed, or a third party tool should be used like the Hummingbird InetD FTP Server.

4.11.5 Hummingbird FTP server installation

The Hummingbird FTP daemon is included in the Hummingbird InetD. This package also includes some UNIX daemons and as an additional feature, the Telnet daemon might be useful. To install the package either click the **Installer** or extract the msi file from the package and install it using `msiexec`. This command is a standard command included in Windows 2000 and XP.

4.11.6 Hummingbird InetD configuration

The Hummingbird InetD configuration information is stored in the WINNT tree. The correct Path in our example is:

D:\WINNT\system32\Hummingbird\Connectivity\8.00\Inet\InetD.ini and not the one described in the documentation as InetD.cfg.

An example configuration with FTPD and TELNETD configured is shown in Example 4-37.

Example 4-37 InetD.ini

```
[Global]
Logging=0
[Telnetd]
Program=telnetd
```

```
Parameters=  
Port=23  
MaxServer=8  
Enable=1  
Protocol=TCP  
[Ftpd]  
Program=ftpdw  
Parameters=  
Port=21  
MaxServer=4  
Enable=1  
Protocol=TCP
```

For a proper user authentication, add the Group FTPaccess to the Active Directory, LDAP, or the local account database. If this group does not exist, the access is granted for every Windows user account. The Disklevel security applies as on normal NTFS drives.

Note: Implement the group FTPAccess to make the FTP server secure.

To assign directories to which the users can connect during an FTP (or Telnet) session, configure this path in the user profile. The next time the user logs on, the starting directory will be the one assigned in the profile settings.

4.11.7 Microsoft IIS Server installation (FTP server)

To use the FTP server from Microsoft IIS, it is not necessarily required to install the whole IIS Pack. The installer allows you to just install the FTP server feature. The Microsoft FTP server feature fits nearly seamlessly into the system.

To configure the Microsoft FTP Server automatically, its recommended to make use of the Active Directory Scripting Interface (ADSI) or a similar function like WMI.

To get the user directories in the FTP server up and running, virtual directories must be set up in the FTP server. To do so, follow these steps:

1. Create a personal folder in the filesystem, if it does not exist already.
2. Create a virtual directory and map the personal folder to the virtual directory. The name of the virtual directory must match exactly the name of the user (case sensitive).
3. Remove anonymous authentication and enable write access to the virtual directory.
4. Set NTFS permissions to the personal folder.

Important: The FTP users need to have the right to log on locally.

For additional configuration of Microsoft's FTP server, consult the Microsoft IIS server documentation.

To create virtual directories on an automated basis, you can use a script like the following sample.

Example 4-38 Virtual directories in IIS FTP server

```
option Explicit

Dim Site, SitePath, Computer, IISOBJ,
Dim DirName, DirPath, vDir

Site = "MSFTPSVC/1"
Computer = "LocalHost"
Dirname = "marc"
DirPath = "e:\ftp\marc"

SitePath = "IIS://" & Computer & "/" & Site & "/ROOT"
set IISOBJ = GetObject(SitePath)

Set vDir = getObject(Sitepath & "/" & DirName)
Set vDir = vRoot.Create("IISFtpVirtualDir", DirName)
vDir.AccessRead = true
vDir.Path = DirPath
vDir.SetInfo
```

4.12 DHCP server migration

The Dynamic Host Configuration Protocol (DHCP) is an Internet protocol for automating the configuration of computers that use TCP/IP. DHCP can be used to automatically assign IP addresses, to deliver TCP/IP stack configuration parameters such as the subnet mask and default router, and to provide other configuration information such as the addresses for printer, time, and news servers.

DHCP's purpose is to enable individual computers on an IP network to extract their configurations from a server (the "DHCP server") or servers. The overall purpose of this is to reduce the work necessary to administer a large IP network. The most significant piece of information distributed in this manner is the IP address.

4.12.1 Software requirements

In order to migrate the configuration, the following requirements apply to the OS/2 Server:

- ▶ The OS/2 Server is up and running.
- ▶ The DHCP server is installed.
- ▶ The DHCP server is configured properly and running.

For the Windows server, the following requirement applies:

- ▶ The Windows server is up and running.

4.12.2 Migration scenario

The following section describes the steps to migrate a DHCP server from OS/2 to Windows. The migration scenario is:

1. Decrease the lease time on the OS/2 Server. In this way the clients will update the configuration sooner after the new server is on line.
2. Stop the OS/2 DHCP server.
3. Migrate the configuration from OS/2 Server to the Windows server.
4. Start the DHCP server on Windows.

Important: Stop the OS/2 DHCP server before starting the DHCP server on Windows. In general, two DHCP servers should not be run on the same subnet.

4.12.3 Source platform

Before migrating DHCP services, it is recommended to decrease the lease time on your OS/2 DHCP server. This way, it is ensured that clients will get changes sooner, and eventually reserved addresses for some clients can be reassigned by the new DHCP server.

The OS/2 DHCP server configuration file can be found in the server's ETC directory (which usually is C:\MPTN\ETC). The file is named DHCP.D.CFG and contains all information about the services the OS/2 DHCP server provides to the subnetworks for which it is configured.

Example 4-39 dhcpcd.cfg

```
logFileName dhcpcd.log
logFileSize 100
numLogFiles 10
logItem SYSERR
```

```

logItem OBJERR
logItem WARNING
logItem INFO
leaseExpireInterval 1 Minutes
leaseTimeDefault 24 Minutes
pingTime 1 Seconds
reservedTime 5 Minutes
usedIPAddressExpireInterval 1000 Seconds
statisticSnapshot 1
updateDNSA "nsupdate -f -h%s -s"d;a;*;a;a;%s;s;%s;3110400;q""
releaseDNSA "nsupdate -f -h%s -s"d;a;%s;s;%s;0;q""
(ARecKeyInfo somedomain.local 127.0.0.1
supportBOOTP no
supportUnlistedClients both
allRoutesBroadcast no
UserMatchesVendorClass no
servertype dhcp

appendDomainName yes
canonical no
proxyARec no
#vendor PXEClient
subnet 192.168.25.0 255.255.255.0 192.168.25.10-192.168.25.200 (alias=SOMENAME
{
    supportUnlistedClients no
    client 0 0 192.168.25.30
    client 0 0 192.168.25.31
    client 0 0 192.168.25.32
    client 0 0 192.168.25.33
    client 0 0 192.168.25.34
    client 0 0 192.168.25.35
    client 0 0 192.168.25.36
    client 0 0 192.168.25.37
    client 0 0 192.168.25.38
    client 0 0 192.168.25.39
    client 0 0 192.168.25.40
    option 6 192.168.25.2
    option 3 192.168.25.1
    option 15 dhcp.somedomain.local
}

```

In this example, the server would not give an IP address to any clients because unlisted clients would not be served (`supportUnlistedClients no`) and there are no listed clients in the database. This setting was made only for testing purposes to not conflict with other DHCP servers in the testing environment.

4.12.4 DHCP server installation

The Server versions of Windows 2000 and XP include a DHCP server which is installable through **Add Programs** in the **Windows Control Panel**. To perform an automated installation see 2.1.3, “DHCP server” on page 23.

4.12.5 DHCP server configuration

While migrating to the Windows platform, it can be tedious to fill in all configuration data by hand, so it is recommended to use the **netsh** command, which is a command-line scripting utility that allows you, either locally or remotely, to display or modify the network configuration of a Windows machine that is currently running. **netsh** also provides a scripting feature that allows you to run a group of commands in batch mode against a specified Windows machine.

Example 4-40 Sample netsh script for DHCP server

```
Dhcp Server 127.0.0.1 Set AuditLog "D:\WINNT\System32\dhcp"
Dhcp Server 127.0.0.1 Set DatabaseBackupInterval 60
Dhcp Server 127.0.0.1 Set DatabaseBackupPath "D:\WINNT\System32\dhcp\backup"
Dhcp Server 127.0.0.1 Set DatabaseCleanupInterval 1440
Dhcp Server 127.0.0.1 Set DatabaseLoggingFlag 1
Dhcp Server 127.0.0.1 Set DatabaseName "dhcp.mdb"
Dhcp Server 127.0.0.1 Set DatabasePath "D:\WINNT\System32\dhcp"
Dhcp Server 127.0.0.1 Set DatabaseRestoreFlag 0
Dhcp Server 127.0.0.1 Set DetectConflictRetry 0

Dhcp Server 127.0.0.1 add scope 192.168.25.0 255.255.255.0 "SOMENAME" ""

# we dont want the dhcp Server to serv now, so disabling this scope
Dhcp Server 127.0.0.1 Scope 192.168.0.0 set state 0

Dhcp Server 127.0.0.1 Scope 192.168.25.0 Add iprange 192.168.25.10
192.168.25.200

Dhcp Server 127.0.0.1 Scope 192.168.25.0 add excluderange 192.168.25.30
192.168.25.40

Dhcp Server 9.3.4.12 Scope 192.168.0.0 set optionvalue 3 IPADDRESS
"192.168.25.3"
Dhcp Server 9.3.4.12 Scope 192.168.0.0 set optionvalue 6 IPADDRESS
"192.168.25.2"
Dhcp Server 9.3.4.12 Scope 192.168.0.0 set optionvalue 15 STRING
"dhcp.somedomain.local"
```

Since it is not very convenient to create scripts like this manually, a REXX script can be used to build the **netsh** import script for you.

For additional information regarding **netsh** consult the Microsoft online help.

4.13 DDNS server migration

The Domain Name System (DNS) is a distributed Internet directory service. DNS is used mostly to translate between domain names and IP addresses, and to control Internet e-mail delivery. Most Internet services rely on DNS to work, and if DNS fails, Web sites cannot be located and e-mail delivery stalls. DDNS is a Dynamic DNS service that allows you to assign a fixed machine name to a dynamic IP address. Dynamic DNS provides the ability to change the IP address of a domain name to point to your dynamically allocated IP address.

4.13.1 Software requirements

In order to migrate the configuration, the following requirements apply to the OS/2 Server:

- ▶ The OS/2 Server is up and running.
- ▶ The DNS server is installed, configured properly, and running.

For the Windows server, the following requirements applies:

- ▶ The Windows server is up and running.

4.13.2 Migration scenario

This is an overview of the possibilities for migrating DHCP servers.

Migration scenario using DHCP

The migration scenario using DHCP is easier, and it does not affect the clients. The migration steps are:

1. Decrease the IP lease time on the DHCP server so the clients will update the IP configuration sooner.
2. Build a secondary DNS on Windows and replicate the configuration.
3. After the DNS configuration is replicated, reconfigure the Windows DNS server to be a primary server and the OS/2 Server to be a secondary DNS server.
4. After the Windows DNS server is up and running, change the DHCP configuration so the clients receive only one DNS server, which is the Windows DNS server.

Note: The OS/2 DNS configuration has to be modified to allow a secondary DNS to replicate the configuration.

The scenario works as follows. At the first logon of a client, it receives and uses the old DNS address (OS/2 Server) from the DHCP server. After the new DNS server is up and running (Windows server), the DHCP configuration is changed with the new DNS address. When a client is reconfigured by DHCP or the lease time expires, it requests from the DHCP server a new IP configuration. The DHCP responds with the new IP configuration including the new DNS server, and the clients will use the Windows DNS server instead the OS/2 Server.

Migrating scenario without DHCP

In this situation there is no smooth migration, because it affects the clients. The network administrator has to manually modify the network client configuration.

The migration steps are:

1. Migrate the DNS configuration from OS/2 DNS server to Windows DNS server (you can replicate this as well using the Windows server as secondary first).
2. Start the Windows DNS server.
3. After the Windows DNS server is up and running, the OS/2 DNS server can be shut down.

4.13.3 Source platform

On the OS/2 DDNS Server, the configuration data is stored in the ETC and ETC\NAMEDB directories. In the ETC directory, only one file is stored for DDNS usage. This file has the obvious name DDNS.DAT. Only authentication keys are stored in this file. These keys are used by the clients to identify themselves against the DDNS server while the client is updating its own IP address.

All other information regarding the DDNS server is stored in the %ETC%\NAMEDB directory.

The **namedb.bt** file holds the boot configuration for the DDNS server. This configuration points to the used files, which are to be loaded by the server. The cache file is used for caching.

Example 4-41 namedb.bt

```
; ***** IBM DDNS Server Administrator *****  
; This file was written by the IBM DDNS Server Administrator on 04-Jun-03  
; ***** IBM DDNS Server Administrator *****
```

```
primary somename.local C:\\MPTN\\ETC\\namedb\\dnstf0000.dom dynamic secured
primary 4.3.9.in-addr.arpa C:\\MPTN\\ETC\\namedb\\dnstf0000.rev dynamic
secured
cache . C:\\MPTN\\ETC\\namedb\\named.ca
```

The **dnstxt.cfg** file holds information about all configured zones for this server. In this example are the reverse and forward lookup zone parameters.

Example 4-42 dnstxt.cfg

```
; ***** IBM DDNS Server Administrator *****
; This file was written by the IBM DDNS Server Administrator on 04-Jun-03
; ***** IBM DDNS Server Administrator *****
4.3.9.in-addr.arpa (
notify=yes
notify.delayTime=60
notify.retryTime=30
notify.retryNumber=3
timeSync=yes
timeSync.toSecondaries=yes
safeWrite=yes
sigDel=no
ttlSet=no
deferUpdCnt=100
incrTime=300
keyToSec=yes
sepDynStatic=yes
reverseMapping=yes
)
somedomain.local (
notify=yes
notify.delayTime=60
notify.retryTime=30
notify.retryNumber=3
timeSync=yes
timeSync.toSecondaries=yes
safeWrite=yes
sigDel=no
ttlSet=no
deferUpdCnt=100
incrTime=300
keyToSec=yes
sepDynStatic=yes
reverseMapping=yes
)
DDNSAdministratorClient (
gui.warn=yes
gui.write=yes
gui.num=100
```

```
gui.lease=3600
gui.pad=3110400
gui.reinit=1
gui.sepdata=3
)
```

If not named otherwise, the `dnsf0000.dom` and `dnsf0000.rev` files store the dynamic forward, and the reverse lookup information including the serial and the TTL times for the zone itself.

Example 4-43 dnsf0000.dom

```
; ***** IBM DDNS Server Administrator *****
; This file was written by the IBM DDNS Server Administrator on 04-Jun-03
; ***** IBM DDNS Server Administrator *****
$ORIGIN somedomain.local.
@ IN SOA pdc. (
    2
    10800
    3600
    604800
    86400 )
somedomain.local. IN KEY 80 0 1
AQPLEFH6yFZZGjodKMudsNaDZu//H7ik0o7hGAFyYr87X1ZyM3pc9BZP+YzG5oJ4qPLNEV2Ip0JwEKS
Jtb49F+8Z
@ IN NS pdc.
$INCLUDE C:\\MPTN\\ETC\\namedb\\dnsf0000.sta
```

The two files referenced by the `dnsf0000.dom` and `dnsf0000.rev` files are the static entries for the domain. In this case, the files are named `dnsf0000.sta` and `dnsf0001.sta`.

Example 4-44 dnsf0000.sta

```
; ***** IBM DDNS Server Administrator *****
; This file was written by the IBM DDNS Server Administrator on 04-Jun-03
; ***** IBM DDNS Server Administrator *****
$ORIGIN somedomain.local.
pdc. IN A 127.0.0.1
bdc IN A 9.3.4.11
ns-updates IN CNAME pdc.
pdc IN A 9.3.4.9
dhcp IN CNAME pdc.
ns IN CNAME pdc.
ddns IN CNAME pdc.
```

4.13.4 Target platform

An easy way to migrate from OS/2 to another platform, in this case Windows, is to build up a new DNS server and configure this machine to be the secondary DNS server to the existing, primary one. The whole configuration will then be replicated.

4.13.5 DDNS server installation

Like the DHCP server, there is a DDNS server included in Windows 2000 and XP server versions as well. For the installation, use either *Add Programs* in the Windows Control Panel, or do an automated installation as seen in 2.1.5, “DNS server” on page 23.

4.13.6 DDNS server configuration

To configure the server you can use the GUI, do basic settings, and then replicate the zone information.

A different approach, and a way to do it unattended, is to use the **DNSCMD** command, which is part of the Windows Server Resource Kit. The syntax to use with **DNSCMD** is very descriptive.

Example 4-45 Sample DNSCMD script

```
DNSCMD /ZoneDelete . /DsDel /f
DNSCMD /ResetForwarders 9.3.4.2

DNSCMD /ZoneAdd somedomain.local /DsPrimary
DNSCMD /Config somedomain.local /AllowUpdate 1
DNSCMD /Config somedomain.local /SecureSecondaries 0

DNSCMD /RecordAdd somedomain.local pdcA 9.3.4.17
DNSCMD /RecordAdd somedomain.local timesrv CNAME pdc.somedomain.local

DNSCMD /ZoneAdd 4.3.9.in-addr.arpa /DsPrimary
DNSCMD /Config 4.3.9.in-addr.arpa /AllowUpdate 1
DNSCMD /Config 4.3.9.in-addr.arpa /SecureSecondaries 0
DNSCMD /RecordAdd 4.3.9.in-addr.arpa 17 PTR pdc.somedomain.local
```

This sample script will set up the DDNS server with the most important information.

For additional information about DNSCMD and the DDNS server configuration, see the Microsoft documentation.

4.14 Summary

After performing the steps described in this chapter, the basic infrastructure supplied by the OS/2 Server will have been replicated in a Windows 2000 domain. Information such as user definitions and profiles, printer definitions, and all other objects from the OS/2 domain should now be available to the client systems.

In the next chapter, we provide some additional information on migrating common middleware such as database systems, communications servers, and so on.



Migrating the software stack to Windows 2000

This chapter provides an overview of recommendations and activities to migrate the major IBM middleware products currently implemented, and which are used by a majority of customers on OS/2 to their equivalent product versions on Windows 2000.

5.1 Migrating IBM Universal Database

Migrating the IBM DB2 from one platform to another is a complex task and might be time consuming. It is highly recommended to research and thoroughly test the procedures before making any changes to your production environment. In addition, a backup of all data should be performed.

Typically, there are a number of applications that use DB2 as their data store, and migration becomes more of an application migration issue rather than a database migration.

The following sections describe the migration steps, some procedures, and useful tips. Another good source for information regarding migrating a DB2 database is the DB2 User Guide.

Important: For detailed information, step by step procedures, and how-to's visit:

<http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/document.d2w/report?fn=db2v7dmfrm3toc.htm>

5.1.1 Migration scenario

The migration scenario involves the following steps:

1. Install and configure the target platform (including patches if necessary).
2. Choose a time when the DB2 server is lightly used.
3. Export the data from the source DB2 server.
4. Import the data to the target DB2 server.
5. Change the application links to match the new configuration.

5.1.2 Exporting and importing the data

Compatibility is important when exporting, importing, or loading data across platforms. There are several options available for moving the databases from one platform to another:

1. Moving data across platforms

- PC/IXF File Format

PC/IXF is the recommended file format for transferring data across platforms. PC/IXF files allow the load utility or the import utility to process (normally machine dependent) numeric data in a machine-independent fashion. For example, numeric data is stored and handled differently by Intel and other hardware architectures, such as mainframes.

- Delimited ASCII (DEL) File Format. DEL files can have differences based on the operating system on which they were created. These differences include:
 - Row separator characters
 - UNIX based text files use a line feed (LF) character.
 - Non-UNIX based text files use a carriage return/line feed (CRLF) sequence.
 - End-of-file character
 - UNIX based text files do not have an end-of-file character.
 - Non-UNIX based text files have an end-of-file character (X'1A').
- WSF file format
 - Numeric data in WSF format files is stored using an Intel machine format. This format allows Lotus WSF files to be transferred and used in different Lotus operating environments (for example, in Intel based and UNIX based systems).

2. Moving data using the db2move tool

This tool facilitates the movement of large numbers of tables between DB2 databases located on workstations. The tool queries the system catalog tables for a particular database, and compiles a list of all user tables. It then exports these tables in a PC/IXF format. The PC/IXF files can be imported or loaded to another local DB2 database on the same system, or can be transferred to another workstation platform, and imported or loaded to a DB2 database on that platform.

3. Moving data with DB2 Connect™

If you are working in a complex environment in which you need to move data between a host database system and a workstation, you can use DB2 Connect, the gateway for data transfer from the host to the workstation, as well as the reverse.

4. Moving data between typed tables

The DB2 export and import utilities can be used to move data out of, and into, typed tables. Typed tables may be in a hierarchy. Data movement across hierarchies can include:

- Movement from one hierarchy to an identical hierarchy
- Movement from one hierarchy to a sub-section of a larger hierarchy
- Movement from a sub-section of a large hierarchy to a separate hierarchy

5. Using replication to move data

Replication allows you to copy data on a regular basis to multiple remote databases. If you need to have updates to a master database automatically

copied to other databases, you can use the replication features of DB2 to specify what data should be copied, which database tables the data should be copied to, and how often the updates should be copied. The replication features in DB2 are part of a larger IBM solution for replicating data in small and large enterprises.

6. Using the Data Warehouse Center to move data

You can use the Data Warehouse Center (DWC) to move data from operational databases to a warehouse database, which users can query for decision support. You can also use the DWC to define the structure of the operational databases, called sources. You can then specify how the operational data is to be moved and transformed for the warehouse. You can model the structure of the tables in the warehouse database, called targets, or build the tables automatically as part of the process of defining the data movement operations. The Data Warehouse Center uses the following DB2 functions to move and transform data:

- a. SQL: You can use SQL to select data from sources and insert the data into targets. You also can use SQL to transform the data into its warehouse format. You can use the Data Warehouse Center to generate the SQL, or you can write your own SQL.
- b. Load and export utilities: You can use these DB2 utilities to export data from a source, and then load the data into a target. These utilities are useful if you need to move large quantities of data.

5.2 Migrating IBM e-Network Communications Server

Communications Server provides an essential foundation for networked computing by supporting the most widely used networking technologies, enabling customers and business partners to build client and server applications independent of networking protocol or hardware. The full implementation of APPN (end node and network node), HPR, and DLUR, along with the integrated SNA gateway capabilities, allows the Communications Server to participate in either a host (hierarchical) or peer-to-peer distributed network environment.

5.2.1 Source platform configuration

The Communications Server on OS/2 is configured to have an Enterprise Extender Link to the S/390® and local SNA links to the clients. This is a widely used configuration. With this configuration you do not have the problem of needing special hardware such as routers, which are capable of building a DLSW Tunnel or an X.25 link to the mainframe.

5.2.2 Migration scenario

The migration steps include:

1. Install Communications Server on Windows.
2. Export the Communications Server configuration on OS/2.
3. Convert the configuration files from OS/2.
4. Import the configuration files on Windows.
5. Stop the OS/2 Communications Server.
6. Start the Windows Communications Server.

5.2.3 Communications Server installation

The automated installation of Communications Server can be found in chapter 2.5.2, “IBM Communication Server” on page 44.

5.2.4 Migrating the configuration

No matter how Communications Server on OS/2 is configured, IBM provides a utility that is able to convert any Communications Server configuration files from OS/2 to Windows. This utility is available at:

<http://www-3.ibm.com/software/network/commserver/downloads/enhancements/csos2.html>

Export the Communications Server configuration

To export the configuration data on OS/2, open an OS/2 command prompt and change to the CMLIB directory (usually C:\CMLIB) and type **cmrecord <configuration name>**. The command **cmrecord** will extract the named configuration out of the Communications Server and save it with **.RSP** as a suffix in the same directory, if not otherwise specified. For help on the additional parameters for **cmrecord**, type **cmrecord** without any parameters. **cmrecord** now has generated a response file, which you can convert using the conversion utility.

Convert the configuration files from OS/2

The conversion can be done on any Windows operated machine. To install the migration utility, open a Windows command prompt, and change to the installation directory (usually c:\migrate).

Important: If the files contain a SDLC or X.25 DLC, make sure that the corresponding PROTOCOL.INI file is included in the same directory as the .RSP file. Use the same file name as the .RSP; for example, MIGRATE.RSP and MIGRATE.INI. If no file of the same name is found, the migration utility defaults to PROTOCOL.INI.

To migrate a single file, type the following command at a Windows command prompt:

```
oocmigcm <source.rsp> [dest.acg]
```

Table 5-1 Options for oomigcm

Parameter	Description
source.rsp	Specifies the name of the response file to migrate
dest.acg	Specifies the name of the output file to create. You can use a fully-qualified name to put the new file in another directory. If you do not specify a name, the output is placed in MIGRATE.ACG.

The output .ACG file is placed in the same directory as the source .RSP file. Log files (.LOG) are placed in the directory from which you ran the utility.

Other files in the directory will be ignored. Output .ACG files have the same name as their corresponding source .RSP files. For example, MIGRATE.RSP will be migrated to MIGRATE.ACG.

Attention: No password security fields are migrated.

Import the configuration files on Windows

To import the generated .ACG file on Windows you have to put it into the Communications Server configuration directory (usually C:\IBMCS\PRIVATE) and define this new configuration as your default configuration.

More Information

For more information about Communications Server see the these redbooks:

- ▶ *IBM eNetwork Communications Server for Windows NT Version 6.0 Enhancements*, SG24-5232-00

- ▶ *IBM eNetwork Communications Server for OS/2 Warp Version 5.0 Enhancements*, SG24-2147-00
- ▶ *IBM Communication Controller Migration Guide*, SG24-6298-00

5.3 Migrating Lotus Domino

In the following sections we will describe the Lotus Domino migration from OS/2 to Windows. We will describe how to migrate from Lotus Domino version 5.x on OS/2 to Lotus Domino version 5.x to Windows.

Note: If you are running Lotus Domino version 4.x on OS/2, we recommend to upgrade to Lotus Domino version 5.x and then to migrate to a Windows server. The upgrade process is not within the scope of this book, please refer to the Lotus documentation.

Note: If you want to use Lotus Domino version 6.x on Windows, we recommend to first migrate from version 5.x on OS/2 and then to upgrade to version 6.x on Windows. Lotus Domino version 6.x does not exist on OS/2.

5.3.1 Migration scenario

The migration steps include:

1. Install Lotus Domino on Windows, at the time of writing this book the release for version 5 is 5.0.12.
2. Copy the notes\data directory from OS/2 to Windows through FTP or NetBIOS or a backup/restore procedure.
3. Copy the notes.ini file from OS/2 to Windows in the notes\data directory.
4. Change the ownership to the notes user for the notes\data directory.
5. Modify the notes.ini file to reflect the new path for the notes\data directory.
6. Start the Lotus Domino server on Windows.

Note: In order for the migration to be transparent for clients, we can change the DNS entry for the Lotus Domino server to reflect the new IP address. If you are not using a DNS server, you have to stop the OS/2 Server and move the IP address to the Windows server.

5.3.2 Migrating the configuration

The steps to perform the actual migration are simple and can be seen in 5.3.1, “Migration scenario” on page 183.

5.4 Migrating IBM HTTP Server

IBM fortunately has ported this product to many platforms, so a migration is simple and straight forward.

5.4.1 Software requirements

In order to migrate the configuration, the following requirements apply to the OS/2 Server:

- ▶ The OS/2 Server is up and running.
- ▶ The IBM HTTP Server is installed.
- ▶ The IBM HTTP Server is configured properly and running.

For the Windows server, the following requirement applies:

- ▶ The Windows server is up and running

5.4.2 Migration scenario

The migration steps include:

1. Copy the Web documents from OS2 to Windows.
2. Copy and modify the configuration file *httpd.conf*.
3. Start the IBM HTTP Server on Windows.
4. Update the DNS entry with the new Web server IP address, or stop the OS2 server, and set the IP address on the Windows Web server.

5.4.3 Installing IBM HTTP Server

To install the IBM HTTP Server on Windows, you have to first install the Java Developer Kit 1.3 from IBM, which is available at the IBM Developers Web site. Be sure to install all parts of the JDK before installing the HTTP Server. In the example, the IBM HTTP Server version 1.3.26.1, which was available at: <http://www-3.ibm.com/software/webservers/htpservers/> while writing this book, was used. This version comes very close to the 1.3.20 on the source platform. Once IBM Java 1.3 is installed, you can proceed by installing the HTTP Server for Windows.

Note: Install Java 1.3 JDK before you install the Web server.

To install this version, open a command prompt and change to the directory to where the install package is. Now type `java -jar setup.jar` and you will be guided through the installation process by the Java installer.

5.4.4 Migrating the IBM HTTP Server

The configuration file `httpd.conf` in the `conf` directory is very similar on both platforms. The main differences are the absolute directories and modules.

Copy the document directory to the target platform. This is usually the `htdocs` directory on both sides. You should change all absolute path information in the entire `httpd.conf` file:

F:/IBM-HTTPD becomes D:/Program Files/IBM HTTP Server

A simple find and replace on this file, maybe with a REXX procedure, is a good choice for changing these parameters. It does not matter if you use forward or backward slashes in the lines containing absolute paths.

The modules are a bit difficult if used. If not, comment out the `LoadModule` statements and the module configuration statements `<IfModule module_name>`.

5.5 Migrating TSM Client

OS/2 uses the ADSM client. At the time of writing this book the latest version of TSM is 5.1.5. We have a TSM server installed on an AIX server version 5.1.5. In order to successfully migrate the ADSM client, you have to have at least TSM server version 5. If you have an earlier version of TSM or ADSM server, you need to upgrade the server to support the TSM clients. The TSM server upgrade is beyond the scope of this book.

5.5.1 Software requirements

In order to migrate the configuration, the following requirements apply for the OS/2 Server:

- ▶ The OS/2 Server is up and running.
- ▶ The ADSM client is installed and configured properly.

For the Windows server, the following requirements apply:

- ▶ The Windows server is up and running.

- ▶ The TSM client is installed.

5.5.2 Migration scenario

The migration scenario is:

1. Copy the *dsm.opt* file to the Windows server through FTP or NFS.
2. Start the TSM client.

The TSM client and server version 5.x has a feature useful in migration scenarios. TSM client allows you to access the backups of another node while already connected with your account. In this case, it is useful to access the OS2 backup (when needed) without modifying the *dsm.opt* file or the *dsm.sys* file.

5.5.3 Migrating the configuration

Since the configuration is forward compatible, only the above two steps are required. Be aware that some of the files backed up on OS/2 will become useless on Windows, such as some OS/2 specific configuration files. Also, extended attributes used on OS/2 will be lost on Windows.

5.6 Summary

This short chapter has described some of the considerations for migrating various middleware components from OS/2 to Windows 2000. Since each product has its own repository for required configuration information, the ease of creation of an equivalent configuration for the new platform is product specific. In some case, the configuration information can be ported very easily, and in other cases, it may need to be manually rebuilt.

Migration to Linux

The chapters in this part of the book describe a step by step migration to a Linux environment. Data gathered from the OS/2 domain as described in Chapter 3, “Starting the OS/2 Server migration” on page 63, is used and imported to the Linux and Samba V3 environment.

Chapter 6, “Migrating OS/2 Servers to Linux and Samba” on page 189, addresses the steps to fully migrate the OS/2 domain and LAN servers, providing the basic infrastructure.

Chapter 7, “Migrating the software stack to Linux” on page 267, briefly describes the migration considerations for the most common middleware that often exists in OS/2 Server environments.



Migrating OS/2 Servers to Linux and Samba

This chapter describes the migration of the core functions and features from an IBM OS/2 Warp Server domain to Linux as the target platform, including the specifics on SuSE or Red Hat when appropriate.

This chapter covers:

- ▶ LDAP Directory organization and structure setup
- ▶ OS/2 domain objects migration: Domain, Server, Group, User, Directory, Printer, Serial
- ▶ Explores areas of limitations or options for the migration scenarios from OS/2 to Samba
- ▶ Log on assignment considerations
- ▶ Client printing considerations
- ▶ Access control limitations and features from Samba

Before performing the steps in this chapter, the migration should be prepared including data extraction, and retrieving and modifying the domain definition of your OS/2 domain as discussed in Chapter 3, “Starting the OS/2 Server migration” on page 63.

6.1 LDAP directory organization

This first step in migrating the OS/2 domain resources to Linux is to design the target LDAP structure. This section explores the design and container considerations for an enterprise deployment of a distributed environment. The directory solution presented here is incomplete, and not considered a final design, but it is used as the basis for our migration discussion.

The following topics are covered in this section:

- ▶ Overview of directory structure
- ▶ Creation of base enterprise objects in the directory
- ▶ Creation of branch specific areas in the directory
- ▶ Import of the basic directory elements

6.1.1 Directory structure

This section is intended to provide a basis for a common understanding of the directory structure used. This redbook does not serve as a design discussion or reference for directory design practices. The root of the directory is the global context of the company where all objects will be contained. This context is for a name of somedomain.local where this could be actually something like company.com.

Within this directory, the individual locations, which are referred to as branches in this scenario, are organized into organizational units. These individual organizational units (OU) contain the objects for the location or branch.

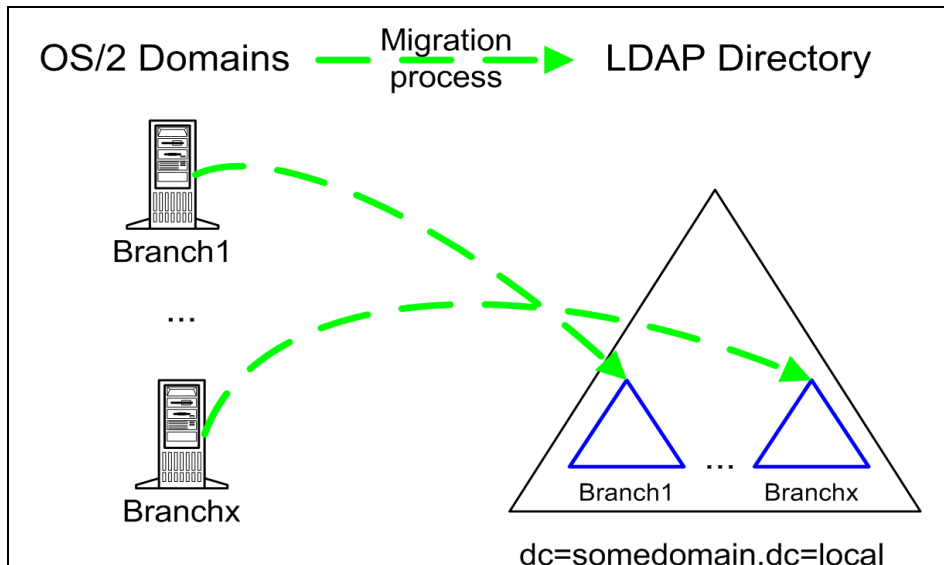


Figure 6-1 LDAP Directory design for transition of OS/2 branch domains

This example design can be modified as required or desired. Additionally, the basic design of the LDAP directory structure enables the integration with a Windows Active Directory Services (ADS) structure.

6.1.2 Enterprise objects

Considering the design of a basic LDAP directory, the basic tree for the enterprise consists of the following:

Central	This OU is the base container for user and group definitions used in a centralized way. Here you can find groups or users that are specific for a service or have been defined in all source domains (for example, administrator accounts, FTP users, and so on).
Systems	Servers and workstations are stored as objects in the directory to put them into an organizational, geographical, or other context. The subsidiary OUs are defined for the different types of workstations (notebooks, standard desktops, specialized workstations) and servers (file, print, domain controllers, application server, and so on).
GPO	Container for group policy objects. This container holds all GPO of the enterprise.

Branch The branch OU is the base object for our migration scenario. All migrated branches are transferred to this context. In larger environments it may be good practice to add a geographic structure like West or East. In this scenario this is omitted for simplification.

Each branch consists of the following OU:

Groups Group definitions of the source OS/2 are transferred here. In the migration process, we will describe concepts to allow a separation depending on their purpose:

Access This can contain groups used to define access control lists (ACL) on resources in this OU.

Organization These groups usually specify membership according to organizational principles, project groups, or distribution lists for e-mail.

Application Application services like Citrix Metaframe or IBM Workspace On-Demand often use group memberships to assign applications to certain users. These application groups would be found here after migration.

Print As for applications, we define print groups that assign shared printer queues to users. We will migrate these types of groups into this OU.

Users All user accounts for the branch will be found in this OU.

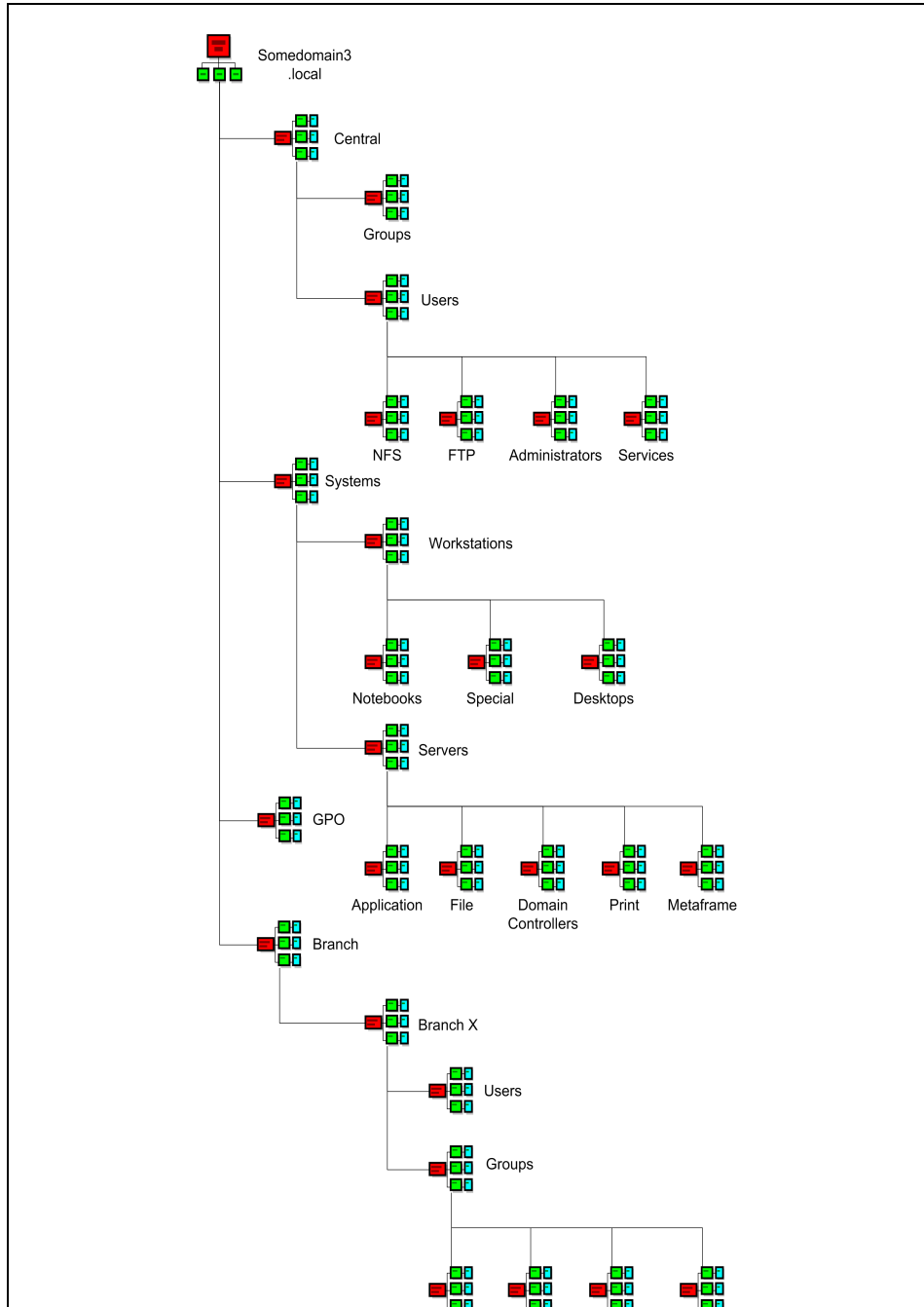


Figure 6-2 LDAP Directory Organizational Unit layout for somedomain.local

6.1.3 Importing basic directory elements and objects

For the most part, the directory modifications will be applied through the LDIF files. There are two main methods that we will use for importing LDIF files: adding and modifying. Both methods use the command `ldapmodify`

Adding LDAP entries with an LDIF file

To import an LDIF file adding new entries, the following command is offered as an example:

```
# ldapmodify -a -f datafile.ldif -D "cn=root,dc=somedomain,dc=local" -w password
```

Modifying LDAP entries with an LDIF file

To import an LDIF file modifying existing entries, the following command is offered as an example:

```
# ldapmodify -f datafile.ldif -D "cn=root,dc=somedomain,dc=local" -w password
```

Review the manual pages for details on the use of the `ldapmodify` command. Note from the above that the `-a` parameter specifies that entries are being added rather than modified. This parameter would be omitted when merely modifying existing directory objects.

Starting with a blank OpenLDAP tree

If you are starting with a blank OpenLDAP tree, at a minimum the following must be imported as shown in Example 6-1.

Example 6-1 Example basetree.ldif file

```
# Organization for Example Corporation
dn: dc=somedomain3,dc=local
objectClass: dcObject
objectClass: organization
dc: somedomain3
o: Example Corporation
description: The Example Corporation

# Organizational Role for Directory Manager
dn: cn=root,dc=somedomain3,dc=local
objectClass: organizationalRole
cn: root
description: Directory Manager
```

Importing this establishes the base organization and role objects for the directory tree.

Setting up the base Organizational Units

Importing the following will set up the base directory tree presented in this chapter as shown in Example 6-2.

Example 6-2 Example baseou.ldif file

```
dn: OU=GPO,DC=somedomain,DC=local
changetype: add
description: Container for Group policy objects
objectClass: organizationalUnit
ou: GPO

dn: OU=Branch,DC=somedomain,DC=local
changetype: add
description: Container for all branches
objectClass: organizationalUnit
ou: Branch

dn: OU=Systems,DC=somedomain,DC=local
changetype: add
description: Base container for computer and server objects
objectClass: organizationalUnit
ou: Systems

dn: OU=Servers,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Server objects
objectClass: organizationalUnit
ou: Servers

dn: OU=Metaframe,OU=Servers,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for Terminal Server objects
objectClass: organizationalUnit
ou: Metaframe

dn: OU=File,OU=Servers,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for file server objects
objectClass: organizationalUnit
ou: File

dn: OU=Print,OU=Servers,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for print server objects
objectClass: organizationalUnit
ou: Print

dn: OU=Domain Controllers,OU=Servers,OU=Systems,DC=somedomain,DC=local
```

```
changetype: add
description: Container for Domain controllers
objectClass: organizationalUnit
ou: Domain Controllers

dn: OU=Application,OU=Servers,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for application servers like DB2, Notes,...
objectClass: organizationalUnit
ou: Application

dn: OU=Workstations,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Client computer objects
objectClass: organizationalUnit
ou: Workstations

dn: OU=Notebooks,OU=Workstations,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for notebook computer objects
objectClass: organizationalUnit
ou: Notebooks

dn: OU=Desktops,OU=Workstations,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for standard desktop computer objects
objectClass: organizationalUnit
ou: Desktops

dn: OU=Special,OU=Workstations,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for non-standard workstation objects
objectClass: organizationalUnit
ou: Special

dn: OU=Central,DC=somedomain,DC=local
changetype: add
description: Centrally defined user and group objects
objectClass: organizationalUnit
ou: Central

dn: OU=Users,OU=Central,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Users

dn: OU=FTP,OU=Users,OU=Central,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
```

```
ou: FTP

dn: OU=NFS,OU=Users,OU=Central,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: NFS

dn: OU=Administrators,OU=Users,OU=Central,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Administrators

dn: OU=Services,OU=Users,OU=Central,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Services

dn: OU=Groups,OU=Central,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Groups
```

At this point, the directory is now accessible through the root distinguished name, and the basic enterprise organizational objects are set up in the directory. The directory is now ready for migrating OS/2 systems information.

6.1.4 LDAP directory maintenance

The package `SMBLDAP-TOOL` is commonly used for the integration of Samba and the proper maintenance of the LDAP structure. This package is available at: <http://www.samba.idealx.org/>

As of the writing of this redbook, the `SMBLDAP` tools package was not updated to the new LDAP schema changes that were introduced in Samba 3.

LDAP object management

During the operation of Samba, the automatic creation of objects (user objects for computers auto-joining the domain, for example) are created in the root context of the LDAP tree where the Samba server is directed. As a result, these entries do not conform to the LDAP organizational structure proposed in this redbook. It is recommended that the use of the `SMBLDAP` tools as well as supporting scripts and processes be utilized to maintain this organization. The `SMBLDAP` tools are well suited for enterprise-specific needs, and the scripts are easily customizable.

6.2 Migrating the OS/2 domain

The migration of the OS/2 domain data consists of two core steps:

1. Import branch-unique organizational units into the directory.
2. Configure the Samba server parameters for the domain name.

6.2.1 Organizational Units for each branch

The following file needs to be imported into the directory, appropriately modified for each branch, in preparation for the migration of the domain information.

Example 6-3 Example branchou.ldif file

```
dn: OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Branch1

dn: OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Users

dn: OU=Groups,OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Groups

dn: OU=Application,OU=Groups,OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
description: Container for groups assigning applications to users
objectClass: organizationalUnit
ou: Application

dn: OU=Access,OU=Groups,OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
description: Container for groups granting access to resources
objectClass: organizationalUnit
ou: Access

dn: OU=Print,OU=Groups,OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
description: Groups for granting access to printer queues
objectClass: organizationalUnit
ou: Print

dn: OU=Organization,OU=Groups,OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
```



```
description: Groups defining organisational membership of users (useable as
DL)
objectClass: organizationalUnit
ou: Organization
```

This enters the branch-unique organizational units, and readies them as targets for migrating OS/2 Server domain information.

6.2.2 Overview of OS/2 domain mapping to Samba

The migration of the domain name from the OS/2 Warp Server to the Samba server is a simple initial step. This involves the configuration of the Samba server to report and respond to the domain name currently in use on the OS/2 domain servers.

6.2.3 Samba domain configuration

As configured earlier in this book, the branch server is a primary domain controller. To configure the domain for which Samba is the primary domain controller, modify the `/etc/samba/smb.conf` file as follows:

```
workgroup = {os2DomainName}
```

Changing the value of this entry to the source OS/2 domain name will make the target Samba server become the primary domain controller for this domain.

Note: Be aware that running two PDCs for the same domain name on the same network segment and the same protocols will likely cause problems. Either the Samba services or the OS/2 LAN Services will need to be disabled or shut down before activating this change.

6.3 Migrating server definitions

The migration of the OS/2 Server definition is rather simple consisting of one core step:

- ▶ Configure the Samba server parameters for the server name.

6.3.1 Overview of OS/2 Server name mapping to Samba

The migration of the server names from the OS/2 Servers to the Samba server is another simple step. This involves the configuration of the Samba server to report and respond to the server NetBIOS names currently in use on the OS/2 Servers.

6.3.2 Additional OS/2 Server services

The OS/2 Warp Server services provide a variety of functions. As part of the migration, there are a set of other server-related functions that might require migration also. The following is a brief overview of these functions, and possible migration solutions and considerations.

Replicator service

The Replicator service provides the ability to synchronize a directory structure from one OS/2 Server to another. Samba provides no function of this type, but the Linux operating system does in the form of *rcp* and other utilities. If the services of Replicator are used, explore these options for implementation in the final solution.

Time source service

The Time source service provides the ability to provide a consistent calendar time to the workstations during logon to the OS/2 domain. Samba provides the ability to serve time data such as with the **NET TIME** command. The configuration of this is time zone is sensitive for Samba. Also, a cross-platform standard time solution is NTP, and this is recommended as a starting point for consideration.

NetRun service

The NetRun service provides the ability to run a command administratively on the OS/2 Server from a remote workstation. This feature is limited in functionality, but very useful to many OS/2 customers. Samba does not provide a similar service, but the Linux operating system does in the forms of *rsh*, *ssh*, and other utilities. If the services of NetRun are used, explore these options for implementation in the final solution. Due to increased security capabilities, *ssh* is recommended as the starting point for examination.

6.3.3 Configuring Samba server name

To configure the server name that Samba responds to, modify the `/etc/samba/smb.conf` file as follows:

```
netbios name = {os2ServerName}
```

Changing the value of this entry to the source OS/2 Server name will make the target Samba server report this server name upon Samba restart.

Note: Be aware that running two systems with the same server name on the same network segment and same protocols will likely cause problems. Either the Samba services or the OS/2 LAN services will need to be disabled or shut down before activating this change.

Multiple NetBIOS names

If the source OS/2 Server is using the feature of responding to multiple NetBIOS names, the following parameter can be configured for Samba to respond to additional NetBIOS names:

```
netbios aliases = {netBIOSName1} {netBIOSName2} {...}
```

Multiple names are separated using a space.

6.4 Migrating groups

Migration of groups is critical to the proper operation and management of the target system.

6.4.1 Overview of OS/2 group mappings to Samba

The OS/2 group is a domain object with a set of attributes. The core attributes and concepts map to Linux and Samba users, but are used a bit differently than that of the OS/2 domain servers. Also, Linux and Samba add attributes for integration into these services. The following table overviews the group object attributes and the mappings we used.

Table 6-1 Transformation matrix for Samba group objects

LDAP attribute	Source OS/2 attribute	Transition steps
dn	GROUP.NAME	The OS/2 attribute is formatted in an LDAP style distinguished name including the complete path.
cn	GROUP.NAME	The OS/2 attribute is formatted in an LDAP style distinguished name including the complete path.
gidNumber		A unique number assigned to each group from the transform.group file
description	COMMENT	Some additional description may be available in the COMMENT field, so we use this as a best match.
memberUID	USER.NAME	The OS/2 user ID(s) as members of this group.

These mappings provide the basic functionality required. Note the gidNumber attribute is now required and is a unique integer for the Linux platform in the context of groups.

6.4.2 Preparation for migration

In our migration example, LSMT writes the group definitions to a file named getgrps1.log.

Example 6-4 OS/2 group definitions from example OS/2 domain (getgrps1.log)

```
OPT;NAME          ;COMMENT          ;
;ADMINS           ;                    ;
;BOOKREAD         ;                    ;
;BOOKWRITE        ;                    ;
;GROUPID          ;Default Group ID  ;
;GUESTS           ;                    ;
;LOCAL            ;                    ;
;PRINTER          ;Printer Group    ;
;SERVERS          ;System ID - Server ;
;TRANSITION       ;                    ;
;USERS            ;                    ;
```

Tip: At the time of migration, it is recommended to review the current design of group usage in your domain. You may change the naming conventions, helping you to identify the purpose of groups more easily. You can use groups more extensively because the OS/2 LAN Server restriction to 254 groups is not a limitation for Samba servers. Because LSMT provides the data in an ASCII format that you can modify very easy, you can also add new groups rather than only migrating existing ones.

The basic approach to migrating the groups from the source OS/2 domain to the target Samba server is to parse the LSMT output file containing the OS/2 group definitions and produce an LDAP LDIF file for importing into the directory.

The design of the directory included four organizational units (OU) for group objects:

- | | |
|----------------|---|
| OU=Access | The container holding groups objects that grant access to resources directories and files. |
| OU=Print | The container holding groups objects that specify access rules to printer objects. |
| OU=Application | The container holding groups that grant access to published applications (for example, Citrix Metaframe). |

OU=Organization The container holding groups defined for the membership to a particularly group of persons in an enterprise view. These include distribution lists, project teams, or workgroups.

To map the given groups, the **setgroups.cmd** script uses the first column (OPT) to map them into the specific context. The following table describes this mapping:

Table 6-2 Mapping group definitions using the OPT column

OPT	Action
<blank>	This line will be ignored in the transformation process. With this option you do not have to remove unwanted groups from the export file.
A	This group definition is treated as an access group. This group is migrated to the OU=Access.
O	This group definition describes an organizational Group. It is migrated to the OU=Organization.
P	This group definition describes a group granting access to print queues. It is migrated to the OU=Print.
X	This group definition is treated as an application group. This group is migrated to the OU=Application.

Taking the given example, we modified it and added one new group that we need in the Samba LDAP directory.

Example 6-5 Example OS/2 groups domain group mapping modifications

```

OPT;NAME           ;COMMENT           ;
  ;ADMINS           ;                   ;
A ;BOOKREAD        ;                   ;
A ;BOOKWRITE       ;                   ;
  ;GROUPID          ;Default Group ID  ;
  ;GUESTS           ;                   ;
  ;LOCAL            ;                   ;
P ;PRINTER          ;Printer Group     ;
  ;SERVERS          ;System ID - Server ;
A ;TRANSITION      ;                   ;
  ;USERS            ;                   ;
O ;BRANCH1         ;All users of branch 1 ;

```

The sample **setgroups.cmd** command file converts the modified LSMT output into an LDIF file of data for the directory. This command file uses a transform.group file, which is to be created and provided by the enterprise to

properly define the group ID numbers for the group names. An example of this mapping follows:

Example 6-6 Example transform.group file for group LDIF creation

```
ADMINS 1000
BOOKREAD 1001
BOOKWRITE 1002
GROUPID 1003
GUESTS 1004
LOCAL 1005
PRINTER 1006
SERVERS 1007
TRANSITION 1008
```

Note that the transform.group file contains multiple lines consisting of the OS/2 group name and the assigned group ID number. The command issued to produce the LDIF file is:

```
setgroups.cmd smb getgrps1.log setsmbgroups.ldif branchID transform.group
```

The options to the **setgroups.cmd** files are as follows:

- ▶ **smb**: Specifies that this invocation is to produce SMB targeted output
- ▶ **getgrps1.log**: The group output from LSMT
- ▶ **setsmbgroups.ldif**: The output file to produce
- ▶ **branchID**: The ID of the branch
- ▶ **transform.group**: The group name to group the ID number mapping file

The following is an example LDIF file for the sample OS/2 domain's group objects for importing:

Example 6-7 Example setsmbgroups.ldif output file

```
dn:
CN=ADMINS,OU=Organization,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: ADMINS
gidNumber: 1000
objectClass: group

dn:
CN=BOOKREAD,OU=Access,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: BOOKREAD
gidNumber: 1001
objectClass: group
```

```
dn:
CN=BOOKWRITE,OU=Access,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: BOOKWRITE
gidNumber: 1002
objectClass: group

dn:
CN=GROUPID,OU=Application,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: GROUPID
gidNumber: 1003
objectClass: group
description: Default Group ID

dn:
CN=GUESTS,OU=Application,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: GUESTS
gidNumber: 1004
objectClass: group

dn:
CN=LOCAL,OU=Organization,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: LOCAL
gidNumber: 1005
objectClass: group

dn:
CN=PRINTER,OU=Print,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: PRINTER
gidNumber: 1006
objectClass: group
description: Printer Group

dn:
CN=SERVERS,OU=Access,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: SERVERS
gidNumber: 1007
objectClass: group
description: System ID - Server
```

```
dn:  
CN=TRANSITION,OU=Organization,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,  
DC=local  
changetype: add  
cn: TRANSITION  
gidNumber: 1008  
objectClass: group
```

Importing this LDIF file will create the groups in the LDAP directory.

6.4.3 Steps to follow for groups

To perform the migration of group definitions from OS/2 to Samba with an LDAP directory, follow these steps:

1. Create the export file `getgrps1.log` using the LSMT.
2. Modify the entries and add a A, O, P, or X in the column OPT for the groups you want to transfer to the target domain.
3. Change descriptions, group names, or add additional groups you need in Samba and LDAP for you branch.
4. Run the command `setgroups.cmd` with the following parameters:

```
setgroups.cmd smb getgrps1.log setsmbgroups.ldif branchID transform.group
```
5. Import the group definitions into the directory using the `ldapmodify` command.

6.5 Migrating users and passwords

Migration of users is core to the operation of the target system. Users will be stored in the centralized LDAP directory in the example migration scenario. New with Samba 3.0, the `sambaSamAccount` LDAP auxiliary object is now the preferred object for the LDAP interaction.

6.5.1 Overview of OS/2 user mapping to Samba

The OS/2 user is a rich domain object with many attributes and related capabilities. The core attributes and concepts map to Linux and Samba users, but many of the related features of the OS/2 user do not map. The following section overviews the user object attributes and the mapping we used:

Table 6-3 Samba LDAP attributes and OS/2 attribute mappings with details

Target: LDAP attribute	Source: OS/2 attribute	Transition details
dn	NAME	The OS/2 attribute is formatted in an LDAP style distinguished name including the complete path.
uid	NAME	The OS/2 attribute is formatted in an LDAP style distinguished name including the complete path.
objectClass		
cn	NAME	The OS/2 attribute is formatted in an LDAP style distinguished name including the complete path.
gidNumber		The number was hard coded to 100.
homeDirectory	HOME_DIR	This attribute defines the mount point assigned to the home directory for Linux Clients.
uidNumber		This is a number that is unique for each user. In our case we used their employee numbers to define their uidNumber.
sambaHomePath	HOME_DIR	This attribute defines the server path assigned to the home directory for Linux Clients.
sambaHomeDrive	HOME_DIR	This attribute defines the drive letter assigned to the home directory for other clients. We can map it directly to the first character of the OS/2 HOME_DIR attribute.
sambaLoginScript		This value was hard coded as login.cmd.
sambaProfilePath		Specifies a path to the user's profile. This value can be a null string, a local absolute path, or a UNC path.
description	USR_COMMENT	Some additional description may be available in the USR_COMMENT field, so we use this as the best match.

Target: LDAP attribute	Source: OS/2 attribute	Transition details
displayName	COMMENT	There is no one-to-one correspondence for this attribute. If the COMMENT attribute was used in a standard way, for example, to specify the full name of a user, then it could be parsed and used to display the user's first name, for instance.
sambaLMPassw ord		The LANMAN password 16-byte hash stored as a character representation of a hexadecimal string was extracted from the GETPWD.LOG file that was created by LSMT.

The following lists attributes not used for the example migration:

Table 6-4 OS/2 user attributes not directly mapped to Samba

OS/2 Attribute	Transition steps
PASSWORD_AGE	Samba does not currently use this OS/2 user attribute data.
PRIV	Samba does not currently use this OS/2 user attribute data.
FLAGS	Samba does not currently use this OS/2 user attribute data.
SCRIPT_PATH	The attribute could be used at the sambaLogonScript, but for our model it was not used.
AUTH_FLAGS	Samba maps this functionality with acctFlags. It was left out of the script as it was not a required attribute and the user needs to decide on how to use the Samba attributes.
FULL_NAME	Samba does not current use this OS/2 user attribute data.
PARMS	Samba does not currently use this OS/2 user attribute data.
WORSTATION	Samba does not currently use this OS/2 user attribute data.
LAST_LOGON	Samba does not currently use this OS/2 user attribute data.
LAST_LOGOFF	Samba does not currently use this OS/2 user attribute data.
ACCT_EXPIRES	Samba does not currently use this OS/2 user attribute data.
MAX_STORAGE	Samba does not currently use this OS/2 user attribute data.
RESTRICTED_HOU RS	Samba does not currently use this OS/2 user attribute data.
LOGON_HOURS	Samba does not currently use this OS/2 user attribute data.

OS/2 Attribute	Transition steps
BAD_PW_COUNT	Samba does not currently use this OS/2 user attribute data.
NUM_LOGONS	Samba does not currently use this OS/2 user attribute data.
LOGON_SERVER	Samba does not currently use this OS/2 user attribute data.
COUNTRY_CODE	Samba does not currently use this OS/2 user attribute data.
CODE_PAGE	Samba does not currently use this OS/2 user attribute data.

The above attributes, which currently do not directly map into the Samba user object, or are not currently used by Samba, can be implemented with customer scripting solutions enhancing the Samba deployment. As an example, the FULL_NAME attribute of the OS/2 user object can be mapped to an attribute on a related schema object in the LDAP directory. Samba should coexist with these types of directory extensions without a problem.

6.5.2 Preparation for migration

Six users were marked for input into the LDAP directory as identified as follows:

Example 6-8 Example LSMT output for users, modified for use with setusers.cmd

```

OPT;NAME      ;PASSWORD;PASSWORD_AGE;PRIV      ;HOME_DIR
; ...
A ;ANDREI     ;****      ;870047      ;User      ;U:\PDC\E$\LANHOMES\ANDREI
; ...
;BDC         ;****      ;162218     ;User      ;-none-
; ...
;GUEST       ;****      ;1375390    ;Guest     ;-none-
; ...
A ;LEIF       ;****      ;1372736    ;User      ;U:\PDC\E$\LANHOMES\LEIF
; ...
A ;MARC       ;****      ;1372735    ;User      ;U:\PDC\E$\LANHOMES\MARC
; ...
;MICHAEL     ;****      ;8652       ;User      ;H:\LNXSLES\MICHAEL
; ...
;MIKE        ;****      ;150749     ;User      ;R:\PDC\C$\HOME\MIKE
; ...
A ;OLIVER     ;****      ;1372735    ;User      ;U:\PDC\E$\LANHOMES\OLIVER
; ...
;PDC         ;****      ;1375391    ;User      ;-none-
; ...
A ;RICHARD    ;****      ;1372735    ;User
;U:\PDC\E$\LANHOMES\RICHARD ; ...
;USERID      ;****      ;426648862  ;Administrator;-none-
; ...

```

```
A ;WYNAND ;**** ;242169 ;User ;U:\PDC\E$\LANHOMES\WYNAND  
; ...
```

The sample **setusers.cmd** command file converts the modified LSMT output into an LDIF file of data for the directory. This command file uses a **transform.user** file, which is created and provided by the enterprise to properly define the group ID numbers for group names. An example of this mapping follows:

Example 6-9 Example transform.user file for group LDIF creation

```
ANDREI 8768  
LEIF 987987  
MARC 1201  
OLIVER 234443  
RICHARD 865797961  
WYNAND 4294967293
```

Note that the **transform.user** file contains multiple lines consisting of the OS/2 user ID and the assigned user ID number. The command issued to produce the LDIF file is:

```
setusers.cmd smb getusers.log setsmbusers.ldif branchID getpwd.log  
transform.user
```

The options to the **setusers.cmd** files are as follows:

- ▶ **smb**: Specifies that this invocation is to produce SMB targeted output.
- ▶ **getusers.log**: The user output from LSMT
- ▶ **setsmbusers.ldif**: The output file to produce
- ▶ **branchID**: The ID of the branch
- ▶ **getpwd.log**: The password output from LSMT
- ▶ **transform.user**: The user ID to user ID number mapping file

Note that the **setusers.cmd** file takes as a parameter the **getpwd.log** output from LSMT. IBM OS/2 Warp Server uses an encrypted hashed value of a user's password. This is created by taking the user's plaintext password, capitalizing it, and either truncating to 14 bytes or padding to 14 bytes with null bytes. This 14 byte value is used as two 56-bit DES keys to encrypt a "magic" 8 byte value, forming a 16 byte value, which is stored by the server and client. This hashed password is part of the user object and stored in the accounts database, *NET.ACC*. Windows NT encryption consists of doing an MD4 hash on a Unicode version of the user's password. This also produces a 16 byte hash value that is non-reversible. This hash value, exported from the OS/2 domain for each user ID, is imported directly into the LDAP directory for each user.

The following is an example LDIF file for importing the sample OS/2 domain's user objects:

Example 6-10 Example setsmbusers.ldif output file

```
dn: CN=ANDREI,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
uid: ANDREI
userid: ANDREI
objectClass: sambaSamAccount
objectClass: account
objectClass: posixAccount
cn: ANDREI
gidNumber: 100
homeDirectory: /home/ANDREI
uidNumber: 8768
sambaSID: S-1-5-21-0123456789-0123456789-0123456789-8768
sambaHomePath: \\PDC\ANDREI
sambaHomeDrive: U:
sambaLogonScript: logon.cmd
sambaProfilePath:
displayName: Andrei_Vlad
sambaLMPassword:CD017457761C8B05AAD3B435B51404EE
```

```
dn: CN=LEIF,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
uid: LEIF
userid: LEIF
objectClass: sambaSamAccount
objectClass: account
objectClass: posixAccount
cn: LEIF
gidNumber: 100
homeDirectory: /home/LEIF
uidNumber: 987987
sambaSID: S-1-5-21-0123456789-0123456789-0123456789-987987
sambaHomePath: \\PDC\LEIF
sambaHomeDrive: U:
sambaLogonScript: logon.cmd
sambaProfilePath:
displayName: Leif_Braeuer
sambaLMPassword:32DD5DAB4DC507A4AAD3B435B51404EE
```

```
dn: CN=MARC,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
uid: MARC
userid: MARC
objectClass: sambaSamAccount
objectClass: account
objectClass: posixAccount
cn: MARC
gidNumber: 100
```

homeDirectory: /home/MARC
uidNumber: 1201
sambaSID: S-1-5-21-0123456789-0123456789-0123456789-1201
sambaHomePath: \\PDC\MARC
sambaHomeDrive: U:
sambaLogonScript: logon.cmd
sambaProfilePath:
displayName: Marc_Schneider
sambaLMPassword:30C38B207E9B137BAAD3B435B51404EE

dn: CN=OLIVER,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
uid: OLIVER
userid: OLIVER
objectClass: sambaSamAccount
objectClass: account
objectClass: posixAccount
cn: OLIVER
gidNumber: 100
homeDirectory: /home/OLIVER
uidNumber: 234443
sambaSID: S-1-5-21-0123456789-0123456789-0123456789-234443
sambaHomePath: \\PDC\OLIVER
sambaHomeDrive: U:
sambaLogonScript: logon.cmd
sambaProfilePath:
displayName: Oliver_Mark
sambaLMPassword:617093781CC21A60AAD3B435B51404EE

dn: CN=RICHARD,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
uid: RICHARD
userid: RICHARD
objectClass: sambaSamAccount
objectClass: account
objectClass: posixAccount
cn: RICHARD
gidNumber: 100
homeDirectory: /home/RICHARD
uidNumber: 865797961
sambaSID: S-1-5-21-0123456789-0123456789-0123456789-865797961
sambaHomePath: \\PDC\RICHARD
sambaHomeDrive: U:
sambaLogonScript: logon.cmd
sambaProfilePath:
displayName: Richard_Spurlock
sambaLMPassword:E4301A7CD8FDD1ECAAD3B435B51404EE

dn: CN=WYNAND,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local

```
changetype: add
uid: WYNAND
userid: WYNAND
objectClass: sambaSamAccount
objectClass: account
objectClass: posixAccount
cn: WYNAND
gidNumber: 100
homeDirectory: /home/WYNAND
uidNumber: 4294967293
sambaSID: S-1-5-21-0123456789-0123456789-0123456789-4294967293
sambaHomePath: \\PDC\WYNAND
sambaHomeDrive: U:
sambaLogonScript: logon.cmd
sambaProfilePath:
description: Standard Bank User
displayName: Wynand_Pretorius
sambaLMPassword:D851BE004D8658DFAAD3B435B51404EE
```

Importing this LDIF file will create the users in the LDAP directory.

6.5.3 Group membership

The groups and users have been created in the LDAP directory and the next step is to add the user IDs to the groups.

The groups have already been created in our migration example. The next step is to add the user ID members to each group. This is accomplished by adding the user ID as a member of the LDAP group object, and this is accomplished through LDIF files.

Using the LSMT generated output files, modify, add, or delete entries, and use it as the input file for the transition script **setgrpmem.cmd**

Important: LSMT adds three columns for the groups USERS, GUESTS, and ADMINS to the export file. These groups are not normal groups as you cannot add users to these groups. Any changes to these columns are ignored within the migration.

To migrate the membership to the LDAP directory, again set an A in the first column, and optionally modify the appropriate column. In case groups have been added to the file, add additional columns to the file and mark the membership as required.

Tip: Remove the columns for the groups ADMINS, GUESTS, USERS and all groups not migrated. Otherwise, the resulting LDIF file generates an error because a group cannot be found.

In contrast to OS/2, LDAP needs the distinguished name for the group. OS/2 only supplies the common name. This is the reason for creating the group lookup database group-d.csv, which we created as part of the group migration step. Having the modified LSMT file and this database ready, you can start creating the LDIF file for group membership using the following command:

```
setgrpmem smb getgrps2.log setgroupmembers.ldif branchID
```

The used input and generated output files are shown in the following examples:

Example 6-11 Modified getgrps2.log ready to import

```
* Do not modify a user from the ADMINS, GUEST, SERVERS or USERS groups *
OPT;USERS
;BOOKREAD;BOOKWRITE;GROUPID;LOCAL;PRINTER;SERVERS;TRANSITION;BRANCH1;
A ;ANDREI ; X ; ; ; ; ; ; X ;
X ;
;BDC ; ; ; ; ; ; X ; ;
;
;GUEST ; ; ; ; ; ; ; ;
;
A ;LEIF ; X ; ; ; ; X ; ; X ;
X ;
A ;MARC ; X ; ; ; ; X ; ; X ;
X ;
A ;OLIVER ; ; X ; ; ; ; X ; ; X ;
X ;
;PDC ; ; ; ; ; ; X ; ;
;
A ;RICHARD ; X ; ; ; ; ; ; X ;
X ;
;USERID ; ; ; X ; ; ; ; ;
;
A ;WYNAND ; X ; ; ; ; ; ; X ;
X ;
```

The following example is the group-db.csv output for our migration scenario.

Example 6-12 Group lookup database group-db.csv

```
BOOKREAD;CN=BOOKREAD,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
BOOKWRITE;CN=BOOKWRITE,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
PRINTER;CN=PRINTER,OU=Print,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
TRANSITION;CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
BRANCH1;CN=BRANCH1,OU=Organization,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
```

The following example is the resulting output file for the group membership import step:

Example 6-13 Resulting setgroupmembers.ldif

```
dn:
CN=BOOKREAD,OU=Access,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
memberUID: ANDREI
-
dn:
CN=TRANSITION,OU=Organization,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,
DC=local
changetype: modify
add: member
memberUID: ANDREI
-
dn:
CN=BOOKREAD,OU=Access,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
memberUID: LEIF
-
dn:
CN=PRINTER,OU=Print,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
memberUID: LEIF
-
dn:
CN=TRANSITION,OU=Organization,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,
DC=local
changetype: modify
add: member
memberUID: LEIF
-
```

```

dn:
CN=BOOKREAD,OU=Access,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
memberUID: MARC
-
dn:
CN=PRINTER,OU=Print,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
memberUID: MARC
-
dn:
CN=TRANSITION,OU=Organization,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,
DC=local
changetype: modify
add: member
memberUID: MARC
-
dn:
CN=BOOKWRITE,OU=Access,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
memberUID: OLIVER
-
dn:
CN=PRINTER,OU=Print,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
memberUID: OLIVER
-
dn:
CN=TRANSITION,OU=Organization,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,
DC=local
changetype: modify
add: member
memberUID: OLIVER
-
dn:
CN=BOOKREAD,OU=Access,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
memberUID: RICHARD
-
dn:
CN=TRANSITION,OU=Organization,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,
DC=local
changetype: modify

```

```

add: member
memberUID: RICHARD
-
dn:
CN=BOOKREAD,OU=Access,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
memberUID: WYNAND
-
dn:
CN=TRANSITION,OU=Organization,OU=Groups,OU=branch1,OU=Branch,DC=somedomain,
DC=local
changetype: modify
add: member
memberUID: WYNAND
-

```

This LDIF file can be imported to add the users to the desired groups using the `ldapmodify` command.

6.5.4 Logon assignments

IBM OS/2 Warp Server domains use the feature of a domain controller database (DCDB) to store alias and logon assignment information. Taking a closer look at this database reveals a directory tree shared by every domain controller running the DCDB-replicator. Clients are able to access this database through the share `IBMLAN$`. Samba does not implement this resource or functionality. There are several concepts to do this in a Samba environment:

- ▶ Copy the DCDB subdirectory to each Samba server to provide a “read-only” backward compatibility for OS/2 clients.
- ▶ Migrate from drive letters and use UNC path names only, and let the user connect his drives using the Windows Explorer and persistent connection.
- ▶ Provide all resources in a distributed file system protecting the branches with discreet group based ACLs, preventing users to see forbidden resources.
- ▶ Use the general logon script that calls a user specific routine for performing assignments.

Providing Logon Services for OS/2 clients

When logging onto a Samba domain, an OS/2 client requires certain information to be configured in order to avoid error messages being presented to the user:

- ▶ The name of the primary domain controller for the domain
- ▶ A home directory with a certain syntax the OS/2 clients can interpret

- ▶ Access to the DCDB to process the logon assignments and the optional PROFILE.CMD

The first requirement is usually provided by the Samba server.

The second requirement cannot be fulfilled for both environments. OS/2 and Windows NT use a different syntax when defining the home directory of a user. When an OS/2 client logs on to a Samba domain with a user account having a home directory defined, they will likely receive the following error message:

```
NET8191: Your home directory could not be set up
```

To avoid this, consider moving the assignment of a users' home directory to a logon script.

If for some reason OS/2 clients still need access to some kind of DCDB, access to these files and PROFILE.CMD can be provided through the following steps:

1. Create a directory on a Samba domain controller.
2. Share this directory as IBMLAN\$ giving all domain users read permissions by modifying the /etc/samba/smb.conf configuration file.
3. Copy the directory x:\IBMLAN\DCDB of the OS/2 primary domain controller into this directory:

```
xcopy \\pdc\ibmlan$\dcdb \\sambaserver\ibmlan$\dcdb /h /o /t /s /e /r /v
```

6.5.5 Steps to follow

To perform the migration of user definitions from OS/2 to Samba and LDAP, follow these steps:

1. Get the export file getusers.log using the LSMT.
2. Modify the entries and add an A in the column OPT for the users you want to transfer to the target domain.
3. Change descriptions, names, privilege, or other attributes as you need them in the LDAP directory for your branch.
4. Run the command **setusers.cmd** with the following parameters:

```
setusers.cmd smb getusers.log setsmbusers.ldif branchID getpwd.log transform.user
```

5. Import the user definitions into the LDAP directory with **ldapmodify**

At this step, your user objects with passwords are migrated to the target domain without any group memberships or logon assignments.

6. Get the export file `getgrps2.log` using the LSMT as described in Chapter 3, “Starting the OS/2 Server migration” on page 63.
7. Modify the entries and add an A in the column OPT for the users’ group memberships you want to transfer to the target domain.
8. Change memberships or add new group as you need them in the LDAP directory for your branch.
9. Get the `group-db.csv` database the script it needs to translate OS/2 group names to LDAP names.
10. Run the command `setgrpmem.cmd` with the following parameters:

```
setgrpmem smb getgrps2.log setgroupmembers.ldif branchID
```
11. Import the user definitions into the LDAP directory with the `ldapmodify` command.
12. Complete the logon assignments process from Chapter 6.5.4, “Logon assignments” on page 217.

6.6 Migrating directories and access controls

Following the user and group migration, the directories and access controls are logically next. The process of migrating the directories and access controls consists of the following steps:

1. Define the shares and the associated share-point access controls.
2. Create the directories for the shares on the Linux system.
3. Copy the data from the OS/2 aliases to the Samba shares.

6.6.1 Overview of access controls with Samba

The access control models of OS/2 and Samba are fundamentally different. Samba V3.0 provides four key facilities. For the example migration, share level access controls (the simplest to map) will be covered.

Restriction: The migration example provides for access control at the share level only. File or subdirectory level access controls are not covered here.

Information regarding access controls in Samba can be found in the Samba Project Documentation document dated June 6, 2003 (see <http://de.samba.org/samba/devel/docs/html/>). Refer to Chapter 13., “File, Directory and Share Access Controls” for additional details and information on additional options.

A detailed exploration of the options herein is out of the scope of this redbook, and the current state of the art is ever changing. The reader is encouraged to explore the state of Samba and Linux-level access controls for the distribution and kernel levels being deployed on the branch servers.

For the example migration, the following access control mappings from OS/2 to Samba were applied:

<u>OS/2 permission</u>	<u>Samba permission</u>
READ	Read
WRITE	Write
CREATE	Write
DELETE	Write
ATTRIBUTE	Write
PERMISSION	Admin
EXECUTE	Read

These are applied to the share definitions through the readlist, writelist, and adminlist values for each share.

6.6.2 Overview of Samba directory shares

Each directory share in Samba is defined by a section in the `/etc/samba/smb.conf` configuration file. Samba provides a wide range of configuration options for defining and tuning shares. The following is the basic share definition that is used for the migration scenario:

Example 6-14 Example directory share definition for `/etc/samba/smb.conf`

```
[share_name]
readlist = readUserID @readGroupID
writelist = writeUserID @writeGroupID
adminlist = adminUserID @adminGroupID
comment = the share comment
path = /shares/share_name
directory mask = 0770
dos file mode = 0770
nt acl support = no
security mask = 0770
case sensitive = no
public = no
writeable = yes
printable = no
```

The following defines each entry for a share, and the assigned data for the basic migration scenario:

Table 6-5 Samba directory share section keywords and values

Samba share entry	Entry description
[share_name]	The share section entry header. All entries between this entry and the next [...] entry is considered to be configuration details for this share.
readlist = readUserID @readGroupID	Defines the users and groups (groups are preceded by the @ character) which can read the data shared by this resource.
writelist = writeUserID @writeGroupID	Defines the users and groups (groups are preceded by the @ character) which can write the data shared by this resource.
adminlist = adminUserID @adminGroupID	Defines the users and groups (groups are preceded by the @ character) which have unrestricted access to the data shared by this resource.
comment = the share comment	A description of this share viewable by users.
path = /shares/share_name	The path to the entry point of the share data on the Linux server file system.
directory mask = 0770	The octal user, group, and other permissions mask used when creating UNIX directories.
dos file mode = 0770	The octal user, group, and other permissions mask used to allow a user who has write access to the file to modify the permissions on the file.
nt acl support = no	Specifies that the Samba daemon will not attempt to map UNIX permissions into Windows NT access control data.
security mask = 0770	The octal user, group, and other permissions mask used to control the UNIX permission bits modified when a Windows NT client is manipulating the UNIX permissions on a file.
case sensitive = no	Specifies that all file name lookups will be case insensitive.
public = no	Specifies that the share is not publicly accessible as a guest without a password.
writable = yes	Specifies that users may write and modify files on the shared resource.
printable = no	Specifies that this resource is not a spooler resource for printing.

Each share in the migration scenario will use these parameters. As Samba provides a rich selection of options for configuring and tuning shared resources, it is recommended that the Samba Project Documentation document the manual page for the smb.conf file to be reviewed.

6.6.3 Create the share point directories

The share points for the Samba shares can be anywhere within the file system on the Linux server. For the migration example in this chapter, the shares for Samba's use are set up as follows:

```
/shares
/shares/netlogon
/shares/profile
/shares/os2alias1
/shares/os2alias2
/shares/...
/shares/os2aliasX
```

The /shares directory is a base directory for the Samba shares structure. With Linux file systems, the options for configuring the file systems and mount points are rather flexible. The /shares directory can be a base where other directories from around the file systems are mounted or linked to. This provides a convenient location to centrally access directory resources around the system for Samba's usage.

Each of the directories required for migrated shares should be created before defining the shares to Samba to make certain that the activated share is valid.

6.6.4 Define shares and access controls

Migration of aliases is central to the function of the target system. OS/2 aliases will be converted to Samba shares during the migration.

The definition of shares in Samba is controlled through the Samba configuration file /etc/samba/smb.conf.

For the example migration, two LSMT output files are used for this migration step:

Example 6-15 Example of the LSMT output for the aliases (three wrapping lines)

```
OPT;NAME      ;TYPE      ;SERVER      ;PATH
;REMARK
;MAXUSES;QUEUE      ;PRIORITY;DEVICE_POOL ;
;BOOK      ;Files ;PDC      ;F:\BOOK
;
;startup;65535 ;Unknown ;Unknown ;Unknown ; ;Within Domain;At server
```



```

;LANSHARE;Files ;BDC ;E:\LANSHARE
;
;Within Domain;At server
startup;65535 ;Unknown ;Unknown ;Unknown ;

```

Example 6-16 Example of the LSMT output for the alias access controls (four wrapping lines)

```

* List of all ACLs of existing Aliases, allowed Options U=update D=delete
OPT;ALIAS ;AUDIT ;ADMINS ;BOOKREAD;BOOKWRITE;GROUPID ;GUESTS ;LOCAL
;PRINTER ;SERVERS ;TRANSITION;USERS ;ANDREI ;AUSRES26;BDC ;GUEST
;LEIF ;MARC ;OLIVER ;PDC ;RICHARD ;USERID ;WYNAND ;
A ;BOOK ;-none-; ; RG ; RWC DAG ; ; ;
; ; ; ; ; ; ; ; ;
; ; ; ; ; ; ; ; ;
A ;LANSHARE ;-none-; ; ; ; ; ; ;
; ; ; RWCXDAPG ; ; ; ; ; ; ;
; ; ; ; ; ; ; ; ;
P ;PRINT_Q ;-none-; ; ; ; ; ; ;
; CPG ; ; ; ; ; ; ; ; ;
; ; ; ; ; ; ; ; ;

```

These two LSMT output files are used to produce a resulting shell script to modify the Samba configuration file `/etc/samba/smb.conf`. The command issued to produce the LDIF file is:

```
setsmbdiralias.cmd getacl.log setDirAlias.sh getalias.log
```

The options to the `setsmbdiralias.cmd` files are as follows:

- ▶ `getacl.log`: The access control output from LSMT
- ▶ `setdiralias.sh`: The output file which modifies the `/etc/samba/smb.conf`
- ▶ `getalias.log`: The alias output from LSMT

The following is an example shell script file for the sample OS/2 alias to Samba share mapping for modifying the Samba configuration file `/etc/samba/smb.conf`:

Example 6-17 Example `setdiralias.sh` output file

```

perl modini.pl /etc/samba/smb.conf SREMOVE "[BOOK]"
perl modini.pl /etc/samba/smb.conf SADD "[BOOK]"
perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "readlist" "@BOOKREAD
@BOOKWRITE"
perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "writelst" "@BOOKWRITE"
perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "adminlist" ""
perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "comment" ""
perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "path" "/shares/BOOK"
perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "directory mask" "0770"
perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "dos file mode" "0770"
perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "nt acl support" "no"

```

```

perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "security mask" "0770"
perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "case sensitive" "no"
perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "public" "no"
perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "writeable" "yes"
perl modini.pl /etc/samba/smb.conf KADD "[BOOK]" "printable" "no"

perl modini.pl /etc/samba/smb.conf SREMOVE "[LANSHARE]"
perl modini.pl /etc/samba/smb.conf SADD "[LANSHARE]"
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "readlist"
"@TRANSITION"
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "writelst"
"@TRANSITION"
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "adminlist"
"@TRANSITION"
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "comment" ""
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "path"
"/shares/LANSHARE"
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "directory mask"
"0770"
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "dos file mode" "0770"
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "nt acl support" "no"
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "security mask" "0770"
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "case sensitive" "no"
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "public" "no"
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "writeable" "yes"
perl modini.pl /etc/samba/smb.conf KADD "[LANSHARE]" "printable" "no"

```

This shell script is to be executed on the Linux server hosting the Samba server service as the root user ID.

Note: The Samba daemon will need to be refreshed for these changes to take effect without waiting for the 60 second re-read delay. On most Linux servers, issue the command:

```
# service smb restart
```

This shell script uses a Perl utility script `modini.pl` created to simplify the management of sections, keys, and values in a traditional text INI file such as `/etc/samba/smb.conf`. Note that this script uses the `IniFiles` Perl module.

Example 6-18 Utility Perl script for text INI file section, key, and value management

```
#!/usr/local/bin/perl -w

use Config::IniFiles;
$bDebugging = "0";
$returnValue = "";
```

```

$sINIFile = $ARGV[0];
$sCommand = $ARGV[1];
$sSection = $ARGV[2];
$sKeyword = $ARGV[3];
$sValue = $ARGV[4];

if ($bDebugging)
{
    print 'INIFile: ', $sINIFile . "\n";
    print 'Command: ', $sCommand . "\n";
    print 'Section: ', $sSection . "\n";
    print 'Keyword: ', $sKeyword . "\n";
    print ' Value: ', $sValue . "\n";
}

$oConfigFile = new Config::IniFiles( -file => $sINIFile );

# Section: Add

if ($sCommand eq 'SADD')
{
    if ($bDebugging) { print "DEBUG: Section:Add\n"; }
    $sReturnValue = $oConfigFile->AddSection($sSection);
    $oConfigFile->WriteConfig($sINIFile);
}

# Section: Remove

if ($sCommand eq 'SREMOVE')
{
    if ($bDebugging) { print "DEBUG: Section:Remove\n"; }
    $sReturnValue = $oConfigFile->DeleteSection($sSection);
    $oConfigFile->WriteConfig($sINIFile);
}

# Keyword: Remove

if ($sCommand eq 'KVREMOVE')
{
    if ($bDebugging) { print "DEBUG: KeywordValue:Remove\n"; }
    $sReturnValue = $oConfigFile->delval($sSection, $sKeyword);
    $oConfigFile->WriteConfig($sINIFile);
}

# Keyword: Set

if ($sCommand eq 'KVSET')
{
    if ($bDebugging) { print "DEBUG: KeywordValue:Set\n"; }

```

```
$sReturnValue = $oConfigFile->setval($sSection, $sKeyword, $sValue);
if (defined $sReturnValue eq "")
{
    $sReturnValue = $oConfigFile->newval($sSection, $sKeyword, $sValue);
}
$oConfigFile->WriteConfig($sINIFile);
}
```

6.6.5 Copy the data from the OS/2 aliases to the Samba shares

The next step of migrating the file resources consists of copying the alias data. Because both platforms support essentially the same SMB protocol, the migration can be done directly. You can select from several options to get the data from OS/2 to Samba. Examples include:

- ▶ XCOPY on OS/2 to Samba. This requires that the OS/2 system is or can be configured for TCPBEUI communications.
- ▶ Backup and restore programs like TSM. By using such a facility you can prepare the new server offsite.
- ▶ NFS mount on Linux to the OS/2 Server. This requires the availability of the OS/2 NFS server.

The copying of the data needs to be completed independent of user access to provide data integrity.

6.6.6 Migrating DASD limits

There is no direct migration path of OS/2 Warp Server DASD limits to Samba. The OS/2 WarpServer DASD limits are defined like ACLs on a directory level. Limiting a directory means that the amount of data stored in this directory tree may not exceed the defined amount. This is defined independent of the owner of the file.

Samba currently provides no limits and relies on the underlying operating system. Linux on the other hand handles its quotas on a volume level. The amount of storage available for a user may not exceed a certain value regardless where it will be stored. Using Linux quota services sounds in some ways only reasonable for home directories, where the owner of a file is mostly the owner of the directory. Because there is currently no direct mapping for the LAN Server DASD Limits to Samba, we leave the handling of limits to be determined by the reader.

6.6.7 Steps to follow

In summary, the following steps are necessary to migrate the file resources from the OS/2 Servers to the Samba server. Our example simplifies the steps and assumes that all needed servers of the OS/2 and Samba systems are online. You may adapt these steps to your migration workflow:

1. Generate the `getalias.log` and `getacl.log` using the LSMT procedures.
2. Run the transition script for file aliases:

```
setsmbdiralias.cmd getacl.log setDirAlias.sh getalias.log
```
3. Execute the `setDirAlias.sh` shell script as root user ID on the Linux server hosting the Samba services.
4. Prepare for data migration. Remove obsolete backups.
5. Copy the data from the OS/2 Servers to the Samba server.

6.7 Migrating printers

The printer resources are logically next. The process of migrating the printer resources consists of the following steps:

1. Define the operating system print queues.
2. Define the printer shares.

6.7.1 Client printing considerations

There are a number of considerations for workstation printing configuration. An example of a configuration, which will be problematic when moving from an OS/2 print resource to a Samba print resource, is the configuration and operation of network printers.

Important: Migrating to Linux requires that the printing services, as configured and used on the deployed workstations, interact with the OS/2 Server on a share level for ease of migration to Samba. As an example, a local system port is assigned to a remote print share, such as:

```
net use lpt3: \\server_name\printer_share
```

Additionally, applications can be configured to use UNC names to deliver print output to a remote server print share.

It is recommended that the Samba's printing support be explored and tested thoroughly for the customer workstation environment. Each client type, be it OS/2 Warp 4, Windows 2000, or others, bring a unique set of printing concerns that

must be resolved for a migration to be complete and successful. Reviewing the information provided in the Samba Project Documentation document is a must, along with the printing support documentation of the chosen Linux distribution. This will enable the unique printing requirements of the customer environment to be fully configured and addressed.

Samba's printing solutions and options will likely provide a solid, functional printing solution for workstation clients. Many current functions will migrate successfully to the Samba server. One such example is the OS/2 workstation print queue driver share PRINTDRV. Defining a PRINTDRV share in the Samba configuration, and populating the share with current printer driver resources will continue to effectively serve the OS/2 workstations for printer driver setups and updates.

6.7.2 Print queue options

Samba provides a set of printing options. Two basic print queue/share definition options are covered here. Both are BSD-based printing solutions. For further information on Samba 3.0's printing solutions, refer to the extensive details covered in the Samba Project Documentation document.

There are two basic ways to define the print queue shares to Samba:

- ▶ System print queue automatic enumeration and publishing
- ▶ Share definitions

System print queues

Samba can be configured to enumerate the system print queues. Any print queues defined to the Linux server system and specified in the `/etc/printcap` configuration file will be shared by Samba at startup. This provides a simple and easy way to provide branch printing resources to attached workstations. This simplicity comes with a price of limited flexibility or configuration customization from Samba's sharing in that the print queues are all treated equally from an access and control perspective.

Share definitions

Samba also provides similar share definitions as the file shares for printers. A share is defined for each print queue on the Samba server. Each queue can be defined with unique configuration settings this way. The migration approach presented here covers the basics of this approach.

Note: The basic Samba configuration provided in this redbook supports both options. Print queues defined to the Linux server and Samba print shares will be available with the base Samba configuration.

6.7.3 Overview of Samba printer shares

Using share definitions for printers, each printer share in Samba is defined by a section in the `/etc/samba/smb.conf` configuration file. Samba provides a wide range of configuration options for defining and tuning shares. The following is the basic share definition, which is used for the migration scenario.

Example 6-19 Example printer share definition for `/etc/samba/smb.conf`

```
[share_name]
comment = the share comment
path = /shares/spooler/share_name
browseable = yes
printable = yes
writeable = no
guest ok = yes
```

The following defines each entry for a share and the assigned data for the basic migration scenario:

Table 6-6 Samba Printer share section keywords and values

Samba share entry	Entry description
[share_name]	The share section entry header. All entries between this entry and the next [...] entry are considered to be configuration details for this share.
comment = the share comment	A description of this share viewable by users.
path = /shares/spooler/share_name	The path to the entry point of the share for printer data on the Linux server file system. All shares in our migration scenario for printers are stored or linked into a common /shares/spooler directory
browseable = yes	Specifies that this share is included in a NET VIEW report or a browse list.
printable = yes	Specifies that this resource is a spooler resource for printing.
writeable = no	Specifies that users may not write and modify files on the shared resource.
guest ok = yes	Specifies that this resource is accessible without a user ID and password, and thus with guest credentials.

Each share in the migration scenario will use these parameters. As Samba provides a rich selection of options for configuring and tuning shared resources,

it is recommended that the Samba Project Documentation document, and the manual page for the *smb.conf* file be reviewed.

6.7.4 Define print queue shares

Migration of printers is central to the function of the target system. OS/2 print aliases will be converted to Samba print shares during the migration.

The definition of shares in Samba is controlled through the Samba configuration file */etc/samba/smb.conf*.

For the example migration, two LSMT output files are used for this migration step:

Example 6-20 Example of the LSMT output for the aliases (three wrapping lines)

```

OPT;NAME      ;TYPE      ;SERVER      ;PATH
;REMARK
;MAXUSES;QUEUE      ;PRIORITY;DEVICE_POOL ;
;BOOK      ;Files ;PDC      ;F:\BOOK
;
;Within Domain;At server
startup;65535 ;Unknown ;Unknown ;Unknown ;
;LANSHARE;Files ;BDC      ;E:\LANSHARE
;
;Within Domain;At server
startup;65535 ;Unknown ;Unknown ;Unknown ;

```

Example 6-21 LSMT output for the alias access controls (four wrapping lines)

```

* List of all ACLs of existing Aliases, allowed Options U=update D=delete
OPT;ALIAS      ;AUDIT ;ADMINS ;BOOKREAD;BOOKWRITE;GROUPID ;GUESTS ;LOCAL
;PRINTER ;SERVERS ;TRANSITION;USERS ;ANDREI ;AUSRES26;BDC ;GUEST
;LEIF      ;MARC ;OLIVER ;PDC      ;RICHARD ;USERID ;WYNAND ;
A ;BOOK      ;-none-; ; RG ; RWCDAG ; ; ;
;
;
;
A ;LANSHARE ;-none-; ; ; ; ; ;
;
; ; RWCXDAPG ; ; ; ; ;
;
;
P ;PRINT_Q ;-none-; ; ; ; ; ;
; CPG ; ; ; ; ; ; ;
;
; ; ; ; ; ; ;

```

These two LSMT output files are used to produce a resulting shell script to modify the Samba configuration file */etc/samba/smb.conf*. The command issued to produce the LDIF file is:

```
setsmbprnalias.cmd getacl.log setPrnAlias.sh getalias.log
```

The options to the *setsmbprnalias.cmd* files are as follows:

- ▶ `getacl.log`: The access control output from LSMT
- ▶ `setprnalias.sh`: The output file, which modifies the `/etc/samba/smb.conf`
- ▶ `getalias.log`: The alias output from LSMT

The following is an example shell script file for the sample OS/2 print alias to Samba print share mapping for modifying the Samba configuration file `/etc/samba/smb.conf`:

Example 6-22 Example `setprnalias.sh` output file

```
perl modini.pl /etc/samba/smb.conf SREMOVE "[PRINT_Q]"
perl modini.pl /etc/samba/smb.conf SADD "[PRINT_Q]"
perl modini.pl /etc/samba/smb.conf KADD "[PRINT_Q]" "comment" ""
perl modini.pl /etc/samba/smb.conf KADD "[PRINT_Q]" "path"
"/shares/spooler/PRINT_Q"
perl modini.pl /etc/samba/smb.conf KADD "[PRINT_Q]" "browseable" "yes"
perl modini.pl /etc/samba/smb.conf KADD "[PRINT_Q]" "printable" "yes"
perl modini.pl /etc/samba/smb.conf KADD "[PRINT_Q]" "writeable" "no"
perl modini.pl /etc/samba/smb.conf KADD "[PRINT_Q]" "guest ok" "yes"
```

This shell script is to be executed on the Linux server hosting the Samba server service as the root user ID.

Note: The Samba daemon will need to be refreshed for these changes to take effect without waiting for the 60 second re-read delay. On most Linux servers, issue the command:

```
# service smb restart
```

This shell script uses the Perl utility script `modini.pl` created to simplify the management of sections, keys, and values in a traditional text INI file, such as `/etc/samba/smb.conf`. Note that this script uses the `IniFiles` Perl module.

6.7.5 Steps to follow

In summary, the following steps are necessary to migrate the print resources from the OS/2 Servers to the Samba server. You may adapt these steps to your migration scenario.

1. Generate the `getalias.log` and `getacl.log` using the LSMT procedures.
2. Run the transition script for print aliases:


```
setsmbprnalias.cmd getacl.log setPrnAlias.sh getalias.log
```
3. Execute the `setPrnAlias.sh` shell script as root user ID on the Linux server hosting the Samba services.

6.8 Migrating serial devices

OS/2 Warp Server services included the ability to share serial devices. Using that feature, administrators have been able to share bidirectional serial devices like modems within the domain. Samba does not include a comparable feature. Some manufacturers such as those listed below, provide a hardware based solution connecting serial devices over TCP/IP:

- ▶ Equinox Super Serial Ethernet Serial Provider from Alloy Computer Products, found at:
<http://www.alloy.com.au>
- ▶ THINQ Serial Device Server from Quatech INC, found at:
<http://www.quatech.com>

There are drivers for Windows and LINUX available.

6.9 Migrating applications

There is no direct migration path of OS/2 Warp Server public applications to Samba. The administrator can use the public applications to define a folder containing the applications a user should use. There are some third party products or concepts available, which fill this gap:

- ▶ Citrix Metaframe to enable support of Windows applications on the clients desktop. More information can be found at:
<http://www.citrix.com>
- ▶ NetApp suite from 6PAC Consulting providing network applications within a folder. These tools provide different approaches to provide network defined applications for OS/2 and Windows clients, storing configuration in plain INI files or Active Directory. More information can be found in 8.3, "6PAC Network administrative tools" on page 301.
- ▶ Servolution Logon Client from Comtarsia. By replacing the Windows 2000 logon interface these clients can use features of an extended Active Directory schema including network applications. More information can be found in 8.5, "Servolution" on page 345.

6.10 NFS migration

The Network File System (NFS) was developed to allow machines to mount a disk partition on a remote machine as if it were on a local hard drive. This allows for fast, seamless sharing of files across a network.

The advantage of NFS today is that it is a mature standard, well understood, and supported across a variety of platforms.

On the OS/2 platform, NFS is used to share data between different OS platforms, specially UNIX. In the following, we will describe a way to move or translate the NFS server configuration from OS/2 to Red Hat Linux and SuSE Linux.

Note: The Red Hat and SuSE configuration is the same for NFS server, so the following examples apply to both Linux distributions.

6.10.1 Software requirement

In order to migrate the configuration, the following requirements applies to the OS/2 Server:

- ▶ The OS/2 Server is up and running.
- ▶ The OS/2 NFS server is installed, configured properly, and running.

For the Linux server, the following requirements applies:

- ▶ The Linux server is up and running.
- ▶ The NFS server is installed.

6.10.2 Migration scenario

Figure 6-3 shows the migration scenario for NFS servers. The migration scenario is described here:

Important: For the Linux command, you must be logged in as *root*.

- ▶ The OS/2 Server exports the directory *f:\nfs* for public access.
- ▶ Everyone has write access on *f:\nfs*.
- ▶ The Linux server exports the path */opt/public* for public access as shown in Example 6-23.

Example 6-23 Exporting a file system

```
#echo "/opt/public *(rw,no_root_squash)" >> /etc/exportfs
#exportfs -ra
```

- ▶ Everyone has write access to */opt/public*.
- ▶ The OS/2 NFS export is mounted in read-write mode on Linux in the path */mnt/os2* by running the following command on Linux:
mount <os2ipaddress>:nfs /mnt/os2

assuming the /mnt/os2 directory already exists.

- ▶ The files from the OS/2 Server are copied over the network to the Linux server by running the command:

```
cp -pr /mnt/os2/* /opt/public
```

- ▶ The NFS configuration is refreshed, by running the command:

```
exportfs -ra
```

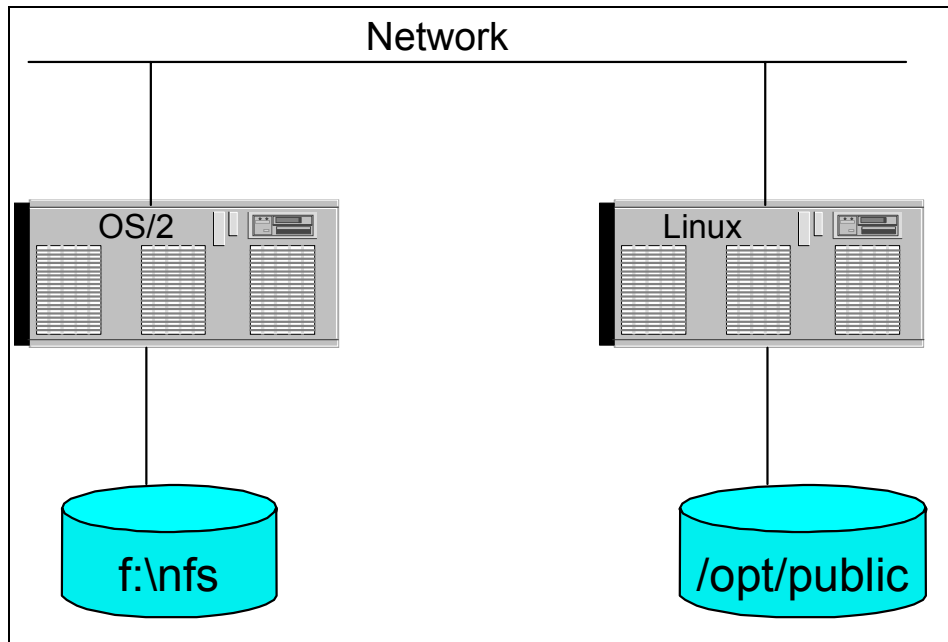


Figure 6-3 Migration scenario

6.10.3 Configuration file for OS/2 Server

The OS/2 configuration file `c:\MPTN\etc\export` is shown in Example 6-24.

Example 6-24 OS/2 NFS configuration file

```
f:\nfs -alias nfs -rw # NFS on PDC  
f:\nfs public
```

6.10.4 Configuration file for Linux server

The Linux configuration file `/etc/exportfs` is shown in Example 6-25.

Example 6-25 Linux NFS configuration file

```
/opt/public *(rw,no_root_squash)
```

Run the command `exportfs -ra` to re-export the file systems.

Note: If the NFS exports are the same, and Linux takes the OS/2 IP address, then there is no modification required for clients. The client does not “know” what OS the server has.

6.10.5 Advanced configuration

For more information about Network File System (NFS), performance, scalability, and security, please read the following documentation:

<http://www.ibiblio.org/pub/Linux/docs/HOWTO/NFS-HOWTO>

- ▶ Linux NFS man pages:
<http://www.linux.org/docs/lp/howto/NFS-HOWTO/server.html>

6.11 FTP migration

File Transfer Protocol (FTP) is a simple and common way to exchange files over the Internet.

6.11.1 Software requirements

In order to migrate the configuration, the following requirements apply to the OS/2 Server:

- ▶ The OS/2 Server is up and running.
- ▶ The FTP server is installed, configured properly, and running.

For the Linux server, the following requirements apply:

- ▶ The Linux server is up and running.
- ▶ The FTP server is installed, *wu-ftpd* for Red Hat respectively *vsftpd* for SuSE.

6.11.2 The migration scenario

The migration scenario is:

1. Create the users on Linux server.
2. Configure the FTP server.
3. Copy the data for each user.
4. Change the ownership of the files for each user.
5. Start the Linux FTP server.
6. Stop the OS/2 FTP server.

Attention: For Linux commands you have to be logged in as *root*.

6.11.3 SuSE FTP configuration

The SuSE SLES 8.0 uses the *vsftp* (Very Secure FTP) package as its FTP server. It is installed by default, and it can be started or stopped from *inetd.conf* file. The configuration file is */etc/vsftpd.conf*.

You can create users either by using the **yast** tool or by using the **useradd** command.

Note: When you create users, specify the home directory for each user.

One important option is to either allow “chroot” or not. In the FTP configuration file, you must uncomment the line *chroot_list_enable=YES*. The FTP server configuration file is shown in Example 6-26.

Example 6-26 vsftpd configuration file

```
## The default compiled in settings are very paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
#
# Allow anonymous FTP?
anonymous_enable=no
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
```

```

# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be owned by
# a different user. Note! Using "root" for uploaded files is not
# recommended!
#chown_uploads=YES
#chown_username=whoever
#
# You may override where the log file goes if you like. The default is shown
# below.
xferlog_file=/var/log/vsftpd.log
#
# If you want, you can have your log file in standard ftpd xferlog format
#xferlog_std_format=YES
#
# You may change the default value for timing out an idle session.
idle_session_timeout=600
#
# You may change the default value for timing out a data connection.
data_connection_timeout=120
#
# It is recommended that you define on your system a unique user which the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsecure
#
# Enable this and the server will recognise asynchronous ABOR requests. Not
# recommended for security (the code is non-trivial). Not enabling it,
# however, may confuse older FTP clients.
#async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII
# mangling on files when in ASCII mode.

```

```
# Beware that turning on ascii_download_enable enables malicious remote parties
# to consume your I/O resources, by issuing the command "SIZE /big/file" in
# ASCII mode.
# These ASCII options are split into upload and download because you may wish
# to enable ASCII uploads (to prevent uploaded scripts etc. from breaking),
# without the DoS risk of SIZE and ASCII downloads. ASCII mangling should be
# on the client anyway..
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
ftpd_banner=Welcome to blah FTP service.
#
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#
# You may specify an explicit list of local users to chroot() to their home
chroot_local_user=YES
# users to NOT chroot().
chroot_list_enable=YES
# (default follows)
chroot_list_file=/etc/vsftpd.chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
ls_recurse_enable=YES

pam_service_name=vsftpd
```

6.11.4 Red Hat FTP configuration

The Red Hat ES v 2.1 uses the *wu-ftp* package as its FTP server. It is installed by default with the system. The configuration file is */etc/ftpaccess*. The newer Red Hat distribution uses vsftpd package as its FTP server.

You can create users by using the *redhat-config-users* tool or by using the **useradd** command.

Note: When you create users using the **useradd** command, you must specify the group for that user, otherwise, a group will be create with the same name as the user.

If you choose to **chroot** the users, you must add the users group at the *guestgroup* parameter in the wu-ftp configuration file. The FTP server configuration file is shown in Example 6-27.

Example 6-27 wu-ftp configuration file

```
# This file controls the behavior of the wu-ftp
# ftp server.
#
# If you're looking for a graphical frontend to
# editing it, try kwuftp from the kdeadadmin
# package.

# Don't allow system accounts to log in over ftp
deny-uid %-99 %65534-
deny-gid %-99 %65534-
allow-uid ftp
allow-gid ftp

# The ftpchroot group doesn't exist by default, this
# entry is just supplied as an example.
# To chroot a user, modify the line below or create
# the ftpchroot group and add the user to it.
#
# You will need to setup the required applications
# and libraries in the root directory (set using
# guest-root).
#
# Look at the anonftp package for the files you'll need.
guestgroup ftpchroot ftp

# User classes...
class all real,guest,anonymous *

# Set this to your email address
email root@localhost

# Allow 5 mistyped passwords
loginfails 5

# Notify the users of README files at login and when
# changing to a different directory
readme README* login
readme README* cwd=*

# Messages displayed to the user
message /welcome.msg login
message .message cwd=*
```

```
# Allow on-the-fly compression and tarring
compress      yes          all
tar           yes          all

# Prevent anonymous users (and partially guest users)
# from executing dangerous commands
chmod         no          guest,anonymous
delete        no          anonymous
overwrite     no          anonymous
rename        no          anonymous

# Turn on logging to /var/log/xferlog
log transfers anonymous,guest,real inbound,outbound

# If /etc/shutmsg exists, don't allow logins
# see ftpshut man page
shutdown /etc/shutmsg

# Ask users to use their email address as anonymous
# password
passwd-check rfc822 warn
```

6.11.5 Creating users on Red Hat

To create users on Red Hat you have two options:

- ▶ Using the **redhat-config-users** tool as shown in Figure 6-4.
- ▶ Using the **useradd** command line as shows in Example 6-28.

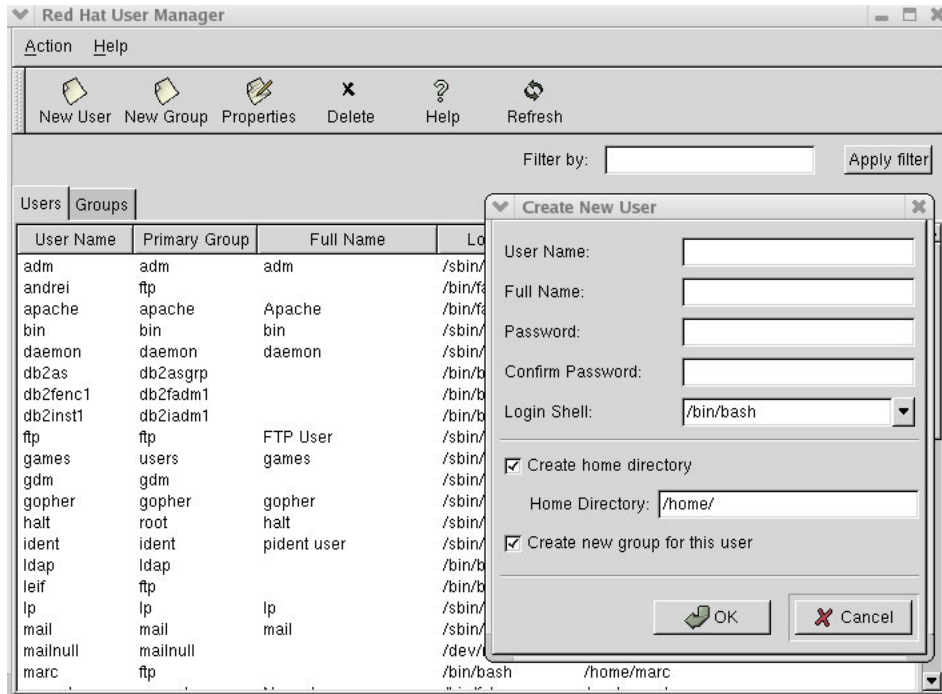


Figure 6-4 Creating users using redhat-config-users tool

Example 6-28 Creating users using command line

```
# useradd -g <group_name> -d <home_dir> <user_name>
# passwd <user_name>
```

6.11.6 Creating users on SuSE

To create users on SuSE you have two options:

- ▶ Using the **yast2** tool as shown in Figure 6-5.
- ▶ Using the **useradd** command line as shown in Example 6-29.

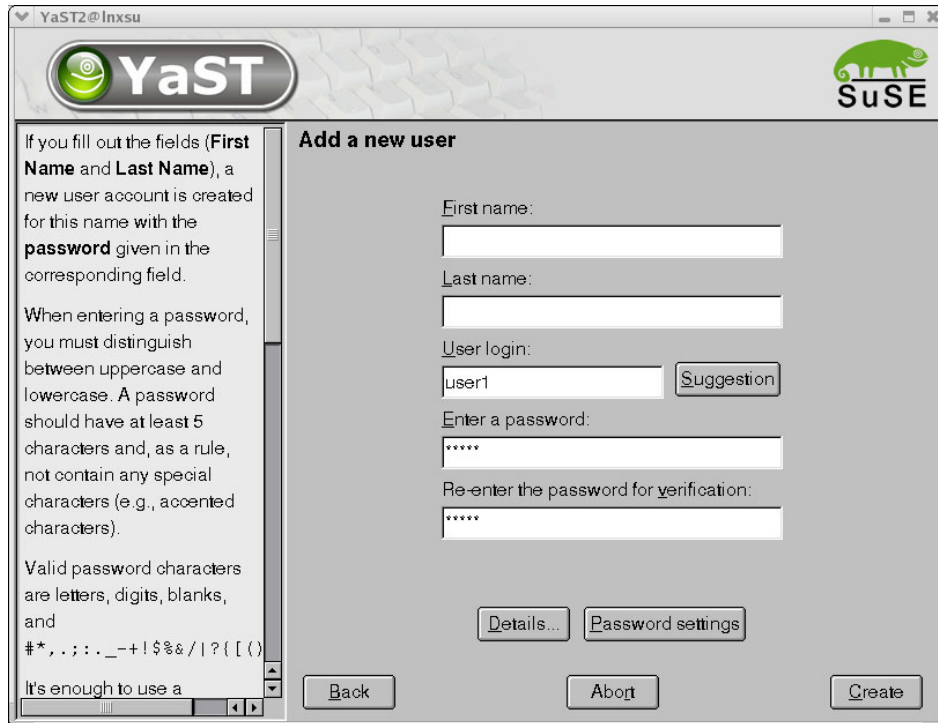


Figure 6-5 Creating users using the *yast2* tool

Example 6-29 Creating users using the command line

```
# useradd -g <group_name> -d <home_dir> <user_name>
# passwd <username>
```

6.11.7 Transfer the data from OS/2 to Linux

To transfer the data between the OS/2 Server and Linux, you can use the FTP protocol. If you have subdirectories, you should choose an FTP client that knows how to transfer a complete directory subtree all at once.

On Linux you can use *gftp* client, which is a graphical client. If you need a command line FTP client, the use of *lftp* client is a good choice. It is not installed by default. You could also choose **wget**, which is installed by default. The FTP clients supports features such as proxy, file transfer resume, and retry, which are useful if files are transferred over slow lines. For more information see the man pages for *lftp* and *wget*.

After the transfer is completed, change the ownership of those files. Use the following command:

```
chown <user1>.<group1> -R /<path>
```

6.12 DHCP migration

The Dynamic Host Configuration Protocol (DHCP) is an Internet protocol for automating the configuration of computers that use TCP/IP. DHCP can be used to automatically assign IP addresses, to deliver TCP/IP stack configuration parameters such as the subnet mask and default router, and to provide other configuration information such as the addresses for printer, time, and news servers.

DHCP's purpose is to enable individual computers on an IP network to extract the configurations from a server (the DHCP server) or servers. In particular, servers that have no exact information about the individual computers until they request the information. The overall purpose of this is to reduce the work necessary to administer a large IP network. The most significant piece of information distributed in this manner is the IP address.

Note: The Red Hat and SuSE configuration is the same for DHCP servers, so the following examples apply to both Linux distributions.

6.12.1 Software requirements

In order to migrate the configuration, the following requirement applies to the OS/2 Server:

- ▶ The OS/2 Server is up and running.
- ▶ The DHCP server is installed, configured properly, and running.

For the Linux server, the following requirements applies:

- ▶ The Linux server is up and running.
- ▶ The DHCP server is installed.

6.12.2 Migration scenario

The following describes the steps to migrate a DHCP server from OS/2 to Linux. The migration scenario is:

1. Decrease the lease time on the OS/2 Server. In this way the clients will update the configuration sooner after the new server is on line.
2. Stop the OS/2 DHCP server.

3. Migrate the configuration from OS/2 Server to Linux server, either using the script that we provide, or using your own script, or working manually.
4. Start DHCP server on Linux.

Important: Stop the OS/2 DHCP server before starting the DHCP server on Linux. Typically, two DHCP servers should not be running concurrently on the same subnet.

Tip: The Linux server must not have the same IP address as OS/2, as long they are on the same subnet.

6.12.3 Configuration file for OS/2

The OS/2 DHCP configuration file `C:\MPTN\ETC\DHCPD.CFG` is show in the following example:

Example 6-30 The dhcpd.cfg file

```
logFileName dhcpd.log
logFileSize 100
numLogFiles 10
logItem SYSERR
logItem OBJERR
logItem WARNING
logItem INFO
leaseExpireInterval 1 Minutes
leaseTimeDefault 24 Minutes
pingTime 1 Seconds
reservedTime 5 Minutes
usedIPAddressExpireInterval 1000 Seconds
statisticSnapshot 1
updateDNSA "nsupdate -f -h%s -s"d;a;*;a;a;%s;s;%s;3110400;q"
releaseDNSA "nsupdate -f -h%s -s"d;a;%s;s;%s;0;q"
(ARecKeyInfo somedomain.local 127.0.0.1
supportBOOTP no
supportUnlistedClients both
allRoutesBroadcast no
UserMatchesVendorClass no
servertype dhcp

appendDomainName yes
canonical no
proxyARec no
#vendor PXEClient
subnet 192.168.25.0 255.255.255.0 192.168.25.10-192.168.25.200 (alias=SOMENAME
```

```
{
  supportUnlistedClients no
  client 0 0 192.168.25.30
  client 0 0 192.168.25.31
  client 0 0 192.168.25.32
  client 0 0 192.168.25.33
  client 0 0 192.168.25.34
  client 0 0 192.168.25.35
  client 0 0 192.168.25.36
  client 0 0 192.168.25.37
  client 0 0 192.168.25.38
  client 0 0 192.168.25.39
  client 0 0 192.168.25.40
```

6.12.4 Using a script to migrate the DHCP configuration

The following provides a Linux script that gathers the DHCP configuration from the OS/2 file, *dhcpsd.cfg*, and creates a DHCP server configuration file for Linux, *dhcpd.conf*.

Note: The script is provided as an example. It is not designed to migrate any DHCP configuration. It migrates only the configuration required to run a DHCP server. You can modify it to suit your configuration.

Attention: Test the script with the following command to make sure it runs properly on Linux:

```
# dos2linux os22linux-dhcp.sh
```

The script works based on the following assumptions:

- ▶ It runs on a Linux platform.
- ▶ There is only one exclude IP range.
- ▶ The exclude IP range is continuous.
- ▶ The exclude IP range is in the same type C IP class.
- ▶ The exclude IP range is not at the beginning or the end of the IP range.

The script has the following features:

- ▶ The path where the *dhcpsd.cfg* and *dhcpd.conf* file are available can be supplied either as variables inside the script as shown in Example 6-31, or as start parameters as shown in Example 6-32.

Important: The first parameter is the path for the OS2 file and the second parameter is the path for the Linux file.

- ▶ If the script does not gather some of data properly, you can overwrite the configuration for that data in the *VARIABLE DATA FOR CUSTOM SCRIPTS* section.
- ▶ For debugging purpose, uncomment the **echo** commands at the end of the script. In this way, the script displays some messages while gathering the data.

Example 6-31 Modify the variables inside the script

```
# vi os22linux-dhcp.sh

### VARIABLE DECLARATION ###
# If you want you can supply the OS2 dhcpcd.cfg path file and
# the Linux dhcpd.conf path file as command line parameter, or
# it can be set within the file.
# Beware: Use one option for both files.
OS2PATH=/mnt/nfs/mptn/etc/dhcpcd.cfg
LNXPATH=/etc/dhcpd.conf

if [ "$1" != "" ]; then
OS2PATH=$1;
fi

if [ "$2" != "" ]; then
LNXPATH=$2;
fi
```

Example 6-32 Start the script with parameters

```
./os22linux-dhcp.sh /mnt/nfs/mptn/etc/dhcpcd.cfg /etc/dhcpd.conf
```

The **os22linux-dhcp.sh** script is shown in Example 6-33.

Example 6-33 The os22linux-dhcp.sh script

```
#!/bin/bash

### VARIABLE DECLARATION ###
# If you want you can supply the OS2 dhcpcd.cfg path file and
# the Linux dhcpd.conf path file as command line parameter, or
# it can be set within the file.
# Beware: Use one option for both files.
OS2PATH=
```



```

LNXPATH=

if [ "$1" != "" ]; then
OS2PATH=$1;
fi

if [ "$2" != "" ]; then
LNXPATH=$2;
fi

# OS2PATH= # The OS2 path for dhcpcsd.cfg file
# LNXPATH= # The LINUX path dhcpcd.conf file
# TEMPDIR= # Temp directory

### VERY IMPORTANT DO NOT REMOVE ###
dos2unix $OS2PATH >/dev/null 2>&1

### VARIABLE DATA FOR CUSTOM SCRIPTS
# If the script is unable to gather the correct information
# from the OS2 dhcpcsd.cfg file please type the correct
# information for each variable

NETWORK_SUBNET=
NETMASK_SUBNET=
START_RANGE_01=
STOP_RANGE_01=
START_RANGE_02=
STOP_RANGE_02=
LEASE_TIME=
MAX_LEASE_TIME=
OPTION_DOMAIN_NAME=
OPTION_DNS_SERVER=
OPTION_ROUTER=
APPEND_DOMAIN_NAME=

### Gathering the information from OS2 file
if [ "$NETWORK_SUBNET" = "" ]; then
NETWORK_SUBNET=`awk '/subnet/ { print $2 }' $OS2PATH`;
fi

if [ "$NETMASK_SUBNET" = "" ]; then
NETMASK_SUBNET=`awk '/subnet/ { print $3 }' $OS2PATH`;
fi

if [ "$START_RANGE_01" = "" ]; then
START_RANGE_01=`awk '/subnet/{ print $4 }' $OS2PATH |cut -d - -f1`;

```

```

fi

if [ "$STOP_RANGE_02" = "" ]; then
STOP_RANGE_02=`awk '/subnet/{ print $4 }' $OS2PATH |cut -d - -f2`;
fi

for i in `awk '/client/ {print $4}' $OS2PATH`
do
IP_A=`echo $i | cut -d. -f1`
IP_B=`echo $i | cut -d. -f2`
IP_C=`echo $i | cut -d. -f3`
IP_D=`echo $i | cut -d. -f4`
if [ "$STOP_RANGE_01" = "" ]; then
DHCP_TMP=`echo $i |cut -d. -f4`
STOP_RANGE_01=`echo $IP_A`.`echo $IP_B`.`echo $IP_C`.`expr $DHCP_TMP - 1 `
fi

if [ "$START_RANGE_02" = "" ]; then
MODIF_START_RANGE_02=yes;
fi

if [ "$MODIF_START_RANGE_02" = "yes" ]; then
DHCP_TMP_01=`echo $i|cut -d. -f4`
START_RANGE_02=`echo $IP_A`.`echo $IP_B`.`echo $IP_C`.`expr $DHCP_TMP_01 +
1`;
fi
done

if [ "$LEASE_TIME" = "" ]; then
LEASE_TIME_TMP=`awk '/leaseExpireInterval/ { print $2 }' $OS2PATH`;

case `awk '/leaseExpireInterval/ { print $3 }' $OS2PATH` in
Minutes)
LEASE_TIME=`expr $LEASE_TIME_TMP \* 60`
;;
Hours)
LEASE_TIME=`expr $LEASE_TIME_TMP \* 3600`
;;
Days)
LEASE_TIME=`expr LEASE_TIME_TMP \* 3600 \* 24`
;;
Months)
LEASE_TIME=`expr LEASE_TIME_TMP \* 3600 \* 24 \* 30`
;;
Years)
LEASE_TIME=`expr LEASE_TIME_TMP \* 3600 \*24 \*30 \* 12`
;;
esac
fi

```

```

if [ "$APPEND_DOMAIN_NAME" = "" ]; then
APPEND_DOMAIN_NAME=`awk ' /appendDomainName/ {print $2}' $OS2PATH`;
fi

if [ "$OPTION_DNS_SERVER" = "" ]; then
OPTION_DNS_SERVER=`awk '/option 6/ { print $3 }' $OS2PATH`;
fi

if [ "$OPTION_ROUTER" = "" ]; then
OPTION_ROUTER=`awk '/option 3/ {print $3}' $OS2PATH`;
fi

if [ "$OPTION_DOMAIN_NAME" = "" ]; then
OPTION_DOMAIN_NAME=`awk '/option 15/ {print $3}' $OS2PATH`;
fi

### Creating the Linux file ###

echo "
subnet $NETWORK_SUBNET netmask $NETMASK_SUBNET {
option domain-name-server "$OPTION_DNS_SERVER";
option routers "$OPTION_ROUTER";
range ${START_RANGE_01} ${STOP_RANGE_01};
range ${START_RANGE_02} ${STOP_RANGE_02};" > $LNXPATH ;

if [ "$APPEND_DOMAIN_NAME" = "yes" ]; then
echo "option domain-name $OPTION_DOMAIN_NAME;" >> $LNXPATH;
fi

if [ "$LEASE_TIME" != "" ]; then
echo "default-lease-time $LEASE_TIME;" >> $LNXPATH;
fi

echo "}" >> $LNXPATH
#For debugging you may uncomment to check in the script runs properly

#echo OPTION_DOMAIN_NAME=$OPTION_DOMAIN_NAME
#echo OPTION_ROUTER=$OPTION_ROUTER
#echo OPTION_DNS_SERVER=$OPTION_DNS_SERVER
#echo APPEND_DOMAIN_NAME=$APPEND_DOMAIN_NAME
#echo LEASE_TIME=$LEASE_TIME
#echo NETMASK_SUBNET=$NETMASK_SUBNET
#echo NETWORK_SUBNET=$NETWORK_SUBNET
#echo START_RANGE_01=$START_RANGE_01
#echo STOP_RANGE_01=$STOP_RANGE_01
#echo START_RANGE_02=$START_RANGE_02
#echo STOP_RANGE_02=$STOP_RANGE_02

```

6.12.5 DHCP configuration file for Linux

Important: *Root* privileges are required to modify the files.

The DHCP configuration file */etc/dhcpd.conf* is shown in Example 6-34.

Example 6-34 /etc/dhcpd.conf file

```
subnet 192.168.25.0 netmask 255.255.255.0 {
    option domain-name "dhcp.somedomain.local";
    option domain-name-servers 192.168.25.2;
    option routers 192.168.25.1;
    range 192.168.25.10 192.168.25.29;
    range 192.168.25.41 192.168.25.200;
}
```

For debugging purposes the file */var/lib/dhcp.leases* gives information about the leased IP address, the hardware address, and the lease time for each address as shown in Example 6-35.

Example 6-35 /var/lib/dhcp.leases

```
lease 192.168.25.10 {
    starts 5 2003/06/13 18:08:20;
    ends 6 2003/06/14 06:08:20;
    hardware ethernet 00:04:ac:9d:6c:70;
}
```

Note: This migration scenario does not affect the clients. The clients are not aware of the change.

6.12.6 Advanced configuration

For more information about DHCP server, performance, scalability, and security, please read the following documentation:

- ▶ <http://www.ibiblio.org/pub/Linux/docs/HOWTO/mini/DHCP>
- ▶ <http://www.isc.org/products/DHCP/dhcp-v3.html>

Linux DHCP man pages

6.13 DNS migration

The Domain Name System (DNS) is a distributed Internet directory service. DNS is used mostly to translate between domain names and IP addresses, and to control Internet e-mail delivery. Most Internet services rely on DNS to work, and if DNS fails, Web sites cannot be located and e-mail delivery stalls.

Note: The Red Hat and SuSE configuration is the same for DNS servers, so the following examples apply to both Linux distributions.

Attention: For SuSE distributions, we need to upgrade the DNS package called bind-9.1.x to bind-9.2.x, so the dynamic DNS will work. Please read 2.1.5, “DNS server” on page 23.

6.13.1 Software requirements

In order to migrate the configuration, the following requirement applies to the OS/2 Server:

- ▶ The OS/2 Server is up and running.
- ▶ The DNS server is installed, configured properly, and running.

For the Linux server, the following requirements applies:

- ▶ The Linux server is up and running.
- ▶ The DNS server is installed.

6.13.2 Migration scenario

There are different approaches to migrate DNS services, which will now be covered.

Migration scenario using DHCP

The migration scenario using DHCP is easier, and it does not affect the clients. The migration steps are:

1. Decrease the IP lease time on the DHCP server, so the clients will update the IP configuration sooner.
2. Create a secondary DNS on a Linux server and replicate the configuration.
3. After the DNS configuration is replicated reconfigure the Linux DNS server to be a primary DNS server.

4. After the Linux DNS server is up and running, change the DHCP configuration so the clients receive only one DNS server, which is the Linux DNS server.

Note: The OS/2 DNS configuration has to be modified to allow a secondary DNS to replicate the configuration.

The scenario works in the following way. At first logon the clients receive and use the old DNS address (OS/2 Server) from the DHCP server. After the new DNS server is up and running (Linux server) the DHCP configuration is changed with the new DNS address. When a client logs on again, or the lease time expires, it requests from the DHCP server a new IP configuration. The DHCP responds with the new IP configuration including the new DNS server, and the clients will use the Linux DNS server as shown in Figure 6-6.

Migrating scenario without DHCP

In this situation there is no smooth migration, because it affects the clients. The network administrator has to manually modify the client network configuration to point to the new DNS server.

The migration steps are:

1. Migrate the DNS configuration from OS/2 DNS server to Linux DNS server.
2. Start the Linux DNS server.
3. After the Linux DNS server is up and running, the OS/2 DNS server can be stopped.

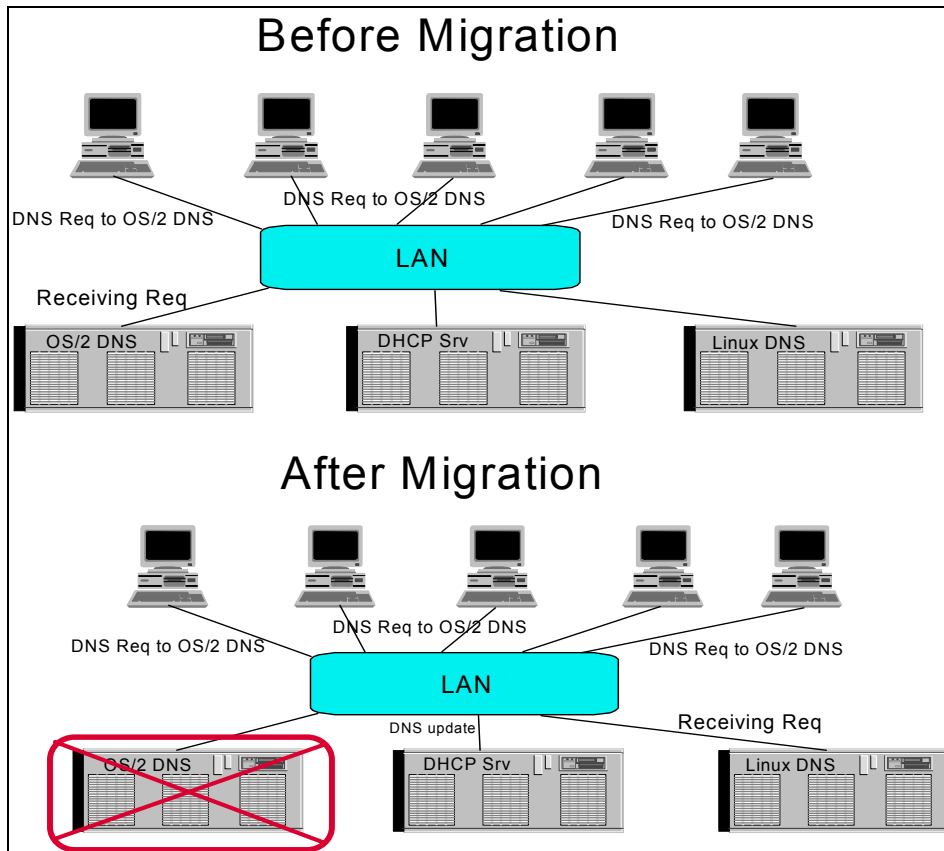


Figure 6-6 Migration scenario using DHCP

6.13.3 Creating a secondary DNS

To create a secondary DNS on Linux, modify the file `/etc/named.conf` and add an entry for the domain `somedomain.local`. Then specify a secondary domain as show in Example 6-36.

Example 6-36 DNS slave example

```
zone "somedomain.local" {
    type slave;
    file "/var/named/somedomain.local.hosts";
    masters {
        192.168.25.2;
    };
    allow-transfer {
        9.3.4.16;
    };
}
```

```
};
```

Attention: After the DNS configuration is migrated to Linux, change the Linux DNS to master mode.

6.13.4 DNS configuration files for OS/2

The DNS server configuration is in the file `c:\mptn\etc\namedb\dnsexst.cfg`. The file is shown in Example 6-37.

Example 6-37 The DNS server configuration file

```
; ***** IBM DDNS Server Administrator
*****
; This file was written by the IBM DDNS Server Administrator on 13-Jun-03
; ***** IBM DDNS Server Administrator
*****
; This file was written by the IBM DDNS Server Administrator on 04-Jun-03
4.3.9.in-addr.arpa (
notify=yes
notify.delayTime=60
notify.retryTime=30
notify.retryNumber=3
timeSync=yes
timeSync.toSecondaries=yes
safeWrite=yes
sigDel=no
ttlSet=no
deferUpdCnt=100
incrTime=300
keyToSec=yes
sepDynStatic=no
reverseMapping=no
)
somename.local (
notify=yes
notify.delayTime=60
notify.retryTime=30
notify.retryNumber=3
timeSync=yes
timeSync.toSecondaries=yes
safeWrite=yes
sigDel=no
ttlSet=no
deferUpdCnt=100
incrTime=300
```



```

keyToSec=yes
sepDynStatic=no
reverseMapping=yes
)
DDNSAdministratorClient (
gui.warn=yes
gui.write=yes
gui.num=100
gui.lease=3600
gui.pad=3110400
gui.reinit=1
gui.sepdata=3
)

```

The zone configuration file is show Example 6-38.

Example 6-38 The somedomain.local zone file

```

; ***** IBM DNS Server Administrator *****
; This file was written by the IBM DNS Server Administrator on 04-Jun-03
; ***** IBM DNS Server Administrator *****
$ORIGIN somename.local.
pdc. IN A 127.0.0.1
bdc IN A 9.3.4.11
ns-updates IN CNAME pdc.
pdc IN A 9.3.4.9
dhcp IN CNAME pdc.
ns IN CNAME pdc.
ddns IN CNAME pdc.

```

6.13.5 DNS configuration files for Linux

The DNS server configuration is in the file */etc/named.conf*. This file contains all configuration information regarding the daemon such as:

- ▶ IP address to bind to, port to listen to, and so on
- ▶ The zone configuration; the file that keeps the zone records; who has access to modify the zone, and so on
- ▶ Other options such as logging, secret keys, and so on

The */etc/named.conf* file is shown in Example 6-39.

Example 6-39 /etc/named.conf file

```
// generated by named-bootconf.pl

options {
    directory "/var/named";
    /*
     * If there is a firewall between you and nameservers you want
     * to talk to, you might need to uncomment the query-source
     * directive below. Previous versions of BIND always asked
     * questions using port 53, but BIND 8.1 uses an unprivileged
     * port by default.
     */
    // query-source address * port 53;
};

//
// a caching only nameserver config
//
controls {
    inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};

zone "." IN {
    type hint;
    file "named.ca";
};

zone "localhost" IN {
    type master;
    file "localhost.zone";
    allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "named.local";
    allow-update { none; };
};

include "/etc/rndc.key";

zone "somedomain.local" {
    type master;
    file "/var/named/somedomain.local.hosts";
};
```

The zone files are stored in */var/named* directory. Each zone has its own file, the zone *somedomain.local* is in the file *somedomain.local.hosts*. The file is shown in Example 6-40.

Example 6-40 *somoedomain.local* file

```
$ttl 38400
somedomain.local.      IN      SOA      ns. root.ns. (
                        1055534258
                        10800
                        3600
                        604800
                        38400 )
somedomain.local.      IN      NS       ns
ns.somedomain.local.   IN      A        192.168.25.2
lnxrh.somedomain.local. IN      A        192.168.25.2
lnxsu.somedomain.local. IN      A        192.168.25.3
gw.somedomain.local.   IN      A        192.168.25.1
```

6.13.6 Advanced configuration

The Linux DNS server supports many configurations, and the description of those features are beyond the scope of these book. For more information read the following:

- ▶ <http://www.isc.org/products/BIND/>
- ▶ <http://www.ibiblio.org/pub/Linux/docs/HOWTO/DNS-HOWTO>

Linux DNS man pages

6.14 DDNS migration

Dynamic DNS (DDNS) service allows you to assign a fixed machine name to a dynamic IP address. Dynamic DNS provides you with the ability to change the IP address of your domain name to point to your dynamically allocated IP address.

6.14.1 Software requirements

In order to migrate the configuration, the following requirements apply for the OS/2 Server:

- ▶ The OS/2 Server is up and running.
- ▶ The DNS server is installed, configured properly, and running.
- ▶ The DHCP server is installed, configured properly, and running.

For the Linux server, the following requirements apply:

- ▶ The Linux server is up and running.
- ▶ The DNS server *version 9.2.0* or newer is installed.
- ▶ The DHCP server *version 3.0.1rc7* or newer is installed.

6.14.2 Migration scenario

In the following, we will describe how to migrate a DDNS server configuration from OS/2 to Linux:

1. Migrate the DHCP configuration from OS/2 to Linux as described in Chapter 6.12, “DHCP migration” on page 243.
2. Migrate the DNS configuration from OS/2 to Linux as described in Chapter 6.13, “DNS migration” on page 251.
3. Configure the Linux DDNS server.
4. Stop the OS/2 DDNS server.
5. Start the Linux DDNS server.

Important: When migrating the configuration, be careful not to start the Linux DHCP server while the OS/2 DHCP is still running.

Note: At the time of this writing, Red Hat version v2.1 ES did not have a DHCP version 3.0.1rc7 or newer. For Red Hat servers, the DHCP installation is described in 2.3.5, “DHCP server” on page 35.

Note: The following information is the same for Red Hat and for SuSE.

6.14.3 Configure the Linux DDNS server

The following configures a secure DDNS server, where all the updates are made by the DHCP server in a secure way. The clients have to “ask” the DHCP server for an address and to send its name (usually the computer name) then the DHCP server will update the DNS server for it. In this way, only the DHCP server is allowed to update the DDNS server.

The first step is to generate a secure key as shown in Example 6-41.

Example 6-41 Generating a secure key

```
# dnssec-keygen -a hmac-md5 -b 512 -n USER ddns # where ddns is just a name
# ls -l K*
-rw----- 1 root    root          111 Jun 23 13:01 Kddns.+157+33783.key
```

```
-rw----- 1 root    root          145 Jun 23 13:01
Kddns.+157+33783.private
# cat Kddns.+157+33783.private
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key:
fenYvvPhNksRhjHXS1PErcb5i3cJW5xyPBCI1mI961GyUj0yH37y/Bx1aeiGZyrQsBo1IG6kY5n5bxa
25azMoA==
```

Note: The key is the string after the *key* word, so in our case it would be:
fenYvvPhNksRhjHXS1PErcb5i3cJW5xyPBCI1mI961GyUj0yH37y/Bx1aeiGZyrQsBo1IG6kY5n5bxa25azMoA==

Now edit the file */etc/rndc.conf* with your favorite editor, and add the following as shown in Example 6-42; also modify the existing settings, especially the *<key name>* that is usually named *rndc_key*.

Example 6-42 Adding the key in the file /etc/rndc.conf

```
key "ddns" {
algorithm "hmac-md5";

secret "replacemewithyourgeneratedkey";

};
```

Now edit the file */etc/named.conf* with your favorite editor and add the following as shown in Example 6-43. Also, modify the *<key name>* in the control tab, and add the *allow-update* tag for each domain that DHCP will update, as shown at the end of Example 6-43.

Example 6-43 Modifying the file /etc/named.conf

```
# vi /etc/named.conf
key "rndc_key" {

algorithm "hmac-md5";

secret "replacemewithyourgeneratedkey";
};

zone "dhcp.somedomain.local" {
    type master;
    file "/var/named/dhcp.somedomain.local.hosts";
    allow-update { key "ddns"; };
};
```

Now, edit the file `/etc/dhcpd.conf`, and add the domains that DHCP will update and the secret key as shown in Example 6-44.

Example 6-44 Adding the DNS update configuration in the DHCP server

```
# vi /etc/dhcpd.conf

ddns-update-style interim;

key ddns {
    algorithm HMAC-MD5;
    secret
fenYvvPhNksRhjHXS1PErcb5i3cJW5xyPBCI1mI961GyUj0yH37y/Bx1aeiGZyrQsBo1IG6kY5n5bxa
25azMoA==;
};

zone dhcp.somedomain.local. {
    primary 127.0.0.1;
    key ddns;
}

zone 25.168.192.in-addr.arpa. {
    primary 127.0.0.1;
    key ddns;
}
```

After all modifications that are done to the DDNS and DHCP servers have to be restarted. The configuration files are listed as follows:

- ▶ The file `/etc/rndc.conf` is listed in Example 6-45.
- ▶ The file `/etc/named.conf` is listed in Example 6-46.
- ▶ The file `/etc/dhcpd.conf` is listed in Example 6-47.

Example 6-45 The /etc/rndc.conf file

```
# cat /etc/rndc.conf
/*
 * Copyright (C) 2000, 2001 Internet Software Consortium.
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose with or without fee is hereby granted, provided that the above
 * copyright notice and this permission notice appear in all copies.
 *
 * THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM
 * DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL
 * INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT,
 * INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING
 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
```

```

* NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
* WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
*/

/* $Id: rndc.conf,v 1.7 2001/01/09 21:40:45 bwellington Exp $ */

/*
 * Sample rndc configuration file.
 */

options {
    default-server localhost;
    default-key "ddns";
};

server localhost {
    key "ddns";
};

key "ddns" {
    algorithm hmac-md5;
    secret
"fenYvvPhNksRhjHXS1PErcb5i3cJW5xyPBCI1mI961GyUj0yH37y/Bx1aeiGZyrQsBo1IG6kY5n5bx
a25azMoA=";
};

```

Example 6-46 The /etc/named.conf file

```

# cat /etc/named.conf

options {
    directory "/var/named";
    /*
     * If there is a firewall between you and nameservers you want
     * to talk to, you might need to uncomment the query-source
     * directive below. Previous versions of BIND always asked
     * questions using port 53, but BIND 8.1 uses an unprivileged
     * port by default.
     */
    // query-source address * port 53;
};

//
// a caching only nameserver config
//
controls {
    inet 127.0.0.1 allow { localhost; } keys { ddns; };
};
zone "." IN {

```

```

        type hint;
        file "named.ca";
    };

zone "localhost" IN {
    type master;
    file "localhost.zone";
    allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "named.local";
    allow-update { none; };
};

include "/etc/rndc.key";

zone "somedomain.local" {
    type master;
    file "/var/named/somedomain.local.hosts";
    also-notify {
        9.3.4.16;
    };
    notify yes;
};

zone "dhcp.somedomain.local" {
    type master;
    file "/var/named/dhcp.somedomain.local.hosts";
    allow-update { key "ddns"; };
};

key "ddns" {
    algorithm "hmac-md5";
    secret
"fenYvvPhNksRhjHXS1PErcb5i3cJW5xyPBCI1mI961GyUj0yH37y/Bx1aeiGZyrQsBo1IG6kY5n5bx
a25azMoA=";
};

zone "25.168.192.in-addr.arpa" {
    type master;
    file "/var/named/192.168.25.rev";
    allow-update { key "ddns"; };
};

```

Example 6-47 The /etc/dhcpd.conf file

```

subnet 192.168.25.0 netmask 255.255.255.0 {
    option domain-name "dhcp.somedomain.local";

```



```

        option domain-name-servers 192.168.25.2;
        option routers 192.168.25.1;
        range 192.168.25.10 192.168.25.30;
        range 192.168.25.40 192.168.25.200;
        ddns-domainname "dhcp.somedomain.local";
    }
ddns-update-style interim;

    key ddns {
        algorithm HMAC-MD5;
        secret
fenYvvPhNksRhjHXS1PErcb5i3cJW5xyPBCI1mI961GyUj0yH37y/Bx1aeiGZyrQsBo1IG6kY5n5bxa
25azMoA==;
    };

zone dhcp.somedomain.local. {
    primary 127.0.0.1;
    key ddns;
}

zone 25.168.192.in-addr.arpa. {
    primary 127.0.0.1;
    key ddns;
}

```

For more information please read:

- ▶ <http://www.performancemagic.com/howtos/ddns.php>
- ▶ <http://ops.ietf.org/dns/dynupd/secure-ddns-howto.html>

Linux dhcpd.conf man pages

Linux named.conf man pages

6.15 Summary

After performing the steps described in this chapter, the basic infrastructure supplied by the OS/2 Server(s) will have been replicated in a Linux environment. Information such as user definitions and profiles, printer definitions, and all other objects from the OS/2 domain should now be available to the client systems.

In the next chapter, we provide some additional information on migrating common middleware such as database systems, communications servers, and so on.



Part 4

Tools and scenarios

Part 4 of this book describes additional tools and products that may be used to ease the migration process. Some of these tools have been developed by IBM and made available on an as-is basis. Others are available from IBM partners and software vendors.

For the tools we discuss, we include scenarios that depict how these tools can be used during a server migration.

We did not intend to provide a complete survey of all products in the market place, which may assist with an OS/2 migration, but rather we chose a few representatives of the kinds of products available, and how they can ease the migration process.



Migrating the software stack to Linux

This chapter discusses the migration of some core middleware products typically used on OS/2 onto the Linux platform. It covers IBM Universal Database, e-Network Communications Server, Lotus Domino, IBM HTTP Server, and the Tivoli Storage Manager Client.

7.1 Migrating IBM Universal Database

Migrating the IBM DB2 from one platform to another is a complex task and might be time consuming. It is highly recommended to research and thoroughly test the procedures before making any changes to your production environment. In addition, a backup of all data should be performed.

Typically, there are a number of applications that use DB2 as their data store, and migration becomes more of an application migration issue rather than a database migration.

The following sections describe the migration steps, some procedures, and useful tips. Another good source for information regarding migrating a DB2 database is the DB2 User Guide.

Important: For detailed information, step-by-step procedures, and how-to's, visit:
<http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/document.d2w/report?fn=db2v7dmfrm3toc.htm>

7.1.1 Migration scenario

The migration scenario is steps include:

1. Install and configure the target platform (including patches if necessary).
2. Choose a time when the DB2 server is not heavily utilized.
3. Export the data from the source DB2 server.
4. Import the data to the target DB2 server.
5. Change the application links to match the new configuration.

7.1.2 Exporting and importing the data

Compatibility is important when exporting, importing, or loading data across platforms. There are several options available for moving the databases from one platform to another:

1. Moving data across platforms:

- PC/IXF File Format

PC/IXF is the recommended file format for transferring data across platforms. PC/IXF files allow the load utility or the import utility to process (normally machine dependent) numeric data in a machine-independent fashion. For example, numeric data is stored and handled differently by Intel and other hardware architectures, such as mainframes.

- Delimited ASCII (DEL) File Format. DEL files can have differences based on the operating system on which they were created. These differences include:
 - Row separator characters
 - UNIX based text files use a line feed (LF) character.
 - Non-UNIX based text files use a carriage return/line feed (CRLF) sequence.
 - End-of-file character
 - UNIX based text files do not have an end-of-file character.
 - Non-UNIX based text files have an end-of-file character (X'1A').
 - WSF File Format
 - Numeric data in WSF format files is stored using Intel machine format. This format allows Lotus WSF files to be transferred and used in different Lotus operating environments (for example, in Intel based and UNIX based systems).
2. Moving data using the **db2move** tool
 - This tool facilitates the movement of large numbers of tables between DB2 databases located on workstations. The tool queries the system catalog tables for a particular database, and compiles a list of all user tables. It then exports these tables in PC/IXF format. The PC/IXF files can be imported or loaded to another local DB2 database on the same system, or can be transferred to another workstation platform, and imported or loaded to a DB2 database on that platform.
 3. Moving data with DB2 Connect
 - If you are working in a complex environment in which you need to move data between a host database system and a workstation, you can use DB2 Connect, the gateway for data transfer from the host to the workstation, as well as the reverse.
 4. Moving data between typed tables
 - The DB2 export and import utilities can be used to move data out of, and into, typed tables. Typed tables may be in a hierarchy. Data movement across hierarchies can include:
 - Movement from one hierarchy to an identical hierarchy
 - Movement from one hierarchy to a sub-section of a larger hierarchy
 - Movement from a sub-section of a large hierarchy to a separate hierarchy
 5. Using replication to move data
 - Replication allows you to copy data on a regular basis to multiple remote databases. If you need to have updates to a master database automatically

copied to other databases, you can use the replication features of DB2 to specify what data should be copied, which database tables the data should be copied to, and how often the updates should be copied. The replication features in DB2 are part of a larger IBM solution for replicating data in small and large enterprises.

6. Using the Data Warehouse Center to move data

You can use the Data Warehouse Center (DWC) to move data from operational databases to a warehouse database, which users can query for decision support. You can also use the DWC to define the structure of the operational databases, called sources. You can then specify how the operational data is to be moved and transformed for the warehouse. You can model the structure of the tables in the warehouse database, called targets, or build the tables automatically as part of the process of defining the data movement operations. The Data Warehouse Center uses the following DB2 functions to move and transform data:

- a. SQL: You can use SQL to select data from sources and insert the data into targets. You also can use SQL to transform the data into its warehouse format. You can use the Data Warehouse Center to generate the SQL, or you can write your own SQL.
- b. Load and export utilities: You can use these DB2 utilities to export data from a source, and then load the data into a target. These utilities are useful if you need to move large quantities of data.

7.2 Migrating IBM e-Network Communications Server

Unfortunately, there is no migration utility for Communications Server for Linux.

The configuration must be redone completely since the configuration files are very different, and there is no automatic conversion utility.

For more information about Communications Server for Linux see:

<http://www-3.ibm.com/software/network/commsserver/linux>

7.3 Migrating Lotus Notes® server

In the following sections we describe the Lotus Domino migration from OS/2 to Linux. We describe how to migrate from Lotus Domino version 5.x on OS/2 to Lotus Domino version 5.x to Linux.

Note: If you are running Lotus Domino version 4.x on OS/2, we recommend that you upgrade to Lotus Domino version 5.x, and then to migrate to a Linux server.

Note: If you want to use the Lotus Domino version 6.x on Linux, we recommend to first migrate from version 5.x on OS/2, and then to upgrade to version 6.x on Linux. Lotus Domino version 6.x does not exist on OS/2.

7.3.1 Migration scenario

Note: The migration scenario is the same for both Linux distributions.

The migration scenario is:

1. Install the Lotus Domino on Linux, at the time of writing this book, the release for version 5 is 5.0.12.
2. Copy the notesdata directory from OS/2 to Linux through FTP or NFS, or a backup/restore procedure.
3. Copy the notes.ini file from OS/2 to Linux in the notesdata directory.
4. Change the ownership to notes user for the notesdata directory.
5. Modify the notes.ini file to reflect the new path for notesdata directory.

You can change the ownership of the notesdata directory with the command:

chown notes.notes -R /opt/notesdata, where:

- ▶ User name is notes.
- ▶ The group name is notes.
- ▶ The path to notesdata directory is /opt/notesdata.

Important: Remember to change the notes.ini file to use forward slashes in your paths from \ to / .

Note: In order for the migration to be transparent for clients, we can change the DNS entry for the Lotus Domino server to reflect the new IP address. If you are not using a DNS server, you have to stop the OS/2 Server, and move the IP address to the Linux server.

7.3.2 Migrating the configuration

The steps to perform the actual migration are simple, and can be seen above.

7.4 Migrating IBM HTTP Server

IBM fortunately has ported this great product to many platforms, so a migration is simple and straight forward.

7.4.1 Software requirements

In order to migrate the configuration, the following requirements apply for OS/2 Server:

- ▶ The OS/2 Server is up and running.
- ▶ The IBM HTTP Server is installed, configured properly, and running.

For the Linux server, the following requirements apply:

- ▶ The Linux server is up and running.
- ▶ The IBM HTTP Server is installed.
- ▶ The Java runtime is installed.

7.4.2 Migration scenario

The migration scenario is:

1. Copy the Web information from OS2 to Linux. The Web information is in the same directory `<path>/htdocs/`.
2. Copy and modify the configuration file `httpd.conf` file.
3. Start the IBM HTTP Server on Linux.
4. Update the DNS entry with the new Web server IP address, or stop the OS2 server, and set the IP address on the Linux Web server.

Note: The migration procedure applies to both Linux distributions, Red Hat, and SuSE.

Copying the Web files

The Web files can be copied through FTP or NFS, or a backup and restore procedure. By default, the Web root is set as the `htdocs` directory. If the Web root is not on that path, make sure to change the ownership of the files when you copy the files from OS2 server. The Web files have to be owned by the user who will start the IBM HTTP Server.

Tip: Make sure the Web files are using the relative path and not the full path. Depending on the designs of the Web information, the files and links might be affected by the path descriptor. Linux use the UNIX path descriptor / and OS/2 uses the path descriptor \

Modify the httpd.conf

The main changes to the *httpd.conf* file are:

- ▶ The IP address that the server will listen to (if applicable)
- ▶ The Web root path, by default on Linux it is /opt/IBMHTTP/htdocs

For more information about IBM HTTP Server visit:

<http://www-3.ibm.com/software/webservers/httpservers/library.html#v1319>

7.5 Migration of ADSM Client

OS/2 uses the ADSM Client. At the time of writing this book, the latest version of TSM is 5.1.5. For our scenario, we have a TSM server installed on an AIX Server version 5.1.5. In order to successfully migrate the ADSM Client, you have to have at least TSM Server Version 5. If you have an earlier version of TSM or ADSM server, you need to upgrade the server because of client requirements. The TSM server upgrade is beyond the scope of this book.

7.5.1 Software requirements

In order to migrate the configuration, the following requirements apply for OS/2 Server:

- ▶ The OS/2 Server is up and running.
- ▶ The ADSM client is installed and configured properly.

For the Linux server, the following requirements apply:

- ▶ The Linux server is up and running.
- ▶ The TSM client is installed.

7.5.2 Migration scenario

The migration scenario is summarized as follows:

1. Copy the *dsm.opt* file to the Linux server through FTP or NFS.
2. Run the script, which we provide that extracts the basic information for the TSM client to work, as shown in Example 7-1, or use your own scripts for

more complex configurations, or create the configuration files for TSM client using the wizard.

3. Start the TSM client.

The TSM client and server version 5.x has a feature useful in migration scenarios. TSM client allows you to access the backups of another node while you are still connected with your account. In our case, it is useful to access the OS/2 backup (when needed) without modifying the *dsm.opt* file or *dsm.sys* file.

7.5.3 Migration of the *dsm.opt* file

In the following, we provide a script that extracts the basic information from an OS2 ADSM file, and creates the configurations files for TSM version 5.x.

The script extracts the NODENAME and TCPSERVERADDRESS variables. In the TSM Client Version 5.x, you need to supply the SERVERNAME variable, which is not in the OS2 ADSM configuration file. The SERVERNAME variable is the server name of the TSM server. The script is listed in Example 7-1.

The script takes as a command line parameter the path to the OS2 *dsm.opt* file. The SERVNAME, LNX_PATH_OPT and LNX_PATH_SYS variables must be set up within the script.

Example 7-1 The scrips os22lnxdsm.sh file

```
#!/bin/bash

### VARIABLE DECLARATION ###
# If you want you can supply the OS2 dsm.opt path file

if [ "$1" != "" ]; then
OS2PATH=$1;
fi

LNX_PATH_OPT=/opt/tivoli/tsm/client/ba/bin/dsm.opt
LNX_PATH_SYS=/opt/tivoli/tsm/client/ba/bin/dsm.sys
SERVERNAME=TSMAIX

### VERY IMPORTANT DO NOT REMOVE ####
dos2unix $OS2PATH >/dev/null 2>&1

### VARIABLE DATA FOR CUSTOM SCRIPTS
# If the script is unable to gather the correct information
# from the OS2 dsm.opt file please type the correct
# information for each variable
```

```

NODENAME=
TCPSEVERADDRESS=

### Gathering the information from OS2 file
if [ "$NODENAME" = "" ]; then
NODENAME=`awk '/NODENAME/ { print $2 }' $OS2PATH`;
fi

if [ "$TCPSEVERADDRESS" = "" ]; then
TCPSEVERADDRESS=`awk '/TCPSEVERADDRESS/ { print $2 }' $OS2PATH`;
fi

### Creating the dsm.opt file ###
echo "SERVERNAME          $SERVERNAME" > $LNX_PATH_OPT
echo "SERVERNAME          $SERVERNAME
      TCPSEVERADDRESS      $TCPSEVERADDRESS
      NODENAME             $NODENAME" > $LNX_PATH_SYS

#For debuding you may uncomment to check if the script runs properly
#echo $OS2PATH
#echo $LNX_PATH_OPT
#echo $LNX_PATH_SYS
#echo $SERVERNAME
#echo $NODENAME
#echo $TCPSEVERADDR

```

7.5.4 Migrating the configuration

Since the configuration is forward compatible, only the two above steps are required. Be aware that some of the files backed up on OS/2 will become useless on Linux, like some OS/2 specific configuration files. Also, extended attributes used on OS/2 will be lost on Linux.

7.6 Summary

For the most common IBM middleware that runs on OS/2, there are equivalent products for the Linux platform. This chapter has provided an overview of these products, and highlights the migration considerations for each product's configuration data.



Additional migration tools

This chapter discusses additional tools, which have not been used in the previous sections of this redbook, but provide features or functions that might be utilized during a migration.

This chapter discusses the following set of tools:

- ▶ IBM Tools to facilitate a better integration of OS/2 and Windows domains. These tools are provided as-is and can be found in the package provided along with this book on the IBM Redbook's Web page.
- ▶ Starfire Titan, a Web browser based management tool
- ▶ A set of tools named NetApp from 6PAC Consulting to simplify several tasks during a migration, and also help with integration between OS/2 and Windows systems.
- ▶ The Lieberman tools suite, a popular package that provides facilities that can help with migrations.
- ▶ Comtarsia Servolution, which provides a client oriented approach to a migration.

8.1 Various IBM tools

The following sections describe utilities that are provided on an as-is basis that may help with various aspects of an OS/2 Server migration. The utilities described are:

- ▶ IBM Networks User Account Manager for Microsoft Windows 2000
- ▶ IBM Networks Password Synchronization Tool

8.1.1 IBM Networks UAM for Microsoft Windows 2000

The IBM Networks User Account Manager for Microsoft Windows 2000 will be called IBM Networks User Account Manager or IBM Networks UAM in the rest of this document. IBM Networks User Account Manager is an add-on to the Microsoft Windows 2000 Network server function and enables user accounts and localgroup aliases to be replicated from an IBM OS/2 Warp Server domain. The tool will only replicate the user accounts and local group aliases to a Windows Member Server.

Prerequisites

You must have the following prerequisite components installed on your Windows 2000 Server system before you can install the IBM Networks User Account Manager:

- ▶ Microsoft Windows 2000 Server with at least Service Pack 3 installed.
- ▶ Microsoft Windows 2000 Server configured as an additional or standalone server.
- ▶ An appropriate network adapter device driver installed and working.
- ▶ NetBEUI communication protocol

Installation

1. Log on to the Windows 2000 system with a user name that has administrator privileges.
2. Right-click the **My Computer** icon in the desktop and click **Properties**.
3. Click the **Network Identification** tab.
4. Click **Properties...**
5. Change the Workgroup name to the name of the IBM OS/2 Warp Server domain.
6. Click **OK**.
7. It will ask you to reboot the machine. Do not reboot the machine now.

Note that if you are changing the name of the server also, then you must reboot before starting the installation of IBM Networks User Accounts Manager.

8. Run the installation program **UAM.EXE** which will install the service.
9. As part of the installation the IBM Networks User Account Manager Properties notebook is displayed. This will already contain the name of the server as a persistent user. Do not delete it.
10. Add the names of persistent user accounts by typing the name in the *Persistent Users* box and then click **Add**. Persistent users are accounts that are managed locally on the Windows 2000 Server system, and are not synchronized by the IBM OS/2 Warp Server domain controller.
11. Add the names of persistent localgroup aliases by typing the name in the *Persistent Localgroups* box and then click **Add**. Persistent localgroups are aliases that are managed locally on the Windows 2000 Server system, and are not synchronized by the IBM OS/2 Warp Server domain controller.
12. After this, the install shield wizard will ask you to reboot the system.
13. Click **Yes**.

Note: During installation of this product, a user account named IBMLOGON is created on the 2000 Server. This account is required for the IBMLOGON service to function properly. This account is maintained internally by the IBMLOGON service, and any changes made to this account may cause the service to fail. Also, the name of the Windows 2000 machine is added to the system, and is configured as a persistent user. This is required for the proper functioning of the IBMLOGON service. Any changes made to this account may cause the service to fail.

Creating the server definition on the OS/2 domain

1. Log on to the IBM OS/2 Warp Server domain controller with a user name that has administrator privileges.
2. Start the LAN server administration graphical user interface. The LAN Server Administration window is displayed.
3. Double-click the **domain name** icon. The domain name window is displayed.
4. Double-click **Defined Servers**. The Defined Servers window is displayed.
5. Right-click **Defined Server Template** and then click **Create another...** The Defined Server-Create notebook is displayed.
6. Click the **Identify** tab.
7. In the Server name box, type the server name of the Windows 2000 system.

8. In the Description box, type a descriptive comment. The Description is an optional field.
9. Click **Create**.
10. An icon displaying the Windows 2000 Server name will display in the Defined Servers window.
11. Close the LAN server administration graphical user interface.
12. There is no need to shut down or restart the domain controller.

Starting the service

1. The IBM Networks User Account Manager service is started automatically by the Windows 2000 Service Control Manager when the Windows 2000 system is started. Normally, no manual intervention will be required to control the IBM Networks User Account Manager service.
2. The IBM Networks User Account Manager may be manually stopped and restarted through the Windows 2000 Service Control Manager by doing the following:
 - a. Click **Start** button, point to **Settings**, and then click **Control Panel**. The Control Panel window is displayed.
 - b. Double-click **Services**. The Services window is displayed.
 - c. Click **IBM Networks User Account Manager**. Click **Stop** to stop the service. Click **Start** to restart the service.

Managing users and groups

1. Users and groups should be managed at your IBM OS/2 Warp Server domain controller. Users and groups managed on the domain controller will be synchronized on the Windows 2000 Server. Users and groups managed on the Windows 2000 Server will not be synchronized in the domain. Users and groups that are created on the Windows 2000 Server that are not designated as persistent users or localgroups will be deleted by the IBM Networks User Account Manager service when the database is synchronized with the domain controller.
2. Setting persistent user accounts and localgroup aliases is a method of preserving locally administered accounts on the Windows 2000 Server system. Persistent users and localgroups are not synchronized with the domain controller and must be managed locally on the Windows 2000 Server system. Examples of why a user account might be designated as persistent are:
 - A local Windows 2000 user that has administrative privileges exclusively on the local system.

- A service that requires a local user account. For example, the following services, if installed, require a local user account: Microsoft Internet Information Server, Microsoft Internet Server Web Application Manager, Lotus Go Web Server, Netscape Communicator Server, and LDAP Server.
3. To configure persistent user accounts and localgroup aliases, do the following:
 - a. Click the **Start** button, point to **Settings**, and then click **Control Panel**. The Control Panel window is displayed.
 - b. Double-click **Network**. The Network notebook is displayed.
 - c. Click the **Services** tab.
 - d. From the **Network Services** list, click **IBM Networks User Account Manager**, and then click **Properties**.
 - e. Add or delete the persistent user accounts and localgroup aliases as needed.
 - f. Click **OK** to exit and save the changes or click **Cancel** to exit without saving any changes.

Viewing Event Log messages

The IBM Networks User Account Manager logs messages to the Windows 2000 Event Viewer. Messages logged to the Event Viewer may be informational (status), warnings or errors. To view messages in the Event Viewer, do the following:

1. Click **Start** button, point to **Programs**, and then point to **Administrative Tools**.
2. Click **Event Viewer**.
3. Click **Log** and then click **System**.
4. Click **View** and then click **Refresh**.
5. Messages logged by the IBM Networks User Account Manager will be listed as from IBMLogon in the Source column.
6. Double-click a message entry line to view the message.

8.1.2 IBM Networks Password Synchronization Tool

The IBM Networks Password Synchronization Tool (PST) facilitates the synchronization of passwords between OS/2 and Windows machines. This is a command line tool, and does not add much overhead to the system.

Prerequisites

This tool works on OS/2 Servers and on Windows 2000 Servers. It is not meant for Windows 2000 workstations. At least Service Pack 3 should be installed on a Windows 2000 Server to work properly. Administrator privileges are required to run this tool on both operating systems.

Installation

Extract the zipped file *passwdsync.zip* to any directory. The zip file contains the following:

- ▶ **PASSWDSYNC.EXE**
This is the main executable, which needs to be run on the Windows 2000 Server.
- ▶ **PASSEXP.CMD**
REXX Script to be run on OS/2 Server with which the passwords need to be synced.
- ▶ **PWDEXP.EXE**
Needs to be copied to the OS/2 Server

Extracting passwords from the OS/2 Server

Copy the REXX script file **PASSEXP.CMD** and **PWDEXP.EXE** to the OS/2 Server in any directory. The **PWDEXP.EXE** should be in the system path, or it should be in the current directory from which **PASSEXP.CMD** is run.

Run **PASSEXP.CMD** from the command prompt. The command line syntax for is:

```
passexp [output_file]
```

This will create a file specified by <output file> which will contain the list of all users and their corresponding one-way encrypted passwords. In case no output file is specified, the default is **PASSWD.TXT** in the current directory from which **PASSEXP.CMD** was run.

Synchronizing OS/2 passwords to Windows Server

Copy the **PASSWDSYNC.EXE** to the Windows Server in any directory. To synchronize the passwords from the OS/2 Server to the Windows server, you need the <output file> from **PASSEXP.CMD**.

The command line syntax for **PASSWDSYNC.EXE** is:

```
passwdsync -i <input file> [-v] [ -l [log file] ]
```

where:

- ▶ **-i** specifies the <input file> to be used.

This file is the one created by `passexp.cmd` on OS/2. You can copy the file from the OS/2 machine to a floppy and use the file directly from the floppy by specifying the path. Otherwise, you can copy the file to the Windows machine and use it.

- ▶ **-v** Enables verbose mode

The relevant output is echoed to the screen.

- l** Enables logging.

The relevant output is logged to a file. If no log file is specified the default is `PasswdSync.log`. Verbose and Logging mode are disabled by default.

Note: Before running the password synchronization tool on the Windows Server, the users must be added to the Windows server.

8.2 Starfire Titan

Starfire Titan formalizes and streamlines the process of migrating OS/2 domains. OS/2 migration with Starfire Titan provides a simplified and fast migration process with consistent results across all migrated OS/2 domains.

Beyond migration scenarios, Starfire Titan unifies existing systems management products and practices into a single, simple interface. Interacting with networked systems with Starfire Titan is a significant departure from conventional system operational and administrative practices.

Titan enables system administrators to script and encapsulate the detailed instructions required to complete operational and administrative tasks as Titan Activities. Titan Activities can then be launched to perform these tasks on deployed enterprise systems. Titan performs these tasks on Java supported platforms including Linux, OS/2, Windows NT/2000/XP/2003, AIX, and Workspace On-Demand. Titan can be extended to perform tasks on any platform that contains a Java Virtual Machine (JVM).

Information on Starfire Titan can be obtained from:

<http://www.titan-central.com>

Detailed information on the installation, configuration, customization, and operation of Starfire Titan can be found in the following documents:

- ▶ *Starfire Titan Installation Guide*
- ▶ *Starfire Titan Administrator's Guide*
- ▶ *Starfire Titan Designer's Guide*

Starfire Titan has a variety of applications. Titan is task oriented rather than platform or product oriented, and the objective of Titan is to simplify traditional operations and administrative tasks and to improve the use of existing systems management products. Examples include:

- ▶ User provisioning
- ▶ Password coordination
- ▶ Operations effectiveness
- ▶ Help desk enablement
- ▶ Operating System migration

In the following sections, we provide an overview of the Titan configuration, features, and functions followed by an overview of its operating system migration capabilities.

8.2.1 Configuration

The Starfire Titan system primarily consists of a *Controller* and *Agents*.

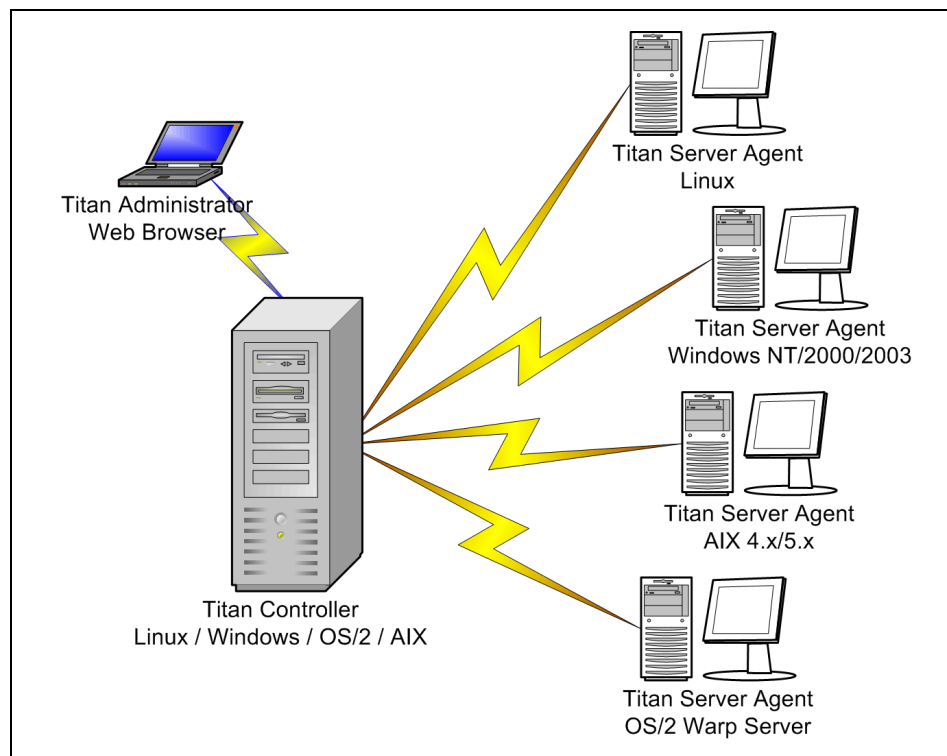


Figure 8-1 Starfire Titan deployment overview

Controller

The controller is a Java application, which resides on a centralized system. This system provides the activity launching and job coordination features of the Titan network. Additionally, the controller provides the browser-based user interface used by Titan administrators.

Agents

The agent is a light-weight Java application, which resides on the distributed enterprise systems. The passive agent uses essentially zero system resources until directed to accomplish a transaction by the controller.

Tools

The transactions dispatched to the agents from the controller complete the task by invoking tools residing on the target system. These tools can be essentially any program, script, or utility which can be invoked to complete an operational or administrative task.

8.2.2 Features and functions

Starfire Titan consists of a variety of features and components including:

Browser-based user interface

Starfire Titan utilizes a browser-based user interface to provide users with a single point of interaction, which is immediately familiar to users. The browser-based user interface helps Titan achieve platform and product independence, and enables access from anywhere to the Titan resources.

Single-image view

Starfire Titan enables users to interact with heterogeneous enterprise resources as if they were a homogeneous resource. From a Titan user's standpoint, the procedure used to change a user's permissions on an OS/2 system is the same for a Linux system.

Activities

Titan Activities are the core of Starfire Titan. The vast amount of information and skills required to roll-out, deploy, and maintain networked computing resources are encapsulated into Titan Activities. The primary function of Titan is to launch Titan Activities that accomplish specific operational and administrative tasks. These tasks can cover multiple platforms and products in a single Activity launch. Titan Activities enable anyone to rapidly and consistently perform any task, on any system, regardless of their skill level or platform expertise. The launch of a

Titan Activity produces one or more Titan Jobs containing the transactions to change the target system.

TitanScript is the language that is used by Activity designers to create Titan Activities. TitanScript is an interpretive language that is syntactically similar to the Java, C, and REXX programming languages. Consult the Starfire Titan Designer's Guide to learn more about TitanScript.

Agents

Titan Agents execute the individual transactions that are contained within a Titan Job. These transactions are dispatched from the Titan Controller. Titan Agents are Java-based applications that reside on the networked systems managed by Titan.

Controller

The Titan Controller is the central nervous system of Titan. The Controller manages and guides all of Titan's functions from the launching of Activities and dispatching of Job transactions to querying and maintaining the Titan Repository. The Controller is the library of enterprise systems skills contained in Activities.

Repository

The Titan Repository takes advantage of IBM DB2's ability to continually grow and change. It stores, receives, and reports updated information about all enterprise objects. The Repository is the library of enterprise information that Titan queries.

Interrogators

Titan Interrogators capture information about enterprise objects and their attributes from Titan-managed systems. The information captured by the Interrogator is imported into the Titan Repository in an XML format.

Tasks and tools

Titan Tools are operating platform specific tools and utilities that are used in conjunction with the individual instructions that are executed on target systems by Titan Agents. These tools can be Starfire supplied tools or enterprise-specific programs and utilities. A Titan Task is a specific behavior of a tool. For example, a tool managing an OS/2 user would have Tasks to create, modify, and delete an OS/2 user.

8.2.3 OS/2 LAN Server migration scenario

Migrating the OS/2 platform LAN server resources to Linux or Windows is a challenging task complicated by the inconsistency in services between the three

platforms. The “mapping” of the available OS/2 resources to the available Linux or Windows resources is the key to effectively migrating from OS/2 to Linux or Windows.

For this migration, the following platforms are defined:

OS/2	OS/2 Warp Server
Linux	Linux with Samba
Windows	Windows Server

The steps to migrating an OS/2 domain using Titan are:

1. Installation and configuration of Titan Controller
2. Installation of Titan Agents on the source OS/2 systems and the target Linux or Windows systems.
3. Importing of OS/2 LAN Server Migration Titan Activity Package
4. Customization of the migration Activities for enterprise-specific mapping requirements
5. Interrogation of source OS/2 domain producing XML data
6. Importing of Domain XML data
7. Launch of Titan Activities for migration
8. Migration of system and user data from OS/2 to the target platform

Steps 5 through 8 are repeated for multiple migrations.

Titan activities

Titan’s behaviors are instantiated through the Titan Activities. For the migration of OS/2 Warp Server resources, the Starfire OS/2 Warp Server Migration Titan Activity Package (TAP) is used. The TAP contains a large variety of Activities from low-level work Activities to the following “top-level” Activities:

- ▶ Migrate OS/2 domain
- ▶ Migrate OS/2 domain - prompted
- ▶ Migrate OS/2 access control
- ▶ Migrate OS/2 directory
- ▶ Migrate OS/2 group
- ▶ Migrate OS/2 printer
- ▶ Migrate OS/2 user

These Activities can be customized to the unique requirements of each enterprise and deployment. The Migrate OS/2 domain activity is used as the launch-point to perform an OS/2 domain migration. This activity will:

1. Prompt for the source OS/2 domain

2. Prompt for the target system type (Linux or Windows)
3. Prompt for the target system from the selected type
4. Compute the mappings and create the Titan Jobs for the migration of the OS/2 resources

Upon the launch of the Migrate OS/2 domain activity, the Titan Jobs will be executed against the target system and create the resources and definitions as customized for the enterprise.

8.2.4 Transformation customization

The core Activities of the OS/2 Warp Server Migration TAP provide a generic base of mapping and migration. This generic configuration can be customized for the enterprise configuration.

Customization of the TAP occurs in two areas:

1. Tools and utilities
2. Activity script

Tools and utilities

The tools and utilities used to complete the migration can be changed as needed. As an example, Starfire provides an INI-file management tool with the TAP. The enterprise environment may currently use another tool that would be preferred. This change can be identified up-front during the customization phase. A variety of tools are used in the migration of OS/2 domains to Linux and Windows platforms. Any of these can be changed or customized as required for the migration scenario.

Activity script

The behavior of the Activities can be customized as desired. Examples of changes include:

- ▶ Removing user prompts where values can be calculated or predetermined
- ▶ Adding prompts for Domain-specific data to be supplied at migration time
- ▶ Data calculation changes for desired unique mapping for the enterprise-specific migration target configuration

Each of the core Activities utilize the Titan Repository as the source of the OS/2 resource object and attribute data elements.

The following is an excerpt of the OS/2 User object migration Activity. This section shows a segment of the Activity where the OS/2 User object attribute

data is being collected from the Titan Repository. Following this collection, the attribute data that is specific and unique for each platform is defined. This additional data is defined as required by the target platform and the tools being used by the Activities.

Example 8-1 Excerpt of the OS/2 User object migration Activity

```
...
-----
// Get object attributes for existing object
-----
userActive#0 = lookupValue("USER", "ACTIVE", "ORGUNIT"=sourceOrgUnit,
"ID"=user#0)
userComment#0 = lookupValue("USER", "COMMENT", "ORGUNIT"=sourceOrgUnit,
"ID"=user#0)
userCountryCode#0 = lookupValue("USER", "COUNTRYCODE", "ORGUNIT"=sourceOrgUnit,
"ID"=user#0)
userExpires#0 = lookupValue("USER", "EXPIRES", "ORGUNIT"=sourceOrgUnit,
"ID"=user#0)
userFullName#0 = lookupValue("USER", "FULLNAME", "ORGUNIT"=sourceOrgUnit,
"ID"=user#0)
userGroups#0 = lookupValue("USER", "GROUPS", "ORGUNIT"=sourceOrgUnit,
"ID"=user#0)
userHomeDirDrive#0 =
lookupValue("USER", "HOMEDIRDRIVE", "ORGUNIT"=sourceOrgUnit, "ID"=user#0)
userHomeDirPath#0 = lookupValue("USER", "HOMEDIRPATH", "ORGUNIT"=sourceOrgUnit,
"ID"=user#0)
userHomeDirReq#0 = lookupValue("USER", "HOMEDIRREQ", "ORGUNIT"=sourceOrgUnit,
"ID"=user#0)
userHomeDirServer#0=lookupValue("USER", "HOMEDIRSERVER", "ORGUNIT"=sourceOrgUnit,
"ID"=user#0)
userID#0 = lookupValue("USER", "ID", "ORGUNIT"=sourceOrgUnit, "ID"=user#0)
-----
// Target Platform Data Modifications - Linux
-----
if(eq(targetPlatformCode, "LNX"))
{
    linuxUID#0 = lookupValue("USER", "UID", "ORGUNIT"=sourceOrgUnit",
"ID"=user#0)
    linuxHomeDir#0 = stringBuild("/home/%0", userID#0)
}
-----
// Target Platform Data Modifications - Windows
-----
if(eq(targetPlatformCode, "W32"))
{
    if(targetIsW32Domain)
    {
        userDomain#0 = "YES"
    }
}
```

```

}
w32HomeDir#0 = stringBuild("\\\\%0\\%1", userHomeDirServer#0,,
stringReplace(userHomeDirPath#0, ":", "$"))
}
...

```

The business and system rules for the mapping of the object attributes for the OS/2 User, Group, and other object types are defined once by the enterprise in the core Activities. These rules will be applied consistently throughout the migration project to each migrated OS/2 domain.

8.2.5 Extraction from OS/2

The Activities query the OS/2 resource data from the Titan Repository. The Repository must be populated prior to the migration of an OS/2 domain. The Titan Interrogator is a thorough and fast means to produce the data for the population of the Repository.

The Titan OS/2 Interrogator is a Java application with OS/2 platform specific code to extract the details of OS/2 domain objects from Users to Groups to Workspace On-Demand Machine Classes. The Titan OS2 Interrogator produces a very complete output of OS/2 Warp Server data.

Example 8-2 Titan OS/2 Interrogator usage options

```

Usage: OrgUnitOS2Extract orgUnit [options]
options:
  -u  extract users
  -g  extract groups
  -al  extract aliases
  -ap  extract applications
  -up  extract user application parameters
  -s  extract servers
  -m  extract machines
  -mc  extract machine classes
  -a  extract access controls
  -all  extract all orgUnit data (default)

  -xml:filename      extract to xml file
  -objmod:filename   extract to object mod file
  -text              extract as text to stdout (default)
  -noxmlheader       do not write orgUnit info header to xml file
  -noobjmodheader    do not write orgUnit info header to object mod file

  -debug            enable debug logging
  -logconf:filename specify log configuration file
  -nobackacc        do not run backacc when querying acs

```

-backaccdays:days	run backacc only if last backacc was run more than specified days prior
-backacchours:hours	run backacc only if last backacc was run more than specified hours prior
-large	use in large orgUnits for memory savings
-gc	periodically request garbage collection when running in large mode

An example command issued on the OS/2 domain controller (or an OS/2 workstation administratively logged onto the Domain to be migrated) to export an OS/2 domain without Workspace On-Demand definitions would be:

```
titanOS2Interrogator SOMEDOMAIN -u -g -al -ap -up -s -a -xml:somedomain.xml
```

The output of the Titan OS/2 Interrogator is an XML-formatted file for import into the Titan Controller. The resulting output file would be imported into the Titan Repository using the Titan Object Import utility.

Example 8-3 Titan Object Import Usage Options

Usage:

```
java net.starfire.titan.Run OBJECTIMPORT
  -s:{import_filename}
  -l:{objectImport_log_filename}
  -c:{objectImport_configuration_filename}
  -command:{titanobjectimportcommand}
```

optional...

```
-nipc
-oatd
```

notes:

nipc - Specifies that no import prechecks are processed during the import which would determine if the object being import exists in the repository

oatd - Specifies that the import file is of type object/attribute type/definition rather than a generic/system object

titanobjectimport - The default behavior for an import of an object
The following are the valid object import types

CREATE - create an object on import; fails if object exists

DELETE - delete an object on import; fails if no object exists

UPDATE - update an object on import; fails if no object exists

UPDATECREATE - update an object on import; if no object exists the object is created on import

The command used to import the XML data produced by the Titan OS/2 Interrogator is:

```
titanobjectimport -l import.log -s somedomain.xml
```

This process can be repeated as needed to “refresh” the Repository data prior to a migration event from OS/2 to Linux or Windows.

8.2.6 Migrating an OS/2 domain

Using Titan, the migration of the OS/2 domain resources to the target platform is accomplished via the launch of Activities.

The prerequisites to the migration step of an OS/2 domain using Titan are:

1. Customization of the migration Activities for enterprise-specific mapping
2. Interrogation of source OS/2 domain producing XML data
3. Importing of domain XML data

Upon completing these steps, the Titan Administrator logs into the Titan Controller from a browser. The following takes the reader through a basic Titan launch scenario using the generic “Migrate OS/2 domain - prompted” activity.

Upon logging into Titan, the Titan Administrator is greeted with the Control Center view of Titan. The top-level Migration Activities are presented here:

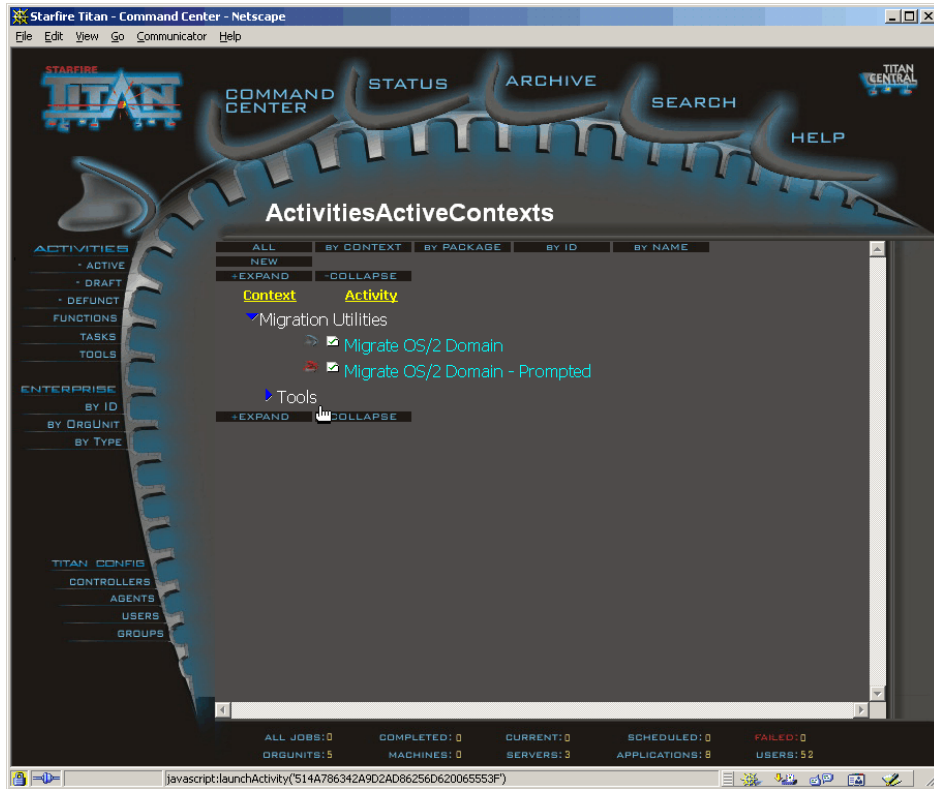


Figure 8-2 Titan Control Center view with migration activities

The launch of an Activity is accomplished by selecting the grey and red icon to the left of the Activity name. In this scenario, the “Migration OS/2 domain - prompted” activity will be launched.

Upon launching the Activity, the Titan Administrator is prompted to select from the OS/2 domains currently available in the Titan Repository:

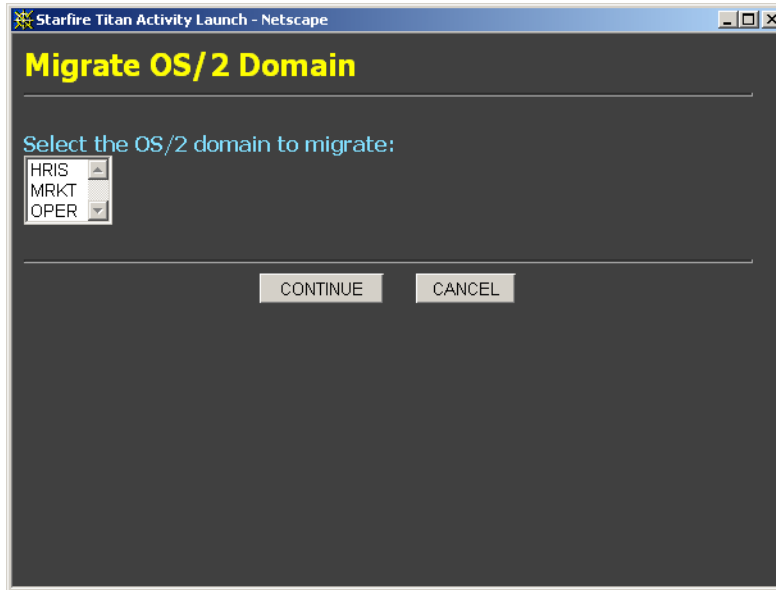


Figure 8-3 Select the OS/2 domain to migrate

Following the selection of the OS/2 domain to migrate, the Titan Administrator is prompted for the target platform. It is likely that most enterprises will select a single platform for all migrations rendering this prompt unnecessary. This is displayed to illustrate that a single set of Activities can target either the Linux or Windows platforms. For this scenario, the Linux platform will be selected.

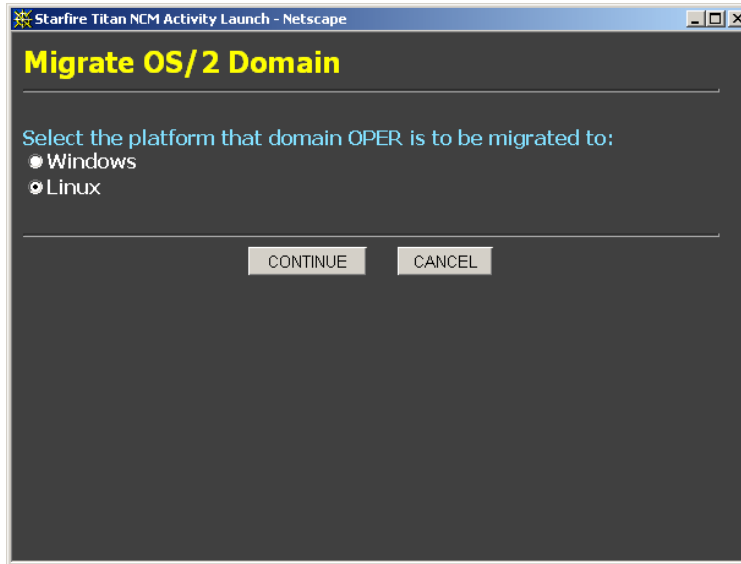


Figure 8-4 Select the target platform

The next step is to select the target Linux server(s) for the migration. This is referenced as an OrgUnit and correlates directly into the Branch LDAP or Active Directory organizational units model presented in this redbook. For this scenario, the TDLINUX OrgUnit will be selected:

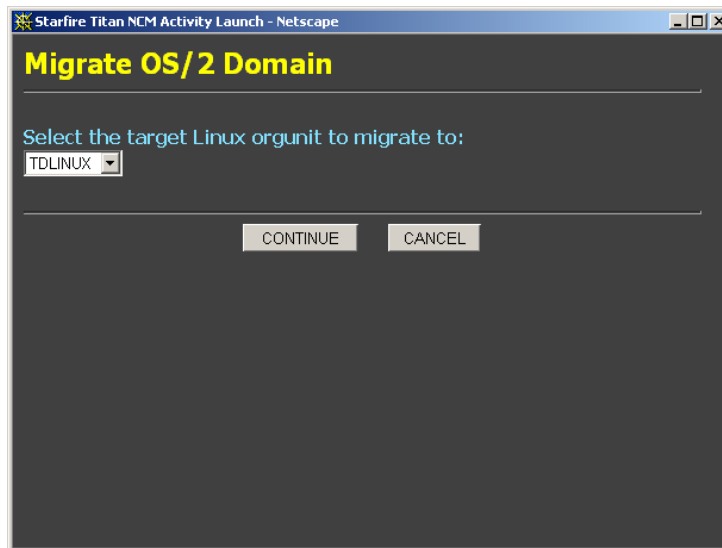


Figure 8-5 Selecting the target Linux OrgUnit

Following this, the OS/2 domain object types to be migrated can be selected. Once again, this is likely to be pre-determined rendering this prompt unnecessary, but is provided here to illustrate the flexibility of the Activity scripting and process flow. At this point, it is possible to select only Users and Groups to be migrated. Likely, all object types would be migrated and are selected for this scenario.

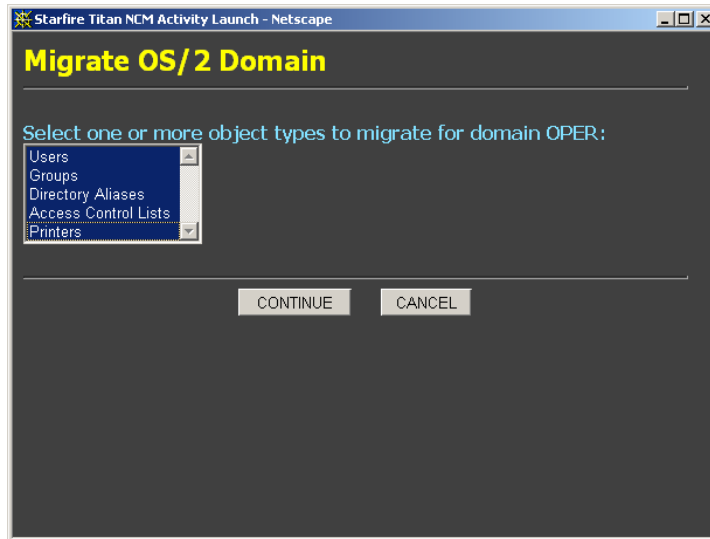


Figure 8-6 Selecting the OS/2 domain object types to migrate

For additional granularity beyond the previously chosen object types, the base migration Activity provides the ability to select all objects or provide the Titan administrator the option to individually select from the currently defined OS/2 objects.



Figure 8-7 Selecting which objects of each type to migrate

Following the “Selected” choice in this scenario, each of the previously selected object types are processed to allow for individual selection of the OS/2 domain objects. Selection of the User objects for migration:

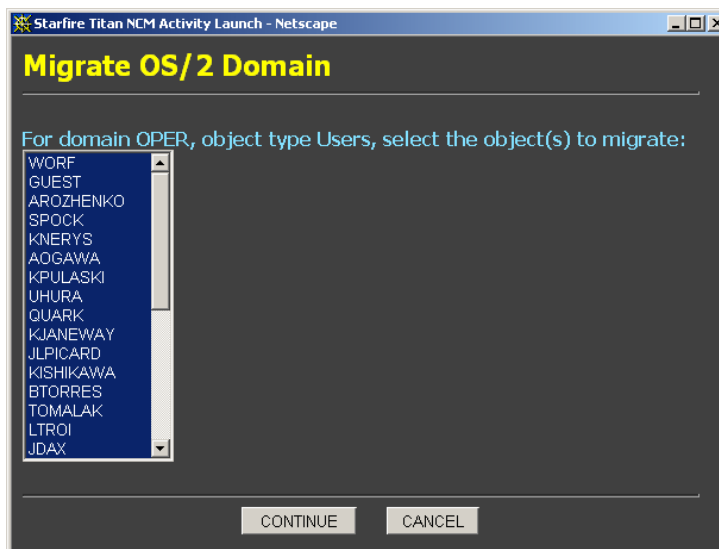


Figure 8-8 Select the user IDs for migration

Then selection of the Group objects for migration:

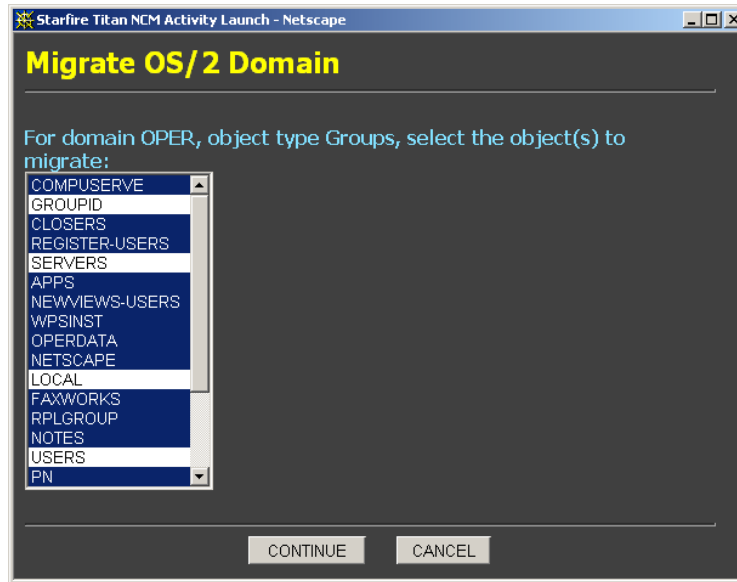


Figure 8-9 Select group IDs - not all groups are selected for migration

Then the selection of directory aliases for migration:

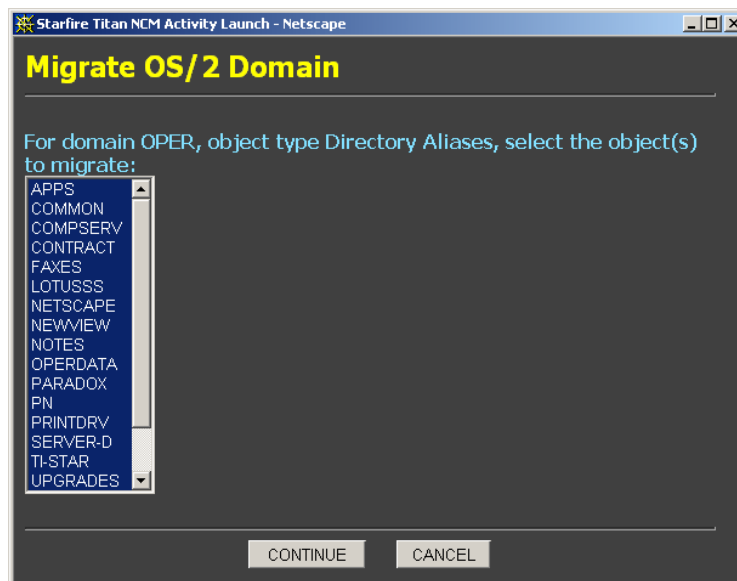


Figure 8-10 Select the directory aliases for migration

And finally, the selection of the printer objects for migration:

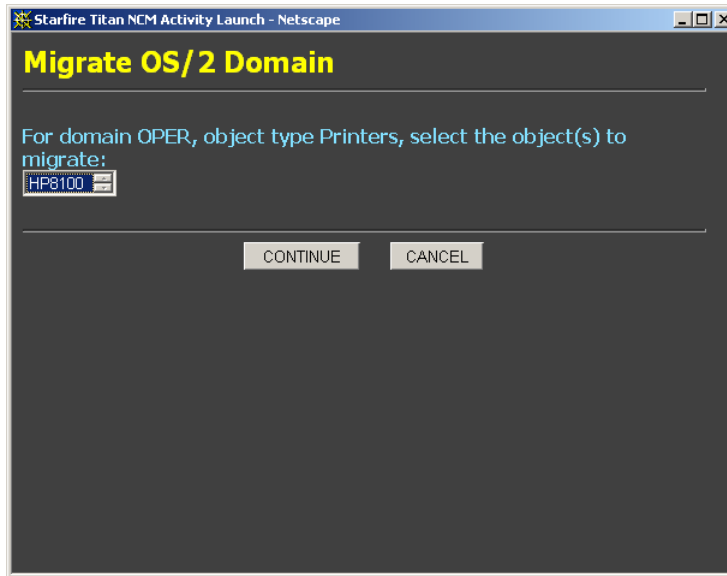


Figure 8-11 Select the printer aliases for migration

This concludes the prompting of the Titan Administrator for information for the migration launch. The Titan Controller then computes the transactions to create the objects as required on the target platform for a successful migration. These transactions are calculated and stored in Titan Jobs for execution.

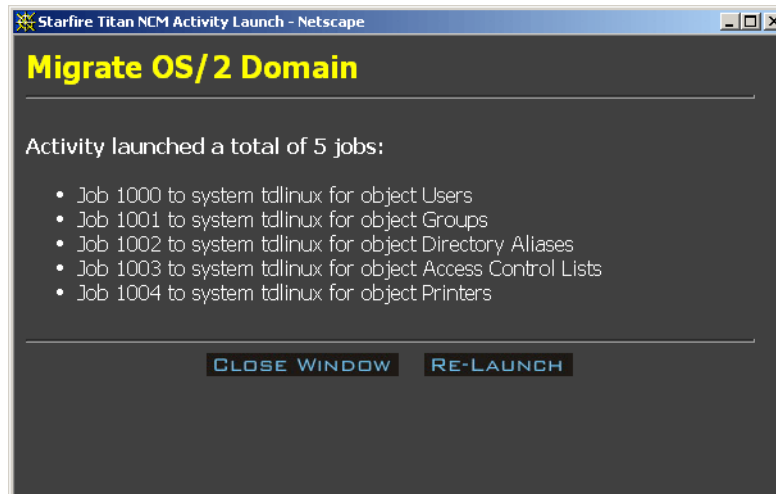


Figure 8-12 Completed migration Activity Launch Summary

The Titan Controller then directs these Jobs for execution at the target system's Agent. In this case it's a system with the hostname `tdlinux`. Each Job is executed, transaction by transaction at the Agent, creating the objects, attributes, and related resource information on the target system.

A final task of each of the Jobs is updating the Repository for the new system's objects and resources. This is accomplished via the Object Modifications contained in the Jobs. These are created at launch time and are executed updating the Repository at the successful completion of the related transaction. Additional details on the Object Modifications is found in the *Starfire Titan Designer's Guide*.

8.2.7 Starfire Titan during and after migration

The migration of OS/2 domain resources can be accomplished quickly and consistently with Starfire Titan. Furthermore, the customization of the migration procedure is flexible within the Titan Activities. Titan's migration process can be extended well beyond the migration of only OS/2 domain resources making a complete migration process possible and the integration into the new platform and applications unified and simple.

Starfire Titan will:

1. Enable the migration process
2. Simplify multiple platform operations and administrative tasks

Enabling migration

Starfire Titan provides a toolkit for the migration of enterprise systems from OS/2 to Linux or Windows platforms. With the customization of the base activities provided, the migration process is simplified as well as completed consistently. If desired, the Activities can be extended to accomplish the data migration step also.

Operational and administrative tasks

Using Starfire Titan, the process of migrating from OS/2 to Linux or Windows can be simplified, faster, and more consistent. However, Titan is not merely a migration tool.

Titan provides a simplified interface to the operations and administrative tasks for systems. With a multiple platform distributed environment, the complexities and challenges include:

1. Staff availability and training limitations
2. Incomplete change coverage in distributed environments

3. Inconsistent systems changes

Starfire Titan can address these and other issues for an enterprise:

1. Titan Activities contain the skills to complete a task - available any time to any one
2. Titan calculates which systems require a change and targets Jobs to all systems
3. Titan changes systems as the Activities are designed, each time on each system

The real value and power of Titan begins when the migration is complete. Beyond the migration activities, Starfire Titan can improve daily operational and administrative task availability, effectiveness, and completeness.

8.3 6PAC Network administrative tools

6PAC Consulting offers a broad range of applications and utilities for the IBM OS/2 LAN Server family, Windows NT and Windows 2000 Server, and LINUX. Having its basis of business in consulting, supporting and project implementation, all solutions emerge from the demands and concepts of customers. Starting as a custom built application for a few installations, some of these advanced to a generalized product. 6PAC focuses in their development process on supporting the daily work of administrators or IT departments providing solutions that are smart, small, and easy to install and to maintain. In the following sections we will describe some tools that target specific environments:

- ▶ Toolset for the migration of OS/2 LAN Server domains to Windows 2000
- ▶ Extending the functionality of Windows 2000 Active Directory or LDAP Servers to provide DCDB features like aliases and public applications for users
- ▶ Support for managed, unattended installation of clients and servers using CID as a model

Each solution is introduced providing only a general overview. For further information including current releases and all other non-migration relevant utilities for Windows and OS/2, 6PAC will be pleased to provide answers. Please see their Web site at <http://www.6pac-ag.com> to find contact information, solutions, and services.

8.3.1 Logon Script Manager offering

One of the still unresolved issues in migrating a domain from OS/2 to Windows is providing logon assignments for users. The Windows NT and the Windows 2000 Active Directory domain model do not provide any mechanisms comparable to IBM's Domain Controller Database (DCDB). With the DCDB, administrators have been able to define resources within the domain and assign these to users. Using nicknames (*aliases*) instead of absolute UNC names, the resolution of resources to distinct servers is postponed to the time users log on. For that reason aliases ease the management and migration of file resources by minimizing the impact for the user. The administrator only needs to update the alias definition keeping the logon assignments untouched. Introducing the Distributed File System (DFS) Microsoft starts to provide a similar domain wide valid name space that is not bound to physical server names and resources. Because DFS is right now only available for clients running Windows operating systems, it is not always the right answer for migration scenarios.

6PAC offers with the Logon Script Manager (LSM) a platform independent approach for logon assignments. The basis of the current release 2.1 was initially developed for a migration project for a specific company and therefore has its main focus right now on Windows 2000 Active Directory domains, supporting Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, and OS/2 as logon clients. LSM implements an easy to manage method of mapping network resources within the logon script. Because of its flexibility LSM can adapt to any concept of logon scripts. You can use a central script for all users, branching to a user specific additional script or define an individual script for each user. Instead of using Windows Explorer and editors, you have a graphical interface to manage the assignments. From the users view, LSM smoothly integrates into existing logon procedures giving a visual feedback while connecting assigned resources.

In the migration scenario in 4.5.5 "Logon assignments" on page 132, we already introduced the concepts using LSM. Each user is assigned to a global logon script (LOGON.CMD) that is stored in the user's attribute `scriptPath`. When logging on, this script is executed and branches to a second, user specific script containing the logon assignments. "Client view" on page 306 contains an example of the structure of these logon scripts. Using this workflow, LSM provides features that the following advantages among others:

- ▶ In the domain, only one logon script exists providing all functions needed on the clients. Logon assignments are thereby separated and a consistent environment is ensured.
- ▶ The user specific part is held in a small second file, simplifying documentation and delegation of logon assignments to another department without providing administrative rights.

- ▶ Because LSM retains the simple batch language and uses the NET USE syntax to define logon assignments, you always keep backward compatibility even if LSM is removed from your domain.
- ▶ Using environment variables where ever possible, LSM implements aliases for server names with its current release, providing full alias support in a future release.

LSM consists of an add-in for the Microsoft Management Console (MMC) extending the Active Directory console for Computers and Users, and scripts and utilities for the clients. Deploying and implementing LSM in your domain is quite easy and will be described in the following sections.

Administrators view

After registering the extension DLL, the administrator can use its usual management console to access the user logon script. Additionally you need to modify the configuration file `logonscriptmanager.ini`. All settings are stored in this INI file which compared to the registry provides a domain wide consistent shared configuration for all administrators. Example 8-4 shows an example of that file.

Tip: Because of the very graduated access profiles in Windows 2000, administrators can grant access to these profiles even for operators or the help desk staff by adding access rights for the directories where logon scripts are stored (usually this is a subdirectory of the NETLOGON share). Create a second directory (for example, LSM) within this share to store the shared ini file and the registered DLL to gain a consistent environment.

Example 8-4 logonscriptmanager.ini

```
[General]
FileServer=%FSRV_001%,%FSRV_002%
FirstDrive=E
IncludeHiddenShares=1
AllowLogonScript=0
AllowUserScript=1
ListSharesOnly=0
;LogonServer=

[Logon]
StartMagic=:START_FILENETUSE
EndMagic=:END_FILENETUSE
Template=logon.tpl

[User]
StartMagic=:START_FILENETUSE
EndMagic=:END_FILENETUSE
```

```
Template=user.tpl
ScriptPath=USERS
```

```
[Translation]
%FSRV_001%=WINDC
%FSRV_002%=WINMEM
```

The ini file contains four sections with the following parameters. We describe only a subset of them in this short introduction:

General	This section contains settings for the general behavior of the extension.
FileServer	With this keyword you specify the name of servers that contain file shares and are used within the management console. Having dozens of servers in a domain, you can use this feature to minimize the list. In the example you can find “real” server names, and “aliases” using environment variables that have to be defined in the translation section.
IncludeHiddenShares	To hide a share from the network neighborhood you can add a dollar sign (\$) to the name. The share is invisible but still assignable to users. By default Windows 2000 creates some of these hidden shares for administrative purposes like C\$ or ADMIN\$ but some customers use this feature to hide home directory shares. If you use hidden shares for user assignments, you may display them by setting this parameter to 1.
AllowLogonScript	With this feature you allow the administrator to use the logon script stored in the scriptPath attribute to assign network resources. If this parameter is set to 0, all settings in the Logon sections are disabled.
AllowUserScript	With this parameter you can enable LSM to support user specific assignment files in a separate directory. This is the recommended configuration for using LSM. If this AllowUserScript is set to 0, all settings in the User sections are disabled.
ListSharesOnly	Legacy clients like Windows NT or OS/2 are not able to net use subdirectories. The assignment of a drive letter is restricted to a server’s share name. To help the administrator build a consistent environment you may consider setting this parameter to 1.

LogonServer	You can define your preferred domain controller that is used to access the NETLOGON share. This may be helpful if you only grant special access to this share on a few domain controllers, while the others remain as read only.
Logon / Users	These two sections describe parameters needed to process related files and directories.
StartMagic	LSM looks for this keyword in a separate line to start parsing the following lines as logon assignments using net use commands.
EndMagic	Related to the StartMagic, this is the corresponding end mark for LSM. Only NET USE commands between these two keywords are treated as manageable assignments.
Template	If there is no logon script assigned to a user, LSM will use this template file to create one. Because these files may differ significantly you can specify one for the Logon and another for the users section.
ScriptPath	For the user specific scripts, LSM needs to know where these files are located. Usually you specify a relative path name, as a subdirectory of the NETLOGON share.
Translation	This section contains an translation of server aliases to their real names. The same translation is necessary for the users within the logon script using the SET command.

After configuring LSM the administrator starts his management console for Active Directory Users and Computers, and adds the Logon Script Manager extension. LSM seamlessly integrates into the provided management utilities by Microsoft and therefor offers its services by extending the task menu for user objects. Selecting the menu item named Logon Script, the administrator is provided a dialog wherein he can add, change or delete logon assignments for the selected user. Figure 8-13 on page 306 shows this dialog. If the configuration file permits access to more than one of the two selectable script files for Logon and Users, the administrator can select one of these for editing.

The combo box includes the complete path name to help identifying the correct file. Below is the list of current assignments in this file. Each line correlates to a NET USE command in the script file. Besides all other options typical for graphical interfaces, LSM provides a filtered and therefore more convenient way to present file resources within the domain. The administrator defines which servers are listed in the resource tree view. Additionally he can use the concept of aliases introduced before, instead of real server names. LSM then extends the

system branch of the tree and offers the share level including hidden shares, if specified. Last but not least, for a legacy-free environment, subdirectories are also displayed allowing the administrator to assign resources like \\windc\lanhomes\wynand to a local drive letter.

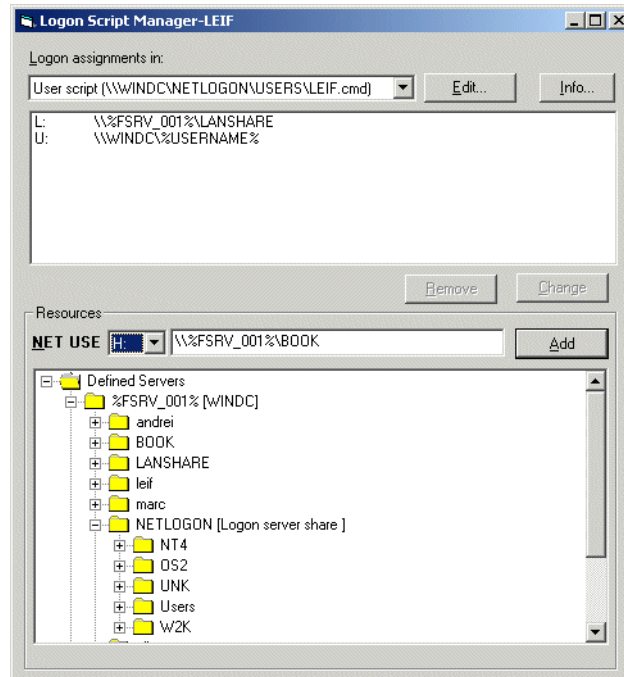


Figure 8-13 Administration of logon assignments with LSM

After completing the administration LSM generates a script providing the appropriate NET USE commands as listed. The resulting CMD file can be found in Example 8-5.

Client view

The second view of LSM is that of the user. During the logon process the client executes the global logon script for the user that branches to the user specific scripts such as the one listed in Example 8-5:

Example 8-5 Example logon script LEIF.CMD for LSM

```
@ECHO OFF
REM *****
REM File      : LEIF.CMD
REM Version  : 2.0
REM Date     : 29 Jun 2003
REM Author   : Leif Braeuer (6PAC Consulting AG)
```

```

REM
REM Description:
REM User specific logon script of logon assignments
REM
REM *****
ECHO Assigning network devices...
IF NOT EXIST %0\..\..\LSM\%OS%\LSMUSE.EXE GOTO START_FILENETUSE
%0\..\..\LSM\%OS%\LSMUSE.EXE %0
GOTO END_FILENETUSE

:START_FILENETUSE
NET USE L: \\BDC\LANSHARE
NET USE U: \\PDC\LEIF
:END_FILENETUSE

:START_PRINTNETUSE
:END_PRINTNETUSE

```

The supplied script template searches for the client part of LSM called **LSMUSE.EXE**. This program substitutes the functionality of the operating system command NET USE, but provides a graphical interface for the user. The environment variable %OS% is used to distinguish the client operating system providing different executables of this utility for OS/2 and Windows. If the script does not run in LSM environments, this utility cannot be found and the script jumps to the label START_FILENETUSE, processing the assignments with the fallback functionality using the NET USE command. Otherwise LSMUSE receives the calling script as input and parses the lines between the magic tokens. In our example these are START_FILENETUSE and END_FILENETUSE. While processing the assignment the program displays a status window like the one shown in Figure 8-14.

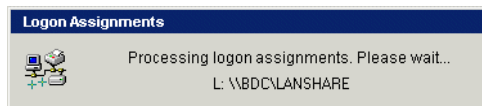


Figure 8-14 Status windows of LSMUSE.EXE

LSMUSE does not provide an error message for each failing connection, but lists failing assignments as a summary. If all connections succeed, the program exits gracefully. Figure 8-15 shows an example of this message box stating that two logon assignment could not be established. The user can follow his given instructions calling the help desk or the administrator for further assistance.

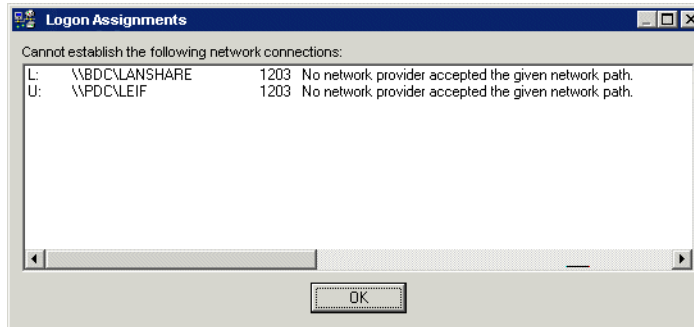


Figure 8-15 Summary of failed connections from LSMUSE

Planned future enhancements

6PAC is planning additional features in the next release of LSM (currently planned for late 2003). This release migrates all definitions for the user into Active Directory or any other LDAP compatible directory. Administrators will be able to define aliases at any level of the directory tree, defining local, branch wide or domain wide resource aliases. With the provided schema extension, user accounts point to the distinguished name of these aliases and map them to devices or mount points for LINUX clients. Additionally a client utility available for OS/2, Windows and LINUX will provide the resource mapping to a given drive letter, LPT port, or mount point for LINUX.

Features at a glance:

- ▶ Full LDAP or Active Directory integration.
- ▶ Schema extension providing a new alias class for file and print resources.
- ▶ Schema extension providing new attributes for user classes to assign aliases to local devices or mount points.
- ▶ Client utility to process these objects for OS/2, Windows, and major LINUX distributions.

8.3.2 OS/2 to Windows migration tools

Many of the concepts described in Chapter 4, “Migrating OS/2 Servers to Windows 2000” on page 87 are based on experience the consultants and engineers of 6PAC gained in the last ten years. As a result of these efforts a toolset is available that assists in all phases of migration or consolidation scenarios. The OS/2 to Windows migration toolset consists of templates, documents, scripts, and work flows accompanying the project team through the whole migration. As the result of customer demands it focusses on the following paths:

- ▶ OS/2 LAN Server 3.x and 4.x to IBM Warp Server for e business
- ▶ OS/2 LAN Server and Warp Server versions to Microsoft Windows NT domains
- ▶ OS/2 LAN Server and Warp Server versions to Microsoft Windows 2000 Server Active Directory
- ▶ OS/2 LAN Server and Warp Server versions to Microsoft Windows Server 2003 Active Directory
- ▶ Microsoft Windows NT domains to IBM OS/2 Warp Server for e-business
- ▶ Microsoft Windows NT domains to IBM OS/2 Warp Server for e-business
- ▶ Microsoft Windows NT to LINUX distributions like Red Hat or SuSE running Samba 2.x
- ▶ OS/2 LAN Server and Warp Server versions to LINUX distributions like Red Hat or SuSE running Samba 2.x

Note: Having Samba version 3.x available on the major distributions, 6PAC will add migration paths for OS/2 and Windows NT and Active Directory domains to the new Samba release including the necessary LDAP integration.

8.3.3 Network application tools

Another unresolved topic in a one-to-one migration scenario to Windows or Samba based domains are public applications. After assigning application definitions to a user, these applications are selectable in a special Workplace Shell folder named Network applications for OS/2 clients, or for Windows clients using the IBM Primary Logon Client for Windows in a provided window.

6PAC supplies a Network Application Toolset (NAT) that uses a very similar approach to the one used for the Logon Script Manager introduced in “Logon Script Manager offering” on page 302. Embedded into a logon script, utilities populate a special purpose folder on the users desktop. The current release of NAT uses INI files to store the application definitions in a server share the client utility has access to (using the NETLOGON share is recommended). Like the DCDB, for each user an assignment table exists from which the client retrieves the list of applications. Having the list and all necessary parameters, the client clears the application folder and populates it with the currently assigned applications. Supporting different client operating systems, there are several additional attributes available to influence the application.

- ▶ Operating system (Currently DOS, OS/2, Windows 9x, Windows NT and Windows 2000)

- ▶ For OS/2 all parameters for DOS and Windows emulation.
- ▶ Icon Files for Windows and OS/2.
- ▶ Working directory, parameters.

If an applications type is not suitable for the clients operating systems, NAT ignores that definition

Additionally 6PAC extended the functionality providing group based application assignments, such as those already known within Citrix Metaframe environments. To use this feature, you need to define a group object for each public application. Modifying the membership in these application groups determines the list of applications for the user. In this mode, administrators can use the Microsoft Management Console for Users and Computers to manage the public application assignments.

In a future release of NAT, 6PAC plans to announce LINUX support and LDAP or Active Directory integration, including:

- ▶ Full OpenLDAP, Active Directory or third party LDAP server integration.
- ▶ Schema extension providing a new application class for public application
- ▶ Schema extension providing new attributes for user and group classes to assign public applications.
- ▶ Enhanced client support including OS/2, Windows 98, Windows 2000, Windows XP, and major LINUX distributions.

Tip: Used on Windows clients, the folder %USERPROFILE%\Startmenu enables you to populate the public applications into the start menu of Windows. The target folder is configurable per operating system.

Restriction: The Network Application Toolset is not designed to support Workspace on Demand environments. Right now it focuses on the classic fat client systems.

8.3.4 Unattended Installation Manager

Migrating client and server systems to the new Windows or LINUX environment, customers still need concepts and utilities to manage unattended installation of operating systems, applications and services. IBM Netview Distribution Manager (NetviewDM) or Tivoli Software Distribution are products within this category. If these products are not used, other solutions are available. For instance, when deploying back-office systems and servers, some administrators use procedures

that are based on scripts, using response files for parametrization and providing a quick, CID-like solution. 6PAC Unattended Installation Manager (UIM) deals with these response files and templates for creating all customized files and scripts needed for the installation of a new system. Because it only works with text files, UIM is very flexible supporting CID-enabled environments. As long as the target system supports text files to run an unattended installation, UIM will manage and supply the necessary files. UIM does not replace existing procedures but eases the use and management of these solutions.

Starting with UIM you need to define Installation templates. These templates reside in a directory containing all files for a single installation. These may include static text files that do not differ between installations for distinct computers, and changeable text files, that contain variables like the name of the machine or a TCP/IP address. Additionally the template directory contains a parameter file uim.ini that contains links to additional packages. These additional packages may be divided into single or multiple choices, that are represented as subdirectories. Figure 8-16 shows an example of packages available for the template Windows 2000 Advanced Server. By selecting the packages needed for installation, naming this selection with a name identifying the system, you can save these system profiles for later usage. As you can see on the right side of the window, some packages contain variables, that need to be defined. All variables found in the package files and shown here can be edited. If response files use the same variable name, the administrator only needs to enter the value once. After selecting the button create, UIM generates the install set, using all selected templates, replacing all variables with the entered values and stores this set in a subdirectory named for the package.

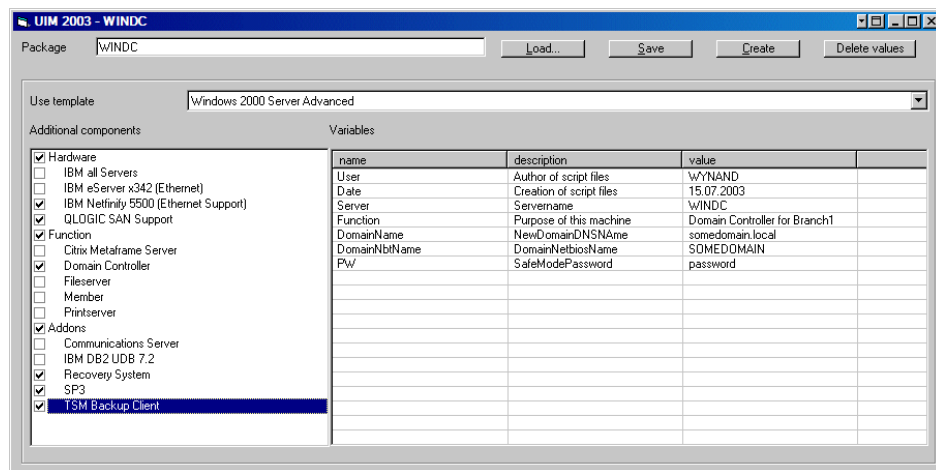


Figure 8-16 Main window of UIM

Looking at Figure 8-16 the administrator is about to generate an install set for the system WINDC. As this system is planned to be a server running Windows 2000, he selected the appropriate template as the basis. This template offers three included packages. The hardware package supplies configuration files needed for driver support (for example, the IBM ServeRAID™ adapter configuration). Additionally UIM found a package defining the role of the system. Selecting the feature **Domain Controller** will include all files needed for a domain controller promotion in the install set. Last but not least the administrator can select additional applications and services to this system. After selecting these packages and features, UIM found references to the following variables in these packages.

User	The name of the user is provided automatically to provide a variable usable to document the author of the install set.
Date	UIM by default uses the current date for documentation purposes.
Server	In our example this variable contains the computer name for the install set.
Function	This is a description used in the scripts and files for documentation.

Selecting the package **Domain Controller**, the following three variables are needed:

DomainName	The DNS name of the Active Directory, in our example we used the name <code>somedomain.local</code> .
DomainNbtName	Additionally, for domain controller promotion, the NetBIOS name of the domain is needed. In our example it is <code>SOMEDOMAIN</code> .
PW	To restore the Active Directory databases in disaster recovery situations a safe mode administrators password is set here.

8.4 Lieberman & Associates

The tool described in this section is the Migration and Synchronization Wizard by Lieberman & Associates. The tool can be used to migrate OS/2 domains to Windows NT or Windows 2000 domain(s). It is designed to handle the migration of large numbers of users and groups in a few hours. Because of the potential variety of an IT environment, this tool is designed to be very flexible. While the Migration Wizard is designed to move your domain information from LAN Server to Windows 2000 Server, it cannot, by itself, resolve all conflicts that may occur when two or more domains are merged. In those cases where domain definitions

are in conflict, the tool will flag the problem and log it for review. It is your responsibility as a LAN Administrator to look at the flags and take steps to resolve the conflict. An example of conflict would be the LAN server user name that exists as a group in the Windows 2000 domain. Another conflict would be insufficient room on the target system to take all the files from an existing alias on the LAN server domain. Each copy of the Migration Wizard is licensed on a per-user count basis sold in increments of 100/250/500 users. Lieberman & Associates owns the Migration Wizard. By using the software, you agree to be bound by the terms of the agreement. Additional information and ordering details are available on the Internet at:

<http://www.lanicu.com/cross/lstnt/index.htm>

Attention: The Lieberman tools work on both Windows NT and Windows 2000 systems. Note that the migration wizard will also work with Microsoft Server 2003. For proper operation on Windows 2000 and 2003, NTLM support must be left enabled. If you select a security policy that disables support for LM hashes, the imported accounts will not work. Before importing accounts into a 2000/2003 system, you must set password policies as follows: password complexity=disabled.

Note that some of the text and directions below may interchange NT and Windows 2000 due to what may appear on various screens of the Lieberman tools.

8.4.1 Migration procedures

This section focuses on the methods required for the actual migration process. Each step will be explained in detail. The migration procedures include the following steps:

1. Install the OS/2 and Migration Wizard programs.
2. Set the Windows 2000 domain definitions.
3. Import OS/2 LAN Server definitions.
4. Resolve or mapping LAN server settings to the Windows 2000 domain.
5. Export data to the Windows 2000 domain.

8.4.2 Installing the Migration Wizard and preparation

The importation of data from LAN server is a multi-step process. The first step is to run the exporter program (LU.EXE) on the LAN server domain to turn all of the LAN server information into a human readable ASCII file. This file is then read into to the Windows 2000 program called Import to be used to build the appropriate Windows 2000 domain entries. Lieberman supplies you with a limited version of the LAN server exporter, or you can use a full version that you

might already have from your purchase of LAN Intensive Care Utilities for IBM LAN Server.

Creating the export file from OS/2 LAN Server

1. Open an OS/2 command session.
2. Log on as an administrator to OS/2 LAN Server.
3. Create a directory on your OS/2 LAN Server machine with the name LU.
4. Copy the ICUDEMO.ZIP and UNZIP.EXE files to the LU directory.
5. From the LU directory, unzip the ICUDEMO.ZIP file using the command:
UNZIP ICUDEMO.ZIP
6. After registration, create the export file of your OS/2 LAN Server using the command:
LU -USER -PASS -GROUP -ALIAS -APP -ACL -V -O:domain.icu
DOMAIN.ICU is the output filename. We have used DOMAIN.ICU for this example.
7. When the LU.EXE has completed its operation on OS/2 LAN Server, copy the output file (DOMAIN.ICU in the example) to the Windows 2000 Workstation or Server running the Migration Wizard.

Patching Windows 2000 Domain Controllers

Included with this wizard is a hot fix for Windows 2000 systems. You must install this hot fix on all your Windows 2000 domain controllers to correct a bug in the Kerberos/NTLM provider selection logic.

Attention: The hot fix is required on Windows 2000 domains running in Native Mode and that do not have Service Pack 3 or greater. If you are not planning on using Native Mode or have Service Pack 3 or later, you DO NOT NEED TO APPLY THIS HOT FIX.

Why the patch is needed

Windows NT and Windows 2000 do logon authentication using a cryptographic hash (one-way function that translates the password into a 16 byte value) of the password you entered. From Windows 2000 or Windows NT machines, new accounts produce two hashes: LM Hash (DES hash) and NT Hash (MD4 hash). When an account is migrated from IBM LAN Server/Warp Server, or the account was created from a downlevel client (Windows 95, 98, ME, OS/2) only the LM hash is created. Due to bug in Windows 2000, if a Windows 2000 client attempts to log on with an account that only has an LM hash, the account will not be allowed to log in. In the event log you will see an error indicating that Kerberos

was unable to build a certificate due to a lack of proper credentials. If you try to log in using a downlevel client, the LM only hash will let you log on OK.

The hot fix corrects the bug in Windows 2000 so that the NTLM authenticator is used instead of Kerberos when only the LM hash is available. This is the expected behavior that you obtain when the patch is installed. The fix is needed on all Windows 2000 domain controllers since there is no way of predicting which domain controller will do the logon for a specific client.

How to apply the hot fix

Copy and run the program: q298861_W2k_SP3_x86_en.exe on each of your domain controllers. The file is located in the application directory where the Migration Wizard software is installed.

Upgrading the registry size

You will need to increase the registry size on your local machine to accommodate the extra information of the systems under management by this program:

1. Right-click the **My Computer** icon normally located in the upper left corner of your screen.
2. Select the **Properties** option on the menu.
3. Select the **Advanced** tab (far right tab) on the System Properties tabbed dialog box.
4. Click the **Performance Options** button.
5. Click the **Virtual Memory group Change...** button

The registry space used by the program varies by the size of the migration being performed. The more users and groups being migrated, the more space required. A good starting size would be 64 Mb. When you open the dialog in step 5, add 64 to the current registry size value as the new maximum registry size (MB).

6. Click the **OK** button to save the new registry setting.
7. Click the **OK** button to close the Performance Options dialog.
8. Click the **OK** button to close the System Properties dialog.
9. If the system requires a reboot, click the **OK** button to confirm the immediate reboot of the system.

Installing the Windows 2000 portion of the Migration Wizard

1. Create a subdirectory on your Windows 2000 workstation or server with the name LSNT.

2. Run the NIMPORT2.EXE program (from the diskette or directory you downloaded it to) to install the Migration wizard.

8.4.3 Step 1: Setting Windows 2000 domain definitions

The creation of NT/2000/XP and LS domain definitions by the Migration Wizard creates a section in the local registry of your machine to hold the information. If you do not create a LAN server domain definition, and attempt to import information, the program will be unable to store the information. If you do not create a 2000 domain definition, the program will not know where to send the exported information. Although you will not be exporting information immediately (there are a few decisions that need to be made with the LAN server data first), you still need to define the NT/2000/XP domain or domains before you use the program.

We use the registry of your workstation for speed and reliability. Because the information is stored in your local machine, that same machine must do all of the import and export operations:

1. Start the Migration Wizard utility by running IMPORT.EXE. Click the **Domains** button. A screen similar to that shown in the following figure will be appear.

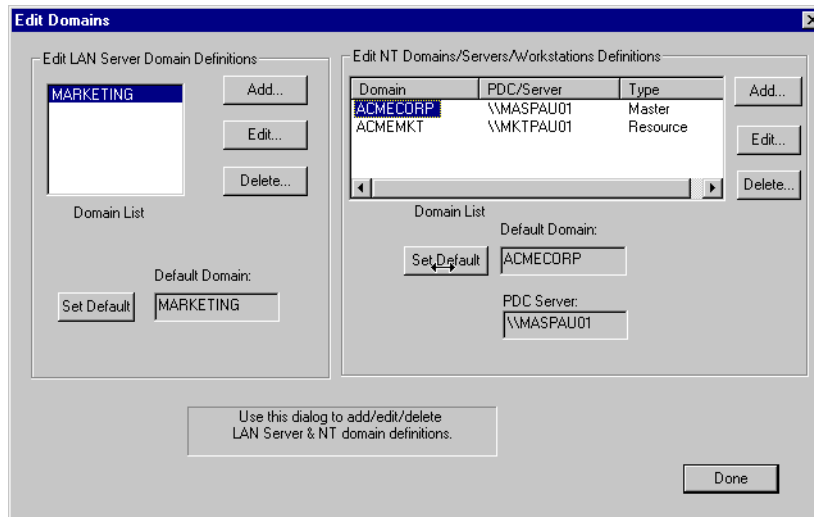


Figure 8-17 Edit domains

The left half is for defining different source LAN server domains. The right side is used to define the destination 2000 domain(s). You must define an NT/2000/XP and at least one LAN server domain. Note: the NT/2000/XP

domain must currently be running so that the primary domain controller can be located.

2. Click the **Add** button on the right side of the dialog box to add a new Windows 2000 domain definition, as shown in Figure 8-18.

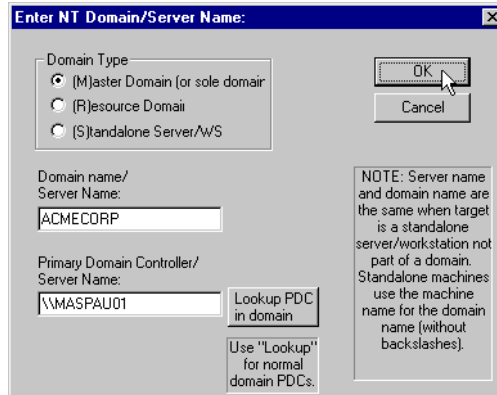


Figure 8-18 Add Windows 2000 domain

3. Enter the name of the domain and click the **Lookup PDC in domain** if this is a domain. If you are exporting to a standalone Windows 2000 machine, then enter the name of the machine (no back slashes) in the Domain name field and enter the server name (with back slashes) in the field normally used for the Primary Domain Controller.
4. Select the **Domain Type**.
5. Enter the Primary Domain Controller Server Name or click **Lookup PDC in Domain**. Click **OK**. Repeat steps 2 and 3 for any additional domain definitions.

If you are having problems locating the PDC (or any domain controller), select the **Domain Type** as **(S)tandalone Server/WS** and enter the name of the domain controller directly.

6. If there is more than a single Windows 2000 domain that you will be using, select the default 2000 domain by highlighting one of the domain entries in the list box and clicking the **Set Default** button. If there is only one 2000 domain, then it will be made the default domain automatically.

The creation of one or more LAN server domain definitions is required so that space is reserved to store the data exported from the LAN server domain. The program supports multiple LAN server domain sections (within the registry of the local workstation) that can be edited and then exported to Windows 2000.

7. To add a new LAN server domain, click the **Add LAN Server Domain** button on the Edit Domains window. Fill in the LAN Server Domain Name. We have

entered Marketing as the LAN server domain in our test case. Your screen should resemble Figure 8-19.

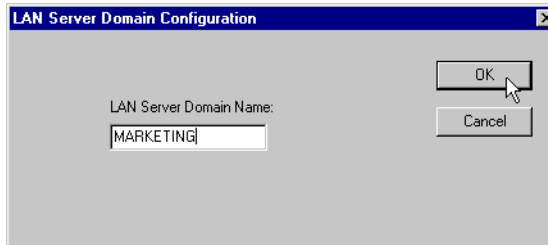


Figure 8-19 LAN Server Domain Configuration

8. Click the **OK** button.
9. When both the LAN Server and Windows 2000 domains have been defined, you can click the **Done** button on the Edit Domains dialog to move to the next step.

8.4.4 Step 2: Import the OS/2 LAN Server data

This step builds the appropriate Windows 2000 domain entries using the LAN Server export file, which is created as described in “Creating the export file from OS/2 LAN Server” on page 314:

1. Start IMPORT.EXE on the Windows Workstation or Server running Migration Wizard.
2. Click the **IMPORT** button.

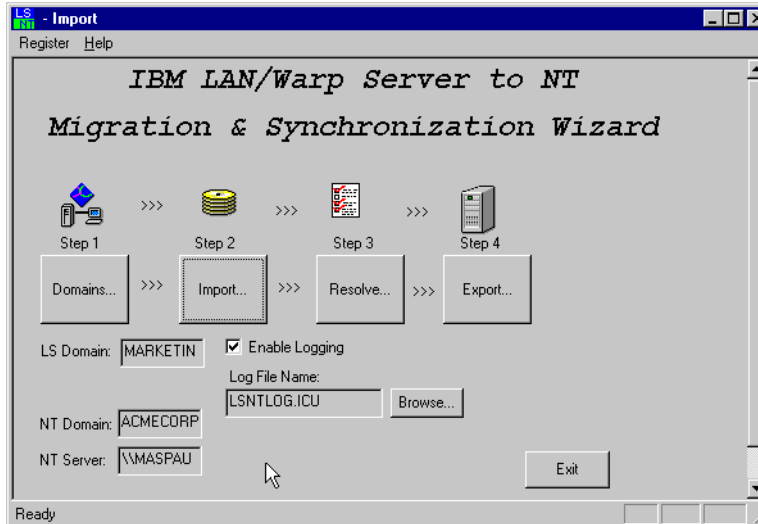


Figure 8-20 Migration Wizard main screen

3. In the Import From Local file to Internal Database, select the appropriate OS/2 Export file (*.icu) using **Browse**.

If your Windows and OS/2 based systems have compatible account systems, you may be able to browse to your previously captured file on your OS/2 system. If not, you should copy the capture file to your Windows machine and browse to the local location where it is stored.

4. In the Import Categories, select **Users**, **Groups**, and **Aliases**.
5. Click the **Start import** button to import the LAN server users, groups, and aliases. Applications are a portion of the Migration Wizard, which is not currently written.

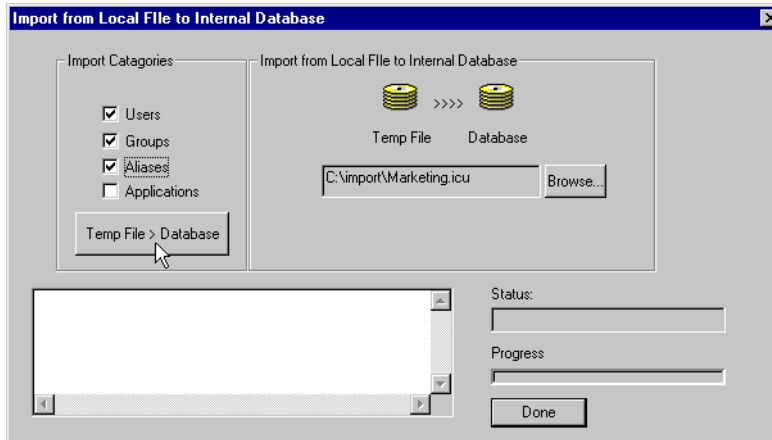


Figure 8-21 Import from Local File to Internal Database

Note: As the import progresses, you will see progress and status. If the application stalls for a long period, there may be an error pop-up dialog box hidden behind the current window. To check the hidden dialog, use ALT+ESC to go through the list of windows.

When the import has been completed, you will see the message in the Status box:

****End Import to Database**.**

6. When complete, click **Done** to return to the Migration Wizard main screen.

The next step is the Resolve Phase where the data is mapped for inclusion into your Windows 2000 domain(s).

8.4.5 Step 3: Resolve

The resolve phase of the Import Wizard is the primary way you decide which users, groups, and aliases are migrated from LAN Server to Windows 2000. Resolve options allow you to select which and when users, groups, aliases, and home directories will be migrated. Because Windows 2000 does not provide automated logon assignments, the Migration Wizard can build logon scripts (CMD and BAT files) which provide the same assignments as LAN Server provides.

The resolve phase has the seven categories as listed below:

- ▶ User accounts

Maintain enable/disable and directory mappings of user accounts. User accounts must be enabled and have their settings modified to reflect their new Windows 2000 hosting.

▶ Groups

Maintains enable/disable and mappings from LAN Server to Windows 2000 group names. This information is needed to correct for the remaining groups from the LAN Server to Windows 2000 standards of group names. The mapping information is also used to move ACLs from LAN Server to Windows 2000.

▶ File aliases

Maintains mapping between file aliases names and UNC's. This allows you to move the location of aliases and data and permissions from LAN Server to Windows 2000 Servers as required.

▶ File ACLs / Directory ACLs

Allows changes to the mappings between LAN Server to Windows 2000 ACLs.

▶ Printer Aliases

Maintains mapping between printer alias names and UNC's. This allows you to move the location of aliases from LAN Server to Windows 2000 Servers as required.

▶ Logon Scripts

The Microsoft requesters do not support central logon assignments. This allows you to specify the location(s) of the scripts and to provide more actions.

▶ Home Directory UNC Shares

Creates public shares for each user's home directory and ACLs

User accounts

1. In the IBM LAN/Warp Server to Migration and Synchronization Wizard click the **Resolve** button. You will see the Resolve Importation Issues window as shown in Figure 8-22.

The Migration Wizard documentation uses the terms **highlight** and **enable** numerous times throughout the procedures. It is important that you understand the terminology. When items are highlighted, it allows you to modify information about those items in the Migration Wizard utility. When items are enabled, it allows the items to be imported into the Windows 2000 registry. You can highlight items and modify their parameters, but that information will not be imported to the Windows 2000 registry for migration unless the items are enabled.

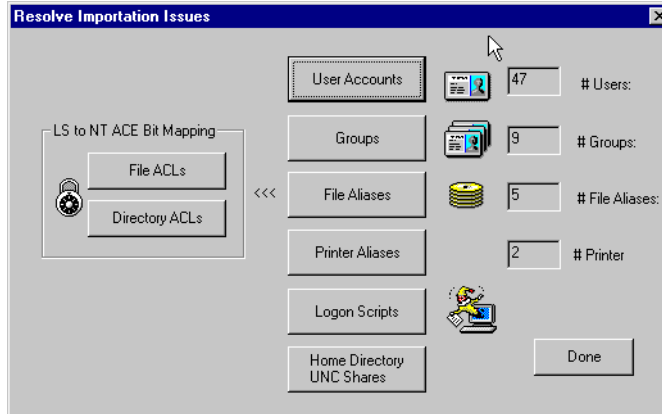


Figure 8-22 Resolve Importation Issues

2. Click the **User Accounts** button. This button causes the Select LAN Server User Accounts to Migrate window to appear as shown in Figure 8-23.

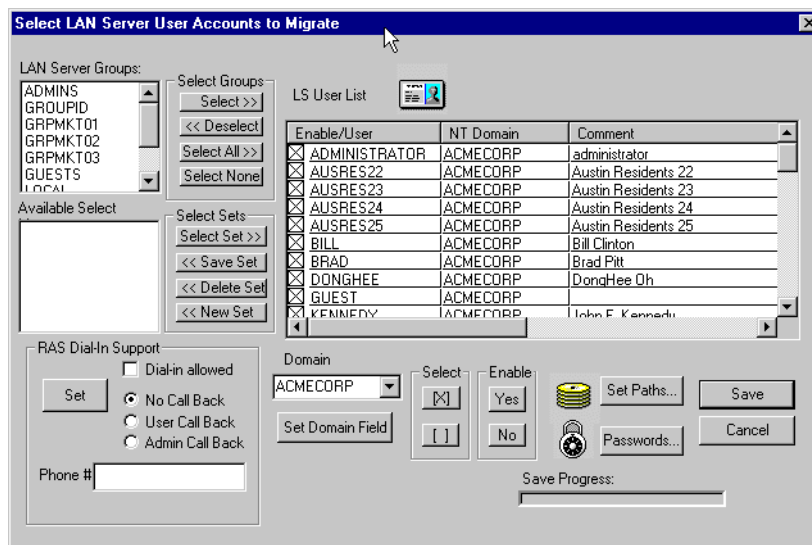


Figure 8-23 Resolve User Accounts

3. In the Select LAN Server User Accounts to Migrate window, enable and disable user accounts for importation to the Windows 2000 Server. Accounts must be enabled to be imported to Windows 2000. To enable or disable highlighted user accounts, use the **Yes** or **No** button in the Enable box. You can double-click on a user entry to flip the enable state of an account. You can

select user accounts individually or by group. To make a change to one or more user accounts, you will need to highlight one or more.

4. If you want to select user accounts by groups, select the group in the LAN Server Groups lists, and click >>. (The >> button selects the highlighted group members.) To deselect, click <<. (The << button deselects the highlighted group members.) This and the following paragraphs about selection are used throughout all the sections under the Resolve step.
5. By highlighting a number of users and then selecting the **New Set** button, you will be asked to enter your own name for this grouping. This grouping is only used during the migration to help you select users more efficiently as opposed to manually selecting them each time.
6. If want to you select all the user accounts, click the **All** (all highlighted). If you deselect all the users, click the **None** (removes highlighted). You can select all the enabled users by clicking on the **[X]** button, and also the opposite, select the disabled users by clicking on the **[]** button.
7. You can add RAS Dial-In Support as shown in Figure 8-23. If you have a RAS Server in your Windows 2000 Domain, you can specify the RAS Dial-In support by user. To enable RAS, highlight the users, and set the Dial-in allowed or Call Back option in the RAS Dial-In Support panel. Users must be enabled before highlighting.
8. To set changes in the RAS Dial-In Support box, click the **Set** button.
9. Each user account has its own information. You can set the paths by clicking the **Set Paths** button. Before clicking the **Set Paths** button, you must highlight the user account(s). Follow the procedures in “Set Paths” on page 323.
10. To set passwords for your users, select one or more user accounts and click the **Password** button. Follow the procedures in “Password” on page 326.
11. After completing these steps, to make the changes click the **Save** button.

Set Paths

After clicking the **Set Paths** button, you will see the Path Mapping window as shown in Figure 8-24.

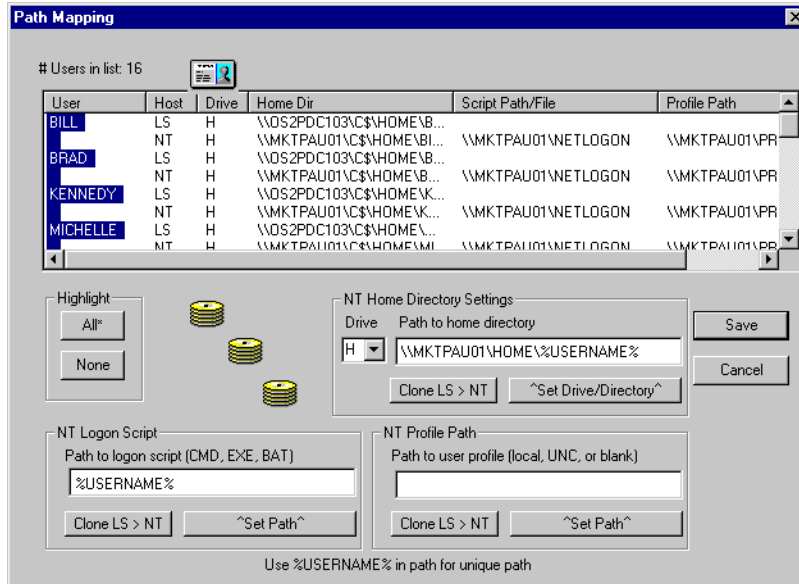


Figure 8-24 Path Mapping

1. Set the drive and path of Home Directory for each user in the WNT Home Directory Settings box.
2. When complete, click the **Set Drive/Directory** button. Or, you can clone an existing OS/2 home directory value for a Windows 2000 by clicking **Clone LS > NT**.

You generally cannot use LS paths because they use the root administrator shares (that is, C\$) within their paths. In Windows NT and about, users are forbidden from using these shares. Consequently, you will either want to created fixed home directory shares (feature provided within this product) such as: \\SERVER\USERNAME, or use a relative path share such as \\SERVER\USERS\USERNAME.

Note: For OS/2 workstations only, you have two choices: leave it blank or enter a correct UNC value here. However, the values you entered will have no impact on the mapping of the OS/2 home directories. The actual home directory mapping must be done through an embedded NET USE statement in the “resolve logon script” part described in “CMD file settings” on page 341:

- ▶ By leaving it blank, the Windows 2000 user profiles will not contain the home directory values. Therefore, Windows 2000 administrators should be aware of this whenever browsing these user profiles. In addition, you cannot refer to some variables - %HOMEPATH%, %HOMEDRIVE%, %SHOMEPATH% - in the logon script described in “CMD file settings” on page 341.
- ▶ By setting this home directory path value here, the Windows 2000 user profiles will contain the correct home directory values. Therefore, you can refer to the variables - %HOMEPATH%, %HOMEDRIVE%, %SHOMEPATH% - in the logon script described in “CMD file settings” on page 341. However, this will cause a “harmless” error NET8191 when users log on to the Windows 2000 domain from OS/2 workstations through a logon script created in the 8.4.8 “Logon scripts” on page 337.

For instance, below is the NET8191 error while running a logon script to map the home directory M: during the OS/2 logon process:

```
[C:\]NET USE M: \\MASPAU01\HOME1
The command completed successfully.
NET8191: Your home directory could not be set up.

The command completed successfully.
```

3. In the Windows 2000 Logon Script window, enter the path to the logon scripts for the highlighted user(s), and click the **Set Path** button. Or, you can clone an existing OS/2 home directory value for a Windows 2000 by clicking **Clone LS > NT**.
4. In the Windows 2000 Profile Path window, enter the path to the user profiles for the highlighted user(s), and click the **Set Path** button. In most cases you can leave this field blank. The migration tool will automatically assume the default path. Or, you can clone an existing OS/2 home directory value for a Windows 2000 by clicking **Clone LS > NT**. This field does not perform any function on platforms other than Windows 2000.
5. When complete, click the **Save** button.

Password

After clicking the **Password** button found in Figure 8-23, you will see the Set Password Policy window. The Set Password Policy window is shown in Figure 8-25.

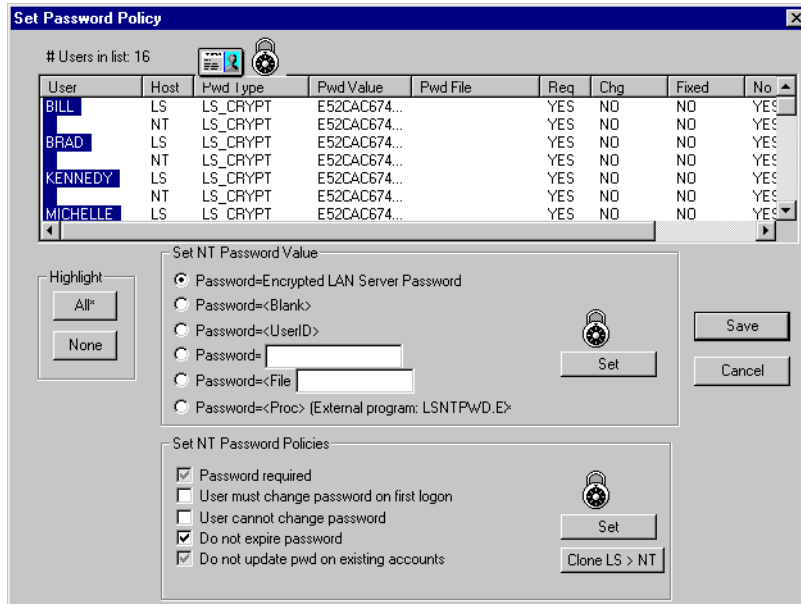


Figure 8-25 Set Password Policy

1. In the Set NT Password Value box, select the value that is desired and click the **Set** button:
 - Password=Encrypted LAN Server Password
To use this option, you must have exported the LAN Server data using the **-PASS** option. If the data was captured properly, you will see a series of letters and numbers in the column marked :Pwd Value. If there was no password data in your data capture, you will see entry:Missing.
This is the preferred option as it will allow existing users to use their LAN server password on the new NT/2000 domain.
 - Password=<Blank>
This option allows you to create accounts that have no password.
 - Password=<UserID>
Sets the each user's password to the user name in all uppercase letters
 - Password=[]

- Sets the each user's password to the typed-in field you provide.
 - Password=<File> []
 - This feature is not yet implemented in the current version of Migration Wizard.
 - Password=<Proc>
 - This feature is not yet implemented in the current version of Migration Wizard.
2. In the Set NT Password Policies box, select the policies that are desired and click the **Set** button. Or you can clone by clicking **Clone LS > NT** button.
- Password required
 - User cannot use a blank password on this account.
 - User must change password on first logon
 - Password will have to be changed when user does the first logon.
 - User cannot change password
 - Use this option for accounts that are not normally exposed to user action.
 - Do not expire passwords.
 - If set, this account will not conform to the password expiration date requirements.
 - Do not update passwords on existing accounts.
 - This option is useful when you do not want to reset passwords on existing Windows 2000 accounts.
3. When complete, click the **Save** button.

Groups

The Groups button is found on the Resolve Importation Issues window as shown in Figure 8-22 on page 322. Figure 8-26 on page 329 shows the dialog to enable and disable groups, as well as maps the LAN server group names to their existing Windows 2000 equivalents. These settings are used for setting the group memberships and for setting ACLs.

Predefined LAN server groups, which are mapped to built-in Windows 2000 Server groups, are handled as shown in Table 8-1.

Table 8-1 Predefined LAN server groups

LAN server groups	Mapped Windows 2000 groups
Servers	Automatically disabled

LAN server groups	Mapped Windows 2000 groups
GroupID	Automatically disabled
Local	Automatically disabled
Admins	Mapped to DomainAdmins
Users	Mapped to DomainUsers
Guests	Mapped to DomainGuests

Also, you can create or map LAN server groups to Windows 2000 domain Local or Global groups. You can enable or disable any LAN server group. When you click the **Save** button, the changes will be written to the local database (registry). There are some differences between local groups and global groups on Windows 2000 Domain as shown in the following table.

Table 8-2 Local Groups vs. Global Group on Windows 2000 Domains

Local Groups	Global Groups
Provide users with permissions or rights	Organize domain users
Can include (from any domain): user accounts and Global groups	Can only include user accounts in the domain where it resides
Cannot include other local groups	Cannot contain local or global groups
Are assigned permissions and rights in the local domain	Are added to a local group to give its members rights
On a computer running Windows 2000 Workstation or a member server, can only be assigned to local resources	Are not assigned to local resources
On a PDC, can be assigned resources on any domain controller in the domain	Must be created on a PDC in the domain where the accounts reside
Built-in Local Groups: Administrators, Backup Operators, Server Operators, Account Operators, Print Operators, Users, Guest, Replicator	Built-in Global Groups: Domain Admins, Domain Users, Domain Guests

The procedures to resolve the Group information are as follows:

1. In the Resolve Importation Issues box, click the **Groups** button.
2. You will see the LAN Server Group Enable/Mapping window as shown in Figure 8-26.

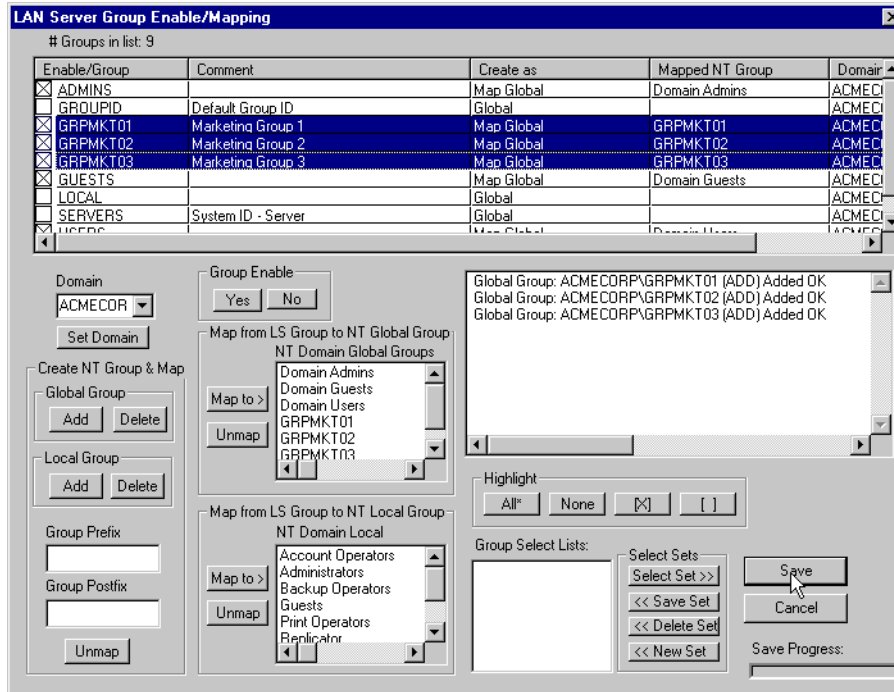


Figure 8-26 LAN Server Group Enable/Mapping

3. To map LAN server groups to a Windows 2000 Global Group:

- Select the LAN Server groups from the list.
- Select the domain in which to map the group.
- Select the **Windows 2000 Global Group(s)**.
- Then click the **Map to** button.

You can unmap the groups by clicking the **Unmap** button.

4. To map LAN server groups to a Windows 2000 Local Group:

- Select the LAN server groups from the list.
- Select the domain in which to map it.
- Select the Windows 2000 Local Group(s).
- Then click the **Map to** button.

You can unmap the groups by clicking the **Unmap** button.

5. It is also possible to create new global or local group by selecting the **Global** or **Local** button under Create Group.

6. When you are finished with the group mappings, click the **Save** button to save your changes, and continue with the next process, or click the **Cancel** button to discard your changes.

File Aliases migration

The File Aliases button is found on the Resolve Importation Issues window as shown in Figure 8-22 on page 322. Figure 8-27 is to enable and disable file aliases, and add, delete, and edit LAN server alias definitions (if you have external aliases on your OS/2 LAN Server, you will need this option.) Also, you can create one or a series of new Windows 2000 shares that will take the place of the LAN server aliases, and map LAN server alias names to existing Windows 2000 UNC definitions. You can also migrate data and permissions from LAN server aliases to Windows 2000 shares here. Before setting the file aliases, one or more aliases must be enabled and highlighted.

Note: Be aware that the Migration Wizard is not transferring the files that reside in the user's home directory. To migrate these files, create a temporary share called HOMEDIR on the OS/2 side and do not apply any ACLs.

After this preparation, the share can be copied like any other share by performing the following steps.

1. In the Resolve Importation Issues window, click the **File Aliases** button.
2. You will see the File Alias Mapping window as shown in Figure 8-27.

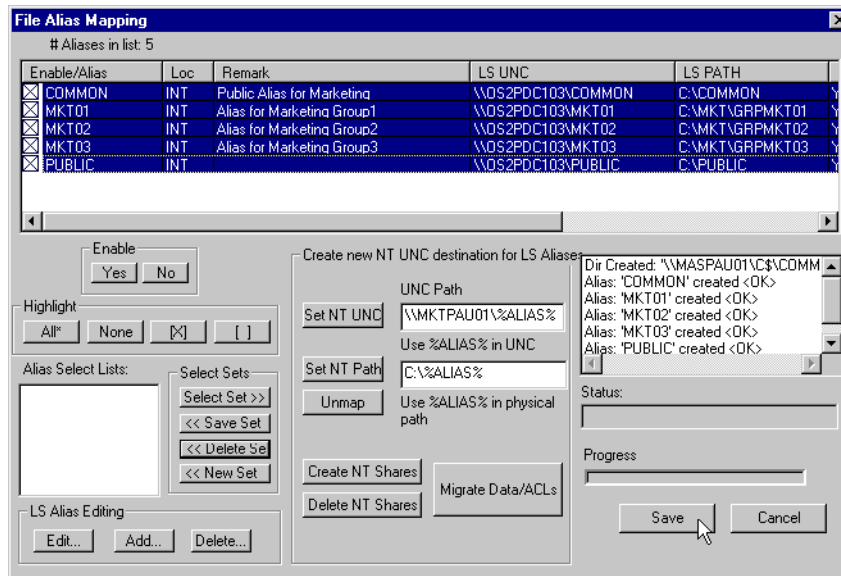


Figure 8-27 File Alias Mapping

3. Highlight the alias that you want to set, and click the **Yes** button in the **Alias Enable** box. You can highlight all or none by clicking **All** or **None** button in the Highlight panel.

External aliases are a feature of OS/2 LAN Server 3.0, and before that it allowed the LAN server administrator to define UNC resources anywhere on the company network as an available alias in the domain. The feature allowed any UNC to be visible as an alias in the domain. Administrators used this feature to provide domain users with access to resources outside of the domain. This feature can be used to allow a LAN server user to access a Windows 2000 resource through the automated logon assignments. As a result of the use of external scripts, there is no information in the DCDB to describe the path information. This means the UNC location will need to be added manually in the Migration Wizard.

4. If you are using OS/2 LAN Server 3.0, and you have one or more external aliases on your OS/2 LAN Server 3.0 domain, you can edit the alias definitions here and add the alias. This migration strategy is usable only by those companies that are still using OS/2 LAN Server 3.0. Because Windows 2000 does not support the external alias, if you want to transfer the data of the external alias, you have to add the alias definition(s). If you do not have any external alias, you can go to step 10.
5. If you want to add the alias, click the **Add** button in the LS Alias Editing window.
6. The Enter New Alias Name window appears. Enter new alias name, and click the **OK** button.
7. You will see the Alias Definition Editor window. Also, you can see the same window by clicking **Edit** button in the LS Alias Editing panel.
8. In the Alias Definition Editor, you can set the location, alias name, and the physical path of the alias. Also, you can enable and disable the alias.
9. When complete, to make the changes click **OK**.

You will see the File Alias Mapping window as shown in Figure 8-27.

10. To map a LAN Server Alias to the existing Windows 2000 Server shares, highlight one or more aliases and enter the UNC Path in the Create New Windows 2000 Alias Destination box and click the **Set Windows 2000 UNC** button. If you are mapping more than one alias, you may want to use the %ALIAS% argument in the UNC path. Then you can see the value of the Map column is changed to YES , and one of the Windows 2000 UNC column is set.
11. To create new Windows 2000 shares for your highlighted aliases, enter the UNC Path in the Create New Windows 2000 Alias Destination and click the **Set Windows 2000 Path** button, and enter the physical path of the share.
12. To create the share(s), click the **Create Windows 2000 Shares** button.

13. You can see the log to check for success or failure on the creates.
14. To delete the Windows 2000 share(s), highlight one or more existing LAN server aliases and click the **Delete Windows 2000 Shares** button.
15. Once you have completed creating your Windows 2000 shares and mapped them to existing LAN Server aliases, you are ready to move your data and ACLs.
16. When complete, to make changes, click the **Save** button. Otherwise, click the **Cancel** button.
17. At this point you need to create (export) the user IDs on Windows 2000, because they must already exist on the target system in order to transfer the ACLs for these users.

This step is described in 8.4.9 “Step 4: Export to the Windows 2000 Domain” on page 343. At this time, only transfer Users, RAS, and Groups. Do not export the logon scripts. Then continue with the next section.

Migrate Data/ACLs

For a successful migration, you must be able to contact both the source LAN server alias UNC and the target Windows 2000 Server share UNC from the computer running the Migration Wizard. To do this work, you must be an administrator for both of them.

If you are migrating the same machine, this step may not be used. Instead of these steps, you should restore the data from your backup instead:

1. Create a user ID named Administrator on your OS/2 LAN Server, if you do not have one. Administrator must have administrative privileges and its password must be the same as the Windows 2000 Administrator's password.
2. If your computer running the Migration Wizard is not in the Windows 2000 domain, log on to the domain as an administrator.
3. In order to migrate the files from the home directories, please be sure to check the note in “File Aliases migration” on page 330.
4. Highlight one or more existing LAN server aliases in the File Alias Mapping window.
5. Click the **Migrate Data/ACLs** button.
6. You will see the Export Data window as shown in Figure 8-28.

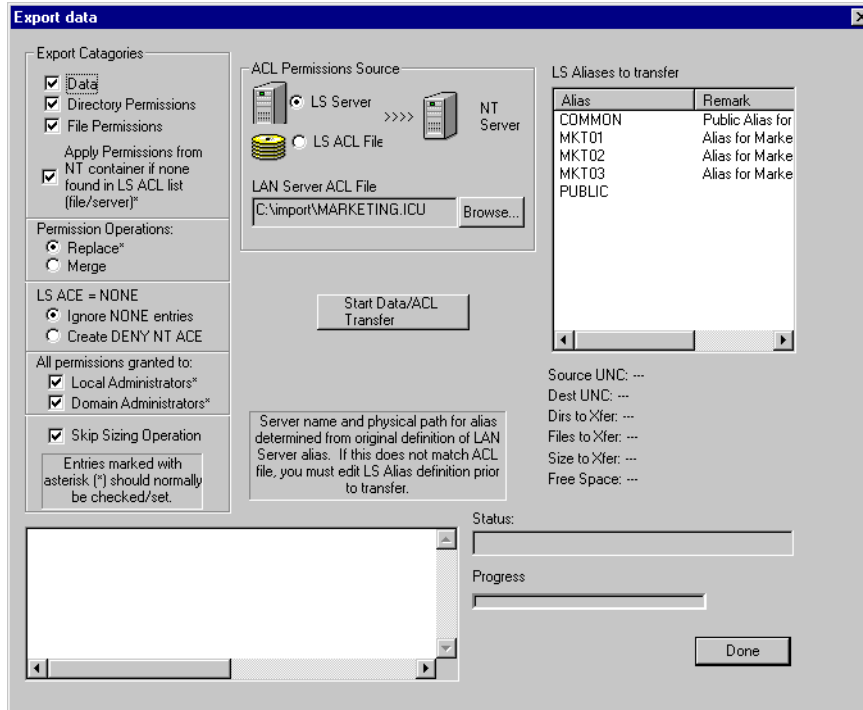


Figure 8-28 Export Data

Notice that there is the alias list to be transferred in the LS Aliases To Transfer section.

7. In the Export Categories section, select the extent of copy:
 - Data
 - Directory permissions
 - File permissions
 - Apply permissions from Windows 2000 if none found in LS ACL list
8. In the Permission Operations, choose **Replace** or **Merge**. Normally, you will use the Replace option unless you want to merge with the permissions that might already be in place. We do not recommend using the Merge option since you might accidentally merge in an ACE for full access to the group Everyone.
9. In LS ACE = NONE, choose **Ignore NONE Entries** or **Create DENY NT ACE**.

Ignore NONE Entries Ignore will simply create the ACE on the Windows 2000 Server and not change the access from the default, which is Everyone having Full Control.

Create DENY NT ACE This will create an ACE on the Windows 2000 Server, which will deny users permissions if they are not specifically mentioned in the LAN Server ACL. In other words, a blank ACL will create permissions on the Windows 2000 Server that excludes all users apart from the local Administrator from this resource.

10. In the All Permissions Granted To section, select **Local Administrators** and **Domain Administrators**. In Windows 2000, administrators do not have the access rights automatically. We recommend adding permissions for both administrators, so that you can confirm the proper transfer of the directories and data.
11. If you want to check the available size, deselect the Skip Sizing Operation. Otherwise, select it.
12. In the ACL Permissions Source, choose the **LS Server** or **LS ACL File**. If you want to transfer ACLs from your LAN server, select **LS Server**. If you want to transfer the ACLs from your export file, select **LS ACL File**. Locate the file using the **Browse** button. At this time, the file must be created using the **-ACL** option on your OS/2 LAN Server.
13. Check the existence of the LAN Server and Windows 2000 Shares.
14. Make sure your X: and Y: drives are unassigned. If you have connections on X and Y drives, disconnect any shares.
15. Make sure you log on to the Windows 2000 Domain as an Administrator and that there is an administrator ID on the OS/2 LAN Server.
16. Finally, click the **Start data/ACL Transfer** button.

You can see not only the status, but also the log to check for success or failure.
17. When complete, click the **Done** button.

8.4.6 File ACLs/Directory ACLs

There are differences between LAN server and Windows 2000 Server in terms of file and directory ACLs. So, you can decide if you keep the current mappings or want to substitute your own mappings. We recommend you leave the mappings by default. If you really want to modify, then before you modify these mappings, we recommend that you fully understand the functions of the bits available in the Windows 2000 file system.

File ACLs

1. In the Resolve Importation Issues window click the **File ACLs of LS to NT ACE Bit Mapping**.

2. You will see the LAN server to Windows 2000 File Permission Mapping window as shown in Figure 8-29.

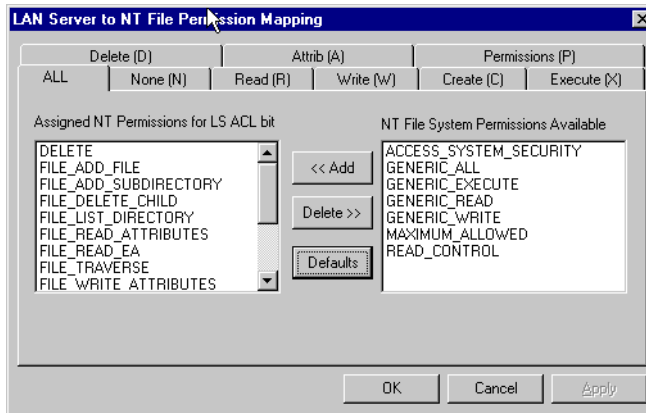


Figure 8-29 File ACLs mapping

3. You can modify by clicking **<< Add**, **Delete >>** or the **Defaults** button for each permission.
4. When complete, make the changes by clicking the **OK** button. Otherwise, click the **Cancel** button.

Directory ACLs

1. In the Resolve Importation Issues window, click the **Directory ACLs of LS to NT ACE Bit Mapping**.
2. You will see LAN server to Windows 2000 Directory Permission Mapping window as shown in Figure 8-30.

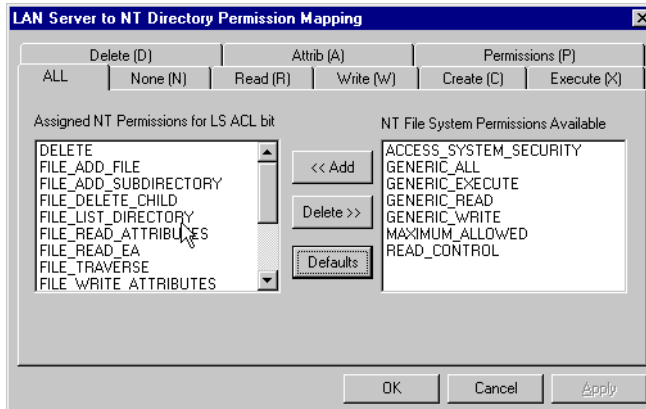


Figure 8-30 Directory ACLs Mapping

3. You can modify by clicking **<< Add**, **Delete >>** or the **Defaults** button for each permission.
4. When complete, make the changes by clicking the **OK** button. Otherwise, click the **Cancel** button.

8.4.7 Printer aliases

1. To resolve your printer aliases, click the **Printer Aliases** button from the Resolve Importation Issues window.
2. Then you will see the Printer Aliases Mapping window as shown in Figure 8-31. Highlight LAN server printer aliases individually or all at once, and then select the **Yes** button under Alias Enable. An X beside the alias name will ensure it is enabled.
3. Under Map From LS Alias To NT UNC Resource, enter a path for the alias on the Windows 2000 domain. Be sure to use %ALIAS% in the path name. Select the **Map To** button to map the alias. The ACME company used a path of \\MKTPAU01\%ALIAS%.
4. From the Alias List, you can scroll to view your aliases and their mappings. Your screen should resemble those shown in the Figure 8-31.

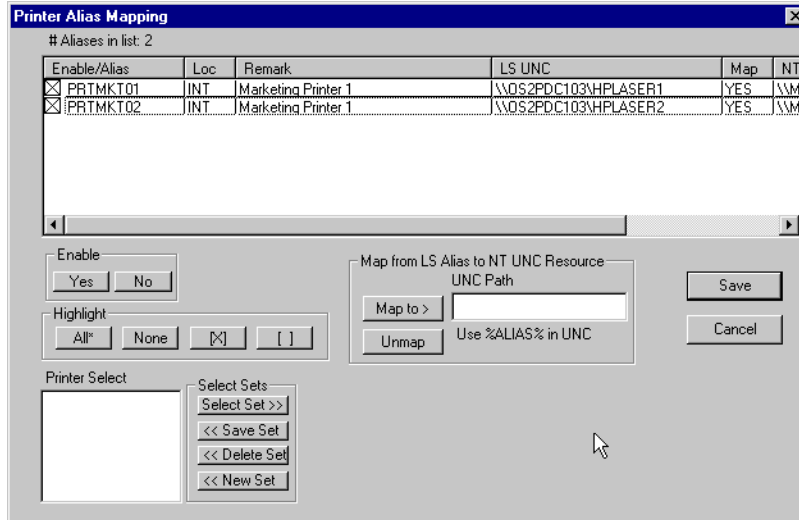


Figure 8-31 Mapping Printer Aliases

5. Select the **Save** button to save your changes and return to the Resolve Importation Issues screen. Select **Cancel** to discard your changes.

8.4.8 Logon scripts

Because Windows 2000 does not provide a domain control database for setting logon assignments for users, users set their logon assignments through *persistent connections* or through a *logon script*. Persistent connections are established by the user when they use File Manager, or the drives object to map a connection. Many administrators prefer to use logon scripts to set up logon assignments as a convenience to the user, and also to control the load on the servers in the network. This Logon Script setting under the Migration Wizard allows you to set the location and name the script or program that is run prior to the user getting control of the console. To set up logon scripts, select the **Logon Script** button from the Resolve Importation Issues window. In the following section are the four parts to the logon script setup:

Enable build

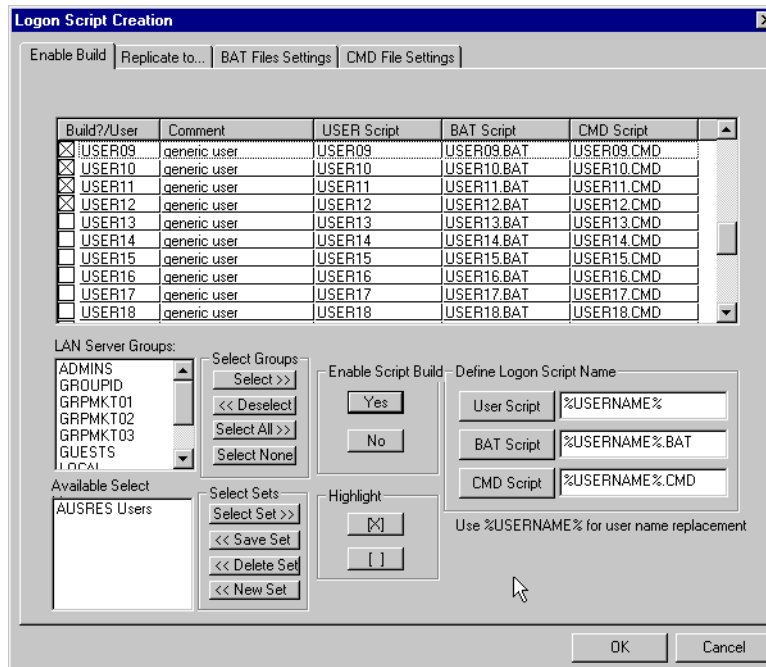


Figure 8-32 Logon script creation

1. Highlight to select the users for which you will need logon scripts. You can do multiple individual highlights with the Control-Click option, or a range of individuals with the Shift-Click option. Users can also be selected by selecting a group. Use the >> button to select all users within a specific group or the << button to deselect all users within a specified group.
2. Ensure they are enabled by selecting the **Yes** button under the Enable Script Build section. An X beside the username will ensure it is enabled for script build.
3. Under Define Logon Script Name, provide the appropriate script names. It is a good idea to leave off the extension of the user script name, so that the correct file will be executed at logon time. The operating system will execute the file with the appropriate extension. For OS/2 and Windows 2000 machines, the *.CMD file will be executed and for DOS and Win95 machines, the *.BAT file will be executed.
4. Click **OK**.

Replicate to...

1. All scripts need to be replicated to the PDC and to all BDCs. Even if no BDC is used, this step must be performed. To add servers for script replication, select **Add Script Location** under the Replicate To... tab of the Login Script Creation window. The Login Script Creation window is shown in Figure 8-32.
2. Highlight the Windows 2000 Server and select the **Select Windows 2000 Server** button as shown in Figure 8-33.
3. Under Script Path Selection, select or enter the appropriate script path. The NETLOGON share is not used because it is a READ-ONLY share. Instead the administrative share of ADMIN\$ is used. This share exists on both LAN server and Windows 2000 domains. If you are using the replicator function to duplicate scripts, you should manually input the path to the export directory.
4. The primary domain controller in the ACMECORP domain was chosen for script replication in the ACME test case. Your screen should resemble Figure 8-33.

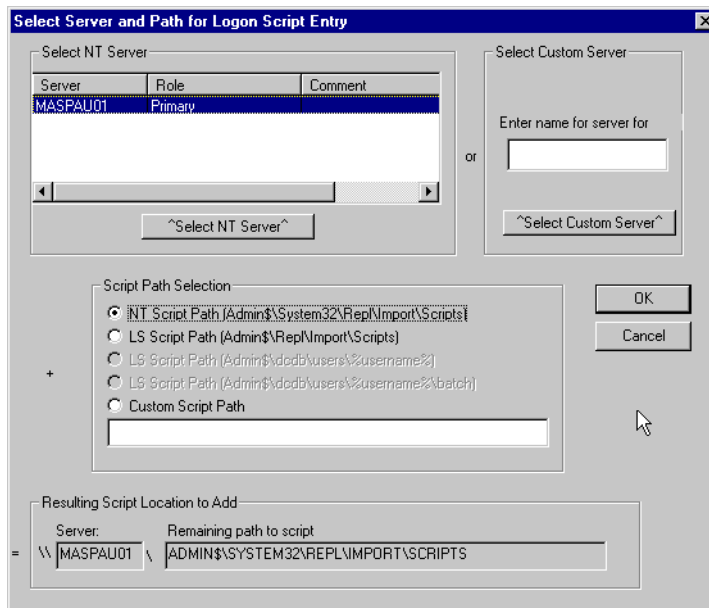


Figure 8-33 Add script location

4. Click **OK** to accept the script location.
5. Once you have added all of the script locations for replication, it is a good idea to click **Verify Servers/Paths** to ensure that all target directories are valid for writing.

BAT file settings

Shown in Figure 8-34 are the default settings for the BAT script files. Windows 95 and DOS clients use BAT logon script files. Your scripts can be customized by editing this screen. For Windows 95 workstations to use home directories, you will need to add to the BAT script:

```
NET USE %S /HOME
```

This can be done by checking the **Enable Connect Home Directory String** check box and ensuring that its entry field contains the NET USE statement described above.

If DOS LAN Services are used as a client instead of Windows 95, the NET USE %S /HOME command does not work. In this case you should place the NET USE command in the trailer because it is the last item from the BAT file that is run. Remember, if your users are authenticating with a Windows 2000 server and using home directories on an OS/2 Server, you need to convert these OS/2 home directories to permanent shares. To access those, embed the NET USE command for the home directory in the trailer section of the script file for all platforms.

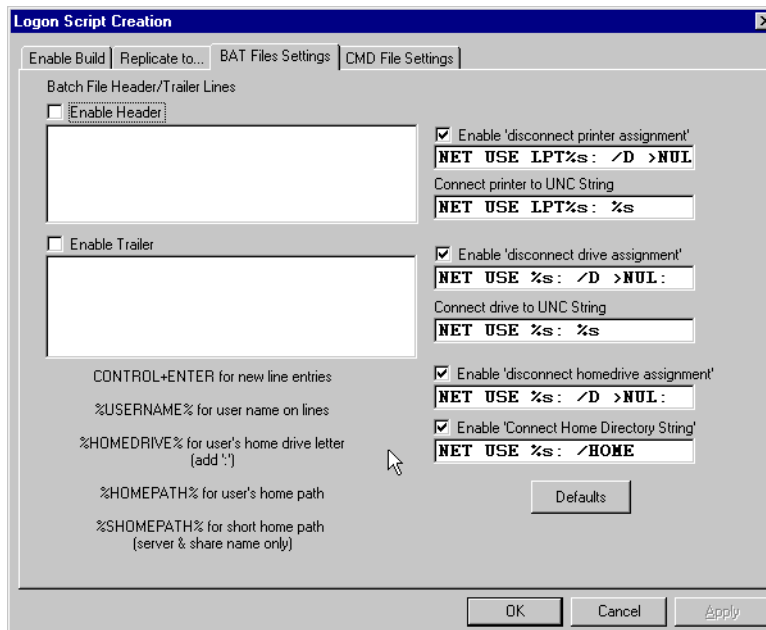


Figure 8-34 BAT file settings under logon scripts

CMD file settings

Shown in Figure 8-35 are the default settings for the CMD script files. Windows 2000 and OS/2 clients use CMD logon script files. Your CMD scripts can be customized by editing this screen. On OS/2 workstations, you will need to connect to the home directory through an embedded **NET USE** command for the home directory. You can place the **NET USE** command in the header or trailer section. You should also disable the Connect Home Directory String. OS/2 workstations cannot use automatic home directory connections when authenticating with a Windows 2000 PDC.

To get home directories to work in one CMD file, regardless of OS/2 or Windows 2000 workstations, do the following:

1. Specify the correct home directory path as described in the “Set Paths” on page 323.
2. Disable the Connect Home Directory String.
3. Use the following line in the trailer of your CMD file:

```
NET USE %HOMEDRIVE%: %SHOMEPAH%
```

The **NET USE** statement is placed in the trailer because of the execution order of the CMD file. The execution order is Header, Selection, and Trailer. In the Selection section, there is an option **Enable Disconnect Home Drive Assignment**. If you place your **NET USE** in the Header, and also have the **Enable Disconnect Home Drive Assignment** selected, you will disconnect your home drive network assignment.

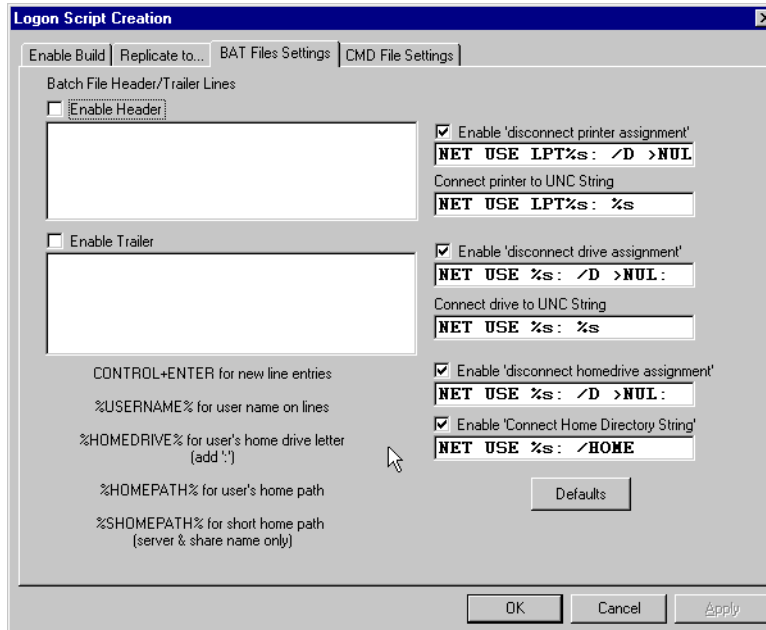


Figure 8-35 CMD file settings for OS/2 and Windows 2000 Clients

Note: For OS/2 users only, this logon script will generate an error NET8191 while mapping the home directory. However, users can ignore this error as it is not true for OS/2 clients. For more details, refer to “Set Paths” on page 323.

Home directory UNC shares

OS/2 workstations cannot connect to Windows 2000 home directories without a little help. The trick is the creation of fixed UNC shares for each user's home directory. The user account should have already been created before setting user/group permissions. Select the **Home Directories** button shown on the Resolve Importation Issues window. Refer to Figure 8-36 while reading the steps below:

1. Highlight users and enable them with the **Yes** button under the Enable Create section.
2. Under Set Parameters for Windows 2000 Home Directory, enter the UNC name and select Set UNC. Enter the path name and select **Set Path**. Enter any comments and select **Comments**.
3. Set the default permissions for the home directories under the Home Directory Share Permissions. If no home directory share permissions are set, then the default of Everyone:All will be used.

4. Select **Create Shares/Permissions** to create shares and permissions for the home directories. Use the logging window provided to ensure changes were successful. Your screen should resemble the figure.

Note: When selecting a share name for the home directories, consider using the following naming convention:

\\Servername\%USERNAME%\$

The following \$ sign results in a share that is not displayed to non-admin users in the Windows 2000 domain.

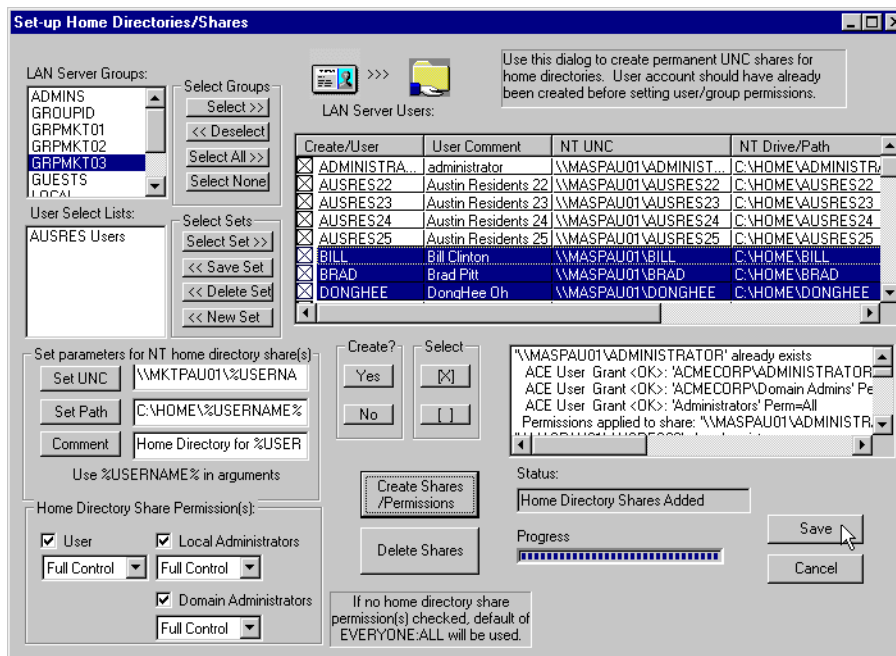


Figure 8-36 Set up home directories and shares

5. Select **Save** to save changes and continue with the next step. Select **Cancel** to discard changes.

8.4.9 Step 4: Export to the Windows 2000 Domain

1. In the IBM LAN/Warp Server to Migration & Synchronization Wizard, click **Export**.
2. You will see Export From Database to Windows 2000 Domain window.

3. Select the categories (Users, RAS Dial-In, Groups, Create Logon Scripts) that you wish to export in the Export Categories.
4. Click the **Database > NT Domain** button to start the export process. Note that the push button will change to a **Stop** button when the export process starts running. If you want to stop the export, just click the **On** the button.
5. As the export process progresses, you will see the status in the Status bar.
6. When the export is complete, you will see the following message in the Status bar:

End: Export to NT Domain

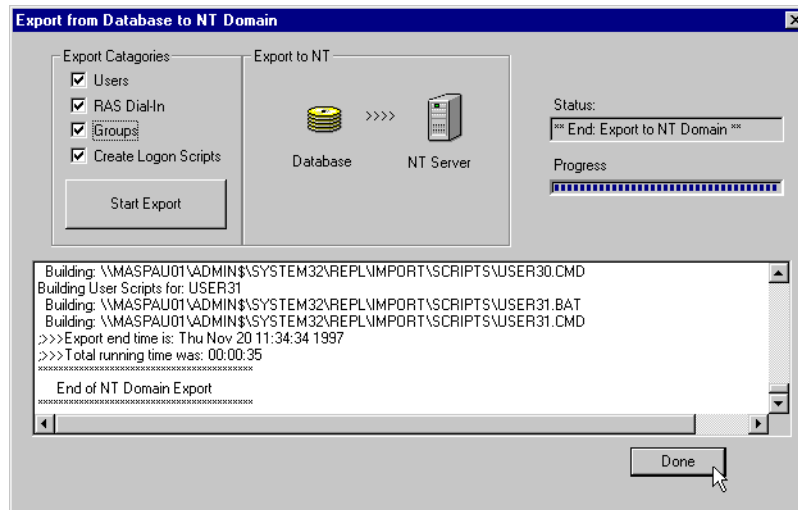


Figure 8-37 Export to Windows 2000 Domain

7. After completing the export, click **Done** as shown in Figure 8-37.

The Migration Wizard is capable of moving both small and very large enterprises to the Windows platform. It is a good idea to do a series of test migrations within a company lab to get familiar with the tools, and to confirm the best operation to be used.

Companies use two strategies in their migration: “Big Bang” and “Phased Migration.” In the Big Bang approach the customer transitions all of their assets from LAN server to an existing Windows NT, or better domain over a weekend. If this option is desired, servers should be staged ahead of time, and accounts and data should already be copied over prior to the cut over date. On the cut over date, the customer should use the tool Robocopy from the NT Resource Kit to synchronize in new and changed data. A new capture from the LAN server

domain can be processed and used to refresh accounts and passwords if desired.

In a Phased Migration, the customer will move their resources over to the NT domain in phases (usually six months to a couple of years). Lieberman & Associates has a product known as *Server-to-Server Password Synchronizer*, which is capable of keeping existing LAN server domains in sync with NT and the above servers and domains. For this approach to work, users will need to be informed to keep their password length and complexity to a standard that is compatible with LAN servers (14 characters or less, and no unprintable characters).

8.5 Servolution

Servolution from Comtarsia is another tool that may be useful during migration. The Servolution migration path is a different approach to migrating OS/2 Servers than other tools. Others concentrate on the server by migrating user database and resources from one platform to the other. The Servolution migration path on the other hand focuses on the user's workplace and makes sure that he always has access to all resources.

This allows for heterogeneous environments to function and thus enables the migration of resources and the user database. This variation allows for a phased approach to migration.

One can decide if and when to migrate resources away from OS/2 and which target platform to migrate to. A heterogeneous network can be run. One can decide where to migrate the user database such as to Windows (ADS) or to LDAP, SAMBA, Domino, and the time to migrate an entire server or an alias to the target system can be flexible.

8.5.1 Overview of Servolution

- ▶ Client support

The Servolution Logon Client makes it possible for a Windows workstation to log on to an OS/2 Server, featuring good and reliable Windows client support.

(The following platforms are supported by the Logon Client: Windows NT, Windows 2000, Windows XP, Terminal Server 2000, Terminal Server 2003, Citrix.):

Servolution Logon Client also provides the ability to directly log on to LDAP or RACF® user management.

- ▶ Smooth migration

A smooth, step-by-step migration path from OS/2 (aliases and home directory) resources to NT, Windows (ADS), Linux, AIX, or Samba servers is possible. There is no pressure for final server configuration decisions to be made at an early stage of the migration.

It is possible to change the client environment, including access control on the target system, during the migration process.

- ▶ No interruption for the user's activity

The entire migration process can be carried out without interrupting the user's productive working time. As opposed to the "all-at-once" migration methods, the user is not prevented from being logged in and accessing his resources. Unforeseen delays before starting with the new system will *not* keep users in a passive state. This is one important cost-saving feature of the Logon Client.

- ▶ Mixed environment

The Servolution Migration Path keeps a mixed (OS/2, Windows ADS, Linux, AIX, Samba) environments synchronized, and all benefits of running a heterogeneous network can be fully realized.

- ▶ RACF or LDAP as User Management

Target user management can be on RACF or LDAP. All OS/2 user definitions like aliases and network applications can be migrated onto LDAP servers. These specific applications will continue to be available on the target user management platform without data inconsistencies or noticeable access limitations.

The reliable maintenance of the user information and definitions is ensured by the Comtarsia LDAP schema-extension. It contains additional object classes and attributes developed specifically for this purpose.

- ▶ Flexible architecture

It is as an easy task to implement additional resource servers or domain controllers during the migration process if the Servolution Migration Path is up and running. The newly attached user database is filled and updated automatically. The Servolution Migration Path makes sure that at each logon session all defined systems are synchronized with the master domain.

- ▶ Flexibility of user management platform:

It is possible to delay the decision of the user management platform. The decision can be made even after all resources are already moved away from the OS/2 Server, since the migration of user management is the last step in the Servolution Migration Path.

It is possible to change the target user management between Windows (ADS), Linux, AIX, Samba, LDAP, and Domino. The Servolution SyncAgents are available for many platforms:

Please see Servolution Product Guide at:
<http://www.servolution.comtarsia.com>

► Non hash migration:

As opposed to other tools that use the hash migration practice, Servolution Migration Path migrates the user without security leaks to the target system, since the LAN server hash is not transferred to the target domain.

In the Servolution Migration Path, the user name and the password will be transferred in a highly secure way (RSA encrypted) between Servolution Migration Path components, and will be finally transformed to a hash by the supported interfaces of the target platform, according to its own security rules. No security level modification is necessary on the target system.

8.5.2 How Servolution works

Servolution stands for the Servolution product family, consisting of the Servolution Logon Client, the SyncProxy, and the SyncAgent.

The Servolution Logon Client 3.1 performs a full OS/2 or LDAP (RACF) logon, controls the Windows logon session, and gives support for assignments of directory and printer aliases, roaming profile, home directory, local group membership, policies, scripts, and network applications. At each logon, the Logon Client sends a sync packet to the SyncProxy server with the entire user information including the password. Afterwards, the Logon Client receives and displays the result of the synchronization process.

The Servolution SyncPacket 1.1 is the technical implementation that allows for user management on OS/2 or LDAP while resources can be placed on Windows, Linux, Samba, and UNIX.

At each logon, the Servolution Logon Client sends a synchronization request/synchronization packet to the server where SyncProxy is running as a centralized distribution point for these; SyncProxy forwards them to the respective SyncAgent(s) on the target system(s).

SyncProxy and SyncAgent(s) handle and process synchronization requests and packets, and therefore are termed *SyncPacket*.

The agent makes sure that the user account exists, a synchronous password is present, suitable access permissions to the home directory are set, the user belongs to the corresponding groups, and many other things.

SyncPacket allows for the complete and fully functional user management to remain on the OS/2 Server as long as user accounts are created and resources

can be moved to Windows, Linux, and UNIX servers without leaving any security gap open, or any data becoming unavailable or lost.

SyncAgents are available for Linux, UNIX, Samba, NT, Windows 2000, Active Directory, Lotus Domino, DB/2, and Oracle.

The entire communication between client and respective proxy and agents is based on RSA encryption with a key size between 512 and 2048-bits.

For production use, it is recommended to use a personally created key instead of the built-in key, which the SyncPacket is using as a default.

The Servolution Logon Client and the Servolution SyncPacket are the core components of the Servolution Migration Path, which is described in this chapter.

More information and comprehensive product manuals about the products and the Migration Path are available on the Servolution product site at:

<http://www.servolution.comtarsia.com>

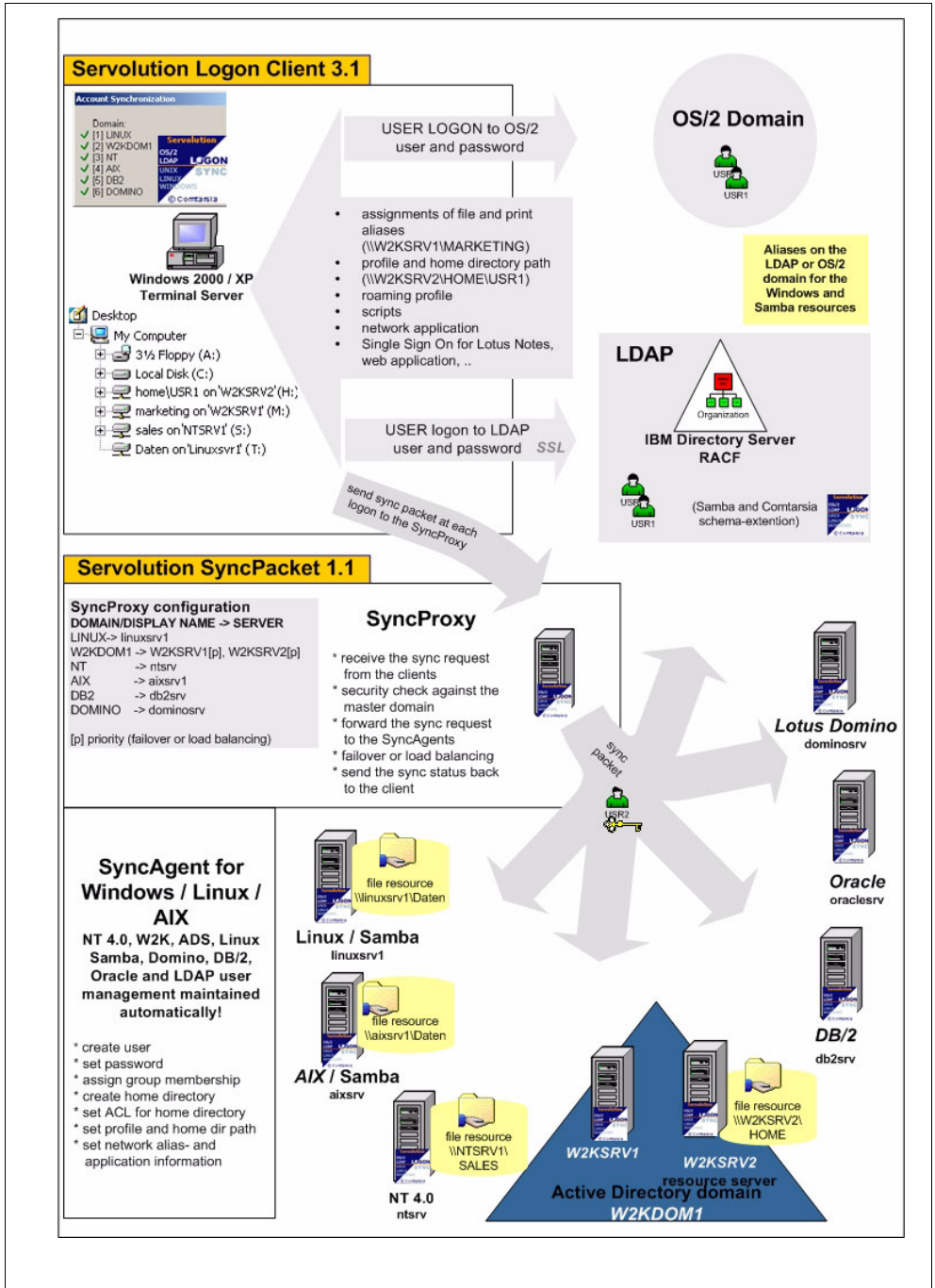


Figure 8-38 Servolution solution overview

8.5.3 Migration scenario using Servolution

The following sections describe a migration scenario using the Servolution solution.

Source

We will describe the necessary steps for all components described in order to give a clear-cut guidelines on how to prepare the initial and the target system to be able to move user management and resources completely away from OS/2 ServerOS/2 Servers using Servolution Logon Client 3.1 and Servolution SyncPacket 1.1 for this purpose.

The network is assumed to be NetBIOS over IP, all client workstations are Windows based and have the Servolution Logon Client installed.

User management is on the OS/2 domain. Users log on onto the OS/2 master domain and access control is determined by the OS/2 groups. All resources are on the OS/2 Server, including home directory, roaming profile, aliases, network applications, policy files, and logon scripts.

For basic information on how to install and configure the Servolution Logon Client please see:

<http://www.servolution.comtarsia.com>

Target

In this case, the target will Linux with user management on an LDAP server, running IBM Directory Server Version 5.1 for AIX, with additional enhancements such as:

- ▶ Samba Schema (optional)
- ▶ Comtarsia Schema

The resources for the users are now on Linux as well, provided by Samba 3.0 either on Red Hat, or SuSE as outlined in earlier chapters.

Clients can be all platforms supported by the Servolution Logon Client and having installed and configured the SyncClient.

For a successful migration when using Servolution products, the following packages and respective versions are required:

- ▶ Servolution Logon Client Version 3.1 or higher
- ▶ Servolution SyncPacket 1.1 (SyncProxy and SyncAgent) or higher

Step by step description

This is an overview of the major steps to follow described in detail later in this section:

1. SyncPacket 1.1 Installation

The first step is the installation of the Servolution SyncPacket on the target system. The SyncProxy server and the SyncAgents have to be installed and configured. For more detailed information see “Enable the Servolution Logon Client SyncClient feature” on page 361.

2. Servolution Logon Client installation and configuration

If Servolution Logon Client 3.x is already installed on the Windows clients, the SyncClient has to be enabled. For more detailed information see 8.5.4 “Installation of the Servolution SyncPacket on the target server” on page 355.

Otherwise, the Servolution Logon Client 3.1 has to be installed on all Windows workstations with the correct SyncProxy configuration.

At this stage each logon is still validated against the OS/2 domain, but the password on the target system is synchronized with Servolution SyncAgent system. The user management and all resources are still on OS/2 as seen in Figure 8-39.

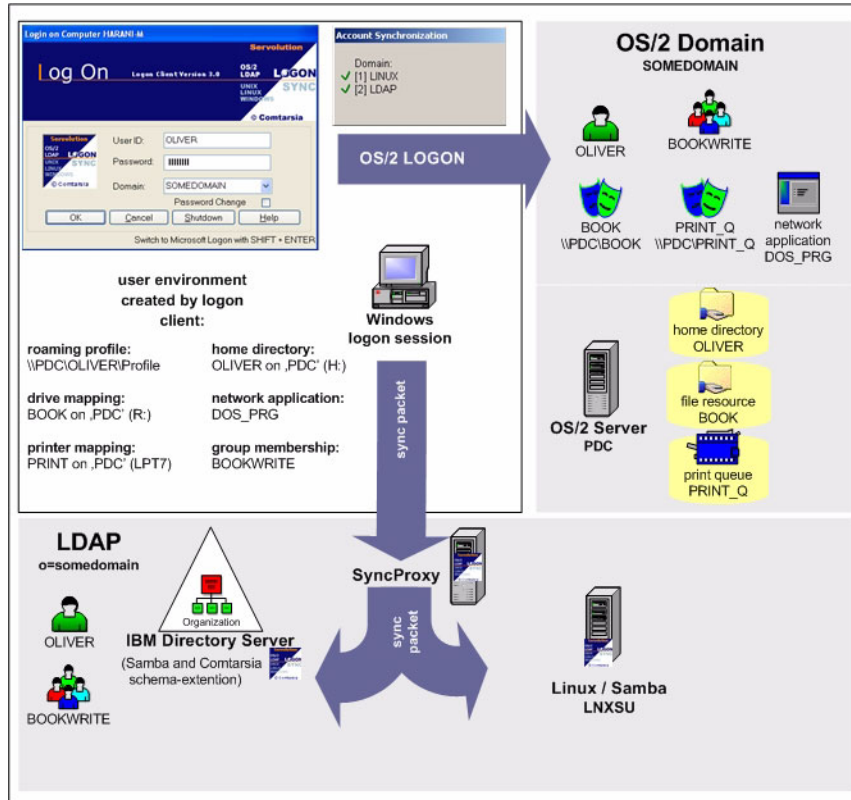


Figure 8-39 Servolution migration step 1

3. Move resources

All Aliases and home directories can now be moved from OS/2 to the target system (Samba) as described in 8.5.8 “Resource migration to Samba” on page 361.

On the next user logon with the Servolution Logon Client, the user session gets the same drive letter for the target network resource.

At this point, all scripts and policy files, usually on:

```
\\PDC\netlogon - C:\ibmlan\rep\import\scripts
```

These should be moved from the OS/2 share to an appropriate public/read share on the Samba system.

This may have an impact on the policy path and the script paths in the client configuration. This change can be done relatively easily with policy files or scripts. As long as the scripts and policy files are in sync, the redefinition of the new path on the client configuration should be completed.

A detailed description can be found in the Servolution Logon Client manual at: <http://www.servolution.comtarsia.com>

At this stage, user management is still on OS/2, but the resources have moved to the Linux server.

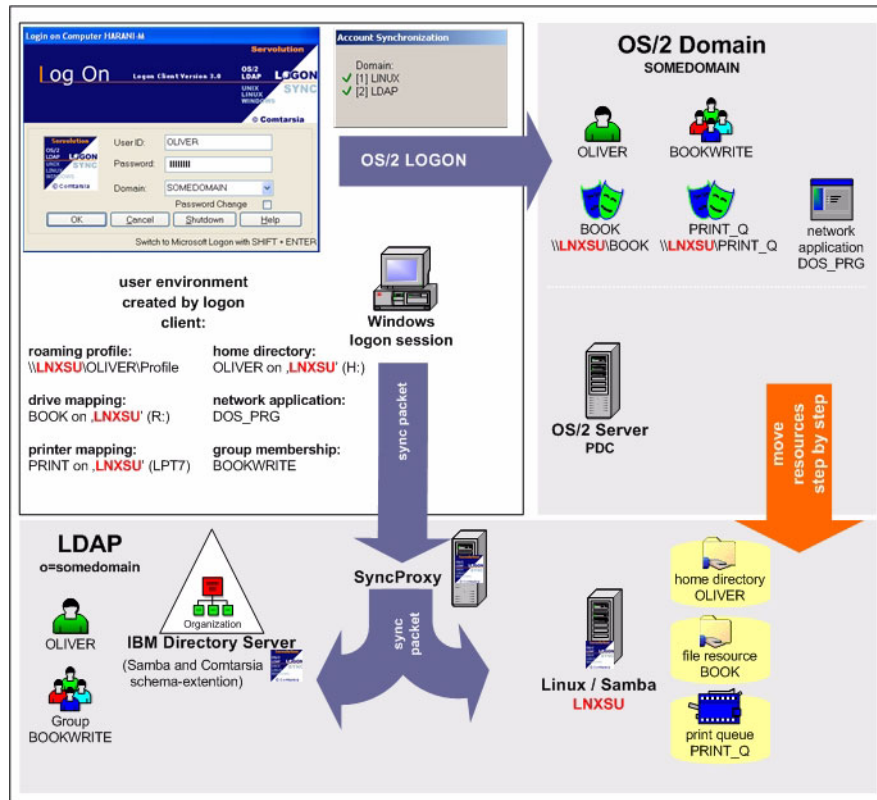


Figure 8-40 Servolution migration step 2

4. Change user management

User management can either be migrated to native Samba or to a central LDAP instance:

a. Samba

At this point in the Servolution Migration Path it is possible to change the client and Samba configuration for a native Samba network, because all user's groups and resources are already on Samba. Users, which during the migration period have not performed any logons, will be migrated by the tool OS2MigrateUsers as described in "OS2MigrateUsers" on page 363.

In this case, the migration process is now completed.

Note: It is not possible to migrate the aliases and the network applications into the Samba domain.

b. LDAP

A complete migration of the OS/2 domain definition to LDAP, including network applications and alias definitions is processed by the OS2LDAPMigration tool. Even users, which during the entire migration time have not performed any logon session, will be migrated at this time. The special information from the OS/2 attributes is stored in appropriate LDAP attributes defined by the Comtarsia schema-extension. For more details see 8.5.9 “User management migration to LDAP” on page 366.

The Logon Client is switched to the LDAP Logon mode with a simple registry setting as described in “Sample Logon Client configuration” on page 371.

At this point, the LDAP SyncAgent must be disabled because LDAP is now the primary user management platform.

Now, the user management is on LDAP, the resources are on Linux, the migration process is now completed and the OS/2 Server can be switched off.

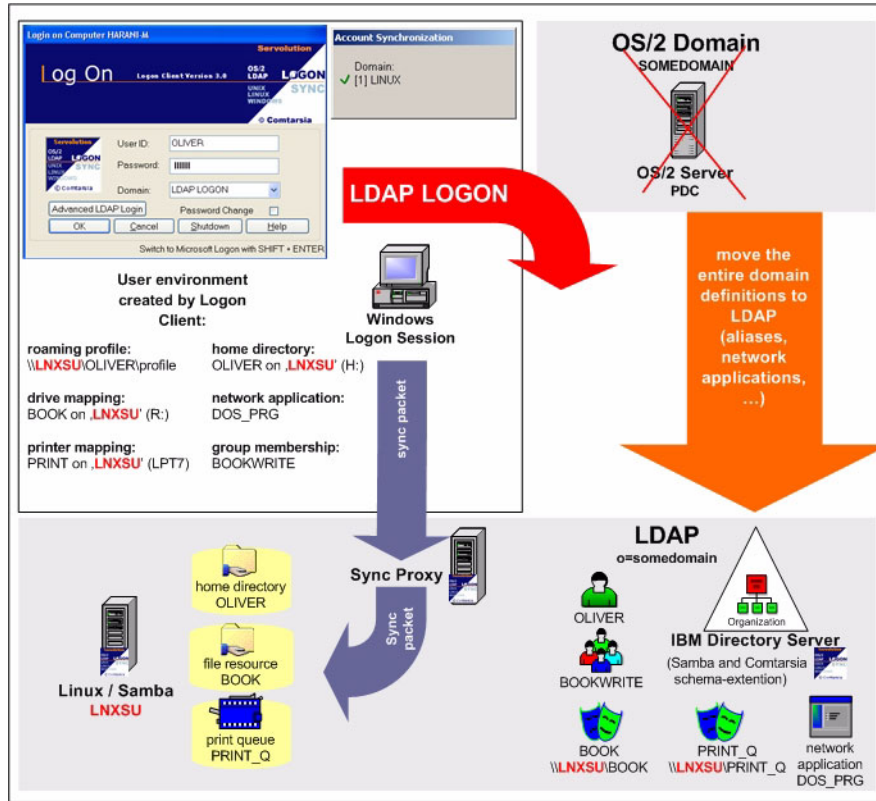


Figure 8-41 Servolution migration step 4

Further documents and tools for user management on LDAP are available at: <http://www.comtarsia.com>

8.5.4 Installation of the Servolution SyncPacket on the target server

At each logon, the Servolution Logon Client sends synchronization requests and synchronization packets to the server where SyncProxy is running as a centralized distributor for them; the SyncProxy forwards them to the respective SyncAgent on the target system.

The SyncProxy and SyncAgent handles and processes the synchronization of requests and packets are therefore is called *SyncPacket*.

SyncPacket allows for the complete and fully functional user management to remain on the OS/2 Server as long as user accounts are created, and resources

can be moved to Windows, Linux, and UNIX servers without leaving any security gap open or any data be unavailable or lost.

8.5.5 Install and configure the Servolution SyncProxy

The SyncProxy has to be installed on a Windows or Linux server.

Start setup by running SP_1_1_xx_x_setup.exe file. (For SyncPacket version 1.1 for Windows)

After entering the user and company name, and specifying the program destination folder, select features you want to install: Servolution SyncProxy and if you also want to run a SyncAgent on this machine choose both.

After successful installation, some configuration needs to take place:

- ▶ Domain synchronization

In the field Add domain the names of the domains or the standalone server name to be synchronized can be specified (Domain Name).

Also, the server where the SyncAgent will run and process the synchronization requests has to be defined here (Server).

In case a standalone server is configured, Failover and Load balancing is disabled. Otherwise, one can configure whether a secondary server (set under Secondary Server) will only be contacted if the primary server stops processing inbound requests (enable **Failover**) or requests are sent to server1 and server2 in alternation instead (enable **Loadbalancing**).

In the field Domain Type are the domain types or the standalone server configuration to set.

- ▶ Security

Set Master Domain Name, it is the authentication domain name. Logon requests are accepted only against this domain.

Server type for password counter checking, if desired, is set here.

If enabled, the SyncProxy executes counter checking, that is, it does not simply accept the user name and password received from the Logon Client, but instead processes a logon against the master domain to reassure password validity. Synchronization requests will be forwarded to SyncAgent only after a successful logon.

If OS/2 is selected, configure the Server Name and IP Address of the OS/2 master server below.

- ▶ Service Control

Startup type (Automatic./Manual) can be specified and SyncProxy service started and stopped.

► Licensing

A license key for testing purposes is provided. Click **Load a new licensekey** in order to browse for a purchased license key according to individual conditions.

8.5.6 Preparation of the Linux target platform

The following sections describe the steps to prepare the target platform.

Configuring Samba 3.0

It is assumed that Samba 3.0 is already installed on the Linux server, either from source or from a binary package.

Set the *smb.conf* file below from Samba 3.0, which is configured according to the migration scenario defined earlier in this redbook

Example 8-6 smb.conf file for Servolution

```
##### Global Settings #####
[global]
workgroup = IBMRES
server string = IBM Residency Samba 3.0
hosts allow = 192.168.2. 127.
log file = /usr/local/samba/var/log.%m
max log size = 50
security = user
encrypt passwords = yes
socket options = TCP_NODELAY
interfaces = 192.168.2.0/24
dns proxy = no

##### Share Definitions #####
[homes]
comment = Home Directories
browseable = no
writable = yes

[LANSHARE]
comment = LANSHARE
path = /shares/lanshare
create mask = 0660
directory mask = 0770
writeable = yes
public = no
```

```
valid users = @transition
```

[BOOK]

```
comment = BOOK
path = /shares/book
create mask = 0660
directory mask = 0770
writeable = yes
public = no
valid users = @bookread, @bookwrite
read list = @bookread, @bookwrite
write list = @bookwrite
```

[PRINT_Q]

```
comment = PRINT_Q
path = /usr/spool/samba
browseable = no
writeable = no
printable = yes
valid users = @printer
```

Samba LDAP schema

Samba 3.0 can be configured to use an LDAP directory as the user database instead of the usual smbpasswd file. Please follow these steps:

1. Import the Samba schema-extension into the directory server.
2. For use with the IBM Directory Server 5.1, the schema file has to be slightly modified; a ready-to-use schema file can be downloaded from the Servolution Web site.
3. An LDAP organizational unit should be created, which will hold all Samba user and machine objects (ou=samba in our example).
4. Next, the following lines have to be added to the global section in the smb.conf file shown in Example 8-7.

Example 8-7 Samba LDAP organizational unit

```
# ldap options
passdb backend = ldapsam:ldap://ldapsrv
ldap delete dn = yes
ldap suffix = o=comtarsia
ldap user suffix = ou=samba
ldap machine suffix = ou=samba
# the admin pwd needs to be set with smbpasswd -w PASSWORD
ldap admin dn = cn=root
```

A comprehensive description of all smb.conf configuration parameters can be found at:

<http://www.hu.samba.org/samba/devel/docs/html/smb.conf.5.html>

8.5.7 Installing the SyncAgent for Linux

SyncAgent for Linux is a member of the Servolution SyncPacket product family.

SyncAgent consists of two modules: the system module and Samba module.

The system module is responsible for maintaining Linux user accounts and has the following functions:

- ▶ Creation of new user accounts including user directories and assigning of necessary file system permissions
- ▶ Directing of group memberships of individual users including the possibility of a GroupMapping list
- ▶ Synchronization of user passwords

The Samba Module is responsible for maintaining Samba user accounts and has the following functions:

- ▶ Creation of new Samba user accounts
- ▶ Synchronization of user passwords

SyncAgent for Linux features a variety of configuration options through which you can customize individual agent functions.

Synchronization of Linux system accounts can be used for terminal users who are directly working on the system (for example, through Telnet or ssh) as well as for users who are using Linux system applications, which make use of system accounts (for example, POP/IMAP server, Web server).

System requirements for installation of Sync Agent Daemon are:

- ▶ SuSE Linux Version 8.1 / SuSE Linux Enterprise Server 8 RC5 or higher
- ▶ Red Hat Linux 8.0 / Red Hat Linux ES U.2.1 or higher

Note: Please note that the supplied program libraries can vary widely with Linux distributions. Therefore, you have to make sure you are using the SyncAgent version that is compiled for your specific distribution.

The requirements for using Sync Agent for Linux are:

- ▶ TCP/IP protocol with static IP configuration
- ▶ Samba 2.2 or 3.0 is required for automatic creation of Samba accounts.

Installation

To install the SyncAgent:

1. Extract the file `sa_linux_X.X.X.tar.gz` into a new directory using the command:

```
tar zxvf sa_linux_1.0.1.tar.gz
```

2. Now change into directory `sa_linux` and execute the installation program with the command:

```
./sainstall
```

Permissions as root are required for installing SyncAgent.

3. During installation you will be asked for the SyncAgent program directory as well as the IP address of the SyncProxy server.

SyncAgent will be installed so that it automatically starts upon rebooting of the Linux system. You can customize this behavior to your requirements by changing the runlevel links.

Starting and stopping of SyncAgent

To start and stop SyncAgent the following script is used:

```
/etc/syncagent/syncagentctl
```

To start it execute the following command as root:

```
/etc/syncagent/syncagentctl start
```

Stopping works similar to starting:

```
/etc/syncagent/syncagentctl stop
```

If you change configuration parameters while SyncAgent is active you have to restart the agent for changes to become effective.

```
/etc/syncagent/syncagentctl restart
```

SyncAgent configuration

The configuration file for SyncAgent for Linux can be found at:

```
/etc/syncagent/syncagent.conf.
```

The default configuration of the Linux SyncAgent suits exactly our needs for this migration, so that there are no changes required.

Note: The SyncAgent creates missing groups automatically. This behavior can be changed in the file `syncagent.conf`.

The IP address of the SyncProxy (in case a different IP address is needed as configured during the installation) can be changed in the syncagent.conf.

If you want to modify the configuration, information about the configuration directives can be found directly in the file syncagent.conf file and in the SyncAgent for Linux manual, available from:

<http://www.servolution.comtarsia.com>

Enable the Servolution Logon Client SyncClient feature

On the Servolution Logon Client, the feature SyncClient has to be enabled, in order to send the SyncPacket to the SyncProxy/SyncAgent at each logon.

The SyncClient needs to be enabled in the Servolution Logon Client Configurator. There are three options to perform this task:

- ▶ Use the GUI
- ▶ Distribute registry files through software distribution method
- ▶ Import registry the information during processing of the logon script

See below for the corresponding windows registry settings to perform options b or c.

```
KEY: HKEY_LOCAL_MACHINE\SOFTWARE\PCS\GINA\  
"EnableSyncClient"=DWORD:1
```

This switch enables the Servolution Synchronization Agent.

```
KEY: HKEY_LOCAL_MACHINE\SOFTWARE\PCS\GINA\SyncProxy\  
"SyncProxy"="192.68.14.245" or "SyncProxy"="syncproxy.comtarsia.com"
```

This parameter defines IP address or host name of the Servolution Sync Proxy server.

The following parameters will be needed, but can remain as follows (the default values):

```
"ProxyPort"=dword:000007d1  
"SyncPacketTTL"=dword:00006  
"ConnectTimeout"=dword:00005
```

8.5.8 Resource migration to Samba

This section describes the necessary steps to migrate the OS/2 definitions onto the Samba server through the Servolution Migration Path.

Migration of aliases

As stated earlier, permissions for users are granted using the group assignments and user to group rather than share associations. The Servolution SyncAgent

creates missing groups on the Linux system automatically. If different groups should be used on Linux than on OS/2, the group mapping function can be used.

To create a share under Linux/Samba, a few simple steps are needed.

1. The share folder or directory has to be created first:

```
mkdir /shares/lanshare/
```

2. The access permission for the directory has to be set accordingly:

```
drwxrws--- 2 root transition 48 Jun 19 15:01 lanshare
```

3. The following section in the file `/etc/samba/smb.conf` needs to be added. The share must be defined in the Samba configuration file `smb.conf` (default under `/etc/samba/`).

Example 8-8 smb.conf for new share on Samba

```
[LANSHARE]
comment = LANSHARE
path = /shares/lanshare
create mask = 0660
directory mask = 0770
writeable = yes
public = no
valid users = @transition
```

4. Use your preferred copy mechanism to move data to the new share.

Now, all of the data has to be transferred to the target Samba share. During this step, it is highly recommended that no user is logged onto either of the servers to guarantee data integrity and consistency. This can be accomplished by stopping the OS/2 Netlogon services, and forcing logoff for all users, for example:

Assure that the Linux file permissions are set correctly after the files are copied from the OS/2 Server.

5. Switch servers

Change the server name in the alias path on the OS/2 Server to point to the Linux resource server with the following command: **NET ALIAS LANSHARE \\LNXSU**

So, the User management is still on the OS/2 Server, while the alias, and therefore the data is now on Linux. Access control is handled through a common group name or using a Group Mapping function.

From this point on the users are allowed to log on again. All data on the new Samba alias is available for users as it was before on OS/2.

Migration of home directory and roaming profile

After successfully migrating all general shares, the users' home directories and their roaming profiles are still left to be migrated.

Again, it is highly recommended that all users or at least the users currently being migrated, are not logged on at the time of migration.

1. Users were allowed to logon until this point. For all users which logged on via Logon Client at least once since the installation of the SyncAgent, a home directory was created on the Samba server. But it is not yet in production, the actively used home directory is still on the OS/2 Server.

2. Use your preferred copy mechanism to move data to the new share.

Not all of the data has to be transferred to the target Samba share. During this step, it is highly recommended, that no user is logged onto either of the servers to guarantee data integrity and consistency. This can be accomplished by stopping the OS/2 Netlogon services and forcing logoff for all users, for example.

Assure that the Linux file permissions are set correctly after the files are copied from the OS/2 Server.

3. Switch servers

The server name in the user object path (on OS/2) to the home directory has to be changed now to point to the home directory on the Linux server.

If the home directory should be assigned to the user at a fixed drive letter, the following syntax can be used: H:\LNXSU\LEIF which will assign Leif's home directory on drive H:.

```
NET USER LEIF /HOMEDIR:H:\LNXSU\LEIF
```

Now, user management is still on the OS/2 Server, while all shares including home directories and roaming profiles are on the Linux site.

From this point on the users are allowed to log on again.

OS2MigrateUsers

This section describes the migration of existing user accounts from OS/2 domains and servers to Linux domains and servers with the Comtarsia user migration tool.

The prerequisites for OS2MigrateUsers are:

- ▶ A fully configured and ready to use Servolution SyncProxy/SyncAgent setup.
- ▶ One Windows workstation with an installed Servolution Logon Client 3.0 or 3.1 with an enabled SyncClient option.

To test the installation, it is recommended that a logon with the configured Logon Client workstation be performed onto the OS/2 Server to assure that the logon user was automatically created on the Linux/Samba server during logon.

If the logon on to the OS/2 Server and the synchronization with the Linux/Samba server worked well, the actual user migration, which is performed by the command line tool **OS2MigrateUsers.exe** can be started. This tool can be found in the Logon Client distribution.

This tool performs an “at once” migration of all OS/2 user accounts including group membership and home directory information.

Run the tool **OS2MigrateUsers.exe** from the command line on the Windows workstation with the following parameters:

Note: There is no need to be logged on onto the OS/2 Server, a local Windows user also works fine.

```
OS2MigrateUsers.exe OS2DOMAIN OS2SERVER ADMIN_USER ADMIN_PWD LOGLEVEL FILTER
```

Table 8-3 OS2MigrateUsers parameters

Parameter	Explanation
OS2DOMAIN (required)	Specify the name of the OS/2 domain you want to migrate.
OS2SERVER (required)	The name of the OS/2 Server you want to migrate.
ADMIN_USER (required)	Specify the name of an user account with Administration privileges on the chosen OS/2 Server.
ADMIN_PWD (required)	The password for the user account specified under "ADMIN_USER"
LOGLEVEL (required)	Specify "1" or "2" (without quotation marks) With the option "1" all migrated users are logged into the file "OS2MigrateUsers_migrated.log" including a migration summary. If option "2" is specified, additionally to the output of level 1, group membership and home directory information of all migrated users is logged into the file.

Parameter	Explanation
FILTER (optional)	Specify a filter string to explicitly select specific user accounts for migration. If no filter is specified, all user accounts are select.

An asterisk can be used at the beginning and at the end of the filter string to select multiple accounts. Examples are:

- ▶ USER* selects all user accounts beginning with the string USER
- ▶ *USER selects all user accounts ending with the string USER
- ▶ *USER* selects all user accounts which are containing the string USER, no matter at what position
- ▶ USER selects the user account with the exact name USER

While the OS/2 user migration tool is running, dots are displayed on the command window to show the progress of the migration. One dot corresponds to fifty user accounts queried from the OS/2 Server. (This progress output just depends on the number of users on the OS/2 Server, and not on the number of users to be migrated.)

Depending on the network speed and on the number of users, the migration process can take from a few seconds up to an hour.

A raw estimate is to calculate one minute per 200 user accounts to be migrated.

All new user accounts created on the Linux/Samba server have a random password set, which will be automatically overwritten with the correct user password by the Servolution SyncAgent at the first logon of the user.

The description of the newly created user is set to SERV_TMP_USER_MIGT, which makes it possible to easily list all new user accounts. After the first logon of a new user the description is changed to SERV_TMP_USER.

After the migration tool has finished its work, a status screen is displayed indicating the result of the operation.

Example 8-9 Status screen of a user migration

```
OS2MigrateUsers finished. Status:
OS/2 Domain                SOMEDOMAIN
OS/2 Server                 PDC
Total users                 7
Users migrated              6
User migration errors       0 (see error log)
Users skipped                1 (Administrator)
```

Users skipped through filter	0 (Filter: *)
Time needed	1 s

There is a total of seven user accounts on the OS/2 Server.

Six user accounts are migrated, while zero are skipped because of the filter setting (*). The user *Administrator* is always skipped for security reasons, independently from the filter settings.

Errors during the migration process can be seen in the row *User migration errors* and are logged verbosely in the file *OS2MigrateUsers_error.log*, which is created by the migration tool in the current working directory.

8.5.9 User management migration to LDAP

The following section describes how to configure an IBM Directory Server 5.1 to act as the central user management platform.

The installation of the Directory Server is assumed to be finished.

Including Comtarsia schema into IBM Directory Server 5.1

First, store the Comtarsia schema file in the server's schema file folder.

With the tool `/usr/bin/ldapxcfg`, under the section **Manage schema files** the Comtarsia schema can be attached to the **Current schema files** in the Directory Server.

Next, the Directory Server has to be restarted, and the relevant object classes and attributes are ready to use.

LDAP SyncAgent

If user management should be migrated to the LDAP server, the Servolution LDAP SyncAgent should be installed at the same time as the other SyncAgents for the resource servers (also see 8.5.8 "Resource migration to Samba" on page 361).

The Servolution LDAP SyncAgent creates and synchronizes user accounts and group membership information into the LDAP directory, which allows for switching the user management from OS/2 to LDAP.

The LDAP SyncAgent can be installed on the same machine as the LDAP server but it also works remotely.

After the SyncAgent is installed and the LDAP synchronization module is enabled, the following configuration parameters needs to be adjusted.

Table 8-4 LDAP SyncAgent parameters

Parameter	Value	Description
LDAPHostname	Ldapsrv	The hostname of the LDAP server
LDAPPort	636	The port address of the LDAP server
LDAPAdminDN	cn=root	A user DN with administration rights on the LDAP server
LDAPAdminPassword	Demo	The password for the administration user
LDAPUserPrefix	cn=	Prefix for user objects
LDAPUserSuffix	,cn=Users	Suffix for user objects
LDAPGroupPrefix	cn=	Prefix for LDAP group definitions
LDAPGroupSuffix	,cn=Users	Suffix for LDAP group definitions

This way user password and group membership information is automatically synchronized between OS/2 and LDAP at each logon while user management is still on the OS/2 Server.

LDAP migration tool

The Servolution OS/2 to LDAP migration tool allows for easy and complete migration of all OS/2 objects and attributes to an LDAP server.

The major difference to the LSMT tools described in 3.3 “Collecting data using LSMT” on page 68, is that just one execution of the program is needed to extract all LAN server objects and attributes, and this imports the data into an LDAP directory in the same step. There is no need for import scripts.

An important advantage of the migration of the user management to LDAP is that all existing OS/2 definitions remain accessible for users without interruption.

The prerequisites are:

- ▶ At least one Windows workstation with a Servolution Logon Client installed and configured.

- ▶ A ready-to-use IBM Directory Server 5.1 (This document describes the migration to the IBM Directory Server 5.1, but any other LDAP server can be used.)
- ▶ One or more OS/2 Servers

Copy the files **OS2LDAPMigration.exe** and **OS2LDAPMigration.ini** to a directory on a Windows workstation and open the ini-file with an editor (for example, Notepad). This figure below shows the empty LDAP structure, which has to be created on the LDAP server before the migration tool is started.create one organization object (o=somedomain):

- ▶ Create a realm for the users and groups (cn=Users).
- ▶ Create *Organizational Units* for shares and network applications (ou=Shares and ou=NetworkApplications).

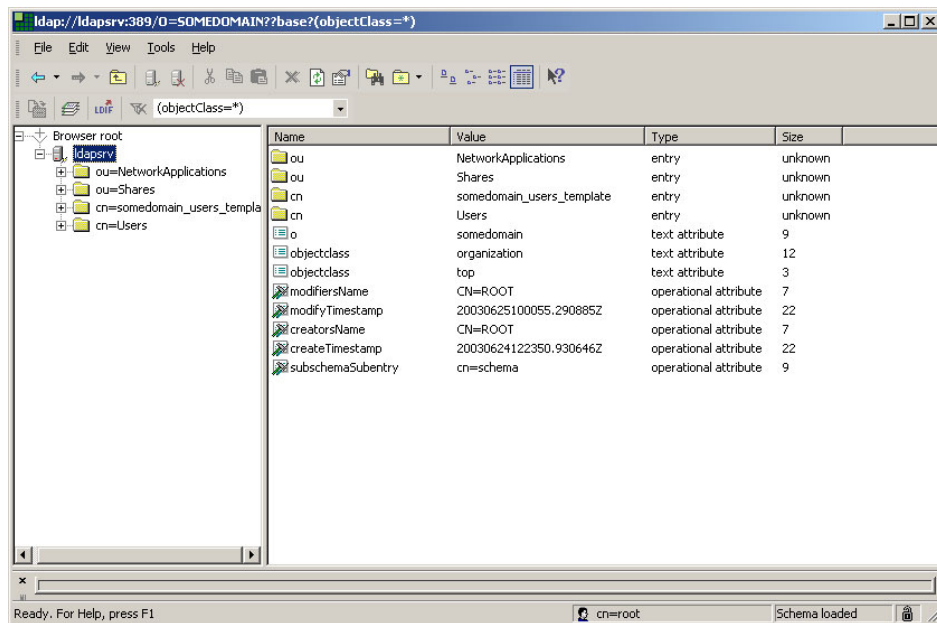


Figure 8-42 LDAP Configuration step 1

The following parameters must be adjusted in the.ini file to fit your environment.

Table 8-5 LDAP configuration parameters

Parameter	Value	Description
LDAPHostname	Ldapsrv	The hostname of the LDAP server

Parameter	Value	Description
LDAPPort	636	The port address of the LDAP server
LDAPAdminDN	cn=root	A user DN with administration rights on the LDAP server
LDAPAdminPassword	demo	The password for the administration user
LDAPUserPrefix	cn=	Prefix for user objects
LDAPUserSuffix	cn=Users	Suffix for user objects
LDAPGroupPrefix	cn=	Prefix for LDAP group definitions
LDAPGroupSuffix	cn=Users	Suffix for LDAP group definitions
LDAPSharePrefix	cn=	Prefix for LDAP share definitions
LDAPShareSuffix	ou=Shares	Suffix for LDAP share definitions
LDAPNWAPrefix	cn=	Prefix for LDAP network applications definitions
LDAPNWSuffix	ou=NetworkApplications	Suffix for LDAP network applications definitions
OS2Domain	SOMEDOMAIN	The name of the OS/2 domain you want to migrate
OS2Server	SOMESERVER	The name of the OS/2 Server you want to migrate
OS2AdminUser	ADMIN	The name of an user account with Administration privileges on the chosen OS/2 Server
OS2AdminPassword	Demo	The password for the OS2AdminUser account
LogLevel	1	"1" for normal logging, "2" for verbose logging

Parameter	Value	Description
Filter	*	Specify a filter string to explicitly select specific user accounts for migration. If no filter is specified, all user accounts are selected.

An asterisk can be used at the beginning and/or at the end of the filter string to select multiple accounts. Examples are:

- ▶ USER* selects all user accounts beginning with the string USER
- ▶ *USER selects all user accounts ending with the string USER
- ▶ *USER* selects all user accounts which are containing the string USER, no matter at what position
- ▶ USER selects the user account with the exact name USER

Now, the migration tool can be executed.

After the LDAP migration tool has finished, all OS/2 objects and attributes are created on the LDAP server at the previously specified locations as shown in Figure 8-43.

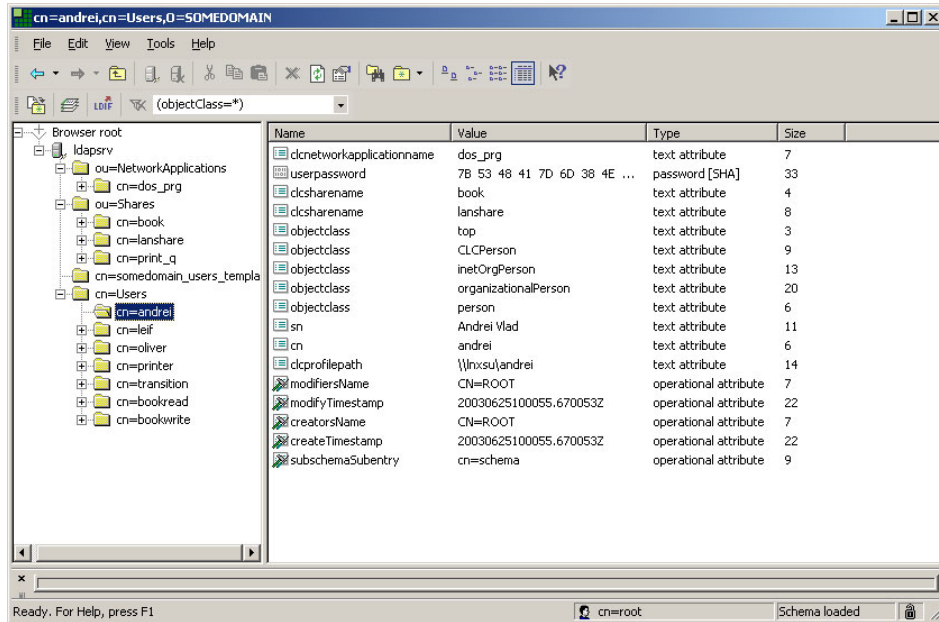


Figure 8-43 LDAP Configuration step 2

For a more detailed description of the migration process from OS/2 to LDAP, please see:

<http://www.servolution.comtarsia.com>

SyncProxy configuration changes

The SyncProxy configuration needs to be changed when the primary user management is on LDAP. This is necessary in order to perform correct counter checking of the sync requests.

In the SyncPacket configurator, in the tab **Security** change the Master Domain Name to LDAP, and in the tab **LDAP** include the same settings as on the Logon Client.

A more detailed description of this process can be found on the Servolution Web site:

<http://www.comtarsia.com>

Sample Logon Client configuration

The following configuration settings are required to switch the Servolution Logon Client to the LDAP Logon mode.

The easiest way to create fully functional LDAP Registry settings is by using the Logon Client Configurator, and afterwards exporting the Registry subkey.

Example 8-10 LDAP Registry settings from Logon Client Configurator

```
KEY: HKEY_LOCAL_MACHINE\SOFTWARE\PCS\GINA\  
"EnableLDAP"=DWORD:1
```

```
KEY: HKEY_LOCAL_MACHINE\SOFTWARE\PCS\GINA\LDAP\  
"LDAPAppendBaseDN"=DWORD:1  
Set Base DN to the correct organisation name, in this example "o=somedomain"  
"LDAPBaseDN"="o=somedomain"  
"LDAPServerTyp"=DWORD:7  
"LDAPUserDNPrefix"="cn="  
"LDAPUserDNSuffix"=",cn=Users"  
User DN Suffix is the remaining "path" between the name and the top of the  
hierarchy, beginning with a ",", in this example ",cn=Users"
```

Hence, a full user DN would be in this example:

```
cn=oliver,cn=Users,o=somedomain
```

The next step is to set the LDAP server name in the Registry or in the Logon Client Configurator:

```
KEY: HKEY_LOCAL_MACHINE\SOFTWARE\PCS\GINA\LDAP\LDAPServers\ldapsrv
```

With this configuration, the Servolution Logon Client is ready for a successful logon to an LDAP server, and all user management is on the LDAP server.

8.6 Summary

This chapter has described several tools available from various sources that may help with different aspects of an OS/2 Server migration. For more information on these tools and their applicability to your environment, please contact the individual vendors.



Linux for OS/2 administrators

This chapter covers some of the basics of Linux administration and security for readers that are familiar with OS/2 administration. It is not intended to be a comprehensive discussion of Linux administration, but rather covers some of the basics to provide the reader with a flavor for the considerations, and the kinds of tools and facilities available for administering Linux systems.

9.1 Linux security

This section describes a few concepts and procedures related to security in a Linux environment. It is not meant to be a comprehensive body of work, but rather to introduce some basic concepts and examples.

Because of expanding global communications and Internet connectivity, more and more people have access to your servers, and not all of these people have good intentions. Therefore, servers need protection against attacks while at the same time granting access to those who need it. Keep in mind that no server, regardless of the operating system, is completely secure.

The levels of security that we discuss in this chapter are:

- ▶ Physical security
- ▶ System security
- ▶ Network security
- ▶ Backup security

9.1.1 Physical security

Physical security applies no matter what operating system is being used. The first step in securing your server is limiting physical access to the machine. Consider all of the following:

- ▶ Lock the server in a special room to which only administrators have access.
- ▶ Lock the server console with a password.
- ▶ Lock the system covers. This way no one has easy access to the inside of your computer. Otherwise, someone could insert another hard drive, boot from it, and potentially gain access to the other drives in the system.
- ▶ Secure the floppy and CD-ROM drives. After you have installed all of the software, consider removing the floppy and CD-ROM from the BIOS boot list.
- ▶ Lock the BIOS setup utility with a password.

Attention: If you enable a power-on password in BIOS, then your system will no longer reboot automatically in the event of a power failure.

9.1.2 System security

Not every user on the system needs root access. Though it is easier to work as root, it should be granted only to those administering the server. If a user does not need access to a resource, do not grant access.

File permissions

In Linux almost every resource (files, directories, symbolic links, disks, modems, and so forth) is considered a *file*, and file permissions give access to the resource. From a shell the permissions of a file are viewed with the command `ls -l` at the command line, as shown in Example 9-1.

Example 9-1 Example of file permissions for /etc/passwd

```
# ls -l /etc/passwd
-rw-r--r-- 1 root  root   873 Apr  4 15:27 /etc/passwd
```

This command gives the long listing format of the file `/etc/passwd`. In addition to the name of each file, it prints the file type, permissions, number of hard links, owner name, group name, size in bytes, and time stamp (by default, this is the time of the last modification). The type and the permission is the cryptic string of letters and dashes at the beginning of the line. The first character of the 10 character long code is the type of the file; in this case it is a dash which means this is a plain file. The possible file types are:

-	Plain file
d	Directory
l	Symbolic link (like a shortcut)
b	Block device (drives)
c	Character device (terminals, modems)

The next nine characters describe the permissions on the file. They are organized in groups of three. The first group gives the permissions of the owner of the file (in this case the user `root`), the second the permissions of the group (in this case the group `root`), and the last three characters give the permissions for any other user on the system.

A group of three characters are built as follows:

- ▶ First character is an `r` which means permission to read the file.
- ▶ Second is a `w` which stands for write permission.
- ▶ The last character is `x` for execute rights on a program or list rights if the file is actually a directory. Also, `s`, `S`, `t`, and `T` are possible values for this character, but these permissions are less frequent and beyond the scope of this book.

In this example, the permissions `-rw-r--r-- root root` means:

- ▶ Read and write access for the user `root`
- ▶ Read rights for anyone who is a member of the group `root`
- ▶ Read rights for any other user on the system

On a Linux system, ordinary users only have write access to their \$HOME directory (also known as ~) and the /tmp directory.

Passwords

Passwords are a ubiquitous means of security, and every company should determine and set password rules based on their security requirements.

Each password has to be chosen with care. There are two components of password strength:

- ▶ **Quantity:** This is simply a minimum number of characters required before a password is acceptable.
- ▶ **Quality:** This is a more complex requirement that dictates that the password must contain a combination of lower and uppercase letters, numbers, or other symbols.

In Linux, the default minimum password length is five, but there is also a maximum length of eight. However, this can and should be changed. Linux offers a range of options to guard against weak passwords, and we detail a number of them in this section. There are also many printed references, as well as Linux Web sites, which discuss password policies and recommended procedures.

Password settings in SuSE

SuSE has a tool for system administration called YaST2. This tool can be used either in text mode or in graphical user mode.

Note: You have to be logged in as root to have access to all areas of YaST2.

To quickly change the password settings, use the graphical YaST2. Click **Start Application -> System -> YAST2**, click **Security and Users**, then **Security Settings** as shown in Figure 9-1.

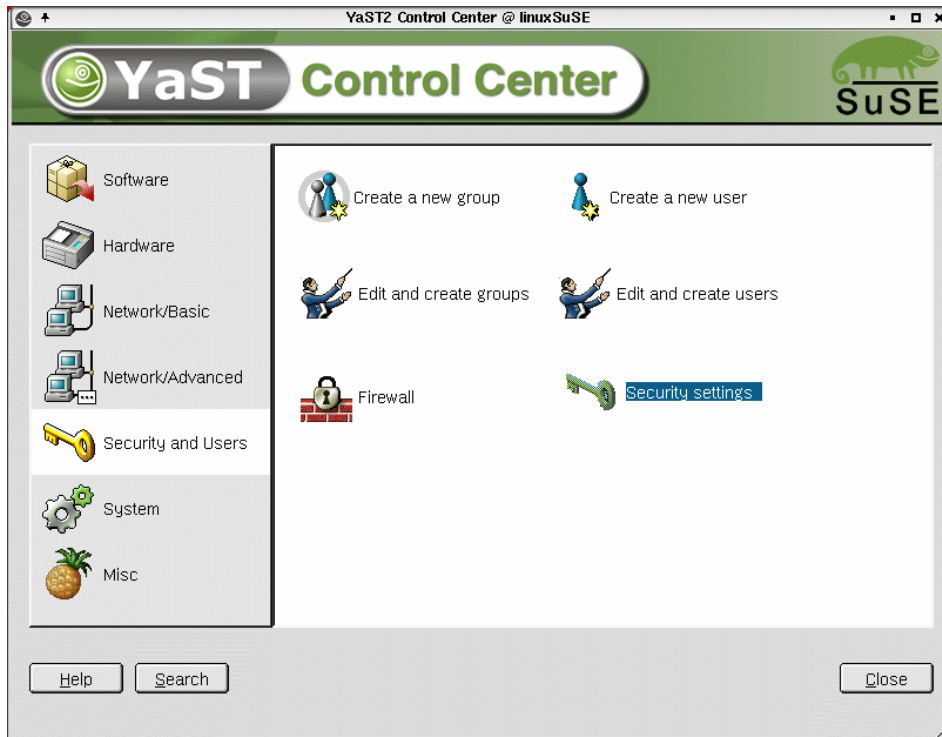


Figure 9-1 Security settings in SuSE

Check **Custom Settings** and click **Next** to display the Password settings window. There are a number of options regarding passwords. General recommendations are shown in Figure 9-2:

- ▶ Enable **Checking new passwords**.
- ▶ Enable **Plausibility test for password**.
- ▶ Enable **Activate MD5 encryption for passwords**.

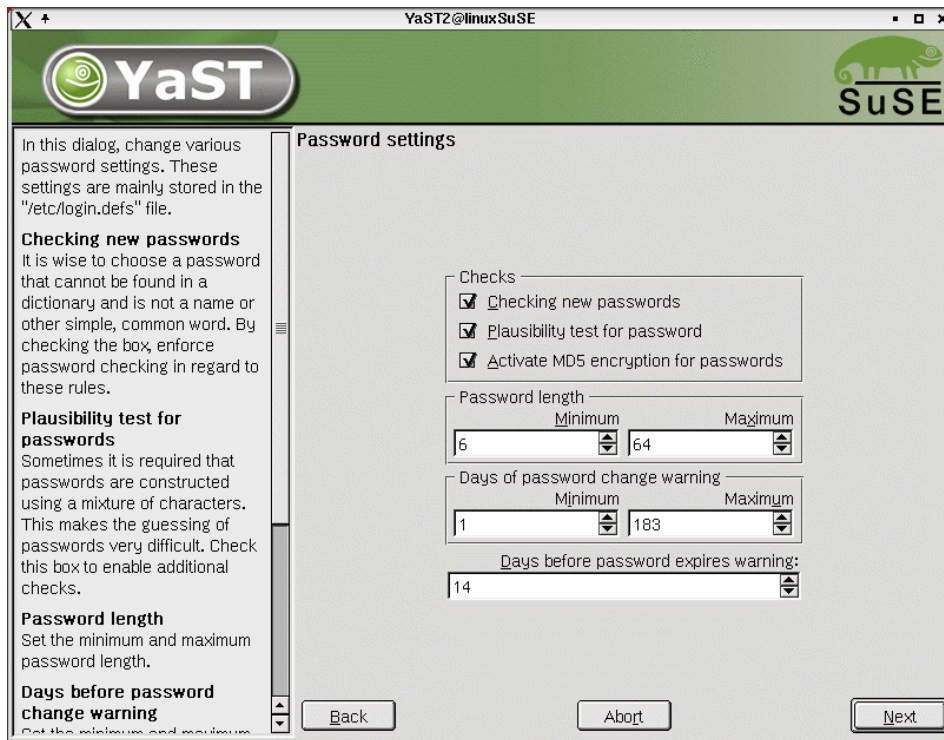


Figure 9-2 Password settings

There are a lot of opinions regarding minimum password length; the consensus seems to be that the password length should be at least six to eight characters. The root password should certainly be eight or more characters in length. Table 9-1 shows different password lengths and the total number of password possibilities, if no restrictions are in place. As you can see, the use of just lowercase letters in a password seriously reduces the number of possible combinations.

In addition to minimum length, you should also change the maximum length to a much higher number than eight; we opted for 64 as shown in Figure 9-2.

Table 9-1 Password length and total possibilities

Password length	Combinations using lowercase letters (26)	Combinations using letters, numbers, and special characters (94)
5	11,881,376	7,339,040,224
6	308,915,776	689,869,781,056
7	8,031,810,176	64,847,759,419,264
8	208,827,064,576	6,095,689,385,410,816

Aside from setting policies regarding the length and complexity of passwords, users should also be forced to change their passwords periodically. A trade-off is that making users change their passwords too often, or requiring passwords to be too complex may result in the user writing down the password, thereby defeating your overly stringent security measures. If twice a year seems a reasonable compromise, set the Maximum for Days of password change warning to 183.

Attention: During our use of YaST2, the Days of password change warning was not set correctly. You can verify that your changes have been saved by viewing the `/etc/login.defs` file, as detailed in “Password settings in Red Hat” on page 379” later in this section.

Click **Next** to view the remaining security options. The default settings, which include a three second log-in delay for failed attempts, and a record of each failed attempt, are selected.

Tip: If you want to increase security even further, investigate switching to Kerberos authentication. There are many sources to learn more about Kerberos, for example, the Kerberos pages of MIT at:

<http://web.mit.edu/kerberos>

Another good source is the Linux Security how-to, which can be found along with numerous other helpful documents at the Linux Documentation Project Web site:

<http://tldp.org/docs.html>

Password settings in Red Hat

For Red Hat 7.2, log in as root and modify the `/etc/login.defs` file directly using an editor of your choice, as shown in Example 9-2.

If your company already has a Linux security policy, make certain to utilize it in conjunction with our recommendations.

Example 9-2 /etc/login.defs file

```
# Password aging controls:
#
#     PASS_MAX_DAYS   Maximum number of days a password may be used.
#     PASS_MIN_DAYS   Minimum number of days allowed between password
changes.
#     PASS_MIN_LEN     Minimum acceptable password length.
#     PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   183
PASS_MIN_DAYS   0
PASS_MIN_LEN     6
PASS_WARN_AGE   14
```

Next, type **setup** at the command prompt to verify that you have enabled MD5 passwords.

1. Select **Authentication Configuration** by pressing Enter.
2. Use the Tab key to navigate to the **Next** option and press Enter. This will display the screen shown in Figure 9-3.
3. Make certain that both **Use Shadow Passwords** and **Use MD5 Passwords** are selected. (You can use the spacebar to select and deselect options.)
4. Press the Tab until **OK** is highlighted; press Enter to accept.
5. Quit the **setup** program.

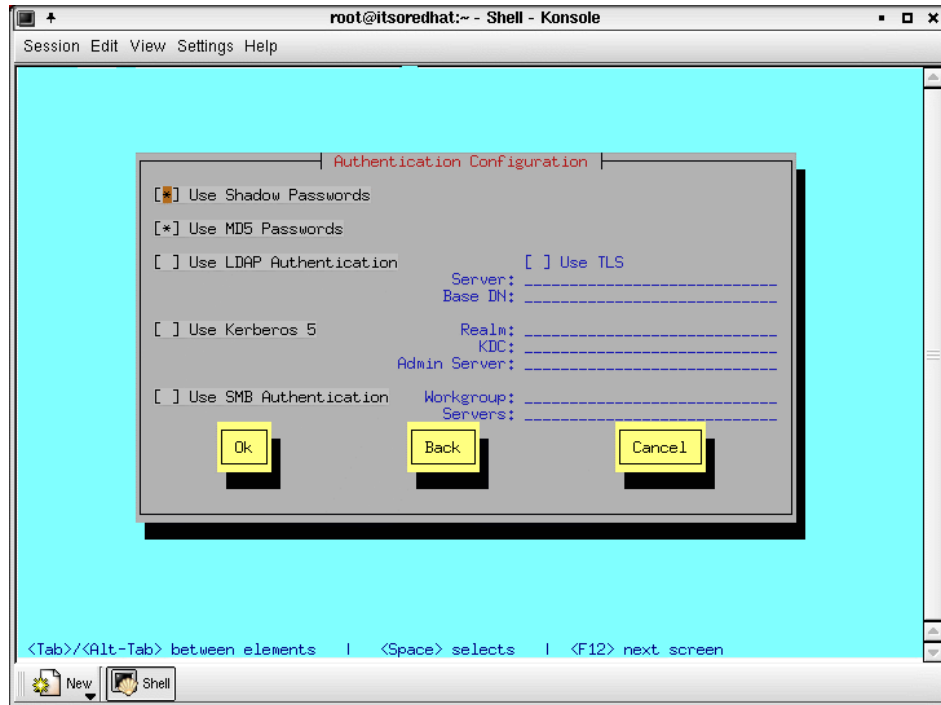


Figure 9-3 Authentication Configuration for Red Hat 7.2

9.1.3 Network security

This section covers both basic and advanced network security. For more information, visit the following Web site:

<http://www.linuxsecurity.com>

Basic network security

In the UNIX system world, software that is able to connect to (exchange information with) other software on the same system or another system is called a daemon. Usually, the daemon listens on a specified IP address and port. A server normally has many daemons running at the same time, such as the ftp daemon, ssh daemon, and so forth. Through these daemons, another system can connect to the server and exchange information.

Daemons are divided into two categories: those started by root user; and the rest, started by other users. The daemons started by root generally listen on ports below 1024.

If a daemon has a programming “bug” or there is an unusual circumstance, such as information coming too fast for the daemon to handle, or reception of an invalid command, and the daemon may crash. When a daemon crashes, it often returns a prompt without requesting a password and whoever is connected at that time with the daemon now has the prompt. If the daemon was started by root, then when it crashes, it returns a root prompt, which is very dangerous. Minimizing the number of daemons run by root is an important step in securing your server.

After the installation of Linux, there are many ports open by default because a number of daemons are automatically started. To increase security, as well as performance, you should stop daemons that you do not need.

Table 9-2 explains some of the frequently used services available for Linux.

Table 9-2 Linux daemons

Name of the service	Observations	Port
crond	It runs user-specified programs at periodically scheduled times. It is useful for log rotation, for example.	N/A
ftpd	This is an ftp (file transfer protocol) daemon common on SuSE. Use it to move files from one server to another. You can use the scp command with an SSH shell.	21
gpm	It adds mouse support to a text console	N/A
httpd	Linux Web server	80
ipchains	Firewall tool	N/A
iptables	Firewall tool	N/A
keytable	It loads the selected keyboard map	N/A
kudzu	This runs a hardware probe akin to plug and play. After you install your server hardware, you can turn this off.	N/A
lpd	Print daemon.	515
network	Activates/Deactivates all network interfaces configured to start at boot time.	
nfs	A file sharing protocol across TCP/IP.	2049
sendmail	An SMTP server.	25
snmpd	A management protocol. You should enable this daemon only if you have implemented SNMP.	161

Name of the service	Observations	Port
ssh	A secure shell for remote administration. Use it to remotely administer the server from a shell.	22
syslog	The facility by which many daemons log messages to various system files.	N/A
telnet	A shell for remote administration. Use SSH for secure remote administration.	23
wu-ftpd	An ftp (file transfer protocol) daemon common on Red Hat. Use it to move files from one server to another. You can use the <code>scp</code> command with an SSH shell.	21
xfs	The X Font Server.	N/A
xinetd	Runs other daemons on demand.	N/A

Starting and stopping daemons

Starting and stopping daemons can be done by logging in as root to KDE and launching the SysV - Init Editor by selecting **Start Application -> System -> Configuration -> SysV Init Editor** on SuSE or **Start Application -> System -> SysV Init Editor** on Red Hat. (See Figure 9-4.)

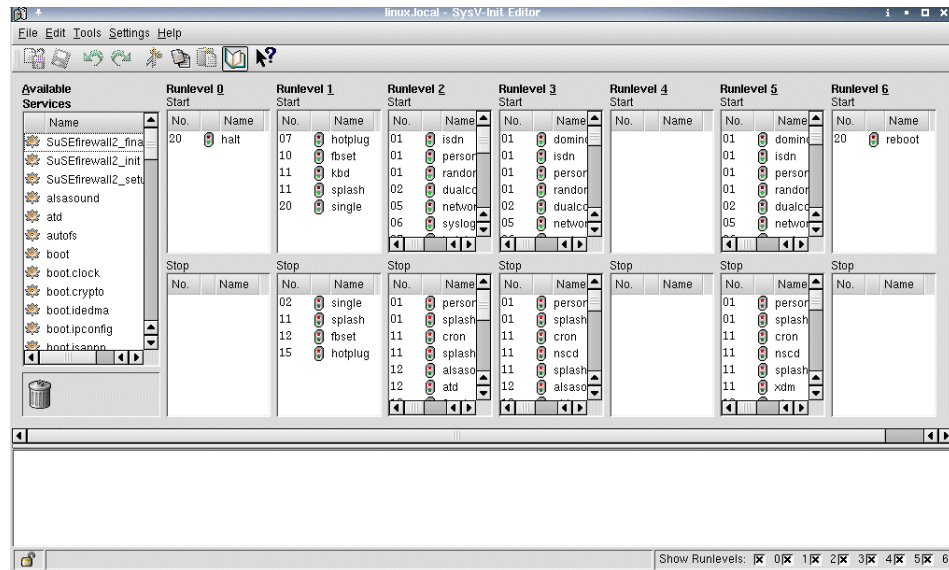


Figure 9-4 SysV Init Editor

Before using the SysV Init Editor, an understanding of Linux runlevels is recommended. Windows really has only two runlevels: Recovery and Normal. *Recovery* is only used when there is a problem with the system. Most of the time Windows runs in *Normal* mode.

Linux usually has six runlevels. Runlevel 0 is used to shut down the server; runlevel 6 is used to restart the server. Runlevel 1 (Single user mode) is used like the Windows recovery mode. Most systems normally run at runlevel 3 (command line) or runlevel 5 (X-Windows).

The top row of boxes in Figure 9-4 shows the services that will start when the system enters each runlevel, the bottom row of boxes show what services will be stopped when the system enters that runlevel.

Note: A service should *not* appear in both the Start and Stop boxes for a runlevel.

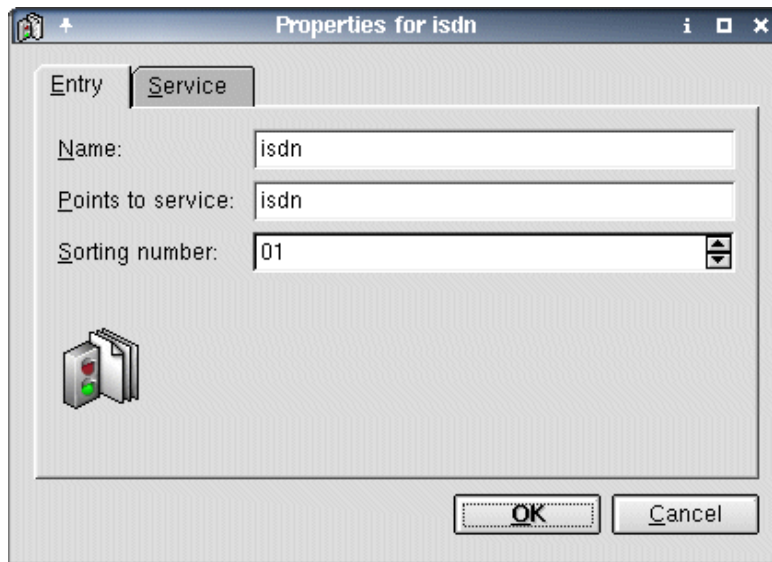


Figure 9-5 Properties for a service

To stop/start a service, click on the service (see Figure 9-5) and then go to the Service tab and click the **Start** or **Stop** button (see Figure 9-6).

To prevent a service from starting when entering a runlevel, drag and drop the service from the runlevel to the Trash can.

To start a service when entering a runlevel, drag and drop the service from the Available Services list to the start box of the appropriate runlevel.

To stop a service when entering a runlevel, drag and drop the service from the Available Services list to the start box of the appropriate runlevel.

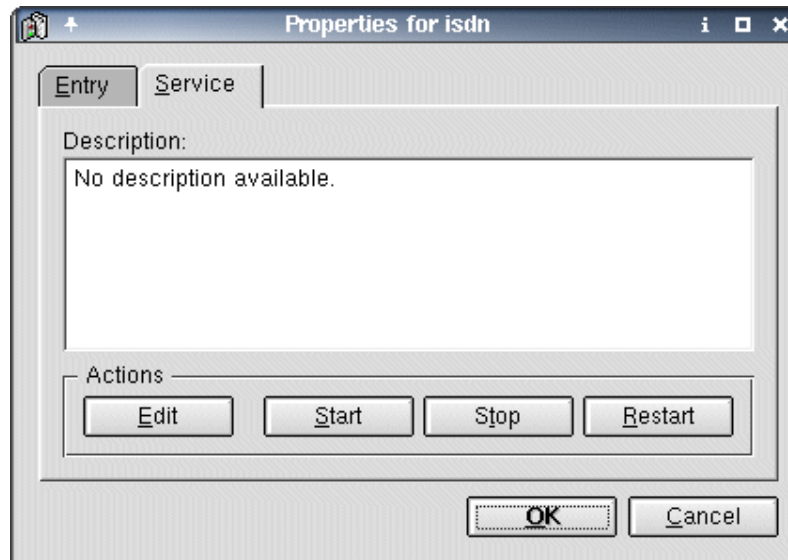


Figure 9-6 Start/Stop a service

Showing running daemons

To see what daemons are listening (accepting connections) on your server, log in as root and issue the command `netstat -a | grep "LISTEN "` as shown in Figure 9-7. In this way, you can always check to see if your daemons are listening.

Note: Linux is case-sensitive, so LISTEN must be uppercase as in this example.

```

Session Edit View Settings Help
itsosuse:~ # netstat -algrep "LISTEN "
tcp      0      0 *:7937          *:*          LISTEN
tcp      0      0 *:7938          *:*          LISTEN
tcp      0      0 *:x11           *:*          LISTEN
tcp      0      0 *:9616          *:*          LISTEN
tcp      0      0 *:25680         *:*          LISTEN
tcp      0      0 *:9617          *:*          LISTEN
tcp      0      0 *:9618          *:*          LISTEN
tcp      0      0 *:9619          *:*          LISTEN
tcp      0      0 *:ftp           *:*          LISTEN
tcp      0      0 *:ssh           *:*          LISTEN
itsosuse:~ # █

```

Figure 9-7 netstat -a | grep "LISTEN " command output

Securing daemons

If a daemon is required to run, control over who (what users) can access the services provided by that daemon should still be enforced. Files such as `/etc/hosts.allow` and `/etc/hosts.deny` are used to limit access to a system's services.

In the file `/etc/hosts.allow`, you can set who *can* connect to your machine on different ports, as shown in Example 9-3.

Example 9-3 `/etc/hosts.allow` file

```

# cat /etc/hosts.allow
sshd: 192.168.1.0/255.255.255.0
sshd: 192.168.234.0/255.255.255.0
in.ftpd: 192.168.0.0/255.255.0.0

```

This means only clients with an IP address between 192.168.1.1 and 192.168.1.254 or 192.168.234.1 and 192.168.234.254 can connect to the `ssh` server, while only those with an IP address between 192.168.0.1 and 192.168.255.254 can connect to your `ftp` server.

The file `/etc/hosts.deny` sets who is *not* allowed to connect to the machine on different ports, as shown in Example 9-4.

Example 9-4 *hosts.deny* file

```
# cat /etc/hosts.deny
sshd: 10.10.10.0/255.255.255.0
in.ftpd: 10.10.99.0/255.255.255.0
```

This means clients between 10.10.10.1 and 10.10.10.254 cannot connect to the **ssh** server, while clients between 10.10.99.1 and 10.10.99.254 cannot connect to the FTP server.

Tip: For best security practices the `/etc/hosts.deny` should contain all: `all deny`. This means that nobody can connect to the daemons protected by `tcpd` (see *man tcpd*) unless they are in the `/etc/hosts.allow`.

Use the **ssh** daemon instead of the **telnet** daemon because SSH encrypts all data between your client and server, and therefore provides another level of security over protocols such as Telnet.

Advanced network security

To remotely administer servers in a very secure manner, use a different physical network if possible. In other words, use different network adapters and different switches.

Note: The administrative network does not have to be a high speed network. You can use older hubs or switches.

Creating a separate network for administration provides the following advantages:

- ▶ You do not have to worry about someone stealing your password.
- ▶ You can update your software through the administration network, so your client will not notice a performance decrease.
- ▶ In case your high speed network fails, you can use the administrator network for a short period of time.

9.1.4 Backup security

Another method to gain access to your information is by stealing backup tapes. In this way, it is possible for someone to read your information, but not to modify it.

There are two ways to back up your server:

- ▶ A tape or a library directly attached to your server

- ▶ A backup server with a tape or library attached to it

Make certain to lock your tapes in a safe place, and if you are using a backup server, be sure to use a username and a password to back up or restore your files.

Operating system patches

Both SuSE and Red Hat provide easy ways to keep you system up-to-date with the latest security patches. See YaST2 for SuSE and RHN (Red Hat Network) Red Hat for more information.

9.2 Linux administration

Linux is a powerful operating system with many capabilities. This can make the administration of Linux seem like a daunting task. However, there are many tools available and administration is easier than it may appear. This section discusses basic administrative tasks, such as creating a partition, creating a file system, creating scripts, and modifying the **crontab** that allows for launching various administrative tasks on a scheduled basis.

9.2.1 Partitions

The tool to create, erase, or modify a partition is **fdisk**. To be able to use it, log in as root to a shell and type **fdisk /dev/sda**, where **sda** is the first SCSI hard disk. If you are not using SCSI, then the first hard disk will be **hda**. To list the partitions on a SCSI hard disk, type **fdisk /dev/sda -l** as shown in Example 9-5.

Example 9-5 Partition list

```
# fdisk /dev/sda -l

Disk /dev/sda: 240 heads, 63 sectors, 2584 cylinders
Units = cylinders of 15120 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1    *           1           821     6206728+   7  HPFS/NTFS
/dev/sda2                822         2584     13328280   f  Win95 Ext'd (LBA)
/dev/sda5                1365         2584     9223168+   b  Win95 FAT32
/dev/sda6                822         1329      3840417   83  Linux
/dev/sda7                1330         1364      264568+   82  Linux swap
```

Partition table entries are not in disk order

Note: Your disk partitions will likely be different from the example.

Linux has the following partition numbering system:

- ▶ From 1 to 4 are primary partitions
- ▶ From 5 to 16 are logical partitions

To view all **fdisk** commands, start **fdisk** interactively with **fdisk /dev/sda**, then type **m** as shown in Example 9-6.

Example 9-6 List of commands

```
Command (m for help): m
Command action
  a  toggle a bootable flag
  b  edit bsd disklabel
  c  toggle the dos compatibility flag
  d  delete a partition
  l  list known partition types
  m  print this menu
  n  add a new partition
  o  create a new empty DOS partition table
  p  print the partition table
  q  quit without saving changes
  s  create a new empty Sun disklabel
  t  change a partition's system id
  u  change display/entry units
  v  verify the partition table
  w  write table to disk and exit
  x  extra functionality (experts only)
```

To delete a partition, follow Example 9-7.

Example 9-7 Deleting a partition

```
Command (m for help): d
Partition number (1-7): 7
```

```
Command (m for help):
```

To create a logical partition, follow Example 9-8.

Example 9-8 Creating a partition

```
Command (m for help): n
Command action
  l  logical (5 or over)
  p  primary partition (1-4)
l
First cylinder (1330-2584, default 1330):
Using default value 1330
```

Last cylinder or +size or +sizeM or +sizeK (1330-1364, default 1364):
Using default value 1364

Command (m for help):

Note: The logical option will only appear for a new partition if an extended partition has already been created.

After creating a partition, change the partition's type. In Linux, the partition type is coded as a number or id. By default, Linux creates a partition with ID 83, which means it is designated as a Linux partition.

In Example 9-9, you can see all the partition types supported by Linux at this time.

Example 9-9 All partition types supported by Linux

Command (m for help): t

Partition number (1-7): 6

Hex code (type L to list codes): l

0 Empty Wizard hid	1b Hidden Win95 FA	64 Novell Netware	bb Boot
1 FAT12 (FAT-	1c Hidden Win95 FA	65 Novell Netware	c1 DRDOS/sec
2 XENIX root (FAT-	1e Hidden Win95 FA	70 DiskSecure Mult	c4 DRDOS/sec
3 XENIX usr (FAT-	24 NEC DOS	75 PC/IX	c6 DRDOS/sec
4 FAT16 <32M	39 Plan 9	80 Old Minix	c7 Syrinx
5 Extended data	3c PartitionMagic	81 Minix / old Lin	da Non-FS
6 FAT16 CTOS / .	40 Venix 80286	82 Linux swap	db CP/M /
7 HPFS/NTFS Utility	41 PPC PReP Boot	83 Linux	de Dell
8 AIX	42 SFS	84 OS/2 hidden C:	df BootIt
9 AIX bootable access	4d QNX4.x	85 Linux extended	e1 DOS
a OS/2 Boot Manag	4e QNX4.x 2nd part	86 NTFS volume set	e3 DOS R/0
b Win95 FAT32	4f QNX4.x 3rd part	87 NTFS volume set	e4 SpeedStor
c Win95 FAT32 (LB	50 OnTrack DM	8e Linux LVM	eb BeOS fs
e Win95 FAT16 (LB	51 OnTrack DM6 Aux	93 Amoeba	ee EFI GPT
f Win95 Ext'd (LB (FAT-12/16/	52 CP/M	94 Amoeba BBT	ef EFI
10 OPUS	53 OnTrack DM6 Aux	9f BSD/OS	f1 SpeedStor
11 Hidden FAT12	54 OnTrackDM6	a0 IBM Thinkpad hi	f4 SpeedStor

12	Compaq diagnost secondary	55	EZ-Drive	a5	BSD/386	f2	DOS
14	Hidden FAT16 <3 raid auto	56	Golden Bow	a6	OpenBSD	fd	Linux
16	Hidden FAT16	5c	Priam Edisk	a7	NeXTSTEP	fe	LANstep
17	Hidden HPFS/NTF	61	SpeedStor	b7	BSDI fs	ff	BBT
18	AST SmartSleep	63	GNU HURD or Sys	b8	BSDI swap		

After creating a partition and setting its type, press **W** to commit the changes to the hard disk drive.

Linux uses a specific partition type (ID 82) for its swap space. When you install Linux make sure you create a swap partition. If you wish you can create more than one swap partition. A swap partition cannot be larger than 2048 MB.

Note: The changes you make to partitions are not committed until you press **w**, so in case of a mistake, press **q** to exit without saving your changes.

9.2.2 File systems

Once a partition is created, it has to be formatted with a file system. To do so, you have to choose how you will format it. For a Linux partition, there are several format choices such as ext2, ext3, jfs, and so on. In Example 9-10, an ext2 file system is created through the shell command line.

Example 9-10 Formatting a Linux partition

```
# mkfs.ext2 /dev/sdb1
mke2fs 1.23, 15-Aug-2001 for EXT2 FS 0.5b, 95/08/09
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
384768 inodes, 769104 blocks
38455 blocks (5.00%) reserved for the super user
First data block=0
24 block groups
32768 blocks per group, 32768 fragments per group
16032 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912
```

```
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

In Example 9-11, the swap partition is formatted.

Example 9-11 Formatting a swap partition

```
#mkswap /dev/sdb2
Setting up swapspace version 1, size = 1036378112 bytes
```

Note: Do not create a swap partition more than twice the size of your RAM memory.

Linux systems have a different way to mount and access partitions as compared with OS/2. Linux systems do not use letters assigned to a partition like OS/2 or Windows, the partition is “mounted” in an empty directory. Once the partition is mounted everything, that is written in that directory is actually written to the partition.

Create a directory where the formatted partition will be mounted, for example type `mkdir /data`, then modify the file `/etc/fstab` by adding the lines shown in Example 9-12, so the partition will be mounted at boot time.

Example 9-12 Adding the partition in file /etc/fstab

<code>/dev/sdb1</code>	<code>/data</code>	<code>ext2</code>	<code>defaults</code>	<code>1 1</code>
<code>/dev/sdb2</code>	<code>swap</code>	<code>swap</code>	<code>defaults</code>	<code>0 0</code>

When you reboot the server, your new file system will be mounted, or you can mount it manually by typing `mount /data` as root.

9.2.3 Scripts

OS/2 and Windows each have a single shell environment by default. Scripts can be written using the batch commands or by using installable languages such as REXX. In a Linux environment, you can also create shell scripts. There are more options for the type of shell that you might use as well as the languages that each support. Shell scripts are a powerful method by which to customize your Linux server. The following script will erase the log files that are more than 2 months old.

Note: Each shell has its own syntax for scripts. The scripts we created are made for the **BASH** shell.

Example 9-13 Log eraser

```
#!/bin/bash

## Log eraser ##
```

```
LPATH=/var/log
NR_OF_DAYS=60

for i in `find $LPATH -atime +$NR_OF_DAYS`
do
rm -f $i
done
```

-
- ▶ The first line `#!/bin/bash` specifies the environment that the script will run in. This line is to be treated as is and should not be modified.
 - ▶ The sixth line sets the variable `LPATH` to equal `/var/log` and the seventh line sets the variable `NR_OF_DAYS` to 60. We recommend that you use variables because it makes it easier to debug your script.
 - ▶ `$LPATH` and `$NR_OF_DAYS` indicate that you wish to use the value of the specified variable.
 - ▶ `find $LPATH -atime +$NR_OF_DAYS` will search in the `/var/log` directory for files older than 60 days.

Note: For more information about `find` consult the man page: `man find`.

- ▶ Next is a `for` loop. For every value of `i`, we will run the command `rm -f $i` which will remove every file specified by the value of `i`.
- ▶ Lines that start with a `#` are comments but there are special cases such as the first line or the comments utilized by the `chkconfig` command.

Save the file as `log_erase.sh`. It is recommended to create a directory, such as `/scripts`, in order to keep scripts in a single location. To be able to execute the script, modify the rights of the file. Run the command `chmod 700 /scripts/log_eraser.sh`. This way, you will be the only one who can read, write, and execute the file. The seven in the `chmod` command comes from adding the numerical values of the read(4), write(2), and execute(1) permissions together: $4+2+1 = 7$. The two zeros in the `chmod` command indicate that the group and the world (all other users) have no rights to the file. This parallels the division of file permissions described in “File permissions” on page 375.

To run the script, type `/scripts/log_eraser.sh` if placed in the `/scripts` directory.

Attention: This script is intended primarily as an example that can be adapted to other situations. Although it works, you might want to consider a more sophisticated algorithm for the management of your log files, or use the built-in `logrotate` daemon.

9.2.4 Crontab

In everyday life, there are several scripts for maintaining your server. **Crontab** is a scheduler in Linux that automates the process of running these scripts. To list the scheduled programs in crontab, log in as root and type **crontab -l**. On a fresh Linux installation, no entries are there, so nothing can be seen or a message like no crontab for root is displayed.

Example 9-14 Crontab example

```
20 * * * * /scripts/script1
20 0 03 08 * /scripts/script2
0 0,6,12,18 * * * /scripts/script3
30 2 * * 6 /scripts/script4
0 0 * * 6 /scripts/log_eraser.sh
*/10 * * * * /scripts/script5
```

In the following is an explanation of the crontab syntax. The six fields are:

Minutes | Hour | Day of the month | Month | Day of the week | Path

The lines in the crontab example have the following meanings:

- ▶ At 20 minutes past each hour run /scripts/script1.
- ▶ At 20 minutes after midnight, on 03 august, run /scripts/script2.
- ▶ At 12 Am, 6 Am, 12 PM and 6 PM, on every day, run /scripts/script3.
- ▶ At 30 minutes after 2 AM, on every Saturday, run /scripts/script4.
- ▶ At midnight, on every Saturday run /scripts/log_eraser.sh.
- ▶ Every 10 minutes run /scripts/script5.

Tip: Be sure the date is set correctly on the server. A week starts on Sunday. Thus, to run a job on a Sunday, enter 0, not 7.

To create a schedule:

- ▶ Log in as root and at the command prompt type **crontab -e**.
- ▶ Press **i** to insert data.
- ▶ Enter a value for all six entries. Use ***** when an entry is not applicable.
- ▶ After you finish, press the Escape key and **:wq** (which means write and quit).

Tip: The crontab uses vi as the default text editor. The commands described here assume the use of vi. There is a graphical front end to cron called KCron.

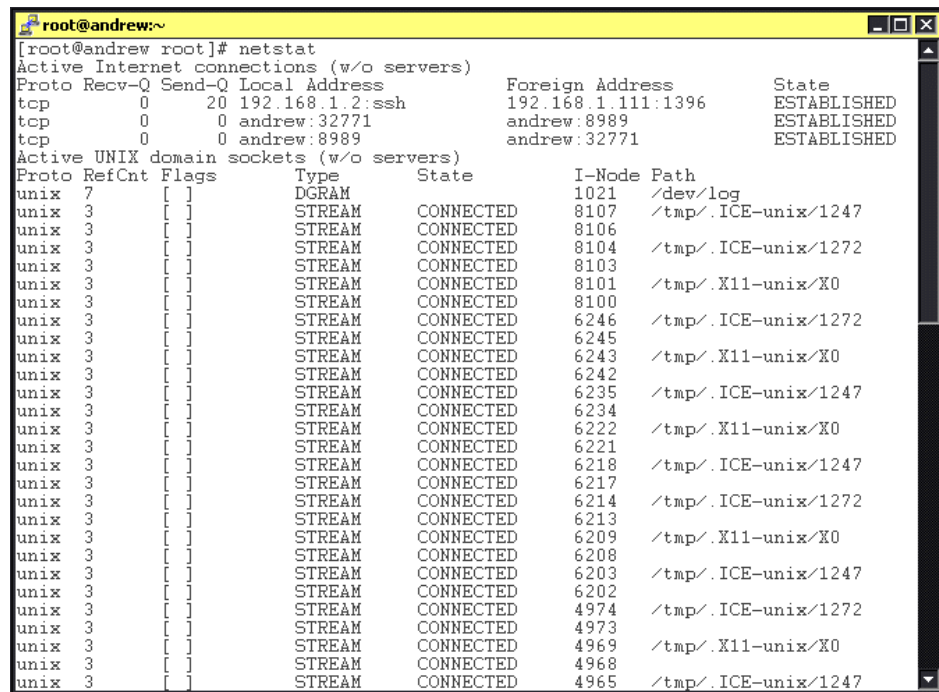
9.2.5 Network status

Sometimes it is very useful to know who is connected to a server and what is happening. In this section the functions of the `netstat` command and the `iptraf` utility are described.

Tip: The information in this section requires some TCP/IP protocol knowledge. Explaining all details in the screen captures is beyond the scope of this book, but enough information is provided to explain the common use of these network status tools. To learn more about TCP/IP, see the IBM Redbook *TCP/IP Tutorial and Technical Overview*, GG24-3376.

Netstat command

Log in as root and type `netstat` to see a screen similar to the one shown in Figure 9-8.



```
root@andrew root]# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 192.168.1.2:ssh        192.168.1.111:1396     ESTABLISHED
tcp      0      0 andrew:32771          andrew:8989            ESTABLISHED
tcp      0      0 andrew:8989          andrew:32771          ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags               Type           State         I-Node Path
unix   7      [ ]                DGRAM          CONNECTED    1021 /dev/log
unix   3      [ ]                STREAM         CONNECTED    8107 /tmp/.ICE-unix/1247
unix   3      [ ]                STREAM         CONNECTED    8106
unix   3      [ ]                STREAM         CONNECTED    8104 /tmp/.ICE-unix/1272
unix   3      [ ]                STREAM         CONNECTED    8103
unix   3      [ ]                STREAM         CONNECTED    8101 /tmp/.X11-unix/X0
unix   3      [ ]                STREAM         CONNECTED    8100
unix   3      [ ]                STREAM         CONNECTED    6246 /tmp/.ICE-unix/1272
unix   3      [ ]                STREAM         CONNECTED    6245
unix   3      [ ]                STREAM         CONNECTED    6243 /tmp/.X11-unix/X0
unix   3      [ ]                STREAM         CONNECTED    6242
unix   3      [ ]                STREAM         CONNECTED    6235 /tmp/.ICE-unix/1247
unix   3      [ ]                STREAM         CONNECTED    6234
unix   3      [ ]                STREAM         CONNECTED    6222 /tmp/.X11-unix/X0
unix   3      [ ]                STREAM         CONNECTED    6221
unix   3      [ ]                STREAM         CONNECTED    6218 /tmp/.ICE-unix/1247
unix   3      [ ]                STREAM         CONNECTED    6217
unix   3      [ ]                STREAM         CONNECTED    6214 /tmp/.ICE-unix/1272
unix   3      [ ]                STREAM         CONNECTED    6213
unix   3      [ ]                STREAM         CONNECTED    6209 /tmp/.X11-unix/X0
unix   3      [ ]                STREAM         CONNECTED    6208
unix   3      [ ]                STREAM         CONNECTED    6203 /tmp/.ICE-unix/1247
unix   3      [ ]                STREAM         CONNECTED    6202
unix   3      [ ]                STREAM         CONNECTED    4974 /tmp/.ICE-unix/1272
unix   3      [ ]                STREAM         CONNECTED    4973
unix   3      [ ]                STREAM         CONNECTED    4969 /tmp/.X11-unix/X0
unix   3      [ ]                STREAM         CONNECTED    4968
unix   3      [ ]                STREAM         CONNECTED    4965 /tmp/.ICE-unix/1247
```

Figure 9-8 netstat output

From left to right, the columns have the following meanings.

► **Proto**

The protocol used by the sockets (TCP, UDP, raw)

- ▶ **Recv-Q**
The count of bytes not copied by the user program connected to this socket
- ▶ **Send-Q**
The count of bytes not acknowledged by the remote host
- ▶ **Local Address**
Address and port number of the local end of the socket
- ▶ **Foreign Address**
Address and port number of the remote end of the socket
- ▶ **State**
The state of the socket. Since there are no states in raw mode and usually no states used in UDP, this column may be blank, but it can be one of several values:
 - ESTABLISHED
The socket has an established connection.
 - SYN_SENT
The socket is actively attempting to establish a connection.
 - SYN_RECV
A connection request has been received from the network.
 - FIN_WAIT1
The socket is closed and the connection is shutting down.
 - FIN_WAIT2
Connection is closed and the socket is waiting for a shutdown from the remote end.
 - TIME_WAIT
The socket is waiting after close to handle packets still in the network.
 - CLOSED
The socket is not being used.
 - CLOSE_WAIT
The remote end has shut down and is waiting for the socket to close.
 - LAST_ACK
The remote end has shut down and the socket is closed but still waiting for acknowledgement.
 - LISTEN
The socket is listening for incoming connections. Such sockets are not included in the output unless you specify the --listening (-l) or --all (-a) option.

- CLOSING
Both sockets are shut down, but we still do not have all our data sent.
- UNKNOWN
The state of the socket is unknown.

Netstat options

The netstat command can be run with options. Some of the options and their meanings are as follows:

- a Show both listening and non-listening sockets; illustrated in Figure 9-9.
- p Show the PID and name of the program to which each socket belongs; illustrated in Figure 9-10.
- s Display summary statistics for each protocol; illustrated in Figure 9-11.

```

root@andrew:~# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0  *:18208                *:*                     LISTEN
tcp      0      0  *:18191                *:*                     LISTEN
tcp      0      0  *:x11                  *:*                     LISTEN
tcp      0      0  *:10000                *:*                     LISTEN
tcp      0      0  *:18192                *:*                     LISTEN
tcp      0      0  *:ftp                  *:*                     LISTEN
tcp      0      0  *:ssh                  *:*                     LISTEN
tcp      0      0  andrew:8989            *:*                     LISTEN
tcp      0      20  192.168.1.2:ssh        192.168.1.111:1396     ESTABLISHED
tcp      0      0  andrew:32771           andrew:8989            ESTABLISHED
tcp      0      0  andrew:8989            andrew:32771           ESTABLISHED
udp      0      0  *:10000                *:*                     LISTEN
udp      0      0  *:990                  *:*                     LISTEN
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags   Type       State         I-Node Path
unix  2      [ ACC ] STREAM LISTENING   1205  /tmp/.font-unix/fs7100
unix  2      [ ACC ] STREAM LISTENING   4795  /tmp/ksocket-root/kdeinit-.0
unix  2      [ ACC ] STREAM LISTENING   4828  /tmp/ksocket-root/klauncherPfMg9b.sla
unix  7      [ ]     DGRAM      1021  /dev/log
unix  2      [ ACC ] STREAM LISTENING   4725  /tmp/.X11-unix/X0
unix  2      [ ACC ] STREAM LISTENING   1149  /dev/gpactl
unix  2      [ ACC ] STREAM LISTENING   4892  /tmp/ncop-root/andrew-04ed-3d267468
unix  2      [ ACC ] STREAM LISTENING   4802  /tmp/.ICE-unix/1247

```

Figure 9-9 netstat -a output

```

root@andrew:~# netstat -p
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp      0      20  192.168.1.2:ssh        192.168.1.111:1396     ESTABLISHED 1701/sshd
tcp      0      0  andrew:32771           andrew:8989            ESTABLISHED 910/cprid
tcp      0      0  andrew:8989            andrew:32771           ESTABLISHED 953/cpd
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags   Type       State         I-Node PID/Program name  Path
unix  7      [ ]     DGRAM      1021  714/syslogd      /dev/log
unix  3      [ ]     STREAM     CONNECTED   8107  1247/kdeinit: dco
47
unix  3      [ ]     STREAM     CONNECTED   8106  1465/kdeinit: konso
unix  3      [ ]     STREAM     CONNECTED   8104  1272/ksmserver    /tmp/.ICE-unix/12

```

Figure 9-10 netstat -p output

```
root@andrew:~  
[root@andrew root]# netstat -s  
Ip:  
 2776 total packets received  
 0 forwarded  
 0 incoming packets discarded  
2753 incoming packets delivered  
1397 requests sent out  
Icmp:  
15 ICMP messages received  
0 input ICMP message failed.  
ICMP input histogram:  
  destination unreachable: 11  
  echo replies: 4  
11 ICMP messages sent  
0 ICMP messages failed  
ICMP output histogram:  
  destination unreachable: 11  
Tcp:  
24 active connections openings  
0 passive connection openings  
0 failed connection attempts  
0 connection resets received  
3 connections established  
1172 segments received  
1322 segments send out  
0 segments retransmited  
0 bad segments received.  
11 resets sent  
Udp:  
25 packets received  
11 packets to unknown port received.  
0 packet receive errors  
60 packets sent  
TcpExt:  
ArpFilter: 0  
17 TCP sockets finished time wait in fast timer  
18 delayed acks sent  
1 delayed acks further delayed because of locked socket  
3 packets directly queued to recvmsg prequeue.  
3 packets directly received from prequeue  
264 packets header predicted  
TCPPureAcks: 439  
TCPHPAcks: 75  
TCPRecovery: 0  
TCPRecovery: 0  
TCPACKReneging: 0  
TCPACKReorder: 0  
TCPACKReorder: 0  
TCPReorder: 0  
TCPReorder: 0  
TCPFullUndo: 0  
TCPPartialUndo: 0  
TCPDACKUndo: 0
```

Figure 9-11 netstat statistic output

IPTraf utility

IPTraf is an IP network statistics utility. It is included in both the Red Hat and SuSE distributions.

Note: You must be logged in as root to run the IPTraf utility.

IPTraf is a console-based network statistics utility for Linux. It gathers a variety of figures, such as TCP connection packet and byte counts, interface statistics and

activity indicators, TCP/UDP traffic breakdowns, and LAN station packet and byte count.

Features

Among the features provided by IPTraf are the following:

- ▶ An IP traffic monitor that shows information on the IP traffic passing over your network. Includes TCP flag information, packet and byte counts, ICMP details, OSPF packet types.
- ▶ General and detailed interface statistics showing IP, TCP, UDP, ICMP, non-IP and other IP packet counts, IP checksum errors, interface activity, packet size counts.
- ▶ A TCP and UDP service monitor showing counts of incoming and outgoing packets for common TCP and UDP application ports.
- ▶ A LAN statistics module that discovers active hosts and displays statistics showing the data activity on them.
- ▶ TCP, UDP, and other protocol display filters, allowing you to view only traffic you are interested in.
- ▶ Logging
- ▶ Support for Ethernet, FDDI, ISDN, SLIP, PPP, and loopback interface types.
- ▶ Utilizes the built-in raw socket interface of the Linux kernel, allowing it to be used over a wide range of supported network cards.
- ▶ Full-screen, menu-driven operation

Protocols recognized

- ▶ IP
- ▶ TCP
- ▶ UDP
- ▶ ICMP
- ▶ IGMP
- ▶ IGP
- ▶ IGRP
- ▶ OSPF
- ▶ ARP
- ▶ RARP

Non-IP packets will simply be indicated as “Non-IP” and, on Ethernet networks, will be supplied with the appropriate Ethernet addresses.

Supported Interfaces

- ▶ Local loopback
- ▶ All Linux-supported Ethernet interfaces

- ▶ All Linux-supported FDDI interfaces
- ▶ SLIP
- ▶ Asynchronous PPP
- ▶ Synchronous PPP over ISDN
- ▶ ISDN with Raw IP encapsulation
- ▶ ISDN with Cisco HDLC encapsulation
- ▶ Parallel Line IP

The information generated by IPTraf can be valuable in making network organization decisions, troubleshooting LANs, and tracking activity of various IP hosts.

Once installed and running on the system, the IPTraf utility will look like Figure 9-12.

```

root@anet: /root
IPTraf
-----
Source          Destination          Packets  Bytes  Flags  Iface
-----
62.231.66.55:ssh 192.168.1.111:1416  >      524    131508 -PA-   eth0
192.168.1.111:1416 62.231.66.55:ssh  >      262    10480  --A-   eth0
24.65.31.164:6699 192.168.1.3:1192  >      501    704072 -PA-   eth2
192.168.1.3:1192 24.65.31.164:6699 >        0        0  ----   eth2
24.65.31.164:6699 192.168.1.3:1192  >      501    704072 -PA-   eth0
192.168.1.3:1192 24.65.31.164:6699 >      255    10248  --A-   eth0
62.231.66.55:1156 61.193.98.91:6699 >        2        128  -PA-   eth2
61.193.98.91:6699 62.231.66.55:1156 >        0        0  ----   eth2
192.168.1.3:1156 61.193.98.91:6699 >        2        128  -PA-   eth0
61.193.98.91:6699 192.168.1.3:1156 >        2        80  --A-   eth0
192.168.1.3:1498 62.137.111.23:6699 >       45    25920  --A-   eth0
62.137.111.23:6699 192.168.1.3:1498 >       28    1120  --A-   eth0
62.231.66.55:1498 62.137.111.23:6699 >       45    25920  --A-   eth2
62.137.111.23:6699 62.231.66.55:1498 >        0        0  ----   eth2
192.168.1.5:1914 216.55.95.34:http >       18    1191  CLOSED eth0
216.55.95.34:http 192.168.1.5:1914  >       26    35175  CLOSED eth0
62.231.66.55:1315 64.12.27.145:5190 >        1        46  -PA-   eth2
64.12.27.145:5190 62.231.66.55:1315 >        0        0  ----   eth2
216.55.95.34:http 192.168.1.5:1914  >       25    35135  --A-   eth2
192.168.1.5:1914 216.55.95.34:http >        0        0  ----   eth2
192.168.1.5:1915 216.55.95.34:http >        5        626  CLOSED eth0
216.55.95.34:http 192.168.1.5:1915 >        4        306  CLOSED eth0
62.231.66.55:1915 216.55.95.34:http >        5        626  --A-   eth2
-----
TCP: 16 entries ----- Active -----

ARP (42 bytes) from 0000c09929c7 to 0050bf345d11 on eth0
ARP (60 bytes) from 0050bf345d11 to 0000c09929c7 on eth0
ARP (42 bytes) from 0000c09929c7 to 00a00c1269ea on eth0
ARP (60 bytes) from 00a00c1269ea to 0000c09929c7 on eth0

Top ----- Elapsed time: 0:00 -----
IP: 2055212 TCP: 2055212 UDP: 0 ICMP: 0 Non-IP: 204
Up/Dn/PgUp/PgDn-scr1 actv win W-chg actv win M-more TCP info X/Ctrl+X-Exit

```

Figure 9-12 IPTraf utility

9.2.6 System logs

The Linux log system is both flexible and powerful, and in many situations, the log information will be very useful.

Logs can be generated by the system or by applications. Linux keeps logs in /var/log unless the administrator changes the path. The program (daemon) responsible for generating the logs is **syslogd**; log entries are caused by events.

Almost every application can send information (events) to the syslogd. The syslogd daemon can be set to start at system boot or not, but we recommend you set syslogd to start when the system boots (this is the default), as shown in Figure 9-13.

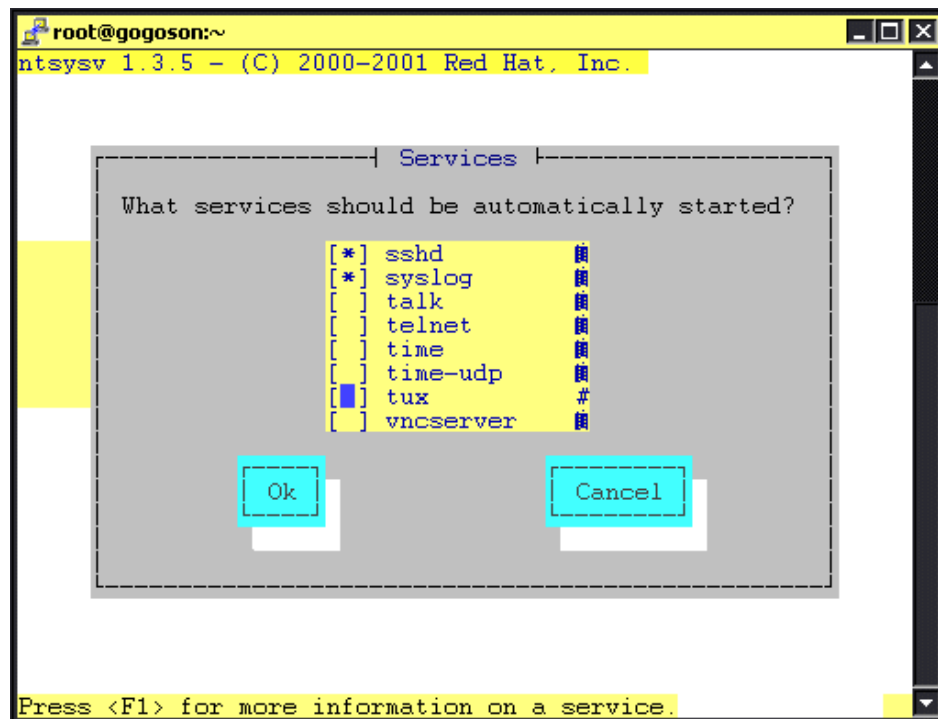
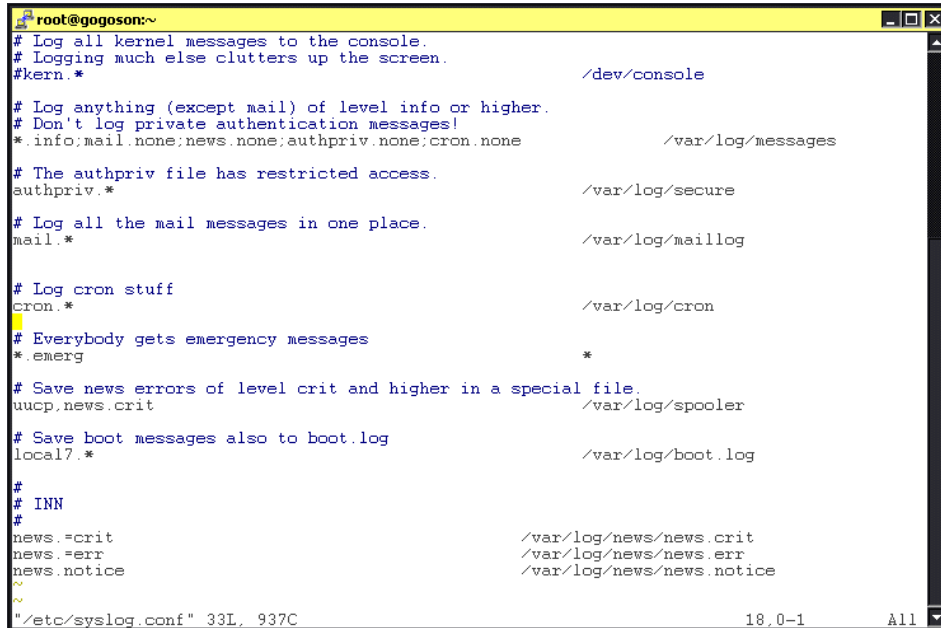


Figure 9-13 Red Hat system services

Figure 9-14 shows the syslogd configuration file /etc/syslog.conf.



```
root@gogoson:~
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;news.none;authpriv.none;cron.none    /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                             /var/log/secure

# Log all the mail messages in one place.
mail.*                                                 /var/log/maillog

# Log cron stuff
cron.*                                                 /var/log/cron

# Everybody gets emergency messages
*.emerg                                               *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                       /var/log/spooler

# Save boot messages also to boot.log
local7.*                                             /var/log/boot.log

#
# INN
#
news.=crit                                           /var/log/news/news.crit
news.=err                                           /var/log/news/news.err
news.notice                                         /var/log/news/news.notice
~
~
"/etc/syslog.conf" 33L, 937C                               18,0-1    All
```

Figure 9-14 Syslog configuration file

By default, all system messages go in the `/var/log/messages` file unless otherwise specified. In the syslog configuration file, there are specifications for other log files for mail, news, and so forth.

The log files can be redirected to other paths by editing the `syslog.conf` or by moving the file and creating a link to the new location.

There are situations when the system administrator wants to see the log information in real time. To do so, log in as root and type at the shell command prompt `tail -f /var/log/messages`. The `tail` command will watch the log file and any information that is written to the log file is displayed in the console window as show in Figure 9-15.

Note: For more information about the `tail` command, type `man tail`.

```

[root@gogoson root]# tail -f /var/log/maillog
Jul  8 09:36:05 gogoson sendmail[1806]: g686a5M01806: from=root, size=227, class=0, nrcpt=
=1, msgid=<200207080636.g686a5M01806@localhost.localdomain>, relay=root@localhost
Jul  8 09:36:06 gogoson sendmail[1806]: g686a5M01806: to=root, ctladdr=root (0/0), delay=0
0:00:01, xdelay=00:00:01, mailer=local, pri=30227, dsn=2.0.0, stat=Sent

[root@gogoson root]# tail -f /var/log/messages
Jul  8 09:31:44 gogoson syslogd 1.4.1: restart.
Jul  8 12:31:23 gogoson sshd(pam_unix)[1977]: session opened for user root by (uid=0)
Jul  8 13:50:56 gogoson ftpd[2109]: wu-ftp - TLS settings: control allow, client_cert all
ow, data allow
Jul  8 13:50:58 gogoson ftp(pam_unix)[2109]: session opened for user root by (uid=0)
Jul  8 13:50:58 gogoson ftpd: 192.168.1.111: root[2109]: FTP LOGIN FROM 192.168.1.111 [192
.168.1.111] root
Jul  8 13:51:36 gogoson ftp(pam_unix)[2109]: session closed for user root
Jul  8 13:51:36 gogoson ftpd: 192.168.1.111: root: QUIT[2109]: FTP session closed

```

Figure 9-15 `tail -f /var/log/messages`

Tip: The syslog daemon can be configured to send the log information to a log server. If you have many Linux servers, you may choose to configure one server to be a log server. In this way all other servers are logging the information to a single target, the log server. For more information about log server see the syslogd man pages.

9.2.7 Remote administration

Linux servers can be administered remotely and there are many software programs available for this. Several of the most commonly used are described here.

Webmin

Webmin is a powerful tool for remotely administering a Linux server. It can be downloaded for free from:

<http://www.webmin.com>

It can be downloaded either as an .rpm package or as a source file. Download the .rpm file, log in as root, and use the Package Manager or rpm command from a shell to install the Webmin software. You can also use rpm from the shell command line.

Once installed, connect from a Web browser to the server:

`http://<server IP address>:10000`

You will be prompted for your username (root) and a password (root's password). After login, the Web browser should look like the example shown in Figure 9-17.

Through the Webmin software, the system administrator can configure the server and its applications from virtually anywhere. The Webmin server configuration page is easy to use and has numerous capabilities, as shown in Figure 9-17.

Attention: We recommend that you use the Webmin software only from an internal network if you do not use SSL authentication.

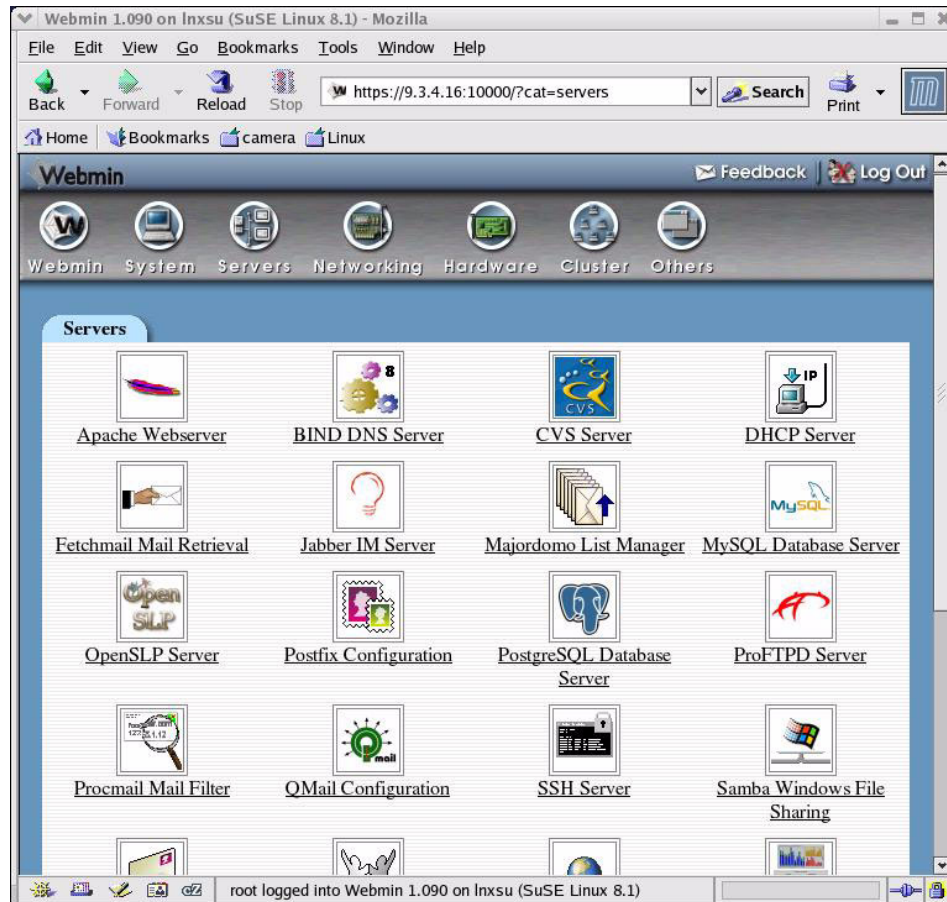


Figure 9-16 Webmin server configuration page

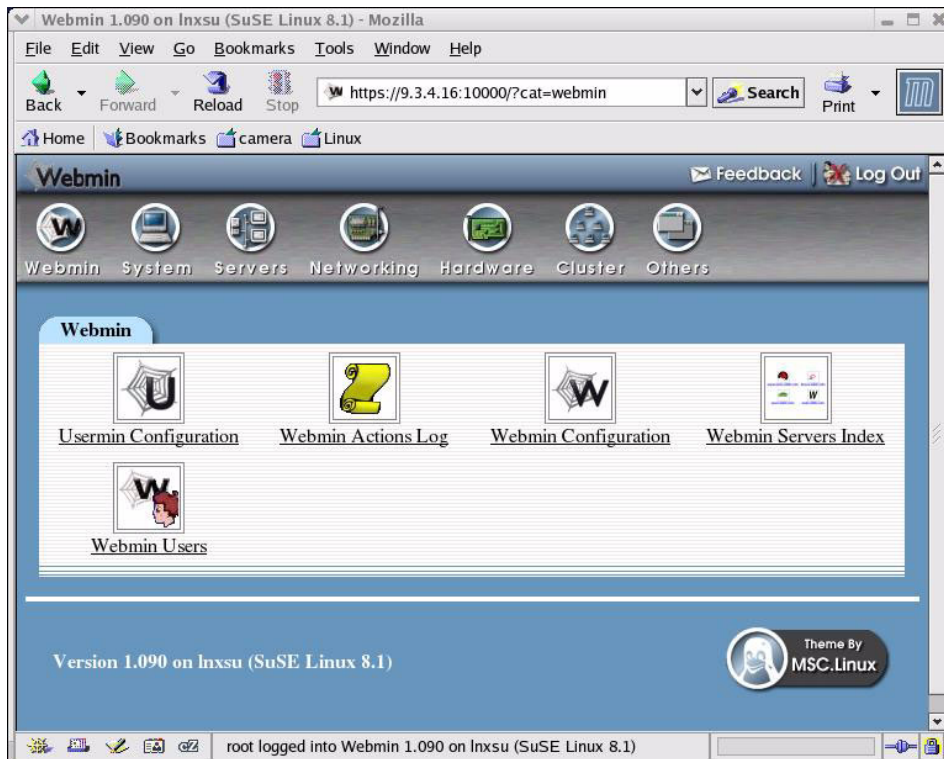


Figure 9-17 Webmin interface

VNC

VNC is another program for remote administration of Linux servers. You can download the VNC tool as well as obtain more information about it at:

<http://www.uk.research.att.com/vnc/index.html>

To install VNC on the Linux machine, download the Linux version, unpack the files (tar xvfz vnc-XX.YY.tar.gz, where XX and YY are version and release numbers) and copy the files to /usr/bin.

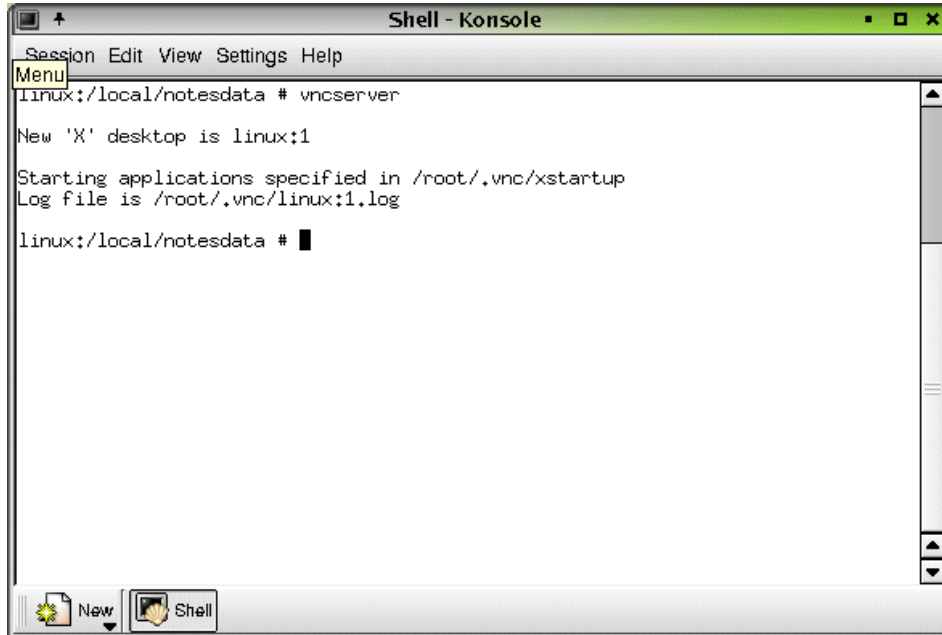


Figure 9-18 Starting VNC Server on Linux

To start the server, run **vncserver** from a shell. This will prompt for a password to be used when connecting from another machine. The machine name and the windows number will be displayed (see Figure 9-18).

To connect to the VNC server, run the VNC viewer on your client and enter the hostname:window (see Figure 9-19) and then click **OK**. Enter the password when prompted.

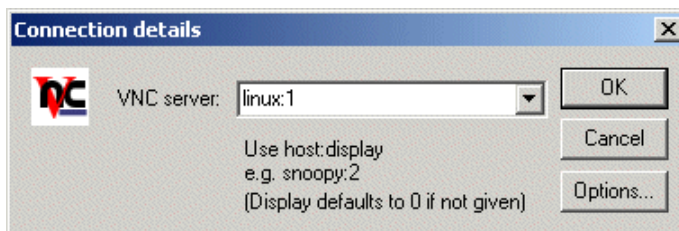


Figure 9-19 VNC viewer

HOBLink X11 for OS/2

Administrators often need to have an X server implementation on their system to access terminal sessions and graphical applications on Linux systems. To access these applications from an OS/2 client, the HOBLink X11 product from

HOB GmbH & Co. KG should be considered. HOBLink X11 for OS/2 is an integrated PC X Server package that allows you to use your PC running OS/2 as an X Window terminal.

For customers entitled to technical support, IBM plans to work with HOB to resolve customer reported problems with OS/2.

For more information, please refer to the HOBLink X11 for OS/2 Web site at: http://www.hob.de/www_us/produkte/connect/X11-OS2.htm

9.3 Summary

For an OS/2 administrator who is not very familiar with Linux operating environments, the prospect of having to administer a Linux server can be somewhat intimidating. This chapter has touched on just a few aspects of Linux administration. It is not intended to be a complete survey of the topic. There are many books and other resources dedicated to the topic. However, the intention here was to introduce some of the administrative commands and facilities to provide the OS/2 administrator with a comfortable feeling that such facilities are not only available but plentiful.

Appendixes

Several scripts were used and described in this book to help facilitate a migration from OS/2. In Appendix A, “Windows 2000 migration related scripts” on page 411, the scripts we used to ease the migration to a Windows 2000 environment are provided. The scripts are also available through FTP from the IBM Redbooks Web site.

Appendix B, “REXX source code” on page 477, contains the source code and related files for the LSMT tool used in Chapter 3, “Starting the OS/2 Server migration” on page 63 to extract configuration information from an OS/2 Server environment.



Windows 2000 migration related scripts

This appendix contains all scripts, protocol files, and response files used in 2.1, “Windows 2000 as a target platform” on page 20, and Chapter 4, “Migrating OS/2 Servers to Windows 2000” on page 87.

CID installation of Windows 2000

For the installation of Windows 2000 systems, we defined a minimalist version of an unattended installation based on the OS/2 CID process. To keep it simple, we only defined the necessary directories, shares and files and completely omitted logging and error checking. Figure A-1 shows the core directory structure including the share points. In all our examples we use the server XFER1 for CID installation.

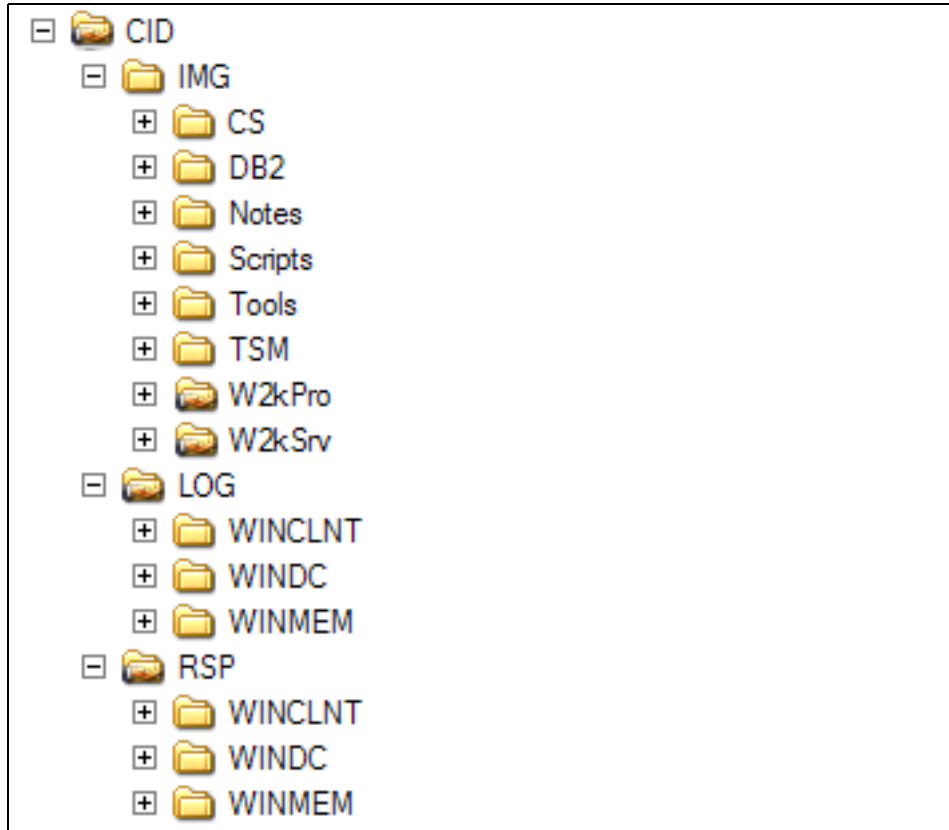


Figure A-1 Core tree of CID structure for Windows 2000 unattended installation

The following shows the shares necessary to follow our installation examples:

Table A-1 Share points for CID installation

Share name	Directory (purpose)
CID	\CID (Root for CID installation images)

Share name	Directory (purpose)
LOG	\CID\LOG (Base directory containing a directory for log files for each client)
RSP	\CID\RSP (Base directory containing a directory for response files for each client)
W2kPro	\CID\IMG\W2kPro (Needed for DOS clients to start unattended installation of Windows 2000 Professional.)
W2kSrv	\CID\IMG\W2kSrv (Needed for DOS clients to start unattended installation of Windows 2000 Server.)

Windows installation related scripts

In this section are all scripts we discussed in 2.1, “Windows 2000 as a target platform” on page 20. Additionally, we have included a few simple command files that ease some steps during installation.

SERVICE.CMD

Post installation routine of maintenance system:

```
@ECHO OFF
REM *****
REM File      : SERVICE.CMD
REM Version   : 2.0
REM Date      : 06/06/03
REM Author    : Leif Braeuer (6PAC Consulting AG)
REM
REM Description:
REM Starts installation of services for Maintenance partition
REM
REM *****

SET SCRIPTS=\\xfer1\cid\img\scripts
SET TOOLS=\\xfer1\cid\img\tools
SET IMG=\\xfer1\cid\img
SET RSP=\\xfer1\rsp\%computername%

%tools%\diskpart.exe /s %rsp%\part.txt
label c: SERVICE
format d: /v:SYSTEM /FS:NTFS /Q < %scripts%\y.txt
format e: /v:DATA /FS:NTFS /Q < %scripts%\y.txt
regedit.exe /s %rsp%\w2k_inst.REG
%tools%\shutdown /l /c /r /T:60
```

W2K.CMD

Start of the Windows 2000 installation within the service partition:

```
@ECHO OFF
REM *****
REM File      : W2K.CMD
REM Version   : 2.0
REM Date      : 06/06/03
REM Author    : Leif Braeuer (6PAC Consulting AG)
REM
REM Description:
REM Starts installation of production environment
REM
REM *****

SETLOCAL
SET SCRIPTS=\\xfer1\cid\img\scripts
SET TOOLS=\\xfer1\cid\img\tools
SET IMG=\\xfer1\cid\img
SET RSP=\\xfer1\rsp%\%computername%
SET SOURCE=\\xfer1\w2ksrv\i386

%SOURCE%\WINNT32.EXE /s:%SOURCE%\ /tempdrive:d:\
/unattend5:%rsp%\%computername%p.txt

ENDLOCAL
```

POST1.CMD

First post installation routine of production system:

```
@ECHO OFF
REM *****
REM File      : POST1.CMD
REM Version   : 2.0
REM Date      : 06/06/03
REM Author    : Leif Braeuer (6PAC Consulting AG)
REM
REM Description:
REM Starts post installation processes phase 1
REM
REM *****

SETLOCAL
SET SCRIPTS=\\xfer1\cid\img\scripts

CALL %Scripts%\sysocmgr.cmd INSTDHCP.TXT
CALL %Scripts%\sysocmgr.cmd INSTDNS.TXT
```



```
CALL %Scripts%\sysocmgr.cmd INSTWINS.TXT
CALL %Scripts%\sysocmgr.cmd INSTFTP.TXT
CALL %Scripts%\promo.cmd
```

POST2.CMD

Second post installation routine of production system:

```
@ECHO OFF
REM *****
REM File      : POST2.CMD
REM Version  : 2.0
REM Date     : 06/06/03
REM Author  : Leif Braeuer (6PAC Consulting AG)
REM
REM Description:
REM Starts post installation processes phase 2
REM
REM *****

SET SCRIPTS=\\xfer1\cid\img\scripts
SET TOOLS=\\xfer1\cid\img\tools
SET IMG=\\xfer1\cid\img
SET RSP=\\xfer1\rsp\%computername%

CALL %Scripts%\sysocmgr.cmd INSTCERTSRV.TXT

msiexec /i "%img%\tsm\Tivoli Storage Manager Client.msi" RebootYesNo="No"
REBOOT="Suppress" ALLUSERS=1 INSTALLDIR="%PROGRAMFILES%\Tivoli\TSM"
ADDLOCAL="BackupArchiveGUI,ApiRuntime,BackupArchiveGuiDeu,AdministrativeCmd"
TRANSFORMS=1033.mst /qn /! *v "%SYSTEMDRIVE%\tsm.log"
COPY %rsp%\dsm.opt "%ProgramFiles%\Tivoli\TSM\baclient"

%img%\notes\501\setup /s /f1%rsp%\notes.iss /f2%SYSTEMDRIVE%\Notes.log

NET GROUP CSAdmins /add /comment:"Administrators for IBM Communications Server"
NET GROUP CSAdmins Administrator /add
NET USE X: %IMG% /persistent:no
NET USE R: %RSP% /persistent:no

x:\cs\setup /s /f1r:\cs.iss /f2%SYSTEMDRIVE%\cs.log
```

SYSOCRMGR.CMD

Script to start unattended installation of additional components:

```
@ECHO OFF
REM *****
```

```

REM File      : SYSOCMGR.CMD
REM Version   : 2.0
REM Date      : 06/06/03
REM Author    : Leif Braeuer (6PAC Consulting AG)
REM
REM Description:
REM Starts installation of additional windows components
REM
REM *****

```

```

SETLOCAL
SET SCRIPTS=\\xfer1\cid\img\scripts
SET TOOLS=\\xfer1\cid\img\tools
SET IMG=\\xfer1\cid\img
SET RSP=\\xfer1\rsp\%computername%

sysocmgr /i:%WINDIR%\INF\SYSOC.INF /u:%rsp%\%1

ENDLOCAL

```

DCPROMO.CMD

Script to start unattended promotion of Domain Controllers:

```

@ECHO OFF
REM *****
REM File      : PROMO.CMD
REM Version   : 2.0
REM Date      : 06/06/03
REM Author    : Leif Braeuer (6PAC Consulting AG)
REM
REM Description:
REM Promotes server to domain controller
REM
REM *****

```

```

SETLOCAL
SET SCRIPTS=\\xfer1\cid\img\scripts
SET TOOLS=\\xfer1\cid\img\tools
SET IMG=\\xfer1\cid\img
SET RSP=\\xfer1\rsp\%computername%

:DCPROMO
  DEL %WINDIR%\DEBUG\DCPROMO.LOG 1>NUL 2>NUL
  DCPROMO /answer:%RSP%\dcpromo.txt

REM
REM The following pause prevents the execution of the next step of installation
before

```

```

REM DCPROMO.EXE completed with an automatic reboot.
REM
:WAIT
ECHO Waiting for DCPromo to complete...
PAUSE
GOTO WAIT

ENDLOCAL

```

Windows installation related response files

In this section you will find all the response files we discuss in 2.1, “Windows 2000 as a target platform” on page 20 to describe the initial installation and setup of the target domain.

WINDC.TXT

Response file for unattended installation of service system. The member server uses the same response files:

```

[Data]
    AutoPartition=1
    MsDosInitiated="0"
    UnattendedInstall="Yes"

[Unattended]
    UnattendMode=FullUnattended
    FileSystem=ConvertNTFS
    OemSkipEula=Yes
    OemPreinstall=Yes
    DriverSigningPolicy=Ignore
    TargetPath=\WINNT

[GuiUnattended]
    AdminPassword=password
    AutoLogon=Yes
    AutoLogonCount=10
    OEMSkipRegional=1
    TimeZone=020
    OemSkipWelcome=1

[UserData]
    FullName=Myname
    OrgName=MyCompany
    ComputerName=WINDC
    ProductId = "?????-?????-?????-?????-?????"

```

```

[Display]
  BitsPerPel=16
  Xresolution=1024
  YResolution=768
  Vrefresh=75

[LicenseFilePrintData]
  AutoMode=PerSeat

[TapiLocation]
  CountryCode=1
  AreaCode=512

[RegionalSettings]
  LanguageGroup=1
  Language=00000409

[GuiRunOnce]
  Command0="\\XFER1\CID\rsp\%computername%\service.cmd"

[Identification]
  JoinWorkgroup=SERVICE

[Networking]
  InstallDefaultComponents=No

[NetAdapters]
  Adapter1=params.Adapter1

[params.Adapter1]
  INFID="PCI\VEN_8086&dev_1229"
  ConnectionName="Ethernet TCPIP"

[NetClients]
  MS_MSCClient=params.MS_MSCClient

[NetServices]
  MS_SERVER=params.MS_SERVER

[NetProtocols]
  MS_TCPIP=params.MS_TCPIP

[params.MS_TCPIP]
  DNS=No
  UseDomainNameDevolution=No
  EnableLMHosts=Yes
  AdapterSections=params.MS_TCPIP.Adapter1

[params.MS_TCPIP.Adapter1]

```

```
SpecificTo=Adapter1
DHCP=No
IPAddress=9.3.4.12
SubnetMask=255.255.254.0
DNSServerSearchOrder=9.3.4.12,9.3.4.2
DefaultGateway=9.3.4.41
WINS=Yes
NetBIOSOptions=1
```

```
[Components]
```

```
accessopt = off
cdplayer = off
cluster = off
charmap = off
deskpaper = off
dialer = off
fp = off
freecell = off
hypertrm = off
iis_common = off
iisdbg = off
iis_doc = off
iis_ftp = off
iis_htmla = off
iis_inetmgr = off
iis_nntp = off
iis_nntp_docs = off
iis_smtp = off
iis_smtp_docs = off
iis_www = off
indexsrv_system = off
media_blindnoisy = off
media_blindquiet = off
media_clips = off
minesweeper = off
mousepoint = off
mplay = off
netcis = off
netcm = off
netcps = off
pinball = off
rec = off
solitaire = off
templates = off
Tenable = on
vol = off
```

```
[TerminalServices]
```

```
ApplicationServer = 0
```

```
[Branding]
BrandIEUsingUnattended = Yes
```

```
[URL]
Home_Page = http://w3.somedomain.local/
```

PART.TXT

Parameter files for DISKPART. This response file contains the commands for the DISKPART utility to create partitions:

```
select disk 0
create partition extended
create partition logical size=2000
create partition logical
select volume 0
remove letter=D
assign letter=F
select volume 2
assign letter=D
select volume 3
assign letter=E
```

W2K_INST.REG

Start installation of Windows 2000 production system after reboot:

```
REGEDIT4
```

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunONCE]
"W2K"="\\\\xfer1\cid\img\scripts\w2k.cmd"
```

WINDCP.TXT

Response file for unattended installation of Domain Controller:

```
[Data]
  AutoPartition=1
  MsDosInitiated="0"
  UnattendedInstall="Yes"

[Unattended]
  UnattendMode=FullUnattended
  FileSystem=ConvertNTFS
  OemSkipEula=Yes
  OemPreinstall=Yes
```

```
DriverSigningPolicy=Ignore
TargetPath=\WINNT

[GuiUnattended]
AdminPassword=password
AutoLogon=Yes
AutoLogonCount=99
OEMSkipRegional=1
TimeZone=020
OemSkipWelcome=1

[UserData]
FullName=Myname
OrgName=MyCompany
ComputerName=WINDC
ProductId = "?????-?????-?????-?????-?????"

[Display]
BitsPerPel=16
Xresolution=1024
YResolution=768
Vrefresh=60

[LicenseFilePrintData]
AutoMode=PerSeat

[TapiLocation]
CountryCode=1
AreaCode=512

[RegionalSettings]
LanguageGroup=1
Language=00000409

[GuiRunOnce]
Command0="\\XFER1\w2ksrv\i386\winnt32 /cmdcons /unattend"
Command1="\\XFER1\rsp\%computername%\post1.cmd"

[Identification]
JoinWorkgroup=PROD

[Networking]
InstallDefaultComponents=No

[NetAdapters]
Adapter1=params.Adapter1

[params.Adapter1]
```

```
INFID="PCI\VEN_8086&dev_1229"  
ConnectionName="Ethernet TCPIP"
```

```
[NetClients]  
MS_MSCClient=params.MS_MSCClient  
  
[NetServices]  
MS_SERVER=params.MS_SERVER  
  
[NetProtocols]  
MS_TCPIP=params.MS_TCPIP  
  
[params.MS_TCPIP]  
DNS=No  
UseDomainNameDevolution=No  
EnableLMHosts=Yes  
AdapterSections=params.MS_TCPIP.Adapter1  
  
[params.MS_TCPIP.Adapter1]  
SpecificTo=Adapter1  
DHCP=No  
IPAddress=9.3.4.12  
SubnetMask=255.255.254.0  
DNSServerSearchOrder=9.3.4.12,9.3.4.2  
DefaultGateway=9.3.4.41  
WINS=Yes  
WinsServerList=9.3.4.12  
NetBIOSOptions=1
```

```
[Components]  
accessopt = off  
cdplayer = off  
cluster = off  
charmap = off  
deskpaper = off  
dialer = off  
fp = off  
freecell = off  
hypertrm = off  
iis_common = off  
iisdbg = off  
iis_doc = off  
iis_ftp = off  
iis_htmla = off  
iis_inetmgr = off  
iis_nntp = off  
iis_nntp_docs = off  
iis_smtp = off  
iis_smtp_docs = off
```



```
iis_www = off
indexsrv_system = off
media_blindnoisy = off
media_blindquiet = off
media_clips = off
minesweeper = off
mousepoint = off
mplay = off
netcis = off
netcm = off
netcps = off
pinball = off
rec = off
solitaire = off
templates = off
Tenable = on
vol = off

[TerminalServices]
ApplicationServer = 0

[Branding]
BrandIEUsingUnattended = Yes

[URL]
Home_Page = http://w3.somedomain.local/
```

WINMEMP.TXT

Response file for unattended installation of member servers:

```
[Data]
  AutoPartition=1
  MsDosInitiated="0"
  UnattendedInstall="Yes"

[Unattended]
  UnattendMode=FullUnattended
  FileSystem=ConvertNTFS
  OemSkipEula=Yes
  OemPreinstall=Yes
  DriverSigningPolicy=Ignore
  TargetPath=\WINNT

[GuiUnattended]
  AdminPassword=password
  AutoLogon=Yes
  AutoLogonCount=99
  OEMSkipRegional=1
```

```

    TimeZone=020
    OemSkipWelcome=1

[UserData]
    FullName=Myname
    OrgName=MyCompany
    ComputerName=WINMEM
    ProductId = "?????-?????-?????-?????-?????"

[Display]
    BitsPerPel=16
    XResolution=1024
    YResolution=768
    Vrefresh=60

[LicenseFilePrintData]
    AutoMode=PerSeat

[TapiLocation]
    CountryCode=1
    AreaCode=512

[RegionalSettings]
    LanguageGroup=1
    Language=00000409

[GuiRunOnce]
    Command0="\\XFER1\w2ksrv\i386\winnt32 /cmdcons /unattend"
    Command1="\\XFER1\rsp\%computername%\post1.cmd"

[Identification]
    JoinDomain=SOMEDOMAIN2
    DomainAdmin = Administrator
    DomainAdminPassword = password

[Networking]
    InstallDefaultComponents=No

[NetAdapters]
    Adapter1=params.Adapter1

[params.Adapter1]
    INFID="PCI\VEN_8086&dev_1229"
    ConnectionName="Ethernet TCP/IP"

[NetClients]
    MS_MSCClient=params.MS_MSCClient

```

```

[NetServices]
    MS_SERVER=params.MS_SERVER

[NetProtocols]
    MS_TCPIP=params.MS_TCPIP

[params.MS_TCPIP]
    DNS=No
    UseDomainNameDevolution=No
    EnableLMHosts=Yes
    AdapterSections=params.MS_TCPIP.Adapter1

[params.MS_TCPIP.Adapter1]
    SpecificTo=Adapter1
    DHCP=No
    IPAddress=9.3.4.14
    SubnetMask=255.255.254.0
    DNSServerSearchOrder=9.3.4.12,9.3.4.2
    DefaultGateway=9.3.4.41
    WINS=Yes
    WinsServerList=9.3.4.12
    NetBIOSOptions=1

[Components]
accessopt = off
cdplayer = off
cluster = off
charmap = off
deskpaper = off
dialer = off
fp = off
freecell = off
hypertrm = off
iis_common = off
iisdbg = off
iis_doc = off
iis_ftp = off
iis_htmla = off
iis_inetmgr = off
iis_nntp = off
iis_nntp_docs = off
iis_smtp = off
iis_smtp_docs = off
iis_www = off
indexsrv_system = off
media_blindnoisy = off
media_blindquiet = off
media_clips = off
minesweeper = off

```

```
mousepoint = off
mplay = off
netcis = off
netcm = off
netcps = off
pinball = off
rec = off
solitaire = off
templates = off
Tsenable = on
vol = off

[TerminalServices]
ApplicationServer = 0

[Branding]
BrandIEUsingUnattended = Yes

[URL]
Home_Page = http://w3.ibm.com/
```

INSTDHCP.TXT

Installation of DHCP server:

```
[NetOptionalComponents]
DHCPServer = 1
```

INSTWINS.TXT

Installation of WINS server:

```
[NetOptionalComponents]
WINS = 1
```

INSTDNS.TXT

Installation of DNS server:

```
[NetOptionalComponents]
DNS = 1
```

INSTFTP.TXT

Installation of FTP server:

```
[Components]
iis_common = on
```

```
iis_ftp = on
iis_inetmgr = on

[InternetServer]
pathFTPRoot = "E:\FTP"
```

INSTWWW.TXT

Installation of Internet Information Server:

```
[Components]
iis_common = on
iis_inetmgr = on
iis_www = on

[InternetServer]
pathWWWRoot = "E:\WWW"
```

DCPROMO1.TXT

Active Directory promotion of first domain controller.

```
[DCINSTALL]
ReplicaOrNewDomain=Domain
TreeOrChild=Tree
CreateOrJoin=Create
NewDomainDNSName=somedomain.local
DNSOnNetwork=yes
DomainNetbiosName=SOMEDOMAIN
AutoConfigDNS=yes
SiteName=CENTRAL
AllowAnonymousAccess=yes
DatabasePath=e:\ntds
LogPath=e:\ntds
SYSVOLPath=e:\sysvol
;
; *****
; Password entry will be deleted after executing DCPROMO
; *****
SafeModeAdminPassword=password
;
CriticalReplicationOnly=No
RebootOnSuccess=Yes
```

DCPROMO2.TXT

Active Directory promotion for additional DC. This file is actually not used in our sample, but shows you how to add additional domain controllers:

```
[DCINSTALL]
;
; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
; All Password entries will be deleted after executing DCPROMO
; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
UserName=Administrator
Password=password
UserDomain=SOMEDOMAIN
;
DatabasePath=E:\NTDS
LogPath=E:\NTDS
SYSVOLPath=E:\SYSVOL
;
SafeModeAdminPassword=password
;
CriticalReplicationOnly=no
ReplicaOrNewDomain=Replica
ReplicaDomainDNSName=somedomain.local
RebootOnSuccess=yes
SiteName=CENTRAL
```

INSTCERTSRV.TXT

Installation of certificate services:

```
[Components]
certsrv = on
certsrv_client = on
certsrv_server = on

[Certsrv_client]
CAmachine = windc.somedomain.local
CAName = WINDC

[Certsrv_server]
CAType = EnterpriseRoot
Country = US
CSPPProvider = "Microsoft Base Cryptographic Provider v1.0"
Description = "Certificate server for Somedomain"
HashAlgorithm = "SHA1"
Locality = "Austin"
Name = WINDC
Organization = Some Company
OrganizationUnit = IT
```

PreserveDB = No
SharedFolder=E:\CAConfig
State = Tx
UseExistingCert = No
ValidityPeriod = 2
ValidityPeriodUnits = Years

CS.ISS

IBM Communication Server installation:

```
[InstallShield Silent]
Version=v5.00.000
File=Response File
[File Transfer]
OverwriteReadOnly=NoToAll
[DlgOrder]
Dlg0=SdWelcome-0
Count=10
Dlg1=SdAskDestPath-0
Dlg2=SdComponentDialog2-0
Dlg3=SdSelectFolder-0
Dlg4=AskText-0
Dlg5=AskText-1
Dlg6=SdStartCopy-0
Dlg7=AskYesNo-0
Dlg8=AskYesNo-1
Dlg9=RebootDialog-0
[SdWelcome-0]
Result=1
[SdAskDestPath-0]
szDir=e:\IBMCS
Result=1
[SdComponentDialog2-0]
Component-type=string
Component-count=19
Component-0=Base Component
Component-1=Documentation
Component-2=AS400 OLE DB Provider
Component-3=Java Access
Component-4=Client Images
Component-5=SDK
Component-6=Base System
Component-7=AS400 System
Component-8=AS400 MRI
Component-9=Base Residual
Component-10=AS400 SelfReg
Component-11=AS400 SelfReg System
Component-12=NT4 LLC2
```

Component-13=WIN2000 LLC2
Component-14=Base NT4
Component-15=Base WIN2000
Component-16=WIN2000 Windir
Component-17=Windir
Component-18=BaseHolder2
Result=1
[SdSelectFolder-0]
szFolder=IBM Communications Server
Result=1
[AskText-0]
szText=CSAdmins
Result=1
[AskText-1]
szText=10
Result=1
[SdStartCopy-0]
Result=1
[Application]
Name=Communications Server
Version=6.1.1
Company=IBM
Lang=0009
[AskYesNo-0]
Result=0
[AskYesNo-1]
Result=1
[RebootDialog-0]
Result=0
Choice=0

NOTES.ISS

Installation of Lotus Notes:

[InstallShield Silent]
Version=v5.00.000
File=Response File
[File Transfer]
OverwriteReadOnly=NoToAll
[DlgOrder]
Dlg0=SdWelcome-0
Count=8
Dlg1=SdLicense-0
Dlg2=SdRegisterUser-0
Dlg3=SdAskDestPath-0
Dlg4=SdSetupType-0
Dlg5=SdComponentDialog2-0
Dlg6=SdSelectFolder-0


```

Dlg7=SdFinish-0
[SdWelcome-0]
Result=1
[SdLicense-0]
Result=1
[SdRegisterUser-0]
szName=Some Company
szCompany=Some Company
Result=1
[SdAskDestPath-0]
szDir=e:\Lotus\Domino
szDir1=e:\Lotus\Domino\Data
gUpgrade=0
Result=1
[SdSetupType-0]
svSetupType=Domino Server
bCustomize=1
Result=303
[SdComponentDialog2-0]
Common Data Files-type=string
Common Data Files-count=2
Common Data Files-0=Common Data Files\Required Administrative Templates
Common Data Files-1=Common Data Files\Optional Templates
Data Files\Required Data Files-type=string
Data Files\Required Data Files-count=1
Data Files\Required Data Files-0=Data Files\Required Data Files\Smarticons
Data Files-type=string
Data Files-count=4
Data Files-0=Data Files\Required Data Files
Data Files-1=Data Files\Modem Command Scripts
Data Files-2=Data Files\Optional Data Files
Data Files-3=Data Files\Readme
DECS-type=string
DECS-count=4
DECS-0=DECS\Server Program Files
DECS-1=DECS\Data Files
DECS-2=DECS\Documentation
DECS-3=DECS\Program Files
Domino as an NT Service-type=string
Domino as an NT Service-count=1
Domino as an NT Service-0=Domino as an NT Service\Uninstaller
Domino Data Files-type=string
Domino Data Files-count=3
Domino Data Files-0=Domino Data Files\Required Domino Data Files
Domino Data Files-1=Domino Data Files\Optional Data Files
Domino Data Files-2=Domino Data Files\Teamroom
Domino Directory NT Sync Services-type=string
Domino Directory NT Sync Services-count=1

```

Domino Directory NT Sync Services-0=Domino Directory NT Sync Services\Help
 Files
 Domino Server Planner-type=string
 Domino Server Planner-count=1
 Domino Server Planner-0=Domino Server Planner\Doc
 Domino Server Program Files\Web Administration-type=string
 Domino Server Program Files\Web Administration-count=2
 Domino Server Program Files\Web Administration-0=Domino Server Program
 Files\Web Administration\Program Files
 Domino Server Program Files\Web Administration-1=Domino Server Program
 Files\Web Administration\Data
 Domino Server Program Files\Dols Download-type=string
 Domino Server Program Files\Dols Download-count=1
 Domino Server Program Files\Dols Download-0=Domino Server Program Files\Dols
 Download\Filesets
 Domino Server Program Files\iNotes Web Access-type=string
 Domino Server Program Files\iNotes Web Access-count=6
 Domino Server Program Files\iNotes Web Access-0=Domino Server Program
 Files\iNotes Web Access\Help
 Domino Server Program Files\iNotes Web Access-1=Domino Server Program
 Files\iNotes Web Access\SameTime
 Domino Server Program Files\iNotes Web Access-2=Domino Server Program
 Files\iNotes Web Access\DataDominoHtm1
 Domino Server Program Files\iNotes Web Access-3=Domino Server Program
 Files\iNotes Web Access\DataInotes
 Domino Server Program Files\iNotes Web Access-4=Domino Server Program
 Files\iNotes Web Access\Data
 Domino Server Program Files\iNotes Web Access-5=Domino Server Program
 Files\iNotes Web Access\Fonts
 Domino Server Program Files\Billing-type=string
 Domino Server Program Files\Billing-count=2
 Domino Server Program Files\Billing-0=Domino Server Program
 Files\Billing\Program Files
 Domino Server Program Files\Billing-1=Domino Server Program Files\Billing\Data
 Domino Server Program Files-type=string
 Domino Server Program Files-count=9
 Domino Server Program Files-0=Domino Server Program Files\Web Administration
 Domino Server Program Files-1=Domino Server Program Files\DataGif
 Domino Server Program Files-2=Domino Server Program Files\Dols Download
 Domino Server Program Files-3=Domino Server Program Files\iNotes Web Access
 Domino Server Program Files-4=Domino Server Program Files\Billing
 Domino Server Program Files-5=Domino Server Program Files\Domino-Directory
 Domino Server Program Files-6=Domino Server Program Files\Domino Mail Directory
 Domino Server Program Files-7=Domino Server Program Files\DataIcons
 Domino Server Program Files-8=Domino Server Program Files\DataDic
 Domino Web Services\Domino Web Services Data-type=string
 Domino Web Services\Domino Web Services Data-count=4
 Domino Web Services\Domino Web Services Data-0=Domino Web Services\Domino Web
 Services Data\Icons

Domino Web Services\Domino Web Services Data-1=Domino Web Services\Domino Web Services Data\Html
 Domino Web Services\Domino Web Services Data-2=Domino Web Services\Domino Web Services Data\Java
 Domino Web Services\Domino Web Services Data-3=Domino Web Services\Domino Web Services Data\Diiop
 Domino Web Services\Program Files-type=string
 Domino Web Services\Program Files-count=1
 Domino Web Services\Program Files-0=Domino Web Services\Program Files\Sec
 Domino Web Services-type=string
 Domino Web Services-count=2
 Domino Web Services-0=Domino Web Services\Domino Web Services Data
 Domino Web Services-1=Domino Web Services\Program Files
 Help-type=string
 Help-count=4
 Help-0=Help\Dols Help
 Help-1=Help\Administration Help
 Help-2=Help\Client Help
 Help-3=Help\Designer Help
 Notes Performance Monitor-type=string
 Notes Performance Monitor-count=1
 Notes Performance Monitor-0=Notes Performance Monitor\System Files
 Notes Program Files\Required Program Files-type=string
 Notes Program Files\Required Program Files-count=12
 Notes Program Files\Required Program Files-0=Notes Program Files\Required Program Files\FT Codepages
 Notes Program Files\Required Program Files-1=Notes Program Files\Required Program Files\Viewers
 Notes Program Files\Required Program Files-2=Notes Program Files\Required Program Files\FT Viewers
 Notes Program Files\Required Program Files-3=Notes Program Files\Required Program Files\Product Registration
 Notes Program Files\Required Program Files-4=Notes Program Files\Required Program Files\Core Notes
 Notes Program Files\Required Program Files-5=Notes Program Files\Required Program Files\FT Files
 Notes Program Files\Required Program Files-6=Notes Program Files\Required Program Files\Generic System Files
 Notes Program Files\Required Program Files-7=Notes Program Files\Required Program Files\Lotus Script
 Notes Program Files\Required Program Files-8=Notes Program Files\Required Program Files\Ini File
 Notes Program Files\Required Program Files-9=Notes Program Files\Required Program Files\Win95 System Files
 Notes Program Files\Required Program Files-10=Notes Program Files\Required Program Files\Network Drivers
 Notes Program Files\Required Program Files-11=Notes Program Files\Required Program Files\Self Registered
 Notes Program Files\Program Files-type=string

Notes Program Files\Program Files-count=2
 Notes Program Files\Program Files-0=Notes Program Files\Program Files\Sec
 Notes Program Files\Program Files-1=Notes Program Files\Program Files\Filters
 Notes Program Files\Java Support\International-type=string
 Notes Program Files\Java Support\International-count=1
 Notes Program Files\Java Support\International-0=Notes Program Files\Java
 Support\International\Security
 Notes Program Files\Java Support-type=string
 Notes Program Files\Java Support-count=3
 Notes Program Files\Java Support-0=Notes Program Files\Java Support\NonOS2Lib
 Notes Program Files\Java Support-1=Notes Program Files\Java
 Support\International Lib
 Notes Program Files\Java Support-2=Notes Program Files\Java
 Support\International
 Notes Program Files\Import Export Engine-type=string
 Notes Program Files\Import Export Engine-count=1
 Notes Program Files\Import Export Engine-0=Notes Program Files\Import Export
 Engine\Filters
 Notes Program Files-type=string
 Notes Program Files-count=6
 Notes Program Files-0=Notes Program Files\Required Program Files
 Notes Program Files-1=Notes Program Files\Program Files
 Notes Program Files-2=Notes Program Files\Java Support
 Notes Program Files-3=Notes Program Files\JIT Debugger
 Notes Program Files-4=Notes Program Files\Import Export Engine
 Notes Program Files-5=Notes Program Files\Additional Network Drivers
 Spell Checker-type=string
 Spell Checker-count=2
 Spell Checker-0=Spell Checker\International Dictionaries
 Spell Checker-1=Spell Checker\English Dictionaries
 Component-type=string
 Component-count=14
 Component-0=Common Data Files
 Component-1=Data Files
 Component-2=DECS
 Component-3=Domino as an NT Service
 Component-4=Domino Data Files
 Component-5=Domino Directory NT Sync Services
 Component-6=Domino Server Planner
 Component-7=Domino Server Program Files
 Component-8=Domino Web Services
 Component-9=Help
 Component-10=Notes Performance Monitor
 Component-11=Notes Program Files
 Component-12=Spell Checker
 Component-13=Summarizer
 Result=1
 [SdSelectFolder-0]
 szFolder=Lotus Applications

```
Result=1
[Application]
Name=Domino
Version=5.0
Company=Lotus
Lang=0009
[SdFinish-0]
Result=1
bOpt1=0
bOpt2=0
```

DB2.RSP

Installation of IBM DB2 Universal Database:

```
* Sample response file for IBM DB2 Universal Database Enterprise Edition
* -----
*
* Comments are made by placing either a * or a # at the start of a line, or by
* placing ** or ## after the start of a line to comment out the rest of that
* line.
*
* For descriptions of DB2 registry variables, please see Appendix F in the
* Administration Guide.
*
* For descriptions of configuration parameters, please see Chapter 20 in the
* Administration Guide.
*
* Do not uncomment selected components (the COMP keywords) unless you change
* the install TYPE to 2. Install type 2 is a custom install.
*
* When installing on a machine that does not have DB2 already installed with
* all NT services already created,
* at least one of the following pairs of keywords is required. If only one
* pair of the following is specified,
* it will be used for any required user name and password pair in the following
* group not explicitly specified:
*
CONTROL_CENTER_SERVER_USERID = db2admin
CONTROL_CENTER_SERVER_PASSWORD = password
*
* ADMIN.USERID
* ADMIN.PASSWORD
*
* DB2.USERID
* DB2.PASSWORD
*
* CTLSRV.USERID
* CTLSRV.PASSWORD
```

```

*
* DW_CTRLDB_USERID
* DW_CTRLDB_PASSWORD
*
* OLAPSK_USERID
* OLAPSK_PASSWORD
* =====

* General Options
* -----
PROD                = UDB_ENTERPRISE
FILE                = E:\SQLLIB
TYPE                = 1

*COMP              = ODBC_SUPPORT
*COMP              = JDBC_SUPPORT
*COMP              = SQLJ_SUPPORT
*COMP              = IBM_JRE
*COMP              = CONTROL_CENTER
*COMP              = EVENT_ANALYZER
*COMP              = WEB_ADMINISTRATION
*COMP              = CONTROL_SERVER
*COMP              = QUERYENABLER
*COMP              = QUERYMONITOR
*COMP              = TRACKER
*COMP              = QUERYADMIN
*COMP              = CONNECT_SERVER_SUPPORT
*COMP              = LDAP_EXPLOITATION
*COMP              = CLIENT_CONFIGURATION_ASSISTANT
*COMP              = COMMAND_CENTER
*COMP              = FIRST_STEPS
*COMP              = SAMPLE_DATABASE
*COMP              = DATABASE_TOOLS
*COMP              = CLIENT_TOOLS
*COMP              = OLAP_STARTER_KIT_SERVER
*COMP              = OLAP_STARTER_KIT_ADDIN
*COMP              = OLAP_STARTER_KIT_DESKTOP
*COMP              = DATA_WH_SERVER
*COMP              = DATA_WH_CONTROL_DB
*COMP              = OEM_ODBC_DRIVERS
*COMP              = DATA_WH_CENTER
*COMP              = INFO_CATALOG_ADMIN
*COMP              = INFO_CATALOG_USER
*COMP              = WEB_INFO_CATALOG_USER
*COMP              = ADMINISTRATION_GUIDE
*COMP              = APPC_CPIC_SNA_SENSE_CODES
*COMP              = COMMAND_REFERENCE
*COMP              = CONNECTIVITY_SUPPLEMENT
*COMP              = DATA_MOVEMENT_GUIDE

```

```

*COMP = CONNECT_ENTERPRISE_QUICK_BEGINNINGS
*COMP = CONNECT_RELEASE_NOTES
*COMP = CONNECT_USERS_GUIDE
*COMP = DQP_ADMINISTRATION_GUIDE
*COMP = DQP_INSTALLATION_GUIDE
*COMP = DQP_USERS_GUIDE
*COMP = QUICK_BEGINNINGS
*COMP = RELEASE_NOTES
*COMP = REPLICATION_GUIDE
*COMP = GLOSSARY
*COMP = INSTALLING_CONFIGURING_SUPPLEMENT
*COMP = MESSAGE_REFERENCE
*COMP = SQL_GETTING_STARTED
*COMP = SQL_REFERENCE
*COMP = SYSTEM_MONITOR_GUIDE
*COMP = WH_CTR_ADMIN_GUIDE
*COMP = WH_MGR_INSTALL_GUIDE
*COMP = ICM_ADMIN_GUIDE
*COMP = ICM_USER_GUIDE
*COMP = QUICK_TOUR
*COMP = BI_TUTORIAL
*COMP = UNIX_QUICK_BEGIN
*COMP = UNIX_CONEE_QUICK_BEGIN
*COMP = ADMIN_SAT_GUIDE_REF
*COMP = TROUBLESHOOTING_GUIDE
*COMP = WHATS_NEW
*CREATE_ICONS = YES or NO (default=YES)
*AUTOSTART_CCA = YES or NO (default=NO)
*AUTOSTART_CONTROL_CENTER = YES or NO (default=NO)
*AUTOSTART_FIRST_STEPS = YES or NO (default=YES)
*REBOOT = YES or NO (default=NO)
*KILL_PROCESSES = YES or NO (default=NO)
*UPGRADE_ODBC_DRIVER_MGR = YES or NO (default=YES)
*REG_PERF_COUNTERS = YES or NO (default=YES)
*REMOTE_PERF_COUNT_UID = char()
*REMOTE_PERF_COUNT_PWD = char()

* Overwrite read-only system files (msvcrt.dll, msvcirt.dll, mfc42.dll,
mfc42u.dll, msvcrt40.dll)
* YES - The read-only attribute will be removed and the file will be updated if
necessary
* NO - The read-only attribute will not be modified and if read-only files are
encountered
* the install will not be able to continue.
* -----
*REMOVE_READ_ONLY_FROM_MS_FILES = YES or NO (default = YES)

* Control Center Server Logon Settings

```

```

* -----
*CONTROL_CENTER_SERVER_USERID   = char(30) or char(14)\char(30) [char(20) or
char(14)\char(20) for Windows NT]
*CONTROL_CENTER_SERVER_PASSWORD = char(14)

```

* Global DB2 Registry Variables

```

* -----
*DB2ACCOUNT                     = BLANK or char(199)
*DB2BIDI                        = BLANK, YES or NO
*DB2BQTIME                      = BLANK or 1 - MAX
*DB2BQTRY                       = BLANK or 0 - MAX
*DB2CHKPTR                      = BLANK, ON or OFF
*DB2CLIINIPATH                  = BLANK or char(260)
*DB2CODEPAGE                    = BLANK or 0 - MAX
*DB2COMM                        = BLANK or APPC, IPXSPX, NETBIOS, NPIPE, TCPIP
*DB2CONNECT_IN_APP_PROCESS     = BLANK, YES or NO
*DB2COUNTRY                     = BLANK or 1 - 999
*DB2DBDFT                      = BLANK or char(8)
*DB2DEFPREP                    = BLANK, ALL, YES or NO
*DB2DISCOVERYTIME              = BLANK or 20 - MAX
*DB2DMNBCKCTRL                 = BLANK, ? or char()
*DB2_ENABLE_LDAP               = BLANK, YES or NO
*DB2IQTIME                     = BLANK or 1 - MAX
*DB2JD_PORT_NUMBER             = BLANK or 1024-65536
*DB2JVVIEW                     = BLANK, ON or OFF
*DB2LDAPHOST                   = BLANK or host name
*DB2LDAP_BASEDN                = BLANK or char()
*DB2LDAPCACHE                  = BLANK or char()
*DB2LDAP_CLIENT_PROVIDER       = BLANK, MICROSOFT or IBM
*DB2LOADREC                    = BLANK or char(260)
*DB2LOCK_TO_RB                 = BLANK or STATEMENT
*DB2NBADAPTERS                 = BLANK or 0, 1, ..., 15
*DB2NBCHECKUPTIME              = BLANK or 0 - 720
*DB2NBDISCOVERRCVBUFS         = BLANK or 16 - MAX
*DB2NBINTRLISTENS              = BLANK or 1 - 10, 1 - 10, ...
*DB2NBRECVBUFSIZE              = BLANK or 4096 - 65536
*DB2NBRECVNCBS                 = BLANK or 1 - 99, 1 - 99, ...
*DB2NBRESOURCES                = BLANK or (0-15,1-254,1-254,1-254), ...
*DB2NSENDNCBS                  = BLANK or 1 - 99
*DB2NSESSIONS                  = BLANK or 5 - 254, 5 - 254, ...
*DB2NBXTRANCBS                 = BLANK or 5 - 254, 5 - 254, ...
*DB2NOEXITLIST                 = BLANK, ON or OFF
*DB2NTNOCACHE                  = BLANK, ON or OFF
*DB2NTPRICLASS                 = BLANK, R or H
*DB2NTWORKSET                  = BLANK or 0-2048, 0-2048
*DB2OLDEVMON                   = BLANK or char()

```



```

*DB2OPTIONS                = BLANK or char():
-/[a,c,e[c[s],n,o,p,s,t,v,w,x] and/or -[f,l,r,z] filename
*DB2PRIORITIES             = BLANK or char()
*DB2RETRY                   = BLANK or 0 - 20000
*DB2RETRYTIME              = BLANK or 0 - 7200
*DB2RQTIME                 = BLANK or 1 - MAX
*DB2ROUTINE_DEBUG          = BLANK, ON or OFF
*DB2SERVICETPINSTANCE      = BLANK or char(8)
*DB2SORCVBUF               = BLANK or 1024-65536
*DB2SORT                   = BLANK or char(260)
*DB2SOSNDBUF              = BLANK or 1024-65536
*DB2SYSPLEX_SERVER         = BLANK, 0 or 1
*DB2SYSTEM                 = char(21)
*DB2_AVOID_PREFETCH        = BLANK, ON or OFF
*DB2_CORRELATED_PREDICATES = BLANK, ON or OFF
*DB2_FALLBACK              = BLANK, ON or OFF
*DB2_FORCE_TRUNCATION      = BLANK, YES or NO
*DB2_GRP_LOOKUP            = BLANK or char()
*DB2_HASH_JOIN             = BLANK, YES or NO
*DB2_INDEX_FREE            = BLANK or 0 - 60
*DB2_LIKE_VARCHAR          = BLANK, YES, NO or 0.0 - 5.0 (exclusive)
*DB2_LOADSORT_STACKSZ     = BLANK or 1 - MAX
*DB2_NEW CORR_SQ_FF        = BLANK, ON or OFF
*DB2_NO_PKG_LOCK           = BLANK, ON or OFF
*DB2_PARALLEL_IO           = BLANK, * or 0-4095, 0-4095, ...
*DB2_PRED_FACTORIZE        = BLANK, YES or NO
*DB2_RR_TO_RS              = BLANK, YES or NO
*DB2_STPROC_ALLOW_LOCAL_FENCED = BLANK, YES or NO
*DB2_STPROC_LOOKUP_FIRST  = BLANK, YES or NO
*DB2_STRIPED_CONTAINERS   = BLANK, ON or OFF
*DB2_USE_JDK12             = BLANK, YES or NO

```

* Default Instance Client Import Profile file

```

* -----
*DB2.CLIENT_IMPORT_PROFILE = filename

```

* Default Instance Auto-start Option

```

* -----
*DB2.AUTOSTART             = YES or NO (default=YES)

```

* Default Instance TCP/IP port number

```

* -----
*DB2.PORT_NUMBER          = 1024 - 65535

```

* Default Instance Logon Settings

```

* -----
*DB2.USERID          = char(30) or char(14)\char(30) [char(20) or
char(14)\char(20) for Windows NT]
*DB2.PASSWORD       = char(14)

* Default Instance DBM CFG settings
* -----
*DB2.AGENTPRI       = SYSTEM or 0 - 6
*DB2.AGENT_STACK_SZ = 8 - 1000
*DB2.ASLHEAPSZ     = 1 - 524288
*DB2.AUDIT_BUF_SZ  = 0 - 65000
*DB2.AUTHENTICATION = CLIENT, DCS, DCS_ENCRYPT, DCE,
DCE_SERVER_ENCRYPT, SERVER, SERVER_ENCRYPT, KERBEROS(Windows 2000 only) or
KRB_SERVER_ENCRYPT(Windows 2000 only)
*DB2.BACKBUFSZ     = 16 - 524288
*DB2.CATALOG_NOAUTH = YES or NO
*DB2.CPUSPEED      = -1 or 1e-10 - 1
*DB2.DATALINKS     = YES or NO
*DB2.DFTDBPATH     = <drive letter>: (not a: or b:, must exist)
*DB2.DFT_ACCOUNT_STR = BLANK or char(25)
*DB2.DFT_CLIENT_COMM = BLANK or APPC, IPXSPX, NETBIOS, TCPIP, NPIPE
*DB2.DFT_MON_BUFPOOL = ON or OFF
*DB2.DFT_MON_LOCK  = ON or OFF
*DB2.DFT_MON_SORT  = ON or OFF
*DB2.DFT_MON_STMT  = ON or OFF
*DB2.DFT_MON_TABLE = ON or OFF
*DB2.DFT_MON_UOW   = ON or OFF
*DB2.DIAGLEVEL     = 0 - 4
*DB2.DIAGPATH      = BLANK or char(215)
*DB2.DIR_CACHE     = YES or NO
*DB2.DIR_OBJ_NAME  = BLANK or char(255) (length of DIR_OBJ_NAME +
DIR_PATH_NAME < = 255)
*DB2.DIR_PATH_NAME = BLANK or char(255) (length of DIR_OBJ_NAME +
DIR_PATH_NAME < = 255)
*DB2.DIR_TYPE      = DCE or NONE
*DB2.DISCOVER      = DISABLE, KNOWN or SEARCH
*DB2.DISCOVER_COMM = BLANK or NETBIOS, TCPIP
*DB2.DISCOVER_INST = ENABLE or DISABLE
*DB2.DOS_RQRI0BLK = 4096 - 65535
*DB2.DRDA_HEAP_SZ  = 16 - 60000
*DB2.FCM_NUM_ANCHORS = -1 or 128 - FCM_NUM_RQB
*DB2.FCM_NUM_BUFFERS = 128 - 65300
*DB2.FCM_NUM_CONNECT = -1 or 128 - FCM_NUM_RQB
*DB2.FCM_NUM_RQB   = 128 - 120000
*DB2.FILESERVER    = BLANK or char(48)
*DB2.INDEXREC      = ACCESS or RESTART
*DB2.INITDARI_JVM  = YES or NO
*DB2.INTRA_PARALLEL = SYSTEM, YES or NO

```

*DB2.IPX_SOCKET = BLANK or 0000 - FFFF
 *DB2.JAVA_HEAP_SZ = 0 - 4096
 *DB2.JDK11_PATH = BLANK or char(255)
 *DB2.KEEPDARI = YES or NO
 *DB2.MAXAGENTS = 1 - 64000
 *DB2.MAXCAGENTS = -1 or 1 - MAX_COORDAGENTS
 *DB2.MAXDARI = -1 or 1 - MAX_COORDAGENTS
 *DB2.MAXTOTFILOP = 100 - 32768
 *DB2.MAX_COORDAGENTS = -1 or 1 - MAXAGENTS (MAX_COORDAGENTS +
 NUM_INITAGENTS cannot be greater than MAXAGENTS)
 *DB2.MAX_LOGICAGENTS = -1 - 64000 (cannot be less than
 MAX_COORDAGENTS)
 *DB2.MAX_QUERYDEGREE = ANY or 0 - 32767
 *DB2.MIN_PRIV_MEM = 32 - 112000
 *DB2.MON_HEAP_SZ = 0 - 60000
 *DB2.NNAME = BLANK or char(8)
 *DB2.NOTIFYLEVEL = 0 - 4
 *DB2.NUMDB = 1 - 256
 *DB2.NUM_INITAGENTS = 1 - NUM_POOLAGENTS (MAX_COORDAGENTS +
 NUM_INITAGENTS < = MAXAGENTS)
 *DB2.NUM_INITDARIS = -1 or 0 - MAXDARI
 *DB2.NUM_POOLAGENTS = -1 or 1 - MAXAGENTS
 *DB2.OBJECTNAME = BLANK or char(48)
 *DB2.PRIV_MEM_THRESH = -1 or 32 - 112000
 *DB2.QUERY_HEAP_SZ = 2 - 524288
 *DB2.RESTBUFSZ = 16 - 524288
 *DB2.RESYNC_INTERVAL = 60 - 60000
 *DB2.ROUTE_OBJ_NAME = BLANK or char(255) (length of SQL_DIR_NAME_SZ)
 *DB2.RQRIOBLK = 4096 - 65535
 *DB2.SHEAPTHRES = 250 - 2097152
 *DB2.SPM_LOG_FILE_SZ = 4 - 1000
 *DB2.SPM_LOG_PATH = BLANK or char(226)
 *DB2.SPM_MAX_RESYNC = 10 - 256
 *DB2.SPM_NAME = BLANK or char(8)
 *DB2.SVCENAME = BLANK or char(14)
 *DB2.SYSADM_GROUP = BLANK or char(30) [char(20) on Windows NT]
 *DB2.SYSCTRL_GROUP = BLANK or char(30) [char(20) on Windows NT]
 *DB2.SYSMAINT_GROUP = BLANK or char(30) [char(20) on Windows NT]
 *DB2.TM_DATABASE = BLANK or char(8)
 *DB2.TPNAME = BLANK or char(64)
 *DB2.TP_MON_NAME = BLANK or char(19)
 *DB2.TRUST_ALLCLNTS = YES, NO or DRDAONLY
 *DB2.TRUST_CLNTAUTH = CLIENT or SERVER
 *DB2.UDF_MEM_SZ = 128 - 60000

* Default Instance DB2 Registry Variables

* -----

*DB2.DB2ACCOUNT = BLANK or char(199)

*DB2.DB2BIDI	= BLANK, YES or NO
*DB2.DB2BQTIME	= BLANK or 1 - MAX
*DB2.DB2BQTRY	= BLANK or 0 - MAX
*DB2.DB2CHKPTR	= BLANK, ON or OFF
*DB2.DB2CLIINIPATH	= BLANK or char(260)
*DB2.DB2CODEPAGE	= BLANK or 0 - MAX
*DB2.DB2COMM	= BLANK or APPC, IPXSPX, NETBIOS, NPIPE, TCPIP
*DB2.DB2CONNECT_IN_APP_PROCESS	= BLANK, YES or NO
*DB2.DB2COUNTRY	= BLANK or 1 - 999
*DB2.DB2DBDFT	= BLANK or char(8)
*DB2.DB2DEFPREP	= BLANK, ALL, YES or NO
*DB2.DB2DISCOVERYTIME	= BLANK or 20 - MAX
*DB2.DB2DMNBCKCTLR	= BLANK, ? or char()
*DB2.DB2IQTIME	= BLANK or 1 - MAX
*DB2.DB2JD_PORT_NUMBER	= BLANK or 1024-65536
*DB2.DB2JVVIEW	= BLANK, ON or OFF
*DB2.DB2LOADREC	= BLANK or char(260)
*DB2.DB2LOCK_TO_RB	= BLANK or STATEMENT
*DB2.DB2NBADAPTERS	= BLANK or 0, 1, ..., 15
*DB2.DB2NBCHECKUPTIME	= BLANK or 0 - 720
*DB2.DB2NBDISCOVERRCVBUFS	= BLANK or 16 - MAX
*DB2.DB2NBINTRLISTENS	= BLANK or 1 - 10, 1 - 10, ...
*DB2.DB2NBRECVBUFSIZE	= BLANK or 4096 - 65536
*DB2.DB2NBRECVNCBS	= BLANK or 1 - 99, 1 - 99, ...
*DB2.DB2NBRESOURCES	= BLANK or (0-15,1-254,1-254,1-254), ...
(0-15,1-254,1-254,1-254), ...	
*DB2.DB2NBSENDNCBS	= BLANK or 1 - 99
*DB2.DB2NBSESSIONS	= BLANK or 5 - 254, 5 - 254, ...
*DB2.DB2NBXTRANCBS	= BLANK or 5 - 254, 5 - 254, ...
*DB2.DB2NOEXITLIST	= BLANK, ON or OFF
*DB2.DB2NTNOCACHE	= BLANK, ON or OFF
*DB2.DB2NTPRICLASS	= BLANK, R or H
*DB2.DB2NTWORKSET	= BLANK or 0-2048,0-2048
*DB2.DB2OLDEVMON	= BLANK or char()
*DB2.DB2OPTIONS	= BLANK or char():
-/[a,c,e[c s],n,o,p,s,t,v,w,x]	and/or -[f,l,r,z]filename
*DB2.DB2PRIORITIES	= BLANK or char()
*DB2.DB2RETRY	= BLANK or 0 - 20000
*DB2.DB2RETRYTIME	= BLANK or 0 - 7200
*DB2.DB2RQTIME	= BLANK or 1 - MAX
*DB2.DB2ROUTINE_DEBUG	= BLANK, ON or OFF
*DB2.DB2SORCVBUF	= BLANK or 1024-65536
*DB2.DB2SORT	= BLANK or char(260)
*DB2.DB2SOSNDBUF	= BLANK or 1024-65536
*DB2.DB2SYSPLEX_SERVER	= BLANK, 0 or 1
*DB2.DB2_AVOID_PREFETCH	= BLANK, ON or OFF
*DB2.DB2_CORRELATED_PREDICATES	= BLANK, ON or OFF
*DB2.DB2_FALLBACK	= BLANK, ON or OFF
*DB2.DB2_FORCE_TRUNCATION	= BLANK, YES or NO

```

*DB2.DB2_GRP_LOOKUP           = BLANK or char()
*DB2.DB2_HASH_JOIN           = BLANK, YES or NO
*DB2.DB2_INDEX_FREE          = BLANK or 0 - 60
*DB2.DB2_LIKE_VARCHAR         = BLANK, YES, NO or 0.0 - 5.0 (exclusive)
*DB2.DB2_LOADSORT_STACKSZ    = BLANK or 1 - MAX
*DB2.DB2_NO_PKG_LOCK         = BLANK, ON or OFF
*DB2.DB2_PARALLEL_IO         = BLANK, * or 0-4095, 0-4095, ...
*DB2.DB2_PRED_FACTORIZE      = BLANK, YES or NO
*DB2.DB2_RR_TO_RS            = BLANK, YES or NO
*DB2.DB2_STRIPED_CONTAINERS  = BLANK, ON or OFF

```

* Administration Server Logon Settings

```

* -----
*ADMIN.USERID                 = char(30) or char(14)\char(30) [char(20) or
char(14)\char(20) for Windows NT]
*ADMIN.PASSWORD               = char(14)

```

* Administration Server ADMIN CFG Settings

```

* -----
*ADMIN.AGENT_STACK_SZ         = 8 - 1000
*ADMIN.AUTHENTICATION         = CLIENT, DCS, DCS_ENCRYPT, DCE,
DCE_SERVER_ENCRYPT, SERVER, SERVER_ENCRYPT, KERBEROS(Windows 2000 only) or
KRB_SERVER_ENCRYPT(Windows 2000 only)
*ADMIN.DIAGLEVEL              = 0 - 4
*ADMIN.DIAGPATH               = BLANK or char(215)
*ADMIN.DISCOVER               = DISABLE, KNOWN or SEARCH
*ADMIN.DISCOVER_COMM          = BLANK or NETBIOS, TCPIP
*ADMIN.FILESERVER              = BLANK or char(48)
*ADMIN.NNAME                   = BLANK or char(8)
*ADMIN.NOTIFYLEVEL            = 0 - 4
*ADMIN.OBJECTNAME              = BLANK or char(48)
*ADMIN.QUERY_HEAP_SZ          = 2 - 524288
*ADMIN.SYSADM_GROUP            = BLANK or char(30) [char(20) on Windows NT]
*ADMIN.SYSCTRL_GROUP           = BLANK or char(30) [char(20) on Windows NT]
*ADMIN.SYSMAINT_GROUP         = BLANK or char(30) [char(20) on Windows NT]
*ADMIN.TPNAME                  = BLANK or char(64)
*ADMIN.TRUST_ALLCLNTS         = YES, NO or DRDAONLY
*ADMIN.TRUST_CLNTAUTH         = CLIENT or SERVER

```

* Administration Server DB2 Registry Variables

```

* -----
*ADMIN.DB2CHKPTR              = BLANK, ON or OFF
*ADMIN.DB2CODEPAGE            = BLANK or 0 - MAX
*ADMIN.DB2COMM                 = BLANK or APPC, IPXSPX, NETBIOS, NPIPE, TCPIP
*ADMIN.DB2COUNTRY              = BLANK or 1 - 999
*ADMIN.DB2DMNBCKCTLR          = BLANK, ? or char()

```

*ADMIN.DB2NBADAPTERS = BLANK or 0, 1, ..., 15
 *ADMIN.DB2NBCHECKUPTIME = BLANK or 0 - 720
 *ADMIN.DB2NBINTRLISTENS = BLANK or 1 - 10, 1 - 10, ...
 *ADMIN.DB2NBRECVBUFSIZE = BLANK or 4096 - 65536
 *ADMIN.DB2NBRECVNCBS = BLANK or 1 - 99, 1 - 99, ...
 *ADMIN.DB2NBRESOURCES = BLANK or (0-15,1-254,1-254,1-254), ...
 (0-15,1-254,1-254,1-254), ...
 *ADMIN.DB2NBSENDNCBS = BLANK or 1 - 99
 *ADMIN.DB2NBSESSIONS = BLANK or 5 - 254, 5 - 254, ...
 *ADMIN.DB2NBXTRANCBS = BLANK or 5 - 254, 5 - 254, ...
 *ADMIN.DB2NTWORKSET = BLANK or 0-2048, 0-2048
 *ADMIN.DB2PRIORITIES = BLANK or char()

* Satellite Control Server

* -----

* These keywords only apply if TYPE=1, or TYPE=2 and COMP=CONTROL_SERVER are specified

* above.

* System will be a dedicated Control Server

* -----

*CTLSRV.DEDICATED_CTLSRV = YES or NO (default=NO)

* Control Server Instance Auto-start Option

* -----

*CTLSRV.AUTOSTART = YES or NO (default=YES)

* Control Server Instance TCP/IP port number

* -----

*CTLSRV.PORT_NUMBER = 1024 - 65535

* Control Server Instance Logon Settings

* -----

*CTLSRV.USERID = char(30) or char(14)\char(30) [char(20) or char(14)\char(20) for Windows NT]

*CTLSRV.PASSWORD = char(14)

* Control Server Instance DBM CFG settings

* -----

*CTLSRV.AGENTPRI = SYSTEM or 0 - 6

*CTLSRV.AGENT_STACK_SZ = 8 - 1000

*CTLSRV.ASLHEAPSZ = 1 - 524288

*CTLSRV.AUDIT_BUF_SZ = 0 - 65000

*CTLSRV.AUTHENTICATION = CLIENT, DCS, DCS_ENCRYPT, DCE,
 DCE_SERVER_ENCRYPT, SERVER, SERVER_ENCRYPT, KERBEROS(Windows 2000 only) or
 KRB_SERVER_ENCRYPT(Windows 2000 only)
 *CTLSRV.BACKBUFSZ = 16 - 524288
 *CTLSRV.CATALOG_NOAUTH = YES or NO
 *CTLSRV.CPUSPEED = -1 or 1e-10 - 1
 *CTLSRV.DATALINKS = YES or NO
 *CTLSRV.DFTDBPATH = <drive letter>: (not a: or b:, must exist)
 *CTLSRV.DFT_ACCOUNT_STR = BLANK or char(25)
 *CTLSRV.DFT_CLIENT_COMM = BLANK or APPC, IPXSPX, NETBIOS, TCPIP, NPIPE
 *CTLSRV.DFT_MON_BUFPOOL = ON or OFF
 *CTLSRV.DFT_MON_LOCK = ON or OFF
 *CTLSRV.DFT_MON_SORT = ON or OFF
 *CTLSRV.DFT_MON_STMT = ON or OFF
 *CTLSRV.DFT_MON_TABLE = ON or OFF
 *CTLSRV.DFT_MON_UOW = ON or OFF
 *CTLSRV.DIAGLEVEL = 0 - 4
 *CTLSRV.DIAGPATH = BLANK or char(215)
 *CTLSRV.DIR_CACHE = YES or NO
 *CTLSRV.DIR_OBJ_NAME = BLANK or char(255) (length of DIR_OBJ_NAME +
 DIR_PATH_NAME < = 255)
 *CTLSRV.DIR_PATH_NAME = BLANK or char(255) (length of DIR_OBJ_NAME +
 DIR_PATH_NAME < = 255)
 *CTLSRV.DIR_TYPE = DCE or NONE
 *CTLSRV.DISCOVER = DISABLE, KNOWN or SEARCH
 *CTLSRV.DISCOVER_COMM = BLANK or NETBIOS, TCPIP
 *CTLSRV.DISCOVER_INST = ENABLE or DISABLE
 *CTLSRV.DRDA_HEAP_SZ = 16 - 60000
 *CTLSRV.FCM_NUM_ANCHORS = -1 or 128 - FCM_NUM_RQB
 *CTLSRV.FCM_NUM_BUFFERS = 128 - 65300
 *CTLSRV.FCM_NUM_CONNECT = -1 or 128 - FCM_NUM_RQB
 *CTLSRV.FCM_NUM_RQB = 128 - 120000
 *CTLSRV.FILESERVER = BLANK or char(48)
 *CTLSRV.INDEXREC = ACCESS or RESTART
 *CTLSRV.INITDARI_JVM = YES or NO
 *CTLSRV.INTRA_PARALLEL = SYSTEM, YES or NO
 *CTLSRV.IPX_SOCKET = BLANK or 0000 - FFFF
 *CTLSRV.JAVA_HEAP_SZ = 0 - 4096
 *CTLSRV.JDK11_PATH = BLANK or char(255)
 *CTLSRV.KEEPDARI = YES or NO
 *CTLSRV.MAXAGENTS = 1 - 64000
 *CTLSRV.MAXCAGENTS = -1 or 1 - MAX_COORDAGENTS
 *CTLSRV.MAXDARI = -1 or 1 - MAX_COORDAGENTS
 *CTLSRV.MAXTOTFILOP = 100 - 32768
 *CTLSRV.MAX_COORDAGENTS = -1 or 1 - MAXAGENTS (MAX_COORDAGENTS +
 NUM_INITAGENTS cannot be greater than MAXAGENTS)
 *CTLSRV.MAX_LOGICAGENTS = -1 - 64000 (cannot be less than
 MAX_COORDAGENTS)
 *CTLSRV.MAX_QUERYDEGREE = ANY or 0 - 32767

*CTLSRV.MIN_PRIV_MEM	= 32 - 112000
*CTLSRV.MON_HEAP_SZ	= 0 - 60000
*CTLSRV.NNAME	= BLANK or char(8)
*CTLSRV.NOTIFYLEVEL	= 0 - 4
*CTLSRV.NUMDB	= 1 - 256
*CTLSRV.NUM_INITAGENTS	= 1 - NUM_POOLAGENTS (MAX_COORDAGENTS +
NUM_INITAGENTS <	= MAXAGENTS)
*CTLSRV.NUM_POOLAGENTS	= -1 or 1 - MAXAGENTS
*CTLSRV.OBJECTNAME	= BLANK or char(48)
*CTLSRV.PRIV_MEM_THRESH	= -1 or 32 - 112000
*CTLSRV.QUERY_HEAP_SZ	= 2 - 524288
*CTLSRV.RESTBUFSZ	= 16 - 524288
*CTLSRV.RESYNC_INTERVAL	= 60 - 60000
*CTLSRV.RQRIOBLK	= 4096 - 65535
*CTLSRV.SHEAPTHRES	= 250 - 2097152
*CTLSRV.SPM_LOG_FILE_SZ	= 4 - 1000
*CTLSRV.SPM_LOG_PATH	= BLANK or char(226)
*CTLSRV.SPM_MAX_RESYNC	= 10 - 256
*CTLSRV.SPM_NAME	= BLANK or char(8)
*CTLSRV.SVCENAME	= BLANK or char(14)
*CTLSRV.SYSADM_GROUP	= BLANK or char(30) [char(20) on Windows NT]
*CTLSRV.SYSCTRL_GROUP	= BLANK or char(30) [char(20) on Windows NT]
*CTLSRV.SYSMAINT_GROUP	= BLANK or char(30) [char(20) on Windows NT]
*CTLSRV.TM_DATABASE	= BLANK or char(8)
*CTLSRV.TPNAME	= BLANK or char(64)
*CTLSRV.TP_MON_NAME	= BLANK or char(19)
*CTLSRV.TRUST_ALLCLNTS	= YES, NO or DRDAONLY
*CTLSRV.TRUST_CLNTAUTH	= CLIENT or SERVER
*CTLSRV.UDF_MEM_SZ	= 128 - 60000

* Control Server Instance DB2 Registry Variables

* -----

*CTLSRV.DB2ACCOUNT	= BLANK or char(199)
*CTLSRV.DB2BIDI	= BLANK, YES or NO
*CTLSRV.DB2BQTIME	= BLANK or 1 - MAX
*CTLSRV.DB2BQTRY	= BLANK or 0 - MAX
*CTLSRV.DB2CHKPTR	= BLANK, ON or OFF
*CTLSRV.DB2CLIINIPATH	= BLANK or char(260)
*CTLSRV.DB2CODEPAGE	= BLANK or 0 - MAX
*CTLSRV.DB2COMM	= BLANK or APPC, IPXSPX, NETBIOS, NPIPE, TCPIP
*CTLSRV.DB2CONNECT_IN_APP_PROCESS	= BLANK, YES or NO
*CTLSRV.DB2COUNTRY	= BLANK or 1 - 999
*CTLSRV.DB2DBDFT	= BLANK or char(8)
*CTLSRV.DB2DEFPREP	= BLANK, ALL, YES or NO
*CTLSRV.DB2DISCOVERYTIME	= BLANK or 20 - MAX
*CTLSRV.DB2DMNBCKCTRL	= BLANK, ? or char()
*CTLSRV.DB2IQTIME	= BLANK or 1 - MAX

*CTLSRV.DB2JVVIEW = BLANK, ON or OFF
 *CTLSRV.DB2LOADREC = BLANK or char(260)
 *CTLSRV.DB2LOCK_TO_RB = BLANK or STATEMENT
 *CTLSRV.DB2NBADAPTERS = BLANK or 0, 1, ..., 15
 *CTLSRV.DB2NBCHECKUPTIME = BLANK or 0 - 720
 *CTLSRV.DB2NBDISCOVERRCVBUFS = BLANK or 16 - MAX
 *CTLSRV.DB2NBINTRLISTENS = BLANK or 1 - 10, 1 - 10, ...
 *CTLSRV.DB2NBRECVBUFFSIZE = BLANK or 4096 - 65536
 *CTLSRV.DB2NBRECVNCBS = BLANK or 1 - 99, 1 - 99, ...
 *CTLSRV.DB2NBRESOURCES = BLANK or (0-15,1-254,1-254,1-254),
 (0-15,1-254,1-254,1-254), ...
 *CTLSRV.DB2NSENDNCBS = BLANK or 1 - 99
 *CTLSRV.DB2NBSSESSIONS = BLANK or 5 - 254, 5 - 254, ...
 *CTLSRV.DB2NBXTRANCBS = BLANK or 5 - 254, 5 - 254, ...
 *CTLSRV.DB2NEXITLIST = BLANK, ON or OFF
 *CTLSRV.DB2NTNOCACHE = BLANK, ON or OFF
 *CTLSRV.DB2NTPRICLASS = BLANK, R or H
 *CTLSRV.DB2NTWORKSET = BLANK or 0-2048,0-2048
 *CTLSRV.DB2OLDEVMON = BLANK or char()
 *CTLSRV.DB2OPTIONS = BLANK or char():
 -/[a,c,e[c[s],n,o,p,s,t,v,w,x] and/or -[f,l,r,z] filename
 *CTLSRV.DB2PRIORITIES = BLANK or char()
 *CTLSRV.DB2RETRY = BLANK or 0 - 20000
 *CTLSRV.DB2RETRYTIME = BLANK or 0 - 7200
 *CTLSRV.DB2RQTIME = BLANK or 1 - MAX
 *CTLSRV.DB2ROUTINE_DEBUG = BLANK, ON or OFF
 *CTLSRV.DB2SORCVBUF = BLANK or 1024-65536
 *CTLSRV.DB2SORT = BLANK or char(260)
 *CTLSRV.DB2SOSNDBUF = BLANK or 1024-65536
 *CTLSRV.DB2SYSPLEX_SERVER = BLANK, 0 or 1
 *CTLSRV.DB2_AVOID_PREFETCH = BLANK, ON or OFF
 *CTLSRV.DB2_CORRELATED_PREDICATES = BLANK, ON or OFF
 *CTLSRV.DB2_FALLBACK = BLANK, ON or OFF
 *CTLSRV.DB2_FORCE_TRUNCATION = BLANK, YES or NO
 *CTLSRV.DB2_GRP_LOOKUP = BLANK or char()
 *CTLSRV.DB2_HASH_JOIN = BLANK, YES or NO
 *CTLSRV.DB2_INDEX_FREE = BLANK or 0 - 60
 *CTLSRV.DB2_LIKE_VARCHAR = BLANK, YES, NO or 0.0 - 5.0 (exclusive)
 *CTLSRV.DB2_LOADSORT_STACKSZ = BLANK or 1 - MAX
 *CTLSRV.DB2_NO_PKG_LOCK = BLANK, ON or OFF
 *CTLSRV.DB2_PARALLEL_IO = BLANK, * or 0-4095, 0-4095, ...
 *CTLSRV.DB2_PRED_FACTORIZE = BLANK, YES or NO
 *CTLSRV.DB2_RR_TO_RS = BLANK, YES or NO
 *CTLSRV.DB2_STRIPED_CONTAINERS = BLANK, ON or OFF

 * OLAP Starter Kit Options
 * -----
 * OLAPSK_USERID = char(30) or char(14)\char(30) [char(20) or
 char(14)\char(20) for Windows NT]

* OLAPSK_PASSWORD = char(14)
* OLAPSK_PROD_DB = char(8)
* OLAPSK_DEV_DB = char(8)
* OLAPSK_PATH = char(256) or SKIP_OLAP to skip

* Data Warehousing

* -----

* These keywords are only applicable to Data Warehousing configuration.

* The following keyword only applies if one of the following hold true:
* a) Visual Warehouse is not installed, the DATA_WH_CONTROL_DB component is not selected, but the DATA_WH_SERVER component is selected.
* b) Visual Warehouse does not exist on the system and the DATA_WH_CONTROL_DB component is selected.
* c) Visual Warehouse is installed on the machine, the user has decided not to migrate the Visual Warehouse Control Database, the DATA_WH_CONTROL_DB component is selected, and the DATA_WH_SERVER component is not selected.

*DW_CTRLDB_NAME = char(8)

* The following keywords only apply if one of the following hold true:
* a) Visual Warehouse does not exist on the system and the DATA_WH_CONTROL_DB component is selected.
* b) Visual Warehouse is installed on the machine, the user has decided not to migrate the Visual Warehouse Control Database, the DATA_WH_CONTROL_DB component is selected, and the DATA_WH_SERVER component is not selected.
* c) Visual Warehouse exists on the system, a Visual Warehouse Control Database exists, and one of: DATA_WH_SERVER is selected and DATA_WH_CONTROL_DB is selected; DATA_WH_SERVER is selected and DATA_WH_CONTROL_DB is not selected;
* DATA_WH_SERVER is not selected and DATA_WH_CONTROL_DB is selected.

*DW_CTRLDB_USERID = char(30) [char(20) for Windows NT]
*DW_CTRLDB_PASSWORD = char(14)

* The following keywords only apply if Visual Warehouse is not installed, the DATA_WH_CONTROL_DB component is not selected, but the DATA_WH_SERVER component is selected.

*DW_CTRLDB_PORT_NAME = 1024-65536
*DW_CTRLDB_HOSTNAME = char()
*DW_CTRLDB_TCPIP_NODE = BLANK or char()

* The following keywords only apply if one of the following hold true:
* a) Visual Warehouse does not exist on the system and the DATA_WH_CONTROL_DB

- * component is selected.
- * b) Visual Warehouse is installed on the machine, the user has decided not to
- * migrate the Visual Warehouse Control Database, the DATA_WH_CONTROL_DB
- * component is selected, and the DATA_WH_SERVER component is not selected.

```
*DW_CTRLDB_INSTANCE_NAME      = BLANK or char(8)
*Dw_CTRLDB_SCHEMA              = BLANK or char(30)
WINMEMP.TXT
```

Domain migration related files

Below are import files and protocol files used in Chapter 4, “Migrating OS/2 Servers to Windows 2000” on page 87.

BASEOU.LDIF

Create Base OU prior to first migration:

```
dn: OU=GPO,DC=somedomain,DC=local
changetype: add
description: Container for Group policy objects
objectClass: organizationalUnit
ou: GPO
```

```
dn: OU=Branch,DC=somedomain,DC=local
changetype: add
description: Container for all branches
objectClass: organizationalUnit
ou: Branch
```

```
dn: OU=Systems,DC=somedomain,DC=local
changetype: add
description: Base container for computer and server objects
objectClass: organizationalUnit
ou: Systems
```

```
dn: OU=Servers,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Server objects
objectClass: organizationalUnit
ou: Servers
```

```
dn: OU=Metaframe,OU=Servers,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for Terminal Server objects
objectClass: organizationalUnit
ou: Metaframe
```

```
dn: OU=File,OU=Servers,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for file server objects
objectClass: organizationalUnit
ou: File

dn: OU=Print,OU=Servers,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for print server objects
objectClass: organizationalUnit
ou: Print

dn: OU=Domain Controllers,OU=Servers,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for Domain controllers
objectClass: organizationalUnit
ou: Domain Controllers

dn: OU=Application,OU=Servers,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for application servers like DB2, Notes,...
objectClass: organizationalUnit
ou: Application

dn: OU=Workstations,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Client computer objects
objectClass: organizationalUnit
ou: Workstations

dn: OU=Notebooks,OU=Workstations,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for notebook computer objects
objectClass: organizationalUnit
ou: Notebooks

dn: OU=Desktops,OU=Workstations,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for standard desktop computer objects
objectClass: organizationalUnit
ou: Desktops

dn: OU=Special,OU=Workstations,OU=Systems,DC=somedomain,DC=local
changetype: add
description: Container for non-standard workstation objects
objectClass: organizationalUnit
ou: Special
```

```
dn: OU=Central,DC=somedomain,DC=local
changetype: add
description: Centrally defined user and group objects
objectClass: organizationalUnit
ou: Central
```

```
dn: OU=Users,OU=Central,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Users
```

```
dn: OU=FTP,OU=Users,OU=Central,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: FTP
```

```
dn: OU=NFS,OU=Users,OU=Central,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: NFS
```

```
dn: OU=Administrators,OU=Users,OU=Central,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Administrators
```

```
dn: OU=Services,OU=Users,OU=Central,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Services
```

```
dn: OU=Groups,OU=Central,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Groups
```

BRANCHOU.LDIF

Create Base OU prior to branch domain migration.

```
dn: OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: {DomainName}
```

```
dn: OU=Users,OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Users
```

```

dn: OU=Groups,OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
objectClass: organizationalUnit
ou: Groups

dn: OU=Application,OU=Groups,OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
description: Container for groups assigning applications to users
objectClass: organizationalUnit
ou: Application

dn: OU=Access,OU=Groups,OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
description: Container for groups granting access to resources
objectClass: organizationalUnit
ou: Access

dn: OU=Print,OU=Groups,OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
description: Groups for granting access to printer queues
objectClass: organizationalUnit
ou: Print

dn: OU=Organization,OU=Groups,OU={DomainName},OU=Branch,DC=somedomain,DC=local
changetype: add
description: Groups defining organisational membership of users (useable as DL)
objectClass: organizationalUnit
ou: Organization

```

Migrating groups

Here you find import files and protocol files used in 4.4, “Migrating groups” on page 108.

GROUPS.LDIF

Import file for group definitions:

```

dn: CN=BOOKREAD,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: BOOKREAD
distinguishedName: CN=BOOKREAD,CN=Users,DC=somedomain,DC=local
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: group
name: BOOKREAD
sAMAccountName: BOOKREAD

```

```
dn: CN=BOOKWRITE,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: BOOKWRITE
distinguishedName: CN=BOOKWRITE,CN=Users,DC=somedomain,DC=local
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: group
name: BOOKWRITE
sAMAccountName: BOOKWRITE
```

```
dn: CN=PRINTER,OU=Print,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: PRINTER
description: Printer Group
distinguishedName: CN=PRINTER,CN=Users,DC=somedomain,DC=local
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: group
name: PRINTER
sAMAccountName: PRINTER
```

```
dn: CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: TRANSITION
distinguishedName: CN=TRANSITION,CN=Users,DC=somedomain,DC=local
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: group
name: TRANSITION
sAMAccountName: TRANSITION
```

```
dn: CN=BRANCH1,OU=Organization,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: BRANCH1
description: All users of branch 1
distinguishedName: CN=BRANCH1,CN=Users,DC=somedomain,DC=local
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: group
name: BRANCH1
sAMAccountName: BRANCH1
```

GROUP-DB.CSV

Lookup database for group names:

```
BOOKREAD;CN=BOOKREAD,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
BOOKWRITE;CN=BOOKWRITE,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
PRINTER;CN=PRINTER,OU=Print,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
TRANSITION;CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
```

BRANCH1;CN=BRANCH1,OU=Organization,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local

Migrating users

Here you find import files and protocol files used in 4.5, “Migrating users” on page 113.

CreateUser.vbs

Script using ADSI for import of user definitions:

```
' *****  
' File      : CreateUser.vbs  
' Version  : 2.0  
' Date     : 06/06/03  
' Author   : Leif Braeuer (6PAC Consulting AG)  
'  
' Description:  
' Processes a user.csv file from lsmt to create users  
'  
' *****  
  
'*****  
' Format of input file as exported from lsmt  
' See lsmt documentation for description of attributes  
'  
'      Field          Column  
const OS2_OPT          = 1  
const OS2_NAME         = 2  
const OS2_PASSWORD    = 3  
const OS2_PASSWORD_AGE = 4  
const OS2_PRIV        = 5  
const OS2_HOME_DIR    = 6  
const OS2_COMMENT     = 7  
const OS2_FLAGS       = 8  
const OS2_SCRIPT_PATH = 9  
const OS2_AUTH_FLAGS  = 10  
const OS2_FULL_NAME   = 11  
const OS2_USR_COMMENT = 12  
const OS2_PARMS       = 13  
const OS2_WORKSTATIONS = 14  
const OS2_LAST_LOGON  = 15  
const OS2_LAST_LOGOFF = 16  
const OS2_ACCT_EXPIRES = 17  
const OS2_MAX_STORAGE = 18  
const OS2_RESTRICTED_HOURS = 19  
const OS2_1_LOGON_HOURS = 20
```



```

const OS2_2_LOGON_HOURS    = 21
const OS2_3_LOGON_HOURS    = 22
const OS2_4_LOGON_HOURS    = 23
const OS2_5_LOGON_HOURS    = 24
const OS2_6_LOGON_HOURS    = 25
const OS2_7_LOGON_HOURS    = 26
const OS2_BAD_PW_COUNT     = 27
const OS2_NUM_LOGONS       = 28
const OS2_LOGON_SERVER     = 29
const OS2_COUNTRY_CODE     = 30
const OS2_CODE_PAGE        = 31

Const UF_DONT_EXPIRE_PASSWD = &H00010000
Const UF_ACCOUNTDISABLE     = &H00000002
Const UF_NORMAL_ACCOUNT     = &H00000200

Set WshShell = wscript.CreateObject("WScript.Shell")
Set objArgs = WScript.Arguments
if objArgs.Count <> 2 Then
    Wscript.echo "Missing or wrong arguments."
    wscript.quit
end if

sFile      = objArgs(0)           ' Input file
sDomain    = objArgs(1)         ' Branch/domain name
sDC        = "DC=somedomain2,DC=local"
sBaseOU    = "OU=Users,OU=" & sDomain & ",OU=Branch," & sDC
sDNSname   = "somedomain2.local"

ADS_GRP_USERS   = "CN=Domain Users,CN=Users," & sDC
ADS_GRP_GUESTS  = "CN=Domain Guests,CN=Users," & sDC
ADS_GRP_ADMINS  = "CN=Domain Admins,CN=Users," & sDC
ADS_GRP_PRINT   = "CN=Print Operators,CN=Builtin," & sDC
ADS_GRP_SERVER  = "CN=Server Operators,CN=Builtin," & sDC
ADS_GRP_ACCOUNT = "CN=Account Operators,CN=Builtin," & sDC

const iNumAttributes=31
dim Attribute(31)

'Create a filesystem object
set objFileSystem = CreateObject("Scripting.FileSystemObject")
set objInputFile = objFileSystem.OpenTextFile(sFile)

'Read the input file
i = 0
wscript.echo "Creating objects in LDAP://" & sDNSname & "/" & sBaseOU & "..."
While not objInputFile.AtEndOfStream

```

```

sInput = Trim(objInputFile.ReadLine)
i=i+1
if i>1 Then
    ParseInputFile sInput
    if InStr(UCCase(Attribute(OS2_OPT)),"A") > 0 Then Call CreateUser
end if
Wend

objInputFile.Close
wscript.quit()

'*****
'* Parse input file
'*****
Function ParseInputFile(sIn)
    iBlock=0
    iString=0

    ' Cleanup array
    do while iString<iNumAttributes
        Attribute(iString)=" "
        iString=iString+1
    loop

    iString=0
    iPos=1
    do while(iPos>0 AND iString<iNumAttributes)
        iString=iString+1
        iPos=Instr(iBlock+1,sIn,";")
        if iPos>0 then Attribute(iString) = trim(mid(sIn,iBlock+1,iPos-iBlock-1))
        if iPos=0 then Attribute(iString) = trim(mid(sIn,iBlock+1))
        iBlock=iPos
    loop
end Function

'*****
'* Create user using values from input file
'*****
Sub CreateUser
    ON ERROR RESUME NEXT
    wscript.echo "Processing " & Attribute(OS2_NAME) & "..."

    '* open organizationalUnit
    Set objOU = GetObject("LDAP://" & sDNSname & "/" & sBaseOU)
    If Err.Number Then
        wscript.echo "Error in opening organizationalUnit."
    Exit Sub

```

```

End If

'* Create user object
Set objUsr = objOU.Create("user", "CN=" & Attribute(OS2_NAME))
If Err.Number>0 Then
    wscript.echo "Error creating user."
    Exit Sub
Else

    objUsr.Put "sn", Mid(Attribute(OS2_COMMENT), InStr(Attribute(OS2_COMMENT),
    "_")+1)
    objUsr.Put "givenName", Mid(Attribute(OS2_COMMENT), 1,
    InStr(Attribute(OS2_COMMENT), "_")-1)
    objUsr.Put "displayName", Attribute(OS2_NAME)
    objUsr.Put "description", Attribute(OS2_USR_COMMENT)
    objUsr.Put "userPrincipalName", Attribute(OS2_NAME) & "@ " & sDNSname

    objUsr.put "pwdLastSet", CLng(0)
    objUsr.Put "samAccountName", Attribute(OS2_NAME)

    if Attribute(OS2_MAX_STORAGE) <> "No Limit" Then
        objUsr.Put "maxStorage", CInt(Attribute(OS2_MAX_STORAGE))
    end if
    objUsr.Put "codePage", CInt(Attribute(OS2_CODE_PAGE))
    objUsr.Put "countryCode", CInt(Attribute(OS2_COUNTRY_CODE))

    if Attribute(OS2_WORKSTATIONS) <> "No Restriction" Then
        objUsr.put "userWorkstations", Replace(Attribute(OS2_WORKSTATIONS), " ",
        ",")
    End if

    objUsr.Put "scriptPath", "logon.cmd"

    if Attribute(OS2_HOME_DIR) <> "-none-" Then
        objUsr.put "homeDrive", Left(Attribute(OS2_HOME_DIR),1)
        objUsr.put "homeDirectory", "\\ " & Mid(Attribute(OS2_HOME_DIR), 4,
        InStr(4,Attribute(OS2_HOME_DIR),"\")-4) & _
        "\ " & Attribute(OS2_NAME)
    End if

    if (Attribute(OS2_ACCT_EXPIRES) <> "(null)") And
    (Attribute(OS2_ACCT_EXPIRES) <> "Unknown") Then
        objUsr.accountExpirationDate = ParseDate(Attribute(OS2_ACCT_EXPIRES)) + 1
    End if

    objUsr.SetInfo

    Set objUsr = GetObject("LDAP://" & sDNSname & "/CN=" & Attribute(OS2_NAME)
    & ", " & sBaseOU)

```

```

select case Attribute(OS2_PRIV)
  case "User"
    Add2Group objUsr.distinguishedName, ADS_GRP_USERS
    objUsr.put "primaryGroupID", 513
  case "Guest"
    Add2Group objUsr.distinguishedName, ADS_GRP_GUESTS
    objUsr.put "primaryGroupID", 514
  case "Administrator"
    Add2Group objUsr.distinguishedName, ADS_GRP_ADMINS
    objUsr.put "primaryGroupID", 512
end select

  if InStr(Attribute(OS2_AUTH_FLAGS), "P" ) > 0 Then Add2Group
objUsr.distinguishedName, ADS_GRP_PRINT
  if InStr(Attribute(OS2_AUTH_FLAGS), "A" ) > 0 Then Add2Group
objUsr.distinguishedName, ADS_GRP_ACCOUNT
  if InStr(Attribute(OS2_AUTH_FLAGS), "S" ) > 0 Then Add2Group
objUsr.distinguishedName, ADS_GRP_SERVER
  if InStr(Attribute(OS2_AUTH_FLAGS), "C" ) > 0 Then WScript.Echo " >
COMM_OP_PRIV is not supported."

  'Change Logon Hours Attribute(OS2_RESTRICTED_HOURS)

  objUsr.userAccountControl = UF_NORMAL_ACCOUNT
  if InStr(Attribute(OS2_FLAGS), "D" ) > 0 Then objUsr.userAccountControl =
objUsr.userAccountControl + UF_ACCOUNTDISABLE
  if InStr(Attribute(OS2_FLAGS), "U" ) > 0 Then WScript.Echo " > FLAG :
CANNOT_DEL is not supported."
  if InStr(Attribute(OS2_FLAGS), "N" ) > 0 Then WScript.Echo " > FLAG :
PWD_NOT_REQ is not supported."
  if InStr(Attribute(OS2_FLAGS), "C" ) > 0 Then WScript.Echo " > FLAG :
CANNOT_CHANGE_PWD is not supported."

  objUsr.SetInfo

  wscript.echo " > Done."
End If
End Sub

Function ParseDate( sDateStr )
  ParseDate = CDate(Mid(sDateStr, 5, 6) & "," & Mid(sDateStr, 21, 4))
End Function

Sub Add2Group( sUser, sGroup )
  ON ERROR RESUME NEXT
  Set objGrp = GetObject("LDAP://" & sDNSname & "/" & sGroup)
  objGrp.Add("LDAP://" & sUser)

```

```
Set objGrp = nothing
End Sub
```

USERS.LDIF

Import file for user definitions:

```
dn: CN=ANDREI,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: ANDREI
distinguishedName:
CN=ANDREI,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: user
givenName: Andrei
sn: Vlad
displayName: ANDREI
name: ANDREI
userPrincipalName: ANDREI@somedomain.local
pwdLastSet: 0
sAMAccountName: ANDREI
codePage: 0
countryCode: 0
logonHours:: ///////////////////////////////////////////////////
userAccountControl: 512
scriptPath: logon.cmd
homeDrive: U
homeDirectory: \\PDC\ANDREI
```

```
dn: CN=ANDREI,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: primaryGroupID
primaryGroupID: 513
-
```

```
dn: CN=LEIF,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: LEIF
distinguishedName: CN=LEIF,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: user
givenName: Leif
sn: Braeuer
displayName: LEIF
name: LEIF
userPrincipalName: LEIF@somedomain.local
pwdLastSet: 0
sAMAccountName: LEIF
codePage: 0
```

```

countryCode: 0
logonHours:: ////////////////////////////////////////////////////
userAccountControl: 512
scriptPath: logon.cmd
homeDrive: U
homeDirectory: \\PDC\LEIF

dn: CN=LEIF,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: primaryGroupID
primaryGroupID: 513
-

dn: CN=MARC,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: MARC
distinguishedName: CN=MARC,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: user
givenName: Marc
sn: Schneider
displayName: MARC
name: MARC
userPrincipalName: MARC@somedomain.local
pwdLastSet: 0
sAMAccountName: MARC
codePage: 0
countryCode: 0
logonHours:: ////////////////////////////////////////////////////
userAccountControl: 512
scriptPath: logon.cmd
homeDrive: U
homeDirectory: \\PDC\MARC

dn: CN=MARC,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: primaryGroupID
primaryGroupID: 513
-

dn: CN=OLIVER,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: OLIVER
distinguishedName:
CN=OLIVER,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: user
givenName: Oliver
sn: Mark

```

displayName: OLIVER
name: OLIVER
userPrincipalName: OLIVER@somedomain.local
pwdLastSet: 0
sAMAccountName: OLIVER
codePage: 0
countryCode: 0
logonHours:: //////////////////////////////////////
userAccountControl: 512
scriptPath: logon.cmd
homeDrive: U
homeDirectory: \\PDC\OLIVER

dn: CN=OLIVER,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: primaryGroupID
primaryGroupID: 513
-

dn: CN=RICHARD,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add
cn: RICHARD
distinguishedName:
CN=RICHARD,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: user
givenName: Richard
sn: Spurlock
displayName: RICHARD
name: RICHARD
userPrincipalName: RICHARD@somedomain.local
pwdLastSet: 0
sAMAccountName: RICHARD
codePage: 0
countryCode: 0
logonHours:: //////////////////////////////////////
userAccountControl: 512
scriptPath: logon.cmd
homeDrive: U
homeDirectory: \\PDC\RICHARD

dn: CN=RICHARD,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: primaryGroupID
primaryGroupID: 513
-

dn: CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: add

cn: WYNAND
distinguishedName:
CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=somedomain,DC=local
objectClass: user
givenName: Wynand
sn: Pretorius
displayName: WYNAND
name: WYNAND
userPrincipalName: WYNAND@somedomain.local
description: Standard Bank User
pwdLastSet: 0
sAMAccountName: WYNAND
codePage: 0
countryCode: 0
logonHours:: AAAAAf/gAf/gAf/gAf/gAf/gAAAA
userAccountControl: 512
userWorkstations: PC1,PC2
scriptPath: logon.cmd
homeDrive: U
homeDirectory: \\PDC\WYNAND

dn: CN=Print Operators,CN=Builtin,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
-

dn: CN=Account Operators,CN=Builtin,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
-

dn: CN=Server Operators,CN=Builtin,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
-

dn: CN=WYNAND,OU=Users,OU=branch1,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: primaryGroupID
primaryGroupID: 513
-

GRPMEM.LDIF

Import file for membership definitions:

```
dn: CN=BOOKREAD,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=ANDREI,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=ANDREI,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=BRANCH1,OU=Organization,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=ANDREI,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=BOOKREAD,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=LEIF,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=PRINTER,OU=Print,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=LEIF,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=LEIF,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=BRANCH1,OU=Organization,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=LEIF,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=BOOKREAD,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=MARC,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=PRINTER,OU=Print,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=MARC,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
```

```

dn: CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=MARC,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=BRANCH1,OU=Organization,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=MARC,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=BOOKWRITE,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=OLIVER,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=PRINTER,OU=Print,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=OLIVER,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=OLIVER,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=BRANCH1,OU=Organization,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=OLIVER,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=BOOKREAD,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=RICHARD,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=RICHARD,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=BRANCH1,OU=Organization,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=RICHARD,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=BOOKREAD,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=WYNAND,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local

```

```

-
dn: CN=TRANSITION,OU=Access,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=WYNAND,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-
dn: CN=BRANCH1,OU=Organization,OU=Groups,OU=,OU=Branch,DC=somedomain,DC=local
changetype: modify
add: member
member: CN=WYNAND,OU=Users,OU=Branch1,OU=Branch,DC=somedomain,DC=local
-

```

LOGON.CMD

Global logon script:

```

@ECHO OFF
REM *****
REM File      : LOGON.CMD
REM Version   : 2.0
REM Date      : 06/06/03
REM Author    : Leif Braeuer (6PAC Consulting AG)
REM
REM Description:
REM Central logon script for OS/2, Windows NT and Windows 2000 clients
REM
REM Other operating systems like Windows 9x etc. are not supported
REM
REM *****

ECHO Please wait while logon script is executed...

REM
*****
REM Detect client operating system
REM
CALL %0\..\CHECKOS.CMD

REM
*****
REM Add some environment variables not available in OS2
REM
IF "%SIXPAC.OS%"=="OS2" CALL %0\..\OS2\OS2ENV.CMD %0

REM
*****
REM Synchronize time
REM
NET TIME %LOGONSERVER% /SET /Y 1>NUL 2>NUL

```

```

REM
*****
REM User specific script with logon assignments
REM
REM
IF NOT "%SIXPAC.OS%"=="OS2" NET USE /persistent:no >NUL
IF EXIST %0\..\USERS\%USERNAME%.CMD CALL %0\..\USERS\%USERNAME%.CMD

REM
*****
REM Jump to operating system specific part
REM
GOTO %SIXPAC.OS%
GOTO END

REM
*****
REM
*****
REM Windows 2000 / Windows XP specific part
REM
:w2k
    ECHO Windows 2000 or Windows XP detected...
GOTO END

REM
*****
REM Windows NT 4.0 specific part
REM
:NT4
    ECHO Windows NT 4.0 detected...
GOTO END

REM
*****
REM IBM OS/2 specific part
REM
:OS2
    ECHO IBM OS/2 detected...
GOTO END

REM
*****
REM Unknown operating system
REM
:UNK
    ECHO.
    ECHO Cannot detect operating system. Please call your local support.

```

```
ECHO.  
PAUSE  
GOTO END
```

```
:END
```

CHECKOS.CMD

Script to detect OS:

```
@ECHO OFF  
REM *****  
REM File      : CHECKOS.CMD  
REM Version  : 2.0  
REM Date     : 06/06/03  
REM Author  : Leif Braeuer (SIXPAC Consulting AG)  
REM  
REM Description:  
REM Script used to detect the clients operating system and environment  
REM The result is available in the variable SIXPAC.OS  
REM  
REM OS2 - OS/2 System is assumed  
REM NT4 - Windows NT Version 4.0  
REM W2K - Windows 2000 (Version 5.0) and Windows XP (Version 5.1)  
REM UNK - Unknown operating system  
REM  
REM Other operating systems like Windows 9x etc. are not detected  
REM  
REM *****  
  
SET SIXPAC.OS=UNK  
IF %OS%==__ GOTO NO_WINDOWS  
VER | FIND /i "5.1" >NUL  
IF %ERRORLEVEL%==1 GOTO NOT_XP  
SET SIXPAC.OS=W2K  
GOTO END_OSCHECK  
:NOT_XP  
VER | FIND /i "5.0" >NUL  
IF %ERRORLEVEL%==1 GOTO NOT_W2K  
SET SIXPAC.OS=W2K  
GOTO END_OSCHECK  
:NOT_W2K  
VER | FIND /i "4.0" >NUL  
IF %ERRORLEVEL%==1 GOTO END_OSCHECK  
SET SIXPAC.OS=NT4  
GOTO END_OSCHECK  
:NOT_NT4  
:NO_WINDOWS  
VER | FIND /i "System/2" >NUL
```

```

IF ERRORLEVEL==1 GOTO END_OSCHECK
SET SIXPAC.OS=OS2
:NOT
:END_OSCHECK

```

OS2ENV.CMD

Script to add environment variables:

```

/* REXX
*****
* File      : OS2ENV.CMD
* Version   : 2.0
* Date      : 06/06/03
* Author    : Leif Braeuer (6PAC Consulting AG)
*
* Description:
* Adds the following environment variables missing in OS/2
*
*  COMPUTERNAME - With NET CONFIG inquired NetBIOS name of the client
*  LOGONSERVER  - From the current path inquired Logonserver
*  USERNAME     - With NET CONFIG inquired user name
*  USERDOMAIN  - With NET CONFIG inquired user name
*
*****
*/

'@ECHO OFF'
PARSE UPPER ARG "\\\" LogonServer "\"
CALL VALUE 'LOGONSERVER', "\\\" || LogonServer, 'OS2ENVIRONMENT'
'NET CONFIG REQ | RXQUEUE'
DO QUEUED()
  PARSE UPPER PULL line
  IF POS('MACHINE ID',line)>0 THEN CALL VALUE 'COMPUTERNAME',
SUBSTR(WORD(line,3),3), 'OS2ENVIRONMENT'
  IF POS('USER ID',line)>0 THEN CALL VALUE 'USERNAME', WORD(line,3),
'OS2ENVIRONMENT'
  IF POS('LOGON DOMAIN',line)>0 THEN CALL VALUE 'USERDOMAIN', WORD(line,3),
'OS2ENVIRONMENT'
END

```

SETWINUSERASN.CMD

Script to generate user specific logon scripts:

```

/* REXX
*****
* File      : SETWINUSERASN.CMD
* Version   : 2.0

```

```

* Date      : 06/06/03
* Author    : Leif Braeuer (6PAC Consulting AG)
*
* Description:
* Get user account information from OS/2 domain controller to
* build a batch file that can be executed by the Windows 2000
* domain controller when the user logs on assigns resources on
* identical drive letters/ports the user gets in the OS/2 environment
*
*****
*/

'@ECHO OFF'

call RxFuncAdd 'LoadLsRxutFuncs', 'LSRXUT', 'LoadLsRxutFuncs'
call LoadLsRxutFuncs

PARSE Arg TargetPath

NETUSER = 280
myRc = NetEnumerate(NETUSER, 'userInfo', '')
DO j=1 TO userInfo.0
    UserId=userInfo.j
    SAY "Processing " || UserId || "..."
    CALL GenerateBatch
END

CALL DropLsRxutFuncs
CALL RxFuncDrop 'LoadLsRxutFuncs'

EXIT 0

/* -----*/
GenerateBatch:

    NETUSER      = 280
    myRc = NetGetInfo(NETUSER, 'userInfo', '', UserId)

    NETLOGONASN = 52
    myRc = NetGetInfo(NETLOGONASN, 'logonAsnInfo', '', UserId)

    if myRc=0 then do
        CmdFile= TargetPath || '\'||UserId||'.CMD'
        'DEL ' || CmdFile || ' 1>NUL 2>NUL'
        CALL LineOut CmdFile, "@ECHO OFF"
        CALL LineOut CmdFile, "REM
*****
        CALL LineOut CmdFile, "REM File      : " || UserId || ".CMD"
        CALL LineOut CmdFile, "REM Version : 2.0"

```

```

CALL LineOut CmdFile, "REM Date   : " || Date()
CALL LineOut CmdFile, "REM Author : Leif Braeuer (6PAC Consulting AG)"
CALL LineOut CmdFile, "REM"
CALL LineOut CmdFile, "REM Description:"
CALL LineOut CmdFile, "REM User specific logon script of logon assignments"
CALL LineOut CmdFile, "REM"
CALL LineOut CmdFile, "REM
*****"
CALL LineOut CmdFile, ""
CALL LineOut CmdFile, ":START_FILENETUSE"
/* Get the user logon assignments information */
DO i=1 TO logonAsnInfo.count
  IF logonAsnInfo.i.type="Files alias" THEN DO
    CALL Lineout cmdFile, " NET USE " || logonAsnInfo.i.device || ": " ||
Alias2UNC()
    END
  END
  call Lineout CmdFile, " NET USE " || LEFT(userInfo.HOME_DIR,2) || " \\" ||
WORD(TRANSLATE(userInfo.HOME_DIR,"","\\"),2) || "\" || userId

  CALL LineOut CmdFile, ":END_FILENETUSE"
CALL LineOut CmdFile, ""
CALL LineOut CmdFile, ":START_PRINTNETUSE"
DO i=1 TO logonAsnInfo.count
  IF logonAsnInfo.i.type="Printer alias" THEN DO
    CALL Lineout cmdFile, " NET USE " || logonAsnInfo.i.device || ": " ||
Alias2UNC()
    END
  END
  CALL LineOut CmdFile, ":END_PRINTNETUSE"
CALL LineOut CmdFile, ""
Rc = Stream(CmdFile, 'c', 'close')
end
RETURN

/* -----*/
Alias2UNC:
  NETALIAS = 20
  MyRc = NetGetInfo(NETALIAS, 'AliasInfo', '', logonAsnInfo.i.alias)
RETURN "\\" || aliasInfo.server || "\" || aliasInfo.netname

```

WYNAND.CMD

Example of user logon script:

```

@ECHO OFF
REM *****
REM File   : WYNAND.CMD
REM Version : 2.0

```



```

REM Date      : 29 Jun 2003
REM Author   : Leif Braeuer (6PAC Consulting AG)
REM
REM Description:
REM User specific logon script of logon assignments
REM
REM *****
:START_FILENETUSE
  NET USE L: \\BDC\LANSHARE
  NET USE U: \\PDC\WYNAND
:END_FILENETUSE

:START_PRINTNETUSE
:END_PRINTNETUSE

```

Migrating directories

Here you find import files and protocol files used in 4.5, “Migrating users” on page 113.

GETWINACL.CMD

Script to retrieve ACL on OS/2 servers:

```

/* Get a access control profile for a drive tree */

call RxFuncAdd 'LoadLsRxutFuncs', 'LSRXUT', 'LoadLsRxutFuncs'
call LoadLsRxutFuncs
call RxFuncAdd 'SysLoadFuncs', 'REXXUTIL', 'SysLoadFuncs'
call SysLoadFuncs

Parse Arg outFile basePath

basePath = Strip(basePath)
outFile = Strip(outFile)

'@del 'outfile' 1>NUL 2>NUL'

if LENGTH(basePath)<3 Then basePath=basePath"\
rc = NetGetInfo( 10, 'AccPerm', '', basePath)
if rc <> 0 Then strAcl = ""
else strAcl = FormatACL()
call Lineout outFile, basePath || ";" || strAcl

Call RecurseDir basePath, strAcl

```

```

call DropLsRxutFuncs
call RxFuncDrop 'LoadLsRxutFuncs'
exit

RecurseDir: procedure expose outFile
  PARSE ARG baseDir, strACL
  Say baseDir
  baseDir = STRIP(baseDir,"T","\")
  CALL SysFileTree baseDir || '\*', 'dir.', 'DO'
  DO i = 1 TO dir.0
    rc = NetGetInfo( 10, 'AccPerm', '', dir.i)
    if rc <> 0 Then subAc1 = ""
    else subAc1 = FormatACL()
    if subAc1 <> strAc1 Then call Lineout outFile, dir.i || ";" || subAc1
    CALL RecurseDir dir.i, subAc1
  END
RETURN

FormatACL:
  ac1 = ""
  do fi=1 to AccPerm.count-1
    do fj=fi to AccPerm.count-1
      fk=fj+1
      if AccPerm.fj.ugname > AccPerm.fk.ugname then do
        tempVar = AccPerm.fk.ugname
        AccPerm.fk.ugname = AccPerm.fj.ugname
        AccPerm.fj.ugname = tempvar
        tempVar = AccPerm.fk.access
        AccPerm.fk.access = AccPerm.fj.access
        AccPerm.fj.access = tempvar
      end
    end
  end
  do k=1 to AccPerm.count
    ac1 = ac1 || AccPerm.k.ugname || ":" || AccPerm.k.access || ";"
  end
  return ac1

```

SETWINACL.CMD

Script to prepare import of ACL in Windows:

```

/* */
Parse Arg inFile outFile

defaultAc1 = "Administrators:F SYSTEM:F"

inFile = Strip(inFile)

```

```

outFile = Strip(outFile)

'@del 'outFile' 1>NUL 2>NUL'

Do While Lines(inFile)
  curLine = LineIN(inFile)
  if curLine = '' | Left(Strip(Opt),1) = '*' Then Iterate
  else do
    Parse value curLine With strPath ';' curLine
    i = 0
    strAcl = defaultAcl || " "
    Do While curLine <> ''
      i = i + 1
      Parse value curLine With actValue ';' curLine
      strAcl = strAcl || FormatNTAcl( actValue )
    End
    CALL LineOut outFile, "md " || strPath
    CALL LineOut outFile, "echo y|cacls " || strPath || " /g " || strAcl
  End
End
Exit
Return

/* -----*/
FormatNTAcl:
  PARSE ARG userid:"ace
  ace = Strip(ace,"T","G")
  select
    when userid = "USERS" Then userid = "Domain Users"
    when userid = "ADMINS" Then userid = "Domain Admins"
    when userid = "GUESTS" Then userid = "Domain Guests"
  otherwise nop
end
select
  when ace = "RWCXDAP" Then ace = "F"
  when ace = "R" Then ace = "R"
  when ace = "RX" Then ace = "R"
  when ace = "RWCXDA" Then ace = "C"
  otherwise nop
end
ace = '%"USERDOMAIN%\ ' || userId || ':' || ace || ' '
Return ace

```

SETWINSHARE.CMD

Script to prepare share definitions in Windows:

```
/* */
```

```

Parse Arg inFile outFileDir outFilePrt

inFile = Strip(inFile)
outFileDir = Strip(outFileDir)
outFilePrt = Strip(outFilePrt)

'@del 'outFileDir' 1>NUL 2>NUL'
'@del 'outFilePrt' 1>NUL 2>NUL'

Do While Lines(inFile)
  curLine = LineIN(inFile)
  orgLine = curLine
  Parse Value curLine With Opt ';' curLine
  Select
    When Opt = '' | curLine = '' | Left(Strip(Opt),1) = '*' Then Iterate
    When Translate(Opt) = 'OPT' Then Call GetColumns
    When Translate(Opt) = 'A' Then Call AddShare
    Otherwise Iterate
  End
End
Exit ExitCode
Return

/* -----*/
AddShare:
  i = 0
  Do While curLine <> ''
    i = i + 1
    columnName = Strip(columnNames.i)
    Parse value curLine With actValue ';' curLine
    share.columnName = Strip(actValue)
    If (share.columnName = "Unknown") | (share.columnName = "(null)") Then
share.columnName = ''
    End
    Call CreateCMD
  Return

/* -----*/
GetColumns:
  i = 0
  Do While curLine <> ''
    i = i + 1
    Parse value curLine With columnNames.i ';' curLine
  End
  numColumn = i
Return

```

```

/* -----*/
CreateCmd:

select
  when share.TYPE = 'Files' Then Do
    optional = ""
    if share.REMARK <> "" Then optional = optional || "/remark:" ||
share.REMARK || " "
    if share.MAXUSES <> 65535 Then optional = optional || "/users:" ||
share.MAXUSES || " "
    CALL LineOut outFileDir, "rmtshare \\\" || share.SERVER || "\" ||
share.NETNAME || "=" || share.PATH || optional
    end
  when share.TYPE = 'Printer' Then Do
    if share.REMARK <> "" Then optional = optional || '/remark:' ||
share.REMARK || ' '
    if share.MAXUSES <> 65535 Then optional = optional || "/users:" ||
share.MAXUSES || " "
    CALL LineOut outFilePrt, "rmtshare \\\" || share.SERVER || "\" ||
share.NETNAME || "=" || share.QUEUE || " /printer " || optional
    end
  otherwise SAY share.NAME || ' skipped. Target does not support type ' ||
share.TYPE
end

Return

```

SETWINCOPY.CMD

Script to prepare data migration to Windows:

```

/* */
Parse Arg inFile

inFile = Strip(inFile)

outFileDir = "rcopy.cmd"

'@del 'outFileDir' 1>NUL 2>NUL'

Do While Lines(inFile)
  curLine = LineIN(inFile)
  orgLine = curLine
  Parse Value curLine With Opt ';' curLine
  Select
    When Opt = '' | curLine = '' | Left(Strip(Opt),1) = '*' Then Iterate
    When Translate(Opt) = 'OPT' Then Call GetColumns
    When Translate(Opt) = 'A' Then Call AddShare
    Otherwise Iterate

```

```

        End
    End
    Exit ExitCode
Return

/* -----*/
AddShare:
    i = 0
    Do While curLine <> ''
        i = i + 1
        columnName = Strip(columnNames.i)
        Parse value curLine With actValue ';' curLine
        share.columnName = Strip(actValue)
        If (share.columnName = "Unknown") | (share.columnName = "(null)") Then
share.columnName = ''
        End
        Call CreateCMD
    Return

/* -----*/
GetColumns:
    i = 0
    Do While curLine <> ''
        i = i + 1
        Parse value curLine With columnNames.i ';' curLine
        End
    numColumn = i
Return

/* -----*/
CreateCmd:
    if share.TYPE = 'Files' Then Do
        CALL LineOut outFileDir, "robocopy \\OS2." || share.SERVER || "\" ||
share.NETNAME || " \\WIN." || share.SERVER || "\" || share.NETNAME || " /mir /z
/r:3 /w:30 /np /log+:rcopy.log"
        end
    Return

```



B

REXX source code

This appendix contains all scripts and input files that are used on the OS/2 Server as part of the information extraction or manipulation.

Global source code and input files

The following are REXX routines that are called by many of the other routines in this appendix.

RGUTIL.CMD

The command file is part of the LSMT package. Each of the GET*.CMD programs provided later, call the RGUTIL.CMD to load REXX functions provided by the REXXUTIL.DLL.

Usage

Call RgUtil /M

The /M is to suppress all information.

Source code

Example 9-15 RGUTIL.CMD source code

```
/*-----*\
Register all the REXXUTIL.DLL Functions      (C) IBM
Written by Alain Rykaert , IBM Belgium
\*-----*/

Parse Upper Arg Option .

if Strip(Translate(Option)) = '/M'
then MUTE = 1
else MUTE = 0

Result_Query=RxFuncQuery('SysLoadFuncs')
if Result_Query = 0
then do
    if \MUTE then Say '*** OK, REXXUTIL was registered. ***'
end
else do
    Result_Add = RxFuncAdd( 'SysLoadFuncs', 'RexxUtil',
'SysLoadFuncs')
    if Result_Add = 0
    then do
        if \MUTE then Say '*** OK, REXXUTIL is registered. ***'
        Signal ON Syntax Name Load_Check
        Call SysLoadFuncs
        Load_Check: /* RC of 43 means REXXUTILs not found */
        if RC = 43
```



```

                                then do
                                    Say '*** ERROR: Not able to load the RexxUtils.
                                ,
                                    Say '    Perhaps REXX not installed, or
REXXUTIL.DLL not found in a LIBPATH drive/directory.'
                                    '@Pause'
                                end
                                else Nop
                                Signal OFF Syntax
                                end
                            else do
                                Say '*** ERROR: RexxUtil registration has failed. ***'
                                '@Pause'
                                end
                            end
                        end

                    if \MUTE then say '0A0D'X ' OS/2 version' SYSOS2VER()

                    Exit

/*-----*/

```

RGUTILS.CMD

The command file is part of the LSMT package. Each of the GET*.CMD programs provided later, call the RGUTILS.CMD to load Rexx functions provided by the RXUTILS.DLL.

Usage

Call RgUtils /M

The /M is to suppress all information.

Source Code

Add text here (Body0).

Example 9-16 RGUTILS.CMD source code

```

/*-----*\
Register all the RXUTILS.DLL Functions      (C) IBM
Written by Alain Rykaert , IBM Belgium
\*-----*/

```

Parse Upper Arg Option .

```

if Strip(Translate(Option)) = '/M'
then MUTE = 1
else MUTE = 0

Result_Query=RxFuncQuery('RxLoadFuncs')
if Result_Query = 0
then do
    if \MUTE then Say '*** OK, RxUtils was registered. ***'
    end
else do
    Result_Add = RxFuncAdd('RxLoadFuncs','RXUTILS','RxLoadFuncs')
    if Result_Add = 0
    then do
        if \MUTE then Say '*** OK, RxUtils is registered. ***'
        Signal ON Syntax Name Load_Check
        Call RxLoadFuncs
        Load_Check: /* RC of 43 means RXUTILS not found */
        if RC = 43
        then do
            Say '*** ERROR: Not able to load the RxUtils.'
            Say '    Perhaps REXX not installed,'
            Say '    or RXUTILS.DLL not found in a LIBPATH
drive/directory.'

                '@Pause'
            end
            Signal OFF Syntax
        end
    else do
        Say '*** ERROR: RxUtils registration has failed. ***'
        '@Pause'
    end
end

end

if \MUTE then say '0A0D'X ' RXUTILS version' RXUTILSVER()

Exit

/*-----*/

```

RGLSRXUT.CMD

This file is part of the LSMT package. The **GET*.CMD** commands described later, call the RGLSRXUT.CMD to load LAN Server utility function. If the functions fail to load, the code will verify if the DLL is installed and copy the correct DLL.

Usage

Call RGLSRXUT /M

The /M is to suppress all information.

Source code

Example 9-17 RGLSRXUT.CMD source code

```
/*-----*\
Register all the LSRXUT.DLL Functions      (C) IBM
Written by Alain Rykaert , IBM Belgium
\*-----*/

Parse Upper Arg Option .

Call ChkD11                               /* Check for correct DLL*/

If Strip(Translate(Option)) = '/M'
  Then Mute = 1
  Else Mute = 0

Result_Query = RxFuncQuery('LoadLSRXUTFuncs')
If Result_Query = 0
  Then Do
    If \Mute Then Say '*** OK, LSRXUT was registered. ***'
  End
  Else Do
    Result_Add = RxFuncAdd('LoadLsRxutFuncs', 'LSRXUT',
'LoadLsRxutFuncs')
    If Result_Add = 0
      Then Do
        If \MUTE Then Say '*** OK, LSRXUT is registered. ***'
        Signal On Syntax Name Load_Check
        Call LoadLsRxutFuncs
        Load_Check:          /* RC of 43 means LSRXUT not found*/
        If RC = 43
          Then Do
            Say '*** ERROR: Not able to load the LSRXUT. '
            Say '   Perhaps REXX not installed'
            say '   or LSRXUT.DLL not found in a LIBPATH
drive/directory.'

            '@Pause'
          End
        Else Nop
        Signal Off Syntax
      End
    Else Do
      Say '*** ERROR: LSRXUT registration has failed. ***'
```

```

        '@Pause'
    End
End

Call RxFuncAdd 'WfrxLoadFuncs', 'WfrxUtil', 'WfrxLoadFuncs'
Call WfrxLoadFuncs

If \Mute Then Say 'OAOD'X ' LSRXUT.DLL Version' LsRxUtVer()

Exit

CHKDLL:/* -----*/

LsrDrive = Left(SysSearchPath('PATH', 'NET.EXE'), 2) /* IBMLAN Drive*/

If LsrDrive = ''
Then Do
    Say ' Unable to determine the OS/2 Boot Drive'
    Say ' or the OS/2 Lan Requester Drive.'
    Exit
End

REQ_CSD = SubStr(LineIn(LsrDrive'\ibmlan\syslevel.req'),48,1)
Call Stream lsrdrive'\ibmlan\syslevel.req', 'C', 'Close'

If REQ_CSD = '7'
Then Call Compit 'lsrxut.30' LsrDrive'\ibmlan\netlib\lsrxut.dll'
Else Call Compit 'lsrxut.40' LsrDrive'\ibmlan\netlib\lsrxut.dll'

Call Compit 'wfrxutil.dll' LsrDrive'\ibmlan\netlib\wfrxutil.dll'

Return

COMPIT:/* -----*/

Parse Arg Source Target

If Source = '' | Target = ''
Then Do
    Call Beep 500,200
    Return
End

Q1 = Stream(Source, 'C', 'Query Exists')
If Q1 <> ''
Then Do
    Q1 = Stream(Source, 'C', 'Query DateTime')
    Q2 = Stream(Target, 'C', 'Query DateTime')
    If Q1 \= Q2

```



```

[0;37;44mUAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA; [0m
[0;37;44m Usage : [0;1;44;33m~[0;37;44m
/SRV:Servername_of_DC [0m
[0;37;44m [0m
[0;37;44m [/OUT:Output_File] [/T] [/M] [/GROUPS] [0m
[0;37;44m [/LOG:Log_File] [/PIP:Name_of_the_Pipe] [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU [0m
[WELCOMELOGO]
[2J
[0;37;44mUAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA; [0m
[0;37;44m BBBB BBBB BBBB BBBB [0m
[0;37;44m BBBB BBBB BBBB BBBB [0m
[0;37;44m BBBB BBBB BBBB BBBB [0m
[0;37;44m BBBB BBBB BBBB BBBB [0m
[0;37;44m BBBB BBBB BBBB BBBB [0m
[0;37;44m BBBB BBBB BBBB BBBB [0m
[0;37;44m BBBB BBBB BBBB BBBB [0m
[0;37;44m BBBB BBBB BBBB BBBB [0m
[0;37;44m * * * [0;1;37;44mLAN Server Management
Tools [0;37;44m * * * [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA~ [0m
[GETWELCOME]
[0;37;44m ServerName = [0;1;44;33m~[0;37;44m
[0m
[0;37;44m Output File = [0;1;44;32m~[0;37;44m
[0m
[0;37;44m LOG File = [0;1;44;32m~[0;37;44m
[0m
[0;37;44m N_Pipe Name = [0;1;44;32m~[0;37;44m
[0m
[SETWELCOME]
[0;37;44m ServerName = [0;1;44;33m~[0;37;44m
[0m
[0;37;44m Input File = [0;1;44;32m~[0;37;44m
[0m
[0;37;44m LOG File = [0;1;44;32m~[0;37;44m
[0m
[0;37;44m Check File = [0;1;44;32m~[0;37;44m
[0m
[0;37;44m N_Pipe Name = [0;1;44;32m~[0;37;44m
[0m
[GETUSERS]
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA~ [0m
[0;37;44m Dump all Users to an ASCII file [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU [0m
[GETPASSWORDS]
[0;1;33;41mUAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA; [0m

```

```

[0;1;33;41m   If needed, Enter GETPWD to dump all passwords
[0m
[0;1;33;41mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAUU [0m
[GETPWD]
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [0m
[0;37;44m   Dump all PASSWORDS to an ASCII file [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU [0m
[GETGRPS1]
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [0m
[0;37;44m   Dump all Groups to an ASCII file [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU [0m
[GETGRPS2]
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [0m
[0;37;44m   Dump all Groups & Members to an ASCII file [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU [0m
[GETALIAS]
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [0m
[0;37;44m   Dump all Alias to an ASCII file [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU [0m
[GETAPPL]
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [0m
[0;37;44m   Dump all Applications to an ASCII file [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU [0m
[GETSEL]
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [0m
[0;37;44m   Dump all Selectors to an ASCII file [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU [0m
[GETASSGN]
[0;37;44m Show all GROUPS = [0;1;44;32m~[0;37;44m
[0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [0m
[0;37;44m   Dump all Logon Assignments to an ASCII file [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU [0m
[SETASSGN]
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [0m
[0;37;44m   Set all logon Assignments from an ASCII file [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU [0m
[GETACL]
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [0m
[0;37;44m   Dump all ACLs for all Aliases to an ASCII file [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU [0m
[SETACL]
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [0m
[0;37;44m   Set all access profiles from an ASCII file [0m
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAU [0m
[GETSERVERS]
[0;37;44mAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [0m
[0;37;44m   Dump all Servers to an ASCII file [0m

```



```
ADMINS 1000
BOOKREAD 1001
BOOKWRITE 1002
GROUPID 1003
GUESTS 1004
LOCAL 1005
PRINTER 1006
SERVERS 1007
TRANSITION 1008
```

Source code for retrieving server information

The following REXX procedures retrieve information regarding the various servers in an OS/2 domain.

GETSRVR.CMD

The `Getsrvr.cmd` is part of the LSMT package. The output file was used in this redbook for the migration to Windows or Linux.

Usage

`C:\OS2MIG\GETSRVR.CMD /SRV:PDC /OUT:C:\OS2MIG\GETSRVR.LOG /M`

`/SRV` - The netbios name of the OS/2 domain controller.

`/OUT` - The output file that will be used later in the book.

`/M` - Suppresses logging information to the screen.

Source code

Example 9-21 GETSRVR.CMD

```
/*-----*\
|   GET all SERVERS from a LAN Server 3.0 and higher
|   and dump it to an ASCII File
|                                     (C) Alain Rykaert IBM-Belgium SEP95-MAY96
\*-----*/
```

Parse Arg Option

```
Call INIT           /* Initialisation of DLL's and other stuff*/
Call CHKOPT        /* Check Options & display Welcome*/
Call CHKPWS       /* Check the PWS & Admin name*/
```

```

Call COLUMNS                               /* Read the Columns definition file*/
Call MAIN                                   /* do the main job*/
Call QUIT                                  /* Quit*/

MAIN:/* ----- MAIN: -----*/

Call Time('R')
'if exist' OUTF 'del' OUTF

RC = NetGetInfo(340, 'SERVER', '\\SRVNAME, 'SERVERS')
if RC = 0
then do
    Call RxStemSort 'SERVER'
    Call LineOut OUTF, BANNER
    do i = 1 to SERVER.0
        if i // MAXLINES = 0
            then Call LineOut OUTF, BANNER
            else Nop
        if \MUTE
            then do
                Call SysCurState OFF
                Call SysCurPos 20,0
            end
        else Nop
        say '0909'x ESC'[K Total Servers = ' i/'Server.0 'Name ='
SERVER.i
        RC = NetGetInfo(160, 'SERVERINFO', '\\SERVER.i)
        if RC = 0
            then Call WRITEIT
            else Call LOGIT 'Get NetGetInfo', SERVER.i, RC
        end
    end
else do
    Call LOGIT 'Get Servers', SRVNAME, RC
    Call QUIT
end

if \MUTE then say '0909'x ' Total Time    =' Trunc(Time('E'),2)

Call Stream Directory()\OUTF, 'C', 'CLOSE'

Call SysSetObjectData OUTF, 'ICONFILE='Directory()\Servers.Ico'

Return

WRITEIT:/* ----- WRITEIT: -----*/

SERVERINFO.OPT = Left(' ',COLL.1,' ')      /* Column 1 must be BLANK*/

```

```

OUT = ''
do j = 1 to COLT
  COLNAME = COLN.j
  DATA.j = Left(SERVERINFO.COLNAME, COLL.j, ' ')
  OUT = OUT || DATA.j || ';'
end
Call LineOut OUTF, OUT
Call Stream OUTF, 'C', 'CLOSE'
Return

COLUMNNS:/* ----- COLUMNNS: ----*/

BANNER = ''
i = 0
do while Lines(COLF)
  LLINE = LineIn(COLF)
  if Left(LLINE, 1) = '*',
  | Strip(LLINE) = ''
  then iterate
  else Nop
  i = i + 1
  parse value LLINE with COLN ';' COLL
  COLN.i = Strip(COLN)
  COLL.i = Strip(COLL)
  BANNER = BANNER || Left(COLN.i, COLL.i, ' ') || ';'
end
COLT = i
Call Stream COLF, 'C', 'CLOSE'

Return

CHKOPT:/* ----- CHKOPT: ----*/

SRVNAME = '';
OUTF = 'SERVERS.CSV';
LOGF = 'LSMT.LOG';
PIPE = '';
TRACE = 0;
MUTE = 0;

OPTION = Translate(OPTION)
do while OPTION <> ''
  Parse value OPTION with ARGUMENT ' ' OPTION
  select
  when Left(ARGUMENT,5) = '/SRV:' then SRVNAME = Substr(ARGUMENT,6)
  when Left(ARGUMENT,5) = '/OUT:' then OUTF = Substr(ARGUMENT,6)
  when Left(ARGUMENT,5) = '/LOG:' then LOGF = Substr(ARGUMENT,6)
  when Left(ARGUMENT,5) = '/PIP:' then PIPE = Substr(ARGUMENT,6)
  when Left(ARGUMENT,2) = '/M' then MUTE = 1

```

```

        when Left(ARGUMENT,2) = '/T'      then TRACE   = 1
        otherwise Nop
    end
end

if SRVNAME = '' then signal GETHELP

if \MUTE
then do
    Topic1='GETWELCOME'
    Topic_String.Topic1.1=SRVNAME;
    Topic_String.Topic1.2=OUTF;
    Topic_String.Topic1.3=LOGF;
    Topic_String.Topic1.4=PIPE' ';
    Topic_List='WELCOMELOGO' Topic1 'GETSERVERS';
    Call GETANS
    Parse VALUE SysCurPos() With Old_R Old_C; '@Pause';
    Call SysCurPos Old_R, Old_C; say ESC'[K';
end
else do
    say 'ServerName =' SRVNAME
    say 'OutputFile =' OUTF
    say 'LogFile     =' LOGF
end
Return

CHKPWS:/* ----- CHKPWS: -----*/

RC = NetGetInfo(350, 'WKSTAINFO','')
if RC = 0
then do
    ADMNAME = WKSTAINFO.UserName
    PWSNAME = WKSTAINFO.ComputerName
end
else do
    Call LOGIT 'Get PWS Info', ,RC
    Call Quit
end
Return

INIT:/* ----- INIT: -----*/

Call RgUtil  '/m'                /* Rexx Utilities*/
Call RgUtils '/m'                /* Rexx Utilities*/
Call RgNPipes '/m'              /* Named Pipes*/
Call RgLSRXUT '/m'              /* Lan Server Rexx Utils*/

Parse Upper Source . . P_NAME
PRGN = Filespec('N', Left(P_NAME, Length(P_NAME) -4))

```

```

'@echo off'
ESC   = '1B'x
REDIR = '>NUL 2>NUL'
MAXLINES = 20           /* Number of Lines to separate with a header*/

COLF  = 'SERVERS.INI'   /* Column description file*/
Call  CHKFILE COLF

Resource_File = 'LSMT.RSC'
Call  CHKFILE Resource_File

Return

GETANS:/* ----- GETANS: ----*/

Vars_List =Ansi_Say(Resource_File Topic_List);
Parse VALUE SysCurPos() With Old_R Old_C;
Do While Vars_List <> '';
    Parse VALUE Vars_List With Topic_Id ';' Var_Id ';' Row ';' Column ';';
Color ';' Vars_List;
    Call SysCurPos Row, Column;
    Say x2c(Color) || Topic_String.Topic_Id.Var_Id || '1B'x || '[0m';
End;
Call SysCurPos Old_R, Old_C;

Return

GETHELP:/* ----- GETHELP: ----*/

if \MUTE
then do
    Topic1='GETHELP'
    Topic_String.Topic1.1=PRGN;
    Topic_List=Topic1;
    Call GETANS
end
else say 'Incorrect options.'

Call QUIT

Return

CHKFILE:/* ----- CHKFILE: ----*/

Parse Arg FILE
RC = Stream(FILE, 'C', 'QUERY EXIST')
if RC = ''
then do

```

```

        say ' File' FILE 'not found.'
        Call QUIT
    end
else Nop
Call Stream FILE, 'C', 'CLOSE'

Return

LOGIT:/* ----- LOGIT: -----*/

FUNC = ARG(1); INFO = ARG(2); RCODE = ARG(3)
RC = LLOGIT(LOGF, PIPE, ADMNAME, PRGN, FUNC, INFO, RCODE)

Return

QUIT:/* ----- QUIT: -----*/

Call LineOut 'LSMT.END', PRGN, 1
Call Stream 'LSMT.END', 'C', 'CLOSE'
Call Stream COLF, 'C', 'CLOSE'
Call Stream LOGF, 'C', 'CLOSE'
Call Stream OUTF, 'C', 'CLOSE'

Exit

/*-----*/

```

SERVER.INI

The Server.ini is part of the LSMT package. GETSRVR.COM uses the INI file to generate the output file.

Usage

None

Source code

Example 9-22 SERVER.INI

```

*****
* DO NOT CHANGE THE FIRST 2 COLUMNS ORDER
* AND DO NOT CHANGE THE COLUMNS NAMES
*

OPT           ; 3
NAME         ; 8

```



```
VERSION_MAJOR ; 13
VERSION_MINOR ; 13
TYPE          ; 8
COMMENT       ; 35
ULIST_MTIME   ; 25
GLIST_MTIME   ; 25
ALIST_MTIME   ; 16
USERS         ; 5
DISC          ; 5
ALERTS        ; 10
SECURITY      ; 10
AUDITING      ; 10
NUMADMIN      ; 10
LANMASK       ; 7
HIDDEN        ; 7
ANNOUNCE      ; 8
ANDELTA       ; 8
GUESTACCT     ; 9
USERPATH      ; 8
CHDEVS        ; 6
CHDEVQ        ; 6
CHDEVJOBS     ; 9
CONNECTIONS   ; 11
SHARES        ; 6
OPENFILES     ; 9
SESSOPENS     ; 9
SESSVCS       ; 7
SESSREQS      ; 8
OPENSEARCH    ; 10
ACTIVELOCKS   ; 11
NUMREQBUF     ; 9
SIZREQBUF     ; 9
NUMBIGBUF     ; 9
NUMFILETASKS  ; 12
ALERTSCHED    ; 10
ERRORALERT    ; 10
LOGONALERT    ; 10
ACCESSALERT   ; 11
DISKALERT     ; 9
NETIOALERT    ; 10
MAXAUDITSZ    ; 10
SRVHEURISTICS ; 22
AUDITEVENTS   ; 13
AUTOPROFILE   ; 26
AUTOPATH      ; 30
```

```

then do
  Call RxStemSort 'GROUPS'
  Call LineOut UTF, BANNER
  do i = 1 to GROUPS.0
    if i // MAXLINES = 0
      then Call LineOut UTF, BANNER
    else Nop
    if \MUTE
      then do
        Call SysCurState OFF
        Call SysCurPos 20,0
      end
    else Nop
    say '0909'x ESC'[K Total Groups = ' i/'Groups.0 GROUPS.i

    RC = NetGetInfo(70, 'GroupInfo', '\\SRVNAME, GROUPS.i)

    if RC = 0
      then Nop
    else do
      GroupInfo.Name      = GROUPS.i
      GroupInfo.Comment = ''
    end
    Call WRITEIT
  end
end
else do
  Call LOGIT 'Get Groups', SRVNAME, RC
  Return
end

if \MUTE then say '0909'x ' Total Time   =' Trunc(Time('E'),2)

Call Stream UTF, 'C', 'CLOSE'

Call SysSetObjectData UTF, 'ICONFILE='Directory()\Groups1.Ico'

Return

WRITEIT:/* ----- WRITEIT: ----*/

GROUPINFO.OPT = Left('',COLL.1,' ')          /* Column 1 must be BLANK*/
OUT = ''
do j = 1 to COLT
  COLNAME = COLN.j
  DATA.j = Left(GROUPINFO.COLNAME, COLL.j, ' ')
  OUT = OUT || DATA.j || ';'
end

```

```

Call LineOut OUTF, OUT
Call Stream OUTF, 'C', 'CLOSE'
Return

```

```

COLUMNS:/* ----- COLUMNS: ----*/

```

```

BANNER = ''
i = 0
do while Lines(COLF)
  LLINE = LineIn(COLF)
  if Left(LLINE, 1) = '*',
    | Strip(LLINE) = ''
    then iterate
    else Nop
  i = i + 1
  parse value LLINE with COLN ';' COLL
  COLN.i = Strip(COLN)
  COLL.i = Strip(COLL)
  BANNER = BANNER || Left(COLN.i, COLL.i, ' ') || ';'
end
COLT = i
Call Stream COLF, 'C', 'CLOSE'

Return

```

```

CHKOPT:/* ----- CHKOPT: ----*/

```

```

SRVNAME = ''
OUTF     = 'GROUPS1.CSV'
LOGF     = 'LSMT.LOG'
PIPE     = ''
TRACE    = 0
MUTE     = 0

OPTION = Translate(OPTION)
do while OPTION <> ''
  Parse value OPTION with ARGUMENT ' ' OPTION
  select
  when Left(ARGUMENT,5) = '/SRV:' then SRVNAME = Substr(ARGUMENT,6)
  when Left(ARGUMENT,5) = '/OUT:' then OUTF     = Substr(ARGUMENT,6)
  when Left(ARGUMENT,5) = '/LOG:' then LOGF     = Substr(ARGUMENT,6)
  when Left(ARGUMENT,5) = '/PIP:' then PIPE     = Substr(ARGUMENT,6)
  when Left(ARGUMENT,2) = '/M'      then MUTE     = 1
  when Left(ARGUMENT,2) = '/T'      then TRACE    = 1
  otherwise Nop
  end
end

if SRVNAME = '' then signal GETHELP

```

```

if \MUTE
then do
    Topic1 = 'GETWELCOME'
    Topic_String.Topic1.1 = SRVNAME
    Topic_String.Topic1.2 = OUTF
    Topic_String.Topic1.3 = LOGF
    Topic_String.Topic1.4 = PIPE' '
    Topic_List = 'WELCOMELOGO' Topic1 'GETGRPS1'
    Call GETANS
    Parse VALUE SysCurPos() With Old_R Old_C; 'Pause'
    Call SysCurPos Old_R, Old_C; say ESC'[K'
end
else do
    say 'ServerName =' SRVNAME
    say 'OutputFile =' OUTF
    say 'LogFile     =' LOGF
end
Return

CHKPWS:/* -----*/

RC = NetGetInfo(350, 'WKSTAINFO','')
if RC = 0
then do
    ADMNAME = WKSTAINFO.UserName
    PWSNAME = WKSTAINFO.ComputerName
end
else do
    Call LOGIT 'Get PWS Info', ,RC
    Call Quit
end
Return

INIT:/* -----*/

Call RgUtil  '/m'                /* REXX Utilities*/
Call RgUtils '/m'                /* REXX Utilities*/
Call RgNPipes '/m'              /* Named Pipes*/
Call RgLSRXUT '/m'              /* Lan Server REXX Utils*/

Parse Upper Source . . P_NAME
PRGN = Filespec('N', Left(P_NAME, Length(P_NAME) -4))

'@echo off'
ESC = '1B'x
REDIR = '>NUL 2>NUL'

COLF = 'GROUPS.INI'                /* Column description file*/

```

```

Call CHKFILE COLF

Resource_File = 'LSMT.RSC'
Call CHKFILE Resource_File

Return

GETANS:/* ----- GETANS: ----*/

Vars_List = Ansi_Say(Resource_File Topic_List)
Parse VALUE SysCurPos() With Old_R Old_C
Do While Vars_List <> ''
    Parse VALUE Vars_List With Topic_Id ';' Var_Id ';' Row ';' Column ';'
Color ';' Vars_List;
    Call SysCurPos Row, Column
    Say x2c(Color) || Topic_String.Topic_Id.Var_Id || '1B'x || '[0m'
End
Call SysCurPos Old_R, Old_C

Return

GETHELP:/* ----- GETHELP: ----*/

if \MUTE
then do
    Topic1 = 'GETHELP'
    Topic_String.Topic1.1 = PRGN
    Topic_List = Topic1
    Call GETANS
end
else say 'Incorrect options.'

Call QUIT

Return

CHKFILE:/* ----- CHKFILE: ----*/

Parse Arg FILE
RC = Stream(FILE, 'C', 'QUERY EXIST')
if RC = ''
then do
    say ' File' FILE 'not found.'
    Call QUIT
end
else Nop
Call Stream FILE, 'C', 'CLOSE'

Return

```

```

LOGIT:/* ----- LOGIT: -----*/

    FUNC = ARG(1); INFO = ARG(2); RCODE = ARG(3)
    RC = LLOGIT(LOGF, PIPE, ADMNAME, PRGN, FUNC, INFO, RCODE)

    Return

QUIT:/* ----- QUIT: -----*/

    Call LineOut 'LSMT.END', PRGN, 1
    Call Stream 'LSMT.END', 'C', 'CLOSE'
    Call Stream COLF, 'C', 'CLOSE'
    Call Stream LOGF, 'C', 'CLOSE'
    Call Stream OUTF, 'C', 'CLOSE'
    Exit

```

GROUPS.INI

The Users.ini is part of the LSMT package. GETGRPS1.CMD uses the INI file to create the output file.

Usage

None

Source code

Example 9-24 GROUPS.INI

```

*****
* DO NOT CHANGE THE FIRST 3 COLUMNS ORDER
* AND DO NOT CHANGE THE COLUMNS NAMES
*

OPT          ; 3
NAME         ; 15
COMMENT      ; 40

```

SETGROUPS.CMD

The SETGROUPS.CMD code was used in both the Windows and Linux migration. We have written a simplified piece of code with no error checking to assist in a migration environment.

Usage

C:\OS2MIG\SETGROUPS.CMD [WIN | SMB] [INPUT FILE] [OUTPUT FILE] [BRANCH NAME] [GROUP ID FILE]

[WIN | SMB] - To what platform are you migrating to Windows (WIN) or Linux (SMB)

[INPUT FILE] - Use the GETGRPS1.LOG from the GETGRPS1.CMD

[OUTPUT FILE] - The output file that will be used for the migration. In our case we named our output file SETGOUPS.LDIF

[BRANCH NAME] - The name of the branch that will be migrating

[GROUP ID FILE] - Only needed for Linux. This is a file with all of the groups listed in the organization with a unique number. We manually created a file call TRANSFORM.GROUPS for our migration process.

Source code

Example 9-25 SETGROUPS.CMD source code

```
/* */
call RxFuncAdd 'SysLoadFuncs', 'REXXUTIL', 'SysLoadFuncs'
call SysLoadFuncs

Parse Arg srvType inFile outFile BranchName grpFile

srvType = strip(translate(srvType))
inFile = Strip(inFile)
outFile = Strip(outFile)
BranchName = Strip(BranchName)
grpFile = Strip(grpFile)

dnsDomain = "somedomain.local"
dc = "DC=somedomain,DC=local"
baseDN = "OU=Groups,OU=" || BranchName || ",OU=Branch," || dc
dbFile = "group-db.csv"

grpApp = "OU=Application," || baseDN
grpAcc = "OU=Access," || baseDN
grpPrt = "OU=Print," || baseDN
grpOrg = "OU=Organization," || baseDN

'@del 'outFile' 1>NUL 2>NUL'

Do While Lines(inFile)
  curLine = LineIN(inFile)
```



```

orgLine = curLine
Parse Value curLine With Opt ';' curLine
Select
  When Opt = '' | curLine = '' | Left(Strip(Opt),1) = '*' Then Iterate
  When Translate(Opt) = 'OPT' Then Call GetColumns
  When Translate(Opt) = 'A' Then Call AddGroup grpAcc
  When Translate(Opt) = 'P' Then Call AddGroup grpPrt
  When Translate(Opt) = 'X' Then Call AddGroup grpApp
  When Translate(Opt) = 'O' Then Call AddGroup grpOrg
  Otherwise Iterate
End
End
Exit ExitCode
Return

/* -----*/
AddGroup:
  baseOU = Strip(Arg(1))
  i = 0
  Do While curLine <> ''
    i = i + 1
    columnName = Strip(columnNames.i)
    Parse value curLine With actValue ';' curLine
    group.columnName = Strip(actValue)
    If (group.columnName = "-none-") | (group.columnName = "No Restriction") |
(group.columnName = "Unknown") | (group.columnName = "(null)") Then
      group.columnName = ''
    End
    If srvType = 'WIN' then Call WinCreateLDIF baseOU
    If srvType = 'SMB' then Call SmbCreateLDIF baseOU
  Return

/* -----*/
GetColumns:
  i = 0
  Do While curLine <> ''
    i = i + 1
    Parse value curLine With columnNames.i ';' curLine
  End
  numColumn = i
Return

/* -----*/
WinCreateLDIF:
  Parse Value group.COMMENT With givenName '_' sn
  baseOU = Strip(Arg(1))
  Say group.Name
  Call Lineout outFile, "dn: CN=" || group.NAME || ", " || baseOU

```

```

    Call Lineout outFile, "changetype: add"
    Call Lineout outFile, "cn: " || group.NAME
    if group.COMMENT <> "" Then Call Lineout outFile, "description: " ||
group.COMMENT
    Call Lineout outFile, "distinguishedName: CN="||group.NAME||",CN=Users," ||
dc
    Call Lineout outFile, "objectCategory: CN=Group,CN=Schema,CN=Configuration,"
|| dc
    Call Lineout outFile, "objectClass: group"
    Call Lineout outFile, "name: " || group.NAME
    Call Lineout outFile, "sAMAccountName: " || group.NAME

    Call Lineout outFile, ""
    call Lineout dbFile, group.NAME || ";" || "CN=" || group.NAME || "," ||
baseOU
Return

/* -----*/
SmbCreateLDIF:
    Parse Value group.COMMENT With givenName '_' sn
    baseOU = Strip(Arg(1))
    Say group.Name

    Call Lineout outFile, "dn: CN=" || group.NAME || "," || baseOU
    Call Lineout outFile, "changetype: add"
    Call Lineout outFile, "cn: " || group.NAME
    Call SysFileSearch group.NAME, grpFile, 'gidNum.'
    if gidNum.0 = 1 then Call Lineout outFile, "gidNumber: " ||
strip(word(gidNum.1,2))

    Call Lineout outFile, "objectClass: " || "group"

    /*-----The rest are optional settings-----*/

    if group.COMMENT <> "" Then Call Lineout outFile, "description: " ||
group.COMMENT
    /*-----*/
    Call Lineout outFile, "userPassword: " || "???????"
    /*-----*/
    Call Lineout outFile, ""
    call Lineout dbFile, group.NAME || ";" || "CN=" || group.NAME || "," ||
baseOU
Return

```

GETGRPS2.CMD

The Getgrps2.cmd is part of the LSMT package.

Usage

C:\OS2MIG\GETGRPS2.CMD /SRV:PDC /OUT:C:\OS2MIG\GETGRPS2.LOG /M

/SRV - The netbios name of the OS/2 domain controller

/OUT - The output file that will be used later in the book

/M - Suppress logging information to the screen

Source code

Example 9-26 GETGRPS2.CMD

```
/*-----*/
| GET all GROUPS & Members from a LAN Server 3.0 and higher
| and dump it to an ASCII File
|                                     (C) Alain Rykaert IBM Belgium SEP95-MAY96
|                                     FEB2000
|-----*/
```

Parse Arg Option

```
Call INIT           /* Initialisation of DLL's and other stuff*/
Call CHKOPT         /* Check Options & display Welcome*/
Call CHKPWS        /* Check the PWS & Admin name*/
Call MAIN          /* do the main job*/
Call QUIT          /* Quit*/
```

```
MAIN:/* ----- MAIN: -----*/
```

```
Call Time('R')
'if exist' OUTF 'del' OUTF
Call LineOut OUTF, '* Do not modify a user from the ADMINS, GUEST, SERVERS
or USERS groups *'
```

```
RC = NetEnumerate(70, 'GROUPS', '\\SRVNAME)
if RC = 0
then Call RxStemSort 'GROUPS'
else do
Call LOGIT 'Get Groups', SRVNAME, RC
Return
end
```

```
UL = 1 /* Determine the Maximum USER length*/
/* RC = NetEnumerate(280, 'USERS', '\\SRVNAME) */
RC = WfrxUserEnum('\\SRVNAME,'USERS')
if RC = 0
then do i = 1 to USERS.0
if Length(USERS.i) > UL then UL = Length(USERS.i)
```

```

        end
    else do
        Call LOGIT 'Get Users', SRVNAME, RC
        Return
    end

HEADER = 'OPT;'Left('USERS',UL,' ');'
ALLGRP = ''
EMTGRP = ''

HEADER = HEADER || ALLGRP
do i = 1 to GROUPS.0
    CL = Length(GROUPS.i)
    ALLGRP = ALLGRP || Left(GROUPS.i,CL,' ') || ';' /* All Groups Line*/
    EMTGRP = EMTGRP || Left(' ',CL,' ') || ';' /* Empty Groups Line*/
end
HEADER = HEADER || ALLGRP
Call LineOut UTF, HEADER

/* RC = NetEnumerate(280, 'USERS', '\\SRVNAME) */
RC = WfrxUserEnum('\\SRVNAME,'USERS')
if RC = 0
    then Call RxStemSort 'USERS'
    else do
        Call LOGIT 'Get Users', SRVNAME, RC
        Return
    end

do i = 1 to USERS.0
    if i // MAXLINES = 0
        then Call LineOut UTF, HEADER
    else Nop

    if \MUTE
        then do
            Call SysCurState OFF
            Call SysCurPos 20,0
        end
    else Nop
    say '0909'x ESC'[K Total Users = ' i/'Users.0 USERS.i

OUT = ''

TMPGRP = EMTGRP
RC = NetGetInfo(330, 'MEMBER', '\\SRVNAME, USERS.i)
if RC = 0
    then do
        Call RxStemSort 'MEMBER'
        do j = 1 to MEMBER.0

```

```

        LEN = 1
        do k = 1 to GROUPS.0
            GRPLEN = Length(GROUPS.k)
            if MEMBER.j = GROUPS.k
                then TMPGRP = Overlay(Center('X',GRPLEN,' '),TMPGRP,
LEN)
                    else Nop
                    LEN = LEN + GRPLEN + 1
                end
            end
        end
        else say 'RC :' RC
        OUT = ' ';Left(USERS.i,UL,' ');'TMPGRP
        Call LineOut OUTF, OUT
    end

if \MUTE then say '0909'x ' Total Time =' Trunc(Time('E'),2)

Call Stream OUTF, 'C', 'CLOSE'

Call SysSetObjectData OUTF, 'ICONFILE='Directory()'\Groups2.Ico'

Return

CHKOPT:/* ----- CHKOPT: -----*/

SRVNAME = '';
OUTF = 'GROUPS2.CSV';
LOGF = 'LSMT.LOG';
PIPE = '';
TRACE = 0;
MUTE = 0;

OPTION = Translate(OPTION)
do while OPTION <> ''
    Parse value OPTION with ARGUMENT ' ' OPTION
    select
        when Left(ARGUMENT,5) = '/SRV:' then SRVNAME = Substr(ARGUMENT,6)
        when Left(ARGUMENT,5) = '/OUT:' then OUTF = Substr(ARGUMENT,6)
        when Left(ARGUMENT,5) = '/LOG:' then LOGF = Substr(ARGUMENT,6)
        when Left(ARGUMENT,5) = '/PIP:' then PIPE = Substr(ARGUMENT,6)
        when Left(ARGUMENT,2) = '/M' then MUTE = 1
        when Left(ARGUMENT,2) = '/T' then TRACE = 1
        otherwise Nop
    end
end

if SRVNAME = '' then signal GETHELP

```

```

if \MUTE
then do
    Topic1 = 'GETWELCOME'
    Topic_String.Topic1.1 = SRVNAME
    Topic_String.Topic1.2 = OUTF
    Topic_String.Topic1.3 = LOGF
    Topic_String.Topic1.4 = PIPE' '
    Topic_List='WELCOMELOGO' Topic1 'GETGRPS2'
    Call GETANS
    Parse VALUE SysCurPos() With 01d_R 01d_C; 'Pause'
    Call SysCurPos 01d_R, 01d_C; say ESC'[K'
end
else do
    say 'ServerName =' SRVNAME
    say 'OutputFile =' OUTF
    say 'LogFile    =' LOGF
end
Return

CHKPWS:/* ----- CHKPWS: -----*/

RC = NetGetInfo(350, 'WKSTAINFO', '')
if RC = 0
then do
    ADMNAME = WKSTAINFO.UserName
    PWSNAME = WKSTAINFO.ComputerName
end
else do
    Call LOGIT 'Get PWS Info', ,RC
    Call Quit
end
Return

INIT:/* ----- INIT: -----*/

Call RgUtil  '/m'                /* Rexx Utilities*/
Call RgUtils '/m'                /* Rexx Utilities*/
Call RgNPipes '/m'              /* Named Pipes*/
Call RgLSRXUT '/m'              /* Lan Server Rexx Utils*/

Parse Upper Source . . P_NAME
PRGN = Filespec('N', Left(P_NAME, Length(P_NAME) -4))

'@echo off'
ESC = '1B'x
REDIR = '>NUL 2>NUL'
MAXLINES = 20                    /* Number of Lines to separate with a header*/

COLF = 'GROUPS.INI'              /* Column description file*/

```

```

Call CHKFILE COLF

Resource_File = 'LSMT.RSC'
Call CHKFILE Resource_File

Return

GETANS:/* ----- GETANS: ----*/

Vars_List =Ansi_Say(Resource_File Topic_List);
Parse VALUE SysCurPos() With Old_R Old_C;
Do While Vars_List <> '';
    Parse VALUE Vars_List With Topic_Id ';' Var_Id ';' Row ';' Column ';';
Color ';' Vars_List;
    Call SysCurPos Row, Column;
    Say x2c(Color) || Topic_String.Topic_Id.Var_Id || '1B'x || '[0m';
End;
Call SysCurPos Old_R, Old_C;

Return

GETHELP:/* ----- GETHELP: ----*/

if \MUTE
then do
    Topic1='GETHELP'
    Topic_String.Topic1.1=PRGN;
    Topic_List=Topic1;
    Call GETANS
end
else say 'Incorrect options.'

Call QUIT

Return

CHKFILE:/* ----- CHKFILE: ----*/

Parse Arg FILE
RC = Stream(FILE, 'C', 'QUERY EXIST')
if RC = ''
then do
    say ' File' FILE 'not found.'
    Call QUIT
end
else Nop
Call Stream FILE, 'C', 'CLOSE'

Return

```

```

LOGIT:/* ----- LOGIT: -----*/

    FUNC = ARG(1); INFO = ARG(2); RCODE = ARG(3)
    RC = LLOGIT(LOGF, PIPE, ADMNAME, PRGN, FUNC, INFO, RCODE)

    Return

QUIT:/* ----- QUIT: -----*/

    Call LineOut 'LSMT.END', PRGN, 1
    Call Stream 'LSMT.END', 'C', 'CLOSE'
    Call Stream COLF, 'C', 'CLOSE'
    Call Stream LOGF, 'C', 'CLOSE'
    Call Stream OUTF1, 'C', 'CLOSE'
    Call Stream OUTF2, 'C', 'CLOSE'
    Exit

```

SETGRPMEM.CMD

The SETGRPMEM.CMD code was used in both the Windows and Linux migration. We have written a simplified piece of code with no error checking to assist in a migration environment.

Usage

C:\OS2MIG\SETGRPMEM.CMD [WIN | SMB] [INPUT FILE] [OUTPUT FILE] [BRANCH NAME]

[WIN | SMB] - To what platform are you migrating to Windows (WIN) or Linux (SMB)

[INPUT FILE] - Use the GETGRPS2.LOG from the GETGRPS2.CMD

[OUTPUT FILE] - The output file that will be used for the migration. In our case we named our output file SETGRPMEM.LDIF.

[BRANCH NAME] - The name of the branch that will be migrating.

Source code

Example 9-27 SETGRPMEM.CMD source code

```

/* */
Parse Arg srvType inFile outFile BranchName

srvType = strip(translate(srvType))
inFile = Strip(inFile)

```



```

outFile = Strip(outFile)
BranchName = Strip(BranchName)

dnsDomain = "somedomain.local"
dc = "DC=somedomain,DC=local"
baseDN = "OU=Users,OU=" || BranchName || ",OU=Branch," || dc
dbFile = "group-db.csv"

'@del 'outFile' 1>NUL 2>NUL'

Do While Lines(dbFile)
  curLine = LineIN(dbFile)
  Parse Value curLine With os2name ';' ldapName
  dbGroup.os2name = ldapName
End

Do While Lines(inFile)
  curLine = LineIN(inFile)
  orgLine = curLine
  Parse Value curLine With Opt ';' curLine
  Select
    When Opt = '' | curLine = '' | Left(Strip(Opt),1) = '*' Then Iterate
    When Translate(Opt) = 'OPT' Then Call GetColumns
    When Translate(Opt) = 'A' Then Call AddGroupMember
    Otherwise Iterate
  End
End
Exit ExitCode
Return

/* -----*/
AddGroupMember:
  Parse Value curLine with userid ';' curLine
  i = 1
  Do While curLine <> ''
    i = i + 1
    groupName = Strip(columnNames.i)
    Parse value curLine With actValue ';' curLine
    actValue = Translate(Strip(actValue))

    if groupName = 'ADMINS' | groupName = 'GUESTS' | groupName = 'USERS' then
Iterate

    if (actValue = 'X') & (srvType = 'WIN') then CALL WinCreateLDIF userID,
dbGroup.groupName
    if (actValue = 'X') & (srvType = 'SMB') then CALL SmbCreateLDIF userID,
dbGroup.groupName
  End
Return

```

```

/* -----*/
GetColumns:
  i = 0
  Do While curLine <> ''
    i = i + 1
    Parse value curLine With columnNames.i ';' curLine
  End
  numColumn = i
Return

/* -----*/
WinCreateLDIF:
  Call Lineout outFile, "dn: " || Strip(Arg(2))
  Call Lineout outFile, "changetype: modify"
  Call Lineout outFile, "add: member"
  Call Lineout outFile, "member: CN=" || Strip(Arg(1)) || "," || baseDN
  Call Lineout outFile, "-"
  Call Lineout outFile, " "
Return

/* -----*/
SmbCreateLDIF:
  Call Lineout outFile, "dn: " || Strip(Arg(2))
  Call Lineout outFile, "changetype: modify"
  Call Lineout outFile, "add: member"
  Call Lineout outFile, "memberUID: " || Strip(Arg(1))
  Call Lineout outFile, "-"
Return

```

Source code for USER

The files below are used to retrieve user information from an OS/2 domain.

GETUSERS.CMD

The **Getusers.cmd** is part of the LSMT package. When running the command with all the parameters, the output file will be used in this redbook for the migration to Windows or Linux.

Usage

```
C:\OS2MIG\GETUSERS.CMD /SRV:PDC /OUT:C:\OS2MIG\GETUSERS.LOG /M
```

/SRV - The netbios name of the OS/2 domain controller

/OUT - The output file that will be used later in the book

/M - Suppress logging information to the screen

Source code

Example 9-28 GETUSERS.CMD

```
/*-----*\
|   GET all USERS from a LAN Server 3.0 and higher
|   and dump it to an ASCII File
|
|                                     (C) Alain Rykaert IBM-Belgium SEP95-MAY96
|                                     FEB2000
|-----*\

Parse Arg Option

MaxLines = 9999          /* Number of Lines to separate with a header*/

Call Init                /* Initialisation of DLL's and other stuff*/
Call ChkOpt              /* Check Options & display Welcome*/
Call ChkPws              /* Check the PWS & Admin name*/
Call Columns            /* Read the Columns definition file*/
Call Main                /* do the main job*/
Call Quit                /* Quit*/

MAIN:/* ----- MAIN: -----*/

Call Time('R')
'if exist' OUTF 'del' OUTF

/* RC = NetEnumerate(280, 'USERID', '\\SRVNAME) */
RC = WfrxUserEnum('\\SrvName,'UserID')
If RC = 0
  Then Do
    Call RxStemSort 'UserID'
    Call LineOut OutF, Banner
    Do i = 1 to UserID.0
      If i // MaxLines = 0
        Then Call LineOut OutF, Banner
      Else Nop
    If \MUTE
      Then Do
        Call SysCurState Off
        Call SysCurPos 20,0
      End
    Else Nop
    Say '0909'x ESC'[K Total Users = ' i/'UserID.0 UserID.i
    RC = NetGetInfo(280, 'UserInfo', '\\SrvName, UserID.i)
    If RC = 0
```

```

                Then Call Writeit
                Else Call Logit 'Get NetGetInfo', UserID.i, RC
            End
        End
    Else Do
        Call LOGIT 'Get Users', SRVNAME, RC
        Call QUIT
    End

If \Mute Then Say '0909'x ' Total Time  =' Trunc(Time('E'),2)

Call Stream OutF, 'C', 'Close'

Call SysSetObjectData OutF, 'ICONFILE='Directory()\Users.Ico'

Return

WRITEIT:/* ----- WRITEIT: ----*/

USERINFO.OPT      = Left('',COLL.1,' ') /* Column 1 must be BLANK*/
USERINFO.PASSWORD = Left('*****', COLL.3, ' ')

OUT = ''
do j = 1 to COLT
    COLNAME = COLN.j
    DATA.j = Left(USERINFO.COLNAME, COLL.j, ' ')
    OUT = OUT || DATA.j || ';'
end
Call LineOut OutF, OUT
Call Stream OutF, 'C', 'CLOSE'
Return

COLUMNS:/* ----- COLUMNS: ----*/

BANNER = ''
i = 0
do while Lines(COLF)
    LLINE = LineIn(COLF)
    if Left(LLINE, 1) = '*',
        | Strip(LLINE) = ''
        then iterate
    else Nop
    i = i + 1
    parse value LLINE with COLN ';' COLL
    COLN.i = Strip(COLN)
    COLL.i = Strip(COLL)
    BANNER = BANNER || Left(COLN.i, COLL.i, ' ') || ';'
end
COLT = i

```

```

Call Stream COLF, 'C', 'CLOSE'

Return

CHKOPT:/* -----* /

SRVNAME = ''
OUTF    = 'USERS.CSV'
LOGF    = 'LSMT.LOG'
PIPE    = ''
TRACE   = 0
MUTE    = 0

OPTION = Translate(OPTION)
do while OPTION <> ''
  Parse value OPTION with ARGUMENT ' ' OPTION
  select
  when Left(ARGUMENT,5) = '/SRV:' then SRVNAME = Substr(ARGUMENT,6)
  when Left(ARGUMENT,5) = '/OUT:' then OUTF    = Substr(ARGUMENT,6)
  when Left(ARGUMENT,5) = '/LOG:' then LOGF    = Substr(ARGUMENT,6)
  when Left(ARGUMENT,5) = '/PIP:' then PIPE    = Substr(ARGUMENT,6)
  when Left(ARGUMENT,2) = '/M'   then MUTE    = 1
  when Left(ARGUMENT,2) = '/T'   then TRACE   = 1
  otherwise Nop
  end
end

if SRVNAME = '' then signal GETHELP

if \MUTE
then do
  Topic1='GETWELCOME'
  Topic_String.Topic1.1=SRVNAME;
  Topic_String.Topic1.2=OUTF;
  Topic_String.Topic1.3=LOGF;
  Topic_String.Topic1.4=PIPE' ';
  Topic_List='WELCOMELOGO' Topic1 'GETUSERS';
  Call GETANS
  Parse VALUE SysCurPos() With Old_R Old_C; '@Pause';
  Call SysCurPos Old_R, Old_C; say ESC'[K';
end
else do
  say 'ServerName =' SRVNAME
  say 'OutputFile =' OUTF
  say 'LogFile     =' LOGF
end

Return

```

```

CHKPWS:/* ----- CHKPWS: -----*/

RC = NetGetInfo(350, 'WKSTAINFO','')
if RC = 0
then do
    ADMNAME = WKSTAINFO.UserName
    PWSNAME = WKSTAINFO.ComputerName
end
else do
    Call LOGIT 'Get PWS Info', ,RC
    Call Quit
end
Return

INIT:/* ----- INIT: -----*/

Call RgUtil '/M' /* Rexx Utilities*/
Call RgUtils '/M' /* Rexx Utilities*/
Call RgLRSXUT '/M' /* Lan Server Rexx Utils*/

Parse Upper Source . . P_NAME
PRGN = Filespec('N', Left(P_NAME, Length(P_NAME) -4))

'@echo off'
Esc = '1B'x
Redir = '>NUL 2>NUL'

COLF = 'USERS.INI' /* Column description file*/
Call CHKFILE COLF

Resource_File = 'LSMT.RSC' /* Ansi Topics Resource File*/
Call CHKFILE Resource_File

Return

GETANS:/* ----- GETANS: -----*/

Vars_List = Ansi_Say(Resource_File Topic_List)
Parse VALUE SysCurPos() With Old_R Old_C
Do While Vars_List <> ''
    Parse VALUE Vars_List With Topic_Id ';' Var_Id ';' Row ';' Column ';'
Color ';' Vars_List
    Call SysCurPos Row, Column
    Say x2c(Color) || Topic_String.Topic_Id.Var_Id || '1B'x || '[Om'
End
Call SysCurPos Old_R, Old_C

Return

```

```

GETHELP:/* ----- GETHELP: ----*/

  if \MUTE
  then do
    Topic1 = 'GETHELP'
    Topic_String.Topic1.1 = PRGN;
    Topic_List = Topic1;
    Call GETANS
  end
  else say 'Incorrect options.'

Call QUIT

Return

CHKFILE:/* ----- CHKFILE: ----*/

  Parse Arg File

  If Stream(File, 'C', 'Query Exists') = ''
  Then Do
    say ' File' File 'not found.'
    Call Quit
  End
  Else Nop
Call Stream File, 'C', 'Close'

Return

LOGIT:/* ----- LOGIT: ----*/

  FUNC = ARG(1); INFO = ARG(2); RCODE = ARG(3)
  RC = LLOGIT(LOGF, PIPE, ADMNAME, PRGN, FUNC, INFO, RCODE)

Return

QUIT:/* ----- QUIT: ----*/

  Call LineOut 'LSMT.END', PRGN, 1
  Call Stream 'LSMT.END', 'C', 'CLOSE'
  Call Stream COLF, 'C', 'CLOSE'
  Call Stream LOGF, 'C', 'CLOSE'
  Call Stream OUTF, 'C', 'CLOSE'

Exit

```

USERS.INI

The Users.ini is part of the LSMT package. GETUSERS.COMM uses the INI file to create the output file.

Usage

None

Source code

Example 9-29 USERS.INI

* DO NOT CHANGE THE FIRST 3 COLUMNS ORDER
* AND DO NOT CHANGE THE COLUMNS NAMES
*

```
OPT                ; 3
NAME               ; 9
PASSWORD          ; 8
PASSWORD_AGE     ; 12
PRIV              ; 13
HOME_DIR         ; 45
COMMENT          ; 45
FLAGS            ; 5
SCRIPT_PATH      ; 12
AUTH_FLAGS       ; 10
FULL_NAME        ; 45
USR_COMMENT      ; 45
PARMS            ; 20
WORKSTATIONS     ; 15
LAST_LOGON       ; 24
LAST_LOGOFF      ; 24
ACCT_EXPIRES     ; 24
MAX_STORAGE      ; 11
RESTRICTED_HOURS ; 25
1.LOGON_HOURS    ; 63
2.LOGON_HOURS    ; 63
3.LOGON_HOURS    ; 63
4.LOGON_HOURS    ; 63
5.LOGON_HOURS    ; 63
6.LOGON_HOURS    ; 63
7.LOGON_HOURS    ; 63
BAD_PW_COUNT     ; 15
NUM_LOGONS       ; 15
LOGON_SERVER     ; 12
COUNTRY_CODE     ; 15
CODE_PAGE        ; 12
```



```
* PUT A '*' TO ANY COLUMN TO BE EXCLUDED
*
*****
```

SETUSERS.CMD

The SETUSERS.CMD code was used in both the Windows and Linux migration. We have written a simplified piece of code with no error checking to assist in a migration environment.

Usage

```
C:\OS2MIG\SETUSERS.CMD [WIN | SMB] [INPUT FILE] [OUTPUT FILE] [BRANCH
NAME] [LSMT PASSWORD OUTPUT FILE] [USER ID FILE]
```

[WIN | SMB] - To what platform are you migrating to Windows (WIN) or Linux (SMB)?

[INPUT FILE] - Use the GETUSERS.LOG from the GETUSERS.CMD.

[OUTPUT FILE] - The output file that will be used for the migration. In our case, we named our output file SETUSERS.LDIF.

[BRANCH NAME] - The name of the branch that will be migrating.

[LSMT PASSWORD OUTPUT FILE] - Only needed for Linux. The output file that was created from GETPWD.CMD that you find at <ref GETPWD.cmd>

[USER ID FILE] - Only needed for Linux. This is a file with all the user IDs in the organization with a unique number, such as their personal number. We manually created a file call TRANSFORM.USER for our migration process. For more information on TRANSFORM.USER, look at <ref TRANSFORM.USER>

Source code

Example 9-30 SETUSERS.CMD source code

```
/* */
call RxFuncAdd 'SysLoadFuncs', 'REXXUTIL', 'SysLoadFuncs'
call SysLoadFuncs
/* win|smb getusers.log out.log Branch1 getpwd.log transfrm.usr */
Parse Arg srvType inFile outFile BranchName smbPwdFile smbUsrFile

srvType = strip(translate(srvType))
inFile = Strip(inFile)
outFile = Strip(outFile)
BranchName = Strip(BranchName)
```

```

smbPwdFile = Strip(smbPwdFile)
smbUsrFile = Strip(smbUsrFile)
ksSystemSID = 'S-1-5-21-0123456789-0123456789-0123456789-'

dnsDomain = "somedomain.local"
dc = "DC=somedomain,DC=local"
baseDN = "OU=Users,OU=" || BranchName || ",OU=Branch," || dc

prtOp = "CN=Print Operators,CN=Builtin," || dc
accOp = "CN=Account Operators,CN=Builtin," || dc
srvOp = "CN=Server Operators,CN=Builtin," || dc
domUsr = "CN=Domain Users,CN=Users," || dc
domAdm = "CN=Domain Admins,CN=Users," || dc
domGue = "CN=Domain Guests,CN=Users," || dc

'@del 'outFile' 1>NUL 2>NUL'

Do While Lines(inFile)
  curLine = LineIN(inFile)
  orgLine = curLine
  Parse Value curLine With Opt ';' curLine
  Select
    When Opt = '' | curLine = '' | Left(Strip(Opt),1) = '*' Then Iterate
    When Translate(Opt) = 'OPT' Then Call GetColumns
    When Translate(Opt) = 'A' Then Call AddUser
    Otherwise Iterate
  End
End
Exit ExitCode
Return

/* -----*/
AddUser:
  i = 0
  Do While curLine <> ''
    i = i + 1
    columnName = Strip(columnNames.i)
    Parse value curLine With actValue ';' curLine
    user.columnName = Strip(actValue)
    If (user.columnName = "No limit") | (user.columnName = "-none-") |
(user.columnName = "No Restriction") | (user.columnName = "Unknown") |
(user.columnName = "(null)") Then
      user.columnName = ''
    End
    if srvType = "WIN" then Call Win32CreateLDIF
    if srvType = "SMB" then Call SMBCreateLDIF
  Return
/* -----*/

```

```

GetColumns:
  i = 0
  Do While curLine <> ''
    i = i + 1
    Parse value curLine With columnNames.i ';' curLine
  End
  numColumn = i
Return

/* -----*/
Win32CreateLDIF:
  Parse Value user.COMMENT With givenName '_' sn
  SAY user.NAME
  userDN = "CN=" || user.NAME || "," || baseDN
  Call Lineout outFile, "dn: " || userDN
  Call Lineout outFile, "changetype: add"
  Call Lineout outFile, "cn: " || user.NAME
  Call Lineout outFile, "distinguishedName: " || userDN
  Call Lineout outFile, "objectCategory: CN=Person,CN=Schema,CN=Configuration,"
  || dc
  Call Lineout outFile, "objectClass: user"

  Call Lineout outFile, "givenName: "givenName
  Call Lineout outFile, "sn: " || sn
  Call Lineout outFile, "displayName: " || user.Name
  Call Lineout outFile, "name: " || user.Name
  Call Lineout outFile, "userPrincipalName: " || user.Name || "@" || dnsDomain
  if user.USR_COMMENT <> "" Then Call Lineout outFile, "description: " ||
user.USR_COMMENT
  Call Lineout outFile, "pwdLastSet: 0"
  Call Lineout outFile, "sAMAccountName: " || user.Name
  if user.MAX_STORAGE <> "" Then Call Lineout outFile, "maxStorage: " ||
FORMAT(user.MAX_STORAGE)
  if user.CODE_PAGE <> "" Then Call Lineout outFile, "codePage: " ||
FORMAT(user.CODE_PAGE)
  if user.COUNTRY_CODE <> "" Then Call Lineout outFile, "countryCode: " ||
FORMAT(user.COUNTRY_CODE)
  Call Lineout outFile, "logonHours:: " || ReadLogonHours()
  if POS("D",user.FLAGS) > 0 Then
    Call Lineout outFile, "userAccountControl: " || 514
  else
    Call Lineout outFile, "userAccountControl: " || 512
  if user.WORKSTATIONS <> "" Then
    Call Lineout outFile, "userWorkstations: " || TRANSLATE(user.WORKSTATIONS,
",", " ")
  Call Lineout outFile, "scriptPath: logon.cmd"
  Call Lineout outFile, "homeDrive: " || LEFT(user.HOME_DIR,1)

```

```

Call Lineout outFile, "homeDirectory: \\\" || WORD(TRANSLATE(user.HOME_DIR,"
", "\"),2) || "\" || user.NAME

/* Only usable with ObjRexx enabled in OS/2 with SWITCHRX */
parse version version
if WORD(version,1) = "OBJREXX" & user.ACCT_EXPIRES <> "" Then Do
  expTime = Date('Base', WORD(user.ACCT_EXPIRES,3) || " " ||
WORD(user.ACCT_EXPIRES,2) || " " || WORD(user.ACCT_EXPIRES,5), 'Normal')
  expTime = expTime - Date('Base','01 Jan 1601', 'Normal') +1
  expTime = expTime * 24 * 60 * 60 * 10000000
  call Lineout outFile, "accountExpires: " || format(expTime,,0)
end
Call Lineout outFile, ""

if POS("P",user.AUTH_FLAGS) > 0 Then
  Call WinAddGroupMember "add", prtOp
if POS("A",user.AUTH_FLAGS) > 0 Then
  Call WinAddGroupMember "add", accOp
if POS("S",user.AUTH_FLAGS) > 0 Then
  Call WinAddGroupMember "add", srvOp
if user.PRIV = "Guest" Then Do
  pGroupID = 514
  Call WinAddGroupMember "add", domGue
end
if user.PRIV = "Administrator" Then Do
  pGroupID = 512
  Call WinAddGroupMember "add", domAdm
end
if user.PRIV = "User" Then do
  pGroupID = 513
end
call Lineout outFile, "dn: " || userDN
call Lineout outFile, "changetype: modify"
call Lineout outFile, "add: primaryGroupID"
call Lineout outFile, "primaryGroupID: " || pGroupID
call Lineout outFile, "-"
call Lineout outFile, ""

if pGroupID <> 513 Then Do
  Call WinAddGroupMember "delete", domUsr
end
Return

/* -----*/

SMBCcreateLDIF:
Parse Value user.COMMENT With givenName '_' sn
SAY user.NAME
Call Lineout outFile, "dn: CN=" || user.NAME || "," || basedN

```

```

Call Lineout outfile, "changetype: add"
Call Lineout outfile, "uid: " || user.NAME
Call Lineout outfile, "userid: " || user.NAME
Call Lineout outfile, "objectClass: " || "sambaSamAccount"
Call Lineout outfile, "objectClass: " || "account"
Call Lineout outfile, "objectClass: " || "posixAccount"

Call Lineout outfile, "cn: " || user.NAME

Call Lineout outfile, "gidNumer: 100"
Call Lineout outfile, "homeDirectory: /home/" || user.NAME

Call SysFileSearch user.NAME, smbUsrFile, 'getUsrNum.'
if getUsrNum.0 = '1' then Call Lineout outfile, "uidNumber: "
strip(word(getUsrNum.1,2))
Call Lineout outfile, "sambaSID: " || ksSystemSID ||
strip(word(getUsrNum.1,2))

Call Lineout outfile, "sambaHomePath: \\\" || WORD(TRANSLATE(user.HOME_DIR,
",\"),2) || "\" || user.NAME
Call Lineout outfile, "sambaHomeDrive: " || LEFT(user.HOME_DIR,2)
Call Lineout outfile, "sambaLogonScript: " || "logon.cmd"
Call Lineout outfile, "sambaProfilePath: " || ""
if user.USR_COMMENT <> "" then Call Lineout outfile, "description: " ||
user.USR_COMMENT
Call Lineout outfile, "displayName: " || user.COMMENT

Call SysFileSearch user.NAME, smbPwdFile, 'getPwd.'
if getPwd.0 = '1' then
do
parse var getPwd.1 uName ':' lmpwd
Call Lineout outfile, "sambaLMPassword:" || strip(lmpwd)
end
else Call Lineout outfile, "sambaLMPassword:" || "*****"

/*-----The following will be ignored due to imcomplete doco from samba

Call Lineout outfile, "sambaNTPassword: " || ""
Call Lineout outfile, "sambaPwdLastSet: " || ""
Call Lineout outfile, "sambaPwdCanChange: " || "0"
Call Lineout outfile, "sambaPwdMustChage: " || "0"
Call Lineout outfile, "sambaAcctFlags: " || ""
Call Lineout outfile, "sambaUserWorkstations: " || ""
Call Lineout outfile, "sambaPrimaryGroupSID: " ||
Call Lineout outfile, "sambaDomainName: " ||

-----*/

```

```

    Call Lineout outFile, ""

Return
/* -----*/
ReadLogonHours:
    Base64 = "A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h
i j k l m n o p q r s t u v w x y z 0 1 2 3 4 5 6 7 8 9 + /"
    tResult = ""
    do ti=1 to 7
        actBitmap = LEFT( "", 24, "0" )
        actDay = user.ti.LOGON_HOURS
        do tj=1 TO WORDS( actDay )
            thAllowed = WORD( actDay, tj ) +1
            actBitmap = OVERLAY("1", actBitmap, thAllowed)
        end
        tResult = tResult || actBitmap
    end
    actBitmap = ""
    do ti=0 to 27
        actBitmap = actBitmap || B2X(SUBSTR(tResult,(ti*6)+1, 3))*8 +
B2X(SUBSTR(tResult,(ti*6)+4, 3)) || " "
    end
    tResult = ""
    do ti = 1 To WORDS(actBitmap)
        tResult = tResult || WORD( Base64, WORD(actBitmap,ti)+1)
    end
return tResult

WinAddGroupMember:
    Parse Arg option, WGrpName
    call Lineout outFile, "dn: " || WGrpName
    call Lineout outFile, "changetype: modify"
    call Lineout outFile, option || ": member"
    call Lineout outFile, "member: " || userDN
    call Lineout outFile, "-"
    call Lineout outFile, ""
Return

```

Source code for passwords

The following files are used for migrating passwords from an OS/2 domain.

GETPWD.CMD

The Getpwd.cmd is part of the LSMT package.

Usage

C:\OS2MIG\GETPWD.COM /SRV:PDC /OUT:C:\OS2MIG\GETPWD.LOG /M

/SRV - The netbios name of the OS/2 domain controller.

/OUT - The output file that will be used later in the book.

/M - Suppresses logging information to the screen

Source code

Example 9-31 GETPWD.COM

```
/*-----*/
|   GET all PASSWORDS from a LAN Server 3.0 and higher
|   and dump it to an ASCII File
|                                     (C) Alain Rykaert IBM Belgium SEP95-MAY96
|-----*/

Parse Arg Option

Call INIT           /* Initialisation of DLL's and other stuff*/
Call CHKOPT        /* Check Options & display Welcome*/
Call CHKPWS        /* Check the PWS & Admin name*/
Call MAIN          /* do the main job*/
Call QUIT          /* Quit*/

MAIN:/* ----- MAIN: -----*/

Call Time('R')
'if exist' OUTF 'del' OUTF

/* RC = NetEnumerate(280, 'USERID', '\\SRVNAME) */
RC = WfrxUserEnum('\\SRVNAME,'USERID')
if RC = 0
then do
    'if not exist \\SRVNAME\IBMLAN$\NETPROG\PWDEXP.EXE copy
PWDEXP.EXE \\SRVNAME\IBMLAN$\NETPROG'
    Call RxStemSort 'USERID'
    do i = 1 to USERID.0
        if \MUTE
            then do
                Call SysCurState OFF
                Call SysCurPos 19,20
                say ESC'[K'
                Call SysCurPos 19,20
            end
        else Nop
        say ' UserID : ('i/'USERID.0)' USERID.i
```

```

        'net admin \\SRVNAME '/c PWDEXP' USERID.i '>>' OUTF
    end
end
else Call LOGIT 'NetEnumerate Users :' RC

if \MUTE then say '0909'x ' Total Time =' Trunc(Time('E'),2)

Call Stream OUTF, 'C', 'CLOSE'

Call SysSetObjectData OUTF, 'ICONFILE='Directory()'\UsersPW.Ico'

Return

CHKOPT:/* -----*----- CHKOPT: -----*/

SRVNAME = '';
OUTF = 'USERS.PWD';
LOGF = 'PASSWORD.LOG';
PIPE = '';
TRACE = 0;
MUTE = 0;

OPTION = Translate(OPTION)
do while OPTION <> ''
    Parse value OPTION with ARGUMENT ' ' OPTION
    select
    when Left(ARGUMENT,5) = '/SRV:' then SRVNAME = Substr(ARGUMENT,6)
    when Left(ARGUMENT,5) = '/OUT:' then OUTF = Substr(ARGUMENT,6)
    when Left(ARGUMENT,5) = '/LOG:' then LOGF = Substr(ARGUMENT,6)
    when Left(ARGUMENT,5) = '/PIP:' then PIPE = Substr(ARGUMENT,6)
    when Left(ARGUMENT,2) = '/M' then MUTE = 1
    when Left(ARGUMENT,2) = '/T' then TRACE = 1
    otherwise Nop
    end
end

if SRVNAME = '' then signal GETHELP

if \MUTE
then do
    Topic1='GETWELCOME'
    Topic_String.Topic1.1=SRVNAME;
    Topic_String.Topic1.2=OUTF;
    Topic_String.Topic1.3=LOGF;
    Topic_String.Topic1.4=PIPE' ';
    Topic_List='WELCOMELOGO' Topic1 'GETPWD';
    Call GETANS
    Parse VALUE SysCurPos() With Old_R Old_C; '@Pause';
    Call SysCurPos Old_R, Old_C; say ESC'[K';

```



```

        end
    else do
        say 'ServerName =' SRVNAME
        say 'OutputFile =' OUTF
        say 'LogFile     =' LOGF
    end

Return

CHKPWS:/* ----- CHKPWS: -----*/

RC = NetGetInfo(350, 'WKSTAINFO','')
if RC = 0
then do
    ADMNAME = WKSTAINFO.UserName
    PWSNAME = WKSTAINFO.ComputerName
end
else do
    Call LOGIT 'Get PWS Info', ,RC
    Call Quit
end
Return

INIT:/* ----- INIT: -----*/

Call RgUtil  '/m'                /* Rexx Utilities*/
Call RgUtils '/m'                /* Rexx Utilities*/
Call RgNPipes '/m'              /* Named Pipes*/
Call RgLsrxut '/m'              /* Lan Server Rexx Utils*/

Parse Upper Source . . P_NAME
PRGN = Filespec('N', Left(P_NAME, Length(P_NAME) -4))

'@echo off'
ESC   = '1B'x
REDIR = '>NUL 2>NUL'

Call CHKFILE 'PWDEXP.EXE'      /* External program of Steve Freeman*/

Resource_File = 'LSMT.RSC'
Call CHKFILE Resource_File

Return

GETANS:/* ----- GETANS: -----*/

Vars_List =Ansi_Say(Resource_File Topic_List);
Parse VALUE SysCurPos() With Old_R Old_C;
Do While Vars_List <> '';

```

```

        Parse VALUE Vars_List With Topic_Id ';' Var_Id ';' Row ';' Column ';'
Color ';' Vars_List;
    Call SysCurPos Row, Column;
    Say x2c(Color) || Topic_String.Topic_Id.Var_Id || '1B'x || '[0m';
    End;
    Call SysCurPos Old_R, Old_C;
    Parse VALUE SysCurPos() With Old_R Old_C;
    Call SysCurPos Old_R, Old_C;

    Return

GETHELP:/* ----- GETHELP: ----*/

    if \MUTE
    then do
        Topic1='GETHELP'
        Topic_String.Topic1.1=PRGN;
        Topic_List=Topic1;
        Call GETANS
    end
    else say 'Incorrect options.'

    Call QUIT

    Return

CHKFILE:/* ----- CHKFILE: ----*/

    Parse Arg FILE
    RC = Stream(FILE, 'C', 'QUERY EXIST')
    if RC = ''
    then do
        say ' File' FILE 'not found.'
        Call QUIT
    end
    else Nop
    Call Stream FILE, 'C', 'CLOSE'

    Return

LOGIT:/* ----- LOGIT: ----*/

    Parse Arg LOGT
    if \MUTE
    then say ESC'[0;1;32m' LOGT ESC'[0m'
    else say LOGT
    LOGT = '('Date('E') Left(Time(),5) ADMNAME Substr(PWSNAME,3)')' LOGT
    Call LineOut LOGF, LOGT
    Call Stream LOGF, 'C', 'CLOSE'

```

```

if PIPE <> ''
then do
    parse value RxNPOpen(PIPE) with RC HANDLE .
    if RC <> 0
    then Return
    else do
        RC = RxNPWrite(HANDLE, LOGT)
        if RC <> 0
        then Return
        else Call RxNPClose HANDLE
    end
end
else Nop

Return

QUIT:/* ----- QUIT: -----*/

Call LineOut 'LSMT.END', PRG, 1
Call Stream 'LSMT.END', 'C', 'CLOSE'

Call Stream LOGF, 'C', 'CLOSE'
Call Stream OUTF, 'C', 'CLOSE'

Exit

```

Source code for access control lists

The following files are used to migrate information regarding access control lists.

GETSMBACL.CMD

The Getsmbacl.cmd is part of the LSMT package.

Usage

```
C:\OS2MIG\GETSMBACL.CMD /SRV:PDC /OUT:C:\OS2MIG\GETSMBACL.LOG /M
```

/SRV - The netbios name of the OS/2 domain controller

/OUT - The output file that will be used later in the book

/M - Suppers logging information to the screen

Source code

Example 9-32 GETACL.CMD

```
/*-----*\
|   GET all access control lists of all aliases definEnd on a server
|   LS 3.0 and higher and dump into an ASCII file.
|   (C) Alain Rykaert IBM Belgium & Hermann Pauli IBM Germany SEP95-MAY96
|                                           OCT97
|-----*\

LUserId = 8                /* max.length of any user ID in your Dom*/
                          /* LS 2.0 and 3.0 LID <= 8, LS 4.0 <= 15 */

Parse Arg Option

Call Init                  /* Initalisation of DLL's and other stuff*/
Call ChkOpt                /* Check Options & display Welcome*/
Call ChkPWS                /* Check the PWS & Admin name*/
Call Main                  /* Do the main job*/
Call Quit                  /* Quit*/

MAIN:/* ----- MAIN: -----*/

Call Time('R')
'If exist' OUTF 'del' OUTF
Call LineOut OUTF, '* List of all ACLs of existing Aliases,',
                || ' allowed Options U=update D=delete'

/* Prepare the output tables banner */

Call GetBanner
Comment = '* type of alias :'

Call LineOut OutF, GULst
NumAlias = 0

Call RxStemSort 'ALIASFiles'
Do i = 1 to ALIASFiles.0
  Call Status 'F' ALIASFiles.i i
  Call GetInfo ALIASFiles.i
End
Say

Call RxStemSort 'ALIASPrint'
Do i = 1 to ALIASPrint.0
  Call Status 'P' ALIASPrint.i i
  Call GetInfo ALIASPrint.i
End
Say
```

```

Call RxStemSort 'ALIASSerial'
Do i = 1 to ALIASSerial.0
  Call Status 'S' ALIASSerial.i i
  Call GetInfo ALIASSerial.i
End

If \MUTE Then Say '0909'x ' Total Time  =' Trunc(Time('E'),2)

Call Stream OutF, 'C', 'Close'

Call SysSetObjectData OutF, 'ICONFILE='Directory()'\ACL.Ico'

Return

GetInfo:/* ----- GetInfo: ----*/

Parse Arg W_Alias
OUT = EmptyLst
OUT = OVERLAY(W_ALIAS,OUT,5)

/* Retrieve info about all Aliases (we need server and resource info) */
RC = NetGetInfo(NETALIAS, 'AliasInfo', '\\\SRVNAME, W_Alias)
If RC = 0
  Then Nop
  Else Do
    Call LOGIT 'NetGetInfo Alias', W_Alias ,RC
    Call Quit
  End

/* find the correct ressource, depEndEnd on alias type */
Select
  When AliasInfo.type = 'Files'   Then Alias_Res = AliasInfo.path
  When AliasInfo.type = 'Printer' Then Alias_Res = '\print\AliasInfo.queue
  When AliasInfo.type = 'Serial'  Then Alias_Res = '\comm\AliasInfo.queue
  Otherwise nop
End

/* Get ACP for alias */
RC = NetGetInfo(NETACCESS, 'ACP', '\\\ || AliasInfo.server, Alias_Res)
If RC = 0
  Then Do
    NumAlias = NumAlias + 1
    If NumAlias // MAXLINES = 0
      Then Call Lineout OUTF, GULst
    Else nop

    Do k = 1 to ACP.count
      UserGroup = ACP.k.ugname
      If GUPos.UserGroup = 0

```

```

        Then nop                                /* user not in list*/
        Else Do                                  /* add found ACP info to output line*/
            OUT = OVERLAY(Strip(ACP.audit),OUT,LPre_Banner-6)
            LCol = POS(';','GULst,GUPos.UserGroup)-GUPos.UserGroup
            LACPT =
CENTER(ACP.k.access,Max(LCol,LENGTH(ACP.k.access)))
            OUT = OVERLAY(LACPT,OUT,GUPos.UserGroup)
        End
    End
    Call LineOut OUTF, OUT
End

Else If Left(strip(RC),4) = '2222'              /* no ACP found*/
    Then Do
        NumAlias = NumAlias + 1
        If NumAlias // MAXLINES = 0
            Then Call Lineout OUTF, GULst
            Else nop

        Call Lineout OUTF, OUT
    End

    Else Call LOGIT 'NetGetInfo Access', W_User || '/' || W_Alias, RC
Return

Status:/* ----- Status: -----*/

Parse ARG S_Typ S_ALIAS S_Number
Select
    When S_Typ = 'F' Then S_Text = 'File '
    When S_Typ = 'P' Then S_Text = 'Print '
    When S_Typ = 'S' Then S_Text = 'Serial'
    Otherwise nop
End

Do
    Say '0909'x ESC'[K' S_Text S_Number '=' S_ALIAS
    If \MUTE
        Then Do
            Call SysCurState OFF
            parse value SysCurPos() with row col
            row = row - 1
            Call SysCurPos row,0
        End
    Else Nop
End
Return

GetBanner:/* ----- GetBanner:----*/

```

```

/* get defined groups */
RC = NetEnumerate(70, 'GROUP', '\\SRVNAME)

If RC <> 0
Then Do
    Call LOGIT 'NetEnum. Group' , 'Server \' || SRVNAME, RC
    Call Quit
End
Else Call RxStemSort 'GROUP'

/* get defined userids */
RC = NetEnumerate(280, 'USERID', '\\SRVNAME)

If RC <> 0
Then Do
    Call LOGIT 'NetEnum. User' , 'Server \' || SRVNAME, RC
    Call Quit
End
Else Call RxStemSort 'USERID'

/* Get list of all aliases definEnd on server */
RC = NetEnumerate(NETALIAS, 'ALIASFiles' , '\\SRVNAME,1)
If RC <> 0 & substr(RC,1,3) <> '234 ' /* 234 = no File Alias def.*/
Then Do
    Call LOGIT 'NetEnum. F-Alias' , 'Server \' || SRVNAME , RC
    Call Quit
End
Else Call RxStemSort 'ALIASFiles'

RC = NetEnumerate(NETALIAS, 'ALIASPrint' , '\\SRVNAME,2)
If RC <> 0 & substr(RC,1,3) <> '234 ' /* 234 = no File Alias def.*/
Then Do
    Call LOGIT 'NetEnum. P-Alias' , 'Server \' || SRVNAME , RC
    Call Quit
End
Else Call RxStemSort 'ALIASPrint'

RC = NetEnumerate(NETALIAS, 'ALIASSerial', '\\SRVNAME,4)
If RC <> 0 & substr(RC,1,3) <> '234 ' /* 234 = no File Alias def.*/
Then Do
    Call LOGIT 'NetEnum. S-Alias' , 'Server \' || SRVNAME , RC
    Call Quit
End
Else Call RxStemSort 'ALIASSerial'

/* prepare first non user/group columns of the banner */
Pre_Banner = Left('OPT' || ';' || 'ALIAS', LAlias + 4, ' ')
EmptyLst = Left(' ' || ';' , length(Pre_Banner), ' ')

```

```

Pre_Banner = Pre_Banner || ';' || 'AUDIT ;'
EmptyLst   = EmptyLst   || ';' || ' ' || ';'
LPRe_Banner = length(Pre_Banner)

GULst      = Pre_Banner

GUPos.     = 0                               /* initialize all GUPos to 0*/

/* build banner using the groups */
Do i=1 to GROUP.0
  GU       = Group.i
  GUPos.GU = length(GULst) + 1
  GULst    = GULst || Left(GU,max(8,length(GU)), ' ') || ';'
  EmptyLst = EmptyLst || Left(' ',max(8,length(GU)), ' ') || ';'
End

/* build banner using the users */
Do i=1 to USERID.0
  GU       = USERID.i
  GUPos.GU = length(GULst) + 1
  GULst    = GULst || Left(GU,max(8,length(GU)), ' ') || ';'
  EmptyLst = EmptyLst || Left(' ',max(8,length(GU)), ' ') || ';'
End

Return

CHKOPT:/* ----- CHKOPT: -----*/

SRVNAME = '';
OUTF    = 'ACL.CSV';
LOGF    = 'LSMT.LOG';
PIPE    = '';
TRACE   = 0;
MUTE    = 0;

OPTION = Translate(OPTION)
Do While OPTION <> ''
  Parse value OPTION with ARGUMENT ' ' OPTION
  Select
  When Left(ARGUMENT,5) = '/SRV:' Then SRVNAME = Substr(ARGUMENT,6)
  When Left(ARGUMENT,5) = '/OUT:' Then OUTF    = Substr(ARGUMENT,6)
  When Left(ARGUMENT,5) = '/LOG:' Then LOGF    = Substr(ARGUMENT,6)
  When Left(ARGUMENT,5) = '/PIP:' Then PIPE    = Substr(ARGUMENT,6)
  When Left(ARGUMENT,2) = '/M'   Then MUTE    = 1
  When Left(ARGUMENT,2) = '/T'   Then TRACE   = 1
  otherwise Nop
End
End

```



```

If SRVNAME = '' Then signal GETHELP

If \MUTE
Then Do
    Topic1 = 'GETWELCOME'
    Topic_String.Topic1.1 = SRVNAME;
    Topic_String.Topic1.2 = OUTF;
    Topic_String.Topic1.3 = LOGF;
    Topic_String.Topic1.4 = PIPE' ';
    Topic_List = 'WELCOMELOGO' Topic1 'GETACL';
    Call GETANS
    Parse VALUE SysCurPos() With Old_R Old_C; 'Pause';
    Call SysCurPos Old_R, Old_C; Say ESC'[K';
End
Else Do
    Say 'ServerName =' SRVNAME
    Say 'OutputFile =' OUTF
    Say 'LogFile    =' LOGF
End
Return

CHKPWS:/* ----- CHKPWS: -----*/

RC = NetGetInfo(350, 'WKSTAINFO','')
If RC = 0
Then Do
    ADMNAME = WKSTAINFO.UserName
    PWSNAME = WKSTAINFO.ComputerName
End
Else Do
    Call LOGIT 'Get PWS Info', ,RC
    Call Quit
End
Return

INIT:/* ----- INIT: -----*/

Call RgUtil '/m' /* Rexx Utilities*/
Call RgUtils '/m' /* Rexx Utilities*/
Call RgNPipes '/m' /* Named Pipes*/
Call RgLsRXUT '/m' /* Lan Server Rexx Utils*/

Parse Upper Source . . P_NAME
PRGN = Filespec('N', Left(P_NAME, Length(P_NAME) -4))

'@echo off'
ESC = '1B'x
REDIR = '>NUL 2>NUL'

```

```

MAXLINES = 20                /* Number of Lines to separate with a header*/

LMinAss = 5                  /* min. length for logon assignment col*/
LAlias  = 9                  /* max.length of alias col*/
                               /* 9 entries per alias posible          */

NETACCESS = 10              /* code for LSREXX API*/
NETALIAS  = 20
NETWKSTA  = 350

NumAlias  = 0                /* number of aliases processed*/

Resource_File = 'LSMT.RSC'
Call CHKFILE Resource_File

Return

GETANS:/* ----- GETANS: ----*/

Vars_List =Ansi_Say(Resource_File Topic_List);
Parse VALUE SysCurPos() With Old_R Old_C;
Do While Vars_List <> '';
    Parse VALUE Vars_List With Topic_Id ';' Var_Id ';' Row ';' Column ';'
Color ';' Vars_List;
    Call SysCurPos Row, Column;
    Say x2c(Color) || Topic_String.Topic_Id.Var_Id || '1B'x || '[0m';
End;
Call SysCurPos Old_R, Old_C;

Return

GETHELP:/* ----- GETHELP: ----*/

If \MUTE
Then Do
    Topic1='GETHELP'
    Topic_String.Topic1.1=PRGN;
    Topic_List=Topic1;
    Call GETANS
End
Else Say 'Incorrect options.'

Call QUIT

Return

CHKFILE:/* ----- CHKFILE: ----*/

```

```

Parse Arg FILE
RC = Stream(FILE, 'C', 'QUERY EXIST')
If RC = ''
  Then Do
    Say ' File' FILE 'not found.'
    Call QUIT
  End
Else Nop
Call Stream FILE, 'C', 'CLOSE'

Return

LOGIT:/* ----- LOGIT: -----*/

FUNC = ARG(1); INFO = ARG(2); RCODE = ARG(3)
RC = LLOGIT(LOGF, PIPE, ADMNAME, PRGN, FUNC, INFO, RCODE)

Return

QUIT:/* ----- QUIT: -----*/

Call LineOut 'LSMT.End', PRGN, 1
Call Stream 'LSMT.End', 'C', 'CLOSE'

Call Stream LOGF, 'C', 'CLOSE'
Call Stream OUTF, 'C', 'CLOSE'

Exit

/*-----*/

```

GETWINACL.CMD

The Getsmbacl.cmd is not part of the LSMT package. This code was created to retrieve the ACL of an OS/2 Servers directory.

Usage

C:\OS2MIG\GETWINACL.CMD [OUTPUT FILE] [BASE PATH]

[OUTPUT FILE] - The output file will be created.

[BASE PATH] - Where to start the directory ACL search.

Source code

Example 9-33 GETWINACL.CMD source code

```
/* Get a access control profile for a drive tree */

call RxFuncAdd 'LoadLsRxutFuncs', 'LSRXUT', 'LoadLsRxutFuncs'
call LoadLsRxutFuncs
call RxFuncAdd 'SysLoadFuncs', 'REXXUTIL', 'SysLoadFuncs'
call SysLoadFuncs

Parse Arg outFile basepath

basePath = Strip(basePath)
outFile = Strip(outFile)

'@del 'outfile' 1>NUL 2>NUL'

if LENGTH(basePath)<3 Then basePath=basePath\"
rc = NetGetInfo( 10, 'AccPerm', '', basePath)
if rc <> 0 Then strAc1 = ""
else strAc1 = FormatACL()
call Lineout outFile, basePath || ";" || strAc1

Call RecurseDir basePath, strAc1

call DropLsRxutFuncs
call RxFuncDrop 'LoadLsRxutFuncs'
exit

RecurseDir: procedure expose outFile
  PARSE ARG baseDir, strACL
  Say baseDir
  baseDir = STRIP(baseDir,"T","\")
  CALL SysFileTree baseDir || '\*', 'dir.', 'DO'
  DO i = 1 TO dir.0
    rc = NetGetInfo( 10, 'AccPerm', '', dir.i)
    if rc <> 0 Then subAc1 = ""
    else subAc1 = FormatACL()
    if subAc1 <> strAc1 Then call Lineout outFile, dir.i || ";" || subAc1
    CALL RecurseDir dir.i, subAc1
  END
RETURN

FormatACL:
  ac1 = ""
  do fi=1 to AccPerm.count-1
    do fj=fi to AccPerm.count-1
      fk=fj+1
```

```

    if AccPerm.fj.ugname > AccPerm.fk.ugname then do
        tempVar = AccPerm.fk.ugname
        AccPerm.fk.ugname = AccPerm.fj.ugname
        AccPerm.fj.ugname = tempvar
        tempVar = AccPerm.fk.access
        AccPerm.fk.access = AccPerm.fj.access
        AccPerm.fj.access = tempvar
    end
end
end
do k=1 to AccPerm.count
    ac1 = ac1 || AccPerm.k.ugname || ":" || AccPerm.k.access || ";"
end
return ac1

```

Source code for aliases

The following files help with the migration of share and printer aliases.

GETALIAS.CMD

The Getalais.cmd is part of the LSMT package.

Usage

```
C:\OS2MIG\GETALIAS.CMD /SRV:PDC /OUT:C:\OS2MIG\GETALIAS.LOG /M
```

/SRV - The netbios name of the OS/2 domain controller.

/OUT - The output file that will be used later in the book.

/M - Suppers logging information to the screen.

Source code

Example 9-34 GETALIAS.CMD

```

/*-----*\
|   GET all ALIAS from a LAN Server 3.0 and higher
|   and dump it to an ASCII File
|                                     (C) Alain Rykaert IBM Belgium SEP95-MAY96
\*-----*/

```

Parse Arg Option

```

Call INIT /* Initialisation of DLL's and other stuff*/
Call CHKOPT /* Check Options & display Welcome*/

```

```

Call CHPWS                               /* Check the PWS & Admin name*/
Call COLUMNS                             /* Read the Columns definition file*/
Call MAIN                                 /* do the main job*/
Call QUIT                                  /* Quit*/

MAIN:/* ----- MAIN: -----*/

Call Time('R')
'if exist' OUTF 'del' OUTF

Type = 1
Line = 0

do while Type < 5
  if Type = 1 then TypeName = 'File '
  if Type = 2 then TypeName = 'Print '
  if Type = 4 then TypeName = 'Serial'

  Line = Line + 1
  RC = NetEnumerate(20, 'AliasName', '\\SRVNAME, Type)
  if RC = 0
    then do
      Call RxStemSort 'AliasName'
      Call LineOut OUTF, BANNER
      do i = 1 to AliasName.0
        if i // MAXLINES = 0
          then Call LineOut OUTF, BANNER
          else Nop
        if \MUTE
          then do
            Call SysCurState OFF
            Call SysCurPos 20 + Line,0
          end
        else Nop

        say '0909'x ESC'[K Type = ' TypeName 'Total Alias = '
i/'AliasName.0 ALIASNAME.i

        RC = NetGetInfo(20, 'ALIASINFO', '\\SRVNAME, AliasName.i)
        if RC = 0
          then Call WRITEIT
          else Call LOGIT 'NetGetInfo Alias', AliasInfo.i, RC
      end
    end
  else do
    if Substr(RC,1,3) = '234 '
      then do
        if \MUTE
          then do

```

```

                Call SysCurState OFF
                Call SysCurPos 19 + Line,0
            end
            else Nop
            say '0909'x ' Type =' TypeName 'Total Alias =' 0
        end
    else do
        Call LOGIT 'Get' TypeName 'Alias',SRVNAME,RC
    end
end
    Call LineOut OUTF, ' '
    Type = Type * 2
end

if \MUTE then say '0909'x ' Total Time =' Trunc(Time('E'),2)

Call Stream OUTF, 'C', 'CLOSE'

Call SysSetObjectData OUTF, 'ICONFILE='Directory()\Alias.ico'

Return

WRITEIT:/* ----- WRITEIT:
-----*/

ALIASINFO.OPT = Left('',COLL.1,' ')          /* Column 1 must be BLANK*/

OUT = ''
do j = 1 to COLT
    COLNAME = COLN.j
    DATA.j = Left(ALIASINFO.COLNAME, COLL.j, ' ')
    OUT = OUT || DATA.j || ';'
end
Call LineOut OUTF, OUT
Call Stream OUTF, 'C', 'CLOSE'

Return

COLUMNS: /* ----- COLUMNS: -*/

BANNER = ''
i = 0
do while Lines(COLF)
    LLINE = LineIn(COLF)
    if Left(LLINE, 1) = '*',
        | Strip(LLINE) = ''
        then iterate
    else Nop
    i = i + 1
end

```

```

        parse value LLINE with COLN ';' COLL
        COLN.i = Strip(COLN)
        COLL.i = Strip(COLL)
        BANNER = BANNER || Left(COLN.i, COLL.i, ' ') || ';'
    end
    COLT = i

Return

CHKOPT:/* ----- CHKOPT: -----*/

SRVNAME = '';
OUTF = 'ALIAS.CSV';
LOGF = 'LSMT.LOG';
PIPE = '';
TRACE = 0;
MUTE = 0;

OPTION = Translate(OPTION)
do while OPTION <> ''
    Parse value OPTION with ARGUMENT ' ' OPTION
    select
        when Left(ARGUMENT,5) = '/SRV:' then SRVNAME = Substr(ARGUMENT,6)
        when Left(ARGUMENT,5) = '/OUT:' then OUTF = Substr(ARGUMENT,6)
        when Left(ARGUMENT,5) = '/LOG:' then LOGF = Substr(ARGUMENT,6)
        when Left(ARGUMENT,5) = '/PIP:' then PIPE = Substr(ARGUMENT,6)
        when Left(ARGUMENT,2) = '/M' then MUTE = 1
        when Left(ARGUMENT,2) = '/T' then TRACE = 1
        otherwise Nop
    end
end

if SRVNAME = '' then signal GETHELP

if \MUTE
then do
    Topic1='GETWELCOME'
    Topic_String.Topic1.1=SRVNAME;
    Topic_String.Topic1.2=OUTF;
    Topic_String.Topic1.3=LOGF;
    Topic_String.Topic1.4=PIPE' ';
    Topic_List='WELCOMELOGO' Topic1 'GETALIAS';
    Call GETANS
    Parse VALUE SysCurPos() With Old_R Old_C; '@Pause';
    Call SysCurPos Old_R, Old_C; say ESC'[K';
end
else do
    say 'ServerName =' SRVNAME
    say 'OutputFile =' OUTF

```



```

        say 'LogFile    =' LOGF
    end
Return

CHKPWS:/* -----*/

RC = NetGetInfo(350, 'WKSTAINFO','')
if RC = 0
then do
    ADMNAME = WKSTAINFO.UserName
    PWSNAME = WKSTAINFO.ComputerName
end
else do
    Call LOGIT 'Get PWS Info', ,RC
    Call Quit
end
Return

INIT:/* -----*/

Call RgUtil  '/m'           /* Rexx Utilities*/
Call RgUtils '/m'           /* Rexx Utilities*/
Call RgNPipes '/m'         /* Named Pipes*/
Call RgLsRXUT '/m'        /* Lan Server Rexx Utils*/

Parse Upper Source . . P_NAME
PRGN = Filespec('N', Left(P_NAME, Length(P_NAME) -4))

'@echo off'
ESC    = '1B'x
REDIR  = '>NUL 2>NUL'
MAXLINES = 999             /* Number of Lines to separate with a header*/

COLF    = 'ALIAS.INI'      /* Column description file*/
Call CHKFILE COLF

Resource_File = 'LSMT.RSC'
Call CHKFILE Resource_File

Return

GETANS:/* -----*/

Vars_List =Ansi_Say(Resource_File Topic_List);
Parse VALUE SysCurPos() With Old_R Old_C;
Do While Vars_List <> '';
    Parse VALUE Vars_List With Topic_Id ';' Var_Id ';' Row ';' Column ';';
Color ';' Vars_List;
    Call SysCurPos Row, Column;

```

```

        Say x2c(Color) || Topic_String.Topic_Id.Var_Id || '1B'x || '[0m';
    End;
    Call SysCurPos Old_R, Old_C;

    Return

GETHELP:/* ----- GETHELP: ----*/

    if \MUTE
    then do
        Topic1='GETHELP'
        Topic_String.Topic1.1=PRGN;
        Topic_List=Topic1;
        Call GETANS
    end
    else say 'Incorrect options.'

    Call QUIT

    Return

CHKFILE:/* ----- CHKFILE: ----*/

    Parse Arg FILE
    RC = Stream(FILE, 'C', 'QUERY EXIST')
    if RC = ''
    then do
        say ' File' FILE 'not found.'
        Call QUIT
    end
    else Nop
    Call Stream FILE, 'C', 'CLOSE'

    Return

LOGIT:/* ----- LOGIT: ----*/

    FUNC = ARG(1); INFO = ARG(2); RCODE = ARG(3)
    RC = LLOGIT(LOGF, PIPE, ADMNAME, PRGN, FUNC, INFO, RCODE)

    Return

QUIT:/* ----- QUIT: ----*/

    Call LineOut 'LSMT.END', PRGN, 1
    Call Stream 'LSMT.END', 'C', 'CLOSE'
    Call Stream COLF, 'C', 'CLOSE'
    Call Stream LOGF, 'C', 'CLOSE'
    Call Stream OUTF, 'C', 'CLOSE'

```

Exit

/*-----*/

ALIAS.INI

The Alais.ini is part of the LSMT package. GETALIAS.CMD uses the INI file to create the output file.

Usage

None

Source code

Example 9-35 ALIAS.INI

```
*****
* DO NOT CHANGE THE FIRST 2 COLUMNS ORDER
* AND DO NOT CHANGE THE COLUMNS NAMES
*

OPT          ; 3
NAME         ; 8
REMARK      ; 50
SERVER      ; 11
NETNAME     ; 8
LOCATION     ; 13
MODE        ; 17
MAXUSES    ; 7
TYPE       ; 7
QUEUE     ; 10
PATH      ; 45
PRIORITY  ; 8
DEVICE_POOL ; 12
*
*****
```

SETSMBDIRALIAS.CMD (Samba Only)

The SETSMBDIRALIAS.CMD code was used only for the Linux migration. We have written a simplified piece of code with no error checking to assist in a migration environment.

Usage

C:\OS2MIG\SETSMBDIRALIAS.CMD [SMBACL FILE] [OUTPUT FILE] [ALIAS FILE]

[ACL FILE] - Use the GETSMBACL.LOG from the GETSMBACL.CMD

[OUTPUT FILE] - The output file that will be used for the migration. In our case we named our output file setDirAlais.sh

[ALIAS FILE] - Use the GETALIAS.LOG from the GETALIAS.CMD

Source code

Example 9-36 SETSMBDIRALIAS.CMD source code

```
/**/
/* trace ?r */

/* getacl.log setDirAlais.sh getalias.log */
Parse Arg infile outfile getAliasFile

PM = 'perl modini.pl /etc/samba/smb.conf'
'@del 'outfile' 1>NUL 2>NUL'
do while Lines(INPFILE)
  LLINE = strip(TRANSLATE(LineIN(INPFILE)))
  OLINE = LLINE
  parse value LLINE with OPT ';' LLINE
  OPT = strip(OPT)

  select
    when OPT = '*' | LLINE = '' then Iterate
    when Left(OPT,1) = '*' then Iterate
    when OPT = 'OPT' then Call COLUMNS
    when OPT = 'A' then Call updDirACL
    otherwise nop
  end
end

Return

updDirACL:

  parse value LLINE with Alias ';' ACL.Audit ';' LLINE
  alias = strip(alias)
  ACL.Audit = strip(ACL.Audit)
  if ACL.Audit = '-NONE-' then ACL.Audit = 'N'
  ACLNum = 0
  ACP_Set. = 0
  i = 0
  readLst = ''
```

```

writeLst = ''
adminLst = ''
do while LLine <> ''
  i = i + 1
  UserGroup = UserGroupByCol.i
  parse value LLINE with ActACP ';' LLINE
  ActACP = strip(ActACP)
  if ActACP <> '' then
    do
      ACLNum = ACLNum + 1
      ACL.ACLNum.UGname = strip(UserGroup)
      ACL.ACLNum.access = translate(strip(ActACP))
      ACP_Set.UserGroup = 1

/*          say Alias ACLNum ACL.ACLNum.UGname ACL.ACLNum.access */

      if POS('G',ACL.ACLNum.access) > 0 then ACL.ACLNum.UGname =
'@' || ACL.ACLNum.UGname

      if POS('P',ACL.ACLNum.access) > 0 then
        adminLst = adminLst || ' ' || ACL.ACLNum.UGname

      if verify('RX',ACL.ACLNum.access) <> '1' then
        readLst = readLst || ' ' || ACL.ACLNum.UGname

      if verify('WCDA',ACL.ACLNum.access) <> '1' then
        writeLst = writeLst || ' ' || ACL.ACLNum.UGname

    end
  else nop
end

Call SysFileSearch alias, getAliasFile, 'aliasGet.'
Parse var aliasGet.1 waste ';' watste ';' waste ';' waste ';' pathLst ';'
waste

Call Lineout outFile, pm || ' SREMOVE "[" || alias || "]"'
Call Lineout outFile, pm || ' SADD "[" || alias || "]"'
Call Lineout outFile, pm || ' KADD "[" || alias || "]" "readlist" "'
||strip(readLst)|| ''
Call Lineout outFile, pm || ' KADD "[" || alias || "]" "writelst" "'
||strip(writelst)|| ''
Call Lineout outFile, pm || ' KADD "[" || alias || "]" "adminlist" "'
||strip(adminlst)|| ''
Call Lineout outFile, pm || ' KADD "[" || alias || "]" "comment" ""'
Call Lineout outFile, pm || ' KADD "[" || alias || "]" "path" "/shares/' ||
alias || ''

```

```

    Call Lineout outFile, pm || ' KADD '[' ||alias|| ']' "directory mask"
"0770"
    Call Lineout outFile, pm || ' KADD '[' ||alias|| ']' "dos file mode"
"0770"
    Call Lineout outFile, pm || ' KADD '[' ||alias|| ']' "nt acl support" "no"
    Call Lineout outFile, pm || ' KADD '[' ||alias|| ']' "security mask"
"0770"
    Call Lineout outFile, pm || ' KADD '[' ||alias|| ']' "case sensitive" "no"
    Call Lineout outFile, pm || ' KADD '[' ||alias|| ']' "public" "no"
    Call Lineout outFile, pm || ' KADD '[' ||alias|| ']' "writeable" "yes"
    Call Lineout outFile, pm || ' KADD '[' ||alias|| ']' "printable" "no"
    Call Lineout outFile, ''

    ACL.count = ACLNum

    Return

COLUMNS: /*----- COLUMNS: ----*/

    UGLst = OPT || ';' || LLINE

    parse value LLINE with . ';' . ';' LLINE

    i = 0
    do while LLine <> ''
        i = i+1
        parse value LLINE with UserGroupByCol.i ';' LLINE
        UserGroupByCol.i = strip(UserGroupByCol.i)
    end
    UserGroupByCol.0 = i /* UserGroupByCol.: user/group name retrieveable */
                        /* via col number */

    Return

GETANS: /*----- GETANS: ----*/

    Vars_List =Ansi_Say(Resource_File Topic_List);
    Parse VALUE SysCurPos() With Old_R Old_C;
    Do While Vars_List <> '';
        Parse VALUE Vars_List With Topic_Id ';' Var_Id ';' Row ';' Column ';'
Color ';' Vars_List;
        Call SysCurPos Row, Column;
        Say x2c(Color) || Topic_String.Topic_Id.Var_Id || '1B'x || '[0m';
    End;
    Call SysCurPos Old_R, Old_C;

    Return

```

SETSMBPRNALIASE.CMD (Samba Only)

The SETSMBPRNALIASE.CMD code was used only for the Linux migration. We have written a simplified piece of code with no error checking to assist in a migration environment.

Usage

C:\OS2MIG\SETSMBPRNALIASE.CMD [ACL FILE] [OUTPUT FILE] [ALIAS FILE]

[ACL FILE] - Use the GETACL.LOG from the GETACL.CMD

[OUTPUT FILE] - The output file that will be used for the migration. In our case we named our output file setPrnAlais.sh

[ALIAS FILE] - Use the GETALIAS.LOG from the GETALIAS.CMD

Source code

Example 9-37 SETSMBPRNALIASE.CMD source code

```
/**/
/* trace ?r */

/*      getacl.log setPrnAlais.sh getalias.log */
Parse Arg infile outfile getAliasFile

PM = 'perl modini.pl /etc/samba/smb.conf'
'@del 'outfile' 1>NUL 2>NUL'
do while Lines(INPFILE)
  LLINE = strip(TRANSLATE(LineIN(INPFILE)))
  OLINE = LLINE
  parse value LLINE with OPT ';' LLINE
  OPT = strip(OPT)

  select
    when OPT = '' | LLINE = '' then Iterate
    when Left(OPT,1) = '*' then Iterate
    when OPT = 'OPT' then Call COLUMNS
    when OPT = 'P' then Call updPrnACL
    otherwise nop
  end
end

Return

updPrnACL:

  parse value LLINE with Alias ';' ACL.Audit ';' LLINE
```

```

alias = strip(alias)
ACL.Audit = strip(ACL.Audit)
if ACL.Audit = '-NONE-' then ACL.Audit = 'N'
ACLNum = 0
ACP_Set. = 0
i = 0
readLst = ''
writeLst = ''
adminLst = ''
do while LLine <> ''
    i = i + 1
    UserGroup = UserGroupByCol.i
    parse value LLINE with ActACP ';' LLINE
    ActACP = strip(ActACP)
    if ActACP <> '' then
        do
            ACLNum = ACLNum + 1
            ACL.ACLNum.UGname = strip(UserGroup)
            ACL.ACLNum.access = translate(strip(ActACP))
            ACP_Set.UserGroup = 1

/*            say Alias ACLNum ACL.ACLNum.UGname ACL.ACLNum.access */

            if POS('G',ACL.ACLNum.access) > 0 then ACL.ACLNum.UGname =
'@' || ACL.ACLNum.UGname

            if POS('P',ACL.ACLNum.access) > 0 then
                adminLst = adminLst || ' ' || ACL.ACLNum.UGname

            if verify('RX',ACL.ACLNum.access) <> '1' then
                readLst = readLst || ' ' || ACL.ACLNum.UGname

            if verify('WCDA',ACL.ACLNum.access) <> '1' then
                writeLst = writeLst || ' ' || ACL.ACLNum.UGname

        end
    else nop
end

Call SysFileSearch alias, getAliasFile, 'aliasGet.'
Parse var aliasGet.1 waste ';' waste ';' waste ';' waste ';' pathLst ';'
waste

Call Lineout outFile, pm || ' SREMOVE "[" || alias || "]"'
Call Lineout outFile, pm || ' SADD "[" || alias || "]"'
Call Lineout outFile, pm || ' KADD "[" || alias || "]" "comment" ""'
Call Lineout outFile, pm || ' KADD "[" || alias || "]" "path"
"/shares/spooler/' || alias || ''

```



```

Call Lineout outFile, pm || ' KADD '[' || alias || ']' "browseable" "yes"
Call Lineout outFile, pm || ' KADD '[' || alias || ']' "printable" "yes"
Call Lineout outFile, pm || ' KADD '[' || alias || ']' "writeable" "no"
Call Lineout outFile, pm || ' KADD '[' || alias || ']' "guest ok" "yes"
Call Lineout outFile, ''

ACL.count = ACLNum

Return

COLUMNS: /*----- COLUMNS: ----*/

UGLst = OPT || ';' || LLINE

parse value LLINE with . ';' . ';' LLINE

i = 0
do while LLine <> ''
    i = i+1
    parse value LLINE with UserGroupByCol.i ';' LLINE
    UserGroupByCol.i = strip(UserGroupByCol.i)
end
UserGroupByCol.0 = i /* UserGroupByCol.: user/group name retrieveable */
/* via col number */

Return

GETANS: /*----- GETANS: ----*/

Vars_List =Ansi_Say(Resource_File Topic_List);
Parse VALUE SysCurPos() With Old_R Old_C;
Do While Vars_List <> '';
    Parse VALUE Vars_List With Topic_Id ';' Var_Id ';' Row ';' Column ';';
Color ';' Vars_List;
Call SysCurPos Row, Column;
Say x2c(Color) || Topic_String.Topic_Id.Var_Id || '1B'x || '[0m';
End;
Call SysCurPos Old_R, Old_C;

Return

```

SETWINSHARE.CMD (Windows only)

The SETWINSHARE.CMD code was used only for the Windows migration. We have written a simplified piece of code with no error checking to assist in a migration environment.

Usage

C:\OS2MIG\SETWINSHARE.CMD [INPUT FILE] [OUTPUT DIR FILE] [OUTPUT PRN FILE]

[INPUT FILE] - Use the GETALIAS.LOG from the GETALIAS.CMD

[OUTPUT DIR FILE] - Creates a command file to create directory shares

[OUTPUT PRN FILE] - Creates a command file to create printer shares

Source code

Example 9-38 SETWINSHARE.CMD source code

```
/* */
Parse Arg inFile outFileDir outFilePrt

inFile = Strip(inFile)
outFileDir = Strip(outFileDir)
outFilePrt = Strip(outFilePrt)

@del 'outFileDir' 1>NUL 2>NUL
@del 'outFilePrt' 1>NUL 2>NUL

Do While Lines(inFile)
  curLine = LineIN(inFile)
  orgLine = curLine
  Parse Value curLine With Opt ';' curLine
  Select
    When Opt = '' | curLine = '' | Left(Strip(Opt),1) = '*' Then Iterate
    When Translate(Opt) = 'OPT' Then Call GetColumns
    When Translate(Opt) = 'A' Then Call AddShare
    Otherwise Iterate
  End
End
Exit ExitCode
Return

/* -----*/
AddShare:
  i = 0
  Do While curLine <> ''
    i = i + 1
    columnName = Strip(columnNames.i)
    Parse value curLine With actValue ';' curLine
    share.columnName = Strip(actValue)
    If (share.columnName = "Unknown") | (share.columnName = "(null)") Then
share.columnName = ''
    End
```

```

    Call CreateCMD
Return

/* -----*/
GetColumns:
    i = 0
    Do While curLine <> ''
        i = i + 1
        Parse value curLine With columnNames.i ';' curLine
    End
    numColumn = i
Return

/* -----*/
CreateCmd:

    select
        when share.TYPE = 'Files' Then Do
            optional = ""
            if share.REMARK <> "" Then optional = optional || "/remark:" ||
share.REMARK || " "
            if share.MAXUSES <> 65535 Then optional = optional || "/users:" ||
share.MAXUSES || " "
            CALL LineOut outFileDir, "rmtshare \" || share.SERVER || "\" ||
share.NETNAME || "=" || share.PATH || optional
            end
        when share.TYPE = 'Printer' Then Do
            if share.REMARK <> "" Then optional = optional || '/remark:" ||
share.REMARK || ' ' '
            if share.MAXUSES <> 65535 Then optional = optional || "/users:" ||
share.MAXUSES || " "
            CALL LineOut outFilePrt, "rmtshare \" || share.SERVER || "\" ||
share.NETNAME || "=" || share.QUEUE || " " || optional
            end
        otherwise SAY share.NAME || ' skipped. Target does not support type ' ||
share.TYPE
    end

Return

```

SETWINCOPY.CMD (Windows Only)

The SETWINCOPY.CMD code was used only for the Windows migration. We have written a simplified piece of code with no error checking to assist in a migration environment.

Usage

C:\OS2MIG\SETWINCOPY.CMD [INPUT FILE]]

[INPUT FILE] - Use the GETALIAS.LOG from the GETALIAS.COMDares

Source code

Example 9-39 SETWINCOPY.CMD source code

```
/* */
Parse Arg inFile

inFile = Strip(inFile)

outFileDir = "rcopy.cmd"

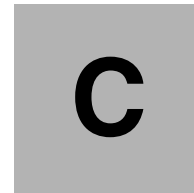
'@del 'outFileDir' 1>NUL 2>NUL'

Do While Lines(inFile)
  curLine = LineIN(inFile)
  orgLine = curLine
  Parse Value curLine With Opt ';' curLine
  Select
    When Opt = '' | curLine = '' | Left(Strip(Opt),1) = '*' Then Iterate
    When Translate(Opt) = 'OPT' Then Call GetColumns
    When Translate(Opt) = 'A' Then Call AddShare
    Otherwise Iterate
  End
End
Exit ExitCode
Return

/* -----*/
AddShare:
  i = 0
  Do While curLine <> ''
    i = i + 1
    columnName = Strip(columnNames.i)
    Parse value curLine With actValue ';' curLine
    share.columnName = Strip(actValue)
    If (share.columnName = "Unknown") | (share.columnName = "(null)") Then
share.columnName = ''
    End
    Call CreateCMD
  Return
/* -----*/
GetColumns:
  i = 0
```

```
Do While curLine <> ''
  i = i + 1
  Parse value curLine With columnNames.i ';' curLine
  End
  numColumn = i
Return

/* -----*/
CreateCmd:
  if share.TYPE = 'Files' Then Do
    CALL LineOut outFileDir, "robocopy \\OS2." || share.SERVER || "\" ||
share.NETNAME || " \\WIN." || share.SERVER || "\" || share.NETNAME || " /mir /z
/r:3 /w:30 /np /log+:rcopy.log"
  end
Return
```



Additional material

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

`ftp://www.redbooks.ibm.com/redbooks/SG246631`

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG246631.

Using the Web material

The additional Web material that accompanies this redbook includes the following files:

<i>File name</i>	<i>Description</i>
OS2SVR.zip	Files, tools and scripts described in book
README	A text file describing the content of the zip file and disclaimers on its use.

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder. Once unzipped, five folders will be created as described below:

<i>Folder name</i>	<i>Description</i>
Appx1	Folder containing source of files described in Appendix 1.
Appx2	Folder containing files described in Appendix 2, and modified versions of some LSMT scripts.
Ch4	Files and scripts described in Chapter 4., "Migrating OS/2 Servers to Windows 2000" on page 87.
LSMT	A zip file containing the files required for LSMT. Some of these files have been modified for this book and those modifications are in the Appx2 folder.
Tools	Folder containing the passwdsync and UAM tools as described in Chapter 4., "Migrating OS/2 Servers to Windows 2000" on page 87, and Chapter 8., "Additional migration tools" on page 277.

Attention: These files are provided on an "as is" basis and have not been thoroughly tested. Please use accordingly.

Abbreviations and acronyms

ACL	Access Control List	FTPD	File Transfer Protocol Daemon
ADS	Active Directory Services		
ADSM	Adstar Distributed Storage Manager	GB	Giga Byte
API	Application Program Interface	HPFS	High-Performance File System
AS/400	Application System/400®	HTTP	Hyper Text Transfer Protocol
ASCII	American Standard Code for Information Interchange	I/O	Input/Output
BDC	Backup Domain Controller	IBM	International Business Machines Corporation
BIOS	Basic Input Output System	IEEE	Institute of Electrical and Electronics Engineers
CHMOD	Change Mode	ITSO	International Technical Support Organization
CHOWN	Change Owner		
CID	Configuration, Installation, Distribution	JDK	Java Development Kit
CPU	Central Processing Unit	JRE	Java Runtime Environment
CRLF	Carriage Return Line Feed	KB	Kilo Byte
CSV	Comma-Separated Value/Variable	LAN	Local Area Network
DASD	Direct Access Storage Device	LDAP	Lightweight Directory Access Protocol
DB	DataBase	LDIF	LDAP Definition Input File
DCDB	Domain Controller DataBase	LPD	Line Print Daemon
DDNS	Dynamic Domain Naming System	LSMT	LAN Server Management Tools
DFS	Distributed File System	MB	Megabyte (1,024 kilobytes)
DHCP	Dynamic Host Configuration Protocol	MMC	Microsoft Management Console
DLL	Dynamic Link Library	MPTS	Multi Protocol Transport Services
DNS	Domain Naming System	MQ	Message Queueing
DOS	Disk Operating System	MS	Microsoft
EMEA	Europe, Middle East, Africa	MSI	Microsoft Software Installer
FAT	File Allocation Table	NAS	Network Attached Storage
FTP	File Transfer Protocol [Internet]	NetBEUI	NetBIOS Extended User Interface

NetBIOS	Network Basic Input/Output System
NFS	Network File System
NFSD	Network File System Daemon
NT	New Technology
NTFS	New Technology File System
OS/2	Operating System/2®
OU	Organizational Unit
PDC	Primary Domain Controller
REXX	Restructured Extended Executor
RPM	Red Hat Program Module
SAM	Security Accounts Manager
SAN	Storage Attached Network
SDK	Software Development Kit
SES	Security Enabling Services
SID	Security Identifier
SLES	Suse Linux Enterprise Server
SMB	Server Message Block
SQL	Sequential Query Language
SSL	Secure Socket Layer
SYSLOG	System Log
TB	Tera Byte
TCP/IP	Transmission Control Protocol/Internet Protocol
TSM	Tivoli Storage Manager
TTL	Time To Life
UAM	User Authentication Method
UDB	Universal DataBase
UNC	Universal Naming Convention
UNIX	AT&T Bell Laboratories Operating System
UPM	User Profile Management
USERID	User Identification
WINS	Windows Internet Naming Service [Microsoft]
WSoD	Workspace on Demand

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 564. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *Beyond DHCP*, SG24-5280-01
- ▶ *The OS/2 Warp 4 CID Software Distribution Guide*, SG24-2010-00
- ▶ *Migrating to OS/2 Warp Server for e-business*, SG24-5135-00
- ▶ *The OS/2 Warp 4 CID Rapid Deployment Tools Migration and Installation Scenarios*, SG24-2012-00
- ▶ *TCP/IP Implementation in an OS/2 Warp Environment*, SG24-4730-00
- ▶ *OS/2 Warp Server for e-business*, SG24-5393-00
- ▶ *Migration Options for OS/2 Warp Server for AS/400 and OS/400 Integration for Novell NetWare*, REDP-0020-00
- ▶ *Using Tivoli Storage Manager to Back Up Lotus Notes*, SG24-4534-02

Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ IBM Software Choice Web site
<http://www.software.ibm.com/os/warp/swchoice>
- ▶ IBM Business Integration Software
<http://www-3.ibm.com/software/integration>
- ▶ IBM Tivoli Storage Manager
<http://www-3.ibm.com/software/tivoli/products/storage-mgr/>
- ▶ IBM Object REXX
<http://www-3.ibm.com/software/awdtools/obj-rexx/>

- ▶ IBM Web Servers - IBM HTTP Server
<http://www-3.ibm.com/software/webservers/httpservers/>
- ▶ IBM Communications Server for OS/2 Warp
<http://www-3.ibm.com/software/network/commserver/downloads/enhancements/csos2.html>
- ▶ IBM Communications Server for Linux
<http://www-3.ibm.com/software/network/commserver/linux>
- ▶ IBM LanDP whitepaper
<http://www-3.ibm.com/software/network/landp/library/whitepapers.html>
- ▶ 6PAC Consulting AG
<http://www.6pac-ag.com>
- ▶ Titan Central
<http://www.titan-central.com>
- ▶ Lieberman and Associates
<http://www.lanicu.com/cross/lsnt/index.htm>
- ▶ Comtarsia Servolution
<http://servolution.comtarsia.com>
- ▶ WebMin
<http://www.webmin.com>
- ▶ Virtual Network Computing
<http://www.uk.research.att.com/vnc/index.html>
- ▶ HOBLink X11 for OS/2
http://www.hob.de/www_us/produkte/connect/X11-OS2.htm
- ▶ Citrix Metaframe
<http://www.citrix.com>
- ▶ Red Hat Linux Kickstart HOW-TO
<http://www.tldp.org/HOWTO/KickStart-HOWTO.html#toc6>
- ▶ SuSE Client Install HOW-TO
<http://www.tldp.org/HOWTO/Network-Install-HOWTO-5.html>
- ▶ Linux NFS HOW-TO
<http://www.ibiblio.org/pub/Linux/docs/HOWTO/NFS-HOWTO>
- ▶ Linux DNS HOW-TO
<http://www.ibiblio.org/pub/Linux/docs/HOWTO/DNS-HOWTO>

- ▶ The Linux Documentation Project
<http://tldp.org/docs.html>
- ▶ Linux Security
<http://www.linuxsecurity.com>
- ▶ NFS HOW-TO
<http://www.linux.org/docs/lpd/howto/NFS-HOWTO/server.html>
- ▶ DHCP Relay HOW-TO
<http://download.freeswan.ca/x509patches/dhcprelay/ipsec-dhcp-howto-4.html>
- ▶ DHCP Mini HOW-TO
<http://www.tldp.org/HOWTO/mini/DHCP/>
- ▶ OpenLDAP
<http://www.openldap.org>
- ▶ Berkeley DB
<http://www.sleepycat.com/download/index.shtml>
- ▶ SAMBA
<http://www.samba.org>
- ▶ Samba Project Documentation
<http://de.samba.org/samba/develop/docs/html>
- ▶ Kerberos
<http://web.mit.edu/kerberos>
- ▶ Hobbes OS/2 tools download site
<http://hobbes.nmsu.edu/cgi-bin/h-browse?sh=1&dir=//pub/os2/util/network/lan/srv>
- ▶ LDIF RFT
<http://www.ietf.org/rfc/rfc2849.txt>
- ▶ Microsoft Active Directory Branch Office Guide
<http://www.microsoft.com/technet/prodtechnol/ad/windows2000/deploy/adguide/default.asp>
- ▶ Microsoft Active Directory Services Interface
http://msdn.microsoft.com/library/en-us/netdir/ads/active_directory_service_interfaces_adsi.asp
- ▶ Microsoft 2000 Resource Kits
<http://www.microsoft.com/windows2000/techinfo/reskit/default.asp>
- ▶ Microsoft Step-by-step Guide to Dfs

<http://www.microsoft.com/technet/prodtechnol/windows2000serv/howto/dfsguide.asp>

- ▶ Microsoft disk limits - Best Practices

http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/entserver/sag_DQbest_practices.asp

- ▶ Understanding windows 2000 Disk Quotas

<http://www.techsupportalert.com/pdf/t1729.pdf>

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

Numerics

6PAC Consulting 133, 158, 232, 277, 301

A

access 77
access control 142
Access Control Entries (see ACE)
Access Control List (see ACL)
accountExpires user object 116
ACE 142
ACL 66, 77, 82, 91, 104, 142, 321, 332, 334
active connections 397
Active Directory 19–20, 35, 43, 85, 89, 155, 191, 345, 348
 preparation 100
 Scripting Interface 166
 user objects 118
activities 285, 287, 300
activity package 287
AD4UNIX 160
 schema master 160
administration of Linux 373
ADSI (see Active Directory Scripting Interface)
Adstar Distributed Storage Manager 9, 28, 273
agents 284, 286
ALIAS.INI 545
aliases 222–223, 278, 321, 330, 336
anaconda 29
anti-virus 12
Apache 8
application definitions 80, 82
applications 177, 267, 309
apply API 66
APPN 180
ASCII files for LSMT 65
assignments 132, 137
authentication 104, 114
autoinst.xml 36
autoyast 36

B

backup 150

backup security 374
baseou.ldif file 101, 449
BASH shell 392
basic user object
 accountExpires 116
 logonHours 117
 primaryGroupid 117
 userAccountControl 117
before migrating
 DOMAINSCOPE 16
 ENABLEDNS 15–16
 NetBIOS name resolution 15
 NetBIOS name server 15–16
 printing 13
 RFC coded names 16
 TCPBEUI 13, 15
 virus protection 12
 WINS 16
Berkeley DB 52–53
bind-9.1.x 42
branchou.ldif file 101, 451
broadcast mode 15
building Samba 55
building Berkeley 53
building OpenLDAP 54

C

cacls command 145–146
checkos command 133, 467
chmod command 393
CID 21, 311, 412
Citrix Metaframe 91, 158, 310
CMLIB 181
cmrecord 181
Communications server 7, 27, 44, 180
 MIGRATE.RSP 182
 migration to Linux 270
 migration utility 181
 PROTOCOL.INI 182
Comtarsia 158, 277
Comtarsia schema 350, 366
controller 284, 286, 292, 299
CreateUser.vbs 454

- creating partitions 389
- crontab 394
- CS.ISS 429
- CS/Linux 44, 47
 - installing 48
- CSADMINS group 27
- CSV files 67
- csvde command 115
- customizing the Linux server 392

D

- daemon 381
- DASD limits 82, 152–153, 226
- data
 - extracting 64
 - importing 64
 - manipulating 64
- data migration 332
- DB2 178, 268
 - (see also Universal Database)
 - DB2 Connect 179, 269
 - db2move 179, 269
 - delimited ASCII 179, 269
 - for OS/2 7
 - installation 44
 - moving data across platforms 178, 268
 - PC/IXF 178, 268
 - Windows 44
 - WSF 179, 269
- DB2.RSP 435
- db2cc 44
- DCDB 104–106, 120, 217–218, 301
- dcpromo command 23, 105, 416
- DCPROMO1.TXT 427
- DCPROMO2.TXT 428
- DDNS server 171, 257
 - DDNS.DAT 172
 - dnsex.t.cfg 173
 - dnst0000.dom 174
 - dnst0000.rev 174
 - for OS/2 7
 - named.bt 172
 - NAMEDB 172
- defined servers 279
- deleting partitions 389
- DES hash 314
- dest.acg file 182
- DFS 106, 302

- DHCP server 23, 35, 43, 167, 257
 - DHCPD.CFG 168
 - for OS/2 7
- dhcpd.conf file 260
- directory service 161
- DLUR 180
- DNS 15
- DNS daemon 34
- DNS server 16, 23, 34, 42, 171
- dnsex.t.cfg file 173
- domain 68
 - controller 104
 - definitions 316
 - migration 100
 - structure 100
- Domain Name Services (see DNS)
- domain structure 90
- DOMAIN.ICU file 314
- DOMAINSCOPE 16
- Domino server 3, 8, 27, 49, 270, 345
 - installation 49
 - on Linux 376
 - securing 376
 - starting automatically 384
 - stopping automatically 384–385
- DOSEMU 10
- dsm.opt file 273
- DWC 180, 270
- Dynamic DNS (see DDNS server)

E

- e-business xix
- ENABLEDNS 15–16
- e-Network Communications server 3, 7, 27, 44, 180
 - MIGRATE.RSP 182
 - migration to Linux 270
 - migration utility 181
 - PROTOCOL.INI 182
- Enterprise Extender 180
- environment 114
- event viewer 281
- export file 314
- exports file 159, 162
- extracting data 64

F

- fdisk command 388
- file

- aliases 321, 330
- permissions 375
- shares 78, 148
- sharing protocol 382
- systems 391
- find command 393
- firewall tool 382
- formatting partitions 391
- FTP
 - Hummingbird InetD 165
 - Internet Information Server 165
 - TCPNBK.LST 164
- ftp daemon 34, 41
 - for OS/2 6
- FTP server 22, 29, 33, 41, 271, 382
- FTPAccess group 166

G

- getacl command 77
- getalias command 78, 539
- getall command 66
- getappl command 80
- getgrps1 command 72, 109, 496
- getgrps1.log 113
- getgrps1.log file 109
- getgrps2 command 72, 504
- getpwd command 74, 524
- getsmbacl command 529
- getsvr command 69, 489
- getusers command 66, 74, 512
- getwinacl command 144, 471, 537
- gidNumber 202
- GPO (see group policy objects)
- group policy object 26, 91
- group-db.csv 112–113, 453
- groups 72, 91, 113, 127, 203, 321
 - id 202
- GROUPS.INI 501
- groups.ldif file 452
- grpmem.ldif 463

H

- hash 347
- hash format 164, 210
- HOBLINK X11 for OS/2 406
- hot fix 315
- HPR 180
- HTTP server 3, 8, 28–29, 50, 184, 272

- httpd.conf 185
- installation 50
- modules 185
- httpd.conf file 185
- Hummingbird 165
 - Maestro NFS Server 8.0 160

I

- IBM middleware 19
- IBMLAN\$ 104, 217
- IBMLAN.INI 13–14
- IBMLOGON 279
- identification 114
- IIS (see Internet Information Server)
- IMPORT.EXE 316
- importing data 64
- INI file for LSMT 66
- installed features for OS/2 4
- INSTCERTSRV.TXT 428
- INSTDHCP.TXT 426
- INSTDNS.TXT 426
- INSTFTP.TXT 426
- INSTWINS.TXT 426
- INSTWWW.TXT 427
- Internet Information Server 22, 165
 - user directories 166
 - virtual directory 166
- interrogators 286, 290
- intrusion 12
- IP traffic 398
- IPTraff utility 395, 398

J

- Java 28, 44
- JDK 28, 184
- jobs 286, 299
- JVM 283

K

- Kerberos 108, 379

L

- LAN Intensive Care Utilities 314
- LAN Manager authentication 104
- LAN Server Management Tools (see LSMT)
- LANDP 9
- LDAP xx, 11, 26, 35, 43, 56–58, 61, 90–91,

- 160–161, 190, 202–203, 206, 210, 213, 218, 281, 295, 301, 308, 345–346
- ldapmodify command 194, 219
- LDIF 110–111, 122, 125, 127, 194, 202, 210
- ldif.err file 101, 113
- ldif.log file 101, 113
- ldifde command 100–101, 115–116, 140
- Lieberman 277, 312
- Linux
 - administration 373, 388, 403
 - migrating to 187
 - partitions 388
 - security 374
 - streams 46
 - Web server 382
- Linux daemons 382
 - ftpd 382
 - httpd 382
 - lpd 382
 - nfs 382
 - sendmail 382
 - snmpd 382
 - ssh 383
 - syslog 383
 - telnet 383
 - wu-ftp 383
 - xfs 383
 - xinetd 383
- listing partitions 389
- listing scheduled tasks 394
- local groups 279
- logging 401
 - configuration file 402
 - events 401
 - log files 402
 - log information in real time 402
 - starting automatically 401
 - system messages 402
- logon assignments 132, 137
- logon client 158
- logon command 133, 137
- logon script manager 133, 302
- logon scripts 133, 321, 337
- logon services for OS/2 clients 105, 217
- logon.cmd file 465
- logonHours user object 117
- Lotus Domino server 3, 8, 27, 49, 270, 345
 - installation 49
- LPD 155
- ls command 375
- LSM 305
- LSMT 64, 109, 125, 148, 156, 202, 210, 213, 218, 223, 367
 - access 77
 - application 80
 - ASCII files 65, 67
 - domain 68
 - file and printer shares 78
 - groups 72
 - installing 65
 - serial devices 79
 - servers 69
 - users 74
- LSMT command
 - getacl 77
 - getalias 78
 - getall 66
 - getappl 80
 - getgrps1 72, 109
 - getgrps2 72
 - getpwd 74
 - getsvr 69
 - getusers 66, 74
- LSMT INI file 66
- LSMT.RSC 483
- LSMUSE.EXE 307
- LSRXUT.DLL 66
- LSRXUTIL.DLL 138

M

- management console 22
- management protocol 382
- manipulating data 64
- MD4 hash 314
- member server 104, 107
- Microsoft installer package 28
- Microsoft Management Console 22, 303, 310
- middleware 19
- miffs 34
- MIGRATE.RSP 182
- migrating DASD limits 152
 - Microsoft quota services 152
 - Precise/StorageCentral SRM 153
 - Quota and File Sentinel 153
- migrating data 178, 268
- migrating data to Windows
 - backup and restore programs 150

- robocopy 150
- XCOPY command 150
- migrating directories 471
- migrating file shares 148
 - rmtshare 148
 - setwinshare 149
 - setwinshares.cmd 148
- migrating to Linux 187
- migrating to Windows 2000 85
- migrating users to Windows
 - access control 142
 - csvde 123
 - LDIFDE 123
 - ldifde 123
 - logon assignments 132, 137
 - logon script 133
 - passwdsync 130
 - passwords 129
 - pwdexp 130
 - pwdimp 130
- Migration and Synchronization Wizard 312
- monitoring
 - IP traffic 398
 - network traffic 395
- mounting
 - a partition 392
- multiple NetBIOS names 201

N

- name resolution 15
- name server switch 57
- named.bt file 172
- named.conf file 259
- NAMEDB 172
- names
 - NetBIOS 15, 108
- native mode 106
- NBNS (see NetBIOS name server)
- NCB 15
- NET ACCESS 145
- NET TIME command 200
- net user
 - authentication 114
 - environment 114
 - identification 114
 - statistics 114
- net user command 114
- NET.ACC file 210

- NetApp 232, 277
 - suite 158
- NetBIOS
 - name resolution 15
 - name server 15–16
 - names 15, 108, 199, 201
- NetBIOS over TCP/IP 15, 350
- netdom.exe 107
- Netfinity Manager for OS/2 11
- NETLOGON command 305, 309
- NetRun service 200
- netsh command 170
- netstat command 385, 395
- NetviewDM 310
- network
 - application toolset 309
 - applications 309, 350
 - control block 15
 - printer 155
 - protocols 4
 - security 374, 381
 - statistics 398
 - status 395
 - traffic 395
- new user objects 115
- NFS 159
 - exports 159
 - Hummingbird exports file 162
 - TCPNBK.LST 160
- NFS daemon 34, 41
 - for OS/2 6
- NFS server 29, 34, 41, 161, 226, 232, 271
- NIS 160
- notes.ini file 271
- NOTES.ISS 430
- NTFS 21, 166
- NTP server 200

O

- ObjectREXX 52
- oocmigcm command 182
- OpenLDAP 51–54, 56, 58, 194
 - building 54
- OpenMOTIF 46
- organizational units 90–91, 190, 198, 295, 358
- OS/2
 - base installation 4–7, 9, 11
 - Communications server 7

- DB2 7
- DDNS server 7
- DHCP server 7
- ftp daemon 6
- LANDP 9
- Netfinity 11
- network protocols 4
- NFS daemon 6
- partition table 4
- public applications 6
- resources 5
- TCP/IP 13
- Tivoli Storage Manager 9
- user accounts 5
- user objects 118
- OS2ENV.CMD 468
- OS2LDAPMigration tool 354
- OS2MigrateUsers 363
- OU (see organization units)

P

- package manager 403
- PART.TXT 420
- partition table for OS/2 4
- partitions 389
 - creating 389
 - deleting 389
 - formatting 391
 - Linux 388
 - listing 389
 - mounting 392
 - swap 391
 - types 390
- passwdsync command 140
- password hash 164, 210
- password synchronization 130
- password synchronization tool 281
- passwords 129, 374
- patch 314
- PDC
 - emulator 105
- persistent 279
- persistent local groups 279
- persistent users 279
- physical security 374
- policy files 350
- policy object 26
- ports 382

- POST1.CMD 414
- POST2.CMD 415
- Primary Domain Controller 317
- primary logon client 309
- primaryGroupid user object 117
- print queue 228
 - options 156
 - share 156–157
- PRINT\$ 155
- PRINTDRV 228
 - share 155
- printer
 - aliases 321, 336
 - migration 13
 - network 155
 - shares 78
- printers 155
- printing 382
- product stack in sample environment 3
- PROFILE.CMD 105–106, 120, 218
- PROTOCOL.INI 14–15, 182
- public applications 82
 - for OS/2 6
- pwdexp command 130, 282
- pwdimp command 130

R

- RACF 346
- rcopy command 154
- Red Hat 19, 45, 51, 233
 - authentication configuration 380
 - daemons 385
 - ftp 383
 - services 385
 - setting password length 379
 - system services 401
- Redbooks Web site 564
 - Contact us xxiv
- registry size 315
- relative identifier 117
- remote administration 383
- replicate to 339
- replicator service 200
- repository 286, 288
- resolve phase 320
- resources 375
 - for OS/2 5
- response files 417

- restore 150
- REXX 64, 66–68, 82, 110, 122
- RFC coded names 16
- RGLSRXUT.CMD 480
- RGUTIL.CMD 478
- RGUTILS.CMD 479
- RID 117
- rmtshare command 148, 157
- rndc.conf file 259
- robocopy command 150, 152–153
- root access 374
- root certificate server 26
- RSA encryption 348
- rsh command 200
- runlevels 384
- running daemons on demand 383

S

- sAMAccountName 110
- Samba xx, 19, 51–52, 59, 187, 199, 201, 203, 206, 209, 218, 220, 222, 345, 348
 - aliases 222
 - building 55
 - configuring 200
 - domain configuration 199
 - Schema 350
 - schema 56
 - server 35
 - shares 222, 228, 362
- Samba server 43
- sambaSamAccount object 206
- sample scenario 3
- SAN 11
- scheduled tasks 394
- scheduler 394
- secure shell
 - remote administration 383
- securing the Domino server 376
- security 12, 374
 - active connections 397
 - advanced network security 387
 - backup 374
 - backup security 387
 - daemon 381
 - file permissions 375
 - Kerberos authentication 379
 - MD5 passwords 380
 - network 374, 381

- open ports 382
- passwords 374, 376
- physical 374
- ports 382
- power-on password 374
- root access 374
- running daemon as root 382
- running Domino server in Linux 376
- securing your backups 387
- setting password length 378–379
- setting permissions 393
- shadow passwords 380
- system 374
- user access 376
- Security Accounts Manager 126
- security identifier 117
- serial devices 79, 82, 232
- SERVER.INI 494
- servers 69
- SERVICE.CMD 413
- services 394
- Servolution 158, 232, 277
 - Logon Client 347, 361
 - migration path 345
- setdiralias.sh command 223
- setgroups command 111, 113, 203–204, 206, 501
- setgrpmem command 127, 140, 214, 510
- setprnalias.sh command 231
- setsmbdiralias command 223, 545
- setsmbprnalias command 230–231, 549
- setting permissions 393
- setusers command 140, 210, 218, 519
- setwinacl command 146, 153, 472
- setwincopy command 150, 153, 475, 553
- setwinshare command 157, 473, 551
- setwinshare-print command 156
- setwinshare-printer command 158
- setwinuserasn command 138, 468
- SFU (see Windows services for UNIX)
- shadow passwords 380
- shares
 - file 78
 - printer 78
- shell
 - bash 392
- shell for remote administration 383
- showexp command 163
- showmount command 163
- SID 117

- smb.conf file 200, 229–231, 357, 362
- SMBLDAP tool 52
- SMTP server 382
- SNA 44, 180
- Software Choice 8
- source.rsp file 182
- ssh command 200, 359, 386
- SSL 47, 50, 61
- Starfire 277, 283
- starting daemons 383
- statistics 114, 398
- stopping daemons 383
- Storage Area network 11
- SuSE 19, 36, 45
 - creating a new group 377
 - creating a user 377
 - ftp daemon 382
 - MD5 encryption 377
 - password settings 376
 - setting password length 378
 - starting daemons 383
 - stopping daemons 383
- swap partitions 391
- switchrx command 120
- SyncAgent 347, 354, 359, 361, 365–366
- SyncClient 361
- SyncPacket 348, 355
- SyncProxy 347, 361
- sysocmgr command 415
- system logs 401
- system security 374
- SysV Init Editor 384

T

- target platforms 19
- TCP/IP 13, 395
- TCPBEUI 13, 15
- TCPNBK.LST file 160, 164
- timesouce service 200
- Titan 277, 283
 - activities 285, 287, 300
 - activity package 287
 - agents 284, 286
 - controller 284, 286, 292, 299
 - interrogator 290
 - interrogators 286
 - jobs 286, 299
 - repository 286, 288

- tools 286
- TitanScript 286
- Tivoli Software Distribution 310
- Tivoli Storage Manager 9, 226, 273
- Tivoli Storage Manager client 3, 28, 51
- tools 286
- tools and scenarios 265
- transform.group file 203–204
- transform.group.file 488
- transform.user file 488
- troubleshooting 400
 - network 400
- types of partitions 390

U

- unattended installation 21
- Unattended Installation Manager 310–311
- UnitedLinux 45
- Universal Database 3, 26, 44, 178, 268
 - (see also DB2)
 - migration 178
- user
 - accounts 320
 - user access 376
 - user accounts for OS/2 5
 - User Accounts Manager 278
 - user authentication 104
 - user directories 166
 - user objects 115
 - accountExpires 120
 - ACCT_EXPIRES 120
 - CODE_PAGE 119
 - codePage 119
 - COMMENT 118
 - COUNTRY_CODE 119
 - countryCode 119
 - description 118
 - displayName 118
 - dn 118
 - FLAGS 119
 - givenName 118
 - HOME_DIR 120
 - homeDirectory 120
 - homeDrive 120
 - LOGON_HOURS 119
 - logonHours 119
 - MAX_STORAGE 119
 - maxStorage 119

- NAME 118–119
- pwdLastSet 118
- sAMAccountName 119
- SCRIPT_PATH 120
- scriptPath 120
- sn 118
- unicodePwd 118
- userAccountControl 119
- userPrincipalName 118
- userWorkstations 119
- USR_COMMENT 118
- WORKSTATIONS 119
- userAccountControl user object 117
- useradd command 240–241
- users 74, 123, 129, 140, 210, 218, 240–241
 - persistent 279
- USERS.INI 518
- users.ldif file 459

V

- virtual directory 166
- virus protection 12
- virus scanner 13
- Visual Scripting Host 121
- vs-ftpd 41, 235

W

- W2K.CMD 414
- W2K_INST.REG 420
- Warp Server for e-business xix
- Warp Server for e-business (see OS/2)
- Webmin utility 403
- WebSphere 50
- WebSphere Business Component Composer 9
- WebSphere MQ 10
- WINDC.TXT 417
- WINDCP.TXT 420
- Windows 2000 85
 - domain controller 104
 - event viewer 281
 - member server 104
 - server 20
- Windows 2000 Server 20
- Windows Active Directory 35, 43
- Windows Services for Unix 160
- Windows services for Unix 162
 - exports 163
 - nfsshare.exe 163

- NIS 162
 - PCNFS 162
- WINMEMP.TXT 423
- WINS 15–16, 23
- workplace shell 309
- Workspace On-Demand 290
- wu-ftpd 33, 235
- WYNAND.CMD 470

X

- XACT-SMB 107
- XCOPY command 150, 226
- XML 286–287, 291

Y

- yast2 36, 241, 376

IBM



Redbooks

OS/2 Server Transition

(1.0" spine)
0.875" <-> 1.498"
460 <-> 788 pages



OS/2 Server Transition



Redbooks

Extract data from OS/2 Servers

Migrate OS/2 domains to Windows 2000

Migrate OS/2 domains to Linux with Samba 3.0

OS/2 Servers have been a stable and powerful platform more many years and are depended upon by many businesses. This is especially true in the banking industry where OS/2 servers are trusted to run the software that supports the branch office environment. However, as the industry looks to renew its branch office operations, many banks are looking to make a transition from their OS/2 Server to a platform with wider industry support.

The two logical target platforms for this transition are a Microsoft or Linux solution. Both of these solutions have there own benefits and costs.

This IBM Redbook provides a technical guide for OS/2 administrators to help them plan for and implement a transition to what ever platform is right for their business. Using this redbook as a guide, OS/2 administrators and technical personnel will be able to develop and implement a plan for a smooth transition from their current OS/2-based domains to either a Microsoft Windows and Active Directory solution, or a Linux-based solution utilizing LDAP and SAMBA V3.0 for file and print sharing, or in many cases, a mixed environment containing both platforms.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks