

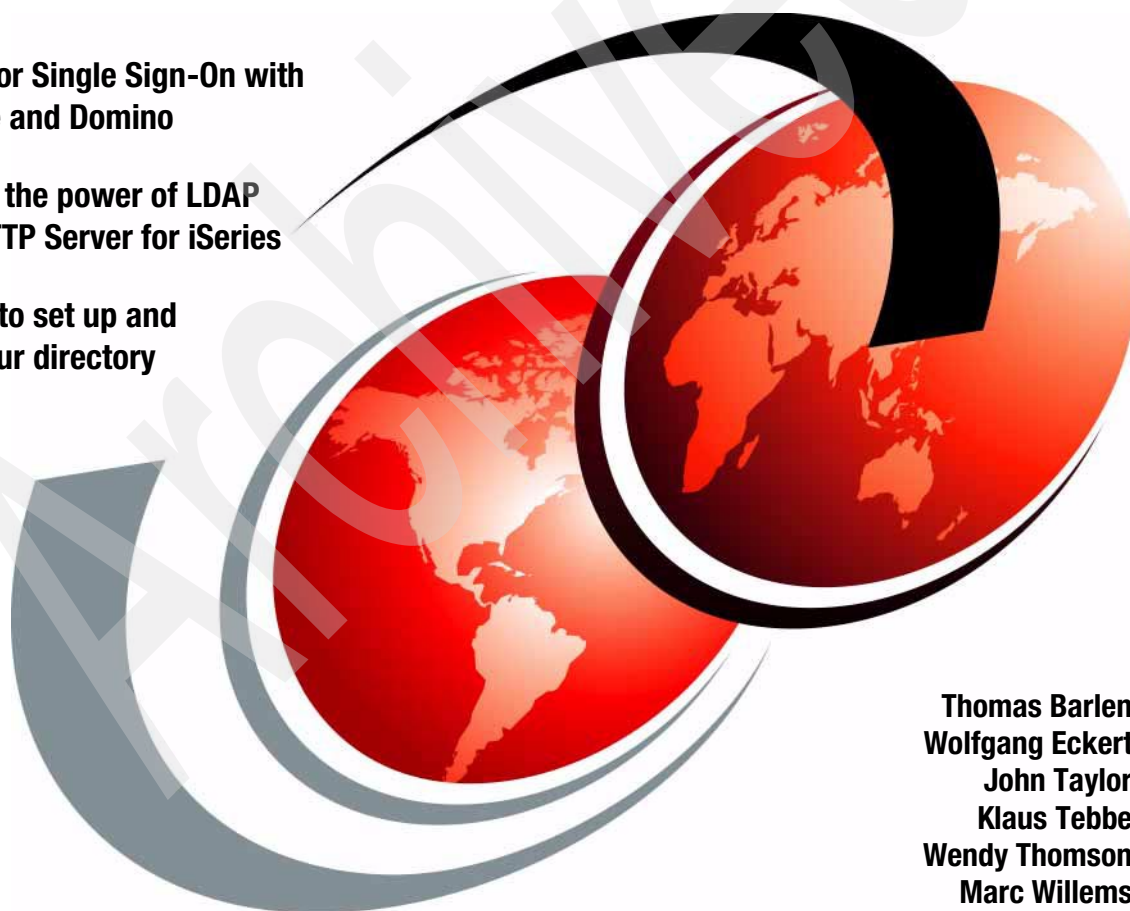
# Implementation and Practical Use of LDAP

on the IBM  iSeries Server

Use LDAP for Single Sign-On with  
WebSphere and Domino

Experience the power of LDAP  
with the HTTP Server for iSeries

Learn how to set up and  
manage your directory



Thomas Barlen  
Wolfgang Eckert  
John Taylor  
Klaus Tebbe  
Wendy Thomson  
Marc Willems





International Technical Support Organization

**Implementation and Practical Use of LDAP on the  
IBM @server iSeries Server**

April 2002

Archived

**Take Note!** Before using this information and the product it supports, be sure to read the general information in “Notices” on page xi.

### **First Edition (April 2002)**

This edition applies to Version 5 Release 1 Modification 0 of the Operating System/400 - 5722-SS1.

Comments may be addressed to:  
IBM Corporation, International Technical Support Organization  
Dept. JLU Building 107-2  
3605 Highway 52N  
Rochester, Minnesota 55901-7829

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2002. All rights reserved.

Note to U.S Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.



# Contents

<b>Notices</b> .....	xi
Trademarks .....	xii
<b>Preface</b> .....	xiii
The team that wrote this redbook .....	xiv
Comments welcome .....	xvi
<b>Part 1. Introduction</b> .....	1
<b>Chapter 1. Directory concepts</b> .....	3
1.1 Directories .....	5
1.2 Advantages of using a directory .....	9
1.3 Directory components .....	11
<b>Chapter 2. Planning your directory</b> .....	17
2.1 Defining the directory content .....	20
2.1.1 Defining directory requirements .....	20
2.2 Data design .....	21
2.2.1 Sources for data .....	22
2.2.2 Characteristics of data elements .....	22
2.2.3 Related data .....	23
2.3 Organizing your directory .....	23
2.3.1 Schema design .....	23
2.3.2 Namespace design .....	25
2.3.3 Naming style .....	27
2.4 Securing directory entries .....	28
2.5 Designing your server and network infrastructure .....	32
2.5.1 Availability, scalability, and manageability requirements .....	33
2.5.2 Topology design .....	33
2.5.3 Replication design .....	36
2.6 Implementation planning .....	37
<b>Chapter 3. The redbook example scenario</b> .....	39
3.1 Scenario overview - Stage 1 .....	40
3.1.1 Stage 2 - The evolution .....	41
3.1.2 Stage 3 - Growing the business .....	43
3.1.3 Stage 4 - Merging businesses .....	45
3.1.4 Stage 5 - The enterprise directory .....	48
3.1.5 The scenario Directory Information Tree .....	49

<b>Part 2. Configuration and administration</b>	<b>53</b>
<b>Chapter 4. OS/400 LDAP Directory Services</b>	<b>55</b>
4.1 Implementation and component overview	57
4.1.1 OS/400 Directory Services jobs	59
4.1.2 Saving and restoring Directory Services information	61
4.2 Installation prerequisites	62
4.2.1 Hardware requirements	62
4.2.2 Software requirements	62
4.3 Configuring OS/400 Directory Services	64
4.3.1 First-time configuration	64
4.3.2 Reconfiguring Directory Services	74
4.4 Publishing system information	75
4.4.1 Publishing OS/400 system information	76
4.5 Publishing the System Distribution Directory	79
4.5.1 System Distribution Directory to LDAP mapping	79
4.5.2 Scenario objectives	81
4.5.3 Setting up SDD publishing	81
4.5.4 Excluding entries from being published	91
4.6 Publishing printer information	92
4.7 Setting up directory replication	106
4.7.1 Stage 2 set-up scenario	106
4.8 Setting up directory referrals	122
4.8.1 Specifying a default server for directory referrals	123
4.8.2 Creating explicit directory referral entries	125
4.9 Securing LDAP traffic	126
4.9.1 When to secure what service	127
4.9.2 Installation prerequisites	127
4.9.3 Scenario characteristics	128
4.9.4 Enabling SSL for the LDAP server	128
4.9.5 Enabling SSL for the Directory Services publishing client	138
4.9.6 Enabling SSL for the Directory Services client	143
4.10 Directory auditing support	143
4.10.1 Setting up auditing for Directory Services	144
4.10.2 Audit entry type	146
<b>Chapter 5. Managing an LDAP directory on iSeries</b>	<b>149</b>
5.1 Different ways to manage your directory	150
5.2 Using the DMT to manage the directory	150
5.2.1 Directory Management Tool installation	151
5.2.2 Connecting to the LDAP server	152
5.2.3 Setting up the DMT for using SSL	158
5.2.4 Adding organization entries using the DMT	165

5.2.5	Using the DMT to browse the directory schema . . . . .	170
5.2.6	Using the DMT to view the directory change log . . . . .	175
5.3	Using LDAP utilities to manage the directory . . . . .	176
5.3.1	The ldapadd and ldapmodify utilities . . . . .	176
5.3.2	The ldapsearch utility . . . . .	181
5.3.3	The ldapdelete utility . . . . .	184
5.3.4	The ldapmodrdn utility . . . . .	185
5.3.5	Using LDAP utilities to view the directory change log . . . . .	185
5.4	Exporting and importing information via Operations Navigator . . . . .	187
5.4.1	Exporting directory data . . . . .	188
5.4.2	Importing directory data . . . . .	192
5.5	Writing your own application to manage your directory . . . . .	195
5.6	Accessing directory information . . . . .	195
5.6.1	Searching the directory with the ldapsearch utility . . . . .	195
5.6.2	Searching the directory from a browser . . . . .	195
5.6.3	Searching the directory with applications . . . . .	198
5.6.4	Searching the directory with your own applications . . . . .	198
5.7	Controlling access to directory entries . . . . .	198
5.7.1	How does access control work? . . . . .	199
5.7.2	Implementation tasks summary . . . . .	200
5.7.3	Creating an access control list (ACL) group . . . . .	201
5.7.4	Adding the ACL group to the entry's ACL . . . . .	205
5.7.5	Enabling the directory for attribute-level permissions . . . . .	217
5.7.6	Editing object and attribute authorities . . . . .	219
5.7.7	Changing the protection level for the userPassword attribute . . . . .	225
<b>Chapter 6.</b>	<b>IBM HTTP Server for iSeries LDAP support . . . . .</b>	<b>229</b>
6.1	Introduction . . . . .	230
6.2	Scenario characteristics . . . . .	231
6.2.1	Prerequisites . . . . .	232
6.3	User authentication . . . . .	232
6.3.1	Setting up LDAP authentication for the Original server . . . . .	233
6.3.2	Setting up LDAP authentication for the powered by Apache server . . . . .	244
6.4	Configuration support . . . . .	254
6.4.1	Setting up the LDAP configuration support for the Original server . . . . .	255
6.4.2	LDAP configuration support for powered by Apache server . . . . .	270
<b>Chapter 7.</b>	<b>Setting up LDAP on Domino server for iSeries . . . . .</b>	<b>291</b>
7.1	Domino Directory implementation and components . . . . .	292
7.1.1	The role of Directory Services in Domino . . . . .	292
7.1.2	The Domino Directory . . . . .	294
7.1.3	Directory Catalog . . . . .	295
7.1.4	Directory Assistance . . . . .	297

7.1.5	Different ways to do directory searches . . . . .	299
7.1.6	Authentication for LDAP clients. . . . .	305
7.2	LDAP on Domino in the redbook scenario . . . . .	306
7.2.1	Scenario characteristics . . . . .	306
7.2.2	Scenario objectives. . . . .	307
7.2.3	Installation prerequisites . . . . .	307
7.3	Configuring LDAP for Domino . . . . .	308
7.3.1	Setting up the Domino server for LDAP . . . . .	308
7.3.2	Changes to the LDAP setup on Domino . . . . .	310
7.3.3	Configuration changes to the Domino LDAP services . . . . .	313
7.4	Using LDIF to exchange directory information . . . . .	313
7.4.1	Scenario objectives. . . . .	313
7.4.2	Exporting LDIF data from the Domino Directory . . . . .	314
7.4.3	Importing LDIF data into an iSeries LDAP . . . . .	315
7.4.4	Exporting LDIF data from iSeries LDAP . . . . .	317
7.4.5	Importing LDIF data into the Domino Directory. . . . .	318
7.5	Connecting directories: The alternative. . . . .	326
7.5.1	Scenario objectives. . . . .	326
7.5.2	Creating the Directory Assistance database . . . . .	327
7.5.3	Setting up directory referrals. . . . .	328
7.5.4	Deploying Directory Assistance and referrals in your domain. . . . .	333
<b>Part 3.</b>	<b>Practical scenarios . . . . .</b>	<b>335</b>
<b>Chapter 8.</b>	<b>Single Sign-On with Domino and WebSphere 4.0 . . . . .</b>	<b>337</b>
8.1	Scenario characteristics . . . . .	338
8.2	Scenario objectives . . . . .	338
8.3	Prerequisites . . . . .	339
8.3.1	Workstation requirements . . . . .	340
8.3.2	iSeries requirements . . . . .	340
8.4	Task summary. . . . .	341
8.5	Single Sign-On security concepts . . . . .	341
8.6	Enabling WebSphere Application Server authentication. . . . .	343
8.6.1	WebSphere Application Server authentication concepts . . . . .	344
8.6.2	Configuring WebSphere Application Server security . . . . .	349
8.6.3	Protecting WebSphere resources . . . . .	354
8.6.4	Verifying WebSphere Application Server authentication. . . . .	366
8.7	Configure Domino HTTP server . . . . .	368
8.8	Enabling Domino authentication . . . . .	370
8.8.1	Domino authentication concepts. . . . .	370
8.8.2	Configure Domino to use iSeries LDAP . . . . .	371
8.8.3	Activating Domino Database security . . . . .	378
8.8.4	Verifying Domino authentication . . . . .	380

8.9	Configuring Single Sign-On . . . . .	381
8.9.1	SSO prerequisites . . . . .	382
8.9.2	Setting up SSO for WebSphere Application Server . . . . .	383
8.9.3	Setting up SSO for Domino . . . . .	389
8.9.4	Verifying Single Sign-On . . . . .	397
8.10	Enabling SSL with SSO . . . . .	402
8.10.1	Enabling WebSphere SSL with SSO . . . . .	402
8.10.2	Enabling Domino SSL with SSO . . . . .	406
8.10.3	Testing SSO between Domino and WebSphere using SSL . . . . .	426
<b>Chapter 9.</b>	<b>LDAP directory: The enterprise directory for mail clients . . . . .</b>	<b>433</b>
9.1	Scenario characteristics . . . . .	435
9.2	Scenario objectives . . . . .	435
9.3	Scenario network and system environment . . . . .	435
9.4	Scenario prerequisites . . . . .	436
9.5	Task summary . . . . .	436
9.6	Configure Notes mail clients to use LDAP . . . . .	437
9.6.1	Configure LDAP address Look-up on the server for all users . . . . .	437
9.6.2	Configure LDAP address look-up on a Notes client . . . . .	437
9.6.3	Searching for e-mail addresses in an LDAP directory . . . . .	441
9.7	Configure Netscape Messenger to use LDAP . . . . .	443
9.8	Configure Outlook mail clients to use LDAP . . . . .	445
<b>Part 4.</b>	<b>Developing directory-enabled applications . . . . .</b>	<b>455</b>
<b>Chapter 10.</b>	<b>Introduction . . . . .</b>	<b>457</b>
10.1	Application programming interfaces . . . . .	458
10.2	Java applications . . . . .	458
10.3	IBM SecureWay Directory Client SDK . . . . .	459
<b>Chapter 11.</b>	<b>Using APIs to directory-enable your applications . . . . .</b>	<b>461</b>
11.1	Overview . . . . .	462
11.2	Programming techniques for using APIs and C functions in ILE RPG . . . . .	463
11.2.1	Prototypes . . . . .	464
11.3	Where to find API and C function documentation . . . . .	465
11.4	API flow when searching a directory . . . . .	466
11.5	API flow when updating a directory entry . . . . .	471
<b>Chapter 12.</b>	<b>Using the JNDI to search and update the directory . . . . .</b>	<b>477</b>
12.1	The JNDI . . . . .	478
12.2	Scenario characteristics and objectives . . . . .	480
12.3	Sample application environment . . . . .	481
12.4	Application overview . . . . .	482

12.4.1 JNDIServlet servlet . . . . .	483
12.4.2 ChangeDirEntry servlet . . . . .	485
12.4.3 AuthenticatedUser class . . . . .	487
12.4.4 LdapAttributes class . . . . .	487
12.4.5 Base64 class . . . . .	489
12.4.6 Obtaining the sample application code . . . . .	489
12.5 Searching the directory . . . . .	490
12.5.1 Creating the directory context . . . . .	490
12.5.2 Performing the search . . . . .	492
12.5.3 Processing the search results . . . . .	494
12.6 Changing a directory entry . . . . .	495
12.6.1 Creating the directory context . . . . .	496
12.6.2 Getting the attributes . . . . .	498
12.6.3 Performing the modification . . . . .	499
<b>Part 5. Appendixes . . . . .</b>	<b>503</b>
<b>Appendix A. Problem determination . . . . .</b>	<b>505</b>
OS/400 Directory Services hints and tips . . . . .	506
Traces . . . . .	506
Symptoms . . . . .	506
Directory Management Tool (DMT) hints and tips . . . . .	508
Domino problem determination hints and tips . . . . .	510
Debugging Domino LDAP . . . . .	510
Processing referrals . . . . .	517
Adding users to an access control list (ACL) . . . . .	517
Domino WebSphere SSO hints and tips . . . . .	517
IBM HTTP Server for iSeries LDAP hints and tips . . . . .	518
Traces . . . . .	518
Configuration support . . . . .	519
<b>Appendix B. Extending your directory schema . . . . .</b>	<b>521</b>
Considerations when extending the schema . . . . .	522
Finding the schema definition files on the iSeries server . . . . .	522
The redbook scenario - extending the schema . . . . .	523
Scenario objectives . . . . .	523
Scenario prerequisites . . . . .	523
Modifying your directory schema . . . . .	524
<b>Appendix C. OS/400 and Domino LDAP history . . . . .</b>	<b>527</b>
OS/400 Directory Services by release level . . . . .	528
The future . . . . .	529
Lotus Domino Directory by release level . . . . .	530
Notes server pre-Release 4.6 . . . . .	530

Domino Release 4.6 . . . . .	530
Domino Release 5. . . . .	531
Domino Release 5.0.1. . . . .	532
Domino Release 5.0.2. . . . .	532
Domino Release 5.0.3. . . . .	532
Domino Release 5.0.4. . . . .	533
Domino Release 5.0.5. . . . .	533
Domino Release 5.0.9. . . . .	533
Domino Release 6.0 . . . . .	533
Domino futures . . . . .	535
<b>Appendix D. Directory futures and trends . . . . .</b>	<b>537</b>
Dominance of LDAP . . . . .	538
Common data definition: DSML . . . . .	539
Directory-enabled applications . . . . .	540
Directory-enabled networks . . . . .	541
Metadirectories . . . . .	541
Directories and databases . . . . .	543
Directories and database tools . . . . .	543
Directories and operating system integration . . . . .	543
More directory content types . . . . .	544
Remote printing . . . . .	544
Directories in Knowledge Management . . . . .	545
Directory and taxonomies . . . . .	545
Pervasive access . . . . .	546
Separating applications from infrastructure . . . . .	546
Extranet/Internet directories . . . . .	546
Shared organizational directories . . . . .	547
Reuniting friends . . . . .	547
Directories and genealogy . . . . .	547
One world-wide directory network . . . . .	548
<b>Appendix E. The BlueNotes Product Suite . . . . .</b>	<b>549</b>
Directory products . . . . .	551
Role in directory projects . . . . .	551
MailStore . . . . .	551
Role in directory projects . . . . .	552
Document Warehouse . . . . .	552
Role in Directory Projects . . . . .	554
Direct Messaging . . . . .	555
Role in directory projects . . . . .	556
More information . . . . .	556
<b>Appendix F. Additional material . . . . .</b>	<b>557</b>

Locating the Web material .....	557
Using the Web material .....	558
System requirements for downloading the Web material .....	558
How to use the Web material .....	558
<b>Abbreviations and acronyms</b> .....	559
<b>Related publications</b> .....	561
IBM Redbooks .....	561
Other resources .....	561
Referenced Web sites .....	562
How to get IBM Redbooks .....	563
IBM Redbooks collections .....	563
<b>Index</b> .....	565



# 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:

AFP™	Environment®	OS/400®	SmoothStart™
AIX®	iSeries™	PAL®	SP™
Application	Language	PartnerWorld®	System/390®
System/400®	Environment®	Perform™	Tivoli®
AS/400®	OfficeVision/400™	pSeries™	WebSphere®
BookMaster®	OfficeVision®	Redbooks(logo)™ 	World Registry™
DB2®	Open Class®	Advanced Function	xSeries™
IBM.COM™	Operating	Printing™	
IBM®	System/400®	Redbooks™	
Integrated Language	OS/2®	SecureWay®	

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

cc:Mail®	Server™	Lotusphere®	Word Pro®
Domino™	Lotus Notes®	Notes®	
Lotus Discovery	Lotus®	Word Pro	

ActionMedia, LANDesk, MMX, Pentium and ProShare 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.

C-bus is a trademark of Corollary, 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.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

# Preface

For several years, many software vendors and customers have realized that a central directory can help consolidate and share information as well as minimize configuration and management efforts. There is virtually no limit as to what a directory can be used for. IBM as well as other companies already directory-enable many of their applications. Typically a directory is used for authentication, sharing configuration information, peoples directories, and so forth. The commonly used protocol to access these directories is the Lightweight Directory Access Protocol (LDAP).

This redbook will help system administrators and programmers to understand the concepts of directories. It also explains the major steps in planning and deploying a directory.

You will learn how to install and configure OS/400 Directory Services with all its features provided with OS/400 Version 5 Release 1. The implementation topics also teach you how to improve availability and scalability by exploiting directory replication and referral services. In a world where security is a key factor in establishing a reliable IT infrastructure, you also gain the skills to enable your LDAP directory server and client applications to securely communicate via the Secure Socket Layer (SSL) protocol.

The redbook continues with detailed information about how to successfully manage your directory. This includes various tools, such as the IBM SecureWay Directory Management Tool and LDAP utilities. Detailed information is provided on how to secure your directory entries and their attributes using the Version 5 Release 1 attribute-level permission enhancements.

Based on a practical scenario that spans across the entire redbook, you will learn how to leverage OS/400 Directory Services to authenticate users and share configuration information with the IBM HTTP Server for iSeries. You discover how Lotus Domino can be enabled for LDAP access and how Lotus Domino and WebSphere Application Server 4.0 exploit the OS/400 LDAP directory for Single Sign-On. Using the directory as an enterprise directory, you gain the knowledge to configure various e-mail client applications to look up e-mail addresses from a single directory.

Once you have discovered the power of a directory, you also want to utilize directory services for your own applications. The redbook describes how to directory-enable your applications using OS/400 application programming interfaces (APIs) and the Java Naming and Directory Interface (JNDI).

## 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, Raleigh Center.



**Thomas Barlen** is an IBM Certified Senior IT Specialist for iSeries and AS/400 systems at the International Technical Support Organization, Raleigh Center. He writes extensively and teaches IBM classes worldwide on all areas of iSeries communications, e-business, and network security. Before joining the ITSO in 1999, he worked in AS/400 software support and as a Systems Engineer in IBM Germany. He has over 13 years of experience in AS/400 networking and system management, as well as LAN and WAN network design and implementation.



**Wolfgang Eckert** is a Software Support Specialist for iSeries communications at IBM Germany. He has been working with the AS/400 platform since its announcement in 1988. He was working as a Senior Hardware Specialist until 1999 and then joined the iSeries Software Support for Communications.



**John Taylor** is the Technical Director of the Typex Group, an Advanced Business Partner of IBM, Lotus, and the AT&T Global Network. Typex specializes in helping businesses to integrate IBM servers with Lotus Notes and in connecting to the Internet. John was an early redbook author on this topic. He is the architect of IBM's BlueNotes family of software products, which provide a Lotus Notes interface to IBM servers. He writes magazine articles and travels the world speaking at customer conferences. John is a board member of the COMMON UK user group. He is a Chartered Civil Engineer by profession and lives in Newcastle upon Tyne, England. He can be reached at [john\\_taylor@typex.com](mailto:john_taylor@typex.com). John has participated remotely in this project and worked primarily on directory planning and BlueNotes topics.



**Klaus Tebbe** is an AS/400 System Support Specialist at IBM Germany. He has 12 years of experience in system support and operational support for AS/400 and System/390. He has worked with Lotus providing Domino for AS/400 software support in Europe since it became available on the AS/400 system.



**Wendy Thomson** has been a contractor with IBM Australia since 1992, working in Australia, Asia, and the U.S. Over this period she has been contracted to IBM Learning Services, ITSO, and PartnerWorld for Developers, iSeries. She has worked for the ITSO Rochester center on redbook residencies and writing educational course material and presentations. She has written Internet-based education courses for the PartnerWorld for Developers, iSeries Team; she has

also written the SmoothStart Perform Guide for IBM Global Services. Also, as an Instructor for IBM Learning Services, she teaches AS/400 and iSeries classes to customers, business partners, and IBM internal staff. On this project Wendy has tested and written about Sign Sign-On for Domino Release 5 and WebSphere Version 4.0. She also wrote about maintaining a single company-wide or cross-company LDAP directory for e-mail address look-up in a multiple mail client environment.



**Marc Willems** is an iSeries Advisory IT Specialist in Belgium. He joined IBM in 1977 and has 10 years of experience in the AS/400 system areas of e-business, networking, and networking security. He is now working for IBM Global Services where he provides technical support to AS/400 and iSeries customers in the areas of his expertise. In this redbook, Marc contributed to the OS/400 LDAP Directory Services and managing an LDAP directory on iSeries chapters.

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

**International Technical Support Organization, Raleigh Center**

Tamikia Barrow, Gail Christensen, Mark Endrei, Peter Kovari, Diane O'Shea, Margaret Ticknor, Jeanne Tucker

**International Technical Support Organization, Rochester Center**

Jenifer Servais

**International Technical Support Organization, Austin Center**

Julie Czubik

**IBM Rochester Development**

Marla Berg, Jim Fall, Pat Fleming, Brant Knudson, John McMeeking, Marion Pitts, Steve Sparrow

**IBM Canada**

Martin Barbeau

**Lotus Software, a division of the IBM Software Group**

Keith Attenborough,

**Messaging and Collaboration Development**

**Lotus and the IBM Software Group**

Terri Warren

## 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](http://ibm.com/redbooks)

- ▶ Send your comments in an Internet note to:

[redbook@us.ibm.com](mailto:redbook@us.ibm.com)

- ▶ Mail your comments to the address on page ii.



# Part 1

# Introduction

Archived





# Directory concepts

Nowadays people and businesses rely on networked computer systems to support distributed applications. These distributed applications might interact with computers on the same local area network, within a corporate intranet, within extranets linking up partners and suppliers, or anywhere on the worldwide Internet. To improve functionality and ease-of-use, and to enable cost-effective administration of distributed applications, information about the services, resources, users, and other objects accessible from the applications needs to be organized in a clear and consistent manner. Much of this information can be shared among many applications, but it must also be protected in order to prevent unauthorized modification or the disclosure of private information.

Information describing the various users, applications, files, printers, and other resources accessible from a network is often collected into a special database that is sometimes called a directory. As the number of different networks and applications has grown, the number of specialized directories of information has also grown, resulting in islands of information that are difficult to share and manage. If all of this information could be maintained and accessed in a consistent and controlled manner, it would provide a focal point for integrating a distributed environment into a consistent and seamless system.

The Lightweight Directory Access Protocol (LDAP) is an open industry standard that has evolved to meet these needs. LDAP defines a standard method for accessing and updating information in a directory. LDAP has gained wide acceptance as the directory access method of the Internet and is therefore also

becoming strategic within corporate intranets. It is being supported by a growing number of software vendors and is being incorporated into a growing number of applications. For example, the two most popular Web browsers, Netscape Navigator/Communicator and Microsoft Internet Explorer, as well as application middleware, such as the WebSphere Application Server or the IBM HTTP server, support LDAP functionality as a base feature.

This chapter introduces the fundamentals of directories and the most commonly used protocol to access directories, the LDAP protocol. You will also learn about the various components a directory is made of.

Part of the information covered in this chapter and further information on LDAP directory concepts and implementations can be found in the following publications:

- ▶ IBM Redbook *Understanding LDAP*, SG24-4986
- ▶ IBM Redbook *LDAP Implementation Cookbook*, SG24-5110
- ▶ IBM Redbook *Using LDAP for Directory Integration: A Look at IBM SecureWay Directory, Active Directory, and Domino*, SG24-6163

Another book that contains good information about directory concepts and architecture is *e-Directories Enterprise Software, Solutions, and Services*, ISBN 0-201-70039-5.

## 1.1 Directories

A directory is a listing of information about objects arranged in some order that gives details about each object. Common examples are a city telephone directory and a library card catalog. For a telephone directory, the objects listed are people; the names are arranged alphabetically, and the details given about each person are address and telephone number. Books in a library card catalog are ordered by author or by title, and information such as the ISBN number of the book and other publication information is given.

In computer terms, a directory is a specialized database, also called a data repository, that stores typed and ordered information about objects. A particular directory might list information about printers (the objects) consisting of typed information such as location (a formatted character string), speed in pages per minute (numeric), print streams supported (for example PostScript or ASCII), and so on.

Directories allow users or applications to find resources that have the characteristics needed for a particular task. For example, a directory of users can be used to look up a person's e-mail address or fax number. A directory could be searched to find a nearby PostScript color printer. Or a directory of application servers could be searched to find a server that can access customer billing information.

The terms *white pages* and *yellow pages* are sometimes used to describe how a directory is used. If the name of an object (person, printer) is known, its characteristics (phone number, pages per minute) can be retrieved. This is similar to looking up a name in the white pages of a telephone directory. If the name of a particular individual object is not known, the directory can be searched for a list of objects that meet a certain requirement. This is like looking up a listing of hairdressers in the yellow pages of a telephone directory. However, directories stored on a computer are much more flexible than the yellow pages of a telephone directory because they can usually be searched by specific criteria, not just by a predefined set of categories.

A directory is often described as a database, but it is a specialized database that has characteristics that set it apart from general-purpose relational databases. One special characteristic of directories is that they are accessed (read or searched) much more often than they are updated (written). Hundreds of people might look up an individual's phone number, or thousands of print clients might look up the characteristics of a particular printer, but the phone number or printer characteristics rarely change.

Because directories must be able to support high volumes of read requests, they are typically optimized for read access. Write access might be limited to system administrators or to the owner of each piece of information. A general-purpose database, on the other hand, needs to support applications, such as airline reservations and banking applications, with relatively high-update volumes.

Because directories are meant to store relatively static information and are optimized for that purpose, they are not appropriate for storing information that changes rapidly. For example, the number of jobs currently in a print queue probably should not be stored in the directory entry for a printer because that information would have to be updated frequently to be accurate. Instead, the directory entry for the printer can contain the network address of a print server. The print server can be queried to get the current queue length if desired. The information in the directory (the print server address) is static, whereas the number of jobs in the print queue is dynamic.

Another difference between directories and general-purpose databases is that most directory implementations still do not support transactions. However, transactions are supported in LDAP and are limited to transactions within the LDAP directory and do not include other transactions (for example, database operations). Transactions are all-or-nothing operations that must be completed in total or not at all. For example, when transferring money from one bank account to another, the money must be debited from one account and credited to the other account in a single transaction. If only half of this transaction completes or someone accesses the accounts while the money is in transit, the accounts will not balance. General-purpose databases usually support such transactions, which complicates their implementation.

Because general-purpose databases must support arbitrary applications such as banking and inventory control, they allow arbitrary collections of data to be stored. Directories may be limited in the type of data they allow to be stored (although the architecture does not impose such a limitation). For example, a directory specialized for customer contact information might be limited to storing only personal information such as names, addresses, and phone numbers. If a directory is extensible, it can be configured to store a variety of types of information making it more useful to a variety of programs.

Another important difference between a directory and a general-purpose database is in the way information can be accessed. Most databases support a standardized, very powerful access method called Structured Query Language (SQL). SQL allows complex update and query functions at the cost of program size and application complexity. Directories, such as an LDAP directory, on the other hand, use a simplified and optimized access protocol that can be used in slim and relatively simple applications.

Because directories are not intended to provide as many functions as general-purpose databases, they can be optimized to economically provide more applications with rapid access to directory data in large distributed environments. If your intended use of the directory is to be read, mostly in a non-transactional environment, both the directory client and directory server can be simplified and optimized.

A request is typically performed by the directory client, and the process that looks up information in the directory is called the directory server. In general, servers provide a specific service to clients. Sometimes a server might become the client of other servers in order to gather the information necessary to process a request.

A directory service is only one type of service that might be available in a client/server environment. Other common examples of services are file services, mail services, print services, Web page services, and so on. The client and server processes might or might not be on the same machine. A server is capable of serving many clients. Some servers can process client requests in parallel. Other servers queue incoming client requests for serial processing if they are currently busy processing another client's request.

An API defines the programming interface a particular programming language uses to access a service. The format and contents of the messages exchanged between client and server must adhere to an agreed-upon protocol.

The Lightweight Directory Access Protocol (LDAP) defines a message protocol used by directory clients and directory servers. The LDAP Protocol uses different messages. For example, a *bindRequest* may be sent from the client to the LDAP server at the beginning of a connection. A *searchRequest* is used to search for a specific entry in the directory.

There are also associated LDAP APIs for the C language and ways to access LDAP from within a Java application. The LDAP APIs can also be used by ILE programming languages, such as ILE RPG. The client is not dependent upon a particular implementation of the server, and the server can implement the directory however it chooses.

LDAP is an open industry standard that defines a standard method for accessing and updating information in a directory. LDAP has gained wide acceptance as the directory access method of the Internet and is therefore also becoming strategic within corporate intranets. It is being supported by a growing number of software vendors and is being incorporated into a growing number of applications.

LDAP defines a communication protocol. That is, it defines the transport and format of messages used by a client to access data in an X.500-like directory. LDAP does not define the directory service itself. When people talk about the LDAP directory, that is the information that is stored and can be retrieved by the LDAP protocol.

The CCITT (Comite Consultatif International Telephonique et Telegraphique or Consultative Committee on International Telephony and Telegraphy, which is now ITU-T, International Telecommunications Union - Telecommunication Standardization Sector) defined the X.500 standard in 1988, which then became ISO 9594, Data Communications Network Directory, Recommendations X.500/X.521 in 1990, though it is still commonly referred to as X.500.

X.500 organizes directory entries in a hierarchical name space capable of supporting large amounts of information. It also defines powerful search capabilities to make information retrieval easier. Because of its functionality and scalability, X.500 is often used together with add-on modules for interoperation between incompatible directory services.

X.500 specifies that communication between the directory client and the directory server uses the Directory Access Protocol (DAP). However, as an application layer protocol, the DAP requires the entire Open Systems Interconnection (OSI) protocol stack to operate. Supporting the OSI stack requires more resources than are available in many small environments. Therefore, an interface to an X.500 directory server using a less resource-intensive or lightweight protocol (LDAP) was desired.

An application client program initiates an LDAP message by calling an LDAP API. But an X.500 directory server does not understand LDAP messages. In fact, the LDAP client and X.500 server even use different communication protocols (TCP/IP vs. OSI). The LDAP client actually communicates with a gateway process (also called a proxy or front end) that forwards requests to the X.500 directory server. This gateway is known as an LDAP server. It services requests from the LDAP client. It does this by becoming a client of the X.500 server. At the beginning, the LDAP server implementations supported both OSI and TCP/IP to be able to translate requests received by LDAP clients to DAP requests required to access X.500 directories. Newer LDAP server implementations, such as the IBM SecureWay Directory server, support only the LDAP protocol to access the directory. The LDAP server on the iSeries server is called Directory Services and implements the IBM SecureWay Directory.

On iSeries, beginning with Version 4 Release 5, both the OS/400 LDAP server and the OS/400 LDAP client are based on LDAP Version 3. You can use a Version 2 client with a Version 3 server. However, you cannot use a Version 3 client with a Version 2 server unless you bind as a Version 2 client and use only Version 2 APIs.

The Windows LDAP client is based on LDAP Version 3.

All LDAP servers share many basic characteristics since they are based on industry standard Request for Comments (RFCs). However, due to implementation differences, they are not all completely compatible with each other when there is not a standard defined. The LDAP server provided by iSeries Directory Services is closely compatible with other LDAP directory servers in the IBM SecureWay product group. However, it may not be as compatible with other LDAP servers.

The majority of data for the LDAP server that Directory Services provides resides in an OS/400 database.

## 1.2 Advantages of using a directory

An application-specific directory stores only the information needed by a particular application and is not accessible by other applications. Because a full-function directory service is complex to build, application-specific directories are typically very limited. They probably store only a specific type of information, do not have general search capabilities, do not support replication and partitioning, and probably do not have a full set of administration tools. An application-specific directory could be as simple as a set of editable text files, or it could be stored and accessed in an undocumented, proprietary manner.

In such an environment, each application creates and manages its own application-specific directory, which quickly becomes an administrative nightmare. The same e-mail address stored by the calendar application might also be stored by a mail application and by an application that notifies system operators of equipment problems. Keeping multiple copies of information up-to-date and synchronized is difficult, especially when different user interfaces and even different system administrators are involved.

What is needed is a common, application-independent directory. If application developers could be assured of the existence of a directory service, then application-specific directories would not be necessary. However, a common directory must address the problems mentioned above. It must be based on an open standard that is supported by many vendors on many platforms. It must be accessible through a standard API. It must be extensible so that it can hold the types of data needed by arbitrary applications. And it must provide full functionality without requiring excessive resources on smaller systems. Since more users and applications will access and depend on the common directory, it must also be robust, secure, and scalable.

When such a directory infrastructure is in place, application developers can devote their time to developing applications instead of application-specific directories. In the same way that developers rely on the communications infrastructure of TCP/IP and remote procedure call (RPC) to free them from low-level communication issues, they will be able to rely on powerful, full-function directory services. LDAP is the protocol to be used to access this common directory infrastructure. Like HTTP (hypertext transfer protocol) and FTP (file transfer protocol), LDAP has become an indispensable part of the Internet's protocol suite.

When applications access a standard common directory that is designed in a proper way, rather than using application-specific directories, redundant and costly administration can be eliminated, and security risks are more controllable. For example, the telephone directory, mail, and Web application as shown in Figure 1-1 on page 11 can all access the same directory to retrieve an e-mail address or other information stored in a single directory entry. The advantage is that the data is kept and maintained in one place. Various applications can use individual attributes of an entry for different purposes permitting that they have the correct authority. New uses for directory information will be realized, and a synergy will develop as more applications take advantage of the common directory.



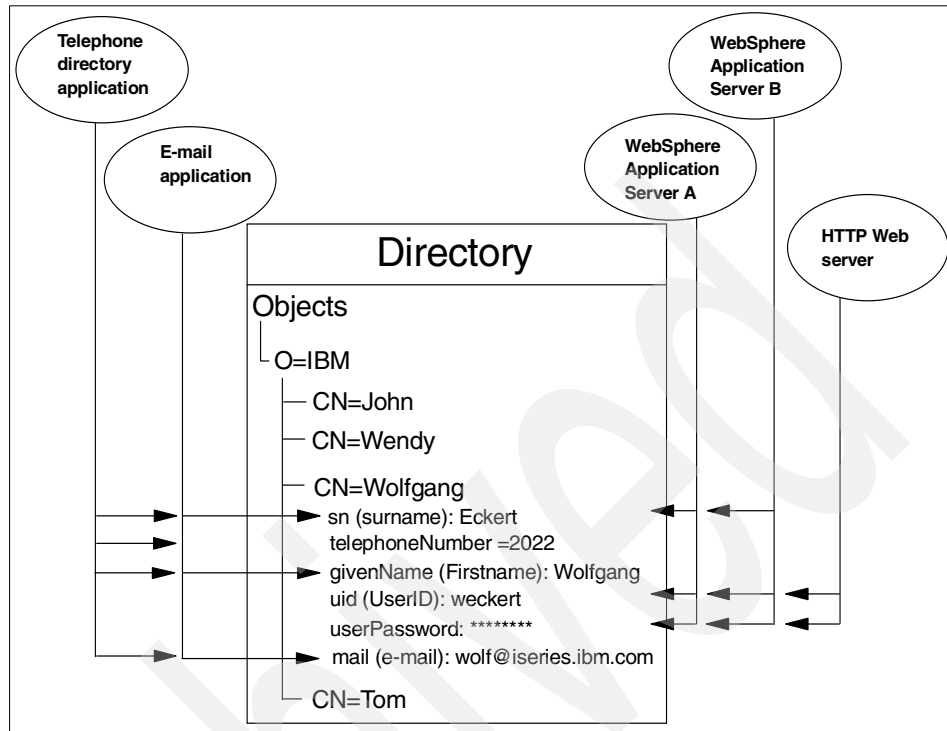


Figure 1-1 Several applications using attributes of the same entry

Storing data in a directory and sharing it amongst applications saves you time and money by keeping administration effort and system resources down. Many IBM applications also utilize directories to centrally store and share information. The number of applications that support LDAP directories is constantly increasing. For example, LDAP directory support, such as for authentication and configuration management, is provided in various IBM Operating Systems, IBM WebSphere Host On-Demand, IBM WebSphere Application Server, Tivoli Policy Director, IBM SecureWay Directory, IBM HTTP server, Lotus Domino, and so forth.

## 1.3 Directory components

A directory contains a collection of objects organized in a tree structure. The LDAP naming model defines how entries are identified and organized. Entries are organized in a tree-like structure called the Directory Information Tree (DIT). Entries are arranged within the DIT based on their distinguished name (DN). A DN is a unique name that unambiguously identifies a single entry. DNs are made

up of a sequence of relative distinguished names (RDNs). Each RDN in a DN corresponds to a branch in the DIT leading from the root of the DIT to the directory entry. A DN is composed of a sequence of RDNs separated by commas, such as `cn=thomas,ou=itso,o=ibm`.

You can organize entries, for example, after organizations and within a single organization you can further split the tree into organizational units, and so forth. You can define your DIT based on your organizational needs as shown in a simple example in Figure 1-2. If you have, for example, one company with different divisions, you may want to start with your company name under the root as the organization (o) and then branch into organizational units (ou) for the individual divisions. In case you store data for multiple organizations within a country, you may want to start with a country (c) and then branch into organizations. For more information on planning a DIT, refer to Chapter 2, “Planning your directory” on page 17.

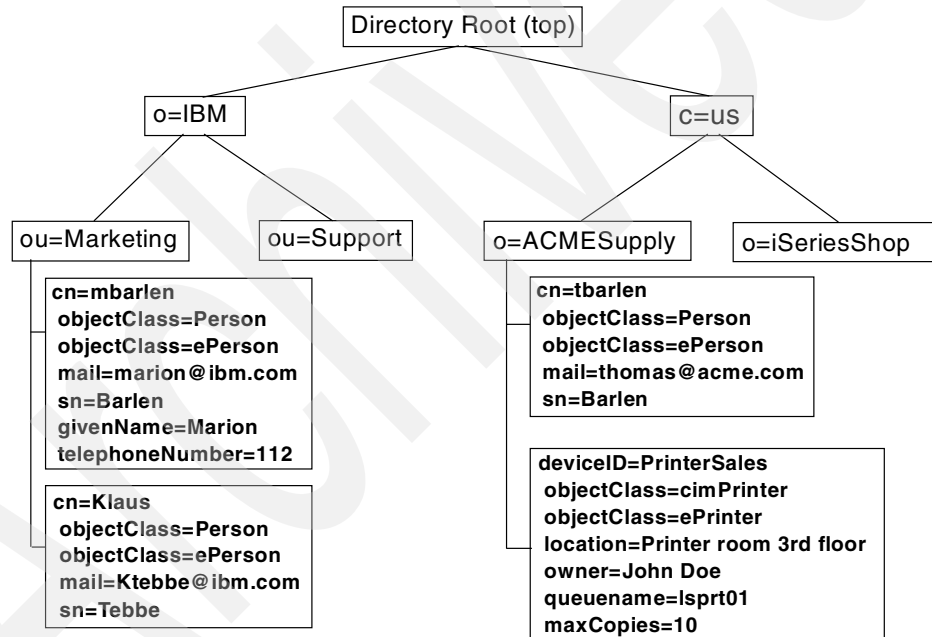


Figure 1-2 Example of a directory information tree

Each object also referred to as an entry in a directory belongs to one or more object classes. An object class describes the content and purpose of the object. It also contains a list of attributes, such as a telephone number or surname, that can be defined in an object of that class. You can publish entries of different object classes under another object as shown in Figure 1-2 where an ePrinter object and a Person object is published under the organization ACMESupply.

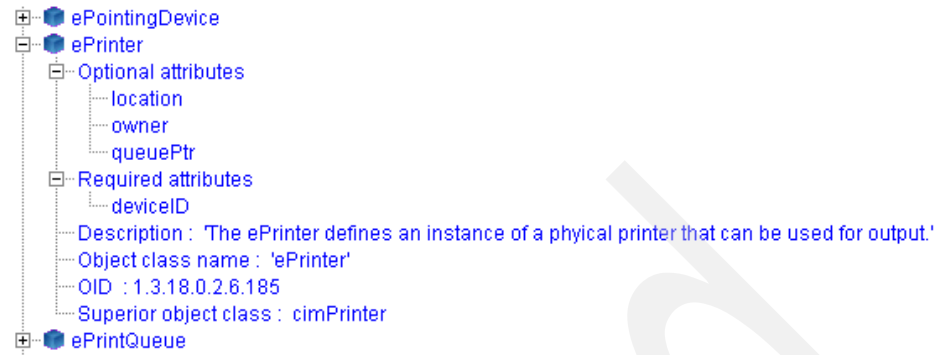


Figure 1-3 ePrinter object class

The object class also defines which of the attributes must be defined (required) when creating an object of this class and which attributes are optional. As shown in Figure 1-3, the object class with the name `ePrinter` has a required attribute `deviceID` and three optional attributes that may or may not be filled in when creating an `ePrinter` object. Object classes can also inherit characteristics, such as attributes from other object classes. In the example of the `ePrinter`, the class inherits all the attributes that are defined in class `cimPrinter`. That means, when you create an `ePrinter` object you have to define the `deviceID` and optionally you can specify the `location`, `owner`, and `queuePtr` attribute of `ePrinter` and all attributes of `cimPrinter`.

Also attributes themselves have certain characteristics as shown in Figure 1-4 on page 14. The surname attribute name, for example, is defined as `sn` and `surName`, and describes a person's family name. The attribute definition specifies also the syntax rules for the attribute value. A telephone number may only contain numbers and hyphens while the surname consists of alpha characters. Other specifications include whether this attribute can contain only one or many values, the matching rules, the Object Identifier (OID), and so forth. The IBM SecureWay Directory product as used on the iSeries server provides also some IBM proprietary extensions. Other manufactures, such as Microsoft, have similar extensions. The IBM extensions on the iSeries server include also an access class, which is used in combination with access control lists (ACLs) to control who can perform a certain action on the attribute value, such as read, write, search, or compare operations.

All the objects and attributes with their characteristics are defined in schemas. The schema specifies what can be stored in the directory. Schema-checking ensures that all required attributes for an entry are present before an entry is stored. Schema-checking also ensures that attributes not in the schema are not

stored in the entry. Optional attributes can be filled in at any time. A schema also defines the inheritance and subclassing of objects and where in the DIT structure (hierarchy) objects may appear. Information about the IBM SecureWay Directory schema can be found at:

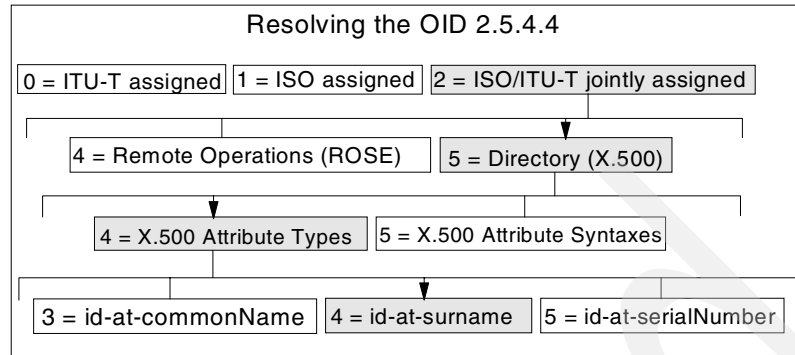
<http://www.ibm.com/servers/eserver/series/ldap/schema>

The screenshot shows a Windows-style dialog box titled "Edit attribute - sn". It has two tabs: "General" and "IBM extensions", with "General" currently selected. The dialog contains several input fields and a checkbox. The "Attribute name" field is a dropdown menu showing "sn". The "Description" field is a text box containing the text: "This is the X.500 surname attribute, which contains the family name of a person." The "OID" field is a text box containing "2.5.4.4". The "Superior attribute" field is a dropdown menu showing "name". The "Syntax" field is a dropdown menu showing "Attribute Type Description syntax". The "Syntax length" field is an empty text box. Below these fields is a checked checkbox labeled "Allow multiple values". Underneath the checkbox is a section titled "Matching rules:" followed by three rows, each with a label and a dropdown menu: "EQUALITY" with "caseIgnoreMatch", "ORDERING" with "caseIgnoreOrderingMatch", and "SUBSTR" with "caseIgnoreSubstringsMatch". At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

Figure 1-4 Attribute definition example

As you have seen in Figure 1-3 on page 13 and Figure 1-4, object classes and attributes including their specifications are defined as OIDs in an ASN.1 notation format. All these OIDs are registered with a public organization, such as the ANSI organization (<http://www.ansi.org>) for the United States. The number notation refers to a hierarchy. For example, the OID 2.5.4.4 resolves into a surName attribute as shown in Figure 1-5 on page 15. For further information on OIDs, their meaning, and the registries, visit the following Web site:

<http://www.alvestrand.no/objectid/index.html>



*Figure 1-5 Example of Object Identifiers as defined by the ANSI organization*



## Planning your directory

The first sections in this chapter describe some guidelines on how the design and implementation of the data and directory tree structure should be done. Then security planning is described, followed by implementing such a directory in a physical infrastructure having scalability, availability, manageability, and maintenance aspects of an LDAP directory deployment in mind.

Discussing low-level details of designing a directory implementation, such as detailed performance tuning aspects or product selection criteria, is beyond the scope of this book. However, this chapter gives you an introductory understanding of what has to be considered when LDAP is to be introduced in an organization.

The discussions that follow in this chapter often refer to typical White Pages directory implementations for people directories. This approach was chosen for the sake of simplicity. Please bear in mind, LDAP is not only suitable for people directories. An LDAP directory can hold almost any kind of information and can therefore be used for a much broader range of applications. The "The Directory-Enabled Networks Initiative" (DEN) is just one example where an LDAP directory is being used for storing network configuration and topology data.

Creating a design that has the flexibility to accommodate changes within the organization is probably the single most important task in implementing a directory service. This will help save time and money as the directory service grows. When designing the directory service, the project can be divided into several smaller projects:

- ▶ Surveying the directory service contents
- ▶ Creating access control strategies
- ▶ Replication and partitioning strategies
- ▶ Network planning (physical planning)

This chapter discusses the four main planning phases when designing an LDAP directory and briefly discusses implementation issues:

- ▶ The first phase, defining directory content, has two components. The first component, defining directory requirements, is about a careful analysis of the main purpose of the directory and the associated considerations to arrive at an overall approach to the directory plan. The second component, data design, is then about understanding the sources and nature of the data, deciding the scope of the data within the directory, and planning the way in which it will integrate with external data.
- ▶ The second phase, organizing your directory, also has two components. The first component, schema design, determines the format in which the data is to be stored. This is analogous to the field data definitions in a relational database. The second component, namespace design, determines the hierarchical structure of the directory. This is analogous to the relationship between individual files and their access paths in a relational database.
- ▶ The third phase, securing directory entries, is all about privacy and security design to ensure that the data in the directory is protected, as well as about allowing applications themselves to be secured by use of the directory. This aspect of the design affects all other aspects.
- ▶ The fourth phase, designing your server and network infrastructure, has two components. The first component, topology design, helps to determine the number and location of directory servers and how the data is distributed among them. The second, optional component, replication design, enables multiple copies of the data to be deployed, which can aid performance.

Surprising as it may seem, with the exception of security, the various major dimensions of design are largely independent of each other.

Some aspects of the design process allow for flexibility when requirements may change in the future. Others are less forgiving and can involve a major upheaval. It is essential to undergo a thorough planning process before starting the live implementation. Do not be misled into thinking, for instance, that because the



IBM SecureWay software is *free*, for example, included in the price of OS/400, it is a lightweight piece of infrastructure. Nothing could be further from the truth. In building an LDAP-enabled directory you are laying the framework for generations of software that are even now beginning to emerge. Think of it rather in the way that the DB2 Relational Database Management System is *free* with OS/400. The directory, like the database, is one of the major building blocks of your infrastructure and some attention to planning at the initial stages will reap rich rewards in time to come.

We have discussed here some aspects of directory design. However, it needs to be pointed out that there is no single correct way to design a directory. To be able to build a more objective picture of the naming methodology, we recommend that several sources of information are compared. Often, vendors will have their own implementation guides that reflect different angles of views for this aspect. See also the related publication “Related publications” on page 561 for information on literature in this area.

Part of the information covered in this chapter and further information on LDAP directory concepts and implementations can be found in the following publications:

- ▶ IBM Redbook *Understanding LDAP*, SG24-4986
- ▶ IBM Redbook *LDAP Implementation Cookbook*, SG24-5110

## 2.1 Defining the directory content

The first phase, defining the directory content, is concerned with what it is that your proposed directory project sets out to achieve and what data is available to help it do so.

### 2.1.1 Defining directory requirements

This section discusses the directory definition requirements that need to be considered when planning a directory implementation.

#### **Application needs**

What type of application(s) will use the directory? Determine what directory-enabled applications are to be deployed and what are their data needs.

Determine the organization's other mission-critical applications. Find out if those applications can directly access and/or update the directory. What are the requirements for manageability and scalability? Will the LDAP service be participating with an X.500 directory service?

#### **User needs**

Determine who needs access to the data as a user. Find out if those users can directly access or even update the directory. Determine the location of clients (users or applications). What expectations are there for privacy concerns? How accurate and up-to-date must the directory content be?

#### **Deployment issues**

What resources will be available for deployment? What people and skills are available? Can this be done as part of another project, for example, messaging migration, or will it require dedicated resources?

#### **Infrastructure constraints**

What hardware configurations are already in use and which, if any, are available to the project? What operating systems, middleware, and applications are in use? Specifically, what directory applications are already available? Obtain a network diagram. Is the directory to be protected behind a firewall or exposed to the Internet?

## 2.2 Data design

Planning the directory's data is the most important aspect of the directory planning activities, and it is probably the most time-consuming aspect as well. A considerable amount of the time spent planning the directory data will most likely be spent surveying the organization to locate all the data stores where directory information is managed. As this survey is performed, expect to find that some kinds of data are not well-managed; some processes may be inefficient, inadequate, or non-existent; and some kinds of data may not be available at all. All of these issues should be addressed before finishing a data-planning phase.

However, we start by looking at the requirements on the data to be used in the directory service. The scope of information required will largely be driven by the application requirement. However, some types of data are better suited for a directory service than others. Ideal candidates in a directory service have some of the following characteristics:

- ▶ A directory service is not a file system, a file server, an FTP server, a Web server, or a relational database. Therefore, large, unstructured objects of data should not be put in the directory. For that kind of data, a server more appropriate for the task should be used. However, it is appropriate to store pointers to these kinds of applications within the directory service through the use of FTP, HTTP, or other types of accesses.
- ▶ The data should typically be read much more often than it is written. This is because directory services usually are tuned for read operations; write operations are more expensive in terms of resource utilization than reads, and they may impact the directory server's performance in typical directory server implementations.
- ▶ Another "rule of thumb" is that the data should typically be accessed from more than just one system or client. For example, an employee's preference settings for a specific application may not be meaningful to put in the directory if that application is only run on the employee's single workstation. If the user wants to run this application on different systems, such as a mail client application, then the application would certainly benefit from a central directory for storing user preferences. This would allow the employee to use the same setup on multiple systems or even platforms within the organization.

Having in mind the types of data suitable and unsuitable for use in a directory, it is now possible to survey what the directory service data will be.

## 2.2.1 Sources for data

Planning the directory content includes deciding which existing data to store in the directory. Survey the organization and identify where the data comes from (such as OS/400 User Profiles, System Distribution Directory (SDD), Windows NT or Novell NetWare directories, Human Resources databases, e-mail systems, and so forth).

When deciding on what to put into the directory, all the owners of data relevant to the contents of the directory should be identified. It is very probable that the information you will be choosing to put in the LDAP directory already resides on some other system in your organization. For example, the Personnel Department most likely already has databases with personnel information. Also be sure to make adequate use of processes already in place to administer that data even in the planned directory service.

Data management and access control are both important when maintaining a directory service. Plans must be made to identify resources for keeping the data up-to-date and identifying resources with the authority to decide on access control policies regarding the data residing in the directory tree.

If data is going to be imported from other sources, develop a strategy for both bulk imports and incremental updates. Try to limit the number of applications that can change the data. Doing this will help ensure the data integrity while reducing the organization's administration.

Identify duplications and data that is not actually used or required. Harmonize the data by eliminating such duplications and discard unnecessary data.

## 2.2.2 Characteristics of data elements

Data is made up of data elements, which possess several characteristics such as format, size, frequency, ownership, relationship with other data elements, etc. For instance, the data element e-mail address has a format of text, has many characters, has possible multiple values, is owned by the IT department, is used by both users and applications and is related to the user's entry. Examine each planned data element to determine its characteristics and which are shared with other elements.

For each piece of data, determine the location where it will be mastered and who owns the data - that is, who is responsible for ensuring that the data is up-to-date.

### 2.2.3 Related data

Remember to plan for related data sources that contain directory-related data but which may not, initially at least, use the directory itself. For example, the Human Resources database must bear a close relationship to entries in a directory containing staff data. Consider appropriate replication and synchronization techniques and procedures to maintain the relationships.

## 2.3 Organizing your directory

Having decided on the type of data to use in the directory service, what the directory will be used for, and how the data will be updated, it is possible to start structuring the data. Structuring data is done by designing both a schema and a namespace. We explain these activities in the sections that follow.

### 2.3.1 Schema design

The schema design plays an important role in your directory implementation and helps you organize the data within a directory.

#### **Directory schema**

A schema is the collection of attribute-type definitions and object class definitions. A server uses these to determine how to match a filter or attribute against the attributes of a specific entry and whether to permit given attribute(s) to be added. This is similar to the data definitions of a relational database system. For more information on schemas, refer to Chapter 1, “Directory concepts” on page 3.

#### **Purpose**

The purpose of a schema is to control the nature and format of the data stored in the directory. This means that schemas can be used for data validation and to control redundant data. A schema is also used by users and applications as the basis for directory search criteria.

#### **Elements of LDAP schemas**

LDAP directory schemas consist of attributes and object classes. A more detailed discussion on schema elements can be found in Chapter 1, “Directory concepts” on page 3.

## Design overview

Schema design involves several stages. Firstly, identify any schemas provided by the applications you have in plan, plus any standard and vendor-supplied schemas. Secondly, select any predefined schemas that meet your needs. Thirdly, plan for any schema extensions.

For each piece of data, determine the name of the attribute(s) that you will use to represent the data in the directory and the object class(es) (the type of entry) that the data will be stored on.

## Predefined schemas

When deciding on the design of the schema, there are a few things to consider. The LDAP specifications include a standard schema for a typical White Pages directory (RFC 2256, A Summary of the X.500(96) User Schema for use with LDAPv3). Vendors ship schemas with their LDAP server products that may include some extensions to support special features they feel are common and useful to their client applications. Work at the Internet Engineering Task Force (IETF) is in progress to create standard schemas for a broad range of applications.

Regardless of the type of information contained in the directory server, the standard schema, some of which is based on the X.500 standard, should not be modified. If this standard schema proves to be too limiting for the intended use, it can be extended to support the unique requirements. Standard schema elements, however, should not be deleted. Doing so can lead to inter-operability problems between different directory services and LDAP clients.

It is important to use a consistent schema within the directory server because LDAP-enabled application clients locate entries in the directory by searching for object classes or attributes and their associated values. If the schemas are inconsistent, then it becomes virtually impossible to locate information in the directory tree efficiently. An example of an inconsistent schema is a situation where an attribute is used to store a specific kind of information, and then later a different attribute is used to store the exact same kind of data, for example when both attributes, `telephoneNumber` and `phone`, contain the same data.

Most LDAP-enabled application clients are designed to work with a specific, well-defined schema. Shrink-wrapped standard applications usually only work with a standard schema. These are important reasons why LDAP-based Directory Services should support at least the standard LDAP schema. Then the schema may be extended as the site discovers site-specific needs that are not met by the standard schema. For a list of iSeries and IBM SecureWay Directory predefined schemas, visit:

<http://www.ibm.com/servers/eserver/iseries/ldap/schema>

## **New schema elements**

The use of a standard schema is beneficial, and specific changes can be made so long as they are additions. You may, however, create your own, private schema. But when doing so, you must take into consideration that compatibility to any other LDAP service may be lost and that your application clients have to be aware of that private schema.

## **2.3.2 Namespace design**

Namespace design is a very important task in planning the directory. It is one of the most difficult to change at a later stage. A namespace is the means by which directory data is uniquely named and referenced. It is the equivalent of the “unique key field” for the entry. The structure of an LDAP namespace is described in Chapter 1, “Directory concepts” on page 3.

### **Purpose**

The namespace provides a way to organize the data. It can be used to partition (group) the data and to provide a basis for replication. It can affect your access control methods. Finally, it is the basic support for directory-enabled applications.

### **Analyzing needs**

Before designing your namespace you need to understand the requirements upon it. Do you need a flat namespace or a hierarchical one? What attributes can be used to name entries? Do you anticipate replication or partitioning? Does a corporate taxonomy (hierarchical map of the organization) exist, and could or should it be used? Might your requirements change over time, for example, with company mergers and acquisitions?

### **Namespace design approach**

Namespace design is done by choosing a directory suffix, branching the directory tree, and finally creating a naming style for the directory entries.

### **Choosing a suffix**

When deciding on suffixes, where a suffix is the root DN of a directory tree, it is a good idea to use the same naming structure for LDAP as is used for X.500. Using the X.500 methodology would lead to choosing a suffix like `o=ibm,c=us` or `ou=raleigh,o=ibm`.

This method will set the root of the directory tree to a specific organization in a specific country or to a specific organization and organizational unit. However, it is not necessary to do this, unless there are plans to participate in an X.500 directory service, since LDAP does not require any specific format for the DN naming convention. In LDAP, the directory suffix can be chosen freely to reflect

the organizations distinct name. Another method that you can use, if the X.500 method does not seem appropriate, is to use the DNS naming model when choosing the directory suffix. This would result in a suffix using the domainComponent attribute, for example: dc=server,dc=company,dc=com.

The design of the directory schema and definition of the suffix makes it possible to start populating the tree. But, before doing so, the naming structure must be put in place. We have divided the discussion on naming structure creation into the two sections that follow:

- ▶ Branching of the directory tree
- ▶ Naming style for the entries

### **Branching the directory tree**

Choosing to branch a directory tree based on the organizational structure, such as departments, can lead to a large administrative overhead if the organization is very dynamic and changes often. On the other hand, branching the tree based on geography may restrict the ability to reflect information about the organizational structure. A branching methodology that is flexible, and which still reflects enough information about the organization, must be created.

Because the structure of organizations often changes considerably over time, the aim should be to branch the tree in such a way as to minimize the number of necessary changes to the directory tree once the organization has changed. Note that renaming a department entry, for example, has the effect of requiring a change of the DNs of all entries below its branch point. This has an undesirable impact on the service for several reasons. Alias entries and certain attributes or ordinary entries, such as `seeAlso` and `secretary`, use DNs to maintain links with other entries. These references are one-way only, and LDAP currently offers no support to automatically update all references to an entry once its DN changes. The impact of renaming branches is illustrated in the following example.

When adding employees to their respective departments, it would be possible to create distinguished names (DN) like `cn=John Smith, ou=Marketing, l=se, dc=xyz.com`. If John Smith should at a later time move to another department, his DN will have to change. This results in changing all entries regarding access rights and more. If John Smith's DN had been set to `cn=John Smith, ou=employees, l=se, dc=xyz.com`, then this would not be a problem. An attribute describing which department he belongs to (`ou=marketing`) could be added to his entry to include this information.

This problem does not arise with OS/400's System Distribution Directory (SDD), which allows both the Department and Location fields to be maintained independently of the person entries with pointers providing the links. A department or location can be updated, renamed, merged, or moved within the



hierarchy and all person entries will immediately display the new value. If you are planning to update the SDD, then the use by the directory administrator of these two fields might be considered, as both provide prompted input in the 5250 interface. With OS/400 SecureWay the standard LDAP-based approach is followed and so, although the Department field can be replicated via an API, the person links are not maintained.

In Domino the standard schema provides Person document fields for company, location, and department, but again they are not externally linked. However, see Appendix E, "The BlueNotes Product Suite" on page 549, for details of how this hierarchy can be built from directory entries as a corporate taxonomy, be re-applied to names.nsf to give users the hierarchical view, be maintained externally by the administrator, and also be used in other Notes applications.

Other criteria that may or should be considered when branching the directory tree include physical or cultural splits in the organization and the nature of the client (human or application).

If your organization has separate units that are either physically separated or have their own management authorities, you might have a *natural* requirement to split and separate parts of the DIT.

A general rule of thumb says that the DIT should be reasonably shallow unless there are strong reasons to design deep branching levels down the directory tree. If the directory information is primarily searched and read by human users - that is, if users manually type in search criteria - the DIT should provide the information in an intuitive manner so that finding information is not limited to system specialists. If, on the other hand, the information is primarily retrieved from programs, other rules more suitable for that application can be followed.

### 2.3.3 Naming style

The first goal of naming is to provide unique identifiers for entries. Once this is achieved, the next major goal should be to make querying of the directory tree intuitive. Support for a naming structure that enables the use of user-friendly naming is desirable. Other considerations, such as accurately reflecting the organizational structure of an organization, should be disregarded if it has a negative effect of creating complex DNs, thus making normal querying non-intuitive. If we take a look at the X.500 view on naming, we see that the X.501 standard specifies that "RDNs are intended to be long-lived so that the users of the Directory can store the distinguished names of objects...", and "it is preferable that distinguished names of objects which humans have to deal with be user-friendly" (excerpt from The Directory – Overview of Concepts, Models and Services, CCITT 1988, cited in RFC 1617).

Multicomponent relative distinguished names can be created by using more than one component selected from the set of the attributes of the entry to be named. This is useful when there are, for example, two persons named John Smith in one department. The use of multicomponent relative distinguished names allows one to avoid artificial naming values such as `cn=John Smith 1` or `cn=John Smith 2`. Attributes that could be used as the additional naming attribute include title, room number, telephone number, and user ID, resulting in a RDN, like `title=Dr, cn=John Smith`, creating a more user-friendly naming model.

A consistent approach to naming people is especially important when the directory stores information about people. Client applications will also be better able to assist users if entries have names conforming to a common format, or at least to a very limited set of formats. It is practical if the RDN follows such a format.

In general, the standard attribute types should be used as documented in the standards whenever possible. It is important to decide, within the organization, which attributes to use for what purpose and not to deviate from that structure.

It is also important that the choice of a naming strategy not be made on the basis of the possibilities of the currently available client applications. For example, it is questionable to use `commonName` of the form "surname firstname" merely because a client application presents results in a more satisfactory order by doing so. Use the best structure for people's names, and adapt or design the client applications accordingly.

## 2.4 Securing directory entries

Having designed the directory tree, we now need to decide on a security policy.

The degree of security controls you require will depend on the nature of the information you are storing. If it is just e-mail addresses then the worst danger of unlimited read capability is *spam* e-mail, and the worst danger of uncontrolled editing is misdirected e-mail. However, if the directory contains gender, home addresses, and Social Security numbers then the dangers are more extensive.

The degree of security you require will also reflect the ways in which clients will be accessing the directory and the methods that will be used to update and manage the directory.

Finally, it needs to reflect an acceptable level of administration effort for security. A security policy should be strong enough to prevent sensitive information from being modified or retrieved by unauthorized users, while simple enough that administration is kept simple so authorized parties can easily access it. Ease of

administration is very important when it comes to designing a security policy. Too complex a security policy can lead to mistakes that either prevent people from accessing information that they should have access to, or allow people to modify or retrieve directory information that they should not have access to.

## **Purpose**

The most basic purpose of security is to protect the data in your directory. It needs to be protected against unauthorized access, tampering with information, and denial of service.

## **Analysis of security requirements**

Try to find answers to the following sorts of questions. Will your directory be read-only? How sensitive is the data? Is replication to multiple locations planned? What privileges might administrators have? How reliable are the users? How will they react to different levels of security? Will they require access across the Internet? Is your network itself secure? How about the machine room?

## **Design overview**

To plan for the required level of security, two basic areas must be considered to answer the following questions: What level of security is needed when clients identify themselves to the directory server, and what methodology will be used when authorizing access to the different kinds of information in the directory? These areas are authentication and authorization.

## **Authentication design**

Conceptually, directory authentication can be thought of as logging into the directory. LDAP terminology, however, usually refers to this operation as *binding* to the directory.

Generally, bind operations consist of providing the equivalent of a user ID and a password. However, in the case of an LDAP directory, the user ID is actually a distinguished name (or a distinguished name derived from a user ID). The distinguished name used to access the directory is referred to as the bind DN.

So, what level of authentication should be considered? There are, generally speaking, three different approaches:

### **No authentication**

This is the simplest approach, which might be perfectly suitable for most directories when all users are equally granted read (or even write) access to all data. There is no need for user authentication when this is the case.

- Basic authentication** This lets the client bind by entering a DN and a password. Using basic authentication will not ensure integrity and confidentiality of the login data since it is being sent over the network in a readable form.
- Secure authentication** Simple Authentication and Security Layer (SASL) is an extensible authentication framework. It was added to LDAP Version 3, and it supports Kerberos and other security methods, like S/Key. SASL provides the ability to securely authenticate LDAP clients and LDAP directory servers. There is an external mechanism in SASL that allows the use of authentication identity information from security layers external to the SASL layer. One possibility is to use the authentication information from SSL. SSL is generally used to secure the connection between a client and a server through the exchange of certificates. The client certificate can be used through SASL as authentication identity. SASL is already used within several Internet protocols including IMAP4 and POP3 (mail server protocols).

It is possible that there is a need for both basic and secure authentication. The choice will be dependent on the security policies in the organization's networks and what type of access rights the different types of clients will have when communicating with the server. For example, when setting up server-to-server communication, it may be valuable to use strong, secure authentication since server-to-server communication will often rely on unrestricted access to each other's tree structures, including individual entry's access settings. On the other hand, for client-to-server communication, where clients only have read access to names, phone numbers, and mail addresses, there is most likely no need for anything but basic authentication.

When using secure authentication, it is possible to choose from different methods depending on the vendors' implementations, for example Kerberos or SSL. If Kerberos is not already deployed in the organization's intranet, then it will probably be sensible to use SSL, since support for SSL is included in most popular LDAP clients. When using SSL, it is possible for the server to authenticate the client by using its server certificate. A server certificate can be thought of as a secure, digital signature that uniquely identifies a server. It has been generated and registered with a trusted certifying authority, also known as a Certificate Authority (CA), such as VeriSign or the IBM World Registry CA. Also, when using server certificates, an encrypted communication can be established between the client and server, enabling a secure basic authentication of the client to the server.

Using SSL server certificates will be particularly interesting when setting up LDAP services on insecure networks, such as the Internet/extranet. This will enable the clients to verify the identity of the server and to encrypt communication of the basic authentication from the clients to the server on the insecure networks.

When using basic authentication, administration of passwords on the directory server will be necessary and may impose some administration overhead. If SSL client certificates are used, then an appropriate infrastructure will be needed to support the certificate generation and administration. This is usually done by separate certificate servers. Client certificate deployment is beyond the scope of this book, but it ought to be mentioned that LDAP supports storing client public keys and certificates in the entries allowing you also to use the directory by mail clients to encrypt e-mail.

## Authorization design

The data in the directory tree will have to be protected in different ways. Certain information must be searchable for everybody, some must be readable, and most of it will be write protected. In LDAP Version 3, there are no defined attributes to handle this. As a result, vendors support their own implementations of authorization. This is done by different implementations of access control lists (ACLs).

ACLs are used to define access rules to the different entries in the directory tree. As an example of an ACL implementation, Example 2-1 shows the IBM SecureWay LDAP directory server's implementation of ACL attribute entries. The pertinent control attributes used here are `aclsource`, `aclpropagate`, and `aclentry`, where the latter, for example, is the attribute that specifies who has access to the entry and what level of access he or she has. In Example 2-1, `cn=John Arnold,ou=employees,o=iserieshop` has read, write, search, and compare (rwsc) rights for normal, sensitive and critical data (the entry is highlighted and split into three lines in the example below).

### *Example 2-1 Sample ACL attribute entry*

---

```
dn: ou=employees, o=iserieshop
objectclass: top
objectclass: organizationalUnit
ou: employees
description: Employees of iSeries Shop
entryowner: access-id:cn=admin,o=iserieshop
inheritoncreate: TRUE
ownerpropagate: TRUE
aclpropagate: TRUE
ownersource: default
aclsource: OU=employees,o=iserieshop
```

```
aclentry: access-id:CN=John
Arnold,OU=employees,0=iseriesshop:object:a:normal:rwc:
sensitive:rwc:critical:rwc
aclentry: group:CN=ANYBODY:normal:rsc
```

---

When setting up access control lists, it is important to do it with the goal of minimizing the administration later on. It is good to try and delegate the access control hierarchically. An example of this could be the following: An individual, say John Arnold, needs to protect sensitive information. Two groups have been created for this purpose, owned by John Arnold (Table 2-1). Entries can be added and deleted by John Arnold to his own groups without intervention of the directory service administrators.

*Table 2-1 ACL structure for Web content administration using two groups*

Group Name	Owner	Group members
cn=editor	cn=John Arnold	cn=user1 cn=user2
cn=readers	cn=John Arnold	cn=user3 ou=techsupport

According to the table, John Arnold has added user1 and user2 to the editor group and user3 and the group called techsupport to the readers group, thus enabling user1 and user2 to edit the contents, and enabling user3 and the people in the techsupport organizational unit to read the contents.

### **Non-directory security considerations**

Other security considerations that are not directly related to directory design but that can help to protect your data include encryption.

You should also ensure that your organization's security audit procedures are updated to reflect the new directory plan.

## **2.5 Designing your server and network infrastructure**

Physical design involves building a network and server infrastructure to support availability, scalability, and manageability. Methods to do this in LDAP are partitioning and replication. Replication is actually not standardized in LDAP Version 3, but most vendors do have an implementation as described in Section 4.7, "Setting up directory replication" on page 106. In this section we concentrate on deployment issues regarding when partitioning and/or replication is appropriate when trying to reach the goals of availability, scalability, and manageability, and what the trade-offs are.

In sizing the directory service, consideration must be given to which clients will be accessing what data, from where, and how often. If there are client applications that use the directory extensively, consideration must be given to ensuring that the network availability and bandwidth are sufficient between the application servers and the directory servers. If there are network bottlenecks, they must be identified because there may be need to replicate data into remote LANs.

### **2.5.1 Availability, scalability, and manageability requirements**

Availability of a directory service may not be an issue in cases where the directory is not business-critical. However, if the use of the service becomes mission-critical, then there is a need to design a highly available system. Designing a highly available system involves more than is supported in LDAP. The components from LDAP that are needed are partitioning and replication. Since high availability involves eliminating single points of failure or reducing their impact, it is necessary to have redundant hardware, software, and networks to spread the risk.

As more and more applications use and rely on a directory service, the need to scale the directory for high-load tolerance increases. Scaling up directory servers is done much the same way, either by increasing availability or by upgrading hardware performance. As is the case when increasing availability, we have to rely on functions outside the LDAP standard as well as LDAP replication and partitioning. The round-robin DNS or the load-balancing router, such as the WebSphere Edge Server, are good tools to scale an LDAP server site.

Manageability aspects involve almost all parts of a directory design. Here is where trade-offs may have to be made regarding scalability, availability, flexibility, and manageability. The level of scalability and availability are both related to cost in hardware and software and, as a drag-along, cost of overall systems management. One important question to ask in a directory design about manageability is whether and how all information providers are able to furnish reliable, correct, and consistent directory data to the LDAP service. If this cannot be assured, there will be a chance for errors and inconsistencies in the LDAP directory data. If such problems are considered critical for the clients using the LDAP service, tools must be provided that can detect and maybe even correct these errors.

### **2.5.2 Topology design**

Topology design concerns the distribution of directory servers. The first choice is between a centralized or a distributed approach. The second choice is between a partitioned and a replicated approach.

## Centralized or distributed?

You can choose to centralize in a single master directory or to distribute the data to additional directory servers.

A simple approach to create a highly-available directory service is to create a master and a replica directory server, each one on its own physical machine. By replicating the data, we have eliminated the single point-of-failure for both hardware and software failures. This solution with a master and one or more replica servers normally provides for high availability for read functions to the LDAP servers. Write requests can only be directed to the master server. If high availability is required for write access, additional effort is necessary. Neither read-only nor read/write replication is supported natively by the LDAP standards, but vendors may have implemented their own mechanisms. Replication solutions can also be constructed using the export/import facilities of LDAP servers or with additional, custom-designed software tools. Also the OS/400 Directory server has its own replication mechanism that is constantly being enhanced.

A mechanism must be added to handle client redirection if one server fails. This can be done manually or semi-automatically by a DNS switchover, or automatically with a load-balancing technique by using a router designed for this. Such a router forwards client requests to one of the servers based on configurable criteria. It is important that the router supports stateful protocols - that is, subsequent requests from the same client need to be forwarded to the same server. There are several products on the market from different vendors to do this, such as IBM's WebSphere Edge Server. This function is also built into Lotus Domino. The IBM eServer iSeries of course allows multiple Domino server instances to run within a single operating system instance.

There is also the issue of network bandwidth and its reliability to take into consideration. In some cases, it may be necessary to distribute a replica into another LAN with slow network connections to the master. This can also be done with any means of replicating an LDAP server (remember that replication is not included in the LDAP standards, thus you have to use vendor product support or your own methods). The primary server for a particular client may be the directory server on the client's own LAN, and the secondary will then be the central master server, accessed over the WAN. The OS/400 replication service can be configured to replicate each time an update has occurred or on a timely basis.

If the method of spreading the risk is used to create high availability, it is possible to partition the directory tree and to distribute it to different locations, LANs, or departments. As a side-effect, depending on how the directory tree is branched and distributed to these servers, each location, department, or LAN administrator could then easily manage their own part of the directory tree on a local machine,



if this is a requirement. If a single server failed in such a configuration, then only a portion of the whole directory would be affected. Partitioning on the iSeries LDAP server is achieved by setting up referrals as described in Section 4.8, “Setting up directory referrals” on page 122.

A combination of these methods could be used to create a dynamic, distributed, highly-available directory service.

### **Partitioned or replicated?**

The second choice for topology design is only applicable when a distributed approach has been selected for the first choice. The options are between a partitioned and a replicated approach. The decision criteria are usually based on performance and availability issues and will be influenced by the size of the directory.

To create a high-availability environment, it is necessary to replicate and/or partition the directory, as discussed in the previous sections. Although not directly related to LDAP, it should be mentioned that adequate systems management tools and skills must be available to run such a fairly complex environment. In addition, one of the manageability concerns regarding replication might be the need to ensure an ample level of consistency. A master LDAP server might have been updated with new information, while a replica server still runs with the old, outdated information. The required level of consistency is largely dependent on the needs of the client applications using the service. If there is a requirement for currency and consistency among replicated servers, additional means must be provided to ensure this.

Replication will also affect back up and disaster/recovery procedures. Processes will be needed to handle recovery of master servers and how synchronization of replicas will be handled. Since replication is outside the current standard for LDAP, it is necessary to study the vendor's implementation in order to find adequate solutions.

Partitioning the directory enables local servers to own their own data, depending on schema and branching design. This increases flexibility when maintaining data, but increases the complexity of referral handling. A clear method of linking the name space together will have to be formulated to ensure consistent referrals in the directory service name space such that the logical name space is still a whole. Also, each local server may have to be administered and maintained locally, requiring staff with operating system and LDAP knowledge.

You should consider partitioning if the directory is very large, if your applications only require local workgroup data, if replication volumes would otherwise be too big, if your WAN is not suited to high volumes, and where future expansion of the service might trigger one of these considerations in the future.

The optimal topology design depends on the applications, the server, the physical network, and the directory namespace.

Remember that each partition needs a partition root, which is the DN of the entry at the top of the naming context, and hence occurs at a branching point in your directory. You may need to revisit your namespace design.

### 2.5.3 Replication design

The replication design stage is only required when, firstly, a distributed approach is chosen to server deployment and, secondly, a replicated approach is chosen over a partitioned approach. Replication aims to improve the reliability and performance of your directory service.

#### Concepts

By making directory data available in more than one location you improve the reliability of the service in the event of server or network failure. You also improve the performance by distributing the load across multiple servers and reducing network traffic.

#### Designing replication

Consider firstly the unit of replication. This concerns which entries and which of their attributes are to be replicated. A subtree of the DIT might form a suitable selection basis. However, the current replication support as provided with OS/400 Version 5 Release 1 supports only replication of the entire directory tree.

Now think about how consistent the data has to be. Must every change be replicated instantly to all servers? Due to the nature of directory data, for example people's phone numbers, it is not usual to impose such a tight restriction, but you might take a different view of removing the entry for a dismissed member of staff. In particular, take a look at Domino's Exclusion Group capability. Think about the sort of replication schedules that might be appropriate for your directory and network. Also, if you replicate Certificate Revocation Lists (CRLs) you may want to replicate information about a revoked certificate instantly.

To ensure initial copies are in place we might use LDAP Data Interchange Format (LDIF) files to import volumes of data in batch. A more incremental approach might be used for subsequent updates. This approach is implemented with the OS/400 LDAP replication support.

What sort of replication strategy is appropriate? Is a master-replica approach suitable, with all changes being driven out from the center? The alternative is a peer-to-peer approach, which allows all servers to update their own data and subsequently to exchange it.

In the iSeries environment, OS/400 allows *shadowing* of the SDD between OS/400 servers. SecureWay allows *replication* to other SecureWay servers. Domino allows replication to other Domino directories. The OS/400 SecureWay API maintains a one-way *synchronized export* of SDD entries to SecureWay, but only for selected fields and not for shadowed entries. The SDD can be synchronized with Domino but only with Domino for AS/400 at Release 4 and Release 5 and only on the same OS/400 instance. For details of additional options to integrate these directories see Appendix E, “The BlueNotes Product Suite” on page 549.

## 2.6 Implementation planning

It is not the function of this book to cover implementation planning in any depth, but the usual considerations will apply.

You will need to consider your choice of directory software. In the case of the iSeries this may seem a foregone conclusion with the availability of SecureWay as a free, high-function integrated part of the operating system and its integration with the System Distribution Directory (SDD). However, for customers using Domino, this too is a valid alternative. An LDAP-enabled Domino Directory server does not yet possess all the function and scalability of SecureWay, most notably in the area of replication with non-Domino LDAP directories, but this is due to be addressed in future releases and it has many other reasons to commend it. If the majority of users are already registered in the Domino Directory then this may be an attractive option. With future releases of Domino it can even become the directory server to a non-Domino infrastructure.

Consider carefully if any migration planning is required, both from non-OS/400 platforms but also in particular from the SDD. In most OS/400 shops the majority of users will already be registered by default in the SDD, for instance as Client Access/400 users, albeit with limited data. However, those customers who are, or have recently been, using OfficeVision/400 may already have a wealth of data here.

The rest of implementation planning is simply the best practice that is applicable to any IT project. Remember to undertake performance and scalability testing and any subsequent tuning that may be required. You should not encounter serious performance concerns with SecureWay, which is built on the solid foundation of DB2/400. Nor should this be an issue with Domino, but there are

significant directory enhancements with Domino 6. See Appendix C, “OS/400 and Domino LDAP history” on page 527 for a checklist of LDAP-related directory functions with the various releases of the OS/400 and Domino Directory support. Develop and document procedures for maintenance of the directory data. Finally, remember to pilot with a representative group of users and to put in place the usual training and helpdesk support before going live.

## The redbook example scenario

The real advantage of implementing Directory Services is reusing directory data for various purposes in different applications. LDAP directory implementations also provide functions to improve scalability and availability. Bearing this in mind, we explain LDAP directory planning, implementation, and management by using an example scenario throughout this book.

This chapter provides an overview of the redbook scenario including the various implementation tasks.

### 3.1 Scenario overview - Stage 1

In our scenario we have a company called iSeries Shop. This company is a legacy AS/400 customer and has moved to iSeries servers. Traditionally, they used OfficeVision/400 (OV/400) to exchange electronic mail between employees. Therefore, they used the System Distribution Directory (SDD) to maintain a list of mail users including typical directory information, such as phone number, address, department, and so forth.

In recent years they established an Internet presence by providing information about the company's products and services using static Web pages. They also switched from OV/400 to PC-based mail clients using the OS/400 Post Office Protocol (POP) services as their mail server. When the company used OV/400, the user enrollment process automatically created a user profile and an SDD entry. Since this method did not work with their POP server anymore, they were looking for a solution that would automatically create an SDD entry for each new user profile on the system. The solution that solved the iSeries Shop's problem was the BlueNotes Directory Synchronisation for OS/400. Using this module, every time a new user profile is created a new SDD entry is added. For more information about this product, refer to Appendix E, "The BlueNotes Product Suite" on page 549.

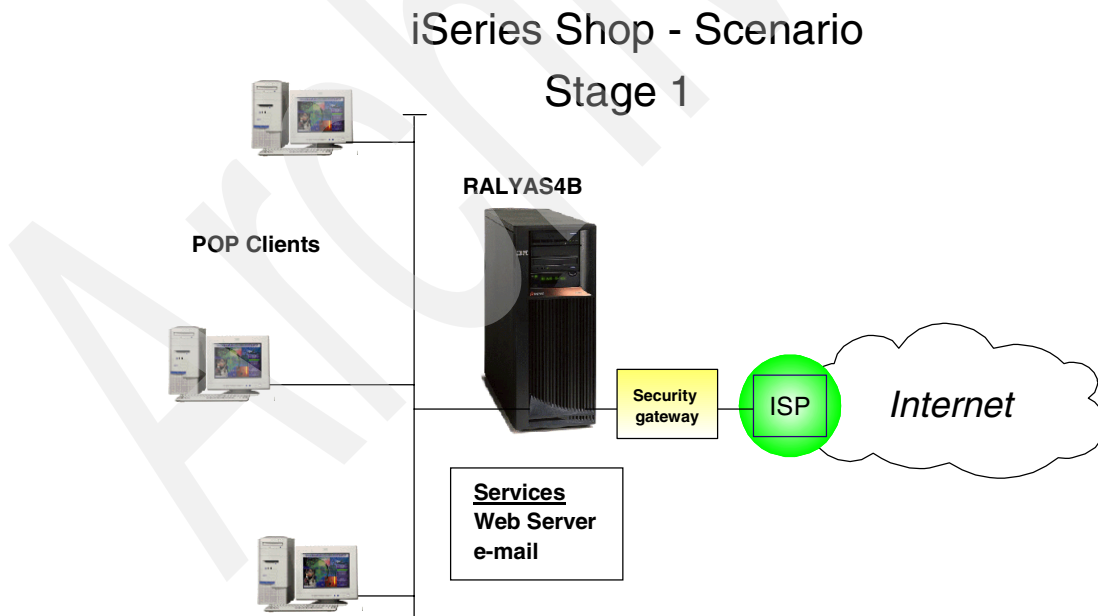


Figure 3-1 Redbook scenario - Stage 1

### 3.1.1 Stage 2 - The evolution

As mail and Web traffic increased, the company's IT infrastructure needed to be changed to provide better availability and performance. The technical support department also wanted to provide special support and help for their premium customers by adding content to their Web servers. In addition they faced another problem, the SDD information could not be used by the POP mail clients to look up e-mail addresses as they used to do with OV/400.

The new solution has the following objectives:

- ▶ Guarantee higher availability and performance of the Web server.
- ▶ Ensure that only premium customers can access restricted information posted by the technical support.
- ▶ Allow POP mail clients to look up information, such as e-mail addresses, phone numbers, and so forth.

To meet the objectives, the company implemented the following solution:

- ▶ A second iSeries server was installed and an HTTP Web server set up. The new Web server is used for load balancing to solve the performance issues. Since the primary (on system RALYAS4B) and new secondary server (on system RCHASM80) host the same information, the availability requirement is also solved. In case that one server goes down, the other server can still serve the Web pages to customers in the Internet. As part of the load-balancing and availability solution, the company installed the WebSphere Edge Server product. This product redirects incoming connection requests based on the server's network load. In addition, the product's caching capabilities serve pages right on the edge of the network and therefore off loads work from the Web servers. For more information about the WebSphere Edge Server product and its features refer to:
  - *WebSphere Edge Server: Working with Web Traffic Express & Network Dispatcher*, SG24-6172
  - *WebSphere Edge Server New Features and Functions in Version 2*, SG24-6511

To minimize the configuration and administration efforts in keeping both Web servers at the same level, the iSeries Shop's IT Department decided to set up an LDAP directory on the iSeries server RALYAS4B. They used the Directory Services that are shipped with OS/400. The OS/400 Directory Services are based on IBM's SecureWay Directory product.

The iSeries Shop used the LDAP support of the IBM HTTP Server for iSeries to read server configuration directives from an LDAP directory. Using this approach, the server configuration needs to be done only once and can be used by multiple Web servers at the same time. For availability reasons they set up a second LDAP directory server on server RCHASM80 as a replica of RALYAS4B's LDAP directory server. The connection between the master and the replica should be secured by SSL.

Refer to Section 6.4, "Configuration support" on page 254 for further information on how to store IBM HTTP Server for iSeries server directives into an LDAP directory.

Setting up a replica is described in Section 4.7, "Setting up directory replication" on page 106.

Securing the connection between master and replica is described in Section 4.9, "Securing LDAP traffic" on page 126.

- As the Technical Support Department wants to sell more service contracts, they want to provide special support information to their premium customers only. To make sure that only those customers have access to the information, they have to register and then use their userid and password to sign onto the Web site. It turned out that the IT Department's decision to set up an LDAP directory server was a very smart move. They can use the same LDAP directory to hold the information necessary to authenticate premium customers with the IBM HTTP Server for iSeries. Without the LDAP directory server, the only way to authenticate the users would be to create a user profile for each customer or store the information in an OS/400 validation list on both Web servers. With the LDAP directory server and its replica there is no need to create the user information twice.

Refer to Section 6.3, "User authentication" on page 232 for further information on how to configure LDAP user authentication for the IBM HTTP Server for iSeries.

- The last objective of stage 2 of this scenario is to allow POP mail clients to look up e-mail addresses or phone numbers as they used to do with OV/400. This problem can easily be solved by utilizing the OS/400 SDD to LDAP publishing support. This support allowed the company to publish the information stored in the SDD into an LDAP directory. Then they set up their mail client's address books to look up a person's directory entry including the e-mail address in the LDAP directory. The Human Resources Department and department managers are keeping employee directory data in the SDD up-to-date.

Refer to Section 4.5, "Publishing the System Distribution Directory" on page 79 for more information on publishing the SDD into an LDAP directory.



For information on how to configure mail clients to look up information in an LDAP directory, refer to Chapter 9, “LDAP directory: The enterprise directory for mail clients” on page 433.

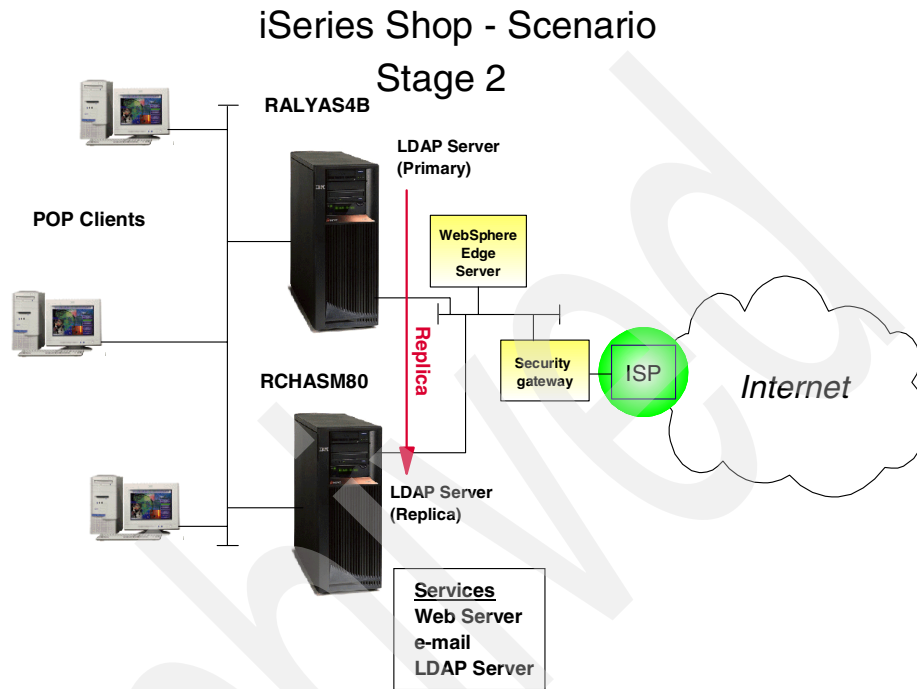


Figure 3-2 Redbook scenario - Stage 2

### 3.1.2 Stage 3 - Growing the business

The business of our iSeries Shop is thriving and they want to expand into new product areas. A small company, ACME Supply, offers exactly what the iSeries Shop is looking for to grow their business. ACME Supply also uses e-mail services. They have Lotus Domino 5.0.9 installed on their server DOMWEBC. In order for both companies to effectively communicate with each other, they want to use e-mail. One problem they need to solve is how employees of one company can look up contact information, such as e-mail addresses and phone numbers of employees of the other company. Since both companies already have directories set up it seems logical to make the required directory information of one company available to the second company in some way. Since this kind of information will not change very often, the companies do not see any reason to establish an online connection between the two directories.

The objective of stage 3 is to allow employees of the iSeries Shop to look up contact information of employees of ACME Supply and vice versa. However, one requirement of both companies is that not all directory information, such as employee number or employee type is visible to employees of the other company.

To meet the objectives, the company implemented the following solution:

Each company only needs names, phone numbers, and e-mail addresses of employees of the other company. To accomplish this, they export the needed information into LDAP Data Interchange Format (LDIF) files. The ACME Supply LDIF file is then transferred to the iSeries Shop and vice versa. This transfer and the corresponding update is performed once a week. Refer to Section 5.4, “Exporting and importing information via Operations Navigator” on page 187 for more information on using LDIF files with OS/400 LDAP Directory Services. For information on how to export and import directory data with Domino refer to Chapter 7, “Setting up LDAP on Domino server for iSeries” on page 291.

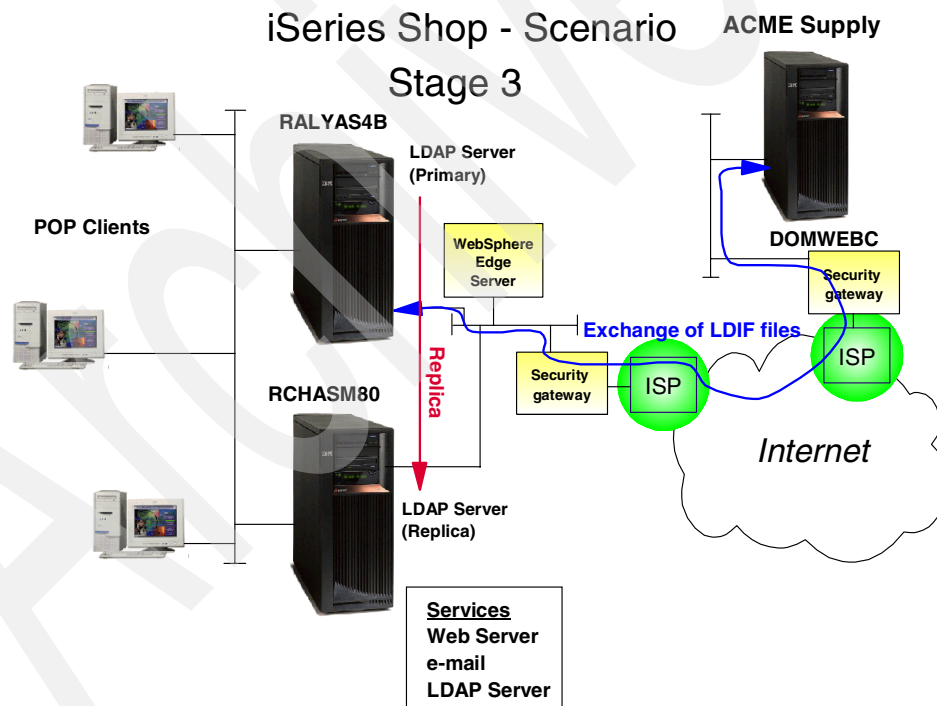


Figure 3-3 Redbook scenario - Stage 3

### 3.1.3 Stage 4 - Merging businesses

In this stage of our scenario the iSeries Shop merged with iSeries Automotive, Inc., a company that offers similar products and is very successful in the market. iSeries Automotive is, of course, also using an iSeries server to run their business applications. Both companies are interested in leveraging each other's customer base to expand their business. The new company is now responsible for all major marketing campaigns. As part of this effort, they deploy new Web applications that are running under WebSphere Application Server and Lotus Domino on their iSeries server. All business partners and customers need to register to use the new applications.

Driven by the market, iSeries Automotive implemented a Customer Relationship Management (CRM) solution on their Domino server. To offer more personalized services, they established a new electronic newsletter service for their customers. As part of the deployment process they also changed the customer registration process to collect information about their customer's preferences, such as the product area they want to receive newsletters about and whether they want to receive the information via e-mail, fax, or the Web. The newsletter service is part of their CRM solution and required that Lotus Notes also be installed. The data for the CRM database is comprised of data from their accounting application database and the customer registration process. They used the Direct Marketing product of the BlueNotes suite to import data into CRM and implement the newsletter service based on user preferences. More information about the BlueNotes product suite can be found in Appendix E, "The BlueNotes Product Suite" on page 549.

The objectives of stage 4 are:

- ▶ Minimize user authentication requests for customers who access different applications.
- ▶ iSeries Automotive Lotus Notes user should also be able to look up e-mail addresses from the LDAP directory.
- ▶ LDAP traffic between application servers should be encrypted.

To meet the objectives, the company implemented the following solution:

- ▶ The Secure Socket Layer (SSL) protocol is used for the LDAP directory server and client to protect the LDAP traffic between DOMWEBA and RALYAS4B.

How to enable SSL for the OS/400 LDAP directory server is described in Section 4.9, "Securing LDAP traffic" on page 126.

How to enable SSL for the Domino Directory is explained in Section 8.10.2, "Enabling Domino SSL with SSO" on page 406.

- ▶ Since iSeries Shop customers are already registered in the LDAP directory, the WebSphere Application Server and Domino server on iSeries

Automotive's server are set up to authenticate users against the existing LDAP directory. This also required that common information about customers registered through the new registration process is published to the existing LDAP directory on RALYAS4B. Information that is relevant to the CRM application only, for example, newsletter preferences, is stored in the CRM database and not published to the LDAP directory. The iSeries Shop's LDAP directory server on RALYAS4B did a once off import from the iSeries Automotive DOMWEBA's LDAP server via LDIF. From now on Human Resources will manage new Notes users. The Web application registration process uses APIs to publish user information into the LDAP directory on the system RALYAS4B.

Refer to Chapter 7, "Setting up LDAP on Domino server for iSeries" on page 291 for information on how export Domino LDAP to the OS/400 LDAP directory server.

- To minimize user authentication requests for customers who access different applications, iSeries Automotive implemented Single Sign-On (SSO) on their WebSphere Application Server and Domino server. Using SSO, both application servers share authentication information. For example, if a customer signs on to a Domino application, the WebSphere Application Server can reuse the authentication information and allow access without displaying a user authentication prompt again.

Refer to Chapter 8, "Single Sign-On with Domino and WebSphere 4.0" on page 337 for more information on how to configure Domino and WebSphere for Single Sign-On.

**Note:** The IBM HTTP Server for iSeries does not support Single Sign-On. Therefore, a customer would have to authenticate once to the Web server on RALYAS4B and another time for the Web applications running on the iSeries Automotive server. The Tivoli Policy Director is a product that also supports Single Sign-On. By combining the SSO capabilities of Tivoli's Policy Director, WebSphere Application Server, and Lotus Domino a user would have to authenticate only once. Features of the Tivoli Policy Director include:

- ▶ Provides access control to Web objects
- ▶ Adds centralized security to your existing Web and TCP/IP applications
- ▶ Enables replication and load balancing
- ▶ Provides a consistent, manageable access control policy
- ▶ Offers extensible authentication and authorization
- ▶ Delivers secure remote access and personalized access
- ▶ Offers one-time authentication capability with access to multiple Web resources
- ▶ Reduces administration costs
- ▶ Supports Public Key Infrastructure (PKI)

For more information about Tivoli products, refer to:

<http://www.tivoli.com>

- ▶ The Notes configuration is changed to allow iSeries Automotive Notes users to look up e-mail addresses of iSeries Shop employees. Refer to Chapter 9, "LDAP directory: The enterprise directory for mail clients" on page 433 for details on how to enable LDAP look-ups for Notes clients.

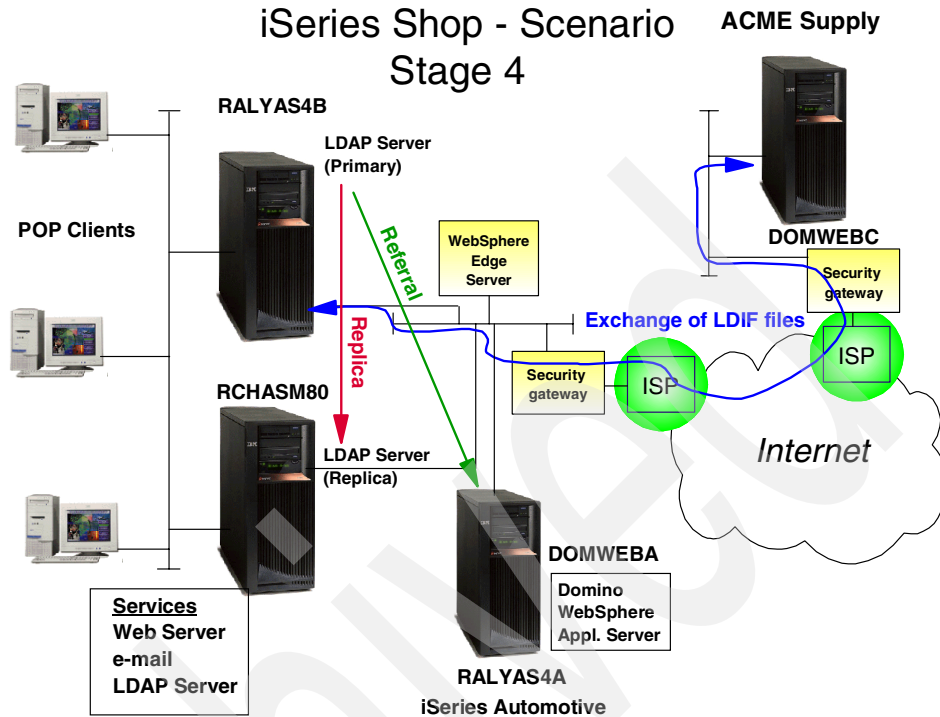


Figure 3-4 Redbook scenario - Stage 4

### 3.1.4 Stage 5 - The enterprise directory

The iSeries Shop and iSeries Automotive have both realized that their IT applications as well as employees benefit from using LDAP Directory Services. However, in the past all changes to directory information, such as a changed phone number, office number, or address were submitted to an administrator who performed the changes in the directory using the Directory Management Tool (DMT). More information about the DMT and other directory management methods are described in Chapter 5, "Managing an LDAP directory on iSeries" on page 149. To create a more efficient and faster way for updating directory changes, the company wants their employees to keep personal data up-to-date. In addition, they want to have a user-friendly interface to search and browse directory entries. The company's sales force should have access to customer and employee information while on the road.

The objectives of stage 5 are:

- ▶ Allow employees to update their personal directory entry. However, they should only be allowed to update certain information while a department manager or the Human Resources Department can update more information.
- ▶ A directory user interface should be developed to allow all employees to search and browse the directory. The solution should not involve a new application to be installed on the client.
- ▶ Travelling sales people should have offline access to customer and employee information.

The following list describes the solutions to meet the stated objectives:

- ▶ Provide a Web-based interface that allows employees to search the directory and initiate an update. The company has chosen to implement the interface with a Java servlet running on the existing WebSphere Application Server. Refer to Chapter 12, “Using the JNDI to search and update the directory” on page 477 for details on how to search and update an LDAP directory using Java.
- ▶ Browsing can be performed by all employees. Updates are controlled by Access Groups, which are different for employees, their managers, and the Human Resources Department.
- ▶ A Domino Directory containing data synchronized from the LDAP directory and the CRM application has become a valuable tool for all travelling sales people. Each sales person replicates the Domino Directory onto the Laptop. This allows access to customer and employee information while on the road and not connected to the company's network. A new module, BlueNotes Directory Taxonomy is introduced. It enables the Domino administrator to use key directory fields, such as department or location, to build a corporate hierarchy taxonomy, and to update the Domino Directory. This gives the Notes user new views of the directory based on the previously defined fields. For example, a sales person can find people of iSeries Automotive's Accounts Department without knowing full names. No new code is needed on the laptop.

### 3.1.5 The scenario Directory Information Tree

After implementing the entire scenario, the LDAP directory hosted on the main iSeries Shop system RALYAS4B has the directory information tree (DIT) shown in Figure 3-5 on page 50.

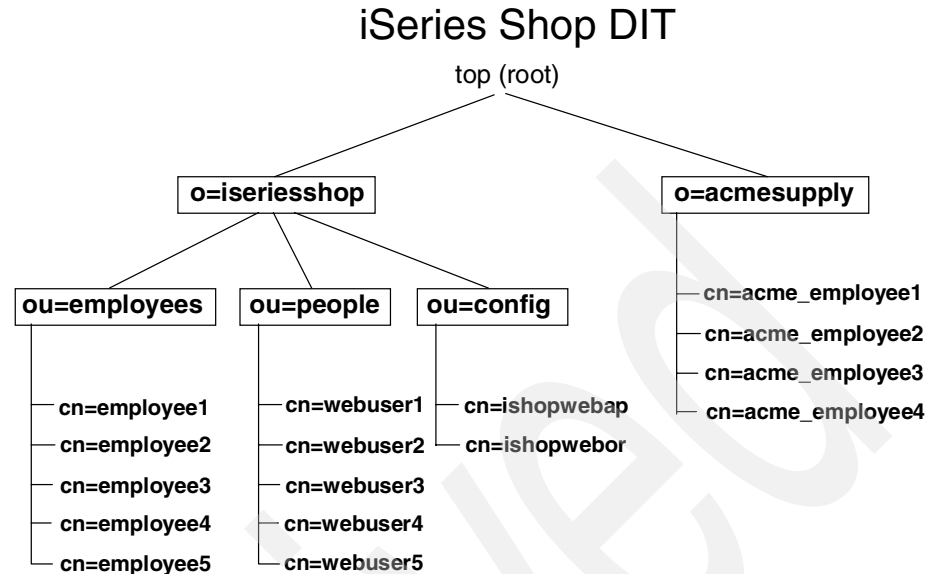


Figure 3-5 iSeries Shop DIT

The iSeries Shop has one organization (o=iseriesshop) under the root of the tree. This organization has three branches:

- ou=employees** This part of the tree holds the employee data initially published from the SDD. Later a user-written application is used to maintain the employee information. The information stored in this organizational unit will be used as a company directory to look up addresses, phone numbers, office numbers, and so on.
- ou=people** This branch of the tree represents a user registry for Web users. The WebSphere Application Server, Domino, and the HTTP Server for iSeries will use the information for user authentication.
- ou=config** The last branch under o=iseriesshop stores HTTP Server for iSeries configuration directives. In this scenario we have two Web servers, RALYAS4A and RCHASM80, that both serve the same information. One is the backup of the other, and when both are up and running they share the load. Both servers are configured to load their configuration from the LDAP directory.



The organization o=acmesupply is used to provide iSeries Shop employees with information about employees of the business partner ACME Supply. The information under this organization will be provided by ACME Supply via LDIF information exchange. The information about ACME Supply employees in the iSeries Shop LDAP directory is limited to employee names, e-mail addresses, and telephone numbers.





## Part 2

# Configuration and administration



## OS/400 LDAP Directory Services

An IBM SecureWay Directory implementation is supported on iSeries. LDAP clients and an LDAP server are provided free with Directory Services. Starting with OS/400 Version 5 Release 1, Directory Services is included with the base operating system. LDAP clients for Windows and OS/400 provide APIs for use by both C and Java applications. The OS/400 client also provides APIs for use by all Integrated Language Environment (ILE) programming languages. LDAP utilities are provided for common administrative tasks, such as searching or modifying the directory, and can be run from the OS/400 QShell command environment or a Windows command prompt. To allow mail clients to search for e-mail addresses of OS/400 users, Directory Services enables System Distribution Directory (SDD) information to be published to an LDAP directory. All IBM SecureWay Directory LDAP server implementations use the IBM Universal Database (UDB). When implemented on iSeries, this results in an LDAP directory that is scalable, robust, and easy to manage. Millions of entries can be added to the directory with little impact on performance. Back up and recovery of the LDAP directory is performed using standard OS/400 administrative procedures. Configuration of the LDAP server is made easy using a wizard within Operations Navigator in the TCP/IP servers folder for your system.

This chapter covers the following topics:

- ▶ Directory services implementation and component overview
- ▶ Configuration of OS/400 Directory Services
- ▶ Publishing system, user, and printer information
- ▶ Replication and referral configuration
- ▶ Setting up directory services for secure communication

## 4.1 Implementation and component overview

Directory Services provides a Lightweight Directory Access Protocol (LDAP) server on iSeries. LDAP runs over Transmission Control Protocol/Internet Protocol (TCP/IP), and is gaining popularity as a directory service for both Internet and non-Internet applications. You perform most setup and administering tasks of the OS/400-based LDAP directory server through the graphical user interface (GUI) of Operations Navigator. You can use Directory Services with LDAP-enabled applications, such as mail applications that look up e-mail addresses from LDAP servers.

Besides the LDAP server, Directory Services also includes:

- ▶ An OS/400-based LDAP client. This client includes a set of application program interfaces (APIs) that you can use in OS/400 programs to create your own client applications.
- ▶ Version 3.2 of the IBM SecureWay Directory Client Software Development Kit (SDK). The SDK includes a Windows LDAP client and the following tools:
  - The IBM SecureWay Directory Management Tool, which provides you with a graphical user interface for managing directory content.
  - Command line utilities (such as, `ldapsearch`, `ldapadd`, and `ldapmodify`).
  - C LDAP APIs (library files, header files, and sample source code).
  - Documentation for the IBM SecureWay Directory Management Tool (DMT) is located in the file `dpagent.htm`. This file is copied to the IBM SecureWay Directory.

From Version 4 Release 5, both the OS/400 LDAP server and the OS/400 LDAP client are based on LDAP Version 3. You can use a Version 2 client with a Version 3 server. However, you cannot use a Version 3 client with a V2 server unless you bind as a Version 2 client and use only Version 2 APIs.

The Windows LDAP client is also based on LDAP Version 3.

Because LDAP is a standard, all LDAP servers share many basic characteristics. However, due to implementation differences, they are not all completely compatible with each other. The LDAP server provided by Directory Services is closely compatible with other LDAP directory servers in the IBM SecureWay product group. However, it may not be as compatible with other LDAP servers.

The data for the LDAP server that Directory Services provides resides in an OS/400 database:

- ▶ The default library for the LDAP server is QUSRDIRDB. This library contains the Directory server's contents.
- ▶ If you configure the directory server to log directory changes, a database library called QUSRDIRCL is created.

If you want to start completely from scratch, clear those two libraries before restarting the wizard.

Beginning with Version 5 Release 1, Directory Services (LDAP) is automatically installed when you install OS/400. The directory server includes a default configuration that automatically starts the directory server when TCP/IP is started. It also starts the publishing of computer (system) information from OS/400 to the directory server. To customize the LDAP directory server's settings for your own use, run the Directory Services Configuration Wizard. You must have \*ALLOBJ and \*IOSYSCFG special authorities to use the wizard. If you want to configure OS/400 security auditing, you must also have \*AUDIT special authority.

Prior to Version 5 Release 1, it was necessary to install the OS/400 Directory Services option of the OS/400 to install the directory server.

**Important if you are using Directory Services prior to Version 5 Release 1:** As noted previously, Directory Services is integrated in the base operating system beginning with Version 5 Release 1, so you do not need to install this option.

However, you will still need to install it for compatibility reasons if you want to be able to do any of the following:

- ▶ Use Version 4 Release 4 of the Operations Navigator with the LDAP server.
- ▶ Call the LDAP APIs from the QDIRSRV library (where they were called from prior to Version 5 Release 1 and a copy still resides) instead of from the QSYS library. Unless your custom applications specifically point to the QDIRSRV library, this will not be an issue. The QUSAPIBD binding directory points to the QSYS library. It is recommended that you use QUSAPIBD with your applications.

If later your application no longer need the LDAP APIs from the QDIRSRV library, you can then uninstall them.



## 4.1.1 OS/400 Directory Services jobs

OS/400 Directory Services and publishing clients are comprised of several jobs. They are:

### **QDIRSRV**

The Directory Services server job QDIRSRV is running in subsystem QSYSWRK. This is the only job running for the directory server. This job will contain information about failing client requests and replication errors. Client errors include all requests that do not complete successfully and might include bind failures, attempts to delete objects that do not exist, schema errors if a client application is missing required attributes, and so on. We should point out that an error message here means only that the server did not return a successful return code to the client. It does not necessarily indicate a failure. For example, an application may assume an object exists and create it only in the event of an error, rather than doing a search first. Look in this job log when you have reason to believe an error has occurred, rather than assuming that a message here indicates an error.

### **QGLDPUBA**

This job also runs under the subsystem QSYSWRK and takes care of the synchronization between the System Distribution Directory (SDD) and the LDAP directory server. This job acts as the publishing agent for user SDD and system information, generating LDAP requests that are put into a publishing queue. However, if changes are made in the LDAP directory, these changes are not synchronized back to the System Distribution Directory.

### **QGLDPUBE**

This job acts as the publishing engine, taking changes from the queue (put there by QGLDPUBA and other publishing agents - printers, user-defined agents), and processes the request using the server, authentication, and location information defined in the agent configuration. If the requests is successful, the change is removed from the queue. If it fails (for example, because the server was down), the request is left in the queue to be retried.

All these jobs are running in the QSYSWRK subsystem. You can check these jobs by either of the following methods:

- Using the OS/400 command **WRKACTJOB SBS(QSYSWRK)**, as shown in Figure 4-1 on page 60.

- Navigate through Operations Navigator to **System -> Work Management -> Server Jobs**.

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help

Work with Subsystem Jobs                                AS4B
                                                         02/08/02 14:07:55

Subsystem . . . . . : QSYSWRK

Type options, press Enter.
  2=Change  3=Hold  4=End  5=Work with  6=Release  7=Display message
  8=Work with spooled files 13=Disconnect

Opt Job      User      Type  -----Status----- Function
---
CRTPFRTA  QSYS      BATCH  ACTIVE  CMD-CRTPFRDTA
QCQEPMON  QSVMS      BATCH  MSGW    PGM-QCQEPMON
QCQRCVDS  QSVMS      BATCH  MSGW    PGM-QCQAPDRM
QDIRSRV   QDIRSRV    BATCH  ACTIVE  PGM-QGLDSVR
QECS      QSVSM      BATCH  ACTIVE  PGM-QNSECSJB
QGLDPUBA  QDIRSRV    AUTO   ACTIVE  PGM-QGLDPUBA
QGLDPUBE  QDIRSRV    AUTO   ACTIVE  PGM-QGLDPUBE
QHODSYM   QHOD       PJ      ACTIVE

More...

Parameters or command
====>
F3=Exit    F4=Prompt  F5=Refresh  F9=Retrieve  F11=Display schedule data
F12=Cancel F17=Top    F18=Bottom

MA b MW
Connected to remote server/host: ralyas4b.iseries.itso.ra1.ibm.com using port: 23

```

Figure 4-1 WRKACTJOB SBS(QSYSWRK) command output

These jobs are running under the QDIRSRV user profile.

**Tip:** The publishing API that causes the System Distribution Directory to get published is called from the QGLDPUBA job. If this job is not running, it will need to be started in order for LDAP to synchronize with the directory entries. This job and the QGLDPUBE job is normally started with the QSYSWRK subsystem; however, if they are ended, they can be restarted with the following commands:

```

QSYS/SBMJOB CMD(CALL PGM(QSYS/QGLDPUBA)) JOBD(QSYS/QGLDPUBA)
USER(*JOB) SYSLIBL(*SYSVAL) CURLIB(*USRPRF) INLLIBL(*JOB)
MSGQ(*NONE)

```

```

QSYS/SBMJOB CMD(CALL PGM(QSYS/QGLDPUBE)) JOBD(QSYS/QGLDPUBE)
USER(*JOB) SYSLIBL(*SYSVAL) CURLIB(*USRPRF) INLLIBL(*JOB)
MSGQ(*NONE)

```

## 4.1.2 Saving and restoring Directory Services information

Directory Services stores information in the following locations:

- ▶ The database library (QUSRDIRDB by default). This library contains the directory server's contents. Use Operations Navigator and display the Directory Services properties to find out whether you are using the default library or a different one. The library name can be found on the Database/Suffixes tab in the Database library parameter.
- ▶ The QDIRSRV2 library, which is used to store publishing information.
- ▶ The QUSRSYS library, which stores various items in objects beginning with QGLD (specify QUSRSYS/QGLD\* to save them).
- ▶ If you configure the directory server to log directory changes, a database library called QUSRDIRCL is created.

If the contents of the directory change regularly, you should save your database library and the objects in it on a regular basis. Configuration data including the directory schema is also stored in the directory /QIBM/UserData/OS400/Dirsrv/.

### Directory Services and OS/400 journaling support

Directory Services uses OS/400 database support to store directory information. Directory Services uses commitment control to store directory entries in the database. This requires OS/400 journaling support. When the server or the LDIF import tool is started for the first time, the following are built:

- ▶ A journal
- ▶ A journal receiver
- ▶ Any database tables needed initially

The journal QSQJRN is built in the database library (default is QUSRDIRDB) that you have configured. The journal receiver QSQJRN0001 is initially created in the database library that you have configured. Your environment, directory size and structure, or save and restore strategy may dictate some differences from the defaults, including how these objects are managed and the size threshold used. You can change journaling command parameters if necessary. You can automatically delete the LDAP journal receivers by performing the following command from an OS/400 command line:

```
CHGJRN JRN(1dap-library/QSQJRN) DLTRCV(*YES)
```

Where 1dap-library is the name of your database library configured for LDAP.

If the change log is configured, you can delete its journal receivers with the following command:

```
CHGJRN JRN(QUSRDIRCL/QSQJRN) DLTRCV(*YES)
```

**Note:** Beginning with Version 5 Release 1, Directory Services also provides security auditing capabilities. However, the QSQJRN has nothing to do with the security audit journal.

## 4.2 Installation prerequisites

In this section we explain what hardware and software prerequisites are needed to successfully install the OS/400 LDAP Directory Services.

### 4.2.1 Hardware requirements

There are no special hardware requirements for LDAP Directory Services. You have only to decide how large your directory will be, so that you can estimate how much storage you need.

The size of the directory depends on the following:

- ▶ The number of attributes in the server's schema
- ▶ The number of entries on the server
- ▶ The type of information that you store on the server

For example, an empty directory that uses the default Directory Services schema requires approximately 10 MB of storage space. A directory that uses the default schema and that contains 1000 entries of typical employee information requires about 30 MB of storage space. This number will vary depending on the exact attributes that you use. It will also increase greatly if you store large objects, such as pictures, in the directory.

### 4.2.2 Software requirements

The following software requirements have to be met:

- ▶ 5722-SS1 - OS/400 V5R1

For Version 5 Release 1 there is no additional software installation to be done because Directory Services is integrated in the base operating system.

For OS/400 releases Version 4 Release 5 and earlier, or for compatibility reasons, you must install the OS/400 - Directory Services option of OS/400.

To install the OS/400 Directory Services option, take these steps:

- a. Insert the CD-ROM that contains OS/400.
- b. Type `G0 LICPGM` on the OS/400 command line, then press Enter.

- c. Choose **option 11** from the Work with Licensed Programs menu; then press Enter.
- d. Enter 1 in the Option field to the left of Option 32, OS/400 - Directory Services, then press Enter.
- e. In the Installation device field, enter the name of the CD-ROM drive where you inserted the OS/400 CD-ROM.
- f. Press Enter.

You can also use the command **RSTLICPGM LICPGM(57xxSS1) DEV(OPT01) OPTION(32)** from the 5250 command line. Where 57xxSS1 is the version of OS/400, for example, 5722SS1 is Version 5 Release 1.

- ▶ Directory Services supports the use of Secure Sockets Layer (SSL) and Digital Certificates for communication security. If you plan to enable SSL on the LDAP directory server you need:
  - Digital Certificate Manager (5722SS1 Option 34).
  - Cryptographic Access Provider 56-bit for AS/400 (5722-AC2) or Cryptographic Access Provider 128-bit for AS/400 (5722-AC3).
  - To use an SSL connection when you administer your LDAP directory server from Operations Navigator, or to use SSL with the Windows LDAP client, you must have the Client Encryption 56-bit (5722CE2) or Client Encryption 128-bit (5722CE3) installed on your PC.
- ▶ IBM AS/400 Operations Navigator
 

You perform most setup and administration tasks of the OS/400-based LDAP directory server through the graphical user interface (GUI) of Operations Navigator. To administer Directory Services, you must have Operations Navigator installed on a PC that is connected to your system.
- ▶ The IBM SecureWay Directory Management Tool (DMT) provides you with a graphical user interface for managing LDAP directory content. The DMT is part of the Windows LDAP client that is included with Directory Services. The client is shipped in an integrated file system directory. The DMT includes also the PC ldap utilities:
  - ldapadd and ldapmodify utilities, which add and modify LDAP directory entries.
  - ldapdelete utility, which removes entries from the LDAP directory.
  - ldapsearch utility, which searches the LDAP directory for entries.
  - ldapmodrdn utility, which allows you to change the relative distinguished name (RDN).

See Chapter 5.2, “Using the DMT to manage the directory” on page 150 for more information and how to install the DMT on the PC.

## 4.3 Configuring OS/400 Directory Services

Beginning with Version 5 Release 1, Directory Services (LDAP) is automatically installed when you install OS/400. The directory server includes a default configuration that automatically starts the directory server when TCP/IP is started. It also starts publishing of computer information from OS/400 to the directory server. To customize the LDAP directory server's settings for your own use, run the Directory Services Configuration Wizard. You must have \*ALLOBJ and \*IOSYSCFG special authorities to use the wizard. If you want to configure OS/400 security auditing, you must also have \*AUDIT special authority.

### 4.3.1 First-time configuration

If your system has not been configured to publish information to another LDAP server and no LDAP servers are known to the TCP/IP DNS server, then Directory Services is automatically installed with a limited default configuration. Directory Services provides a wizard to assist you in configuring the LDAP directory server for your specific situation. You may run this wizard as part of EZ-Setup, or run it later from Operations Navigator. Use this wizard when you initially configure the directory server.

1. Launch the Operations Navigator.
2. Expand the system that you want to use as the LDAP server.
3. Expand **Network** and then **Servers**.
4. Click **TCP/IP**. This will show all the TCP/IP server that exist on the system. They may be running, not running, or even not yet configured, as shown in Figure 4-2 on page 65 for the Directory server.

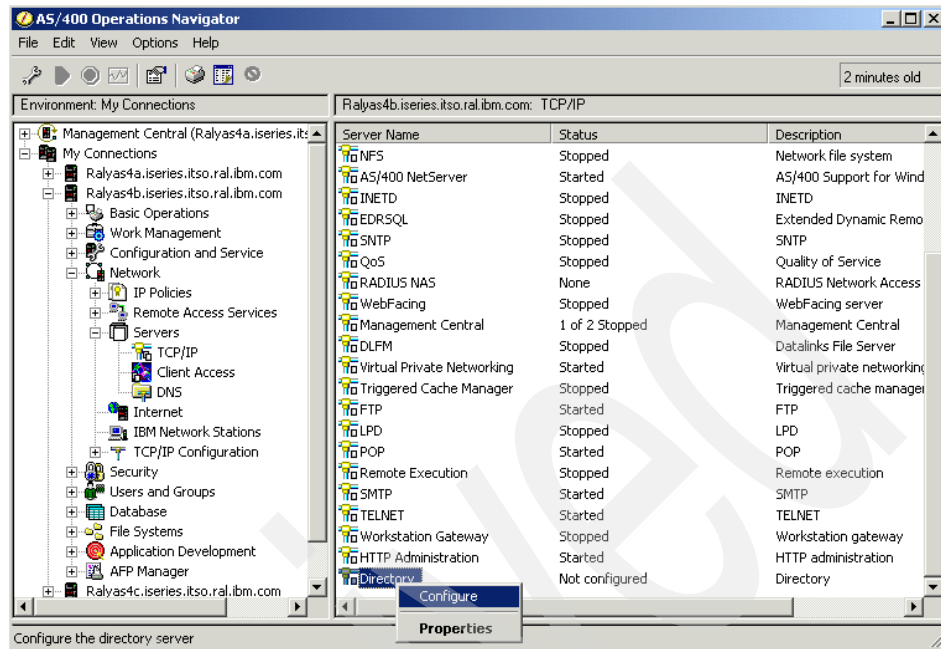


Figure 4-2 Directory server first-time configuration

5. Right-click **Directory** and click **Configure**. This will start the Directory Services Wizard as shown in Figure 4-3.

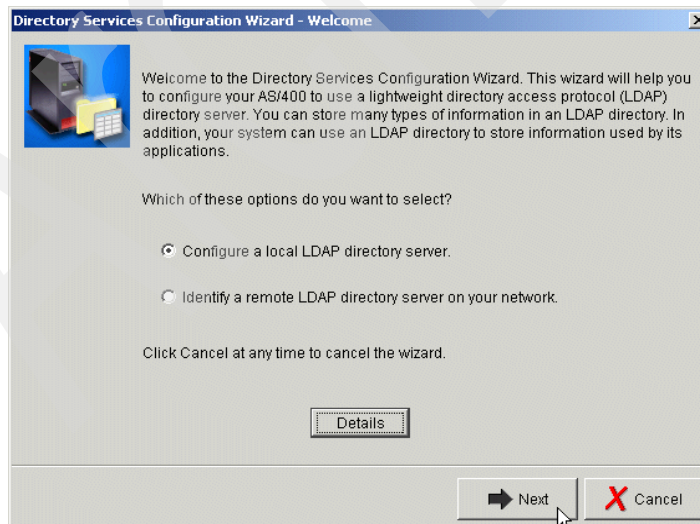


Figure 4-3 Directory Services Configuration Wizard - Welcome window

6. Select **Configure a local LDAP directory server**. This option of the wizard will guide you through the LDAP server settings.

Select the **Identify a remote LDAP directory server on your network** option when you have already set up another LDAP directory in your network. The wizard will then ask for the server address, port, distinguished name, and so forth, of the existing server. It will also ask what kind of information you want to publish to the existing server (SDD, printer, and so forth). Note that you can only use this option when the system has not been configured to publish any information to an LDAP server yet, whether local or remote. We do not use this option in this scenario.

7. Click **Next**.

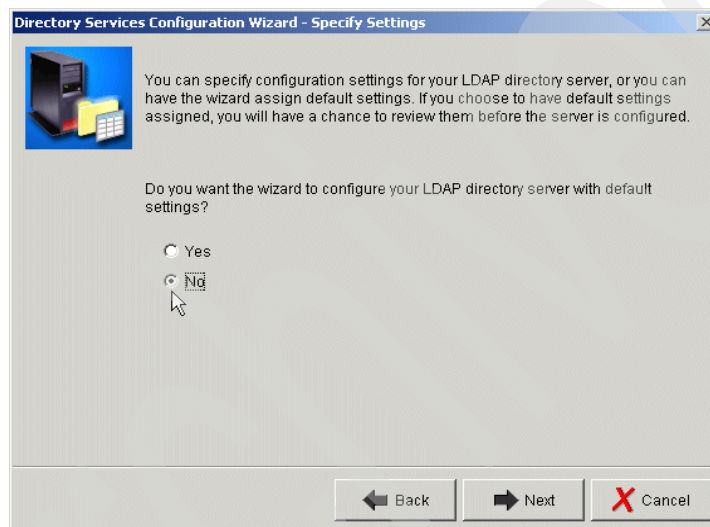


Figure 4-4 Directory server configuration Wizard - Specify Settings

8. Select **No**. If you select **Yes**, then the system will use the TCP/IP system name as a template for the suffix. In this scenario we use another suffix.
9. Click **Next**.





Figure 4-5 Directory Services Configuration Wizard - Specify Administrator DN

Uncheck **System-generated**.

Enter the administrator DN and password.

This user will be the administrator for the directory server. This userid is only used inside the directory server and is not related to an OS/400 user profile.

Note that you can change the administrator DN and password at a later time by opening the Directory Services properties and overriding the information. You need to perform this action with a user profile that has the authorities as listed at the beginning of this section.

10. Click **Next**.

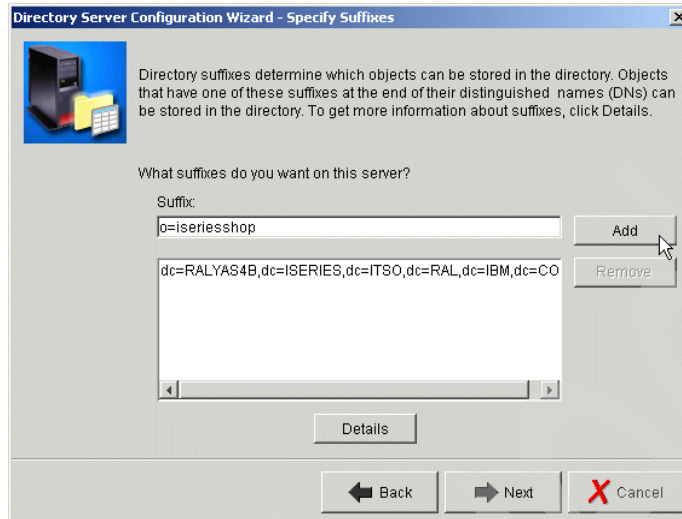


Figure 4-6 Directory Services Configuration Wizard - Specify Suffixes

Here you can see the suffix from the TCP/IP System name. We will not use this suffix. Instead we add our own suffix for our company iSeries Shop as described in Chapter 3, “The redbook example scenario” on page 39.

Enter the suffix you want to use in the Suffix textbox.

Without a suffix you will not be able to create any entry in the directory. In this example, we add a suffix `o=iseriesshop`. Adding the suffix does not automatically add the organization `iseriesshop` to the directory. It rather allows us to add the organization under the root of the directory. You still need to create the organization object in the directory. Beginning with Version 5 Release 1, the system will add the organization to the directory in case you publish system information and the suffix does not exist.

11. Click **Add**.

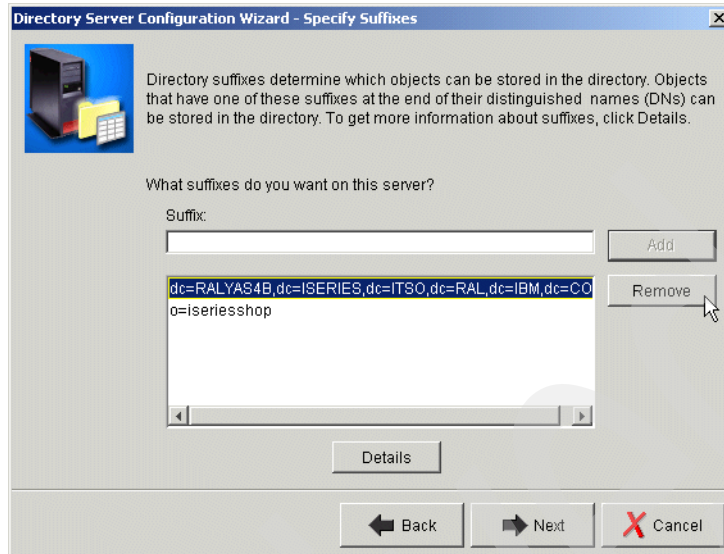


Figure 4-7 Directory Services Configuration Wizard - remove suffix

Since we do not use the system generated suffix, we are going to remove it from the list of suffixes.

12. Select the suffix created from the TCP/IP host name.
13. Click **Remove**.
14. After the TCP/IP host name is removed, click **Next**.



Figure 4-8 Directory Services Configuration Wizard - Specify TCP/IP Preference

15. Select **Yes** for the question of whether you want to start the Directory server when TCP/IP starts.

16. Click **Next**.

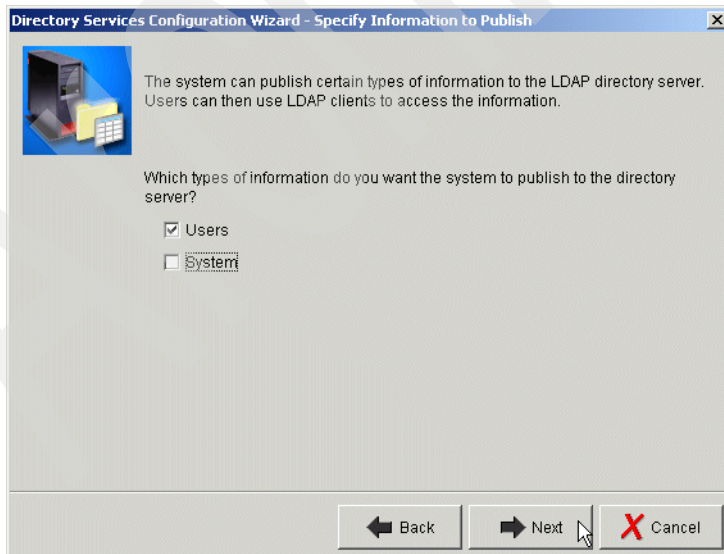


Figure 4-9 Directory Services Configuration Wizard - Specify Information to Publish

In our scenario we want to publish users registered in the System Distribution Directory to the LDAP server.

Select **Users**. Note that the wizard configures the system so that it publishes SDD entries under a DN `cn=users,o=iseriesshop`. In some cases this might cause problems. For example, when configuring Domino ACLs, Domino expects a default naming convention where the `cn` (commonName) is actually the username itself and not a sort of organizational unit. Chapter 8, “Single Sign-On with Domino and WebSphere 4.0” on page 337 covers Domino ACLs and the format it usually expects in more detail. In these cases you may want to publish SDD information into an object class `organizationalUnit`, such as `ou=users,o=iseriesshop`. Changing the publishing settings is described in Section 4.5 “Publishing the System Distribution Directory” on page 79.

17. Click **Next**.

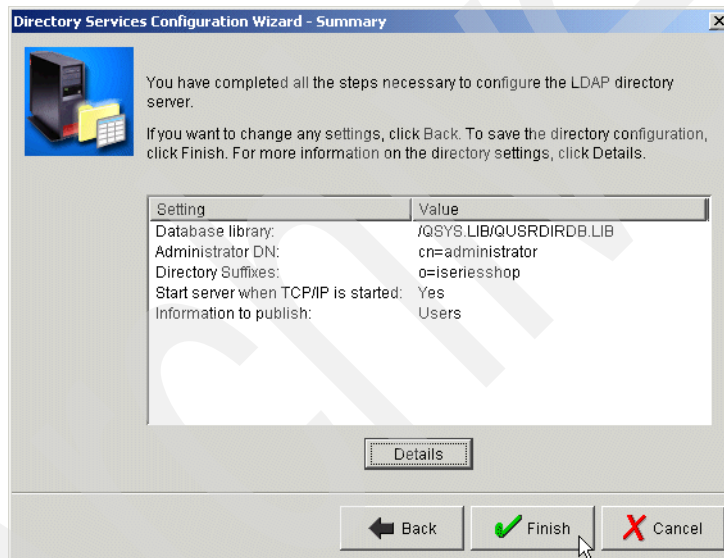


Figure 4-10 Directory Services Configuration Wizard - Summary

18. Click **Finish** to configure the LDAP directory.

19. Start the LDAP directory server via the AS/400 Operation Navigator or with the command **STRTCPSVR SERVER(\*DIRSRV)**.

You can do a first test of the LDAP directory server even if there is no data published to the server. You can do this with the Netscape Web browser or with the IBM SecureWay Directory Management Tool.

To test the connection using a Netscape browser perform the following:

Open a Netscape Web browser and enter the following URL:

`ldap://servername/`

Where `servername` represents your iSeries hostname.

In this example, `ldap://ralyas4b/`.

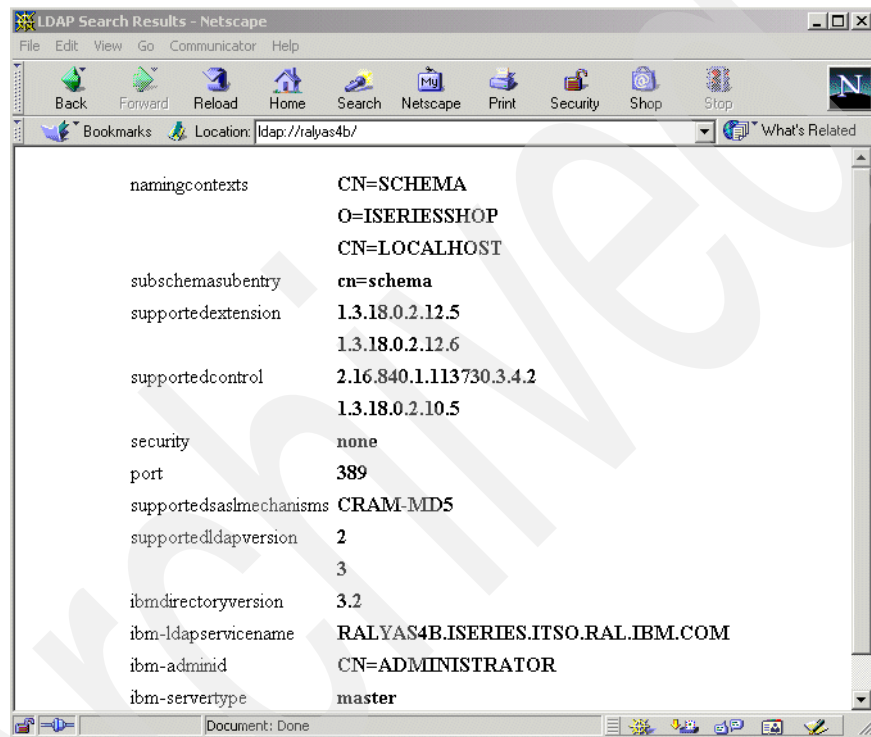


Figure 4-11 Directory server first test

This shows you the base configuration of the Directory server.

**Attention:** This only works with Netscape Web browsers.

The second way of testing the connection is using the Directory Management Tool (DMT) as described in the following steps. For information on how to install the DMT refer to Section 5.2 “Using the DMT to manage the directory” on page 150.

1. Start the IBM SecureWay Directory Management Tool. By default, you start the tool using following path: **Start -> Programs -> IBM SecureWay Directory -> Directory Management Tool.**
2. Add a server connection for your LDAP server as shown in Figure 4-12.

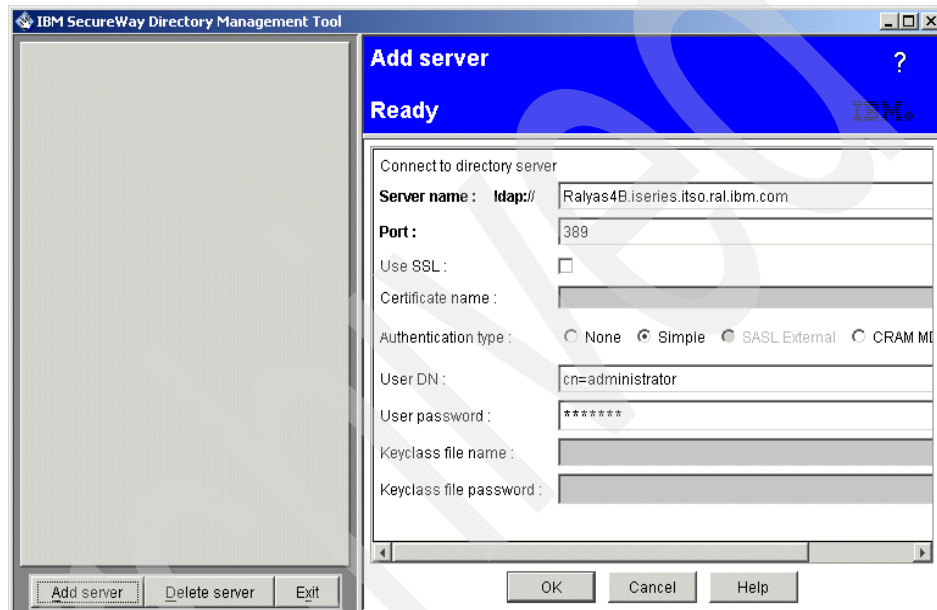


Figure 4-12 Directory server - IBM SecureWay DMT add system

3. Click **OK** to bind to the LDAP server. As we do not have the organization iserieshop added to the directory yet, an error as shown in Figure 4-13 on page 74 is displayed.

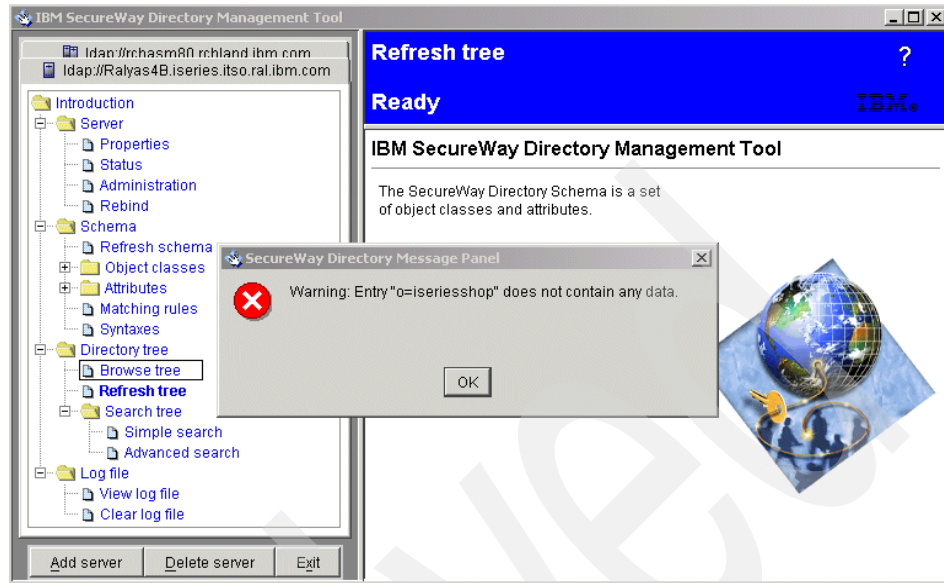


Figure 4-13 Directory server - IBM SecureWay DMT empty directory

This is correct because the LDAP directory server wizard does not publish any information to the directory.

If the Directory server is not started you will receive the message from the IBM SecureWay Directory Management Tool shown in Figure 4-14. Start the Directory server and try again.



Figure 4-14 Directory server - IBM SecureWay DMT server not started error

### 4.3.2 Reconfiguring Directory Services

You may also use the wizard to reconfigure the directory server. Perform the following steps to perform the reconfiguration:

1. Launch the Operations Navigator.



2. Stop the LDAP directory server via the Operation Navigator or with the command **ENDTCPSVR SERVER(\*DIRSRV)**.
3. Expand the system that you want to use as the LDAP server.
4. Expand **Network** and then **Servers**.
5. Click **TCP/IP**.
6. Right-click **Directory** and select **ReConfigure**. This will start the Directory Services Wizard for reconfiguration.

The wizard is basically the same as for the first-time configuration.

**Note:** When you use the wizard to reconfigure the directory server, you start configuring from scratch. The original configuration is deleted rather than changed. The directory data, however, is not deleted even if you delete a suffix that contains data. The data for the deleted suffix remain stored in the library that you selected upon installation (QUSRDIRDB by default). If you add, for example, at a later time the previously deleted suffix again, the data appears again under that suffix. The change log also remains intact, in the QUSRDIRCL library, by default.

## 4.4 Publishing system information

You can configure your system to publish certain OS/400 information into an LDAP directory server on the same system or on a different system. OS/400 will then automatically publish this information to the LDAP directory server every five minutes. Information that you can publish includes system (systems and printers), print shares, and user information. Beginning in Version 5 Release 1, if the parent DN to which the data is being published does not exist, Directory Services will automatically create it. You may have also installed other OS/400 applications that publish information in an LDAP directory. Additionally, you can call application program interfaces (APIs) from your own programs to publish other types of information to the LDAP directory.

**If you are in Version 4 Release 4:** PTF SF55507 will reduce the time between publishing from one hour to five minutes if applied. The PTF is SF55507, and the system needs to be IPLed in order for it to be applied. This only affects publishing of users, not of computer information.

#### 4.4.1 Publishing OS/400 system information

As already mentioned, beginning with Version 5 Release 1 when no LDAP directory or publishing is configured on a system, the Directory Services server automatically performs a basic configuration of the OS/400 Directory Services. This includes adding a default suffix and publishing system information to the directory. The default path under which the system information is published is:

`cn=Computers,dc=RALYAS4B,dc=ISERIES,dc=ITSO,dc=RAL,dc=IBM,dc=COM`

Where the domain components (dc) represent the fully-qualified TCP/IP host name as specified in the CFGTCP menu option 12. System information itself is published under the `cn=Computers` (object class container). All this configuration is done by the system for you when the Directory Services server is started the first time.

The following examples shows the information that OS/400 publishes as system information. The data, as shown in Figure 4-1 on page 76, was captured using a Netscape browser using the URL:

`ldap://ralyas4b/dc=ralyas4b,dc=iseries,dc=itso,dc=ral,dc=ibm,dc=com??sub?`

*Example 4-1 Data published as system information*

---

```
RALYAS4B.ISERIES.ITSO.RAL.IBM.COM
Object Class top
               cimManagedElement
               cimManagedSystemElement
               cimLogicalElement
               eSystem
               eComputerSystem
               publisher
nameformat    IP
Notes         AS/400 computer system
hostname      RALYAS4B.ISERIES.ITSO.RAL.IBM.COM
publishername dc=RALYAS4B,dc=ISERIES,dc=ITSO,dc=RAL,dc=IBM,dc=COM
```

```
OS400
Object Class  top
               cimManagedElement
               cimManagedSystemElement
```

```

                                cimLogicalElement
                                eSystem
                                eOperatingSystem
                                publisher
systemname                      OS400
ostype                          11
version                         5.1.0
publishername                   dc=RALYAS4B,dc=ISERIES,dc=ITS0,dc=RAL,dc=IBM,dc=COM

```

---

In this example there is an entry for the iSeries system itself and, as a child, an entry for the OS/400 release.

Perform the following steps if you want to change the publishing settings for system information:

1. Start Operations Navigator.
2. Select the system you want to change the settings for and then select **Properties**.
3. Select the **Directory Services** tab.

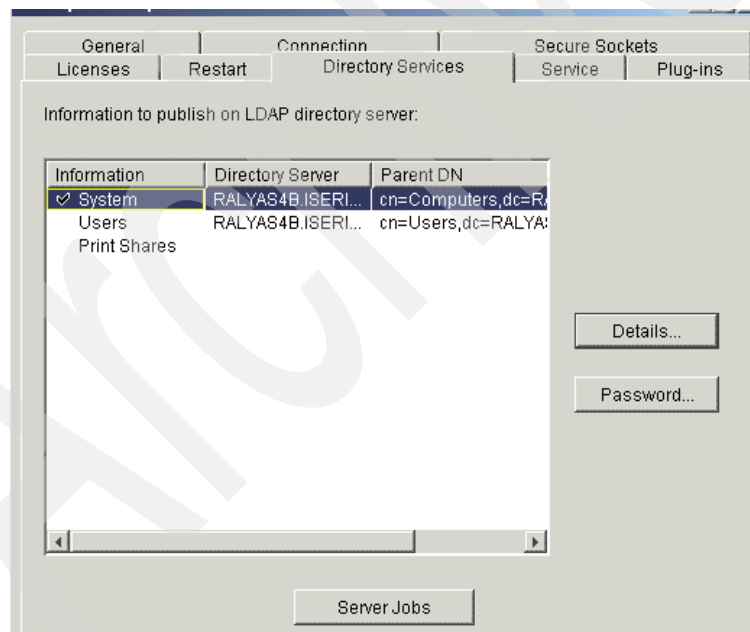


Figure 4-15 System properties window - Directory Services tab

4. Select **System** in the Information column and click **Details**.

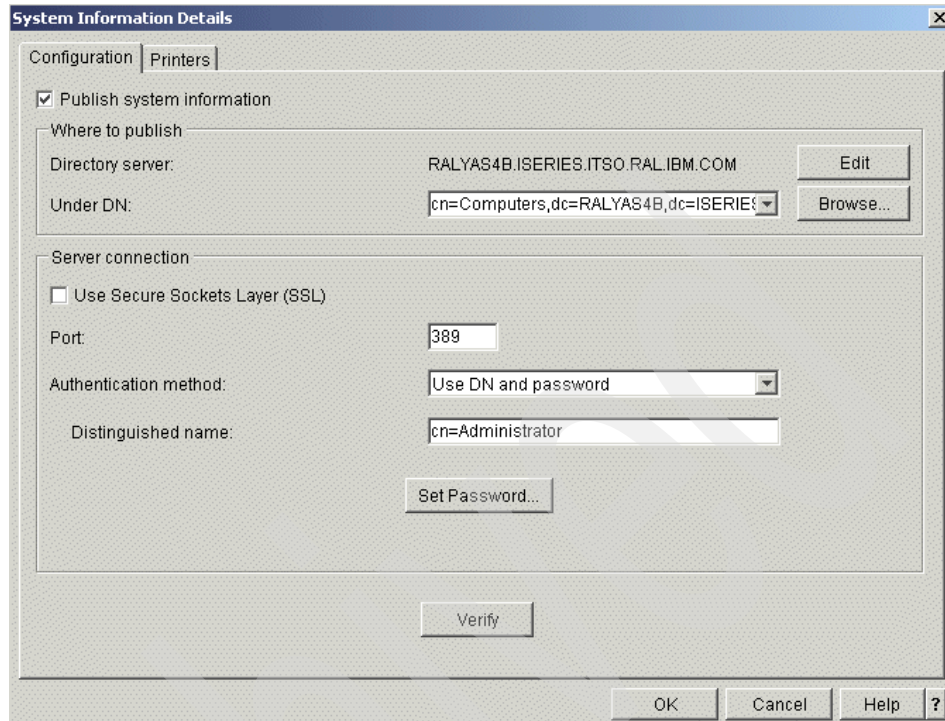


Figure 4-16 System Information Details window

From this window you can specify the LDAP directory server's host name by clicking the **Edit** button next to the name. You can also change the DN under which system information is to be published. For example, if you decide in our scenario to publish system information, you could set the DN to `ou=computers,o=iseriesshop`. The settings you make here strongly depend on your directory tree structure and schema. You can also specify whether the communications from the publishing client to the LDAP directory server should be protected via Secure Sockets Layer (SSL). If you want to use SSL, you need to assign a certificate to the Directory Services publishing (QIBM\_GLD\_DIRSrv\_PUBLISHING) client. By default, the system configures the authentication method for DN and password. The administrator common name is then being used. As an alternative you can also specify to perform authentication via Kerberos.

After you changed your settings and before saving them, you should verify that the connection values and credentials are correct by clicking **Verify**. The verification process also checks whether the publishing DN exists on the target server. If not, it will ask you whether you want to create it. After the DN is created you get a message similar to the one shown in Figure 4-17 on page 79.



Figure 4-17 Directory Services Settings Verification window

Click **OK** to close the status message window.

5. On the System Information Details window click **OK** to save your settings.

## 4.5 Publishing the System Distribution Directory

This section describes how to configure the iSeries server to publish System Distribution Directory (SDD) information into an LDAP directory.

### 4.5.1 System Distribution Directory to LDAP mapping

LDAP uses the distinguished name (DN) as the unique name for an entry. For the System Distribution Directory entries in LDAP, the DN is the common name (cn) combined with the directory path that is configured.

The System Distribution Directory entry is exported to the LDAP directory by using the inetOrgPerson object class and the ePerson object class. Figure 4-1 describes the mapping of System Distribution Directory fields to attributes of the inetOrgPerson and ePerson object class.

Table 4-1 Mapping of System Distribution Directory fields

System Distribution Directory field	LDAP attribute
User profile	uid
Description	description
Last name	sn (surname), cn (common name)

System Distribution Directory field	LDAP attribute
First name	givenName, cn (common name)
Preferred name	cn (common name)
Full name	cn (common name)
User ID	cn (common name)
Department	departmentNumber
Job title	title
Telephone number 1 and 2	telephoneNumber
FAX telephone number	facsimileTelephoneNumber
Office	roomNumber
Address lines 1-4	registeredAddress
SMTP name	mail

The common name (cn) will use the following formats:

- ▶ 'First name' 'Middle Name' 'Last name'
- ▶ 'Preferred name' 'Last name'
- ▶ 'Full name'
- ▶ 'UserID'

For example:

A user with the first name of Jonathan, preferred name of John, middle name of T, last name of Smith, and user ID of JSMITH, would have the following common names:

- ▶ cn=Jonathan T. Smith
- ▶ cn=John Smith
- ▶ cn=Smith, Jonathan T. (John)
- ▶ cn=JSMITH

The distinguished name (DN) of the published entry is the first common name (cn) combined with the directory path.

For example:

If the directory path is ou=employees, o=iseriesshop, the distinguished name (dn) for this user would be cn=Jonathan T. Smith,ou=employees,o=iseriesshop.

If you have two users in the System Distribution Directory that will resolve to the same DN, they will overlay each other in the LDAP server. Sometimes overlaying names is what you want if you are merging multiple OS/400 SDDs into one LDAP server. If you have different users with the same name, ensure they have different distinguished names to prevent overlaying each other.

**Tip:** The best time to clean up name conflicts is before publishing the SDD the first time.

## 4.5.2 Scenario objectives

As mentioned in Chapter 3, “The redbook example scenario” on page 39 the iSeries Shop used OfficeVision/400 for their electronic mail exchange for many years. As all user information is already available in the SDD, it seems logical to reuse the information for e-mail address look up and other purposes via their LDAP directory. The objectives of the iSeries Shop are:

- ▶ Make the user information from the SDD available to mail clients, such as Netscape Messenger or MS Outlook via LDAP Directory Services.
- ▶ At the beginning, all user information will still be maintained in the SDD.

## 4.5.3 Setting up SDD publishing

If you use the Directory Services configuration wizard and have chosen to publish user information, SDD publishing is already set up. However, as mentioned before, the default directory path under which SDD entries will be published would then be `cn=users,o=i seriesshop`. This is not what we want. As shown in Section 3.1.5 “The scenario Directory Information Tree” on page 49, the employee information is supposed to be stored under `ou=employee,o=i seriesshop`. The following steps will change the configuration to meet the scenario requirements:

1. Start the Operations Navigator.
2. Select and right-click **System (RALYAS4B)** in the AS/400 Operations Navigator.

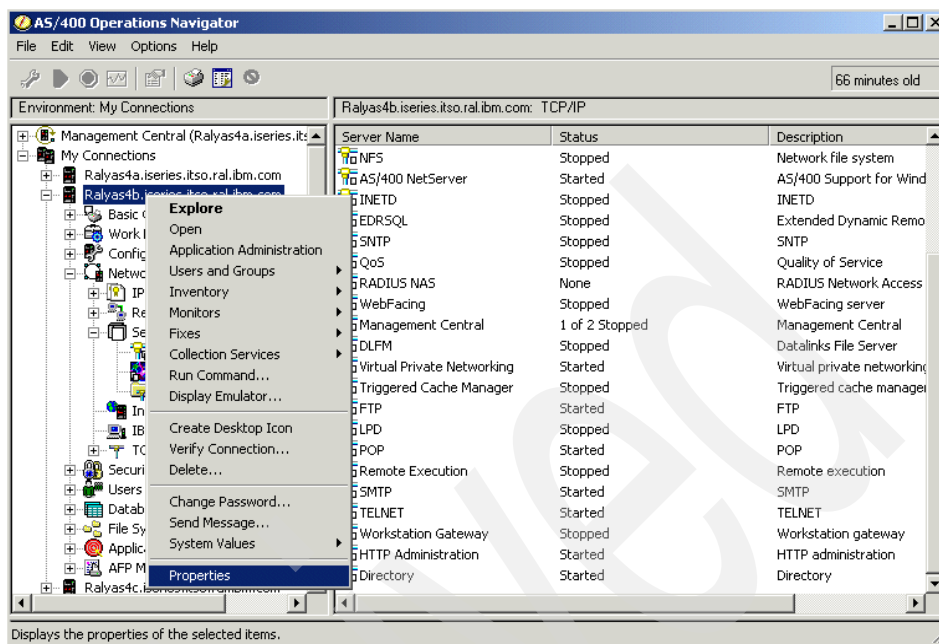


Figure 4-18 Publishing the System Distribution Directory

3. Select **Properties**. This will open the Systems Properties window.
4. Click the **Directory Services** tab in the System Properties window.



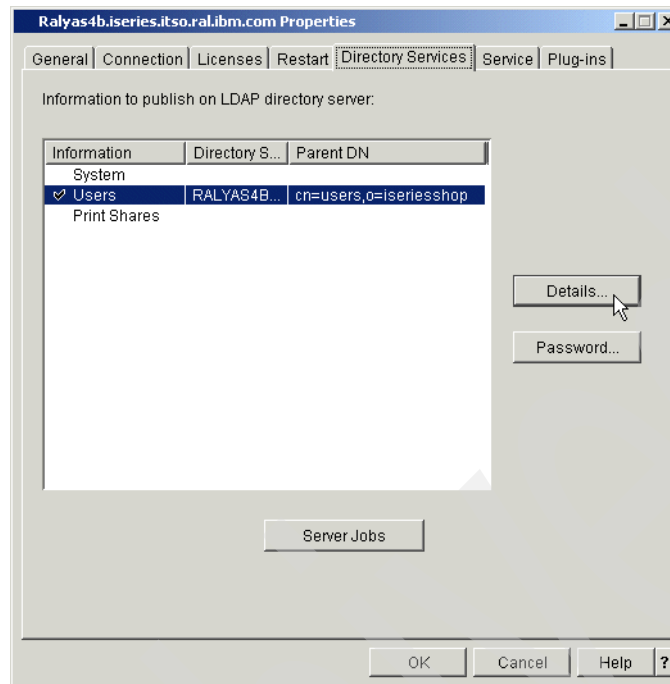


Figure 4-19 RALYAS4B Properties - Directory Services details

5. Select **Users** and click **Details**. This will open the Directory Services user information details window.

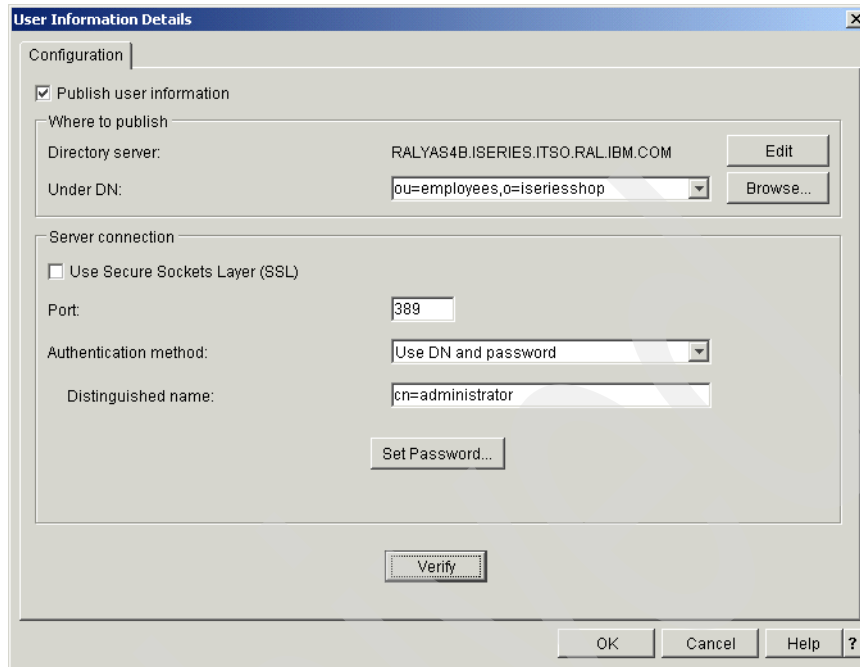


Figure 4-20 RALYAS4B Properties User Information Details

6. Verify that **Publish user information** is checked and enter the directory path in the Under DN field where the System Distribution Directory will be published.

In our scenario, this is ou=employees,o=iseriesshop.

For the moment we do not use SSL, we use the Distinguished Name (DN) and password for authentication.

7. Enter the DN administrator Distinguished name field.

In this scenario we use cn=administrator. This is the DN of the administrator as defined in Figure 4-5 on page 67.

8. Click **Set Password** to set the administrator password.

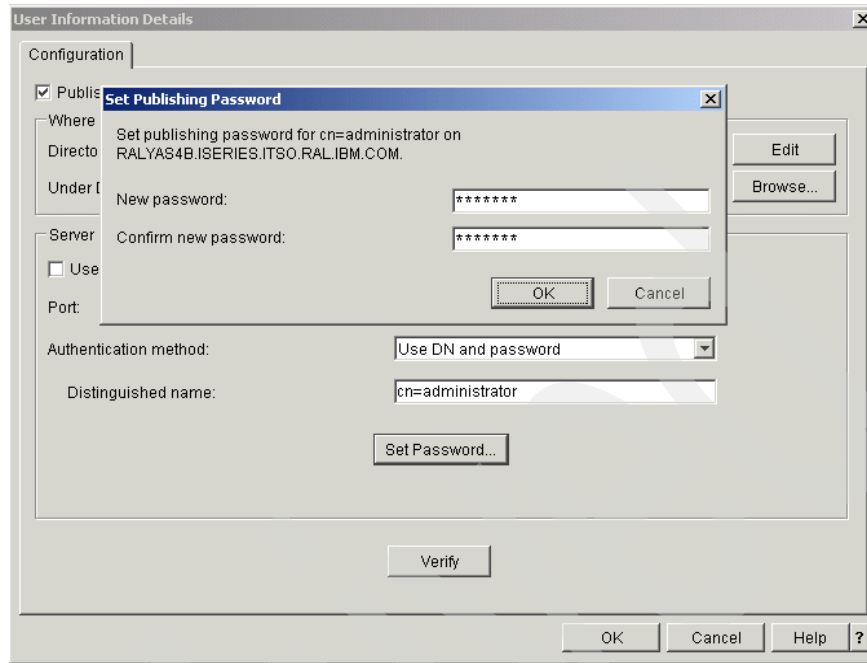


Figure 4-21 RALYAS4B properties Set Publishing Password

9. Enter the password for the DN you specified on the Distinguished name parameter and click **OK** to set the password. This password has to match the password you specified for this DN on the LDAP server.

**Important:** In case you configure SDD publishing so that the SDD gets published to an LDAP directory on another iSeries server, you may need to change the password on the LDAP server first. The case where you need to change the password is when this LDAP server was configured using the configuration wizard with default values. Then a random password gets generated, which cannot be retrieved and therefore needs to be changed to a password you know.

10. Click **Verify** to check the settings in the User Information Details window.

This will attempt the connection to the Directory server by using the distinguished name (DN) name, password, connection method, and port.

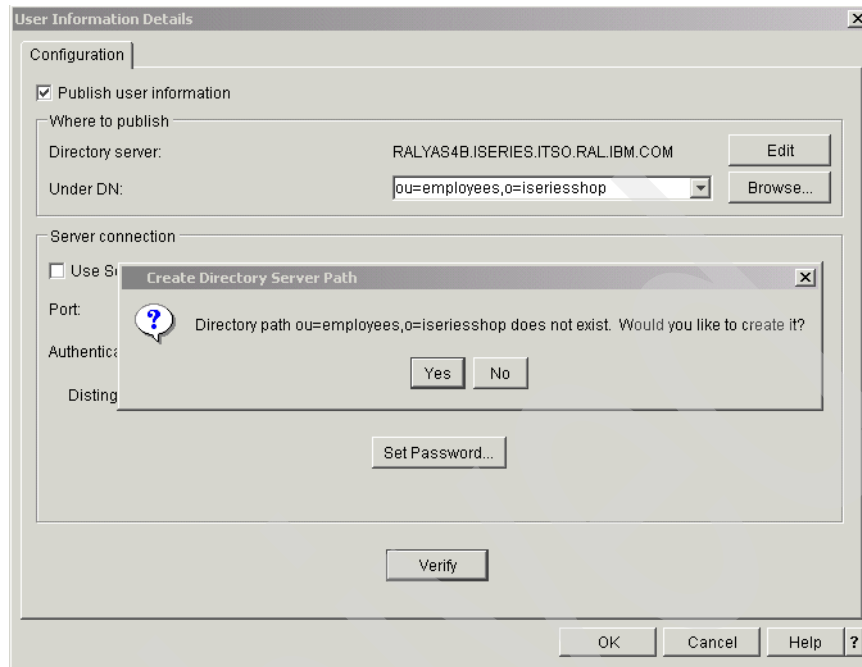


Figure 4-22 RALYAS4B Properties create Directory server Path

11. If the directory path does not exist in the LDAP directory, a message is shown asking whether you want to create the specified directory path. Click **Yes** to create the directory path.

**Tip:** Beginning in Version 5 Release 1, if the parent DN to which the data is being published does not exist, Directory Services will automatically create it.

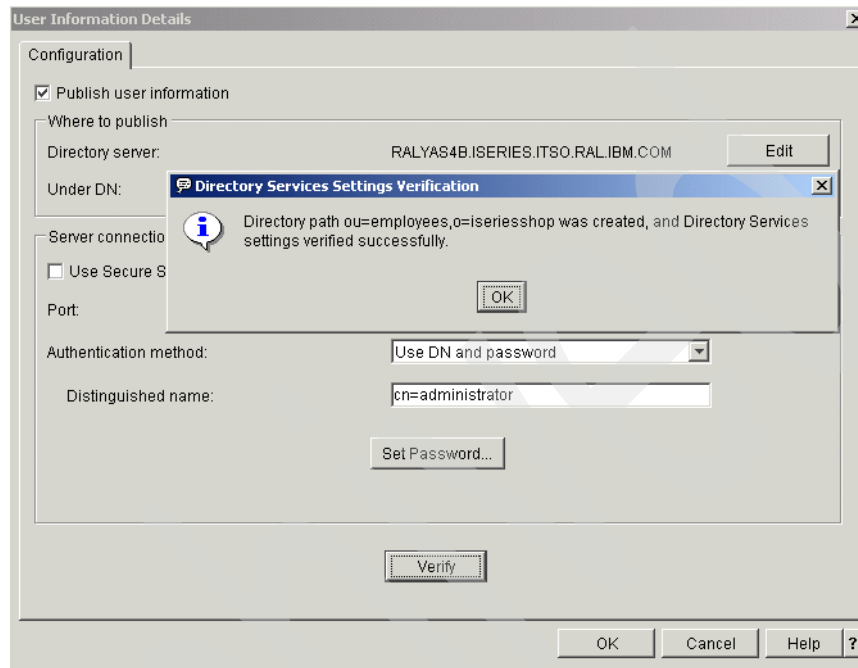


Figure 4-23 RALYAS4B properties - Directory Services Settings Verification

12. Click **OK** when receiving no errors, otherwise correct any errors and re-verify.

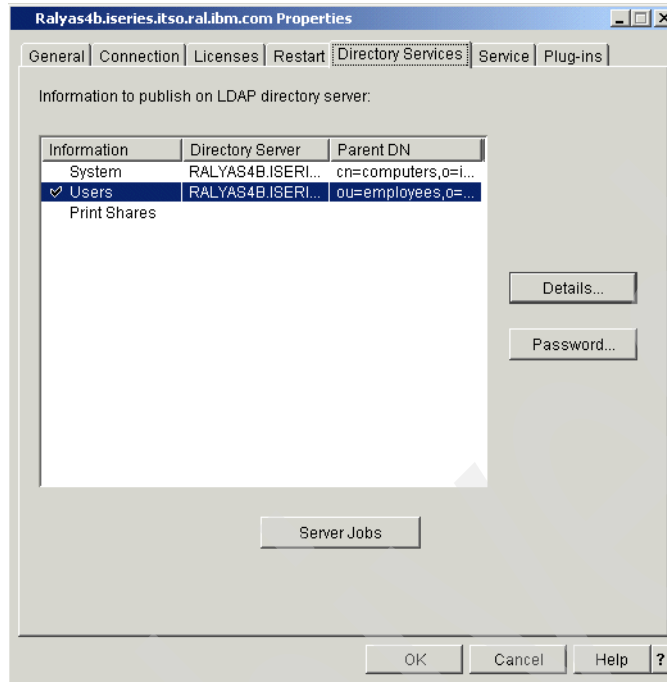


Figure 4-24 RALYAS4B Directory Services for SDD publishing

13. Click **OK**. This will start the publishing of the SDD.

You can check your System Distribution Directory publishing by using various methods.

A simple way is using a Netscape browser session by entering the following URL:

`ldap://RALYAS4B/o=iseriesshop??sub`

This URL shows the complete directory path for `o=iseriesshop`.

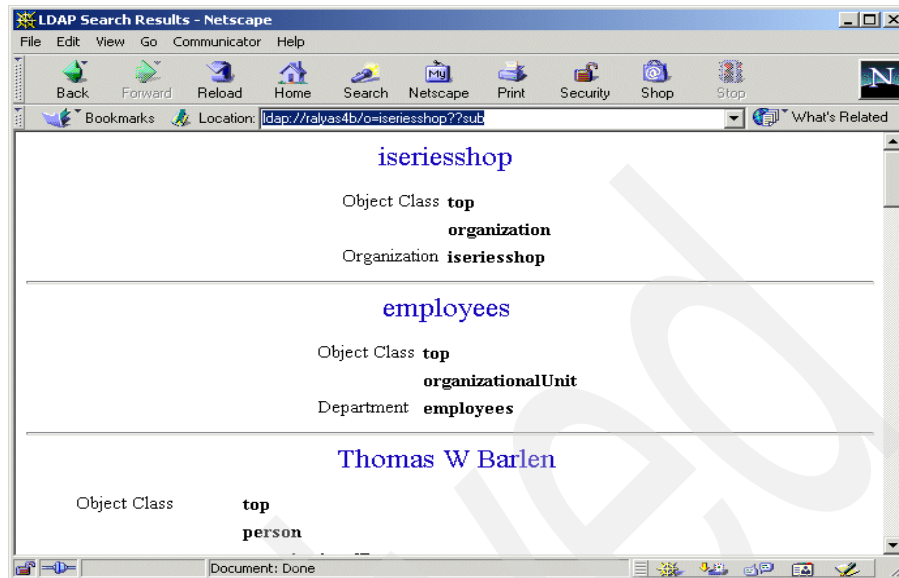


Figure 4-25 Displaying the directory path using a browser

As an alternative you can use the IBM SecureWay Directory Management Tool. Assuming that you are still connected with the DMT, click **Refresh Tree**.

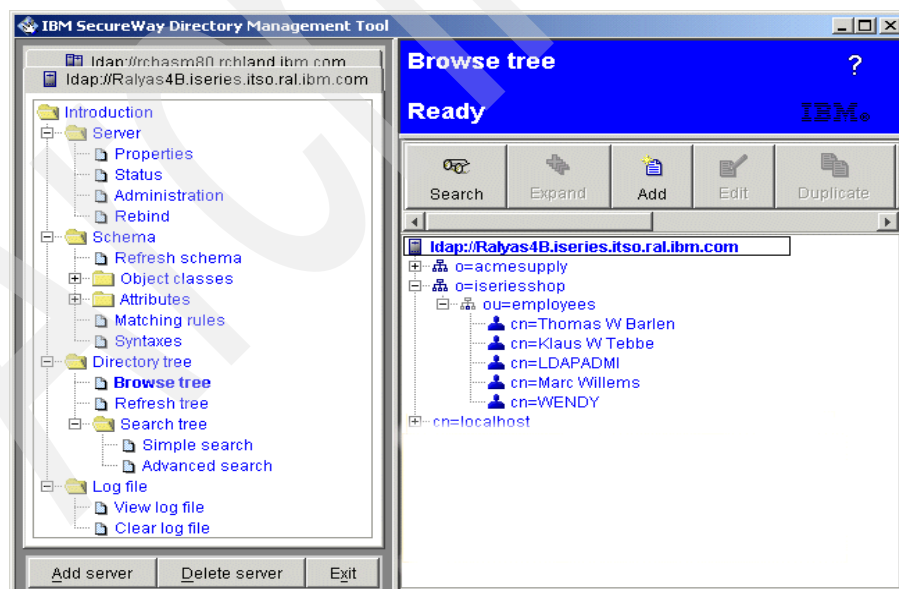


Figure 4-26 Displaying the directory path using the DMT

You can also verify whether the publishing process is working by checking the joblog of the publishing client QGLDPUBA as shown in Figure 4-27.

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help

Display All Messages

Job . . . : QGLDPUBA      User . . . : QDIRSRV      System: AS4B
Number . . . : 090926

>> QSYS/CALL QSYS/QGLDPUBA
Ownership of object QGLDSDDU in QUSRSYS type *USRIDX changed.
Synchronization between the System Distribution Directory and the LDAP
Directory Server completed.
Synchronization between the System Distribution Directory and the LDAP
Directory Server completed.
Synchronization between the System Distribution Directory and the LDAP
Directory Server completed.
Synchronization between the System Distribution Directory and the LDAP
Directory Server completed.
Synchronization between the System Distribution Directory and the LDAP
Directory Server completed.
Synchronization between the System Distribution Directory and the LDAP
Directory Server completed.
- Synchronization between the System Distribution Directory and the LDAP
Directory Server completed.

Press Enter to continue.

F3=Exit  F5=Refresh  F12=Cancel  F17=Top  F18=Bottom

MA  b  MW

```

Figure 4-27 Directory Server iSeries QGLDPUBA joblog

Extending the details of one of the messages also shows how many entries were published to the directory. In the example shown in Figure 4-28, the publishing client published five entries from the SDD to the LDAP directory.

```

Session B - [24 x 80]
File Edit View Communication Actions Window Help

Additional Message Information

-
Message ID . . . . . : GLD0305      Severity . . . . . : 00
Message type . . . . . : Completion
Date sent . . . . . : 02/08/02      Time sent . . . . . : 17:56:11

Message . . . . . : Synchronization between the System Distribution Directory
and the LDAP Directory Server completed.
Cause . . . . . : Synchronization between the System Distribution Directory
and the LDAP Directory Server has completed. 5 System Distribution Directory
updates were successfully exported to the LDAP Directory server. 0 System
Distribution Directory updates were omitted. Check previous messages in the
joblog for more information on the omitted updates.

Press Enter to continue.

F3=Exit  F6=Print  F9=Display message details  F12=Cancel
F21=Select assistance level

MA  b  MW

```

Figure 4-28 Directory Server iSeries QPLDPUBA joblog details



**Attention:** Prior to Version 4 Release 5, there was an API you needed to call to initiate the publishing of the SDD. This is not required anymore and should be avoided. The following program call is using this API and should only be used as directed by IBM service personnel:

```
CALL PGM(QDIRSRV/QGLDSSDD) PARM(*ALL 'cn=administrator' 'password' 0 0 0)
```

#### 4.5.4 Excluding entries from being published

The Synchronize System Distribution Directory to LDAP publishes System Distribution Directory entries to an LDAP directory and keeps the LDAP directory synchronized with changes made in the System Distribution Directory.

The following users from the System Distribution Directory are published:

- ▶ Local users
- ▶ Remote users that have been added to the local system and have a Simple Mail Transfer Protocol (SMTP) address

The System Distribution Directory users that are not published are:

- ▶ Some entries are automatically prevented from being published to LDAP. They are the \*ANY System Distribution Directory entries and some other entries that are IBM-supplied starting with Q (QSECOFR, QDOC, QSYS, QDFTOWN, and QUSER, for example).
- ▶ Remote users that do not have a SMTP address
- ▶ Shadowed users

In some cases you may want to prevent additional users from being published to the LDAP directory. For example, some system or software product users that have to be in the SDD, but are not representing real people.

A specific user can be prevented from being published to LDAP by doing the following:

1. Add the user-defined field QREPL QLDAP to the System Distribution Directory. This only needs to be done once per system.  

```
CHGSYSDIRA USRDFNFLD((QREPL QLDAP *ADD *DATA 4))
```
2. Specify \*N0 as the value for the QREPL QLDAP user-defined field for those users that you do not want to replicate to LDAP. Any other value or absence of the QREPL QLDAP user-defined field will replicate the user. It is recommended that you either leave the QREPL QLDAP value blank or specify \*YES if you want the user to be replicated.

For example, using Work with Directory Entries (WRKDIRE), option 1 to add a user or option 2 to change a user, press the F20 key to specify user-defined fields. When using the **ADDIRE** or **CHGDIRE** commands, specify `USRDFNFLD((QREPL QLDAP *NO))` to prevent the user from being replicated.

3. If the user is already replicated to LDAP, and \*NO is specified in the QREPL QLDAP user-defined field, then the user will be deleted from the LDAP directory. Likewise, if the value of the QREPL QLDAP user-defined field is changed to anything but \*NO, then the user will be added to the LDAP directory.

**Attention:** Note that if changes are made in the LDAP directory, these changes are not synchronized back to the System Distribution Directory.

Additional information for publishing users is located at the following URL:

<http://www.iseries.ibm.com/ldap/ldapfaq.htm>

## 4.6 Publishing printer information

In this section we will use the Windows 2000 Server Active Directory for publishing shared iSeries printers. We will only show the configuration windows needed to add the necessary configuration. The section does not cover the installation and configuration of Microsoft Active Directory. For more information about MS Active Directory refer to *Using LDAP for Directory Integration: A Look at IBM SecureWay Directory, Active Directory, and Domino*, SG24-6163, or the appropriate Microsoft documentation.

To manage the Active Directory server we used the ADSI Edit Microsoft Management Console (MMC) snap-in. See the redbook *Using LDAP for Directory Integration: A Look at IBM SecureWay Directory, Active Directory, and Domino*, SG24-6163, or the Windows 2000 documentation how to set up this snap-in.

The following list summarizes the tasks to set up and publish iSeries printer information:

- ▶ Adding a new context for iSeries printers to Active Directory (AD)
- ▶ Configuring Directory Services to publish print shares
- ▶ Creating additional print shares for new printers
- ▶ Verifying that print shares have been published to AD

The documented steps assume that the Active Directory is already set up and active.

## Adding a new context for iSeries printers to Active Directory

Perform the following steps to add a new context in AD:

1. Start the Microsoft Management Console (MMC) and expand ADSI Edit.

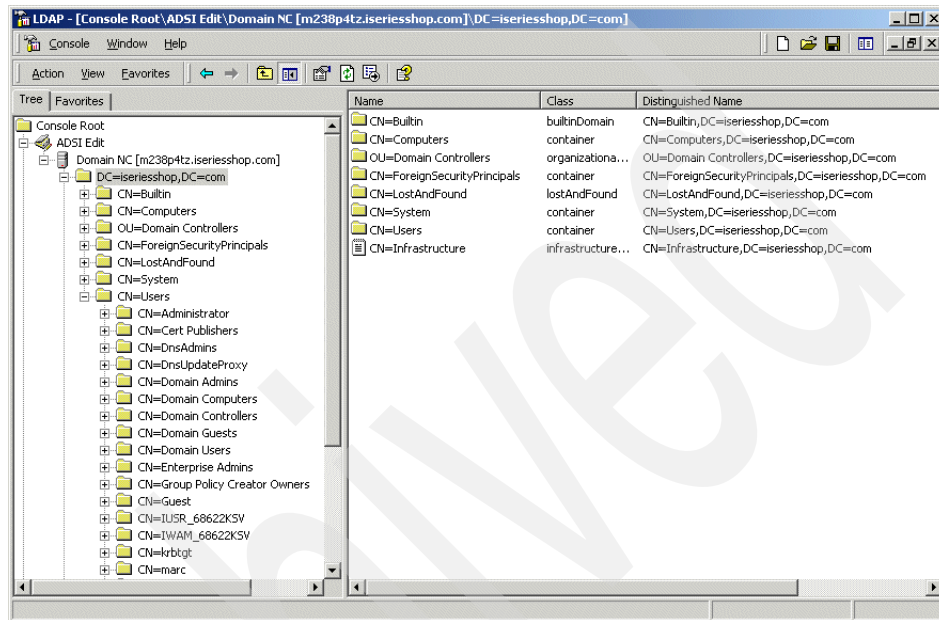


Figure 4-29 Windows 2000 Active Directory console root

2. Select and right-click the parent context under which you want to create the new child context. In this example it is DC=iseriesshop,DC=com.
3. From the context menu select **New -> Object**.

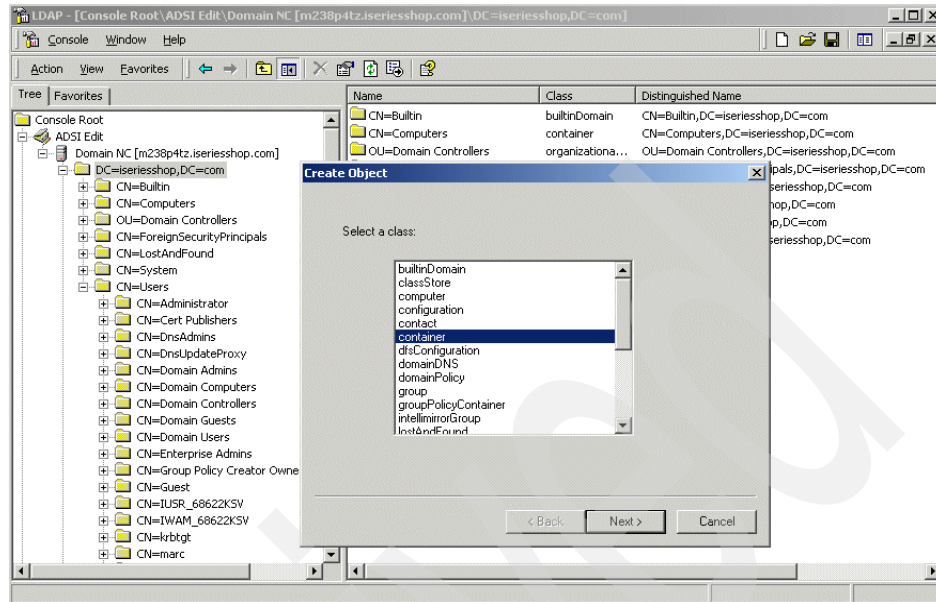


Figure 4-30 Windows 2000 Active Directory console new container

Select the object class for the new context. In this scenario we selected the object class container, which serves as a basic context with no special attributes.

4. Click **Next**.

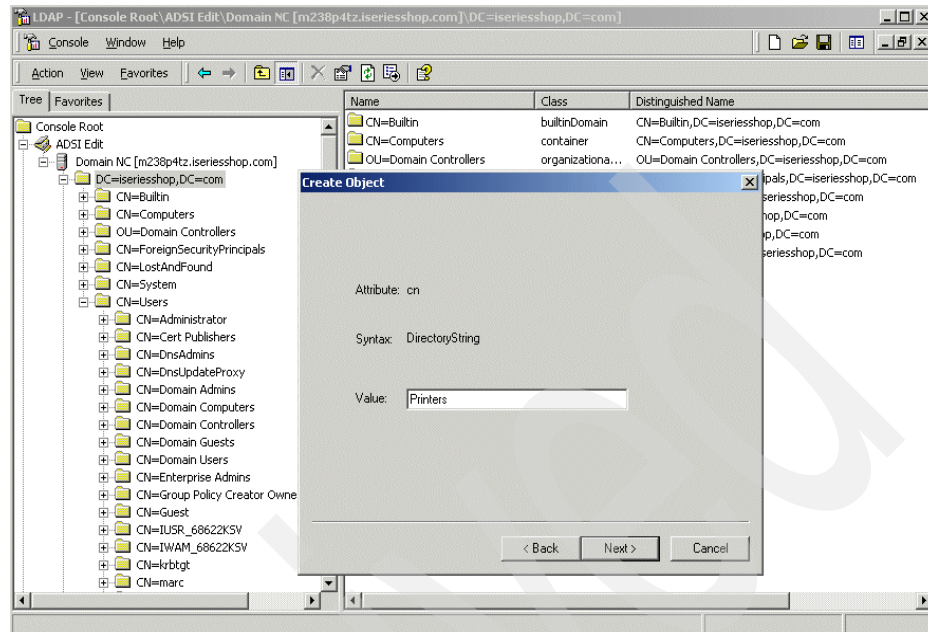


Figure 4-31 Windows 2000 LDAP console new container printer

5. Enter the common name (cn) for the new context of the class container. In this example, Printers.
6. Click **Next**. A new window is shown allowing you to enter values for attributes of the new entry.
7. As the container class has only the cn attribute, we can skip this step and click **Finish**. The new entry (context) Printers is added to the directory, as shown in Figure 4-32 on page 96.

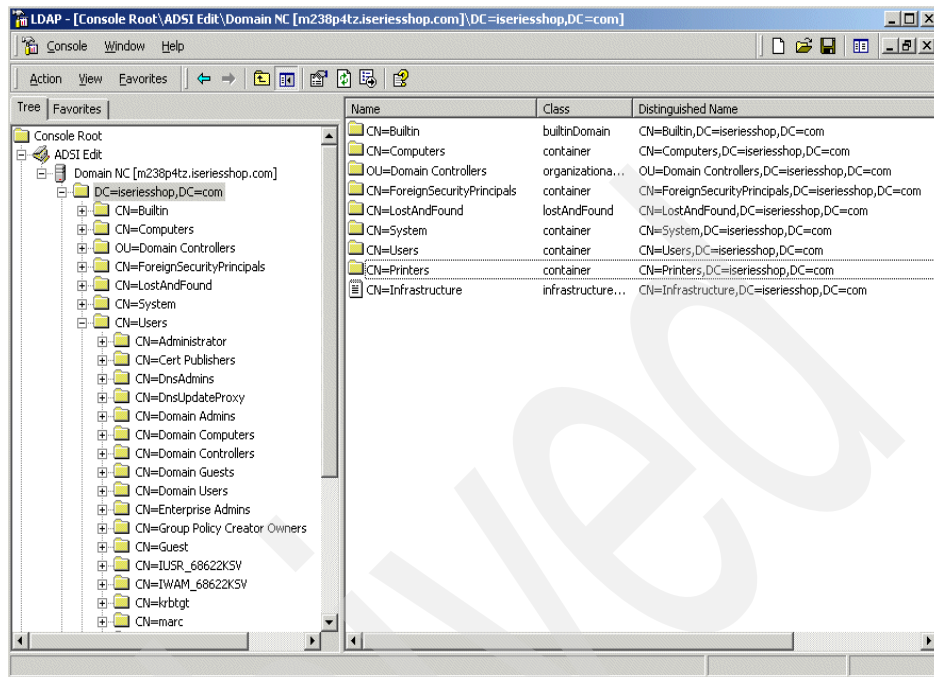


Figure 4-32 Windows 2000 Active Directory console with new entry cn=Printers

## Configuring Directory Services to publish print shares

Now we configure system RALYAS4B for publishing the print shares to the new container. To do this:

1. Launch the Operations Navigator.
2. Select and right-click **System (RALYAS4B)**.
3. Select **Properties**.
4. Click the **Directory Services** tab in the Systems Properties window.

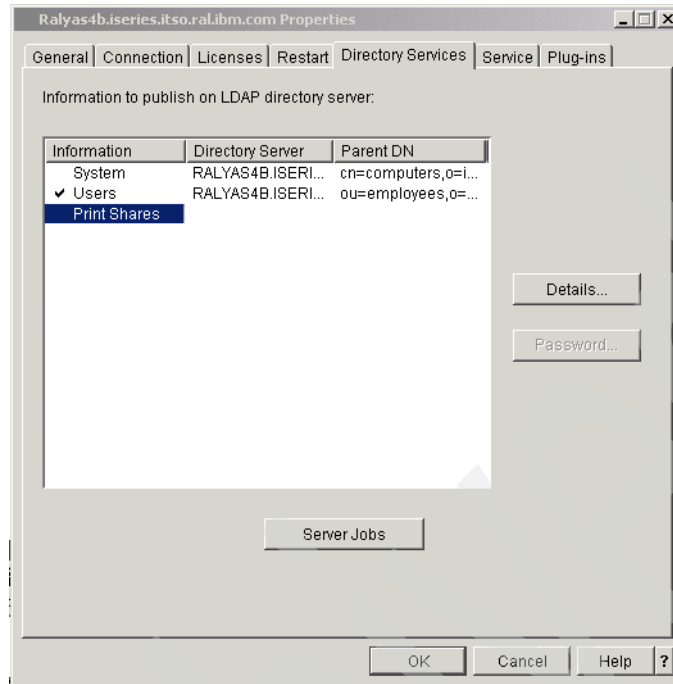


Figure 4-33 RALYAS4B Properties Directory Services Print Shares

5. Select **Print Shares** and click **Details**. This will open the Printer Share Information Details window.

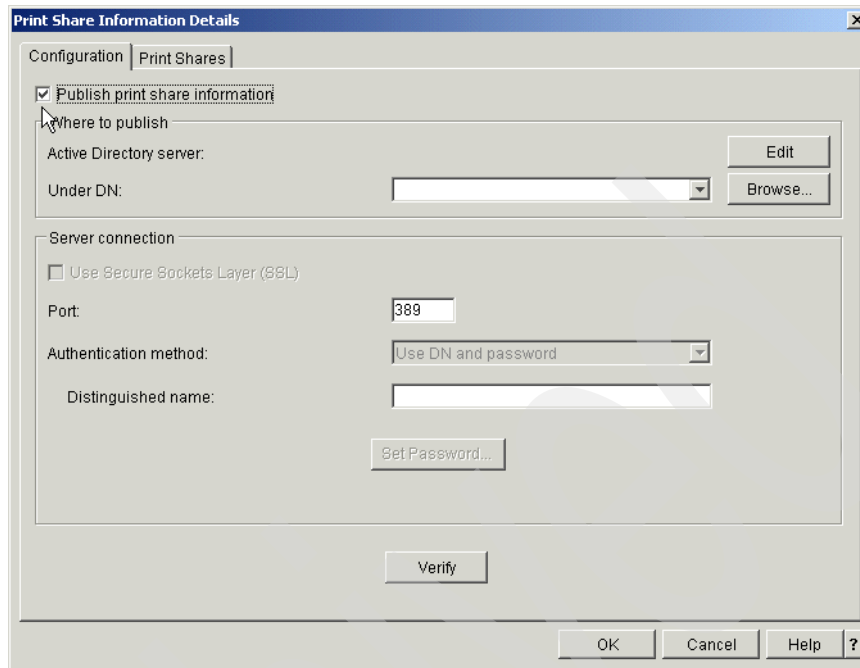


Figure 4-34 RALYAS4B Print Share Information Details window

6. Check **Publish print share information**.
7. Click **Edit** to enter the host name or the IP address of the Active Directory server.

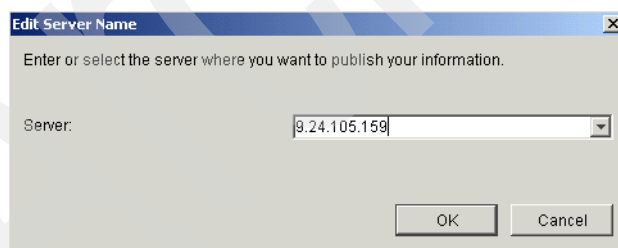


Figure 4-35 RALYAS4B Print Shares Edit Server Name window

8. Enter the Active Directory TCP/IP name or the TCP/IP address.
9. Click **OK**.



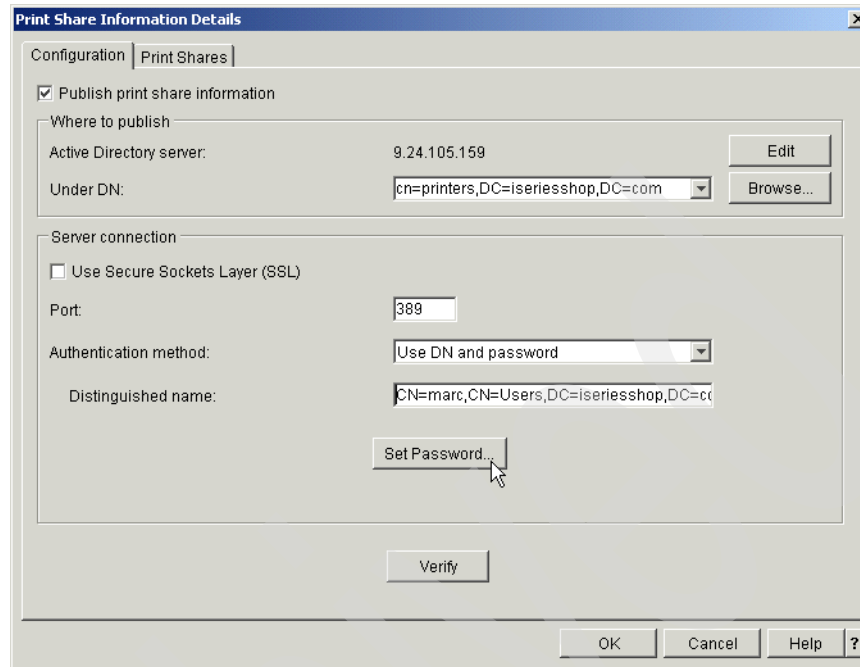


Figure 4-36 RALYAS4B Properties - Print Share Information Details window

10. Enter the distinguished name under which the iSeries printers will be published.

In this example it is the fully qualified DN of the context that was created in “Configuring Directory Services to publish print shares” on page 96, `cn=printers,DC=iseriesshop,DC=com`.

**Tip:** If the Active Directory is running you can select the correct DN in the pull down menu in the Under DN field.

11. Enter the fully distinguished name of the Active Directory administrator in the Distinguished name field. In this example:

`CN=marc,CN=Users,DC=iseriesshop,DC=com`

This DN does not need to be the Windows administrator, but it has to be a DN of a user that has the authority to manage the directory path.

12. Click **Set Password** and enter the password of the user specified in the Distinguished name field.
13. Click **OK** to save the password and return to the Print Shares Information Details window.

14. Click **Verify** to check the connection to the Active Directory server using the distinguished name (DN) name, password, connection method, and port.

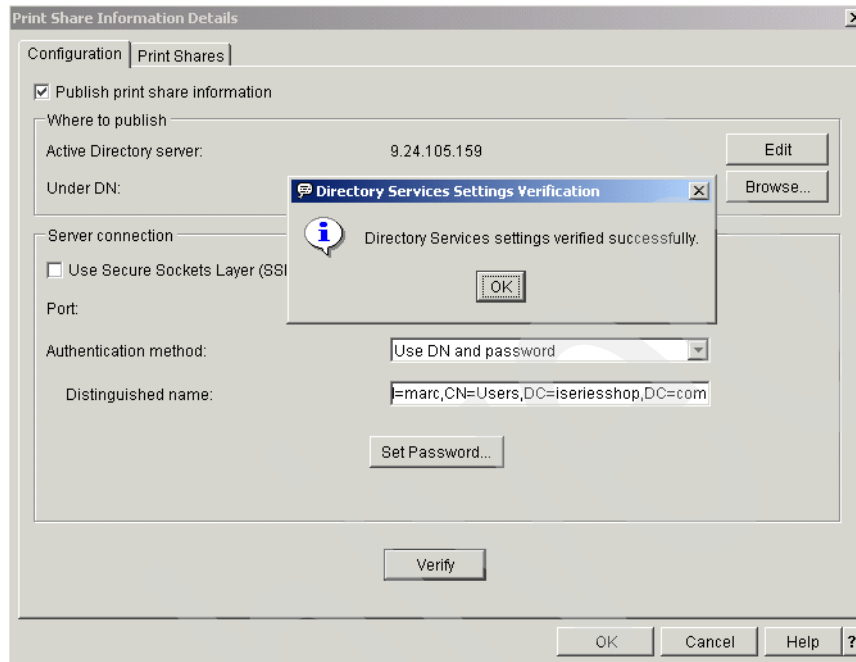


Figure 4-37 RALYAS4B Properties - Directory Services Settings Verification window

15. Click **OK** to close the message window.
16. Click the **Print Shares** tab. This opens the window with the available print shares, as shown in Figure 4-38 on page 101.

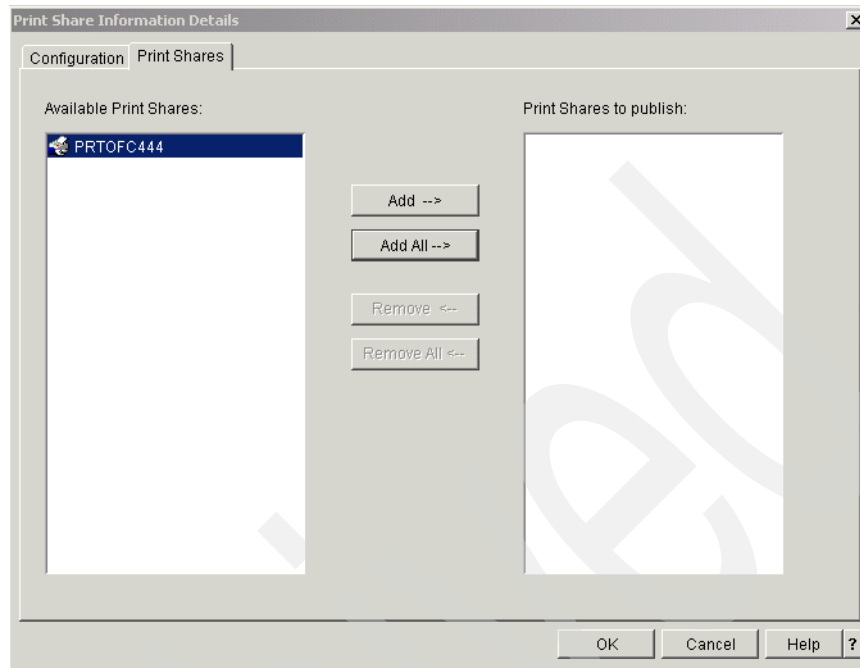


Figure 4-38 RALYAS4B Properties - Directory Services Available Print Shares

In the left pane of the window you can find all the iSeries printers that have been shared, but not published yet.

**Tip:** If the Available Print Shares pane in Figure 4-38 is empty you can still continue with the configuration, but you will receive the following error:



See “Publish new printers to the Active Directory” on page 103 for information on how to publish additional printers to the Active Directory server.

17. Select the printer you want to publish and click **Add** or click **Add All** if you want to publish all the available print shares.

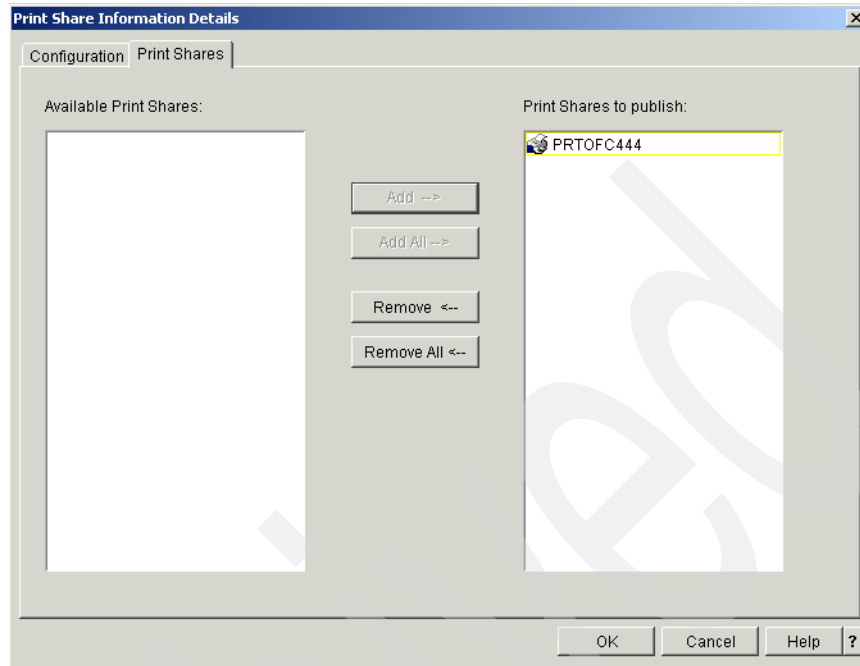


Figure 4-39 RALYAS4B Properties - Directory Services Print Shares to publish

18. The selected and added printers move to the right pane. Click **OK** to save your selections.

19. Click **OK** in the System Properties window. This will start the print share publishing.

If the Operations Navigator do not have the latest PTFs installed you may receive the error shown in Figure 4-40.



Figure 4-40 Date Entry Error window

Install the latest PTFs for Client Access and Operations Navigator and restart the configuration. In our case, we used one PC with an old PTF level, upgrading to PTF level SI01037.

## Publish new printers to the Active Directory

To publish new printers to the Active Directory you have to perform following steps:

1. Launch the Operations Navigator.
2. Expand the **System (RALYAS4B)**.
3. Expand **Basic Operations**.
4. Expand **Printers**.
5. Select and right-click the printer that you want to publish. In this example, Prttest.

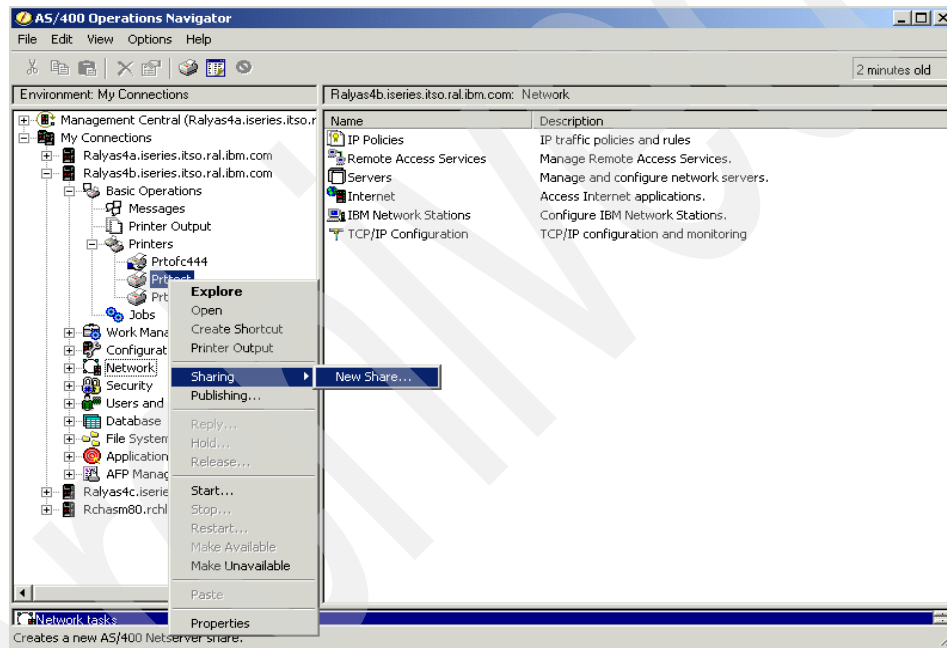


Figure 4-41 RALYAS4B new printer share

6. Select **Sharing** and then **New Share...**

If the printer you want to publish is already shared you can create a new share or use the existing share by selecting the printer share name.

This opens the AS/400 NetServer Print Share window as shown in Figure 4-42 on page 104.

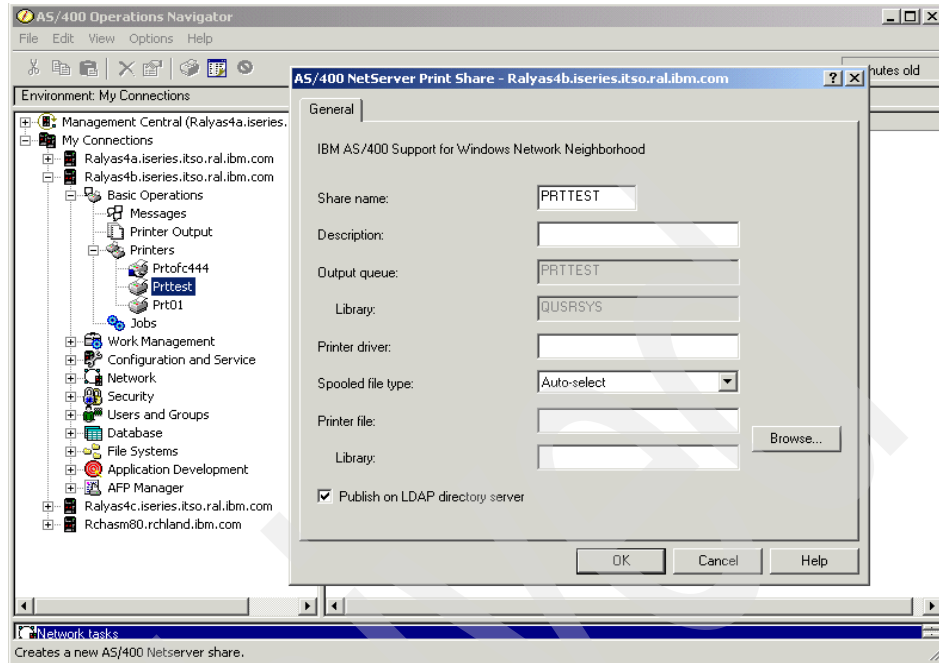


Figure 4-42 RALYAS4B - AS/400 NetServer Print Share

7. Check **Publish on LDAP directory server** at the bottom-left corner of the window.
8. Click **OK** to save the settings. The new print share will be published to the Active Directory server.

## Verifying that print shares have been published to AD

The following methods can be used to verify whether print share information has been successfully published.

- Check the iSeries job QGLDPUBE joblog for a message, as shown in Figure 4-43 on page 105.

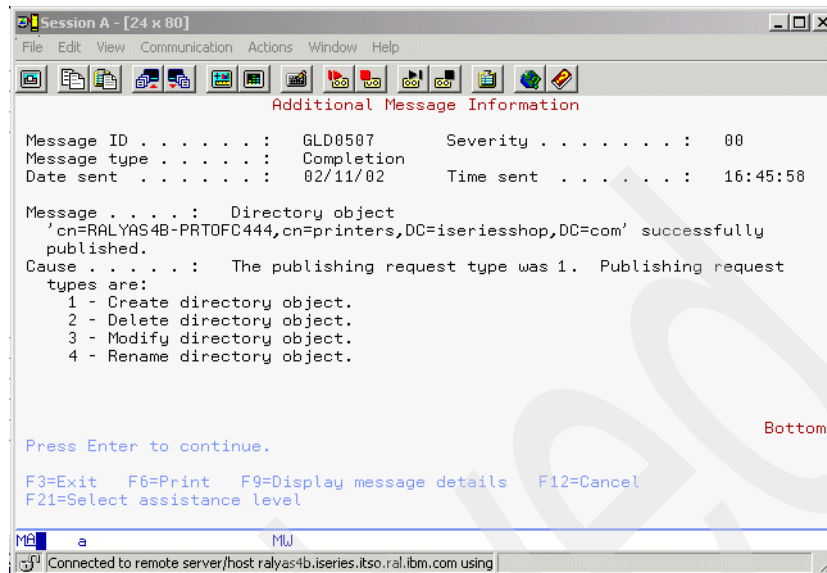


Figure 4-43 QGLDPUBE joglog

- Check the Windows 2000 LDAP console for these updates.

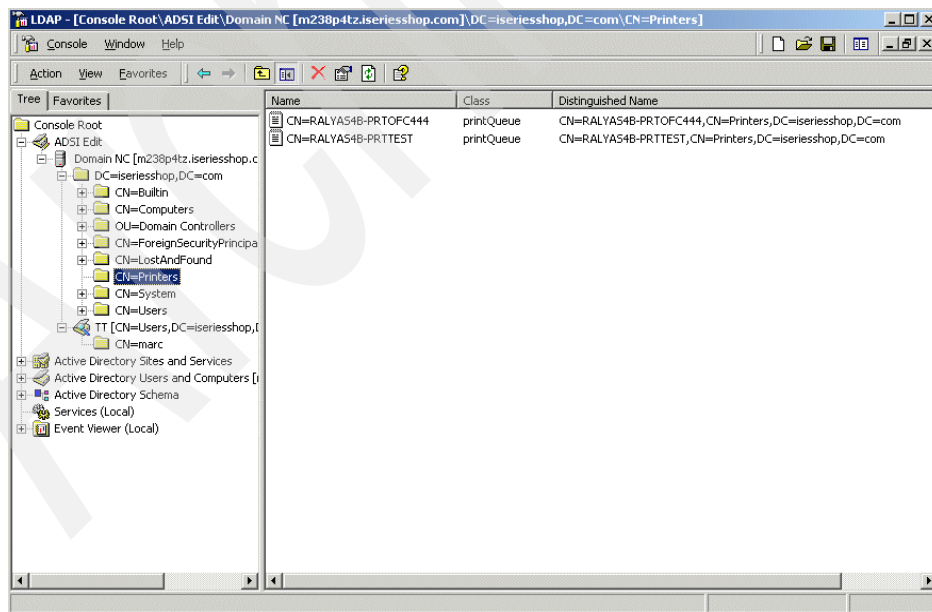


Figure 4-44 Windows 2000 LDAP console with new printer updates

## 4.7 Setting up directory replication

You can set up replicas of the LDAP directory server to directory servers on other iSeries 400 systems. Directory Services uses the standard LDAP Version 3 protocol to replicate.

The information stored on replica LDAP directory servers is identical to the information on your main, or "master," LDAP directory server.

There are two principal benefits to having one or more replicas of your LDAP directory:

- ▶ Replicas make directory searches faster. Instead of having all clients direct search requests to a single master server, you can split requests between the master server and the replica servers.
- ▶ Replicas provide a backup to the master server. If the master server is unavailable, a replica can still fulfill search requests and provide access to directory data.

Replica servers are read-only. When an authorized user attempts to change an entry on a replica server, it refers the request to the master directory server.

**Note:** You cannot replicate between LDAP Version 3 and LDAP Version 2 servers. Therefore, the system that you replicate to must be using the same version of LDAP as the system from which you replicate. Version 4 Release 3 and Version 4 Release 4 of OS/400 support LDAP Version 2. Version 4 Release 5 and later releases support LDAP Version 3.

You can replicate the Directory Services directory to IBM SecureWay Version 3.2 servers on other platforms.

### 4.7.1 Stage 2 set-up scenario

We will now set up replications as described in "Stage 2 - The evolution" on page 41 and shown in Figure 4-45 on page 107.



## Replication scenario

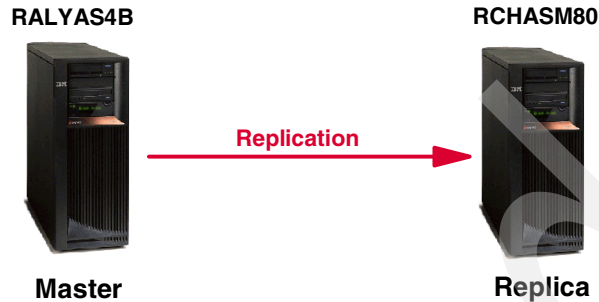


Figure 4-45 Replication scenario

You have to perform the following tasks in this scenario to set up a replica where system RCHASM80 is the replica of system RALYAS4B.

1. Configure the replica server with the Directory server wizard.
2. Set up the LDAP replica server to receive updates from the LDAP master server.
3. Move LDAP directory data from the master LDAP server to the LDAP replica server.
4. Set up the LDAP master server to have a new LDAP replica.

### Setting up the new replica server

Follow these steps to set up a new replica of the directory server:

1. Configure the replica on the server RCHASM80 using the wizard as described in Section 4.3.1 “First-time configuration” on page 64.

**Important:** Both the master and the replica servers must have the same schema and suffixes defined. If, for example, you have the suffixes `o=iseriesshop` and `o=acmesupply` configured on the master, but only `o=iseriesshop` configured on the replica, the replication will fail. Remember, the replica is identical to the master.

Once the basic configuration is complete, proceed with the remaining configuration steps.

2. In Operations Navigator under **System -> Network -> Servers** click **TCP/IP**.

If the server is not already stopped, stop it now. Refresh the status of the server until the status is Stopped.

3. Right-click **Directory** and select **Properties**.
4. Click the **Database/Suffixes** tab.

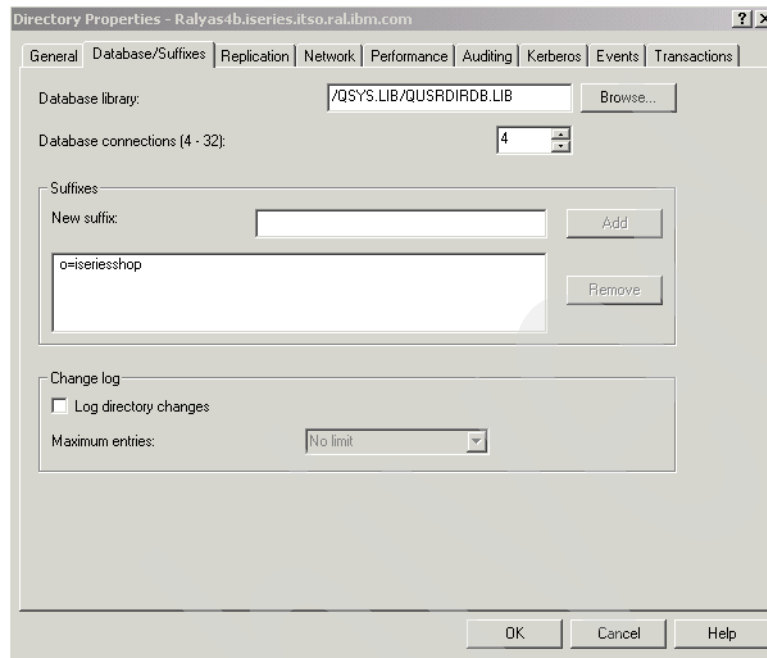


Figure 4-46 New replica server Database/Suffixes tab

If the suffix that you want to replicate is not on the list, add it. We already added the suffix in the wizard. Note that at this stage of the scenario, we have not added the `o=acmesupply` suffix yet. This will be done in stage 3 of the scenario. You then need to change the replication settings accordingly.

5. Click the **Replication** tab. Operations Navigator may prompt you to enter connection information. Enter the directory administrator DN and password for the LDAP server on RCHASM80 and click **OK**.
6. Select **Use as a replica server**.

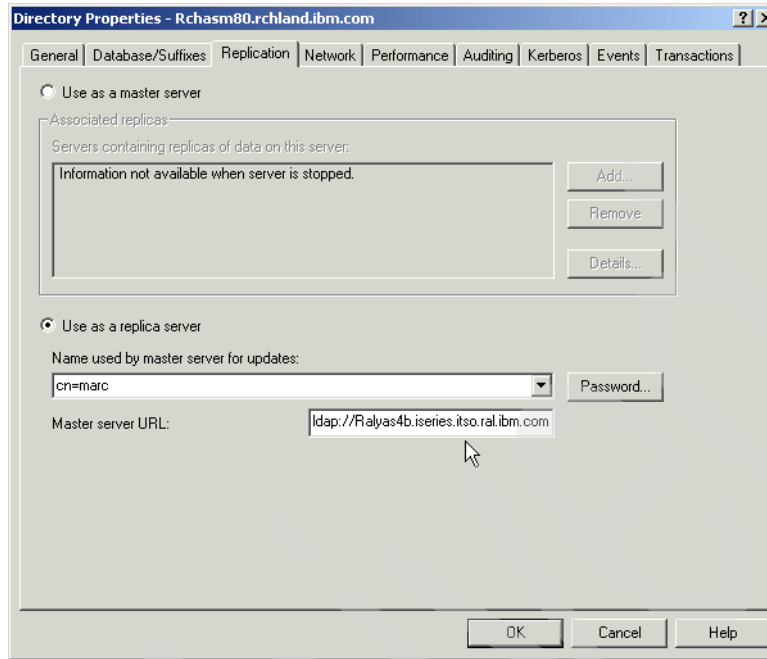


Figure 4-47 New replica server Replication tab

7. In the Name used by master server for updates field, select a name for the master server to use when it logs onto the replica server when it performs updates. This may be a distinguished name (DN) or a Kerberos user.

If you select a DN:

- Click the **Password** button next to the Name used by master server for updates field. Enter a password for the master server to use when it logs onto the replica server to perform updates.

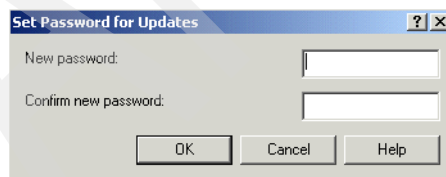


Figure 4-48 Set Password for Updates window

**Note:** You should make note of this password and the name you entered. You will need them when you set up the master server for replication. The name and password are not related to entries within the directory or an OS/400 user profile. If you set up multiple replicas, these values can be different.

If you select **Add Kerberos User:**

- You will be prompted to enter the Kerberos name (in the format LDAP/hostname, where host name is the fully qualified host name of the master server) and the default realm (such as ACME.COM) of the master server.

**Note:** To use Kerberos, you must have Kerberos enabled on both the master and the replica servers.

8. In the Master server URL field, enter the name of the master server in URL format. If your master server uses a port other than the default, enter this port number as part of the URL.
9. If you want to use Secure Sockets Layer (SSL) when replicating, use Digital Certificate Manager to enable SSL for the server. You can start Digital Certificate Manager from the Network tab. For more information on how to enable SSL for directory client and server services, refer to Section 4.9 “Securing LDAP traffic” on page 126.
10. Click the **General** tab.

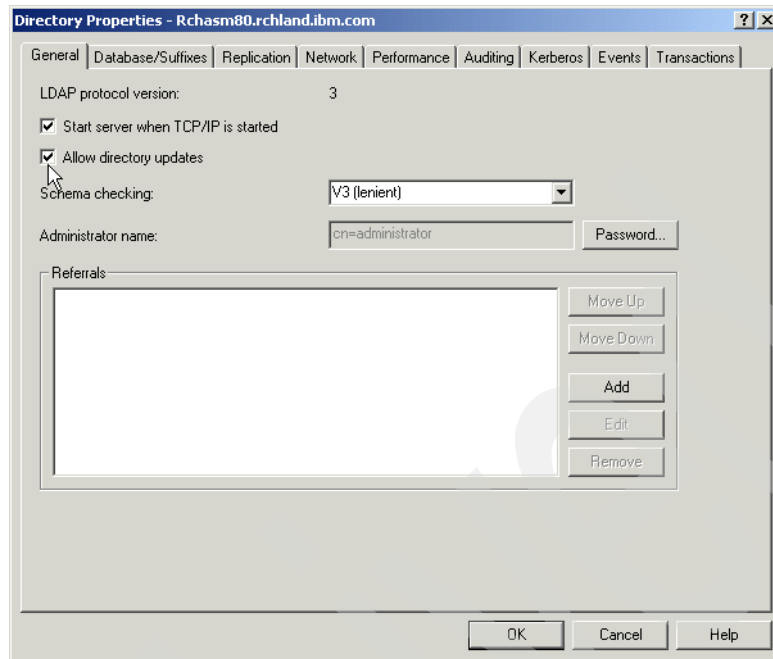


Figure 4-49 Directory Properties - checked Allow directory updates

11. If **Allow directory updates** is unchecked, check it. This allows the data to be imported into the replica directory.
12. Click **OK** to save the replica configuration.

### Moving LDAP directory data from the master to the replica

The LDAP replication will *not* initially load the replica with all the data from the master after the configuration. We have to export the LDAP data from the master and import this data on the replica.

You can skip this step if you do not have any initial data that you want to transfer to the replica server from the master server.

### Set up LDAP data for initial replication on the LDAP master

Before exporting the directory data you have to make sure that the master server does not accept any updates while you are in the process of exporting, transferring, and importing the data on the replica server. Not doing so will result in an inconsistent replication and may cause unpredictable results.

Prevent the master server from being updated by performing the following steps:

1. Using Operations Navigator on the master server (RALYAS4B), expand the system on which the master directory server runs.
2. Expand **Network** and then **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.

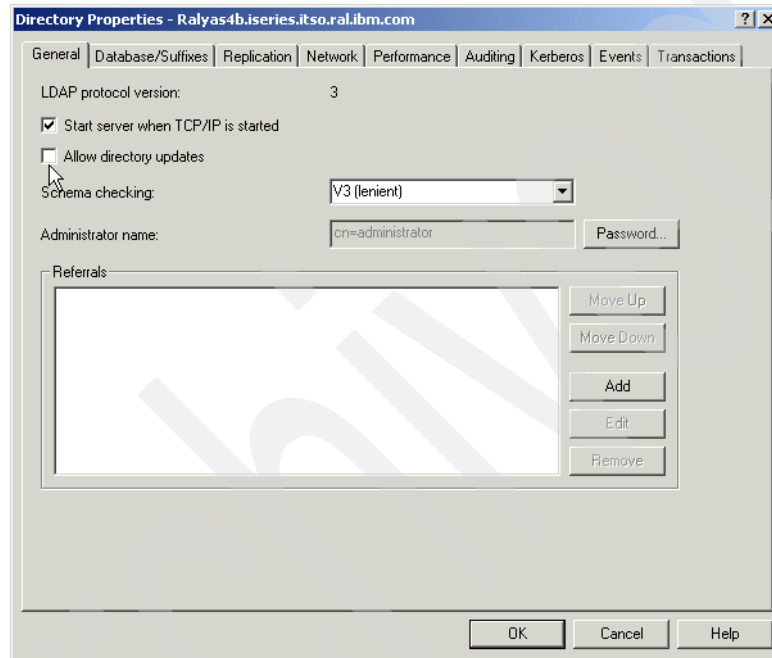


Figure 4-50 Directory Properties - unchecked Allow directory updates

5. Uncheck **Allow directory updates**. This will prevent updates to the directory until the initial export/import is completed.
6. Click **OK**. If the master server was running, you will be asked whether to restart the server now or later. In order to proceed, you need to restart the server now.
7. Right-click **Directory**, select **Tools**, and select **Export File**.

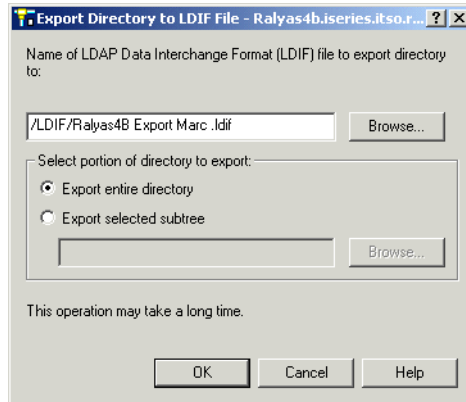


Figure 4-51 Export Directory to LDIF File window

Enter a name and path for the LDIF file, export the complete directory, and click **OK**. An export progress window is displayed.

**Important:** Do not use this option to export only a selected subtree. In this case, the replica would not contain the exact copy of the master server and subsequent updates to entries that do not exist on the replica will fail.

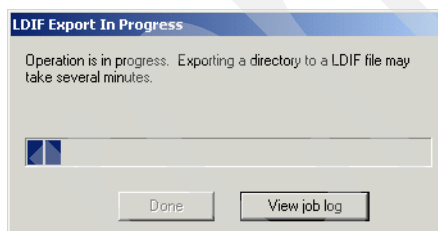


Figure 4-52 LDIF Export In Progress window

8. Click **Done** when finished or **View job log** if errors occur.
9. Transfer the exported file from the master RALYAS4B to the replica RCHASM80. If you use FTP for the file transfer make sure to use ASCII format. The exported file can be anywhere in the IFS of RCHASM80; just remember where you put it.
10. Using Operations Navigator, expand the system on which the replica (RCHASM80) directory server runs.
11. Expand **Network** and then **Servers**. Then click **TCP/IP**.
12. Right-click **Directory**, select **Tools**, and select **Import File**.

**Important:** You can only select Import File if the server is stopped.

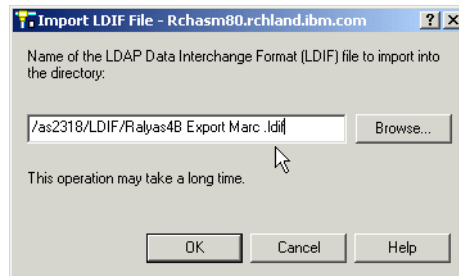


Figure 4-53 Import LDIF File window

13. Select the LDIF file you transferred from the master and click **OK**. A window shows up informing you of the import progress.

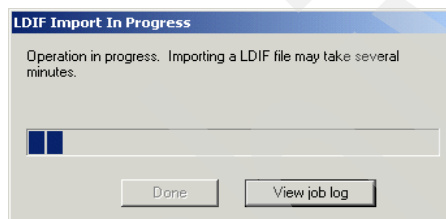


Figure 4-54 LDIF Import In Progress window

14. After the import has completed, a message window is shown with information about how many entries were added or not added. Click **OK** when finished or click **Job log**.

In the Job log you can find information on why items are not added. Most of the time these errors are caused by missing suffixes or a different directory schema. For more information about directory schemas refer to Chapter 1, “Directory concepts” on page 3 or Appendix B, “Extending your directory schema” on page 521.

15. Click **Cancel** to close the LDIF import window.
16. In Operations Navigator start the Directory server on the LDAP replica by right-clicking **Directory** and clicking **Start**.

The new LDAP replica is now ready to receive updates. Now we must tell the LDAP master server to send updates to this LDAP replica.



## Setting up the master server to have a new replica

Follow these steps to set up the master server to have a new replica:

1. Make sure the master server is started.
2. In Operations Navigator, expand the system on which the master directory server (RALYAS4B) runs.
3. Expand **Network** and then **Servers**.
4. Click **TCP/IP**.
5. Right-click **Directory** and select **Properties**.
6. Click the **Replication** tab.

Operations Navigator may prompt you to enter connection information as shown in Figure 4-55.

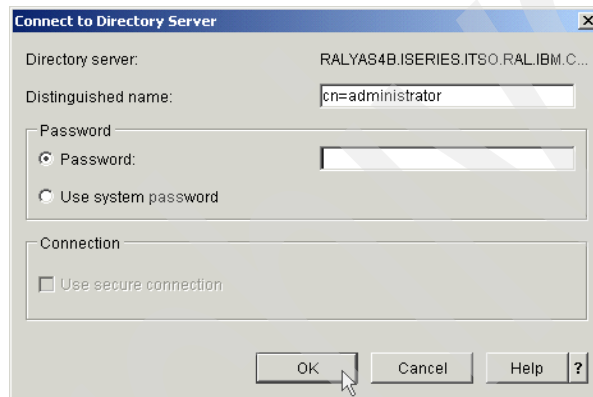


Figure 4-55 RALYAS4B Connect to the Directory Server window

Enter the administrator Distinguished name and password, then click **OK**.

7. Make sure that the Use as a master server box is checked.
8. Click **Add**.

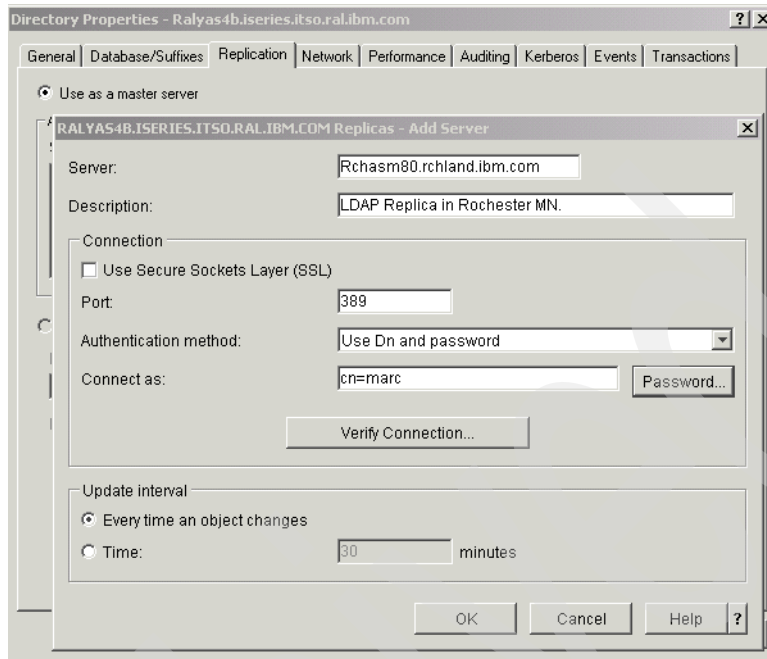


Figure 4-56 Directory Properties - Replication tab

Enter the following information in the Add Server window:

- |                                |  |
|--------------------------------|--|
| <b>Server</b>                  | Enter the fully qualified TCP/IP host name of the replica server. In this case Rchasm80.rchland.ibm.com.   |
| <b>Use Secure Socket Layer</b> | In this example we do not use SSL for the time being. If you want to use SSL to replicate from the master to the replica server, you need to assign a digital certificate to the Directory Services publishing client. |
| <b>Port</b>                    | As we do not use SSL right now, the port remains as its default value 389.   |
| <b>Authentication method</b>   | In this scenario we selected <b>Use Dn and password</b> . Alternatively, you can also use the master server's Kerberos account for authentication.   |
| <b>Connect as</b>              | Enter the name you specified in the replica server as defined in Figure 4-47 on page 109.  |

## Update interval

In this scenario we selected **Every time an object changes** so that the replication is a *realtime* process. If you have a dial-up link between master and replica or you want to transfer the updates as bulk transfers you may consider changing this to **Time** and specifying the interval in minutes.

9. Click **Password** next to the Connect as field and enter the password specified in the replica server as shown in Figure 4-48 on page 109.

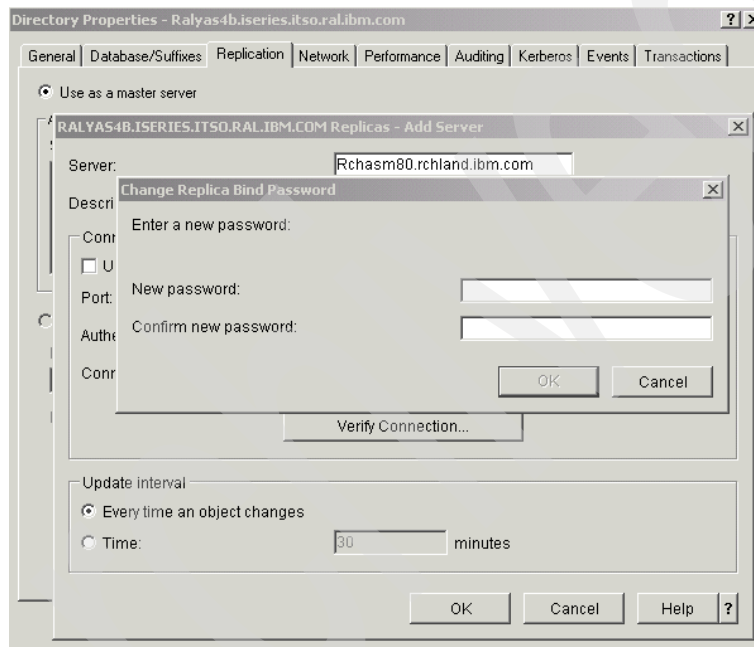


Figure 4-57 Directory Properties - Change Replica Bind Password window

10. After the password is entered click **OK**.
11. Click **Verify Connection....** This will attempt to connect to the replica server by using the distinguished name (DN), password, connection method, and port.

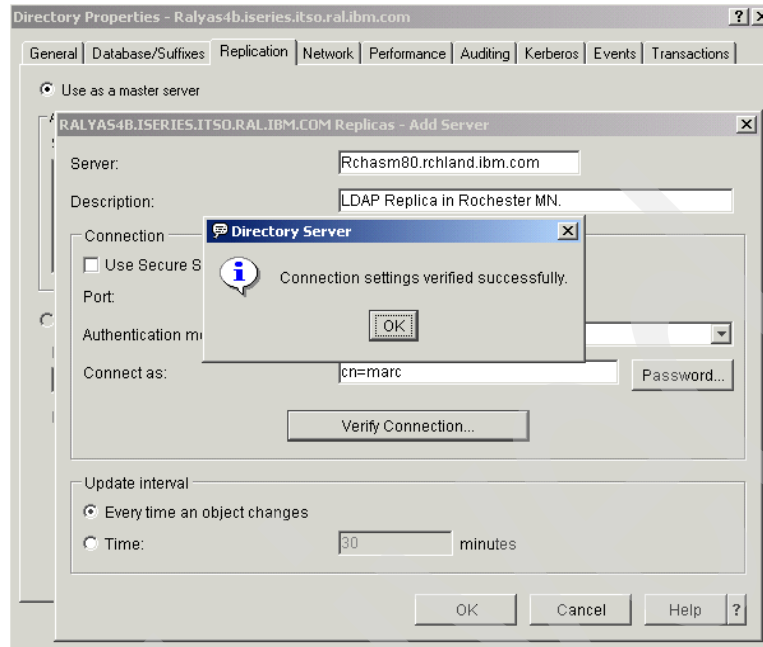


Figure 4-58 Directory Properties - Verify Connection

12. Click **OK** when the connection settings are verified successfully; otherwise correct any errors.
13. Click **OK** in the Add Server window to save the new replica server settings.

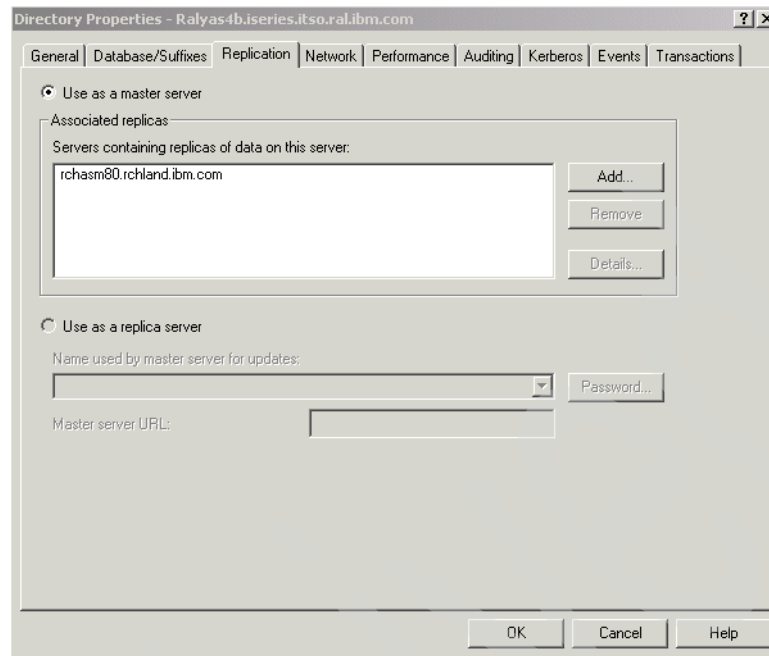


Figure 4-59 *Directory Properties - Replication servers*

The new LDAP replica is shown in the servers containing replicas of data on this server window.

14. Click the **General** tab.

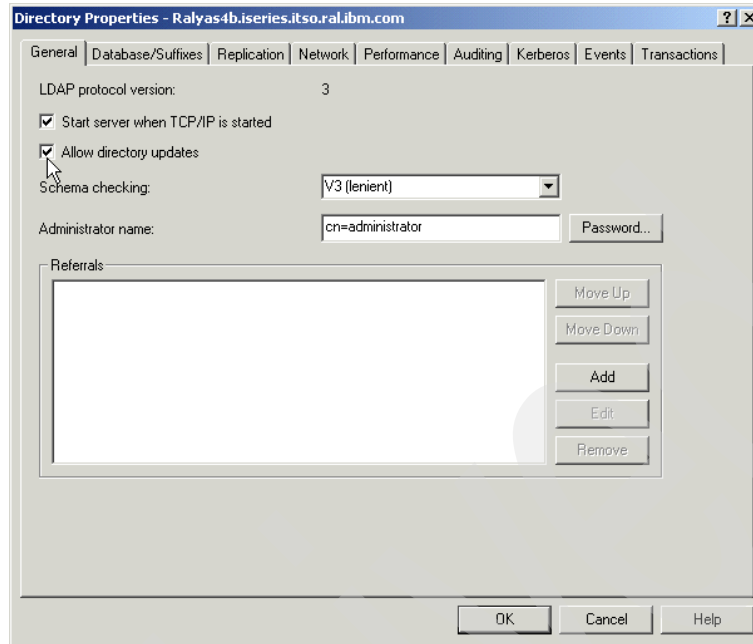


Figure 4-60 Directory Properties - Allow directory updates checked

At this point in the replication set up process, the replica server is started and ready to receive updates. The replica server is also registered with the master server and we are ready to allow updates to the master server again.

15. If **Allow directory updates** is unchecked (it should still be unchecked), check it. This allows the data to be updated in the master directory.

16. Click **OK**.

17. If prompted to restart the server now or later, select to restart the server now.

This completes the LDAP directory set up for the master system (RALYAS4B) and replica system (RCHASM80).

## Testing the LDAP replication

You can test the LDAP replication by performing the following steps:

1. Create a new iSeries user profile on the LDAP master.

In this example we created a user profile on system RALYAS4B with the command:

```
CRTUSRPRF USRPRF(TOREPLICA) STATUS(*DISABLED) TEXT('Test for LDAP
replication')
```

2. Add this user in the System Distribution Directory, with the command:  

```
ADDDIRE USRID(TOREPLIC RALYAS4B) USRD('Test user for Replication.')  
USER(TOREPLICA)
```
3. Check the LDAP master directory for this userid using, for example, the IBM SecureWay Directory Management Tool.

**Tip:** Changes in the System Distribution Directory will be published every five minutes. This is hardcoded and cannot be changed.

On the master system RALYAS4B you can see that the System Distribution Directory published the TORECPLI user to the LDAP directory.

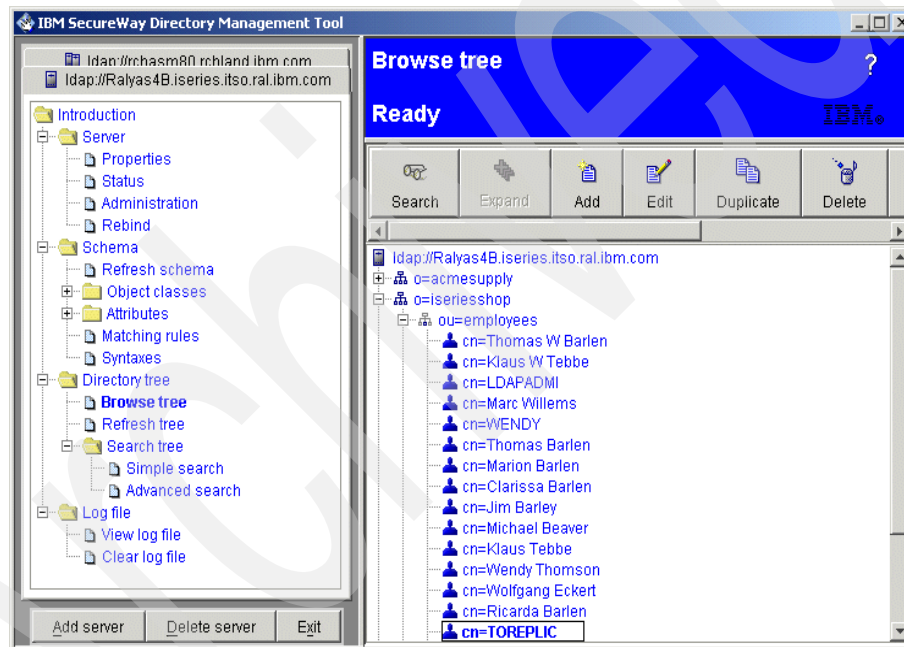


Figure 4-61 IBM SecureWay Directory Management Tool - system RALYAS4B

On the replica system RCHASM80 you see that system RALYAS4B replicated the TORECPLI user to the LDAP directory on RCHASM80.

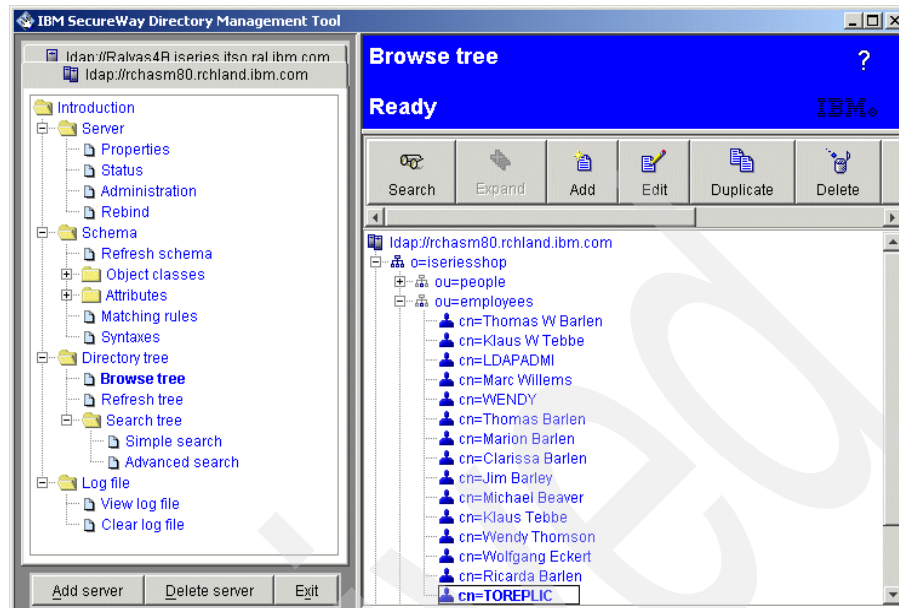


Figure 4-62 IBM SecureWay Directory Management Tool - system RCHASM80

This concludes the replication set up scenario.

## 4.8 Setting up directory referrals

Referrals allow LDAP directory servers to work in teams. If the DN that a client requests is not in one directory, the server can automatically send (refer) the request to any other LDAP server. Directory Services allows you to use two different types of referrals. You can specify default referral servers, where the LDAP server will refer clients whenever any DN is not in the directory. You can also use your LDAP client to add entries to the directory server that have the objectClass referral1. This allows you to specify referrals that are based on what specific DN a client requests.

An important aspect that you need to understand when planning to implement referrals is that the LDAP server that receives a client request is actually not contacting the LDAP server listed as a referral server. It rather returns all configured referral server entries to the client, which in turn retries the initial request on a server returned as a referral, as shown in Figure 4-63 on page 123.



In the example of a default referral, the client submits, for instance, a search request. When the server receives the search request and does not find the entry for `sn=barlen` and `o=automotive`, it returns the `ldap://domweba` reference to the client.

**Important:** Understanding that a client is actually contacting the individual LDAP servers that are returned in a referral response raises another question. What happens if firewalls restrict traffic in your network? You may want to limit clients to access only hosts in a certain subnet. One of these hosts might be an LDAP server that returns referral responses. The client might not be able to contact the server listed in the response as this one is on another subnet. This means that you also need to consider network traffic constraints when implementing referrals.

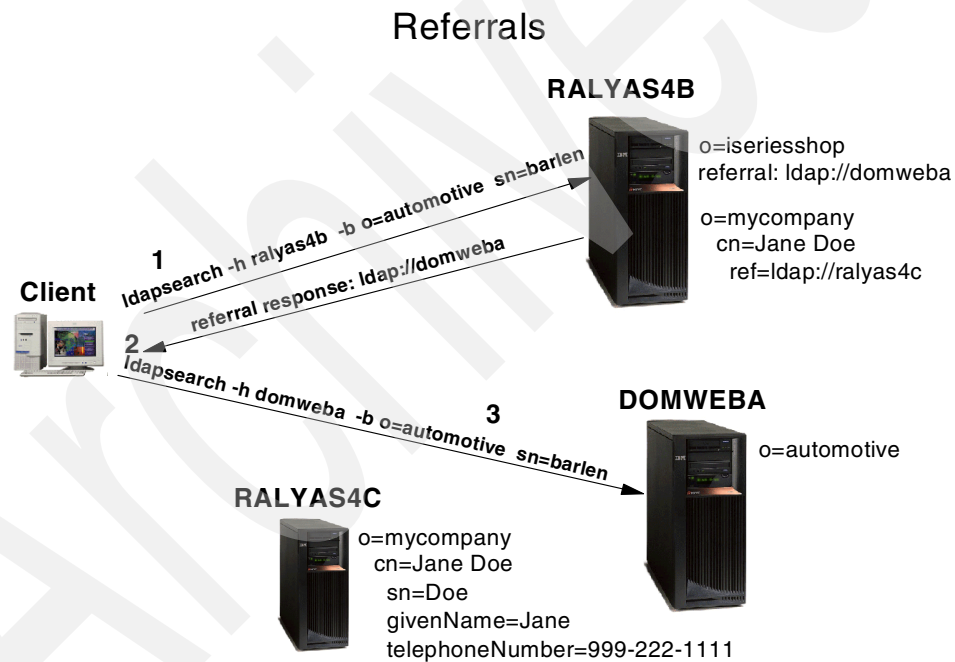


Figure 4-63 Directory referrals

### 4.8.1 Specifying a default server for directory referrals

To assign referral servers for the directory server, perform the following steps:

1. Launch the AS/400 Operations Navigator.
2. Expand the system RALYAS4B.

3. Expand **Network** -> **Servers**.
4. Click **TCP/IP** to display all servers.
5. Right-click **Directory** and click **Properties**.

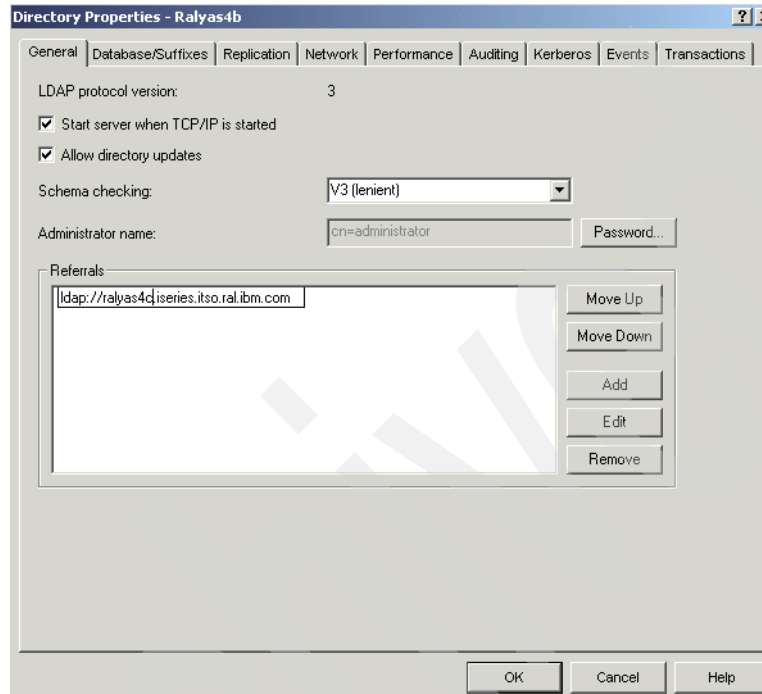


Figure 4-64 RALYAS4B Directory Properties Referrals

6. On the **General** tab click **Add** in the Referrals section.

Enter the fully-qualified TCP/IP host name or TCP/IP address of the referral server. If the referral server does not use the default port, specify the correct port number as part of this host name. In this example we added a default referral for server RALYAS4C in the form of the fully-qualified host name.

7. Click **OK** in the Directory Properties window and restart the server.

In the example shown in Figure 4-64 a single referral entry was added. When a client now connects to the LDAP server RALYAS4B and the directory does not contain the entry the client was looking for, the referral to `ldap://9.24.105.24` is returned to the client. The client is then responsible to retry the request on the server returned as a referral. In case the Referrals list contains multiple entries,

all of them are returned to the client in the order they are listed in the window. The client typically starts its retries with the top entry. You can use the Move Up and Move Down buttons to change the order of the referrals returned to the client.

## 4.8.2 Creating explicit directory referral entries

As opposed to adding default referral servers where all client requests for entries that are not found in the LDAP server the client is connected to, you can also create individual referrals. These referrals consist of a directory entry that has an object class referral. This could be considered as a pointer. The advantage is, for example, that a central directory server can hold just a pointer about people and the directory server that contains more information about them. The actual information, such as names, phone numbers, pictures, and so forth are located on a different server. With individual referral entries you have better control over where your requests will be directed or referred to.

In the following example, a referral entry for an individual object (person) is added to the directory on server RALYAS4B (see Figure 4-63 on page 123). Besides a reference to the server that holds all the details about the person, the new entry contains no further information. We added the entry using an LDIF file and the `ldapadd` utility as follows:

```
dn: cn=Jane Doe,o=mycompany
objectclass: referral
cn: Jane Doe
ref: ldap://ralyas4c.iseries.itso.ral.ibm.com
```

When the client performs now the following search:

```
ldapsearch -h ralyas4b -b o=mycompany "cn=Jane Doe"
```

it get a referral response back from the server RALYAS4B containing the reference to `ldap://ralyas4c.iseries.itso.ral.ibm.com` and retries the search on RALYAS4C. The search results that are displayed on the client would look like:

```
C:\>dapsearch -h ralyas4b -b o=mycompany "cn=Jane Doe"
cn=Jane Doe,o=mycompany
objectclass=ePerson
objectclass=inetOrgPerson
objectclass=organizationalPerson
objectclass=person
objectclass=top
cn=Jane Doe
sn=Doe
givenname=Jane
telephonenumber=999-222-1111
```

As mentioned before, the client takes the referral response returned from RALYAS4B and initiates a new search to the server indicated in the response, in this case the server RALYAS4C. If you want to see the actual referral entry as defined in RALYAS4B's directory you need to include the -M option on the ldapsearch utility as the following example shows:

```
C:\>ldapsearch -h ralyas4b -b o=mycompany "cn=Jane Doe" -M
cn=Jane Doe,o=mycompany
```

This time the displayed search results are from the server where the referral is defined.

**Tip:** Currently, when browsing the DIT with the DMT, it does not display entries that have an object class of referral. In order to search, modify, and delete such entries, you need to use the corresponding ldapsearch, ldapmodify, and ldapdelete utilities with the -M switch.

## 4.9 Securing LDAP traffic

When configuring and testing a new setup, such as our replication scenario, it is always recommended to perform the implementation in stages. For example, you would first set up TCP/IP and test connectivity between master and replica, then you would set up and test the LDAP directory servers and replication. If the replication works, you can continue with securing your traffic between the servers. Even if your master and replica server are in the same corporate network, you want to secure the replication. You wonder why you should put this overhead of encryption to the connection. Well, just think about what kind of data you store in the directory. When you use the LDAP directory for authentication purposes, the userPassword attributes are also replicated. Of course, you may argue that the userPassword is typically encrypted (but it does not have to be) and you do not worry. But other attributes that are protected within the directory might also contain sensitive data, such as employees' home addresses and telephone numbers. During a replication process all this information would flow in the clear over the network. And this is the reason why you should always encrypt replication traffic.

Replication traffic is just one example of why it is important to secure LDAP traffic. You have to decide for your specific situation whether you can afford to live without network traffic security.

This section of the book walks you through the configuration of SSL for the various directory services.

## 4.9.1 When to secure what service

LDAP Directory Services on iSeries consists of one server and two client applications. They are registered under the corresponding application types in Digital Certificate Manager (DCM) in the \*SYSTEM certificate store. Table 4-2 lists these applications and explains what they are used for.

Table 4-2 LDAP Directory Services applications

Application name	Appl. type	Description
Directory Services publishing	Client	This application is used by the publishing engine when secure connections are used to publish user, system, or printer share information, as well as information published under any user defined publishing agents. Application ID: QIBM_GLD_DIRSRV_PUBLISHING.
Directory Services client	Client	This application is used by client applications that use a secure connection but do not specify a keyring file or other DCM application id. If the QSHELL utilities are used with the -Z flag, but do not specify the -K option to identify a keyring file, this application is used. Or if a client calls the ldap_ssl_client_init API and specifies NULL for the keyring parameter, this application is used too. Application ID: QIBM_GLD_DIRSRV_CLIENT.
Directory Services server	Server	This application is used by the server to identify the LDAP server's certificate for secure connections to the server and for replication (master server's connection to the replica server). If server authentication is used, clients must include the certificate issuer in their trusted CA roots. If client and server authentication is used, the server will only trust client certificates issued by trusted CAs associated with this DCM application. Application ID: QIBM_GLD_DIRSRV_SERVER.

## 4.9.2 Installation prerequisites

To use the Secure Socket Layer (SSL) protocol to secure your LDAP traffic you need to meet, in addition, the prerequisites for using LDAP. The following requirements are:

- ▶ 5722-AC2 or AC3 Cryptographic Access Provider (AC2 = 56-bit, AC3 = 128-bit).
- ▶ When using the Directory Management Tool to securely communicate to the LDAP server, you need the full version of the tool. The version that is shipped

with OS/400 does not support SSL. Further details are found in Section 5.2 “Using the DMT to manage the directory” on page 150.

- ▶ 5722-CE2 or CE3 Client Encryption (CE2 = 56-bit, CE3 = 128-bit). This product is used when you want to configure and administer Directory Services from your workstation using an SSL connection with Operations Navigator.

### 4.9.3 Scenario characteristics

In Section 3.1.1 “Stage 2 - The evolution” on page 41 of our scenario, the communication between the master and the replica should be protected. This is especially important as the directory is also used for HTTP Web server and Single Sign-On authentication.

The following Directory Services applications need to be SSL-enabled to meet the scenario objectives:

- ▶ The Directory Services server on RALYAS4B. This allows clients, such as HTTP servers, the WebSphere Applications Server, and Domino to securely authenticate users using LDAP directory entries on the master directory.
- ▶ The Directory Services server on RCHASM80. This allows clients, such as HTTP servers, the WebSphere Applications Server, and Domino to securely authenticate users using LDAP directory entries on the master directory.

### 4.9.4 Enabling SSL for the LDAP server

To enable SSL on the LDAP server on systems RALYAS4B and RCHASM80, you have to perform the following steps via the Digital Certificate Manager. Note that we only show the steps for system RALYAS4B as these steps are identical to the ones to be performed on RCHASM80, except that you need a different server certificate for system RCHASM80.

#### **Assigning a certificate to the LDAP server**

The following steps assign a server certificate to the Directory Services (LDAP) server:

1. Create a server certificate on the LDAP master and the LDAP replica as shown in Figure 4-65 on page 130 for RALYAS4B.

**Note:** In this scenario we used a test certificate from VeriSign. This is a good approach for testing a solution as they are free of charge. At the time this book was written, the test server certificates were valid for 14 days. In a production environment we strongly recommend to use either a certificate issued by a private or local Certificate Authority (CA) if your connections are just in your intranet and in a controlled environment. As soon as you go public or all your various PC clients need to connect to the LDAP securely, you should consider using a certificate issued by a well-known CA, such as VeriSign or RSA Security. In the latter, case this certificate should not be a test certificate.

For a detailed step-by-step description of how to operate DCM and request certificates, refer to IBM @server iSeries Wired Network OS/400 V5R1 DCM and Cryptogrphs, SG24-6168.

The next steps assign the existing certificate to the to the Directory Services server.

2. Open a Web browser and go to the AS/400 Tasks page using the URL:

<http://rallyas4b:2001>

Sign on when prompted with an OS/400 user profile that has \*ALLOBJ and \*SECADM special authorities.

DCM requires that the HTTP \*Admin instance is up and running. You can use Operations Navigator (TCP/IP servers) or the following command to start the \*Admin instance:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

3. From the AS/400 Tasks page click **Digital Certificate Manager**.
4. Click **Select a Certificate Store** on the navigation pane.
5. Select **\*SYSTEM** and click **Continue**.
6. Enter the password for the \*SYSTEM certificate store and click **Continue**.
7. Click **Fast Path** to expand its menu.
8. From the Fast Path menu options on the navigation pane click **Work with server applications**.

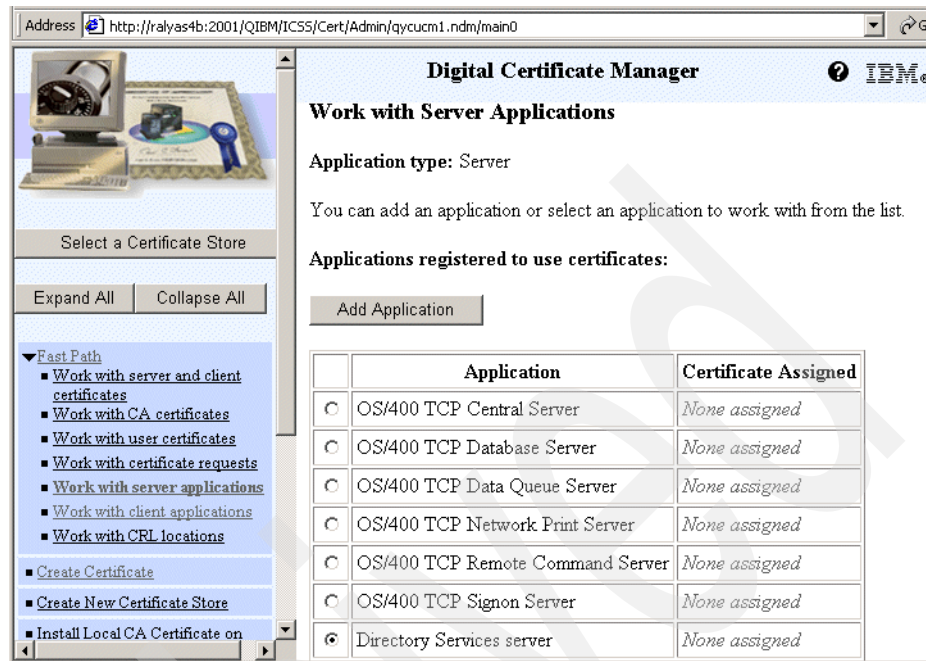


Figure 4-65 Digital Certificate Manager - Work with Server Applications window

- From the list of server applications select **Directory Services server** and click **Work with Application**. This button is found at the end of the displayed list. The application ID for the Directory Services server is QIBM\_GLD\_DIRSRV\_SERVER.



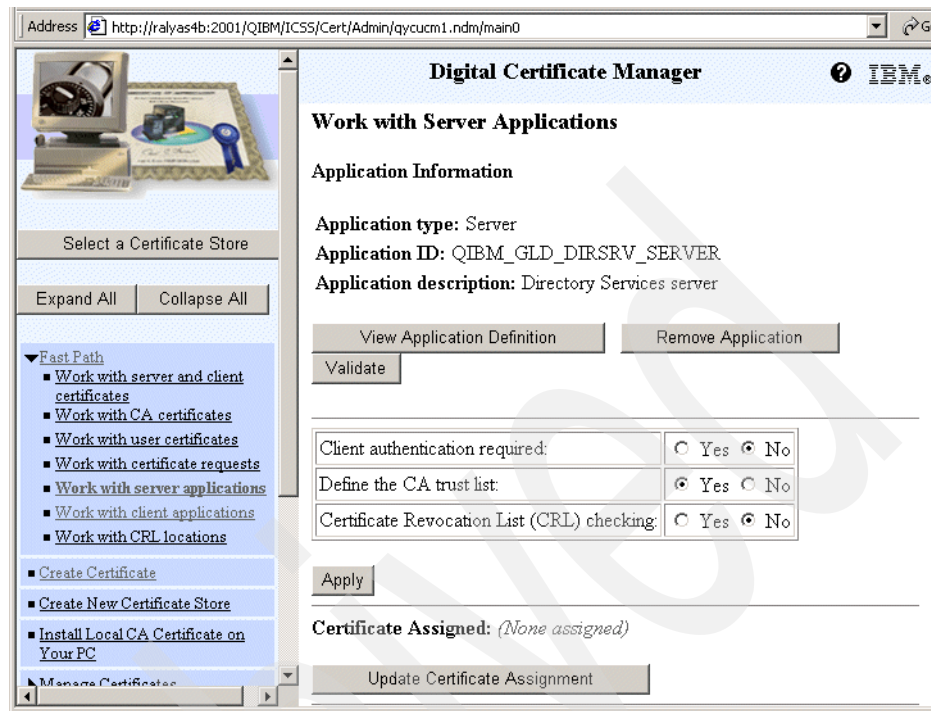


Figure 4-66 Work with Server Applications window

The Work with Server Applications window allows you to change various settings for the selected application. In this case, a new server certificate is going to be assigned to the Directory Services server (LDAP server).

10. Click **Update Certificate Assignment**. The list of available server certificates is displayed, as shown in Figure 4-67 on page 132.

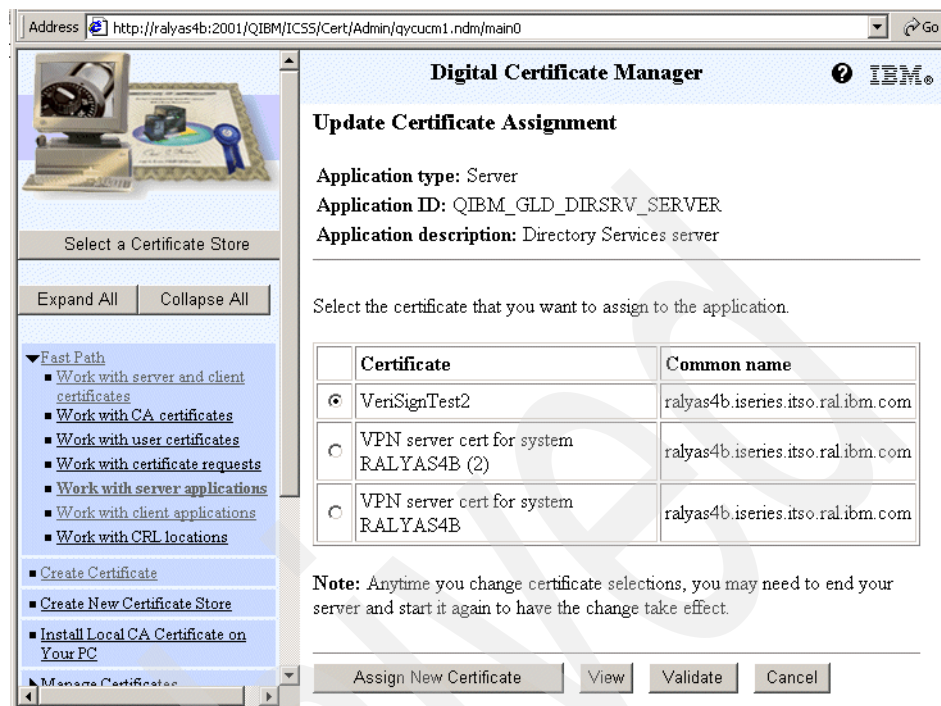


Figure 4-67 Update Certificate Assignment window

11. Select the certificate you want to assign to the Directory Services server and click **Assign New Certificate**. We selected our test certificate we got from VeriSign. A message is shown on this window indicating that the certificate has been successfully assigned.
12. Click **Cancel** to return to the Work with Server Application window.

**Note:** If you want to enable client authentication for the Directory Services server, you also need to define the CA Trust list in the Work with Server Applications window. When client authentication is turned on, the server accepts client connections when the client presents a client certificate that was issued by one of the trusted CAs. The server can not be configured to accept only specific client certificates issued by a trusted CA.

## Changing or verifying SSL for the Directory (LDAP) server

When you assign a certificate to Directory Services server, the Directory properties in Operations Navigator get automatically updated. In the case that you specify in DCM that you do not want to use client authentication, the SSL properties of the Directory server are set to server authentication. When

configuring in DCM that you require clients to authenticate with certificates, the SSL Directory properties are updated to Client and server authentication. Likewise, if you change the SSL settings in Operations Navigator, the settings in DCM are updated too.

The following steps can be used to verify the SSL settings via Operations Navigator:

1. Launch the Operations Navigator.
2. Expand the system RALYAS4B.
3. Expand **Network** and then **Servers**.
4. Click **TCP/IP**.
5. Right-click **Directory** and click **Properties**.
6. Click the **Network** tab.

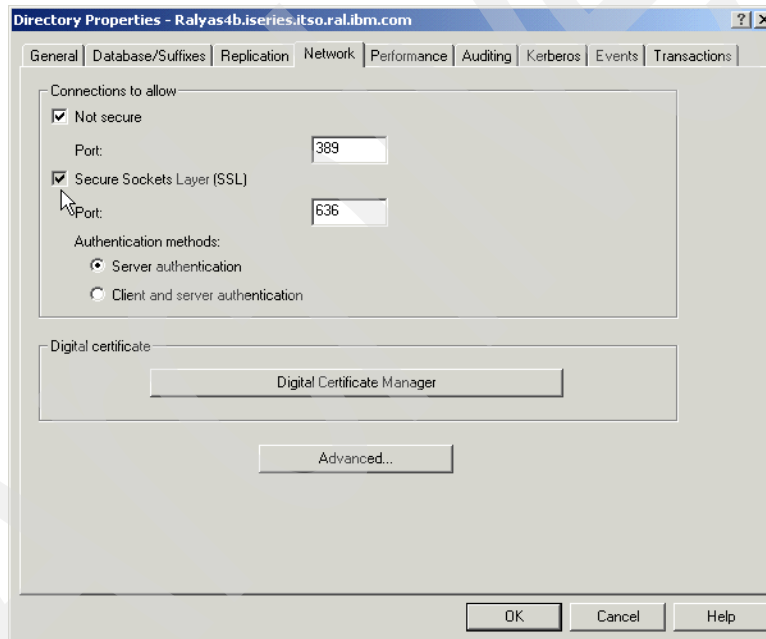


Figure 4-68 RALYAS4B Directory Properties - SSL settings

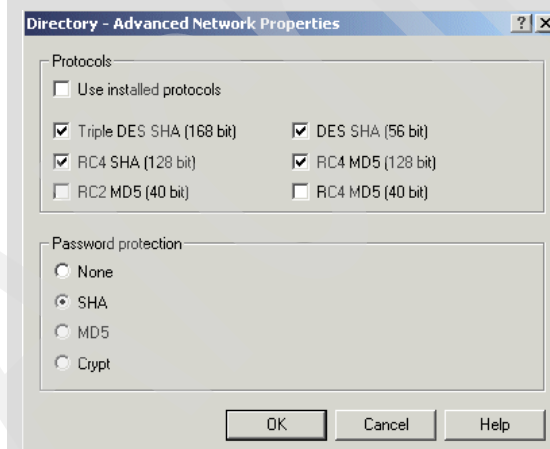
Note that you cannot change the Secure Sockets Layer (SSL) checkbox. As soon as you assign a certificate to the Directory Services server, the SSL checkbox is checked. When removing a certificate assignment, the checkbox is automatically unchecked. The only settings you can change are whether you want to perform server authentication only or server and client authentication. You can launch the Digital Certificate Manager from this window.

7. Select the authentication method. In this scenario we operate with server authentication only.

If you uncheck the Not Secure box then you will allow only SSL connections to the LDAP server, so every client must be SSL-enabled. In this scenario we want non-SSL LDAP clients to be able to query the LDAP server too.

8. Click **OK** to save your settings.

**Tip:** The Advanced button on the Network tab allows you to configure the SSL cipher suites you want to accept during SSL handshake. The SSL ciphers are the authentication and encryption algorithms that can be used to secure the session. If your company's security policy demands a certain cipher, click **Advanced** and select the ciphers you want to allow as shown in the following window. In this example we do not want to accept the relatively weak 40-bit ciphers.



The Directory Services server on RALYAS4B is now ready to accept client requests over a secure connection. To also allow secure connections for the Directory Services server on RCHASM80, you need to perform the same steps as described in this section on that server. Note that our replication process requires that the Directory Services server on RCHASM80 is SSL enabled.

## Configure the master server to replicate over SSL

We now need to tell the LDAP master to use SSL for the replication. To do this:

1. Via Operations Navigator open the Directory properties as described in “Changing or verifying SSL for the Directory (LDAP) server” on page 132.
2. Click the **Replication** tab.

Operations Navigator may prompt you to enter connection information as shown in Figure 4-69.



Figure 4-69 RALLYAS4B - Connect to Directory Server window

Enter the administrator Distinguished name and password, then click **OK**. The list of registered replicas is displayed.

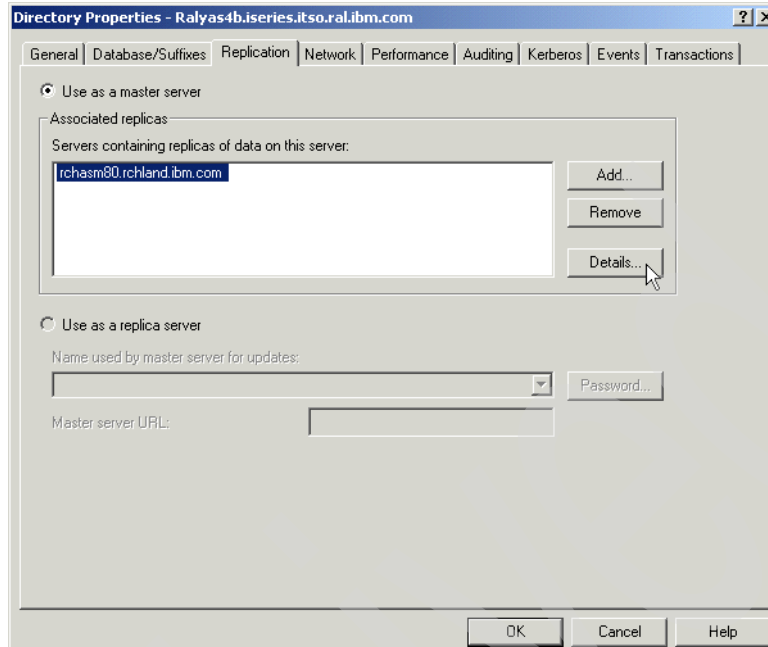


Figure 4-70 RCHASM80 Directory Properties - Replication

3. Select the LDAP replica you want to use SSL for from the Servers containing replicas of data on this server window and click **Details**.

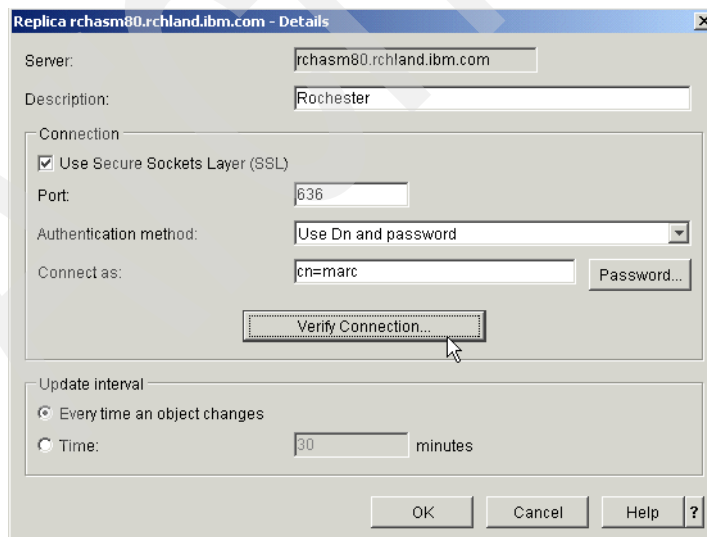


Figure 4-71 RCHASM80 Directory Properties - Replication SSL

Check **Use Secure Sockets Layer (SSL)** to enable SSL for the selected replica. In case you use a port other than the well-known port (636) for secure LDAP connections, specify your port in the Port field.

4. Click **Verify Connection** to verify the connection to the LDAP replica. This will attempt to connect to the replica server by using the distinguished name (DN), password, SSL, and port as shown in Figure 4-72.

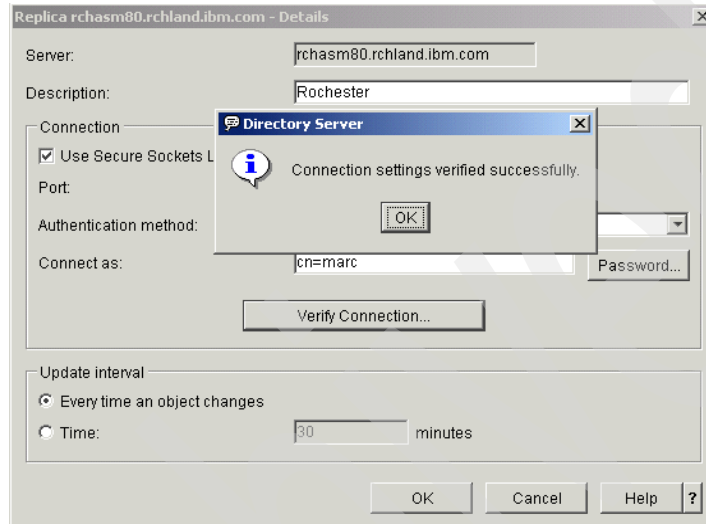


Figure 4-72 Connection verification for replica RCHASM80

5. Click **OK** when the verification completes successfully.
6. Click **OK** in the Directory Properties Replication window to save your changes.

Operations Navigator asks to restart the server, as shown in Figure 4-73.

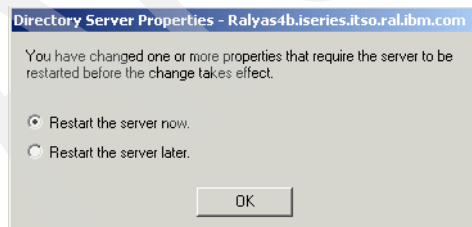


Figure 4-73 Restart window

For the changes to take effect you need to restart the server.

After restarting the LDAP master you can check this SSL connection on the LDAP master and LDAP replica with the command **NETSTAT OPTION(\*CNN)**.

Scroll down until you find the connection to the LDAP replica as shown in Figure 4-74 for RALYAS4B, which is the LDAP master.

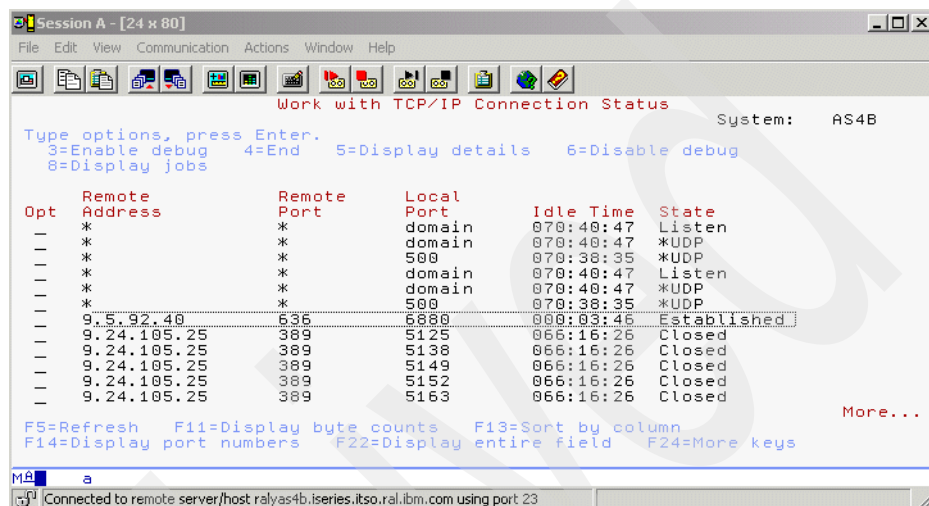


Figure 4-74 RALYAS4B SSL Netstat

This completes the set up for the LDAP replication using SSL data protection.

## 4.9.5 Enabling SSL for the Directory Services publishing client

As explained in Table 4-2 on page 127, the Directory Services publishing client is used to publish information to an LDAP server. If you want to publish, for example, you user (SDD) information over a secure connection then you need to enable SSL for this client.

### Assigning a certificate to Directory Services publishing client

Assigning a certificate to the Directory Services publishing client is only required if the Directory Services (LDAP) server to which the client wants to connect and is configured for server and client authentication. The following steps describe how to assign a certificate to the Directory Services publishing client:

1. In Digital Certificate Manager (DCM) open the \*SYSTEM certificate store as described in "Assigning a certificate to the LDAP server" on page 128.
2. Click **Fast Path** to expand its menu options.
3. From the Fast path navigation pane click **Work with client applications**.



- From the displayed list of client applications select **Directory Services publishing** and click **Work with Application**.

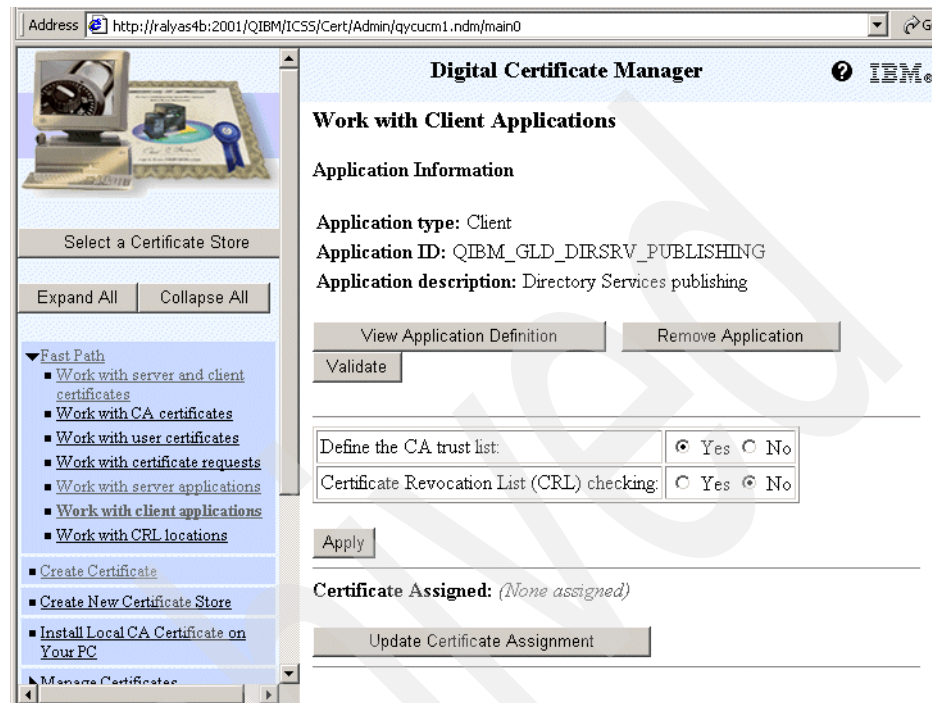


Figure 4-75 Work with Client Applications window

- Click **Update Certificate Assignment**. A list of available certificates is displayed.
- From the list of certificates, select the one you want to use for the Directory Services publishing client and click **Assign New Certificate**.
- Click **Cancel** to return to the Work with Client Applications window.

### Defining the CA Trust for the publishing client

When you want to connect over SSL from the publishing client to the LDAP server, you need to define the list of trusted CAs. That is, you need to tell the publishing client what certificates it can accept from the server. During the SSL handshake the server presents its server certificate to the connecting client. The client now verifies that the presented certificate was issued by a CA it trusts. Perform the following steps to define the CA Trust for the publishing client:

- Start DCM and open the \*SYSTEM certificate store as described in "Assigning a certificate to the LDAP server" on page 128.

2. From the Fast Path navigation pane select **Work with client applications**.
3. From the list of displayed client applications select the **Directory Services publishing** application and click **Work with Application**.

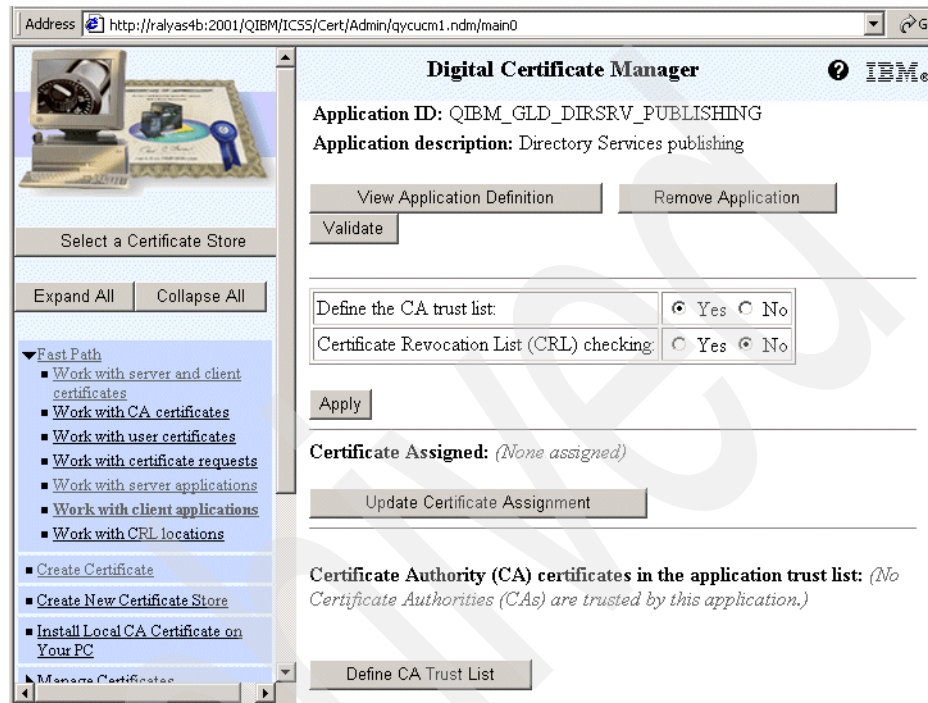


Figure 4-76 Define CA Trust List - Work with Client Applications window

4. Scroll down the Work with Client Applications window and click **Define CA Trust List**.

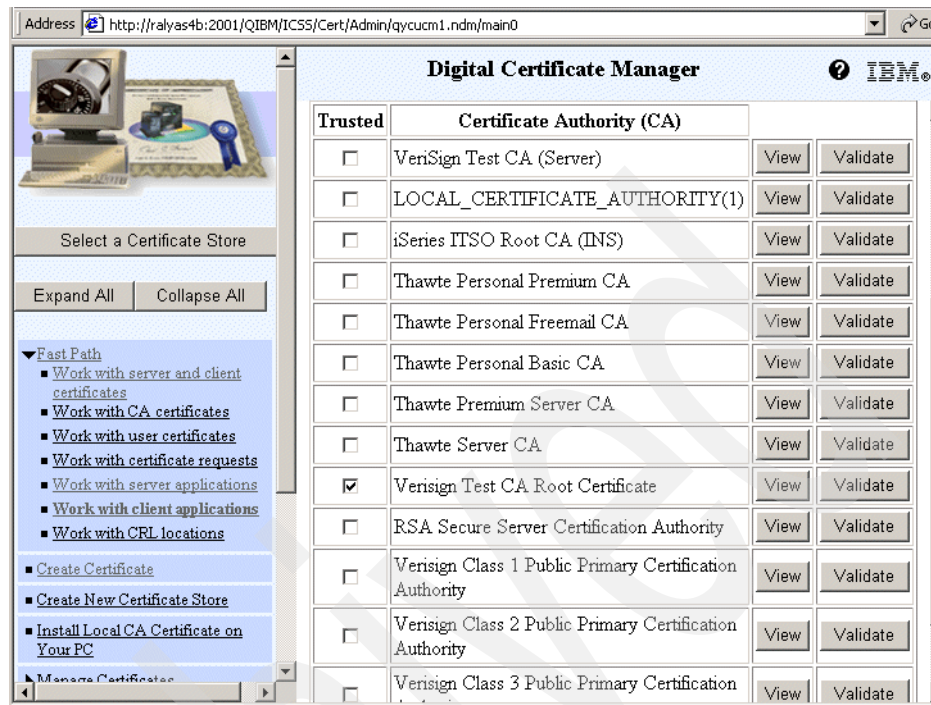


Figure 4-77 Define CA Trust List window

5. Select the CA (that issued the certificate for the LDAP server) and click **OK**. In this example the LDAP server has a certificate assigned that was issued by the VeriSign Test CA.
6. Click **Cancel** to return to the Work with Client Applications window.

When the client now connects to the LDAP server and the server presents a certificate that was *not* issued by the VeriSign Test CA, the connection will fail.

## Changing the publishing settings to use SSL

The last step in enabling the publishing service to connect via SSL to the LDAP server is changing the publishing settings as described in the following steps:

1. Within Operations Navigator right-click the system you want to publish information from.
2. From the displayed context menu select **Properties**.
3. Select the **Directory Services** tab.
4. Select the information you want to change the settings for and click **Details**. In this example, we change the settings for user information.

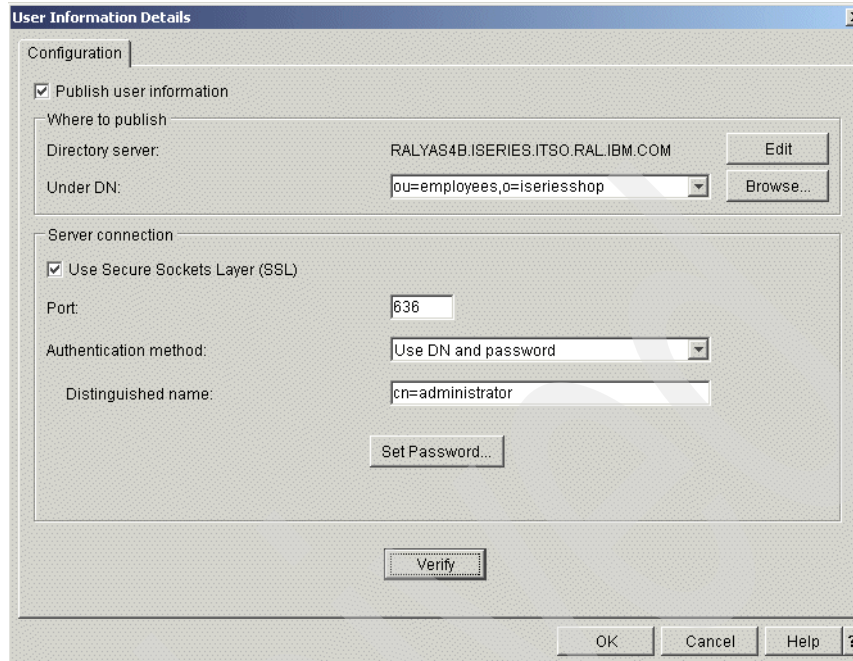


Figure 4-78 User Information Details window

5. Check **Secure Sockets Layer (SSL)**. The port automatically changes from the non-secure port 389 to the secure port 636. Change the port if you configured the server to listen on a different port.
6. Click **Verify** to test the connection to the LDAP server using SSL. You should receive the confirmation message shown in Figure 4-79.



Figure 4-79 Directory Services Settings Verification window

7. Click **OK** to close the window. If you receive a message as shown in Figure 4-80 on page 143 the LDAP server has presented a certificate that is not trusted by the publishing client. Set the correct CA Trust as described in “Defining the CA Trust for the publishing client” on page 139.

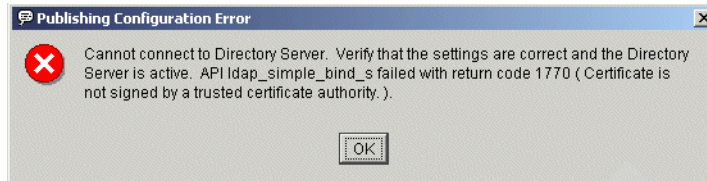


Figure 4-80 Publishing Configuration Error window

8. After successful verification, click **OK** to save the new settings.
9. Click **OK** again to close the system properties window.

You have completed setting up the Directory Services publishing client to connect securely to the LDAP server.

### 4.9.6 Enabling SSL for the Directory Services client

Even if the name of this client is very similar to the one of the publishing client, this client is used by other programs or for other purposes as described in Table 4-2 on page 127. For this client you need to perform the following steps to enable it for SSL:

- ▶ Assign a certificate if the LDAP server the client is connecting to is configured for server and client authentication. The steps of assigning a certificate to the Directory Services client (QIBM\_GLD\_DIRSRV\_CLIENT) application are basically the same as for the publishing client. Refer to “Assigning a certificate to Directory Services publishing client” on page 138 for details on how to assign a certificate. You just need to assign the certificate to the Directory Services client application rather than the Directory Services publishing application.
- ▶ Define the CA Trust List so that your client application accepts only server certificates that were issued by a trusted CA. Details on how to define the CA Trust List can be found in “Defining the CA Trust for the publishing client” on page 139. Just remember to define the CA Trust List for the Directory Services client application and not the Directory Services publishing application.

## 4.10 Directory auditing support

The audit journal can be used to record directory-related incidents. You can use auditing so that you have a record of any operation done to the the directory, including reads of entries or changes to entries.

Beginning with Version 5 Release 1, Directory Services supports OS/400 security auditing. For the auditing function to work, you also need to have security auditing enabled at the system level.

### 4.10.1 Setting up auditing for Directory Services

To audit directory read or change requests you have to set up the auditing journal as described in the *iSeries Security Reference*, SC41-5302. The summary of steps to set up the audit journal for logging directory-related activities is as follows:

1. Create a journal receiver. For example:

```
CRTJRNRCV JRNRCV(QSYS/QAUDJRNRCV) THRESHOLD(100000)
```

2. Create the audit journal. For example:

```
CRTJRN JRN(QSYS/QAUDJRN) JRNRCV(QSYS/QAUDJRNRCV) MNGRCV(*SYSTEM)DLTRCV(*NO)
```

3. Set the QAUDCTL system value to \*OBJAUD. For example:

```
WRKSYSVAL SYSVAL(QAUDCTL)
```

And add the value \*OBJAUD.

To activate Directory Services auditing, perform the following steps:

1. In Operations Navigator under your system, expand **Network -> Servers**.
2. Click **TCP/IP**.
3. Right-click **Directory** and select **Properties** from the context menu.
4. Click the **Auditing** tab.
5. Select the auditing setting that you want to use for your server.

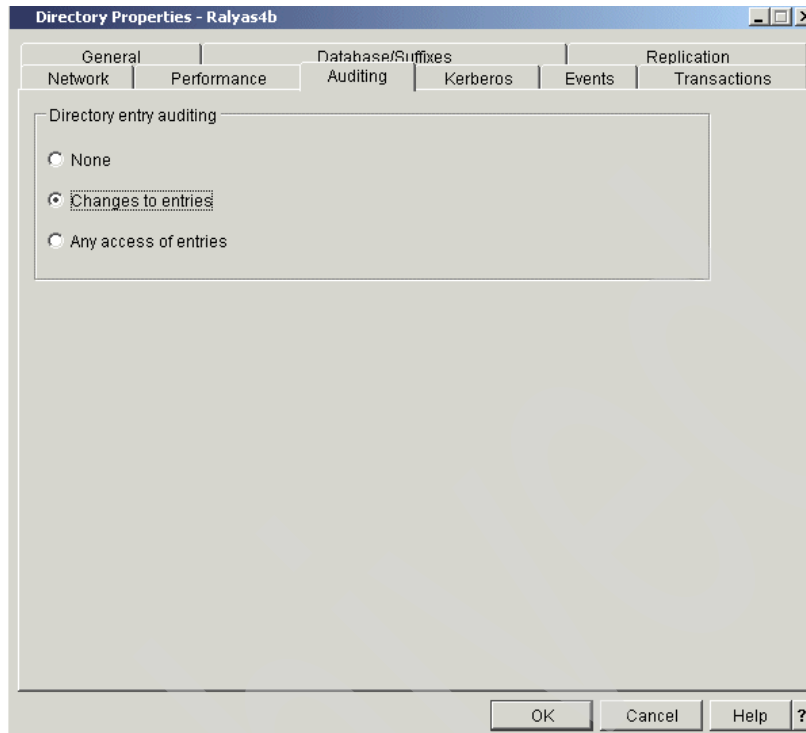


Figure 4-81 Directory Properties window - Auditing tab

You can choose between the following options:

**None**

This is the default and means that no directory requests are logged.

**Changes to entries**

All directory changes are logged into the QAUDJRN. This includes additions, modifications, and deletions using all different kinds of methods, such as the LDAP utilities `ldapadd`, `ldapmodify`, and so forth.

**Any access of entries**

Using this option, all access attempts, whether it is a change (for example, `ldapmodify`) or read (`ldapsearch`), are logged. In a typical environment you would use this option only for a limited time, for example, for debugging problems.

6. Click **OK** to save your settings. Note that changes to auditing settings will take affect as soon as you click OK. There is no need to restart the LDAP directory server.

**Note:** As an alternative to the previously described steps, you can also enable auditing by specifying \*AUDLVL for the system value QAUDCTL and then set QAUDLVL to what you want to audit. For example, setting QAUDLVL to \*AUTFAIL will cause the LDAP server to send an audit record when a bind or an add fails because the bound user does not have authority; if QAUDLVL is \*CREATE, then the server will cut an audit record when an entry is added.

If you are only using \*AUDLVL, then the settings on the Auditing tab of the Directory server properties have no effect. You can have both \*AUDLVL and \*OBJAUD activated.

4.10.2 Audit entry type

As the QAUDJRN logs not only directory service-related entries, you need to know the audit entry type when searching and analyzing the audit journal. The entry type for Directory Services is DI. An example of a journal entry is shown in Figure 4-82.

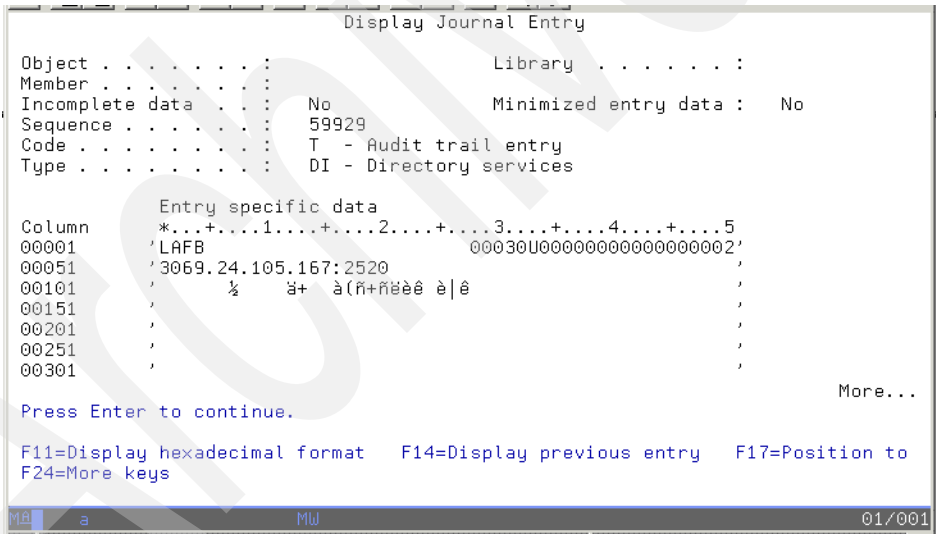


Figure 4-82 Audit journal entry

The example shows a DI journal entry with the following meaning:

- L** The first character represents the entry type and tells us that this is an LDAP operation.
- AF** The next two characters are the operations type, and AF describes an authority failure.



**B** This is a one-character Authority Failure Code field. The value B describes an unauthorized bind attempt.

This is also exactly what happened in our example. We tried to perform an ldapmodify using the wrong password. There is much more information in the journal entry. For example, the scrambled information represents the DN of the user we used to authenticate to the LDAP server. The reason why we cannot display the information correctly is that it is stored in UTF-8 format. You can use the F11 key to switch to a hexadecimal format. In this case the hex value was x'434E3D41444D494E4953545241544F52, which translates into CN=ADMINISTRATOR.

This might be a little bit combersome to interpret. Therefore, for all audit journal entry types, IBM has provided template outfiles you can use to better analyze the audit journal entries. The related audit journal template outfile for the DI entry type is QASYDIJ4 and is available from the QSYS library.



## Managing an LDAP directory on iSeries

The IBM SecureWay Directory, as provided with OS/400 as Directory Services, provides several tools, utilities, and APIs to manage your LDAP directory. In addition to these tools, the iSeries implementation provides some neat extensions in Operations Navigator that allow you to perform certain management tasks even more easily.

This chapter describes various methods and tools to manage your directory data, schema, and authorities.

## 5.1 Different ways to manage your directory

You can manage the LDAP directory information on the iSeries using the following tools:

- ▶ IBM SecureWay Directory Management Tool (DMT)  
The DMT provides you with a graphical user interface for managing LDAP directory content. The DMT is part of the Windows LDAP client that is included with Directory Services.
- ▶ iSeries LDAP utilities  
Directory Services includes five utilities that allow you to perform actions on the LDAP directory server from the QShell command environment on OS/400. These utilities are also shipped with the Windows LDAP client to be used from a Windows workstation.
- ▶ Exporting and importing information through LDIF  
On the iSeries server the Operations Navigator can also be used as a GUI interface for exporting or importing directory data provided with LDAP Data Interchange Format (LDIF) files.
- ▶ Writing your own application to manage your directory  
You can write applications to manage the LDAP directory using the OS/400 LDAP APIs or using the Java Naming and Directory Interface.
- ▶ Third-party tools and programs  
Especially for browsing and searching a directory, there are many tools and programs available in the market. For example, you can use Netscape's address book or Microsoft's Outlook address book to search for entries in the directory.

## 5.2 Using the DMT to manage the directory

The IBM SecureWay Directory Management Tool (DMT) is a graphical tool that allows you to perform the following tasks:

- ▶ Search for a directory entry
- ▶ Add entries
- ▶ Edit entries
- ▶ Duplicate entries
- ▶ Delete entries
- ▶ Create, modify, and delete access control lists (ACLs)

- ▶ Edit the relative distinguished name (RDN) of an LDAP entry
- ▶ Manage and extend your directory schema

The DMT, as shipped with OS/400 Directory Services, is a sort of light-weight version as it does not support SSL connections. With DMT, to connect securely to an LDAP server you need to download the full version of the tool. Details on this topic can be found in Section 5.2.3 "Setting up the DMT for using SSL" on page 158.

More information about the DMT can be found in:

- ▶ The online help provided as an HTML document dparent.htm, stored in a subdirectory of the LDAP client on your PC
- ▶ The SecureWay Directory library at:  
<http://www.ibm.com/software/network/directory/library/>

## 5.2.1 Directory Management Tool installation

The DMT is part of the Windows LDAP client that is included with Directory Services. The client is shipped in the integrated file system (IFS) directory:

`\QIBM\ProdData\OS400\DirSrv\UserTools\Windows`

To install the Windows LDAP client, including the DMT, onto a PC, follow these steps:

1. In Operations Navigator under your system, expand **File Systems**.
2. Click **File Shares** to open a list of defined file shares on the iSeries server.
3. Double-click **Qdirsrv**. The Qdirsrv file share is preconfigured on the iSeries server. It represents the share path `\QIBM\ProdData\OS400\DirSrv`.
4. Double-click **UserTools** to expand the directory.
5. Double-click **Windows**. This is the directory containing the Windows LDAP client setup program.
6. Double-click **setup.exe** to start installing the DMT. Follow the on-screen instructions to complete the installation.

After the installation you can start the IBM SecureWay Directory Management Tool (DMT).

## 5.2.2 Connecting to the LDAP server

Connecting to an LDAP server can be quiet simple or pretty complex depending on the authentication methods you choose. The following steps guide you through the basics of connecting to an LDAP server:

1. From your Windows workstation start the DMT using the path **Start -> Program -> IBM SecureWay Directory -> Directory Management Tool**. This is the default installation path.
2. The DMT is a Java application and it may take a moment until the user interface appears on your desktop. When the tool has started, click **Add server** from the bottom of the navigation pane.

**Note:** When starting the DMT, an error message is displayed indicating that the tool cannot connect to the localhost. This is normal as the default configuration points to the localhost. Just click **OK** and the tool continues starting up.

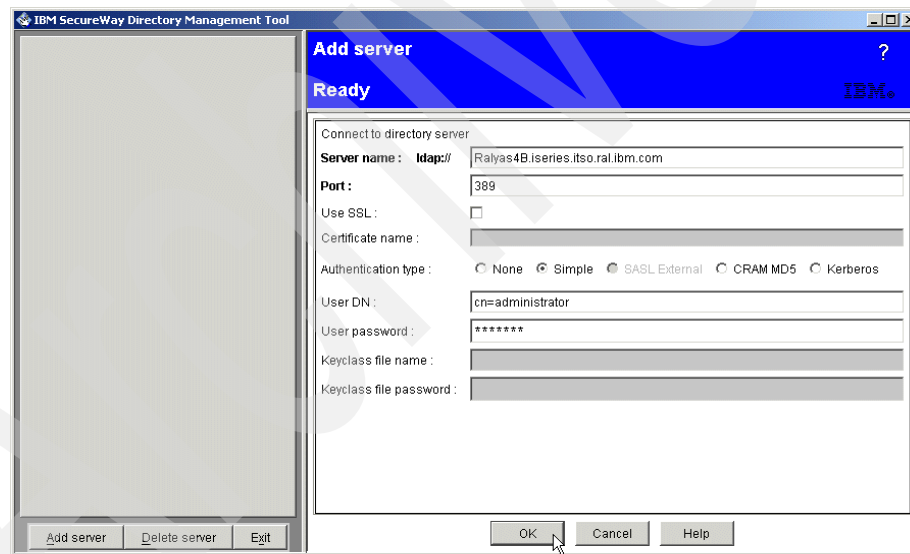


Figure 5-1 IBM SecureWay DMT add system

3. In this example we connect to the server using the administrator distinguished name (DN) to sign on, or in LDAP terms, bind to the LDAP server.

Enter the following information in the Add server window:

<b>Server name:</b> ldap://	LDAP server TCP/IP system name or the TCP/IP address. In this case we connect to our iSeries Shop LDAP server on system RALYAS4B.
<b>Port</b>	389. This is the port the LDAP server listens on. The port automatically toggles between 389 and 636, depending on whether the Use SSL checkbox is checked. 389 is the default; change it to whatever you have configured.
<b>Use SSL</b>	As mentioned earlier, the DMT version shipped with OS/400 does not include SSL support. This means that the necessary Java classes are not included in the package. Checking the Use SSL box with this version of the DMT will result in classNotFound errors. Leave the field unchecked for now. We will describe in Section 5.2.3 "Setting up the DMT for using SSL" on page 158 how to obtain and install the DMT version that supports SSL.
<b>Certificate name</b>	This only used with SSL and specifies the certificate label name of the client certificate to be used for authentication. Refer to Section 5.2.3 "Setting up the DMT for using SSL" on page 158 for more information on how to connect with SSL.
<b>Authentication type</b>	
<b>None</b>	Select <b>None</b> if you want to connect anonymously. Connecting with anonymous access allows you to perform tasks and updates that are allowed by the access group CN=ANYBODY. For more information on authorities refer to Section 5.7 "Controlling access to directory entries" on page 198.
<b>Simple</b>	This is the simplest form of authentication using a DN and a password. In a controlled environment this might be sufficient. However, when using the Simple authentication type, the user and password information flow in the clear over the network. You have to use SSL to protect the user and password information when connecting to the LDAP server.
<b>SASL External</b>	Simple Authentication and Security Layer (SASL) is an authentication framework defined in RFCs. The SASL External option is used with SSL when the

server is configured for client and server authentication. In this case, the client must have a private key (client certificate) that is trusted by the server (issued by a trusted CA). When the SASL External bind is used, the client's identity is formed from the subject DN in the certificate. This is not checked against any entry in the directory, and need not be the DN of an entry in the directory. Verification is handled within the SSL layer. DMT uses the default private key in the specified keyring class file.

### **CRAM MD5**

This bind mechanism uses a challenge string. The client indicates it is using CRAM-MD5 for a specific DN. The server generates a random challenge string and sends it to the client. Both the client and the server hash the challenge string using the password as the key. The client sends its result to the server, which compares the results. To use this mechanism, the server must have access to the clear text password of the `userPassword` attribute, which is stored encrypted in the database. The `QRETSVRSEC` (retain server security data) system value must be set to 1, and password protection in the Directory Services properties must be set to none (which describes the protection applied when the password is retrieved via a search or is replicated).

### **Kerberos**

With the DMT, as provided with OS/400 Version 5 Release 1, Kerberos is still a valid authentication mechanism. However, in the newer version this option is removed. The Kerberos option is also an authentication mechanism supported via the SASL authentication framework.

### **User DN**

Enter the DN of the identity you want to use to bind (authenticate) to the server. When you want to authenticate as the directory administrator, use the DN you entered when configuring Directory Services on your iSeries server as described in Section 4.3 "Configuring OS/400 Directory Services" on page 64. Note that the DMT has some program logic built in that allows you to authenticate users by entering the fully-qualified DN. The DMT searches for entries of all suffixes in



the directory that have the entered a user DN in one of the attributes listed below.

If there is only one word, for example, smith, the DMT searches all suffixes for an entry matching one of these filters:

```
(cn=smith)
(sn=smith)
(uid=smith)
```

If there are two words, for example, John Smith, DMT searches all suffixes for entries matching:

```
(cn=John Smith)
(uid=John Smith)
```

If there are three words, for example, John Smith Jr, DMT searches all suffixes for an entry matching one of these filters:

```
(cn=John Smith Jr)
(uid=John Smith Jr)
```

And, if there happen to be more words, for example, a name with more words, DMT just looks for an entry matching this filter:

```
(cn=a name with more words)
```

**User password**

The password of the user entered in the User DN field.

**Keyclass file name**

This only used with SSL and specifies the name of the key database that holds client certificates and trusted root certificates. Note that the DMT client as shipped with OS/400 does not support SSL. Refer to Section 5.2.3 "Setting up the DMT for using SSL" on page 158 for more information on how to connect with SSL.

**Keyclass file password**

A keyclass along with its content is encrypted. This password is the password that was used to protect (encrypt) the keyclass file.

4. Click **OK** to connect.

The IBM SecureWay Directory Management Tool (DMT) starts and retrieves the LDAP server schema, as shown in Figure 5-2 on page 156.

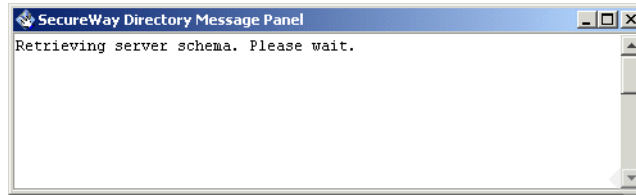


Figure 5-2 IBM SecureWay DMT retrieving server schema

The Directory Management Tool displays servers as tabs at the top of the navigation area. Each tab displays the name of the associated directory server.

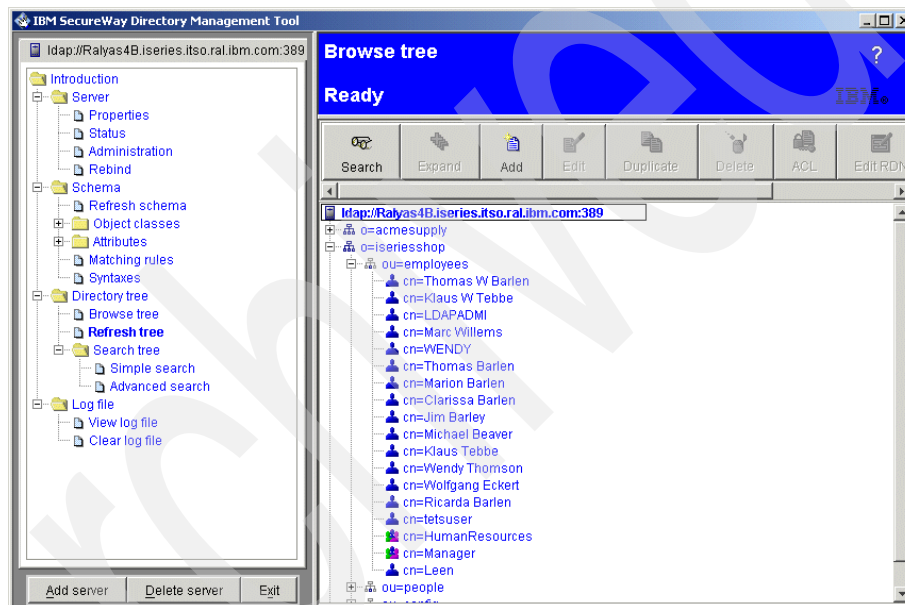


Figure 5-3 IBM SecureWay DMT Browse tree

The retrieved schema is displayed in the IBM SecureWay Directory Management Tool main window, as shown in Figure 5-3.

You are now ready to manage your directory. A few examples of management tasks are covered later in this chapter.

When you exit the DMT it will not remember the servers you specified on the Add server page, as shown in Figure 5-1 on page 152. To permanently add the servers so that every time you start the DMT you bind to the same servers, you need to edit the configuration file `dmt.conf` located in the LDAP client installation subdirectory `LDAP/etc/`.

The following configuration file is used in our redbook scenario to manage the LDAP master and replica servers as described in Section 4.7 "Setting up directory replication" on page 106.

*Example 5-1 LDAP/etc/dmt.conf file*

---

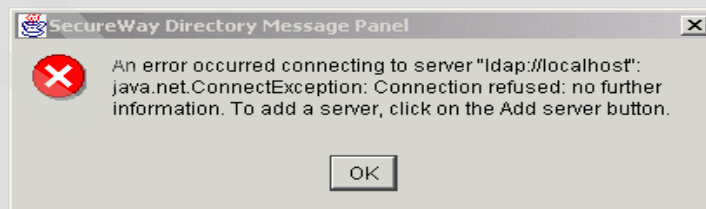
```
#browser=
#toolbar=both
server1.url=ldap://ralyas4b.iseries.itso.ral.ibm.com:389
server1.security.bindDN=cn=adminstrator
server1.security.password=your_password
#server1.security.ssl.keyclass=
#server1.security.ssl.keyclass.password=
#server1.admin.url=http://webserver:80
server2.url=ldap://rchasm80.rchland.ibm.com:389
server2.security.bindDN=cn=adminstrator
server2.security.password=your_password
server3.url=ldap://localhost:389
```

---

When started, the tool connects to the non-SSL port 389 of the ralyas4b.iseries.itso.ral.ibm.com (master) and rchasm80.rchland.ibm.com (replica) directory servers. The DMT binds to the servers with the DN as cn=adminstrator and *your\_password* as password. If you do not want to keep a clear password in the configuration file in an unprotected directory, leave the principal name empty to perform an initial bind to the server as an anonymous user, then rebind within the DMT with the correct DN.

Lines that start with # are comment lines and are not processed.

**Tip:** During our tests we found out that the localhost entry must be in the configuration file, otherwise the IBM SecureWay Directory Management Tool will not start. However, if the localhost entry is in the configuration file you receive the following message:



Just click **OK** to continue. The DMT starts connected to the servers specified in the server1 and server2 entries.

### 5.2.3 Setting up the DMT for using SSL

The version of the IBM SecureWay DMT and directory client SDK that is shipped with OS/400 does not include SSL support.

To use SSL with the DMT you will need to get IBM SecureWay Directory Version 3.2.2 for Windows NT from the IBM SecureWay Directory Web site:

<http://www.ibm.com/software/network/directory>

In order to use SSL as documented in this chapter you need to download at least the following two packages:

- ▶ IBM SecureWay Directory Version 3.2.2 for Windows NT (ldap322win.zip)
- ▶ GSKit V5 SSL Kit (gskit5d-win.zip)

After downloading the packages you need to unzip the ldap322win.zip file first. The extracting process creates in the extraction directory a subdirectory called gskit. Extract the second file, gskit5d-win.zip, into this subdirectory.

**Tip:** If you have previously installed the IBM SecureWay Directory from the iSeries, then you must first remove it using the Windows Control Panel. If you used the dmt.conf file to configure the DMT you must move it to a safe place on your PC before removing the DMT, and you can reuse this file with the new DMT.

Perform the following steps to install the SSL-supporting DMT version after the ZIP files have been extracted:

1. Double-click the **setup.exe** program from the PC directory you unzipped (extracted) the LDAP322WIN.ZIP file from.
2. Follow the on-screen instructions to continue and select **Custom** installation.

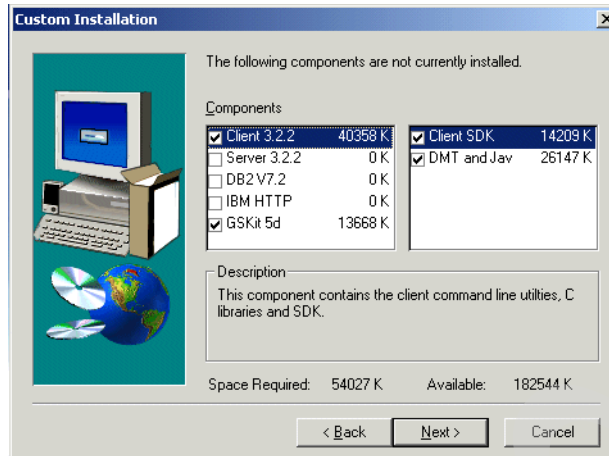


Figure 5-4 DMT with SSL support

3. Check only the **Client 3.2.2** and the **GSKit 5d** options as shown in Figure 5-4 and click **Next**. The GSKit option is required to use the DMT with SSL.
4. Continue with the on-screen instructions to complete the installation. After the installation is finished, restart your PC.

Before you can use SSL on the IBM SecureWay Directory Management Tool you need to create a keyclass file. The keyclass file is used to store trusted root CA certificates as well as personal certificates. In the case of using SSL with the DMT, the CA that issued the server certificate must be in the keyclass file. When the LDAP server has also turned on client authentication, you need to add your client certificate as a personal certificate to the keyclass file too. By default, the keyclass has to be stored in the DMT directory \bin. The default directory would be c:\Program Files\IBM\LDAP\bin\.

Perform the following steps to create a keyclass file:

1. Open your Windows Explorer and expand to the GSKit program directory. By default this is \Program Files\IBM\GSK5\bin\.
2. Start the IBM Key Manager tool by double-clicking the program icon with the name gsk5ikm.exe.

As an alternative you can also use the IBM Key Manager tool that is delivered as part of IBM Client Access Express SSL support and is installed using the Client Access Express Selective Setup.

3. From the IBM Key Management tool main window, select **Key Database File**, as shown in Figure 5-5 on page 160.

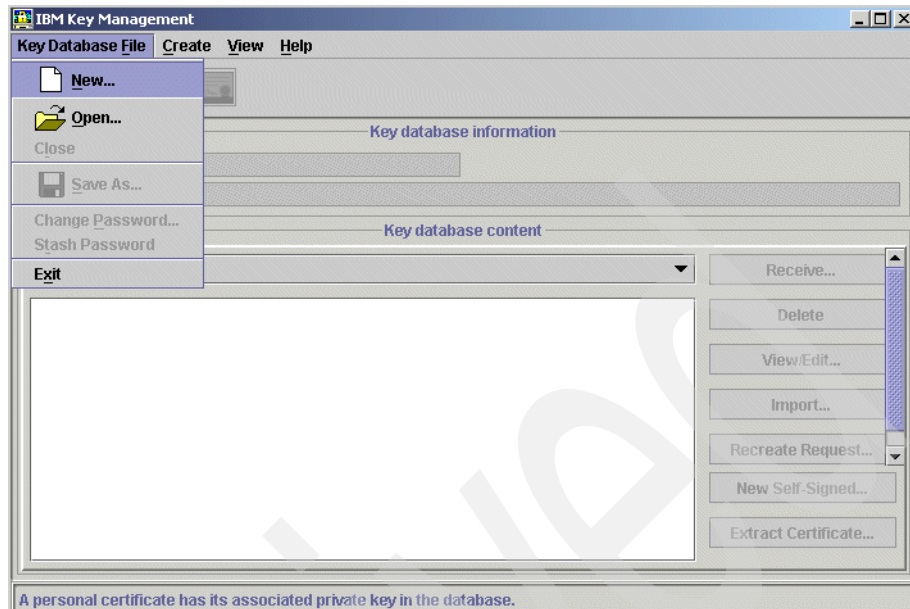


Figure 5-5 IBM Key Management New keyclass file

4. Select **New** from the Key Database File menu.

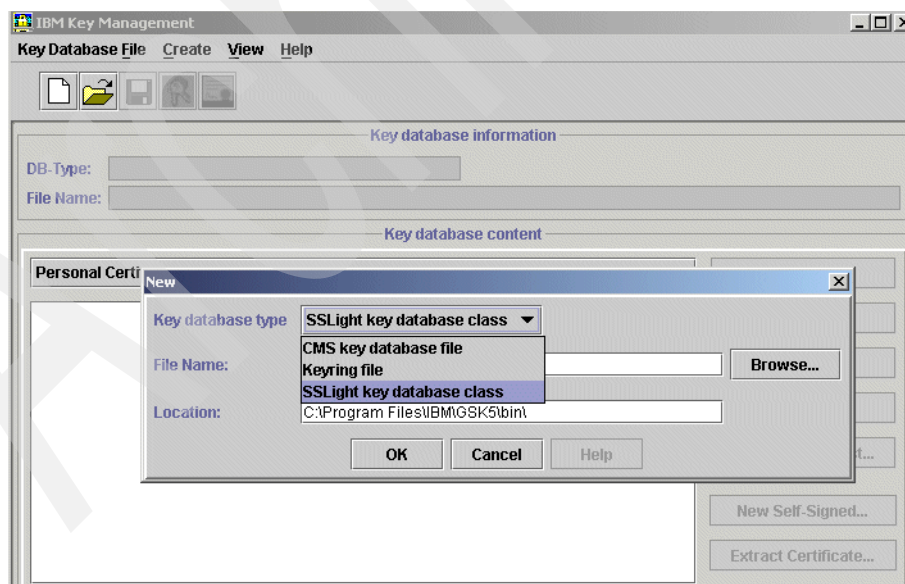


Figure 5-6 IBM Key Management - select SSLight key database class file

5. Select **SSLight key database class** from the Key database type field as shown in Figure 5-6 on page 160.
6. Enter the name of the file name in the File Name field. This file name is case sensitive. In this example, LDAP.class.
7. Enter C:\Program Files\IBM\LDAP\bin\. This must be the \bin directory of the IBM SecureWay Directory Management Tool.
8. Click **OK** to continue. The Password Prompt window is displayed.

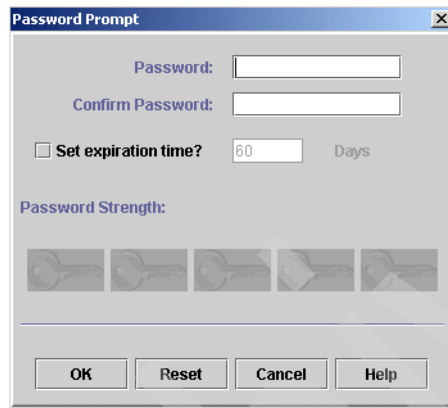


Figure 5-7 IBM Key Management - select Password Prompt

9. Enter the keyclass file password in the Password Prompt window. This password is used to encrypt the keyclass file with its content. Applications, such as the DMT, need to access the keyclass file to verify, for example, if the presented server certificate was issued by a trusted CA. In order for the DMT to decrypt the keyclass file, you need to provide the password you specified on the window, as shown in Figure 5-7.
10. Click **OK**.

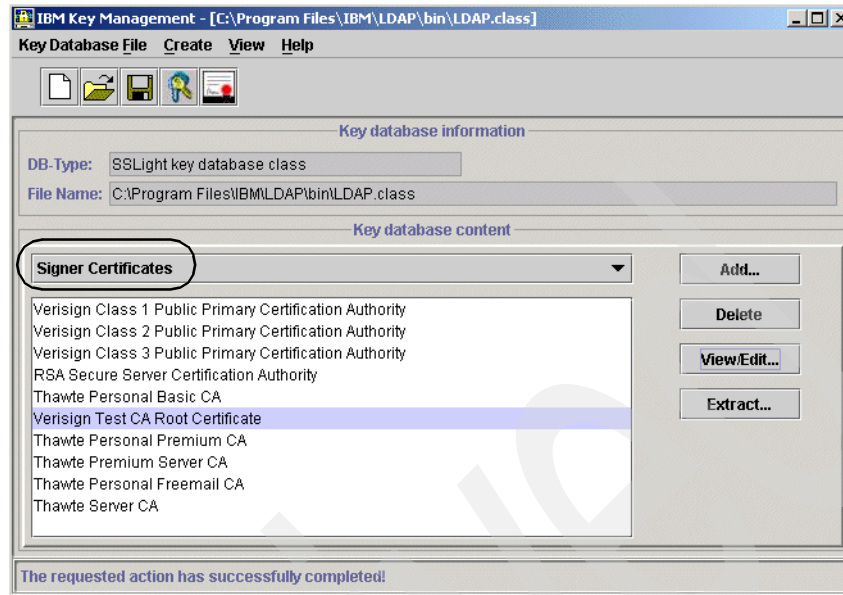


Figure 5-8 IBM Key Management - Signer Certificates

If the LDAP server's certificate is signed by a CA other than the well-known CAs as listed under Signer Certificates, you have to add this CA to the list. We used a server certificate that was issued by the VeriSign Test CA, which was already listed as a trusted signer certificate as Verisign Test CA Root Certificate.

11. Exit the IBM Key Management tool by selecting **Key Database File** and clicking **Exit**.

After saving the keyclass file you can start the IBM SecureWay Directory Management Tool and connect to the LDAP server using SSL as described in the following steps:

1. Start the DMT using the following path: **Start -> Program -> IBM SecureWay Directory -> Directory Management Tool**.
2. Click **Add server**.



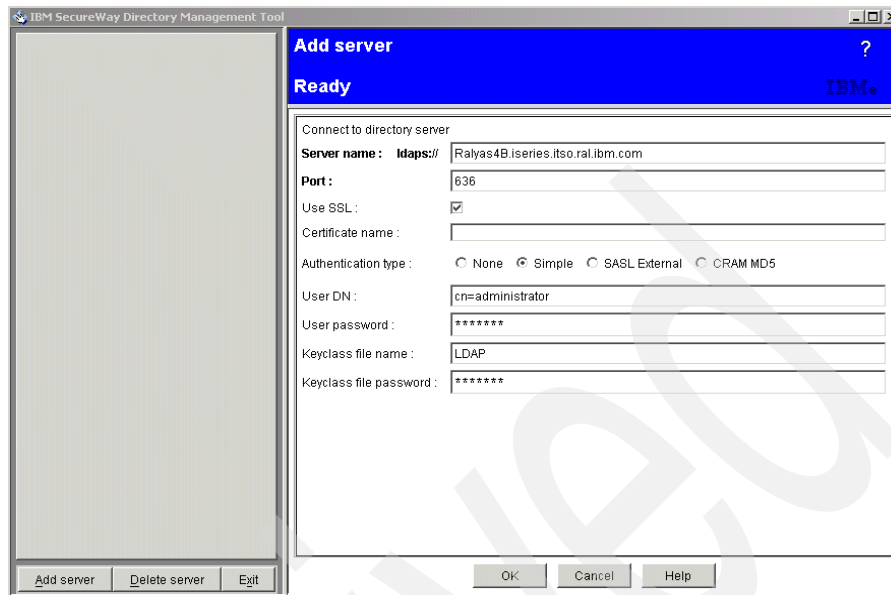


Figure 5-9 IBM SecureWay DMT SSL configuration

3. Enter the following parameters:

**Server name**

Enter the LDAP server TCP/IP system name or the TCP/IP address.

**Port**

After switching from non-secure to SSL, the Port value automatically changes from 389 to 636, the well-known port for secure LDAP. If you configured the LDAP server to listen on a different port specify this port now.

**Use SSL**

Check this checkbox.

**Certificate name**

If the LDAP server requires client authentication, you need to enter the certificate label name of the client (personal) certificate that is stored in the keyclass file. The certificate associated with the label is then presented to the LDAP server during SSL handshake. In this example we do not use SSL client authentication and, thus, leave the field blank.

**Authentication type**

In this scenario we use simple authentication with a user DN and password. Confidentiality of the user and password is given by the use of SSL. Select **Simple**.

<b>User DN</b>	Enter the DN of the identity you want to use to bind (authenticate) to the server. When you want to authenticate as the directory administrator, use the DN you entered when configuring Directory Services on your iSeries server as described in Section 4.3 "Configuring OS/400 Directory Services" on page 64.
<b>User password</b>	The password of the user entered in the User DN field.
<b>Keyclass file name</b>	The name specifies the name of the keyclass file that holds client certificates and trusted root certificates. Use the name of the file as specified in Figure 5-6 on page 160, but without the .class extension or any path information. As mentioned previously, the keyclass file should, by default, be stored in the \bin directory of the DMT. In this scenario the value entered as the Keyclass file name is LDAP, all in uppercase. In case you stored the keyclass file in a directory other than the \bin directory you need to specify the -k switch when starting the DMT, as in <code>dmt -k \mykeydir</code> .
<b>Keyclass file password</b>	A keyclass, along with its content, is encrypted. This password is the password that was used to protect (encrypt) the keyclass file. Enter the password you specified when creating the keyclass file.

4. Click **OK** to connect to the LDAP server using a secure connection.

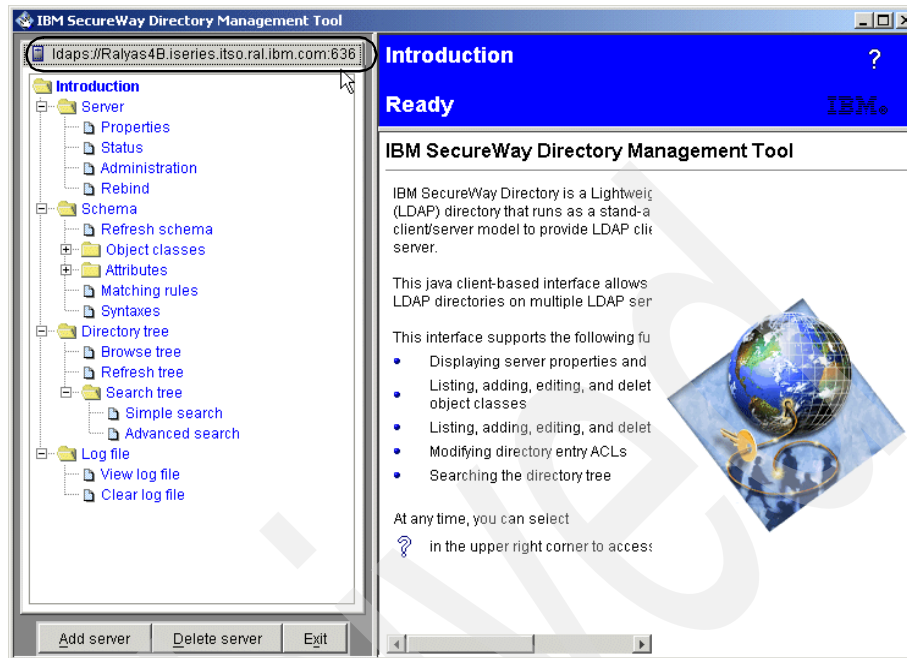


Figure 5-10 IBM SecureWay DMT using SSL

As shown in Figure 5-10, the IBM SecureWay DMT is connected via the ldaps protocol (secure LDAP) to the server on system RALYAS4B using SSL port 636.

For further details about connecting to the LDAP server using the DMT and how to add server connections permanently, refer to Section 5.2.2 "Connecting to the LDAP server" on page 152.

## 5.2.4 Adding organization entries using the DMT

As described in the redbook scenario in Section 3.1.1 "Stage 2 - The evolution" on page 41, the iSeries Shop publishes employee information from the SDD to the LDAP directory. They also use the directory to store user information of their premium customers. The directory path as described in Section 3.1.5 "The scenario Directory Information Tree" on page 49 requires an organization called `iseriesshop` with three branches called `employees`, `people`, and `config`. In addition, an organization of `acmesupply` will be added to accommodate information about employees of ACME Supply. This section shows you how to add organizations and organizational units to your directory information tree (DIT).

**Important:** Prior to adding the organization iseriesshop or acmesupply to the directory as described in the following steps, you need to add their suffixes via Operations Navigator to the Directory properties. Adding suffixes is described in Section 4.3.1 "First-time configuration" on page 64. Without adding the suffixes first, you will not be able to add that entry under the *top* parent DN.

Perform the following steps to add the iseriesshop organization to the DIT:

1. If not already done, start the DMT binding with a DN that has administrator authority.
2. On the navigation pane click **Browse tree**.
3. Select the directory root (the server name) and click **Add** on the toolbar.

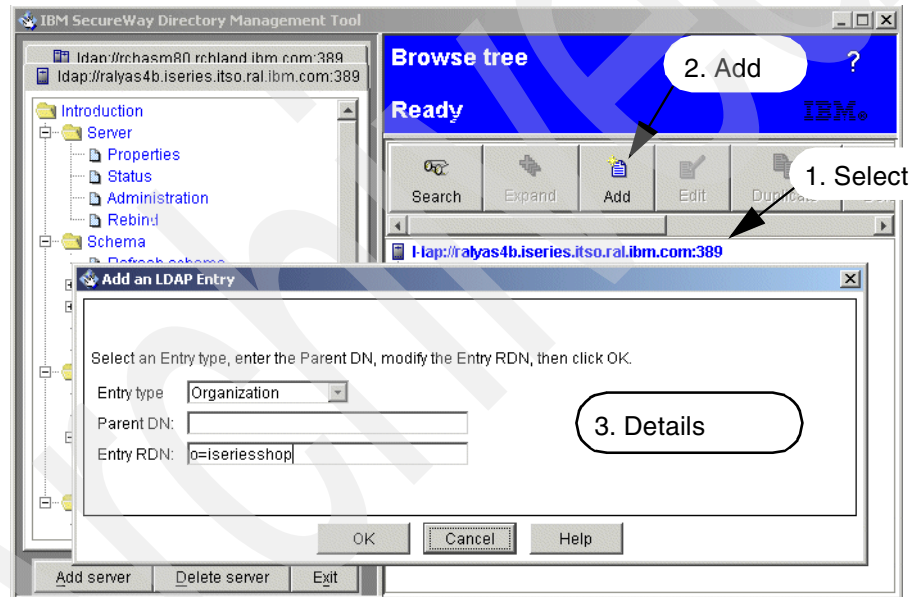


Figure 5-11 Add an LDAP Entry window

4. Enter the following values to add the organization iseriesshop:

<b>Entry type</b>	Organization
<b>Parent DN</b>	Blank (a blank parent DN represents the top (root) of the directory)
<b>Entry RDN</b>	o=iseriesshop

In this case a new entry of type organization is added under the root of the directory. Note that all entries you want to add under the root have to be added as a suffix to the Directory properties. The new entry's relative distinguished name (RDN) is o=iseriesshop where o is the attribute name for an organization.

5. Click **OK** to continue. A window with all attributes for the new entry is displayed.

The screenshot shows the 'Add an LDAP Entry' dialog box. At the top, it says 'To add a new entry, enter values for the attributes, then click Add.' Below this, the 'Object class' is set to 'organization' and the 'DN' is 'o=iseriesshop'. The 'Attributes' tab is active, showing a list of attributes. The attribute 'o:' is bolded and has a value of 'iseriesshop'. Other attributes are non-bolded and empty. Annotations on the right side of the window identify the 'Object class' dropdown as 'Object classes', the 'o:' attribute as 'Required attribute', and the other attributes as 'Optional attributes'.

Figure 5-12 Add an LDAP Entry - Attributes window

The window shown in Figure 5-12 allows you enter information for further attributes. In this example, the organization object class has just one required attribute o (organization). All required attributes in an object class are displayed in bold and must have a value defined. The non-bold attributes are optional and can, but do not have to, be filled in. For more information about the directory schema, object classes, and attributes refer to Section 5.2.5 "Using the DMT to browse the directory schema" on page 170.

6. After filling in all the information you need or want to provide, click **Add** to add the entry to the directory.

Repeat the previous steps to add the acmesupply organization as described in Section 3.1.5 "The scenario Directory Information Tree" on page 49.

## Adding the organizational units using the DMT

Now that the iserieshop organization has been added to the directory, you can add child entries to the new organization as described in the following steps:

1. While you are still browsing the tree, click on the new organization **o=iserieshop**.
2. Click **Add** on the toolbar.

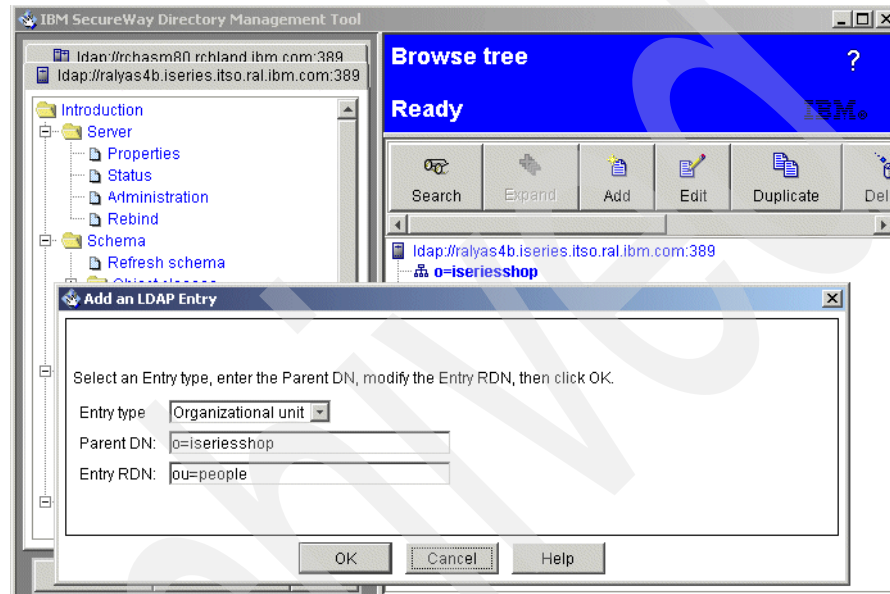


Figure 5-13 Add an LDAP Entry window

3. Enter the following values to add the organizational unit people:

**Entry type**                      Organizational unit.

**Parent DN**                      This will be the parent of the new entry. In this scenario the new entry is published under the parent o=iserieshop.

**Entry RDN**                      ou=people.

Using the previous parameter values, a new organizational unit with the RDN of ou=people is added under the organization o=iserieshop. The fully qualified DN is ou=people,o=iserieshop.

4. Click **OK** to continue. A window with all attributes for the new entry is displayed.
5. After filling in the attribute values click **Add** to add the new entry.

According to the redbook scenario you need to add the remaining two organizational units employees and config by repeating the previous steps.

The steps described in this section apply to any entry you might want to add. You need to decide first under which parent DN you want to publish the new entry. You also need to know what object class you want to use for the new entry and finally provide the name for the new entry. For example, if you want to add a person entry to the directory, you may want to use the ePerson object class. This object class contains all attributes inherited from the person object class in addition to its own attributes. An object class might have several required attributes, which have to be filled in, and several optional attributes you may want to fill in. However, what all attributes have in common are syntax rules for their values. Some values might allow any random character string, while others require a numeric value and others a value in a form of a distinguished name. In any case, if you enter a value that does not match the required attribute syntax you receive an error message, as shown in Figure 5-14.

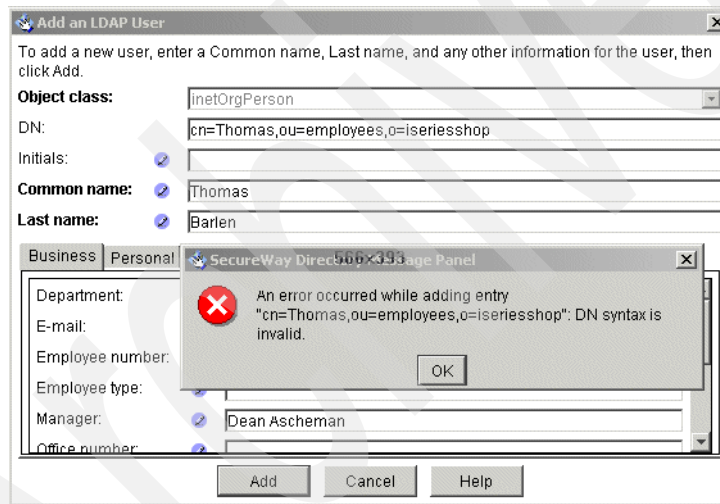


Figure 5-14 Add an LDAP Entry - error message

Figure 5-14 shows an example of adding a new person object of object class inetOrgPerson. Required attributes are Common name (cn) and Last name (sn). In addition, an optional attribute is filled in, the Manager of the new person. When trying to add this entry an error occurred indicating that a DN syntax is invalid. In this example it is the person's manager attribute that has the wrong syntax. The attribute expects a name in the form of a DN, such as cn=Dean Ascheman,ou=employees,o=iseriesshop, rather than a plain name. Sometimes the difficulty is finding out what the actual problem is, especially when filling in values for several attributes. The best way to isolate the problem is to check the directory schema, as described in the next section.

## 5.2.5 Using the DMT to browse the directory schema

You can use the Directory Management Tool to manage the directory schema of your LDAP server. The DMT allows you to browse and extend your directory schema. You should be very careful when extending a directory schema, as you need to have very good knowledge about object classes, attributes, syntaxes, and so forth. For most requirements and needs, the directory schema as provided with the IBM SecureWay Directory and thus with OS/400 Directory Services, has sufficient object classes and attributes to accommodate all information. For more information on extending a directory schema refer to Appendix B, “Extending your directory schema” on page 521.

Browsing a directory schema is a common task you have to perform when working with object classes you are not familiar with. To better understand how to navigate through the schema let us use the following example.

You want to add an entry for employees in your organization. Typically you would select the directory branch under which you want to add the new entry and click **Add** from the toolbar. Then you select the entry type. There are a few entry types already predefined. In this case we select the User entry type and specify the common name for the new entry. When continuing with adding the new entry you get a window, as shown in Figure 5-15 on page 171.



Figure 5-15 Add an LDAP User window

As you can see in Figure 5-15, the User entry type uses several object classes when adding a new entry. They are:

- ▶ inetOrgPerson
- ▶ organizationalPerson
- ▶ Person
- ▶ top
- ▶ ePerson

What actually happens is that not all of the listed object classes are added manually, rather they are inherited. For the User entry type, the inetOrgPerson has its own attributes and inherits from its superior class organizationalPerson, which in turn inherits from the Person class. The Person's superior class is top. That counts for all classes except ePerson. The ePerson object class is an auxiliary object class. This means that ePerson cannot be used on its own. It has to be defined along with another structural object class, such as inetOrgPerson. The new entry has then all the characteristics of all object classes.

All the attributes shown in Figure 5-15 on page 171 are defined in one of the object classes that are listed in the Object class parameter. But how do you find out which attribute belongs to which object class and what the attribute syntax is? You may also want to know what security class a specific attribute has assigned. This can be done by browsing the directory schema as shown in the following steps:

1. Start the DMT binding as administrator to the LDAP server.
2. On the navigation pane expand **Schema** and then **Object classes**.
3. Click **View object classes**. All defined object classes in the directory schema are displayed.
4. Scroll the list of object classes until you find the one you want to have more information about. In this example it is the `inetOrgPerson`.
5. Expand the object class as shown in Figure 5-16 on page 173.

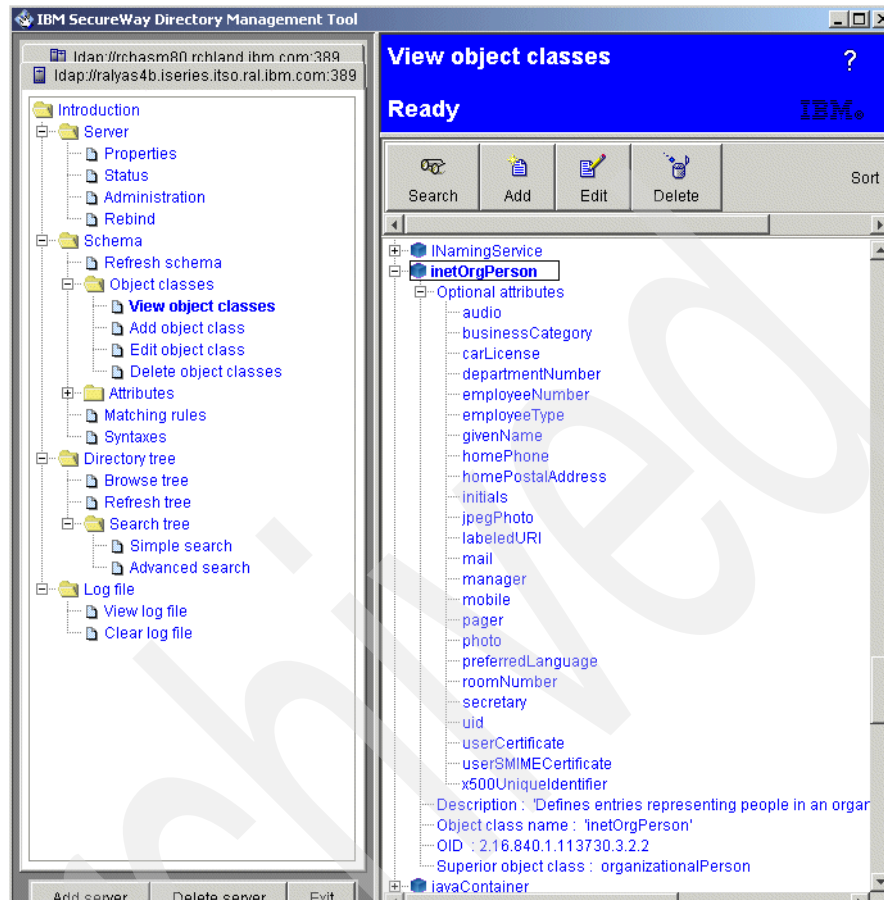


Figure 5-16 *inetOrgPerson* object class definition

As shown in Figure 5-16, the *inetOrgPerson* has only optional attributes. No required attributes are defined for this object class. A description of the object class is provided too. The OID is the object identifier that is unique for this object class. For further details on OIDs refer to Section 1.3 "Directory components" on page 11.

In addition you can see that the *inetOrgPerson*'s superior object class is *organizationalPerson*. This means that the *inetOrgPerson* has all attributes defined as optional attributes plus the attributes inherited from the *organizationalPerson* object class. You can then view the *organizationalPerson* object class to find out what attributes this class has defined and what its superior class is (the *Person* object class).

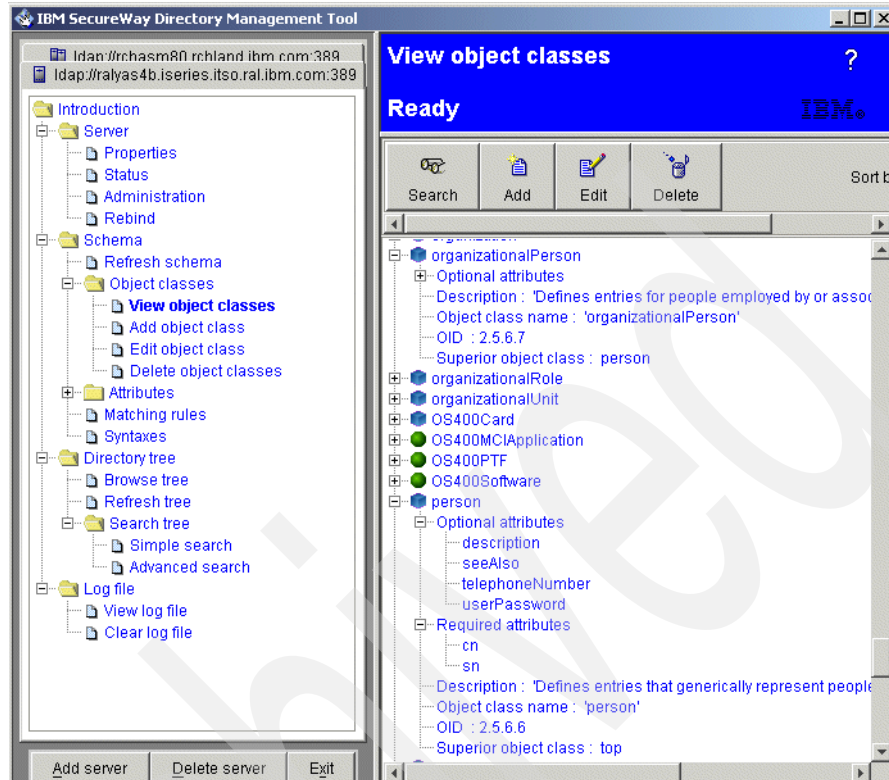


Figure 5-17 *organizationalPerson object class*

Figure 5-17 shows that the `organizationalPerson` inherits from `Person`, which inherits from `top`. In our example, the only required attributes are defined in the `Person` object class and are the `cn` (common name) and `sn` (last name) attributes.

All attributes are defined only once in a directory schema. This means that even if you see, for example, the `userPassword` attribute listed under the `ePerson` and the `Person` object class, it is defined only once in the schema and then referenced by individual object classes.

You can display attribute properties by expanding **Schema -> Attributes** and then clicking **View attributes**. Figure 5-18 on page 175 shows the attribute properties for the `userPassword` attribute.

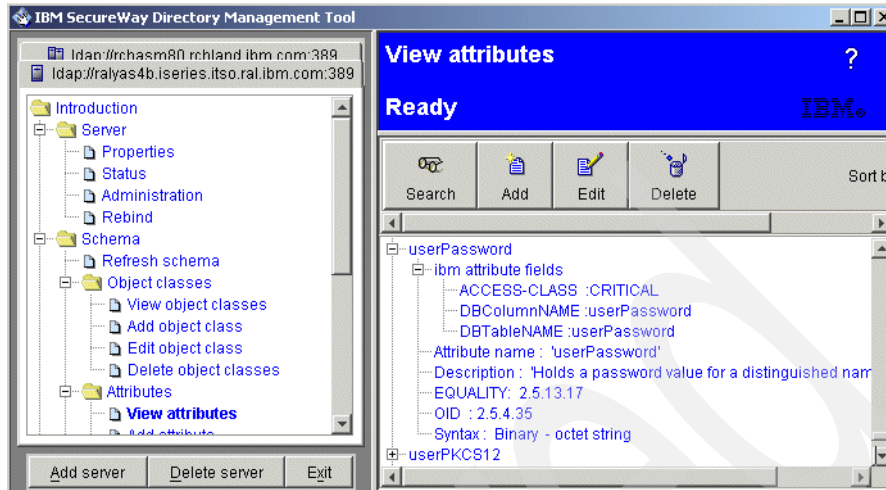


Figure 5-18 userPassword attribute properties

The properties of an attribute provide information about the required syntax, OIDs, access class, and so forth. For more information about the access class of an attribute refer to Section 5.7.1 "How does access control work?" on page 199.

The DMT allows you also to edit, add, or delete object classes and attributes by using the different options in the DMT navigation pane. The complete IBM SecureWay Directory schema can be found at:

<http://www.ibm.com/servers/eserver/iseries/ldap/schema/index.html>

## 5.2.6 Using the DMT to view the directory change log

The change log contains information about activities performed on the directory. It is not related to the security auditing support as described in Section 4.10 "Directory auditing support" on page 143, as it does not log, for example, search requests.

To view the change log, expand the Log file category in the navigation pane and then click **View log file**. The log file for the current server is displayed in the work area. The log displays the most recent log entry at the bottom of the window.

You can also clear the log by clicking **Clear log file**. Clearing the log file removes all log entries for the current server, leaving only a log message stating that the log was cleared.

More information about searching and setting up the change log can be found in Section 5.3.5 "Using LDAP utilities to view the directory change log" on page 185.

## 5.3 Using LDAP utilities to manage the directory

The iSeries Directory Services includes five utilities that allow you to perform actions on the LDAP directory server from the Qshell command environment in OS/400. These utilities use the LDAP APIs. You can use these utilities from the Qshell command line or call them from your programs. You may also find them useful as programming examples. When you install the Windows LDAP client that is included with Directory Services, you also install code that is very similar to the source code for the shell utilities.

The utilities are as follows:

- ▶ The `ldapadd` and `ldapmodify` utilities are used to add and modify LDAP directory entries.
- ▶ The `ldapdelete` utility removes entries from the LDAP directory.
- ▶ With the `ldapsearch` utility you can search the LDAP directory for entries.
- ▶ The `ldapmodrdn` utility allows you to change the relative distinguished name (RDN) of an entry.

Examples of how to use these utilities are covered in the following sections. For all utilities you can use the `-?` switch, for example, `ldapmodify -?`, to obtain command help.

### 5.3.1 The `ldapadd` and `ldapmodify` utilities

The `ldapmodify` utility allows you to change entries or add entries to the LDAP directory server from the QShell command shell on your iSeries server. It uses the `ldap_modify` application program interface (API). The `ldapadd` utility, which uses the `ldap_add` API, works identically to the `ldapmodify` utility with the exception that the `-a` flag is turned on automatically.

Usage:

```
ldapmodify [-a] [-r] -D binddn -w bindpassword [-f file] [dn...]
```

The `-a` (add) and `-r` (replace) switches specify the default behavior for the `ldapmodify` utility. If, for example, you want to add entries and do not want to specify the `changetype` parameter each time, you can use the `-a` switch that by default will then try to add all entries.

-D (bind distinguished name) and -w (password for the the bind DN) are typically required when performing updates to a directory. Make sure that the DN has the proper authority to perform the intended changes.

A typical way for providing input for the LDAP utilities is the use of input files. These files contain information in a special format, the LDAP Data Interchange Format (LDIF). The -f switch followed by the path and file name of the LDIF file is used to input data via a file. If you do not provide entry information from a file through the use of the -f option, the utility will wait to get entries from the keyboard. To break out of the wait press Enter until you receive the QShell prompt (\$) or press the **System Request** key and choose option 2 (end previous request).

**Tip:** Command help with all the possible switches and options is available by entering the ldapmodify command as follows:

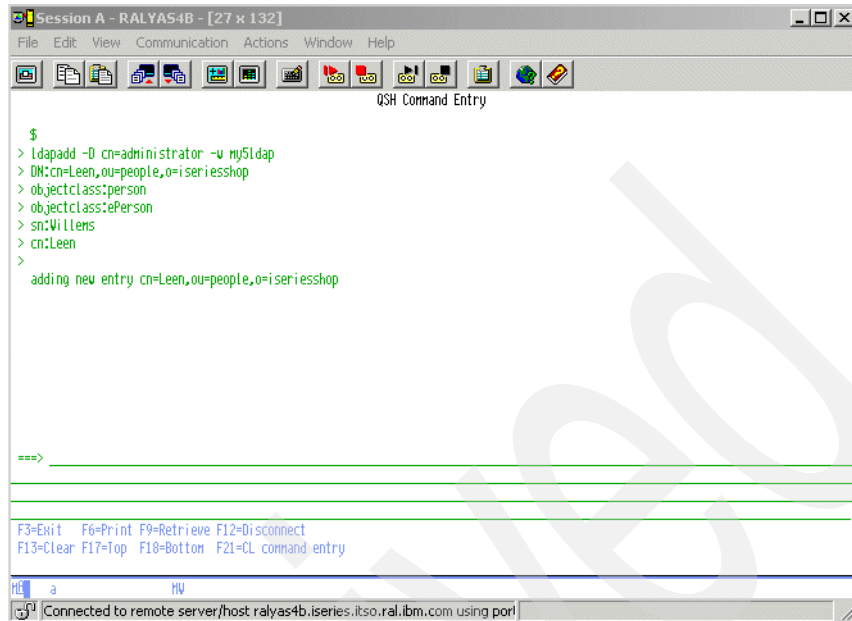
```
ldapmodify -?
```

### Adding an entry interactively using the QShell

The following example shows how to add a new person object entering the information interactively using the QShell. The new entry will be published under the directory path ou=people,o=iseriesshop. As the LDAP server is on the same iSeries server the command is performed on, you do not need to specify the -h (host name) switch. The entry is added using the ldapadd utility.

To add the entry you have to perform following steps:

1. Start the OS/400 Qshell with the command **QSH** on system RALYAS4B, as shown in Figure 5-19 on page 178.



The screenshot shows a QShell terminal window titled "Session A - RALYAS4B - [27 x 132]". The menu bar includes File, Edit, View, Communication, Actions, Window, and Help. The toolbar contains various icons for file operations and system functions. The terminal text shows the following sequence of commands and output:

```
$
> ldapadd -D cn=administrator -w my5ldap
> DN:cn=Leen,ou=people,o=iseriesshop
> objectclass:person
> objectclass:ePerson
> sn:Willems
> cn:Leen
>
  adding new entry cn=Leen,ou=people,o=iseriesshop

==>

F3=Exit  F6=Print  F9=Retrieve  F12=Disconnect
F13=Clear F17=Top  F18=Bottom  F21=CL command entry
```

At the bottom, a status bar indicates the user is logged in as 'a' on host 'HV' and is connected to the remote server 'host ralyas4b.iseries.itso.ral.ibm.com using port'.

Figure 5-19 QShell ldapadd utility

2. Enter `ldapadd -D cn=administrator -w password` and press Enter. The `ldapadd` utility goes into input mode and reads the data from the keyboard.
3. Enter `DN:cn=Leen,ou=people,o=iseriesshop` and press Enter.
4. Enter `objectclass:person` and press Enter.
5. Enter `objectclass:ePerson` and press Enter.
6. Enter `sn:Willems` and press Enter.
7. Enter `cn:Leen` and press Enter.
8. Press Enter to indicate the end of the input stream.

The message adding new entry `cn=Leen,ou=people,o=iseriesshop` as shown in Figure 5-19, is displayed confirming the addition of the new entry.



## Adding an entry using an LDIF file from the QShell

You can also use an LDIF file for input to the **ldapadd** or **ldapmodify** commands as shown in the next example where the same entry as described in "Adding an entry interactively using the QShell" on page 177 is added to the directory.

1. Create a text file to store the LDIF information for the new entry. In this example the file `/ldap/updates/NewEntr.ldif` on server RALYAS4B is created with the command:

```
EDTF STMF('/ldap/updates/NewEntry.ldif')
```

2. In the editor window enter the information as shown in Example 5-2.

*Example 5-2 LDIF file example for adding an entry*

---

```
*****Beginning of data*****
DN:cn=Leen,ou=people,o=iseriesshop
objectclass:person
objectclass:ePerson
sn:Willem
cn:Leen
*****End of Data*****
```

---

3. Save the file.
4. Use the following command on the QShell to add the new entry using a file for the command input:

```
ldapadd -D cn=administrator -w password -f /ldap/updates/NewEntry.ldif
```

**Tip:** The LDAP utility switches are case-sensitive, so `-f` is not equal to `-F`.

The Qshell responds with the following message:

```
adding new entry cn=Leen2,ou=people,o=iseriesshop
```

As an alternative to the `ldapadd` utility you could use the `ldapmodify` utility as shown in the following command to achieve the same result:

```
ldapmodify -a -D cn=administrator -w password -f /ldap/updates/NewEntry.ldif
```

The `ldapsearch` utility or the DMT, as shown in Figure 5-20 on page 180, can be used to verify the results.

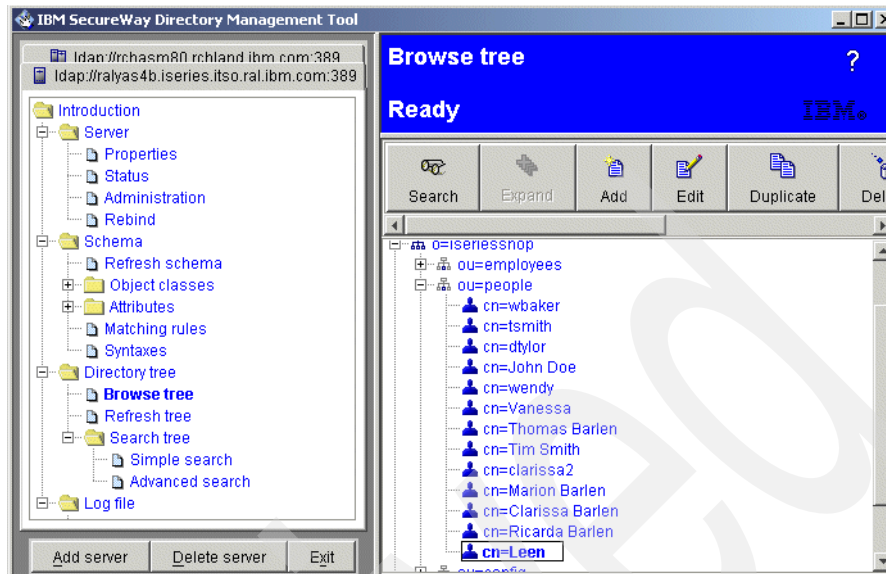


Figure 5-20 Verifying the existence of the new entry using the DMT

## Modifying an entry using an LDIF file from the QShell

As with adding an entry using an LDIF file for input to the `ldapmodify` utility, you can also use an LDIF file to perform updates to an existing entry. When using the LDIF example, as shown in Example 5-2 on page 179, you could simply update the entry by using the `-r` switch on the `ldapmodify` utility. The `-r` switch causes the `ldapmodify` utility to replace all entries found in the LDIF file in the directory.

The LDIF format allows you also to perform updates on certain attributes, add new attributes, and delete attributes, and this with just a single `ldapmodify` command. Example 5-3 shows an example of an LDIF file that performs multiple modifications.

### Example 5-3 LDIF file example of updating entries

```
version: 1

dn: cn=Ricarda Barlen,ou=employees,o=iseriesshop
changetype: modify
add: title
title: Senior staff member
-
replace: employeetype
employeetype: Permanent
-
replace: employeenumber
```

```
employeenumber: 55555553  
-  
delete: homepostaladdress
```

---

The update can be performed using the `ldapmodify` utility with the following switches:

```
ldapmodify -h ralyas4b -D cn=administrator -w password  
-f /as2318/ldif/update.ldif
```

This command string binds with DN, `cn=administrator` (-D), and its password, password (-w), to the LDAP server, `ralyas4b` (-h), and performs the modifications provided in the LDIF input file, (-f) `/as2318/ldif/update.ldif`. The LDIF file, as shown in Example 5-3 on page 180, performs an update for dn: `cn=Ricarda Barlen,ou=employees,o=iserieshop`. The change type is `modify`, indicating that the modification actions can consist of adding, replacing, or deleting attributes values. In this example, the title attribute is added, the `employeetype` and `employeenumber` attributes updated, and the `homepostaladdress` attribute deleted. Note the - character between the different modification entries. This character serves as a separation character and must be included.

For more information about the LDIF format refer to the LDIF RFC 2849 found at:

<http://www.rfc-editor.org/rfcsearch.html>

### 5.3.2 The `ldapsearch` utility

The `ldapsearch` utility is probably the most diverse utility from all five utilities. It can not only be used to just perform a simple search, it can also be used to export data into an LDIF file. This section shows you some examples of how to use the `ldapsearch` utility. All examples shown in this section access a directory as described in Figure 3-5 "iSeries Shop DIT" on page 50.

#### Example 1

In the first example you see how to search for all iSeries Shop employees that belong to the object class `Person`.

```
ldapsearch -h ralyas4b -b ou=employees,o=iserieshop objectclass=person
```

This command connects to the LDAP server on system `ralyas4b` (-h) and searches with the directory path (search base) `ou=employees,o=iserieshop` (-b) for entries of object class `person`.

**Important:** If one of your parameter values contains blanks, such as a DN of `cn=Clarissa Barlen,ou=employees,o=iseriesshop`, you need to enclose the value in double quotes. Taken the previous DN, this would mean that the DN parameter would look like:

```
-D "cn=Clarissa Barlen,ou=employees,o=iseriesshop"
```

The result is a list of all the entries that match the search criteria including all attributes of the entry, as shown in the Example 5-4 output.

*Example 5-4 ldapsearch example 1 output*

---

```
cn=WENDY,ou=employees,o=iseriesshop
objectclass=top
objectclass=person
objectclass=organizationalPerson
objectclass=inetOrgPerson
objectclass=publisher
objectclass=ePerson
cn=WENDY
sn=WENDY
uid=WENDY
description=Wendy
departmentnumber=ITS0
telephonenumber=1234-876-9876
mail=wendy@ISERIES.ITS0.RAL.ICM.COM
publishername=dc=RALYAS4B,dc=ISERIES,dc=ITS0,dc=RAL,dc=IBM,dc=COM
```

---

Example 5-4 shows just one example entry of the list of returned entries. As you can see, all filled-in attributes that can be accessed via anonymous access are shown.

## Example 2

In example 2 you are going to see how to bind to the LDAP server and search for a particular entry. In this case, the entry to be searched is the one shown in Example 5-4. But this time we are not interested in seeing all attributes of that entry, therefore a list of attributes to be returned is specified.

```
ldapsearch -h ralyas4b -D cn=administrator -w password
-b ou=employees,o=iseriesshop cn=wendy cn description mail homepostaladdress
```

This command string binds with DN, `cn=administrator` (-D), and its password, password (-w), to the LDAP server, `ralyas4b` (-h), and searches within the search base `ou=employees,o=iseriesshop` (-b) for an entry that has a common name `cn=wendy`. The only attributes that should be returned are `cn`, `description`, `mail`, and `homepostaladdress`. Note that in example 1 the `homepostaladdress` was not returned as an attribute even if all attributes were returned by the search. The

reason for this is that the homepostaladdress attribute's security class, by default, does not allow access for anonymous searches. Since the search in example 2 was performed with the bind DN cn=administrator, which is the owner of the entry, the protected attribute was also returned. Example 5-5 shows the result of the search request.

*Example 5-5 ldapsearch example 2 output*

---

```
cn=WENDY,ou=employees,o=iserieshop
cn=WENDY
description=Wendy
mail=wendy@ISERIES.ITSO.RAL.ICM.COM
homepostaladdress=Australia
```

---

### Example 3

In the last example for the ldapsearch utility you see how to use the utility to export directory entries into a file. The entries are stored in LDIF format allowing easy import to another directory. In this example, the ldapsearch exports all entries under the organization o=iserieshop, which includes all organizational units.

```
ldapsearch -h ralyas4b -D cn=administrator -w password -L -s sub
-b o=iserieshop objectclass=person >/as2318/ldif/export.ldif
```

This command string binds with DN, cn=administrator (-D), and its password, password (-w), to the LDAP server, ralyas4b (-h), and searches in all subtree levels (-s sub) within the search base o=iserieshop (-b) for entries of an object class person. According to the entries in the DIT, the command searches in the subtree RDNs ou=employees, ou=config, ou=people. The search returns all attributes in LDIF format (-L) and stores the output in file /as2318/ldif/export.ldif. Example 5-6 shows an entry in LDIF format as returned by the search request and stored in the file.

*Example 5-6 ldapsearch example 3 output*

---

```
Edit File: /as2318/ldif/export.ldif
Record :      56  of      294 by 10
Control :
CMD
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+..

dn: cn=WENDY,ou=employees,o=iserieshop
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: publisher
objectclass: ePerson
cn: WENDY
```

```
sn: WENDY
uid: WENDY
description: Wendy
departmentnumber: ITS0
telephonenumber: 1234-876-9876
mail: wendy@ISERIES.ITS0.RAL.ICM.COM
publishername: dc=RALYAS4B,dc=ISERIES,dc=ITS0,dc=RAL,dc=IBM,dc=COM
homepostaladdress: Australia
F2=Save F3=Save/Exit F12=Exit F15=Services F16=Repeat find F17=Repeat
```

---

### 5.3.3 The ldapdelete utility

The ldapdelete utility allows you to delete entries from the directory provided that the DN you are using to bind to the directory has the authority to perform the delete operation. As the other LDAP utilities, the ldapdelete utility takes input from the user's keyboard or from an LDIF input file. This sections shows examples for both input methods.

The first example deletes the entry with the DN cn=Leen, ou=people,o=iseriesshop. The input is provided over the user's keyboard.

```
ldapdelete -D cn=administrator -w password cn=Leen,ou=people,o=iseriesshop
```

The message deleting entry cn=Leen,ou=people,o=iseriesshop is shown indicating that the entry has been deleted.

If you want to do a bulk delete of several entries you may want to use an LDIF file for input to the ldapdelete utility as shown in Example 5-7.

*Example 5-7 LDIF file example for ldapdelete (delete.ldif)*

---

```
CN=leen,ou=people,o=iseriesshop

CN=sparky,ou=people,o=iseriesshop

CN=edwin,ou=people,o=iseriesshop
```

---

Note that the LDIF file contains only the distinguished names of the entries to be deleted. No attributes are required. The command to delete entries using the file is as follows:

```
ldapdelete -D cn=administrator -w password -f /as2318/ldif/delete.ldif
```

### 5.3.4 The ldapmodrdn utility

The ldapmodrdn utility allows you to change the relative distinguished name (RDN) of entries in the LDAP directory. This can be compared to a rename operation of an entry.

You can also perform the changes via input from the keyboard or through a file. The following command changes the DN cn=leen,ou=people,o=iseriesshop to cn=leen new,ou=people,o=iseriesshop by changing the RDN.

```
ldapmodrdn -D cn=adminstrator -w my5ldap  
-r cn=leen,ou=people,o=iseriesshop cn="leen new"
```

The -r switch removes the old RDN. Note that the new RDN contains two words and therefore must be placed in double-quotes.

You can also create a file that contains several RDNs to be changed. They have to be specified in pairs, as shown in Example 5-8.

*Example 5-8 modrdn example*

---

```
cn=leen,ou=people,o=iseriesshop  
cn=leen new  
cn=sparky,ou=people,o=iseriesshop  
cn=sparky new
```

---

When using a file as input for the ldapmodrdn utilities, you do not need to put multiple word values in double-quotes.

Note that you can also use the ldapmodify utility to change the RDN of an entry.

### 5.3.5 Using LDAP utilities to view the directory change log

You may want to track changes and activities, such as replication, to your LDAP directory. You can use the LDAP directory's change log to keep track of changes to the directory.

The change log is located under the special suffix cn=changelog. It is stored in the QUSRDIRCL library.

To enable the change log, follow these steps:

1. Start Operations Navigator and expand your system.
2. Expand **Network** and then **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties** to display the Directory Services properties.

5. Select the **Database/Suffixes** tab.

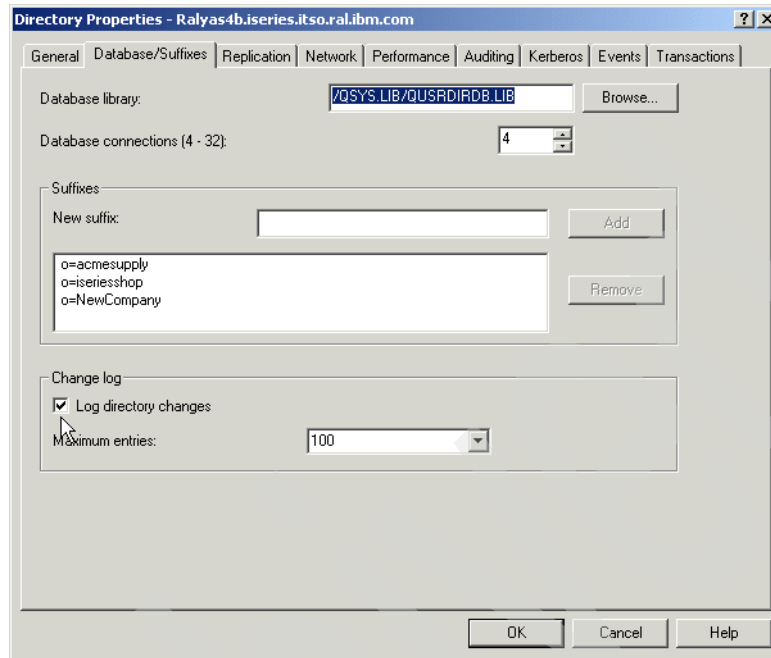


Figure 5-21 Enabling the change log

6. In the Change log section check the **Log directory changes** box to enable logging.
7. In the Maximum entries field specify the maximum number of entries for the change log to keep (this is optional).

**Note:** You should strongly consider specifying a number of maximum entries. If you do not specify a maximum number of entries, the change log will keep all entries and may become very large.

The `changeLogEntry` object class is used to represent the changes applied to the directory server. The set of changes is given by the ordered set of all entries within the changelog container as defined by `changeNumber`. The change log information is *read-only*. Any user who is on the access control list for the `cn=changelog` suffix can search on the entries in the change log. You should only perform searches on the change log suffix, `cn=changelog`.



**Tip:** Do not attempt to add, to change, or delete the change log suffix, even if you have authority to do so. This will cause unpredictable results.

Figure 5-22 shows the use of the `ldapsearch` command-line utility to retrieve all change log entries logged on the server:

```
ldapsearch -h ldaphost -D cn=adminstrator -w yourpassword  
-b "cn=changelog" "(changetype=*)"
```

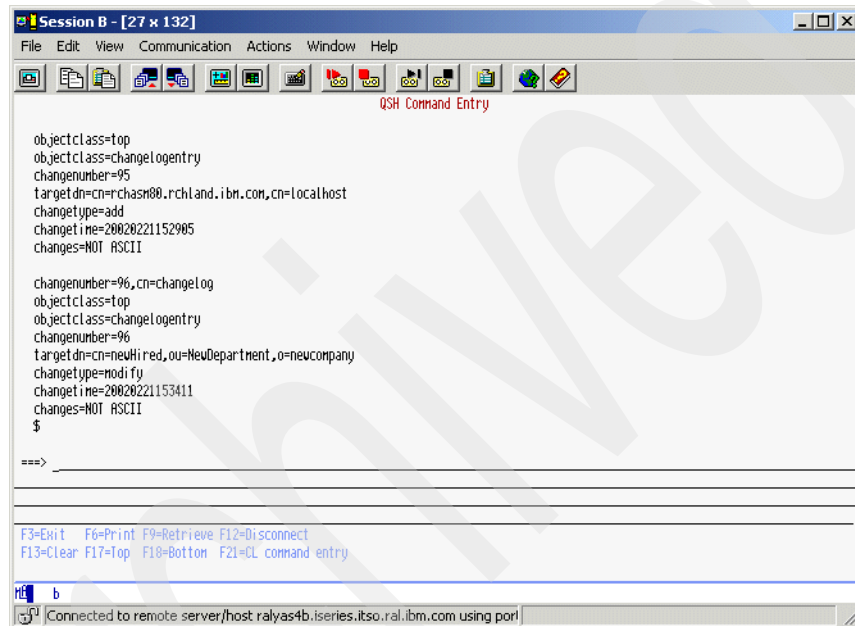


Figure 5-22 Searching the change log using `ldapsearch`

## 5.4 Exporting and importing information via Operations Navigator

The LDAP database can be distributed across multiple platforms by using LDAP Data Interchange Format (LDIF) files. On the iSeries the Operations Navigator is used as a GUI interface for exporting or importing LDIF files. LDIF files are stream files and should be transferred in ASCII format when using FTP.

## 5.4.1 Exporting directory data

Operations Navigator allows you to export the LDAP database to an LDIF file. You can export the entire directory with all entries or a specific subtree. However, whether you export the entire directory or a subtree, the export function always exports the entries with all attributes. You have no way of specifying that you only want to export a subset of attributes for the entries to be exported. You also cannot specify a search criteria to export only certain entries within the selected subtree. The Operations Navigator export function is designed, rather, to allow you to export the directory for backup or replication purposes. For more information about exporting entries that meet certain criteria into LDIF files, refer to Section 5.3.2 "The ldapsearch utility" on page 181.

To export the LDAP database on the iSeries server to an LDIF file you have to perform the following tasks:

1. Launch the Operations Navigator.
2. Expand the **System RALYAS4B** from which you want to export the LDAP database.
3. Expand **Network** and then **Servers**.
4. Click **TCP/IP**.
5. Right-click **Directory** and select **Tools**.
6. Click **Export File**.

The Export Directory to LDIF File window opens as shown in Figure 5-23.

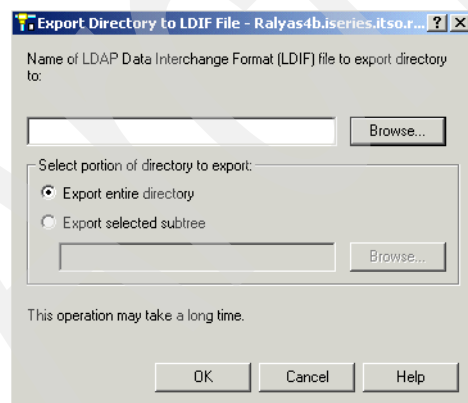


Figure 5-23 Export Directory to LDIF File window

7. In the Name of LDAP Data Interchange Format (LDIF) file to export directory to field you can enter the filename of the export file that you want to use or

you can click **Browse**. This opens a window where you can browse the iSeries Intergrated Files System (IFS).

The two radio buttons let you specify the part of the directory that you want to export to a file:

- Export entire directory

Specifies to export the entire directory to the specified file including all entries with all their attributes.

- Export selected subtree

Specifies the distinguished name (DN) of the subtree to export. This could be a suffix or a lower level object (for example, an organizational unit). If the directory server is started, you can click Browse to view the directory tree. You can then select a DN from the directory tree as shown in Figure 5-24 on page 190.

**Tip:** While the LDIF file is exporting, you must prevent the server from being updated. For information on how to disable updating, refer to a section "Moving LDAP directory data from the master to the replica" on page 111.

8. In this example, we export a subtree. Therefore select **Export selected subtree** and click **Browse**. When prompted to authenticate, use the LDAP directory administrator DN and password.

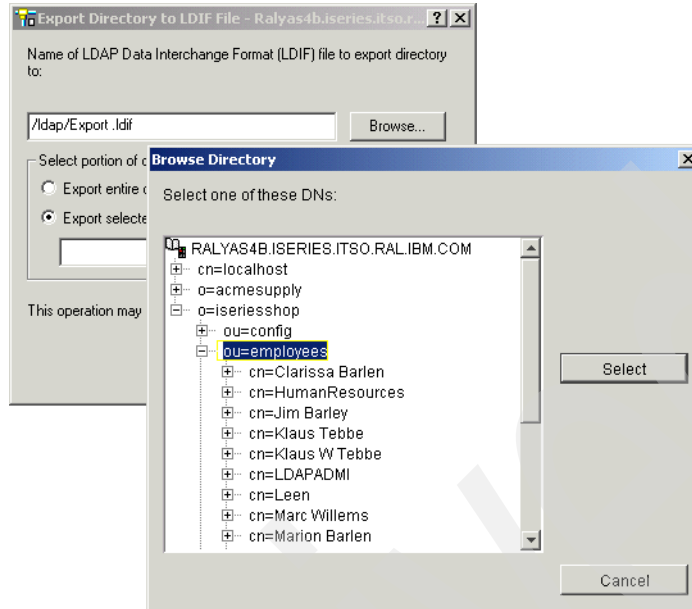


Figure 5-24 Browse Directory window

9. Select the subtree you want to export and click **Select**. In this example, all entries under the subtree `ou=employees,o=iseriesshop` are exported.

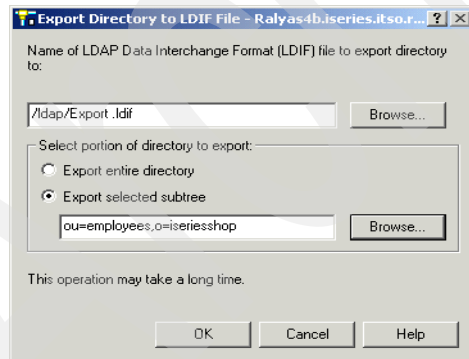


Figure 5-25 Export Directory to LDIF File window

The export options are displayed. As shown in Figure 5-25, the subtree `ou=employees,o=iseriesshop` of the directory will be exported to the `/ldap/Export.ldif` file in the iSeries IFS.

10. Click **OK** to start the export process.

The LDIF Export in Progress window opens.

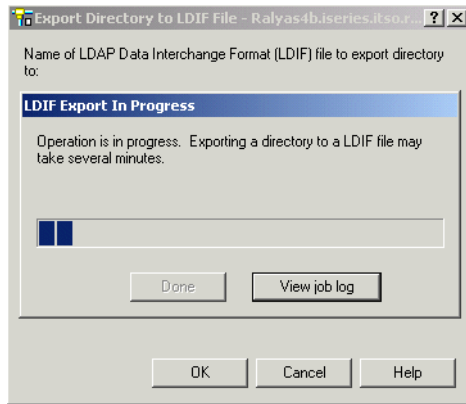


Figure 5-26 LDIF Export in Progress window

11. When the export completes successfully click **Done** to continue, otherwise click **View Job log** to open the joblog from the export LDIF file job.

You can browse the LDIF file using the Operations Navigator or with the OS/400 command:

```
EDTF STMF('/LDAP/Export.ldif')
```

An example of the exported file is shown in Figure 5-27 on page 192.

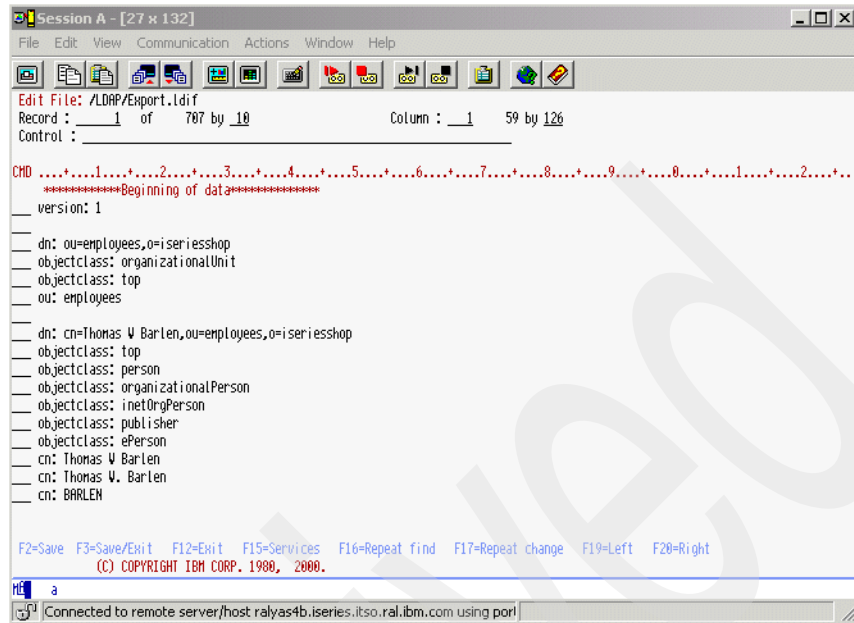


Figure 5-27 LDIF file editor window

**Tip:** If you want to browse or edit this LDIF file from your Windows workstation you must use Wordpad and *not* Notepad as Notepad is not able to interpret control characters correctly.

## 5.4.2 Importing directory data

Before importing data into the iSeries LDAP directory make sure that the proper suffixes are defined and the parent subtree structure exists. Using the import support via Operations Navigator requires that the LDAP directory server is stopped. When high availability is an issue and you cannot afford to stop the directory server, you can also use the `ldapadd` and `ldapmodify` utilities to import the data from your LDIF source file. Refer to Section 5.3.1 "The `ldapadd` and `ldapmodify` utilities" on page 176 for information on how to import data using the LDAP utilities.

Perform the following steps to import an LDIF file into the iSeries LDAP directory using Operations Navigator:

1. If the LDAP directory is started, stop it using one of the following methods:
  - a. Use Operations Navigator by right-clicking the Directory TCP/IP server application and selecting **Stop**.

- b. Use the OS/400 command **ENDTCPSVR SERVER(\*DIRSRV)**.
2. If not already started, launch the Operations Navigator.
3. Expand the **System RALYAS4B** on which you want to import the LDAP information.
4. Expand **Network** and then **Servers**.
5. Click **TCP/IP**.
6. Right-click **Directory** and select **Tools**.
7. Click **Import File**. The Import LDIF File window opens as shown in Figure 5-28.

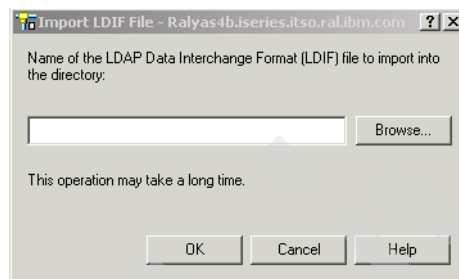


Figure 5-28 Import LDIF File window

8. Enter the name and the path of the LDIF source file in the Name of LDAP Data Interchange Format (LDIF) file to import into the directory field. You can also click **Browse** to browse the iSeries IFS and then select the LDIF file you want to import.
9. After the file is selected, click **OK** to start the LDIF import process. The LDIF Import Progress window appears.

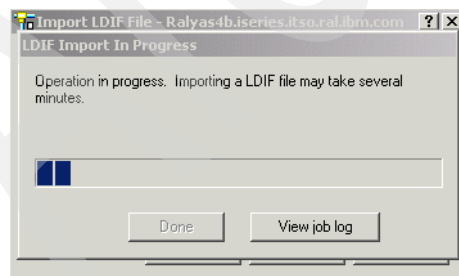


Figure 5-29 Import Directory In Progress window

When the import has finished you receive a completion message, as shown in Figure 5-30 on page 194.

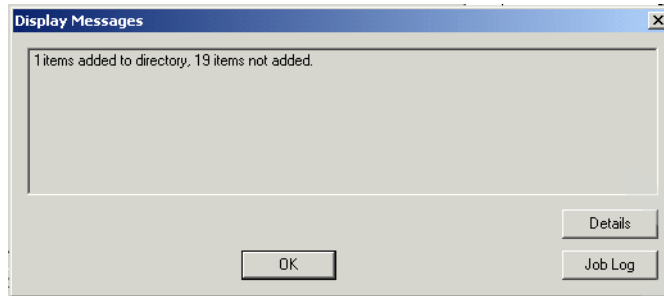


Figure 5-30 Display Messages window

The message provides information about how many entries were imported and for how many entries the import failed.

10. In the case of entries that were not added, click **Details** to open the Detailed Message Information window, as shown in Figure 5-31.

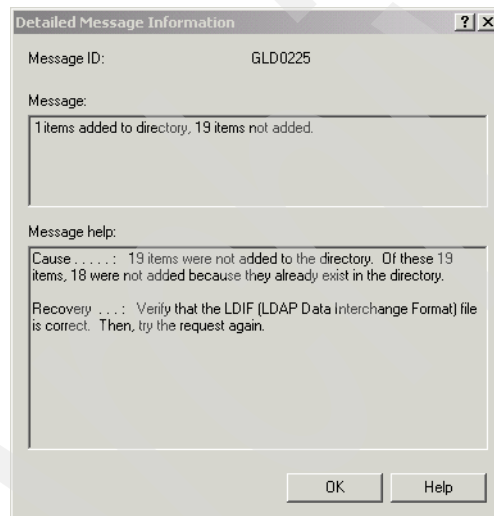


Figure 5-31 Detailed Message Information window

In this example one entry was successfully added but nineteen were not added because they already existed in the directory.

11. To obtain more information about errors that occurred during the import process, you can also click **Job Log** to see the complete job log of the LDIF import job.

This completes the import of directory information via Operations Navigator.



## 5.5 Writing your own application to manage your directory

If you want to write your own application to manage your LDAP directory, you can do so by using Directory Services Application Programming Interfaces (APIs) in Integrated Language Environment (ILE) programming languages, such as ILE RPG or ILE C. For more information about Directory Services APIs and how to use them, refer to Chapter 11, “Using APIs to directory-enable your applications” on page 461.

You can also access and manage your directory from within Java applications. The Java Naming and Directory Interface (JNDI) allows you to access, search, read, and write to an LDAP directory. Refer to Chapter 12, “Using the JNDI to search and update the directory” on page 477 for more information about how to use the JNDI interface.

## 5.6 Accessing directory information

Accessing and using directory information is the sole purpose of keeping information in a directory. In previous sections in this chapter you have seen the tools to manage the directory and publish information into a directory. This section gives you an overview of some of the available methods and tools to access a directory and retrieve information.

### 5.6.1 Searching the directory with the ldapsearch utility

The ldapsearch utility allows you to search for entries in the LDAP directory from the OS/400 QShell command interface or via the Windows command prompt when the IBM SecureWay Directory client has been installed.

Several examples of how to use the ldapsearch utility can be found in Section 5.3.2 “The ldapsearch utility” on page 181.

### 5.6.2 Searching the directory from a browser

Some browsers allow you to search a directory and display the search results on the browser window (Netscape) or open another address book window (Microsoft Internet Explorer). This section shows you some examples of how to search a directory using the browser interface. The results are displayed using a Netscape browser.

You can search a directory by specifying a URL in a browser. The format is defined in RFC1959.

The general search format is:

```
ldap://hostname:port/dn?attributes?scope?filter
```

The parameters have the following meaning:

<b>hostname</b>	The IP address or host name of the LDAP server.
<b>port</b>	Identifies the port the LDAP server is listening on.
<b>dn</b>	Specifies the search base or the entire DN of an entry to be displayed.
<b>attributes</b>	Defines a list of attributes the LDAP server returns in the search results. If this parameter is not specified, all attributes are returned.
<b>scope</b>	Specifies the scope of your search. The choices for this parameter are: <ul style="list-style-type: none"><li><b>sub</b> Searches all subtrees under the search base including the search base itself as specified in the dn parameter.</li><li><b>one</b> Searches one level down of the search base specified in the dn parameter.</li><li><b>base</b> Searches only in the search base specified in the dn parameter. No subtrees are searched.</li></ul>
<b>filter</b>	Specifies the search filter. That is, the search criteria, such as certain object classes or people with a certain last name.

### Example 1

Using the URL in this example, all entries within the subtree `ou=employees,o=iseriesshop` of object class `person` and where the last name starts with `b` are displayed. This command only searches in the `ou=employees,o=iseriesshop` level.

```
ldap://ra1yas4b/ou=employees,o=iseriesshop??one?(&(objectclass=person)(sn=b*))
```

## Example 2

Using the next URL, all entries in and under the search base `o=iseriesshop` are displayed where the phone number contains a 5. Only the last name (`sn`), first name (`givenname`), and the phone number (`telephonenumber`) should be returned in the search results.

```
ldap://ralyas4b/o=iseriesshop?sn,givenname,telephonenumber?  
sub?(telephonenumber=*5*)
```

In this example we also use the wildcard character `*`, which is specified in front and after the search character `5`. This means that all entries that have somewhere in the phone number a `5` are returned. You can use the wildcard characters for any information you may want to find. The output of this example is shown in Figure 5-32.

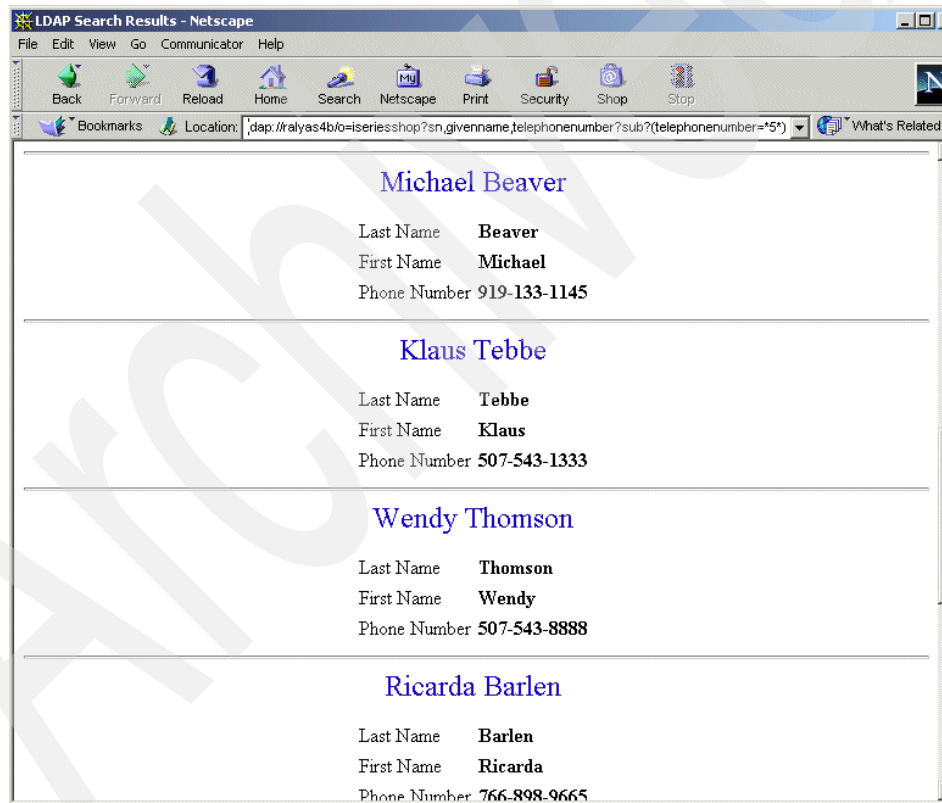


Figure 5-32 Searching the directory using a Netscape browser

### Example 3

The last example searches for entries under the search base `o=iseriesshop` in all subtrees. Only the last name (`sn`), e-mail address (`mail`), and phone number (`telephonenumber`) attributes should be returned. The entries have to belong to the object class `eperson` and the last name *or* the common name has to start with `do`.

```
ldap://ralyas4b/o=iseriesshop?sn,mail,telephonenumber?sub?(&(objectclass=eperson)(|(sn=do*)(cn=do*)))
```

**Note:** The LDAP server only returns attributes that contain a value and where anonymous access is allowed.

## 5.6.3 Searching the directory with applications

Many applications are already directory-enabled. As described in the redbook scenario in Chapter 3, “The redbook example scenario” on page 39, the iSeries Shop uses an LDAP directory for configuration and authentication purposes. For example, WebSphere Application Server and Domino use the LDAP directory for user authentication in a Single Sign-On environment; e-mail clients connect to the LDAP directory to look up e-mail addresses of other people; and the IBM HTTP Server for iSeries uses the directory for loading its configuration directives and for authentication. This redbook contains instructions on how to set up the applications to access an LDAP directory for many of these applications.

## 5.6.4 Searching the directory with your own applications

Applications Programming Interfaces (APIs) for ILE languages as well as a JDNI interface for Java applications is available to directory-enable your own applications. Refer to Section 5.5 “Writing your own application to manage your directory” on page 195 for more information about directory-enabling your own applications.

## 5.7 Controlling access to directory entries

In many cases, you probably would want to restrict access to data on your LDAP directory server. For example, an LDAP server on your company Intranet might contain a telephone directory of company employees. You would probably want all employees to be able to view the data in this directory. Imagine, however, that the president of your company does not want all employees to be able to access her telephone number. In that case, you could create an access control list (ACL). With this ACL you could restrict access to the server entry to only those employees the president wanted to receive calls from.

## 5.7.1 How does access control work?

With ACLs you can control who has the authority to add and delete directory objects. You can also specify whether or not users have the ability to read, write, search, and compare directory attributes. ACLs can be either inherited or explicit. That is, you can use ACLs in one of the following ways:

- ▶ Explicitly set up an ACL for a specific object.
- ▶ Specify that objects inherit ACLs from objects higher up in the LDAP directory hierarchy.

Perhaps the president of the iSeries Shop in the redbook scenario did not want all employees to be able to access her home telephone number. She did, however, want all managers to be able to access it. In such a case, you could make use of an ACL group to simplify granting authority to the managers. ACL groups allow you to grant access to specific groups of users rather than granting authority on an individual basis. This is particularly useful if the same group of people needs access to more than one set of objects. If the same managers that had access to the president's home telephone number, for example, later needed access to salary entries, you could reuse the ACL group. All versions of Directory Services support an access class level permissions model. Under this model, each LDAP attribute type has a classification of *Normal*, *Sensitive*, or *Critical*. The attribute schema files control these classifications. When you add a user to an object's ACL, you specify which classifications the user can read, write, search, and compare. In most schema, the home telephone number would be classified as a *Sensitive* attribute. Therefore, to give the managers in the above example access to the president's home telephone number, you would give them read access to the *Sensitive* attributes in the president's directory object. They would still not be able to access *Critical* information. All versions of Directory Services support setting access class level permissions.

Beginning with Version 5 Release 1, Directory Services also supports an attribute level permissions model. Under this model, you can specify read, write, search, and compare authorities for specific attributes, regardless of their access class. Consider again the above example. Under the attribute level permissions model, you could give the managers read access to the home telephone number attribute, even if they did not have access to *Sensitive* attributes in general.

The attribute level permission model is compatible only with SecureWay Directory Services Version 3.2 and above servers. By default it is not enabled. You have the option of enabling it when you work with ACLs. After it is enabled, the model can be disabled only by reconfiguring the server and restoring the directory database. Before you decide to enable this model, be aware that you will not be able to administer it from any LDAP V2 client (including pre-Version 5 Release 1 versions of Operations Navigator) and that attempting to do so may corrupt ACL entries.

## 5.7.2 Implementation tasks summary

Most of us probably understand why the iSeries Shop's president does not want to have anybody calling her at home. So obviously the decision has been made to allow only managers to see the president's home phone number. Within this section of the chapter you will see how to set up the necessary configuration to achieve this goal. The implementation is divided in to two phases. The following sections summarize the implementation tasks.

### Phase 1

In the first phase you will see how to set up the directory security to allow only managers to see the president's home phone number. The implementation steps described in this phase do not use attribute-level permissions as introduced with OS/400 Version 5 Release 1.

#### ***Advantages***

The advantage of this implementation is that you are backward compatible as you do not use attribute-level permissions. It is also simpler to implement.

#### ***Disadvantages***

One of the major drawbacks of this implementation is that you have to grant read and search access to all Sensitive attributes in the president's entry to managers. You cannot distinguish between the home postal address and the home phone number, which both have the sensitive classification.

Phase 1 implementation tasks:

- ▶ Create an access control list (ACL) group and add members to it.
- ▶ Add the ACL group to the ACL of the president's entry and grant the necessary permissions.

### Phase 2

In phase 2 of the implementation you are going to use attribute-level permissions which allow for a more granular and flexible authority model. The attribute-level permission model allows you grant access to individual attributes within a single entry or a group of entries.

#### ***Advantages***

The main advantage of this implementation is that managers will have access to *only* the home telephone number and no other attributes with a Sensitive classification.

### **Disadvantages**

Besides the restriction that you can only use IBM SecureWay Version 3.2 and higher client versions to administer the directory, there are no other disadvantages.

Phase 2 implementation tasks:

- ▶ Use Operations Navigator to enable attribute-level permissions.
- ▶ Remove the manager's permissions to all Sensitive attributes of the president's entry.
- ▶ Grant managers access to *only* the home telephone number attribute.

**Note:** Most of the implementation tasks to create an ACL group and assign members to it are described using Operations Navigator. You could do the same set up using the Directory Management Tool (DMT). However, using the DMT, you have to enter all values manually, as opposed to using the Operations Navigator where you have a very user-friendly GUI that allows you, for example, to browse for and select members to be added to an ACL group. You will appreciate the Operations Navigator interface even more when you have to enter many members in the form of a full-qualified DN to an ACL group. In this case, just selecting the members from a displayed list is much more convenient than entering all the information manually.

### **5.7.3 Creating an access control list (ACL) group**

To simplify management of access control lists of authorized users, access group objects can be specified in ACL entries (in place of a user). You can create access groups that can be specified as an object owner or used in access control lists.

Perform the following steps to create the access control list group for the iSeries Shop managers:

1. Launch the Operations Navigator.
2. Expand the **System RALYAS4B** and sign on when prompted.
3. Expand **Network** and then **Servers**.
4. Click **TCP/IP** to display the list of TCP/IP servers.
5. Right-click **Directory** and select **ACL Groups**.

If you are not already connected to the directory server, the Connect to Directory Server dialog appears. Connect to the server as administrator, as shown in Figure 5-33 on page 202.

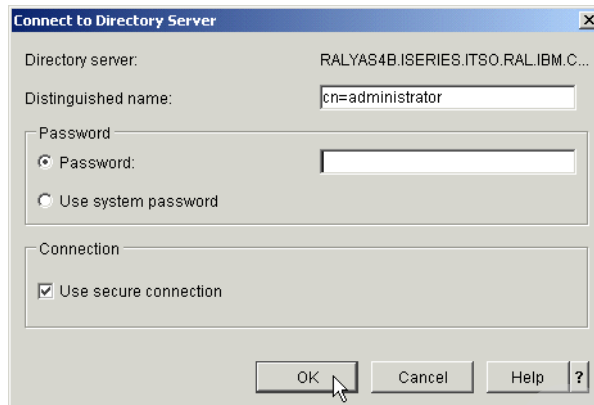


Figure 5-33 Connect to Directory Server window

6. Click **OK** to continue. A list of all defined ACL groups is displayed.

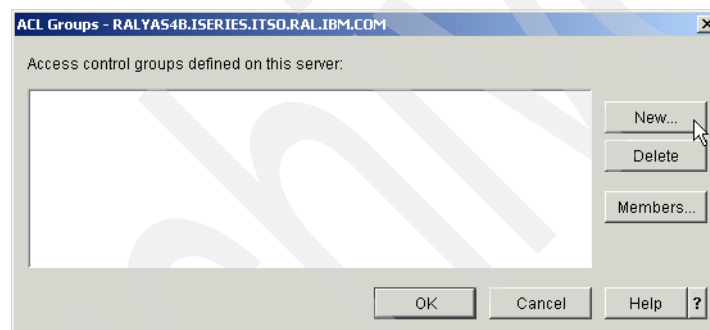


Figure 5-34 ACL Groups window

7. Click **New** to create a new ACL group. The New ACL Groups window appears, as shown in Figure 5-35 on page 203.



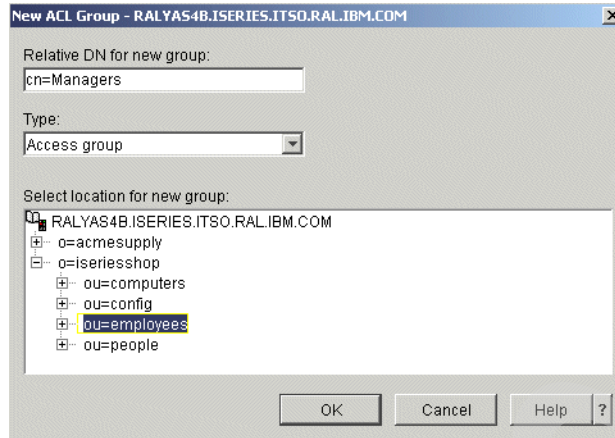


Figure 5-35 New ACL Group window

8. In the New ACL Group window you define the name, type, and location for the new group. Provide the following information:

**Relative DN for new group** Enter the name of the new ACL group as a relative distinguished name (RDN). For this scenario enter **cn=Managers**.

**Type** For Version 4 Release 4 and above, there are two types of ACL groups: access groups and roles. You use access groups to explicitly grant authority to objects. You use roles to track users who perform particular tasks. For our scenario select **Access group**.

**Select location for new group** Specifies the location for the new group. The RDN together with the location resolves into the DN of the ACL group. It is not required to place ACL groups into the directory path you want to protect entries in. It is up to you how to organize your directory. You may want to have a separate subtree that holds all ACL groups or, as in this scenario, place the ACL group into the subtree that holds the president's entry. Select **ou=employees,o=iseriesshop**.

9. Click **OK** to create the ACL group. The ACL Group members window appears, as shown Figure 5-36 on page 204.

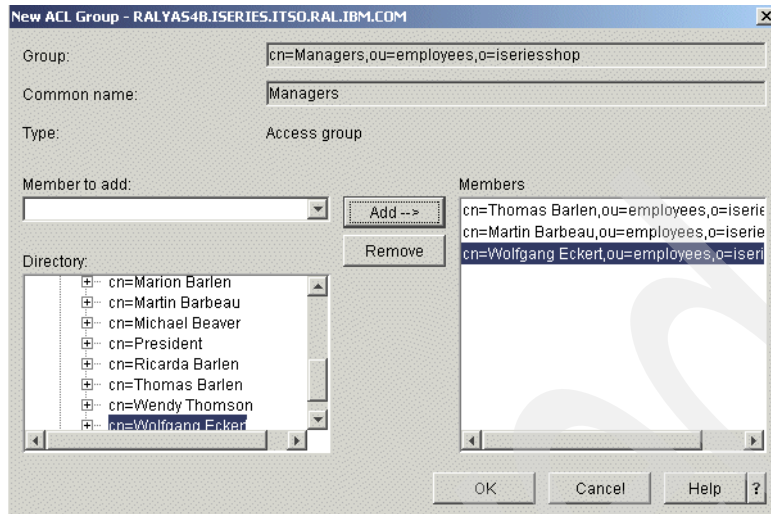


Figure 5-36 New ACL Group window - adding members

10. Select the entries that you want to be a member of the new ACL group. You do this by expanding the directory tree in the **Directory** pane, selecting the entries you want to add, and clicking **Add** to move the selected entries to the **Members** pane. In this scenario, three iSeries Shop managers are added to the `cn=Managers,ou=employees,o=iseriesshop` ACL group.
11. Click **OK** to save the new ACL group with its members.

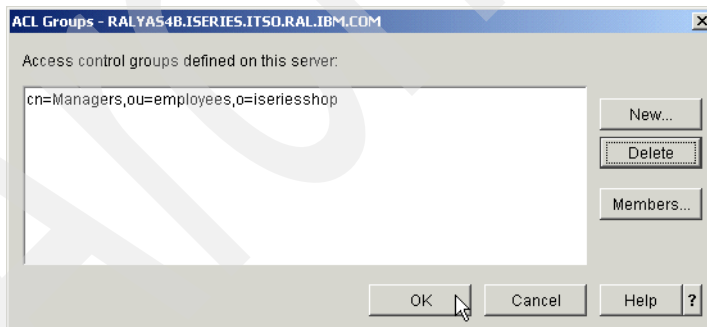


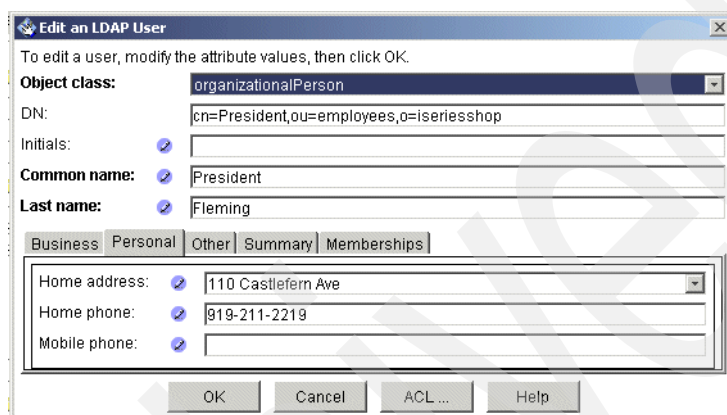
Figure 5-37 ACL Groups window with added ACL group

12. Click **OK** to close the ACL Groups window.

The ACL group for our managers is created. The next task assigns the new ACL group to the ACL of the president's directory entry.

## 5.7.4 Adding the ACL group to the entry's ACL

Before adding the new ACL group to the ACL of the president's directory entry, let us take a look at the directory entry and the attribute classification as defined in the SecureWay Directory schema. This information will help you understand the relationship between the attribute classifications in the directory schema and the ACL of an entry. Figure 5-38 shows the personal information of the president's entry in the directory using the DMT.



The screenshot shows a window titled "Edit an LDAP User" with a close button in the top right corner. Below the title bar is a message: "To edit a user, modify the attribute values, then click OK." The window contains several fields and tabs. The "Object class:" dropdown is set to "organizationalPerson". The "DN:" field contains "cn=President,ou=employees,o=iseriesshop". The "Initials:" field is empty with a checkmark icon. The "Common name:" field contains "President" with a checkmark icon. The "Last name:" field contains "Fleming" with a checkmark icon. Below these fields are five tabs: "Business", "Personal", "Other", "Summary", and "Memberships". The "Personal" tab is selected. Under the "Personal" tab, there are three fields: "Home address:" with a checkmark icon and a dropdown menu showing "110 Castlefern Ave"; "Home phone:" with a checkmark icon and a text field containing "919-211-2219"; and "Mobile phone:" with a checkmark icon and an empty text field. At the bottom of the window are four buttons: "OK", "Cancel", "ACL ...", and "Help".

Figure 5-38 President's directory entry - personal information

The homePhone and the homePostalAddress attributes contain information that not everybody should have access to. With phase 1 of the implementation as described in Section 5.7.2 "Implementation tasks summary" on page 200, object level permissions do not allow you to grant access to only one attribute that is classified as Sensitive. When granting access for managers to sensitive classified information in the president's entry, managers have access to all attributes that are classified Sensitive, not only the homePhone attribute. You can display the attribute classifications using the DMT. To do this you need to expand **Schema -> Attributes -> View** attributes, as shown in Figure 5-39 on page 206.

**Note:** The DMT version shipped with OS/400 Version 5 Release 1 translates many attribute names into a more meaningful description, as shown in Figure 5-38. The attribute name for the home address is actually homePostalAddress. Newer versions of the DMT (for example, Version 3.2.2) show both a meaningful text and the actual attribute name.

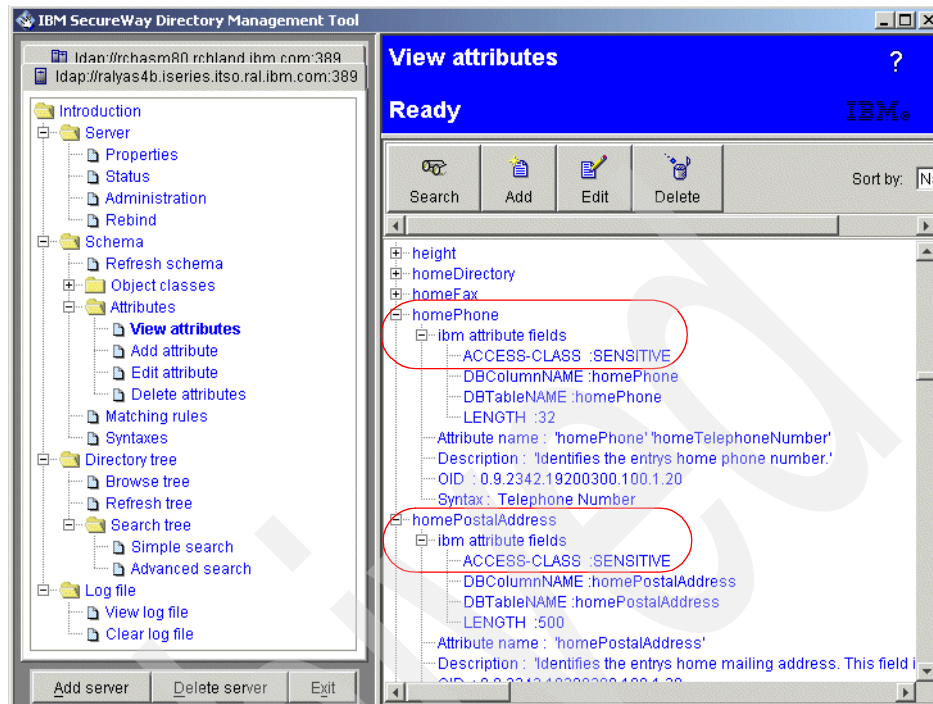


Figure 5-39 *homePhone and homePostalAddress attributes*

The following steps assign the managers' ACL group to the ACL of the president's entry and grant the necessary permissions to allow members of the ACL group to access the president's home telephone number. The entire configuration is performed using Operations Navigator. It is assumed that Operations Navigator is still started and expanded to the TCP/IP servers.

1. From the list of TCP/IP servers, right-click **Directory** and select **Authority**.
2. Expand the directory to find the entry where you want to connect the ACL group to.

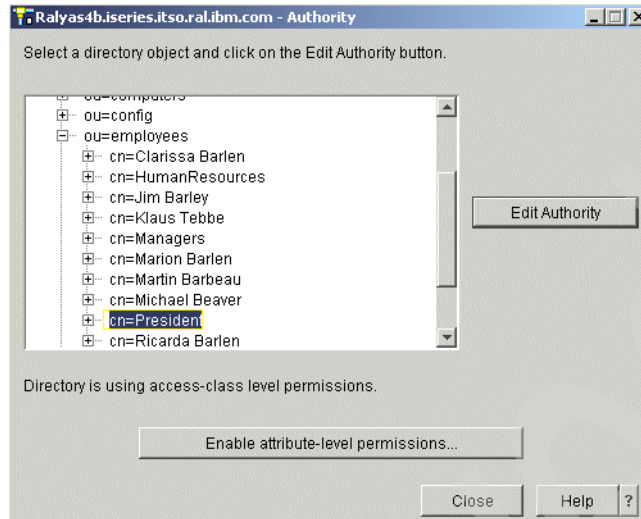


Figure 5-40 Authority window

3. According to our scenario, select **cn=President** and click **Edit Authority**.

The Edit Authority window opens. As we do not change the owner of this entry, click the **ACL** tab as shown in Figure 5-41.

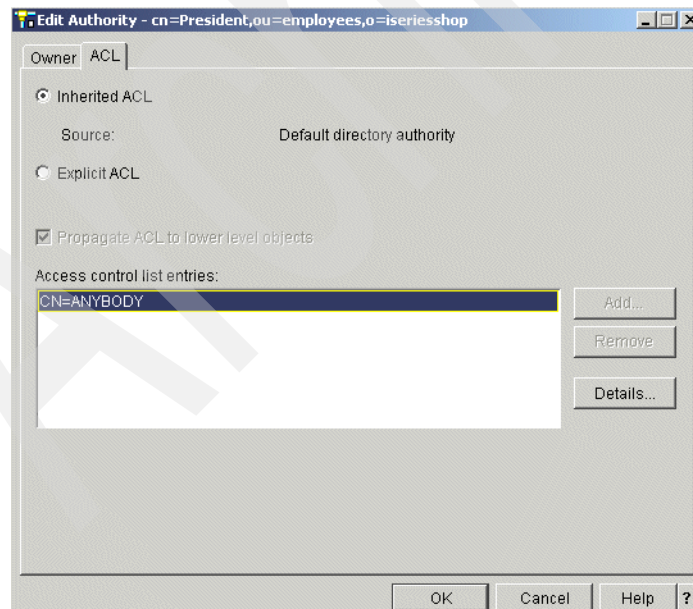


Figure 5-41 Edit Authority window

As you see in Figure 5-41 on page 207, there is already an access control list entry CN=ANYBODY. This is the ACL for anonymous connections to the directory server. You can view the details of this ACL group by selecting **CN=ANYBODY** and then clicking **Details...**. The ACL Entry Details window is shown.

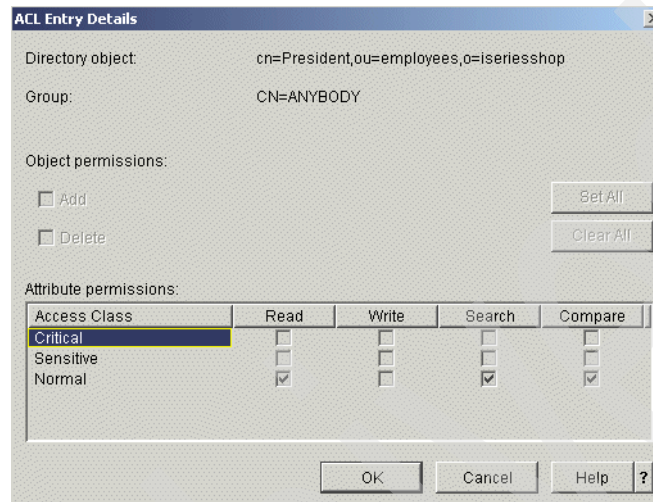


Figure 5-42 ACL Entry Details window for CN=ANYBODY

The details as shown in Figure 5-42 show what permissions anonymous users have on the President's entry. In this case, anonymous users cannot add or delete entries under the president entry. They have the following attribute permissions for Normal classified attributes:

- Read
- Search
- Compare

Anonymous users do not have access to Sensitive or Critical information and, thus, cannot view the home phone number or home postal address. Note that you cannot modify any settings as the entry, by default, inherits the ACL from the directory parent of the president entry. Click **OK** to return to the Edit Authority window.

4. From the Edit Authority window select **Explicit ACL** as shown in Figure 5-43 on page 209. With Explicit ACL selected, you can override ACL settings for the selected entry. This would also allow you to modify the ACL for anonymous users.

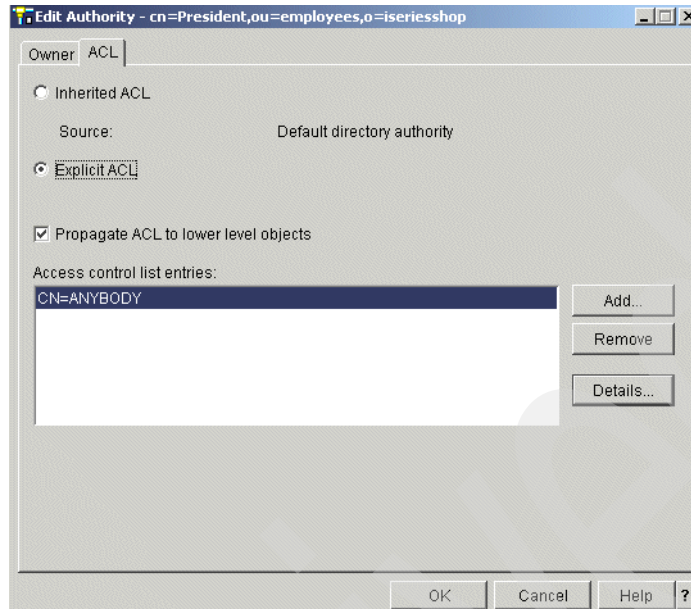


Figure 5-43 Edit Authority window

5. Click **Add** in the Edit Authority window to add a new ACL entry.

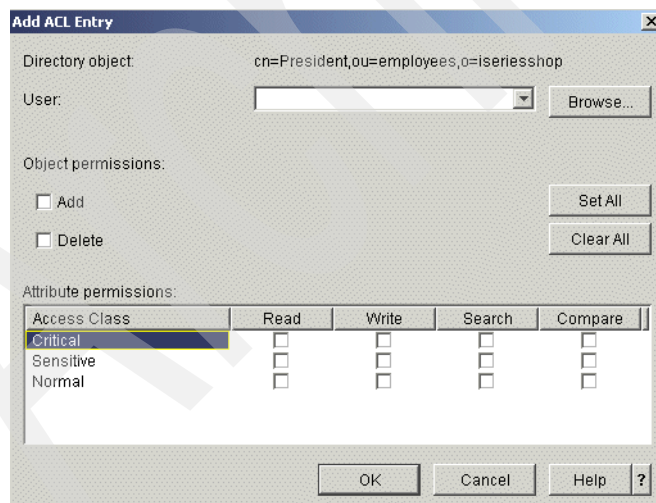


Figure 5-44 Add ACL Entry window

6. Click **Browse**. This option allows you to browse the directory and select the entry you want to add. With Operations Navigator, as described here, you can



browser your directory. When using the DMT, you need to manually add the DN of the entry to be added.

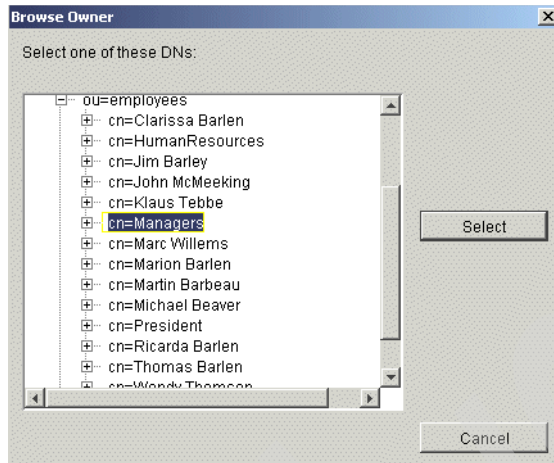


Figure 5-45 Edit Authority - Browse Owner window

Browse the tree until you found the ACL group entry you want to use.

7. Select **cn=Managers** and click **Select**.

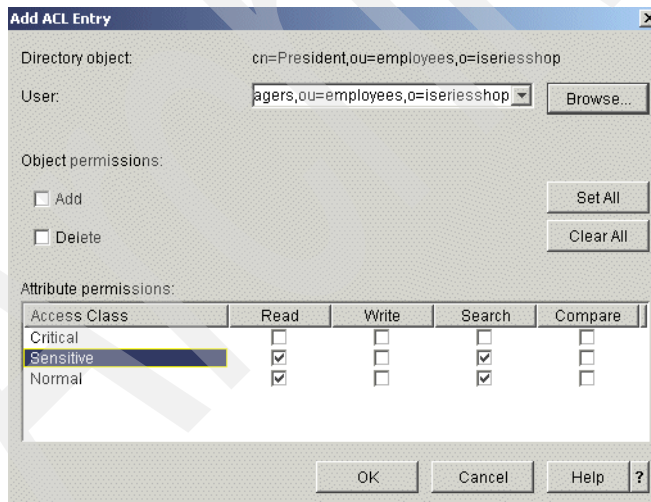


Figure 5-46 Add ACL Entry window

Specify the permissions (or rights) a user or group has to the directory object attributes. The attributes are the contents of the object.



8. In this scenario check **Read** and **Search** for the the Normal and Sensitive access classes.

**Tip:** The check boxes do not overwrite each other. For example, if you checked Critical Read, but not Normal Read you will *not* be able to read the Normal attributes from an entry. You also have to check Normal Read if you want to read also these attributes.

Possible values are:

<b>Read</b>	Specifies you can retrieve attribute values.
<b>Write</b>	Specifies you can modify attribute values in this access class.
<b>Search</b>	Specifies you can search for objects that have an attribute (for example, objects that have a telephone number), but cannot get the value without Read permission.
<b>Compare</b>	Specifies you can compare the attribute values (affects only the LDAP compare API).

**Tip:** If you have Read permission, you would probably want to have Search permission. If you have Read and Search permissions, you would probably want to have Compare permission. Likewise, if you have Compare permission, you would probably want to have Read and Search permissions.

Permissions to object attributes are based on the concept of access classes. The directory server defines the following access classes:

- Critical
- Sensitive
- Normal

The directory schema maps each attribute of an object class to an access class. For example, access to object data is done by first defining an object attribute to be Normal, and then defining a user's access to Normal attributes.

9. Click **OK** to save the object and attribute permissions.

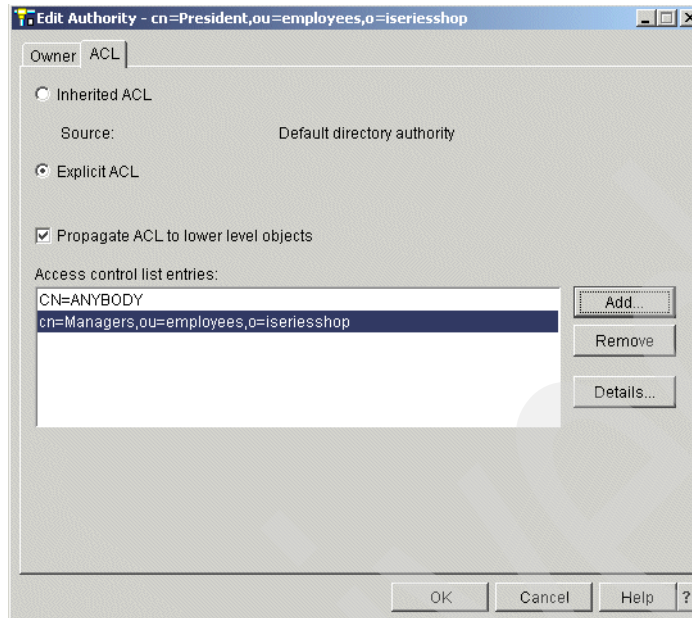


Figure 5-47 Edit Authority window with added ACL entry

10. Click **OK** to close the Edit Authority window.

## Verifying your configuration

The following steps are an example of how you can verify that your ACL setup works as desired:

1. From the OS/400 QShell enter the following command:

```
ldapsearch -b ou=employees,o=iserieshop cn=President
```

This command will search for common name cn=President, as shown in Figure 5-48 on page 213 and binds to the LDAP directory server as anonymous, because there is no bind (-D) switch defined in the command string.

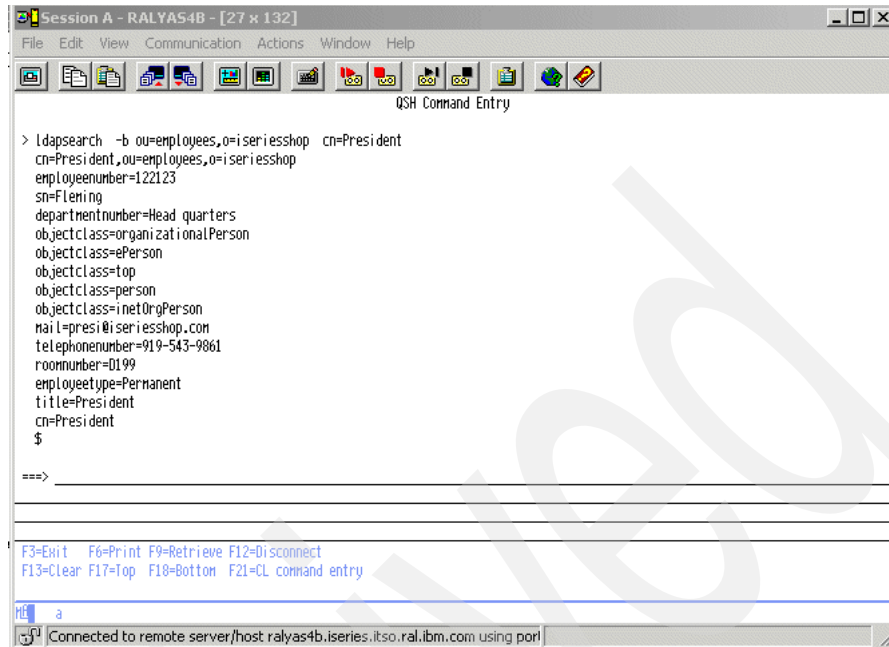


Figure 5-48 Verification without anonymous bind

The attributes homePhone and homePostalAddress are not shown because these two attributes have ACCESS-CLASS=SENSITIVE and anonymous users only have access to Normal attributes.

Now let us bind with a DN of one of the managers and check if he is authorized to view the Sensitive attributes.

2. Enter the following command in the OS/400 command shell:

```
ldapsearch -D "cn=Martin Barbeau,ou=employees,o=iseriesshop" -w password -b ou=employees,o=iseriesshop cn=President
```

This command searches for the common name cn=ThePresident starting in the directory tree at ou=employees,o=iseriesshop and binds to the LDAP directory server as "cn=Martin Barbeau,ou=employees,o=iseriesshop", as shown in Figure 5-49 on page 214.

**Tip for binding:** If you want to bind to the directory with a DN other than the administrator you must give the complete DN of that entry. If the DN contains any blanks you have to put the DN in double quotes as shown in this example.

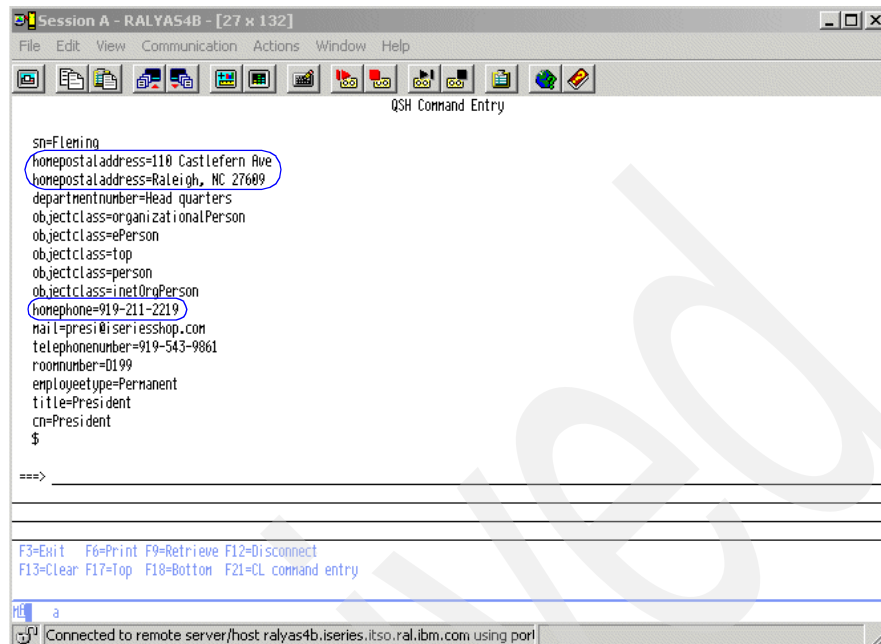


Figure 5-49 Verification with bind as a member of the new ACL group

The attributes homephone and homepostaladdress are shown to the user DN:cn=Martin Barbeau,ou=employees,o=iseriesshop, who is a member of the ACL group cn=Managers.

## Granting an ACL group access to a group of people

You have now seen how to grant an ACL group access to an individual directory entry. For the president's entry this is exactly what we tried to accomplish. But sometimes you might want to authorize a group, let us say the Human Resources Department, to manage entries within a directory subtree including adding and deleting entire entries. To implement this task, you can also create an ACL group and add it to an entry in the directory tree and the ACL will be propagated to the lower level objects in the tree. Following is a summary of steps to allow employees of the Human Resources Department to manage entries under the ou=employees,o=iseriesshop directory subtree.

1. Create an ACL group HumanResources as described in Section 5.7.3 "Creating an access control list (ACL) group" on page 201. The DN of the new ACL group is cn=HumanResources,ou=employees,o=iseriesshop.
2. Add all Human Resources Department members to the HumanResource ACL group.

After the ACL group HumanResources is created, you now have to connect this ACL group to all the iSeries Shop employees as described in the following steps:

3. In Operations Navigator expand **System -> Network -> Servers** and click **TCP/IP**.
4. Right-click **Directory** and select **Authority** to open the Authority window.
5. Expand the directory to **o=iserieshop**.
6. Under o=iserieshop select **ou=employees**. The ou=employees subtree holds all employee entries the Human Resources Department want to manage.

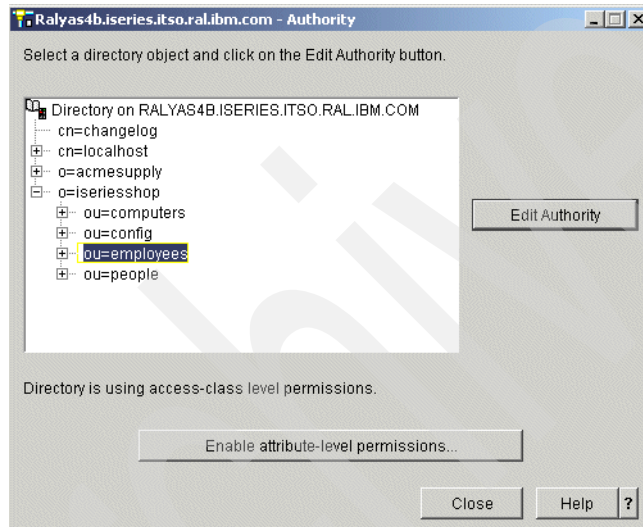


Figure 5-50 Authority window

7. Click **Edit Authority** to edit the ou=employees authority settings.
8. Click the **ACL** tab as shown in Figure 5-50.

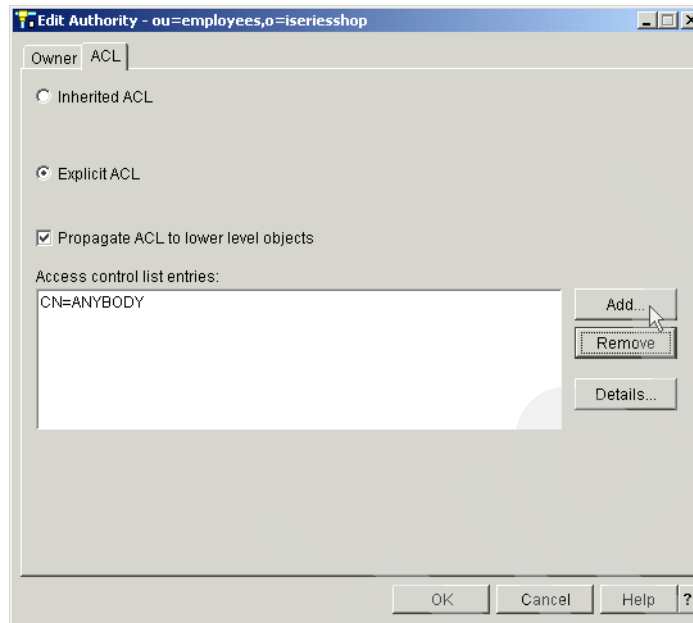


Figure 5-51 Edit Authority window

9. Click **Explicit ACL** and make sure that **Propagate ACL to lower level objects** is checked.

When this box is checked, objects beneath this one can inherit the access control list (ACL) from this object. When not checked, the ACL applies to only this object. Lower level objects that inherit their ACL will inherit them from an object above this one.

10. Click **Add** to add a new entry to the ACL list.
11. Click **Browse** to browse the directory and select the ACL group `cn=HumanResources,ou=employees,o=iseriesshop` that has previously been created.

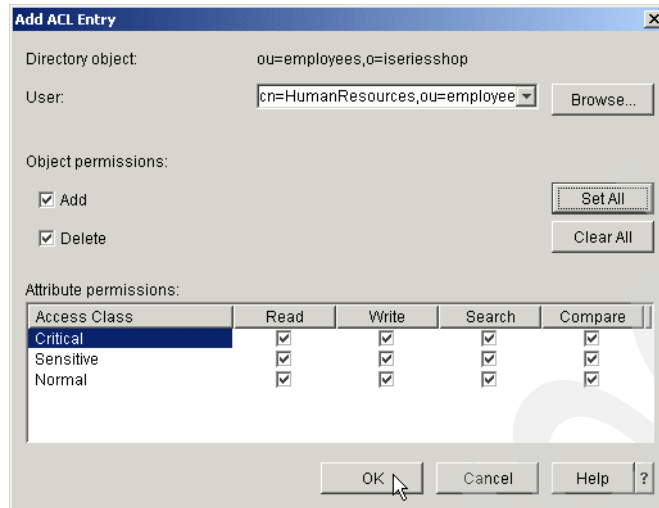


Figure 5-52 Add ACL Entry window

12. Click **Set All** to give the members of the Human Resources Department all the permissions by checking all the object and attributes permissions checkboxes.

Checking the object permissions Add and Delete checkboxes gives the users the right to add and delete entries beneath the ou=employees path.

13. Click **OK** to save the object and attribute permissions.
14. Click **Close** in the Authority window to save your configuration.

Now all members of the ACL group cn=HumanResources, ou=employees, o=iseriesshop have all the permissions to manage iSeries Shop employees.

### 5.7.5 Enabling the directory for attribute-level permissions

Attribute-level permissions are part of an access control list (ACL) model that is new for the LDAP Version 3.2 server. This model allows you to set attribute permissions for the individual attributes of directory objects. It is supported beginning with Version 5 Release 1 of Operations Navigator and Version 3.2 of the Directory Management Tool.

**Note the following before you enable attribute-level permissions:** Using an earlier version of Operations Navigator to administer the LDAP directory server may corrupt ACL entries. If you have applied one of the following PTFs to your earlier version of Client Access Express, you will not be in danger of corrupting ACL entries, but will still not be able to edit them:

- ▶ V4R4M0 Client Access Express: PTF 5769-XE1 SF63193
- ▶ V4R5M0 Client Access Express: PTF 5769-XE1 SF62213

Perform the following steps to enable attribute-level permissions for your LDAP directory:

1. Using Operations Navigator, expand the system your LDAP server runs on.
2. Expand **Network** and then **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Authority**.

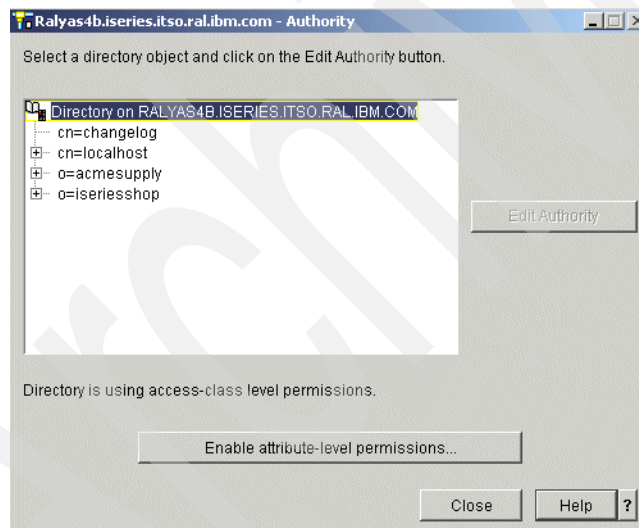


Figure 5-53 Authority window - Enable attribute-level permissions

**Important:** Proceeding with the next step enables attribute-level permissions. This is a one-time task. Once enabled, you can only disable attribute-level permissions by reconfiguring your directory server.

5. Click **Enable attribute-level permissions**. A confirmation message is shown.



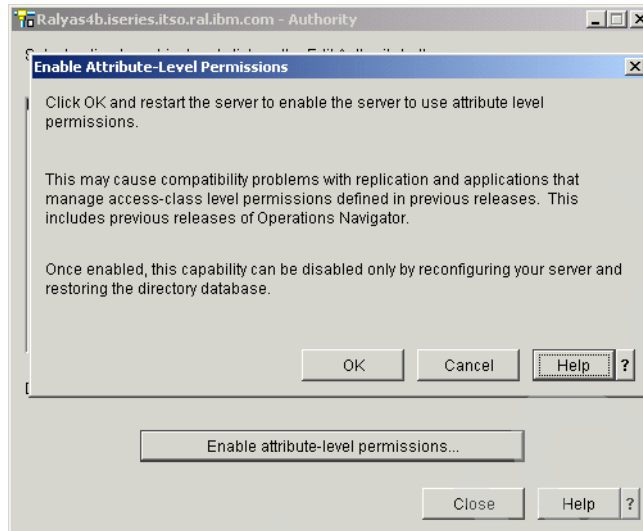


Figure 5-54 LDAP authority - Enable attribute permissions warning

6. Read the information and when ready click **OK**.
7. After attribute-level permissions are enabled, restart the Directory Services server to activate the changes.

### 5.7.6 Editing object and attribute authorities

You can use Operations Navigator or the Directory Management Tool to manage attribute permissions. The tasks shown in this section are performed with Operations Navigator. For information on how to perform the steps with the DMT, refer to the product documentation found at:

<http://www.ibm.com/software/network/directory/library/>

Following our redbook scenario implementation steps as described in Section 5.7.2 "Implementation tasks summary" on page 200, you need to remove the authority for sensitive information from the managers ACL group. After the permissions are revoked you will use the attribute-level permissions scheme to allow managers to view only the home phone number of the company's president. All other sensitive information can then not be accessed by managers anymore.

#### Removing access class level permissions from the ACL

Perform the following steps to revoke the access class permissions that were previously granted when no attribute-level permissions were activated:

1. If not already done so, use Operations Navigator and expand to TCP/IP servers.
2. Right-click **Directory** and select **Authority**.
3. Expand the directory tree to **o=iseriesshop** and then **ou=employees**.

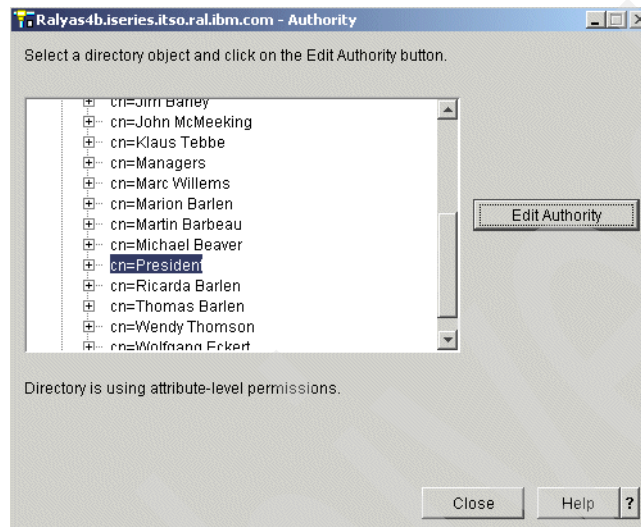


Figure 5-55 Authority window

4. Select **cn=President** and click **Edit Authority**.
5. On the Edit Authority window click the **ACL** tab. The current ACL list entries are displayed.

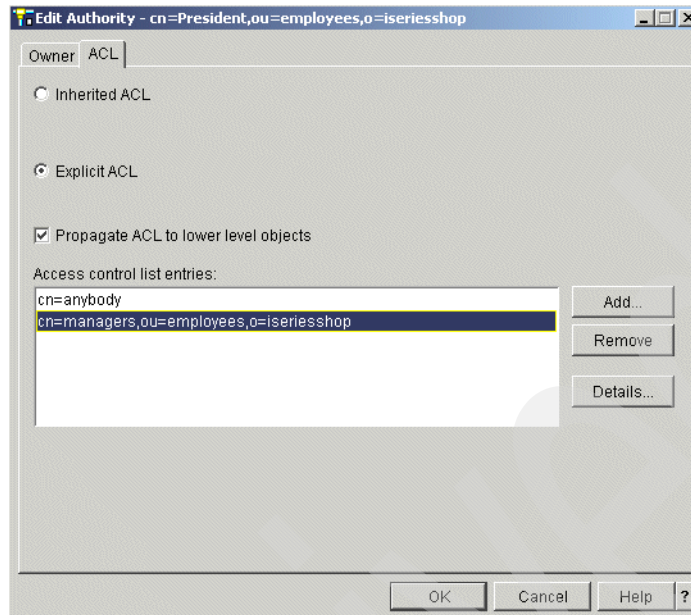


Figure 5-56 Edit Authority window - ACL tab

6. Select the ACL entry **cn=Managers,ou=employees,o=iserieshop** and click **Details**. The ACL Entry Details window is displayed. Note that after enabling the attribute-level permissions, the window layout of the Edit Authority window has changed. The new layout now contains two tabs, the Objects and Attributes tab.
7. Click the **Attributes** tab.

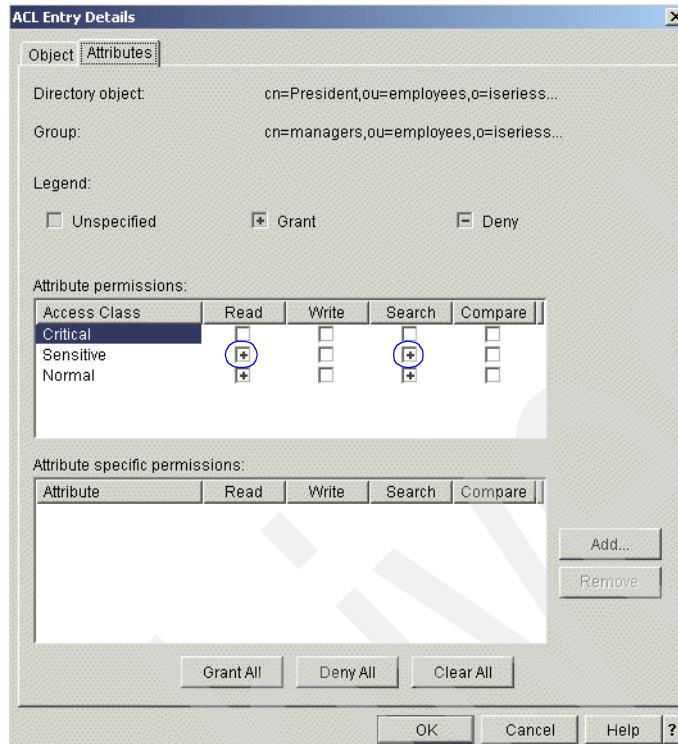


Figure 5-57 ACL Entry Details window - Attributes tab

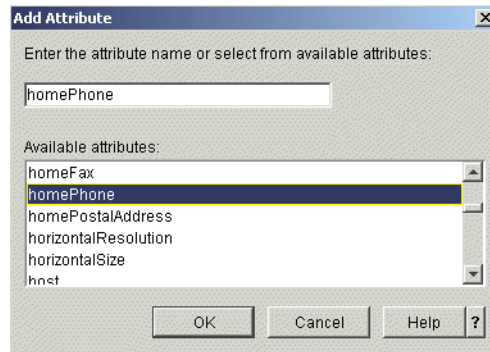
8. Click the **Read** and **Search** check boxes for the Sensitive access class. The + sign (Grant) will turn into a - sign (Deny) revoking access for Sensitive attributes.

Leave the ACL Entry Details window open and continue with the next steps.

### Granting attribute permission for the managers ACL group

The remaining steps of this task grant attribute-level access for the homePhone attribute to the managers ACL group.

9. In the Attribute specific permissions section of the window click **Add**. The Add Attribute window is displayed, as shown in Figure 5-58 on page 223.



*Figure 5-58 Add Attribute window*

10. Select the homePhone attribute from the list of all attributes defined in the directory schema.
11. Click **OK**. As shown in Figure 5-59 on page 224 the access class attribute permissions for sensitive information is set to - (Deny). The homePhone attribute has also been added.

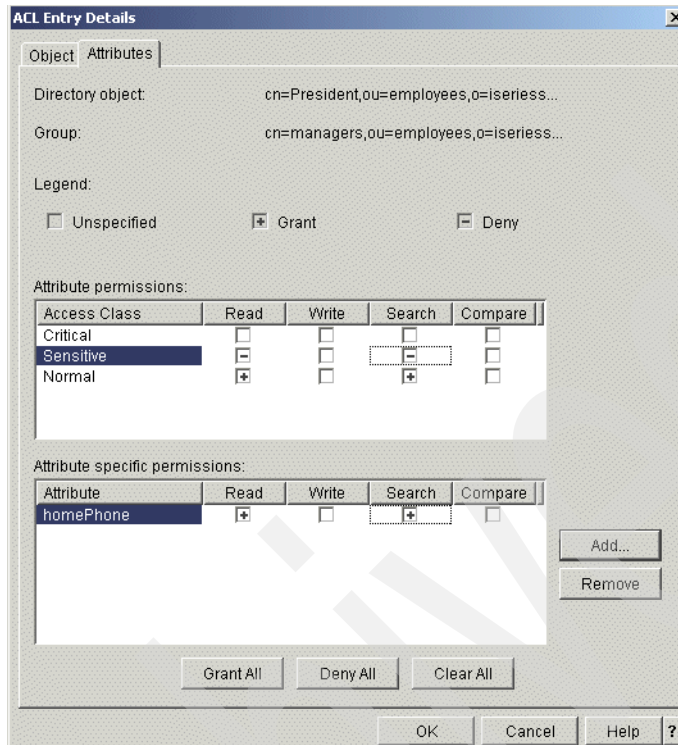


Figure 5-59 ACL Entry Details window with added attribute

12. Click **Read** and **Search** for the homePhone attribute, as shown in Figure 5-59. These settings grant the managers ACL group access to only the home phone number of the president. No other sensitive information can be accessed by the managers ACL group.
13. Click **OK** to save the permissions and return to the Edit Authority window.
14. Click **OK** to save the changes and return to the Authority window.
15. Click **Close** to close the Authority window.

## Verifying the permissions

The following steps are an example of how you can verify that your ACL setup works as desired:

1. From the OS/400 QShell enter the following command:
 

```
ldapsearch -D "cn=Martin Barbeau,ou=employees,o=iseriesshop" -w password -b ou=employees,o=iseriesshop cn=President
```

This command searches for the common name cn=ThePresident starting in the directory tree at ou=employees,o=iseriesshop and binds to the LDAP directory server as “cn=Martin Barbeau,ou=employees,o=iseriesshop”, as shown in Figure 5-60.

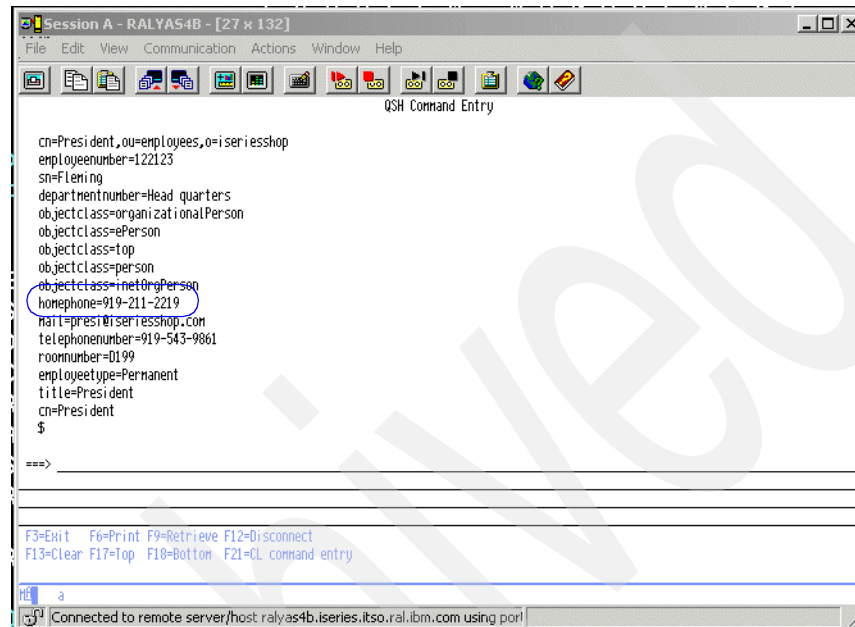


Figure 5-60 Verification of attribute-level permissions

Only the home phone number (homePhone attribute) of the president is shown to the member of the managers ACL group. The home postal address as shown in Figure 5-49 on page 214 does not appear anymore.

With attribute-level permissions you can fine tune directory security on your iSeries server.

### 5.7.7 Changing the protection level for the userPassword attribute

The userPassword attribute in the directory schema is used to store a user's password. By default, the access class for the userPassword attribute is set to Critical. This password is used for several purposes, such as binding to the LDAP directory or when authenticating users via LDAP with the IBM HTTP Server for iSeries, WebSphere Application Server, WebSphere Commerce Suite, Lotus Domino, WebSphere Host On-Demand, and so forth. In most cases it is desirable to store the password in an encrypted form. In an ideal world all applications in the market would support the same encryption algorithm when

performing authentication using the password stored in the userPassword attribute. Unfortunately, this is not always the case. Therefore, you may come into a situation where you need to change the protection level of the userPassword attribute.

With OS/400 Version 5 Release 1 you can use the following protection levels:

<b>None</b>	Returns the actual password value. Returning the actual value requires that the Retain server security data system value (QRETSVRSEC) be set to 1. If the value is set to 0 (default value), one of the encrypted values is returned.
<b>SHA</b>	Secure Hash Algorithm (SHA) returns the password in the form {SHA}binary-data.
<b>MD5</b>	Message Digest 5 (MD5) returns the password in the form {MD5}binary data.
<b>Crypt</b>	The password is returned using the Crypt algorithm. Crypt is a modified DES algorithm. The password is one-way hashed.

By default, the protection level is set to SHA.

The userPassword attribute is stored using the password protection value that is specified at that time. If you change the password protection value later, the password may not be available in the newly specified format, and the most recently stored format will be used instead.

**Important:** Before changing the protection level for the userPassword attribute make sure that you know about the implications this might have for the applications that use this attribute. Some applications might not support the algorithm you have chosen.

The following steps walk you through changing the protection level for the userPassword attribute:

1. Using Operations Navigator, expand the system you want to change the protection level on.
2. Under the system, expand **Network** and then **Servers**.
3. Click **TCP/IP**.
4. Right-click **Directory** and select **Properties**.
5. On the Directory Properties window click the **Network** tab.



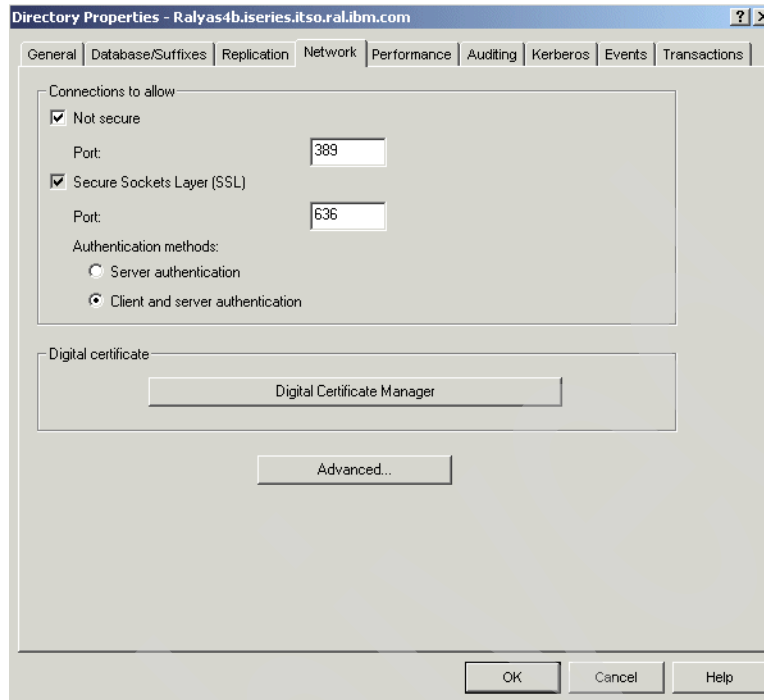


Figure 5-61 Directory Properties window - Network tab

6. Click **Advanced** to open the Directory - Advanced Network Properties window as shown in Figure 5-62.

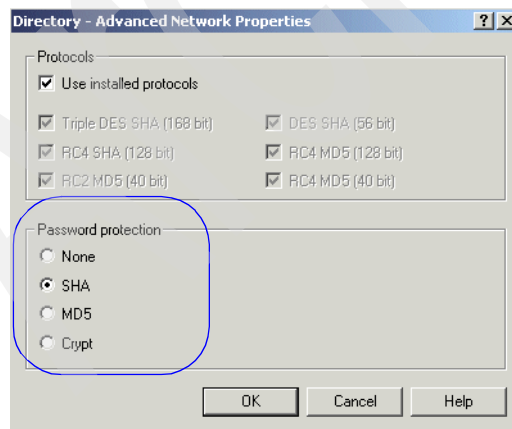


Figure 5-62 Directory - Advanced Network Properties window

7. Select the desired protection level and click **OK** to save your changes.

8. On the Directory Properties window click **OK** to close it.
9. Restart the Directory Services server to activate the new settings.

This completes the steps of changing the userPassword attribute protection level.

## IBM HTTP Server for iSeries LDAP support

This chapter provides information on how the IBM HTTP Server for iSeries (Original) and (powered by Apache) use a directory on an LDAP server for authentication and HTTP server configuration. It guides you through the various steps of planning and setting up the HTTP server to authenticate Web users via user information stored in an LDAP directory. Further, this chapter shows how one or more HTTP servers can load their configuration from a configuration file stored in an LDAP directory.

## 6.1 Introduction

The HTTP Server for iSeries allows you to minimize user and configuration administration and management by leveraging LDAP Directory Services. You can use the LDAP server for authenticating Web users that want to access protected resources on your HTTP Web server. The advantage of using centralized storage for user information is that many different applications, such as WebSphere Application Server, Lotus Domino, and HTTP server, can use authentication information that is kept in a single directory. For example, a user needs to change his password only once for all applications that use LDAP for authenticating users to perform authentication with the changed user password. Authentication verifies that users are who they say they are. A username and password is a basic authentication. Once users are authenticated, it must be determined whether they have the authorization or permission to perform the requested operation on the specific object.

Another feature of the HTTP Server for iSeries allows you to store server configuration directives in an LDAP directory. This is especially useful when operating a cluster of servers that are used for load sharing or backup purposes. In this case, the webmaster has to maintain only one set of configuration directives that are shared by all servers.

**Important:** We would recommend that you start with a powered by Apache server when the following is true:

- ▶ You are setting up a new Web server, Web site, or Web Application.
- ▶ You are not running Domino 5.0.x. Domino 5.0.x releases do not provide plug-in support for the powered by Apache server on iSeries. Domino 6.0 is expected to support a powered by Apache plug-in when it is released.
- ▶ You do not immediately require any products that do not yet support the powered by Apache server, such as WebSphere Commerce Suite, Payment Manager, and so forth.

In the medium-term IBM will bring future enhancements only to the powered by Apache server and will eventually drop the support for the Original server.

## 6.2 Scenario characteristics

The configuration shown in this chapter follows our redbook scenario as described in Section 3.1.1 “Stage 2 - The evolution” on page 41. Let us recap what the iSeries Shop characteristics and requirements are.

- ▶ The iSeries Shop operates an HTTP Web server on their iSeries server. To improve performance and availability, the company installs a second Web server on another iSeries server that is being used for back-up and load balancing purposes. Since the new Web server serves the same information as the existing one, the company wants to maintain the server configuration only in a single place. This approach minimizes the administration effort and allows for easy expansion in case they want to add additional servers to the cluster. To achieve this goal, the iSeries Shop exploits the LDAP configuration support included with the HTTP Server for iSeries product. The details of the set up are described in Section 6.4 “Configuration support” on page 254.
- ▶ The company’s Technical Support Department wants to offer special information to their premium customers over the Web. To ensure that only premium customers have access to the information, the content is protected by the Web server and customers have to authenticate to get access. This means that each customer is registered and needs a user ID and password to sign on. The operation of multiple Web servers raises another question: How can the company make sure that all Web servers have access to the user authentication data without replicating or copying the information to all Web servers? Well, the answer is pretty easy: the IT Department registers all customers in the iSeries LDAP directory. Then they modify the centrally-stored Web server configuration to authenticate Internet users via user information stored in the LDAP directory, as described in Section 6.3 “User authentication” on page 232.

The configuration, as shown in this chapter, shows the implementation on server RALYAS4B as depicted in Figure 6-1 on page 232. Note that the implementation is described for both the Original and the powered by Apache servers.

## iSeries Shop - Scenario HTTP LDAP Support

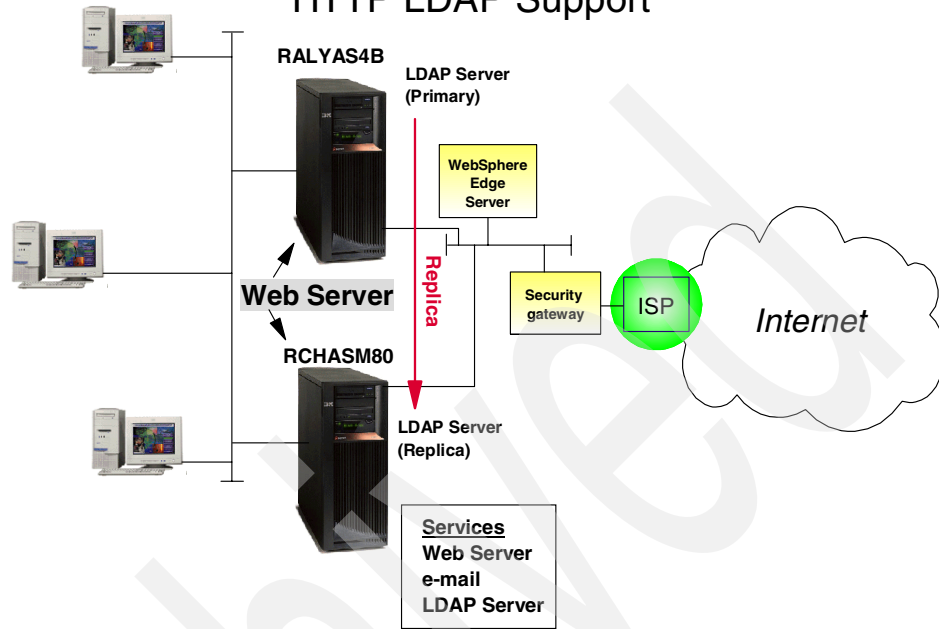


Figure 6-1 Redbook scenario - HTTP LDAP support

### 6.2.1 Prerequisites

The scenario and configuration as described in this chapter was performed on an iSeries server having the following prerequisite software components installed:

- ▶ 5722-SS1 - OS/400 V5R1 with CUM PTF Package C1302510
- ▶ 5722-DG1 - HTTP Server for iSeries
- ▶ 5722-XE1 - Client Access/400 Express for Windows (Service Pack SI01907)

In addition to the listed license programs you need a user profile with at least \*ALLOBJ and \*IOSYSCFG special authorities to perform the setup.

## 6.3 User authentication

A common task LDAP servers are used for is user authentication. We show in this section how to implement this on the IBM HTTP Server for iSeries (Original) and (powered by Apache).

### 6.3.1 Setting up LDAP authentication for the Original server

This section guides you through the steps for setting up the IBM HTTP Server for iSeries (Original) to perform authentication using user information stored in an LDAP directory. The section also includes the steps for setting up directory protection. Following the redbook scenario described in Section 3.1.1 “Stage 2 - The evolution” on page 41, the iSeries Shop Technical Support offers additional information to their registered premium customers. The authentication information for these customers is stored in the LDAP directory on system RALYAS4B and is also made available to the LDAP replica on system RCHASM80.

Perform the following steps to set up LDAP authentication:

1. From a Web browser access the HTTP Administration window on your iSeries server by using the following URL:

<http://ralyas4b:2001/HTTPAdmin>

Sign on when prompted with an OS/400 user profile that meets the requirements, as listed in Section 6.2.1 “Prerequisites” on page 232.

The HTTP Server Administration and Configuration main window is displayed. The HTTP server configuration utility requires that the HTTP \*Admin instance is up and running. You can use Operations Navigator (TCP/IP servers) or the following command to start the \*Admin instance:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

2. Select the **Configuration** tab in the left pane of your HTTP server configuration, as shown in Figure 6-2 on page 234. The iSeries Shop has set up an HTTP instance called ISHOPWEBOR and they stored the information for their premium customers in the directory path /www/ishopweb/premiumsupor.

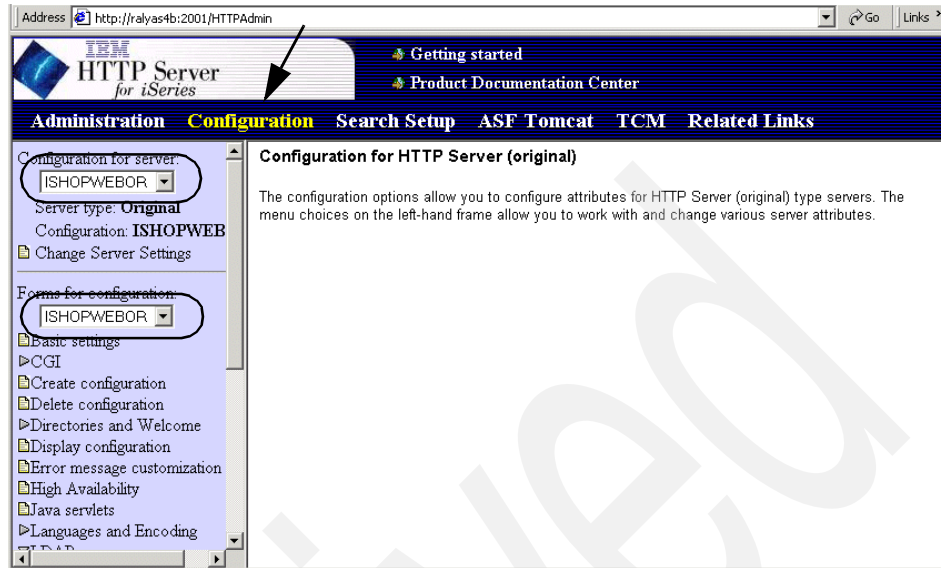


Figure 6-2 Main configuration window

This is the main configuration window where most of the configuration tasks can be done.

3. In the left pane scroll down to the LDAP section and click the triangle next to it to expand its options.
4. Select **Create LDAP server setup** to create a new LDAP server setup as shown in Figure 6-3 on page 235.



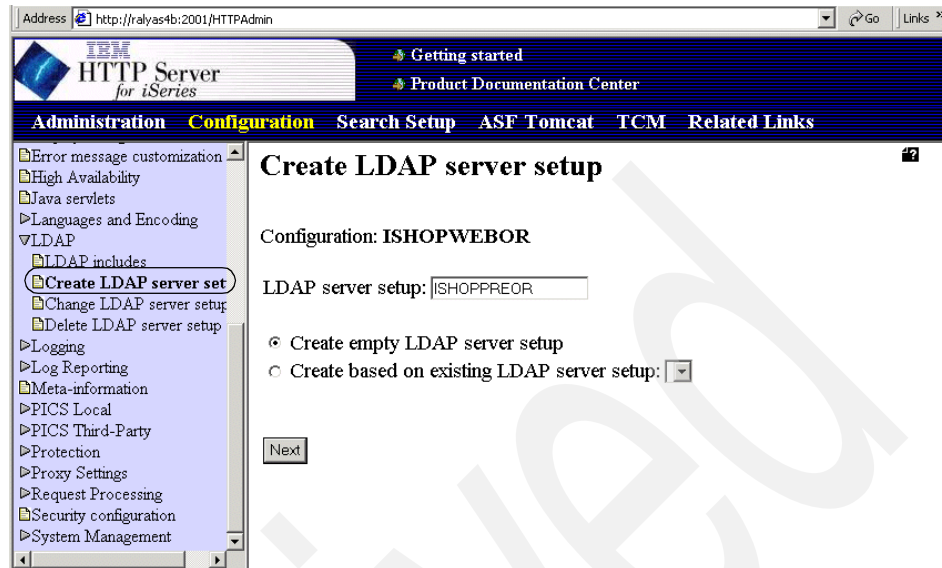


Figure 6-3 Create LDAP server setup window

5. Enter a name for the LDAP server setup and select **Create empty LDAP server setup**.
6. Click **Next** to continue to create the LDAP server setup.

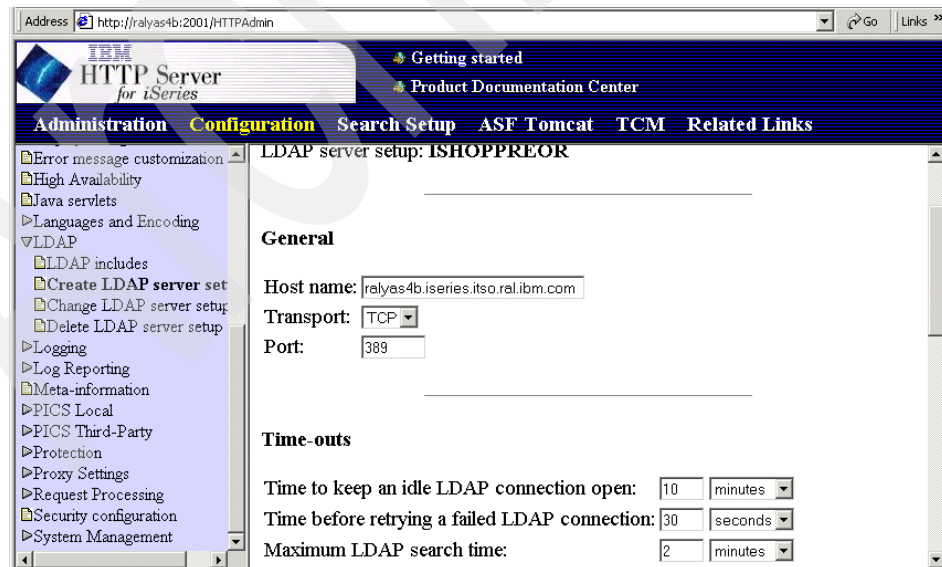


Figure 6-4 LDAP server setup window - General section

7. Enter the host name from your LDAP server. The host name is the name of the system on which the LDAP server is running. The LDAP server can run on your iSeries server or on another system. In this scenario enter the host name of system ralyas4b.iseries.itso.ral.ibm.com.
8. Select **TCP** and enter the port number your LDAP server listens on. The default port number for LDAP non-secure connections is 389. In this scenario, we have an LDAP server (master or replica) set up on the same servers the Web server is running on. Thus we use the TCP non-secure connection. In case your LDAP server resides on a different system, we recommend that you use a secure connection from the Web server to the LDAP server by selecting the SSL transport protocol. Note that the port number for secure sessions (using the SSL protocol) is the well-known port 636.

The parameter in the Time-outs section provide the IBM HTTP server with connection settings for the LDAP server. We recommend that you always start with the default values unless your network and application environment dictates different values.

9. Scroll down to the Server connection detail section as shown in Figure 6-5.

The screenshot shows the IBM HTTP Server for iSeries Administration Configuration page. The browser address bar displays `http://ralyas4b:2001/HTTPAdmin`. The page has a navigation bar with tabs: Administration, Configuration (selected), Search Setup, ASF Tomcat, TCM, and Related Links. On the left, there is a sidebar with a tree view showing configuration options for the selected server (ISHOPWEBOR). The main content area is titled "Server connection detail" and contains two sections: "LDAP authentication type" and "Client connection detail".

**Server connection detail**

LDAP authentication type:

- ☐ None
- ☒ Basic

HTTP server DN:

Password:

**Client connection detail**

User search root:

User search filter:

Delimiters for user data:

Figure 6-5 LDAP Server connection detail

Select **Basic** for the LDAP authentication type you want your server to use when connecting to the LDAP server. Basic requires the IBM HTTP server to provide the distinguished name and password of an entry in the LDAP directory that has the authority to access a user entry's userPassword attribute. Typically the administrator is used, but you can also set up a DN for this HTTP server that has the permission to access the attributes required to authenticate a user.

Enter the distinguished name (DN) and password for your IBM HTTP server DN.

<b>HTTP server DN</b>	cn=administrator
<b>Password</b>	Password of DN used

10. In the Client connection detail section enter the DN of the place in the directory information tree that stores the user information. This is the starting point for the LDAP server to start the search in the User search root field. In this scenario all customers are published to the ou=people,o=iseriesshop branch of the directory. Therefore we entered this value as our user search root:

<b>User search root</b>	ou=people,o=iseriesshop
-------------------------	-------------------------

The user search filter specifies how the HTTP server looks up a user in the LDAP directory. In the scenario we use the following user search filter:

<b>User search filter</b>	(&(objectclass=ePerson)(uid=%v1))
---------------------------	-----------------------------------

This filter specifies that the HTTP server searches for directory entries of an objectclass ePerson and where the uid (user ID) attribute must match the username value entered in the browser's authentication dialog window. Search filters also allow you to use the \* wildcard character if you want to allow a user to enter only a part of the attribute value. For example, it is a successful match if the username entered is BAK and the uid attribute in the directory entry is BAKER. The attribute you want to use for user authentication should uniquely identify a user. That means, you probably want to use only the uid or maybe an e-mail address as a unique identifier and not use wildcards at all. You can also search for user entries using multiple attributes. The following example takes the input from the user (provided in %v1) and searches for users of objectclass ePerson that have this value in the uid or mail attribute. For the mail attribute we allow searching using wildcard characters:

```
(&(objectclass=ePerson)(|(uid=%v1)(mail=%v1*)))
```

To give you another idea how to search for a user, here is an example where the user of objectclass=ePerson has to enter his first and last name separated by a comma in the browser's authentication challenge:

```
(&(objectclass=ePerson)(givenName=%v1)(sn=%v2))
```

To make this example work you also need to specify the delimiter for the user data field; in this case a comma (,).

If you want to grant access to users that are members of a certain group, you can do this by filling in the group search parameter and configuring the protection setup accordingly. Since groups are not used in our redbook scenario, this configuration is not covered in this section. For more information on how groups are used with LDAP and the HTTP (Original) refer to “Using groups for authentication” on page 240.

11. Click **Apply** to save your settings and return to the Create LDAP server setup window.

The remaining steps of the configuration show you how to protect a directory using basic authentication where the users are authenticated via an LDAP directory.

12. On the left pane, scroll down to Protection, and click the triangle next to it to expand this section. Then select **Document protection** as shown in Figure 6-6. Note that we do not show all possible ways of protecting a directory. For simplicity we take the approach of in-line protection setup.

The screenshot shows the IBM HTTP Server for iSeries Administration Configuration window. The left pane is expanded to show the 'Protection' section, with 'Document protection' selected. The main pane displays the configuration for document protection.

Configuration for server: ISHOPWEBOR  
Server type: Original  
Configuration: ISHOPWEB  
Change Server Settings

Forms for configuration: ISHOPWEBOR  
Basic settings  
CGI  
Create configuration  
Delete configuration  
Directories and Welcome  
Display configuration  
Error message customization  
High Availability  
Java servlets  
Languages and Encoding

Example	Path	Group	IP Address
Example	/restricted/*	WEB_MASTERS	9.83.29.4
Example	/Usage/*	PROT-ADMIN	
Example	/documents/*		host.ibm.com

Index: 0 ☐ Insert before ☐ Replace  
☒ Insert after ☐ Remove

URL template: /premiumsup/

Define protection settings: ☒ In-line  
☐ Named protection setup: [v]

Server IP address or host name: [ ]

Authentication options for in-line document protection:  
☒ Always prompt for user/password  
☐ Use SSL client authentication

Apply Reset

Figure 6-6 Document protection window

On this window you specify the URLs to protect on your server. For the URL template enter the directory you want to protect. This entry cannot contain any blank characters. In our case we the following URL template value:

/premiumsup/\*

This is the URL part of the pass directive that maps the request to the directory where the information for our premium customers are stored.

13. Select **In-line** for the protection settings.

Unless you want to process the protection for only a certain host name or IP address, leave the server IP address or host name blank.

14. Select **Always prompt for user/password**.

15. Click **Apply** to proceed to the last step, which takes you to the Protection Setup form where you can define the in-line protection setup as shown in Figure 6-7.

The screenshot shows the IBM HTTP Server for iSeries Administration Configuration window. The address bar displays `http://rallyas4b:2001/HTTPAdmin`. The main navigation bar includes links for **Administration**, **Configuration** (highlighted), **Search Setup**, **ASF Tomcat**, **TCM**, and **Related Links**. The left sidebar shows the configuration tree for the **ISHOPWEBOR** server, with options like **Basic settings**, **CGI**, **Create configuration**, **Delete configuration**, **Directories and Welcome**, **Display configuration**, **Error message customization**, **High Availability**, **Java servlets**, and **Languages and Encoding**. The main content area is titled **Protection realm: iShopPremium**. It contains a section for **Choose one user/password authentication option:** with three radio buttons: **Prompt for user/password using OS/400 user profile**, **Prompt for user/password using validation list**, and **Prompt for user/password using an LDAP server** (which is selected). Below the LDAP option is a field for **LDAP server setup name:** with the value **ISHOPPREOR**. The bottom section is titled **Authorization** and contains a **User ID:** field with the value **%%SERVER%%**.

Figure 6-7 Protection setup window

16. Enter a Protection realm, which is the name you want to use to identify this protection setup to requesters who will be entering a user ID and password.

When the server challenges a requester for username and password, it also includes the name you specify for the protection realm. Most browsers display this name with the prompt. Because different protection setups can use different forms of authentication, having a name associated with the protection setup can help the requester decide which username and password to send back. For our scenario we entered the protection realm of:

iShopPremium

17. Select **Prompt for user/password using an LDAP server**. This option asks users trying to access protected resources for the user ID and password information matching the entries on an LDAP server.
18. Select the LDAP server setup (ISHOPPREOR) you created in step 4 on page 234 from the drop-down list.
19. For Authorization enter the following user ID:

%%Server%%

The %%Server%% value indicates that the HTTP server swaps to the default profile of the server (QTMHHTTP) when accessing Web resources on the server. For more information refer to the online help text.

In the remaining fields you can enter any masks to apply against Get, Post, Put, Delete, or All requests. Entries can consist of usernames, group names, and address templates. This is an optional entry.

20. Click **Apply** to save the settings and return to the Document protection window.

You have now configured your HTTP server (Original) to use LDAP for authentication. You need to restart the server to activate your new configuration.

## Using groups for authentication

Well, you may ask yourself, when does it make sense to implement groups? To answer this question, let us imagine the following case. You have set up an HTTP Web server that authenticates users via entries in an LDAP directory. The directory information tree is as shown in Figure 6-8 on page 241.

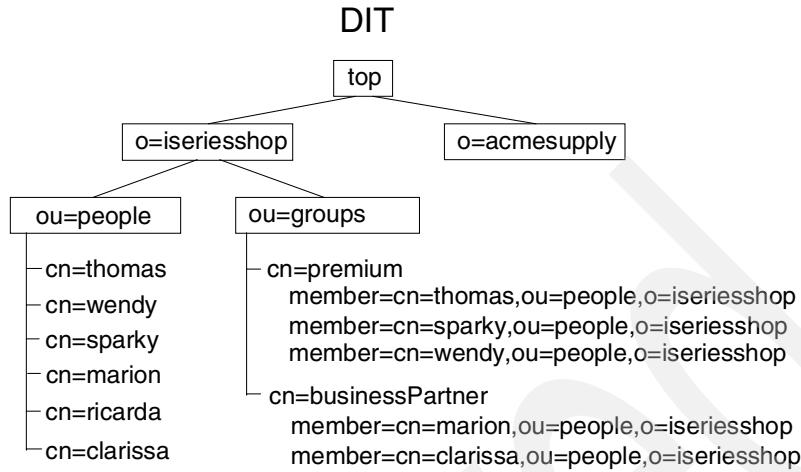


Figure 6-8 Sample DIT for group authentication

All registered users are published under the DN `ou=people,o=iseriesshop`. In addition, we defined two groups (objectclass `groupOfNames`) `premium` and `businessPartner`, which are published under `ou=groups,o=iseriesshop`. Using groups allows you to grant access to different protected resources based on a user's membership in a group. The HTTP server is set up to protect information that only registered premium customers can access. It also has a protection setup that allows business partners to browse restricted areas. Using groups makes the authentication setup pretty easy as shown in the configuration in Figure 6-1.

Example 6-1 Original server group support configuration

```

# IBM HTTP Server configuration file
LDAPInfo AS4B {
    Host ralyas4b.iseries.itso.ral.ibm.com
    Transport TCP
    Port 389
    IdleConnTimeout 10 Minutes
    WaitToRetryConnTime 30 Seconds
    SearchTimeout 2 Minutes
    CacheTimeout 5 Minutes
    ServerAuthType Basic
    ServerDN "cn=adminstrator"
    UserSearchBase "ou=people,o=iseriesshop"
    UserNameFilter "(&(objectclass=person)(uid=%v1))"
    GroupSearchBase "ou=groups,o=iseriesshop"
    GroupNameFilter "(&(cn=%v1)(objectclass=groupOfNames))"
    GroupMemberAttrs "member"
    ClientAuthType Basic
  }
  
```

```

}
Protection Premium {
    PasswdFile %%LDAP:AS4B%%
    ACLOverride Off
    PostMask Premium
    GetMask Premium
    AuthType Basic
    ServerID PremiumCustomers
    UserID %%SERVER%%
}
Protect /premium/* Premium
Protection BP {
    PasswdFile %%LDAP:AS4B%%
    ACLOverride Off
    PostMask businessPartner
    PutMask businessPartner
    GetMask businessPartner
    AuthType Basic
    ServerID BusinessPartner
    UserID %%SERVER%%
}
Protect /bpsupport/* BP
BindSpecific Off
Pass /bpsupport/* /www/restrict/bp/*
Pass /premium/* /www/restrict/prem/*
Pass /public/* /www/web1/public/*
Pass /* /www/web1/index.html
Port 80

```

➔ 3

➔ 4

The configuration as shown in Example 6-1 performs the following:

1. Section **1** of the configuration specifies where the HTTP server searches for user information in the LDAP directory (UserSearchBase). All users are published to the ou=people,o=iseriesshop branch of the tree. The input from the user in the browser's authentication challenge is checked against entries that belong to the objectclass person and have a uid attribute value that matches the user input (UserNameFilter).
2. Section **2** specifies how the HTTP server determines whether the authenticated user belongs to a certain group. In this case the server searches for groups in ou=groups,o=iseriesshop (GroupSearchBase). The search is performed for objects that belong to the groupOfNames objectclass and where the name of the group (cn) matches the value provided with the %v1 variable (GroupNameFilter). Note that the %v1 variable in the group name filter has nothing to do with the %v1 variable used in the username filter; they just happen to have the same name. The variable %v1 in the group name filter is replaced with the group name specified in the protection setup as explained in the next paragraph.



3. The first named protection setup (Premium) as shown in section 3 of the configuration, defines that authentication is performed via users stored in an LDAP directory (PasswdFile). We also specified that only Get and Post request methods can be used on resources that are protected by this named protection setup. The word Premium as specified on the GetMask and PostMask directives (for example, GetMask Premium) represent the name of the group an authenticated user has to be a member of to gain access to the protected resource. This value has to match the name of the group in the LDAP directory.

The actual protection of the URL /premium/ is defined in the Protect /premium/\* Premium directive where the named protection premium is used to protect the URL.

When a user tries to access information via a URL such as `http://hostname/premium/` the HTTP server knows that the URL is protected. It processes the authentication request as defined in the named protection setup. Within the protection setup it is specified that only users that belong to the Premium group can use the Get and Post method. The HTTP server first challenges the user to enter a username and password. Then the server uses UserSearchBase and UserNameFilter to find and authenticate the user. Once the user is authenticated, the server takes the group name as specified in the GetMask and PostMask directives and checks whether the user is a member of this group. The check is performed by using the GroupSearchBase, GroupNameFilter, and GroupMemberAttrs directives. Within the search base, the HTTP server searches for the group passed as variable %v1 and where the objectclass is groupOfNames. If the group is found, the LDAP server returns the attribute, in this case member, as specified in the GroupMemberAttrs directive. The member attribute contains one or more distinguished names of all members in this group. If the HTTP server finds the authenticated user in the returned attribute, the user gets access to the requested information.

The authentication process flow is also important to know when debugging problems. For example, if you open the access log of the server and a user was successfully authenticated, but is not a member of the group that has access to the protected resource, you see an entry like the following:

```
10.24.105.175 - timsmith [22/Feb/2002:13:52:52 +0500] "GET /premium/
HTTP/1.1" 401 250
```

In this case you can already see that a username timsmith is logged, which means the user is successfully authenticated, but the 401 indicates that the user has no access to the requested resource.

4. Section 4 in the configuration defines the setup for protecting information that only business partners can access.

More information about HTTP server LDAP capabilities can be found in the article *Using LDAP with the HTTP Server (Original)* found at:

<http://www.ibm.com/eserver/series/software/http/services/ldapinfo.html>

### 6.3.2 Setting up LDAP authentication for the powered by Apache server

This section guides you through the steps for setting up the IBM HTTP Server for iSeries (powered by Apache) to perform authentication using user information stored in an LDAP directory. The section also includes the steps for setting up directory protection. Following the redbook scenario described in Section 3.1.1 “Stage 2 - The evolution” on page 41, the iSeries Shop Technical Support offers additional information to their registered premium customers. The authentication information for these customers is stored in the LDAP directory on system RALYAS4B and is also made available to the LDAP replica on system RCHASM80. The configuration described in this section also includes the steps to protect a resource.

Perform the following steps to set up LDAP authentication:

1. From a Web browser access the HTTP Administration window on your iSeries server by using the following URL:

<http://rallyas4b:2001/HTTPAdmin>

Sign on when prompted with an OS/400 user profile that meets the requirements as listed in Section 6.2.1 “Prerequisites” on page 232.

The HTTP Server Administration and Configuration main window is displayed. The HTTP server configuration utility requires that the HTTP \*Admin instance is up and running. You can use Operations Navigator (TCP/IP servers) or the following command to start the \*Admin instance:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

2. Select the **Configuration** tab. The iSeries Shop has set up an HTTP instance called ISHOPWEB and they stored the information for their premium customers in the directory path /www/ishopweb/premiumsup.
3. Select the directory path that you want to protect in the navigation pane on the left side of window. In this scenario we selected **Directory /www/ishopweb/premiumsup**.

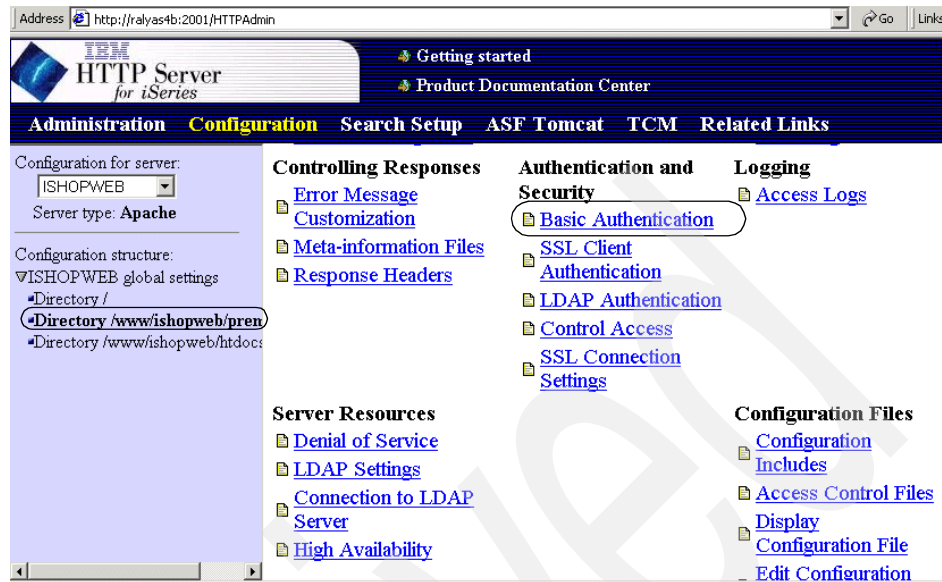


Figure 6-9 Main configuration window

After selecting the directory context on the navigation pane, the available configuration options on the right side of the window refresh.

4. Under Authentication and Security click **Basic Authentication** to define how the server will authenticate users as shown in Figure 6-10 on page 246.

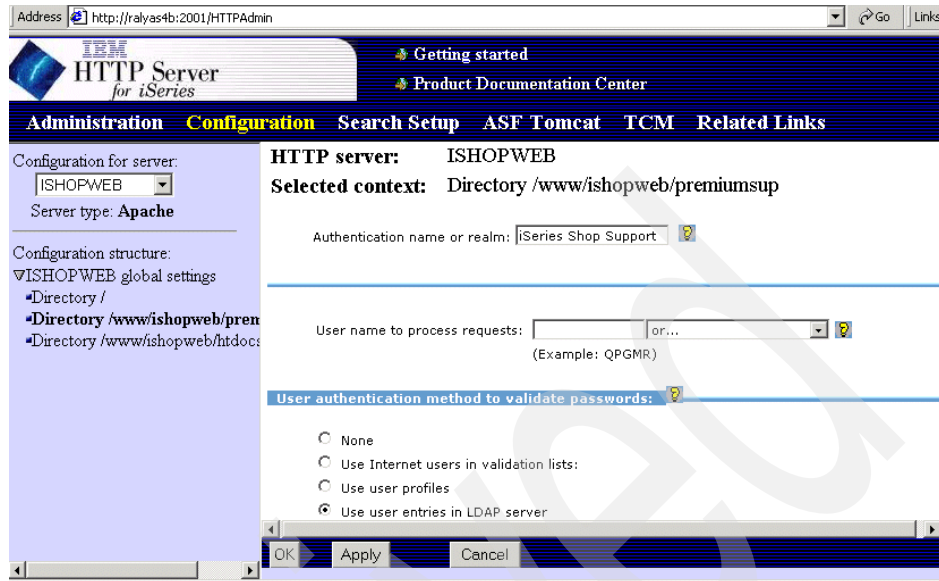


Figure 6-10 Basic Authentication window

Enter the authentication name or realm. This name is usually displayed by a Web browser in a pop-up dialog window when prompting for a username and password to access the requested resource. This information can then be used to determine what username and password to enter.

For username select **Default server profile** and for User authentication method select **Use user entries in LDAP server**.

Note that you cannot process server requests under a client user profile when authenticating users with LDAP or validation lists.

5. Click **OK** to save your changes and return to the main configuration window.
6. Make sure that the directory you want to protect is still selected and click **Control Access**.

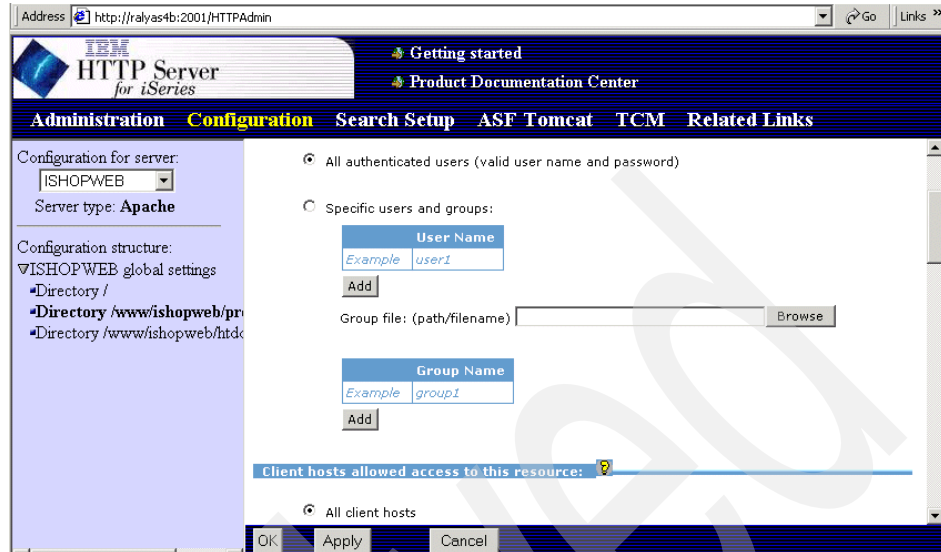


Figure 6-11 Control Access window

On the Control Access window select **All authenticated users (valid username and password)** for users and groups who can access this resource. For client hosts allowed access to this resource, select **All client hosts**.

Use the defaults for the remaining parameters as shown in the following list:

- Client hosts that cannot access this resource: Specific client hosts.
  - Environment variable: No value selected.
  - Method for evaluating access: Allow access before denying access.
7. Click **OK** to save your settings and return to the main configuration window.
  8. Click **LDAP Authentication** to define further details for LDAP authentication, as shown in Figure 6-12 on page 248.

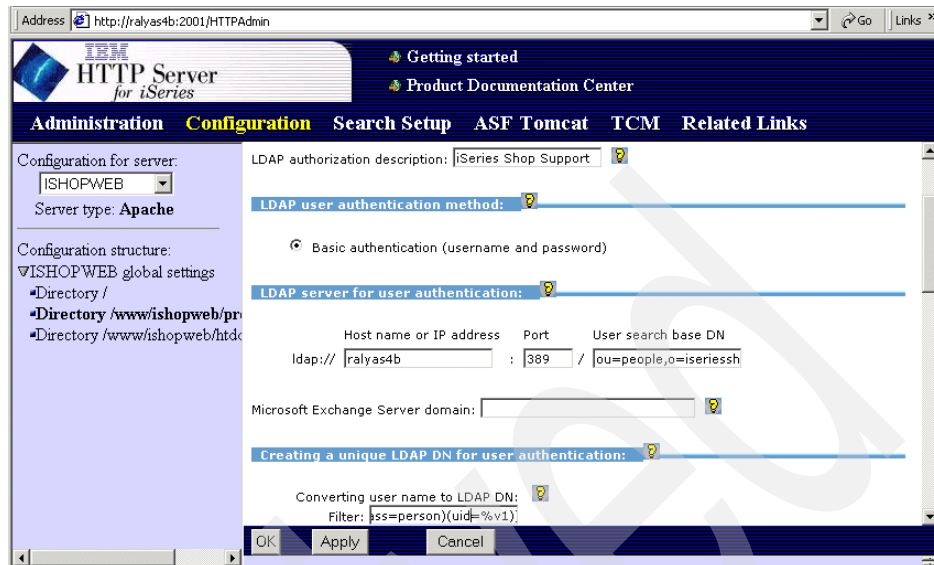


Figure 6-12 LDAP Authentication window

For LDAP authorization description, enter a label to identify an LDAP authentication configuration. The label can be any character string, and can include multiple words with spaces. There is no default value. The value entered is not displayed to the Web user. It is shown in the log files.

Select **Basic authentication (username and password)** as the LDAP authentication method. Currently, this is the only authentication method the HTTP Server for iSeries supports.

The parameter for the LDAP server for user authentication specifies the host where the LDAP directory resides and the location within the directory tree containing the user information. Enter the host name or IP address, the port number, and the user search base DN. In this scenario, we used the following parameters for the ldap field:

**Host name or IP address:** RALYAS4B  
**Port:** 389  
**User search Base DN:** ou=people,o=iseriessh

The section on creating a unique LDAP DN for user authentication specifies how the HTTP server looks up a user in the LDAP directory. In this scenario we use the search filter (&(objectclass=person)(uid=%v1)).

This filter specifies that the HTTP server searches for directory entries of an objectclass Person and where the uid (user ID) attribute must match the username value entered in the browser's authentication dialog window. Search filters also allow you to use the wildcard (\*) character if you want to

allow a user to enter only a part of the attribute value. For example, it is a successful match if the username entered is BAK and the uid attribute in the directory entry is BAKER. The attribute you want to use for user authentication should uniquely identify a user. This means that you probably want to use only the uid or maybe an e-mail address as a unique identifier and not use wildcards at all. You can also search for user entries using multiple attributes. The following example takes the input from the user (provided in %v1) and searches for users of objectclass ePerson that have this value in the uid or mail attribute. For the mail attribute we allow searching using wildcard characters:

```
(&(objectclass=ePerson)(|(uid=%v1)(mail=%v1*)))
```

To give you another idea how to search for a user, following is an example where the user of objectclass=ePerson has to enter his first and last name separated by a comma in the browser's authentication challenge:

```
(&(objectclass=ePerson)(givenName=%v1)(sn=%v2))
```

To make this example work you also need to specify the delimiter for valid field separators, in this case a comma (,).

If you want to grant access to users that are members of a certain group, you can do this by filling in the parameters for configuring group authentication support. Since groups are not used in our redbook scenario, this configuration is not covered in this section. For more information on how groups are used with LDAP and the HTTP server (powered by Apache) refer to "Using groups for authentication" on page 251.

9. Click **OK** to save your configuration and return to the main configuration window.
10. Under Server Resources click **Connection to LDAP Server** to provide options for configuring LDAP connections for authentication and setup as shown in Figure 6-13 on page 250.

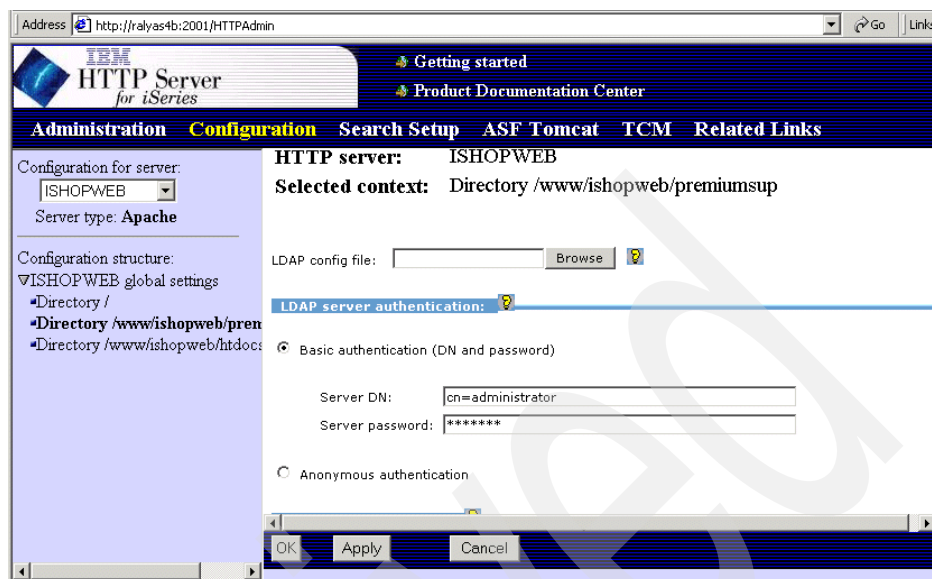


Figure 6-13 Connection to LDAP Server window

On the Connection to LDAP Server window, for LDAP server authentication select **Basic authentication (DN and password)** and enter the following values:

**Server DN:** cn=administrator

**Server password:** Password of LDAP directory administrator as specified on the Server DN parameter

The server DN specifies the distinguished name the application (in this case the Web server) uses to authenticate itself to the LDAP server. The server password specifies the password of the DN specified in the server DN parameter. Once the information is saved, a stashfile is created in the directory /QIBM/UserData/HTTPSVR/LDAP/ISHOPWEB. This file stores the password in an encrypted form and is used by the HTTP Web server to access the password when connecting and authenticating to the LDAP server. This file will also be used when using the LDAP configuration file (ldap.prop) as described in the configuration support section.

Note that the server DN does not need to be the administrator DN, but it has to be a DN of a directory entry that has access to the user entries including the userPassword attribute.

Use the default values on the remaining parameters.

11. Click **OK** to save your configuration and return to the main configuration window.



12. Restart the server to activate the new protection and authentication setup.

Now we have configured our ISHOPWEB IBM HTTP Server for iSeries (powered by Apache) for our iSeries Shop to use the LDAP server to authenticate our premium customers.

## Using groups for authentication

The IBM HTTP Server for iSeries (powered by Apache) also provides support for group authentication. To better understand when and how to use group authentication, let us imagine the following case. You have set up an HTTP Web server that authenticates users via entries in an LDAP directory. The directory information tree is as shown in Figure 6-8.

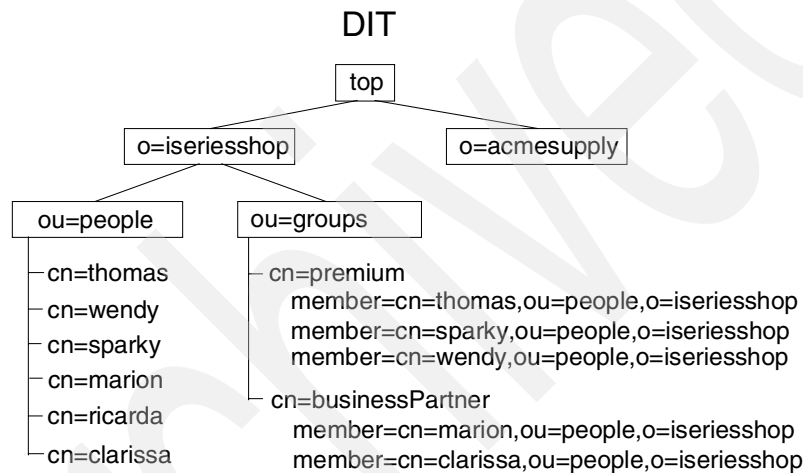


Figure 6-14 Sample DIT for group authentication

All registered users are published under the DN `ou=people,o=iseriesshop`. In addition we defined two groups (objectclass, `groupOfNames`) `premium` and `businessPartner`, which are published under `ou=groups,o=iseriesshop`. Using groups allows you to grant access to different protected resources based on a user's membership in a group. The HTTP server is set up to protect information that only registered premium customers can access. It also has a protection setup that allows business partners to browse restricted areas. Using groups makes the authentication setup pretty easy, as shown in the following configuration.

### Example 6-2 Powered by Apache server group support configuration (excerpt)

```
# Configuration originally created by Apache Setup Wizard Tue Feb 26
23:39:01 GMT+00:00 2002
LoadModule ibm_ldap_module /QSYS.LIB/QHTTPSVR.LIB/QZSRVLDP.SRVPGM
```

```

LoadModule ibm_ldap_include /QSYS.LIB/QHTTPSVR.LIB/QZSRVLDPAP.SRVPGM
Alias /premium/ /www/ishopweb/premium/
Alias /bpsupport/ /www/ishopweb/businesspartner/
Listen 80
<Directory /www/ishopweb/businesspartner>
    AllowOverride None
    AuthName "Business Partner"
    ProfileToken off
    AuthType Basic
    order allow,deny
    allow from all
    require valid-user
    UserID %%SERVER%%
    PasswdFile %%LDAP%%
    ldap.user.authType Basic
    ldap.url ldap://rallyas4b:389/ou=people,o=iseriesshop
    ldap.user.name.filter (&(objectclass=person)(uid=%v1))
    ldap.user.name.fieldSep " \t,"
    ldap.group.memberAttributes member
    ldap.group.url ldap://rallyas4a:389/ou=groups,o=iseriesshop
    ldap.group.name.filter (&(cn=%v)(objectclass=groupofnames))
    LDAPRequire group businessPartner
    ldap.application.authType Basic
    ldap.application.DN cn=administrator
    ldap.application.password.stashFile
/QIBM/UserData/HTTPSVR/LDAP/ISHOPWEB/1014755449800.stash
    ldap.transport TCP
    ldap.version 3
    ldap.waitForRetryConnection.interval 30
    ldap.idleConnection.timeout 600
</Directory>
<Directory /www/ishopweb/premium>
    AllowOverride None
    AuthName "Tom Premium"
    ProfileToken off
    AuthType Basic
    order allow,deny
    allow from all
    require valid-user
    UserID %%SERVER%%
    PasswdFile %%LDAP%%
    ldap.realm TOM_LDAP_AUTH
    ldap.user.authType Basic
    ldap.url ldap://rallyas4b:389/ou=people,o=iseriesshop
    ldap.user.name.filter (&(objectclass=person)(uid=%v1))
    ldap.group.name.filter (&(cn=%v)(objectclass=groupofnames))
    ldap.user.name.fieldSep " \t,"
    ldap.group.memberAttributes member
    ldap.application.authType Basic

```

```

ldap.application.DN cn=administrator
ldap.application.password.stashFile
/QIBM/UserData/HTTPSVR/LDAP/ISHOPWEB/1014754186700.stash
ldap.transport TCP
ldap.version 3
ldap.waitToRetryConnection.interval 30
ldap.idleConnection.timeout 600
ldap.group.url ldap://ralyas4b:389/ou=groups,o=iseriesshop
LDAPRequire group premium 3
</Directory>

```

---

The configuration, as shown in Example 6-2 on page 251, performs the following:

1. Section **1** of the configuration specifies where the HTTP server searches for user information in the LDAP directory (ldap.url). All users are published to the ou=people,o=iseriesshop branch of the tree. The input from the user in the browser's authentication challenge is checked against entries that belong to the objectclass person and have a uid attribute value that matches the user input (ldap.user.name.filter).
2. Section **2** specifies how the HTTP server determines whether the authenticated user belongs to a certain group. In this case the server searches for groups in ou=groups,o=iseriesshop on the LDAP server with the name ralyas4b, and which listens on port 389 (ldap.group.url). The search is performed for objects that belong to the groupOfNames objectclass and where the name of the group (cn) matches the value provided with the %v variable (ldap.group.name.filter). The variable %v in the group name filter is replaced with the group name specified in the LDAPRequire directive as explained in the next paragraph.
3. The first directory context (/www/ishopweb/businesspartner), as shown in Example 6-2 on page 251, defines that authentication is performed via users stored in an LDAP directory (PasswdFile). The entire configuration of how users will be authenticated, how the HTTP server authenticates to the LDAP server (ldap.application.DN and ldap.application.password.stashfile), and how the LDAP directory is searched, is specified in the directory context. The actual group an authenticated user has to be a member of to gain access to the protected resource is specified in the LDAPRequire **(2)** directive. In this case, LDAPRequire group businessPartner, the parameter group defines that the HTTP server uses the group name in combination with ldap.group.name.filter and ldap.group.memberAttributes to determine if the user belongs to the group. As an alternative you can also specify your own search filter on the LDAPRequire directive to determine if the user matches the group requirements. In the latter case, the ldap.group.name.filter is not being used.

When a user now tries to access information via a URL such as <http://hostname/bpsupport/>, the HTTP server knows that the URL is protected. It processes the authentication request as defined in the directory context. The HTTP server first challenges the user to enter a username and password. Then the server uses the `ldap.url` and `ldap.user.name.filter` to find and authenticate the user. Once the user is authenticated, the server takes the group name as specified on the `LDAPRequire` directive and checks whether the user is a member of this group. The check is performed by using the `ldap.group.url`, `ldap.group.name.filter`, and `ldap.group.memberAttributes` directives. Within the search base, the HTTP server searches for the group passed as variable `%v` and where the objectclass is `groupOfNames`. If the group is found, the LDAP server returns the attribute, in this case `member`, as specified in the `ldap.group.memberAttributes` directive. The member attribute contains one or more distinguished names of all members in this group. If the HTTP server finds the authenticated user in the returned attribute, the user gets access to the requested information. Note that without the `LDAPRequire` directive, group authentication will not work. If you use LDAP configuration files as introduced in Section 6.4.2 “LDAP configuration support for powered by Apache server” on page 270 you need to add the `LDAPRequire` directive in the directory context and not in the configuration file.

The authentication process flow is also important to know when debugging problems. For example, if you open the access log of the server and a user was successfully authenticated, but is not a member of the group that has access to the protected resource, you see an entry like the following:

```
10.24.105.169 - thomas [27/Feb/2002:21:13:45 +0000] "GET /bpsupport/
HTTP/1.1" 403 207
```

In this case you can already see that a username `thomas` is logged, which means the user is successfully authenticated, but the 403 indicates that the user has no access to the requested resource.

4. Section 3 in the configuration defines that group authentication is required. This time, the users have to be a member of the premium group.

## 6.4 Configuration support

Another fantastic feature of the IBM HTTP Server for iSeries (Original) and (powered by Apache) is the support of retrieving server configuration directives from an LDAP directory. If you operate just one HTTP Web server you probably do not use this support, but once you start using at least two Web servers you can take full advantage of using the LDAP configuration support. For example, when you use two Web servers for load sharing and back-up purposes like our

iSeries Shop does, then you need to maintain just a single set of configuration directives. These directives are published into an LDAP directory and both servers retrieve the configuration via include directives during server startup time. Sounds good? Well it is good when you follow certain rules.

1. Have your Webmaster set up a test server instance. This instance is used to maintain your main server configuration and test configuration changes before deployment. There should be no doubt about this point, because making changes to a production instance without prior testing is asking for trouble.
2. Once the configuration is properly tested, the configuration directives are published into the LDAP server. And here comes a second aspect into the picture. You can selectively publish information; this means that if you have a common set of directives that you want to use on three Web servers, you just need to publish those. If required, each of the three servers can still maintain their own directives that only apply to the individual server. The common directives are then included from the LDAP server during server startup. There is virtually no limit to the possibilities you have with this support. For example, you can build logical blocks that hold certain configuration directives. With multiple include directives you can then link different blocks of configuration directives together. You may want to consider implementing LDAP server replicas for availability reasons.
3. On each server that is designated to include configuration directives from an LDAP server, you create a basic Web server configuration and then add the necessary include directives to load the rest of the configuration from the LDAP server.

If you follow the previously listed rules, you can build a Web server environment that is easy to maintain and easy to extend. In the following sections of this chapter we show you how to set up the LDAP configuration include support for the HTTP Server for iSeries (Original) and (powered by Apache).

### **6.4.1 Setting up the LDAP configuration support for the Original server**

In our redbook scenario, the iSeries Shop has set up two HTTP Web servers. In this section, we assume that they have a test HTTP server running where they perform all configuration tests and changes before deploying a new configuration to the production servers.

This section guides you through the following configuration tasks:

- ▶ Export the test server configuration into a file.
- ▶ Create an LDIF file to publish the server configuration to the LDAP server.

- Publish the server configuration to the LDAP server.
- Set up a new production server instance and prepare the instance to include directives from an entry in the LDAP directory.
- Start the production server instance.

Figure 6-15 shows the basic flow for and relationships between objects when implementing the HTTP LDAP include support.

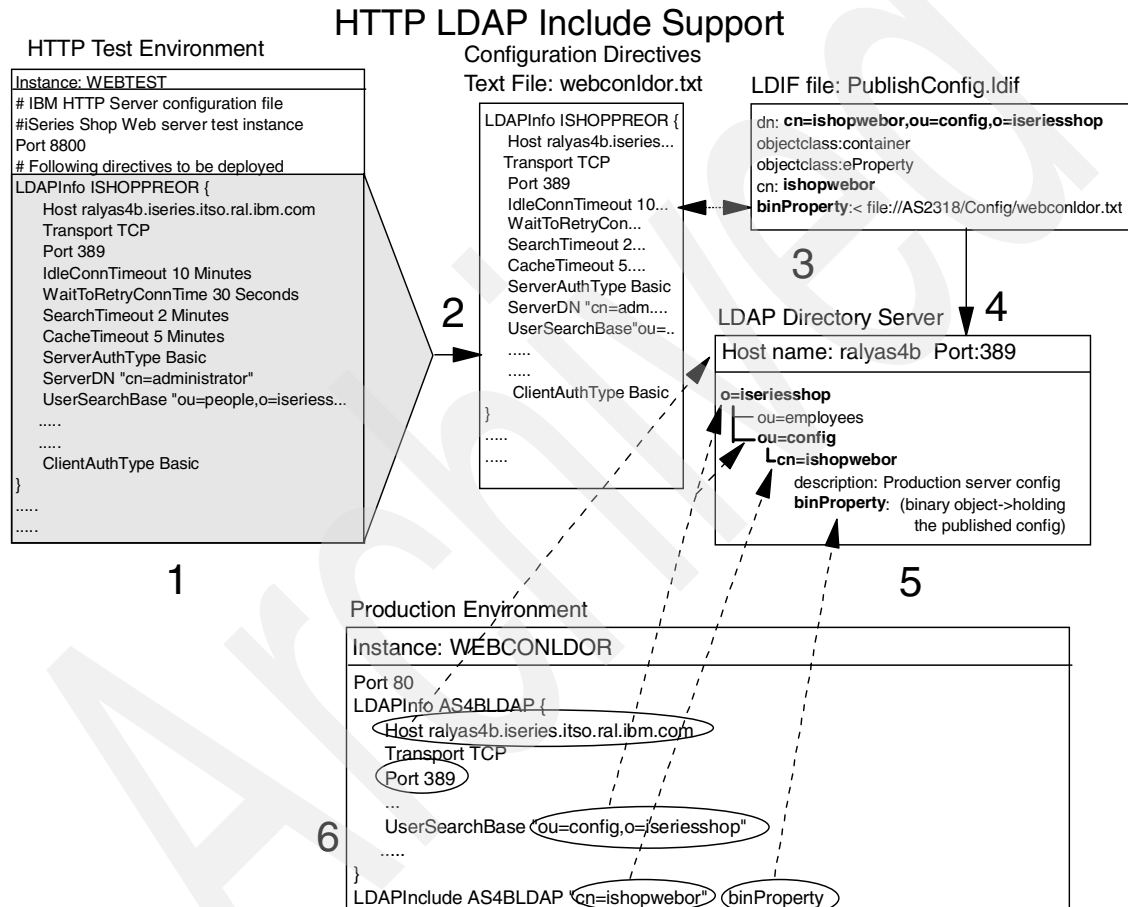


Figure 6-15 HTTP server (Original) configuration overview

At a first glance it might seem like a lot of work to implement the LDAP include support. But you will see that most of the steps are for initial setup only and not difficult at all. Once the initial setup is done, deploying configuration changes basically involves just two steps. The following list describes the various components and steps that are involved in implementing the LDAP include support:

1. This is our HTTP test environment where the iSeries Shop Webmaster configures and tests all the HTTP server configurations and applications. The test instance WEBTEST listens on port 8800 to not interfere with production server ports. The configuration directives that are to be deployed are separated from the test instance unique directives by a comment line.
2. Upon successful completion of the configuration and testing, you copy and paste the server directives into an ordinary text file. In this scenario we used Notepad as the editor. We named the file `webconldor.txt` and placed it on our server RALYAS4B into the IFS directory `/AS2318/Config`.
3. In this step of the implementation, an LDIF file needs to be created. This file, in our scenario named `PublishConfig.ldif`, will be used to publish the configuration file `webconldor.txt` into the LDAP directory. The attribute that holds the configuration is named `binProperty` and is an optional attribute of the `eProperty` object class. The `eProperty` object class itself needs an object class as a required attribute, thus the object class container, which has a required attribute of `cn` (`commonName`). The `cn=ishopwebor` combined with the attributes `ou=config,o=iseriesshop` represent the qualified DN of the directory entry that contains the configuration. The value of the `binProperty` attribute refers to the IFS path and file name of the file that holds the server configuration directives.
4. When the configuration and LDIF files are created, you are ready to publish your server configuration to the LDAP server. In this case, the underlying directory entries `ou=config` and `o=iseriesshop` already exist. The `ldapmodify` utility can be used to perform the initial creation of the Web server configuration entry as well as for periodic updates.
5. This step in Figure 6-15 on page 256 represents the directory entry of the Web server configuration after it has been published to the directory server RALYAS4B.
6. The Web server instance `WEBCONLDOR` is our production Web server. Typically the configuration contains only directives that are unique to that server, for example, the Port directive, and the necessary directives to locate and load the additional configuration directives stored in the LDAP directory. An LDAP server setup needs to be created. It contains information about how the HTTP Web server can reach the LDAP server (host name, port, protocol, and so forth). It also specifies the directory search base that is used for

locating the directory entry within the DIT. As shown in Figure 6-15 on page 256, the LDAPInclude directive has three parameters:

- The first parameter identifies the name of the LDAP server setup. In this scenario it is AS4BLDAP. The LDAPInclude directive contains information about the directory entry's name and attribute, but no information about where the entry is located within the DIT (ou=config,o=iseriesshop). This information is provided in the LDAP server setup.
- In the second parameter you define the LDAP directory entry that contains the attribute that holds the server configuration directives.
- The last parameter refers to the attribute name within the directory entry that stores the configuration directives.

### Creating the WEBCONLDOR.TXT file

At this implementation step, the Webmaster has tested the Web server configuration on the test server instance and is ready to export it. The steps described in this section show you how to easily export the configuration from the test server instance to a text file. This text file will later be published to the LDAP directory.

1. Open a Web browser window and go to the HTTP Administration page of the system your test instance is running on using the following URL:  
<http://ralyas4b:2001/HTTPAdmin>
2. Select the **Configuration** tab and make sure your test server instance is selected.
3. In the navigation pane click **Display configuration**.





Figure 6-16 Display configuration window

The easiest way to copy your server directives is by using the Display configuration option and then selecting and copying all directives you want to have published to the LDAP server.

4. Select all the directives to be published. In this scenario they are separated from the test server specific directives by a comment line. Then press Ctrl-C to copy the directives into your clipboard.
5. Use a text editor to create an ASCII text file. In this scenario we used the Notepad editor and created a file named webconldor.txt, as shown in Figure 6-15 on page 256. Paste your directives from the clipboard into the Notepad editor window by pressing Ctrl-V (or the equivalent option of your editor).

```

LDAPInfo ISHOPPREOR {
  Host ralyas4b.iSeries.itso.ral.ibm.com
  Transport TCP
  Port 389
  IdleConnTimeout 10 Minutes
  WaitToRetryConnTime 30 Seconds
  SearchTimeout 2 Minutes
  CacheTimeout 5 Minutes
  ServerAuthType Basic
  ServerDN "cn=administrator"
  UserSearchBase "ou=people,o=iSeriesShop"
  UserNameFilter "(&(objectclass=ePerson)(uid=%v1))"
  GroupSearchBase "ou=groups,o=iSeriesShop"
  GroupNameFilter
  "(&(cn=%v1)(|(objectclass=groupofnames)(objectclass=groupofuniquenames)))"
  GroupMemberAttrs "member"
  ClientAuthType Basic
}
Protect /premiumsup/ {
  PasswdFile %%LDAP:ISHOPPREOR%%
  ACLOverride off
  PostMask premium
  GetMask premium
  AuthType Basic
  ServerID ishopPremium
  UserID %%SERVER%%
}
BindSpecific off
AccessLog /www/ishopwebor/logs/Accesslog 2000
AccessLogArchive None
AccessLogExpire 0
AccessLogSizeLimit 0
ErrorLog /www/ishopwebor/logs/Errorlog 2000 *DFT *DFT
ErrorLogArchive None
ErrorLogExpire 0

```

Figure 6-17 Editor window

6. Verify the directives again and save the file. Since we will publish the configuration via the QShell LDAP utilities, the file is saved to the IFS file system.

The file that contains your HTTP Web server configuration directives is now created.

## Creating an LDIF file

The easiest way to perform the initial publishing of the configuration and also updates is the approach of using LDAP Data Interchange Format (LDIF) files. This file contains information following a specific syntax that allows an LDAP server to publish certain information to the directory. In this scenario we create a file that is used to create, and later on update, the Web server configuration directives in the LDAP directory, as described in the following steps.

1. On your PC or 5250 session create a new text file using an editor such as Notepad (PC) or EDTF (OS/400).
2. Enter the following information into the new file:
 

```

dn: cn=ishopwebor,ou=config,o=iSeriesShop
objectclass: container
objectclass: eProperty

```

```
cn: ishopwebor  
binProperty:< file:///AS2318/Config/webconldor.txt
```

The first entry is the distinguished name (DN) and represents the name and location of the new entry within the directory. The next two entries specify the object classes the new entry belongs to. The cn (commonName) attribute must also be specified, as it is a required attribute of the container object class. The binProperty attribute's value specifies that a file is to be published from the location specified. In this scenario, it is the webconldor.txt file that holds the server configuration directives.

3. Save the file. We saved the file under the name PublishConfig.ldif into the IFS directory /AS2318/config.

## Publishing the Web server configuration

Now that you have the Web server configuration directives stored in a text file webconldor.txt and the LDIF file PublishConfig.ldif, you can publish the configuration directives to the LDAP directory. In our scenario we used the ldapmodify utility in the OS/400 QShell to perform the publishing process.

1. From a 5250 session, start the QShell by entering the following command:

```
qsh
```

Press Enter. The QShell Command Entry window is shown.

2. Depending on whether you publish the configuration for the first time to or update an existing configuration in the LDAP directory, you need to specify different switches for the ldapmodify command.

### – First-time configuration

Enter the following command to publish the configuration for the first time:

```
ldapmodify -D "cn=administrator" -w "my5ldap"  
-f "/as2318/config/PublishConfig.ldif" -a
```

The QShell responds for a successful publishing with a message like:

```
$  
> ldapmodify -D "cn=administrator" -w "my5ldap" -f  
"/as2318/config/PublishConfig.ldif" -a  
adding new entry cn=ishopwebor,ou=config,o=iseriesshop  
$
```

### – Updating an existing configuration

Enter the following command to update an existing configuration:

```
ldapmodify -D "cn=administrator" -w "my5ldap"  
-f "/as2318/config/PublishConfig.ldif" -r
```

The QShell responds for a successful publishing with a message like:

```
$
> ldapmodify -D "cn=administrator" -w "my5ldap" -f
"/as2318/config/PublishConfig.ldif" -r
modifying entry cn=ishopwebor,ou=config,o=iserieshop
```

\$

This command creates or updates an object ishopwebor as specified in the PublishConfig.ldif file and uploads the HTTP configuration from the webconldor.txt file. The major difference between publishing for the first time or updating an existing entry is the -a (add) and -r (replace) command switches. The switches and their meanings as used in this scenario are as follows:

- |           |   |
|-----------|---|
| <b>-D</b> | The DN used to bind to the LDAP directory. In our scenario it is the user DN (cn=administrator) to access the LDAP server.  |
| <b>-w</b> | The password of the user to access the LDAP server.   |
| <b>-f</b> | Reads the entry modification information from an LDIF file instead of the standard input.<br><br>/as2318/config/PublishConfig.ldif is the path to the LDIF file used. |
| <b>-a</b> | Used on the <b>ldapmodify</b> command to indicate that the utility tries to add entries by default. Using this parameter is the same as using <b>ldapadd</b> .        |
| <b>-r</b> | Specifies that <b>ldapmodify</b> tries to replace existing values by default. This parameter is used if you update the existing object.                               |

The Web server configuration is now published and available to be included by one or more HTTP servers in the network.

## Setting up a new production server instance

Perform the following steps to set up the HTTP Server for iSeries (Original) to use the LDAP support for loading configuration directives:

1. From a Web browser access the HTTP Administration window on your iSeries server by using the following URL:

<http://rallyas4b:2001/HTTPAdmin>

The HTTP Server Administration and configuration main window is displayed. The HTTP server configuration utility requires that the HTTP \*Admin instance is up and running. You can use Operations Navigator (TCP/IP servers) or the following command to start the \*Admin instance:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

2. Select the **Administration** tab and then click **Create HTTP Server** on the navigation pane as shown in Figure 6-18.

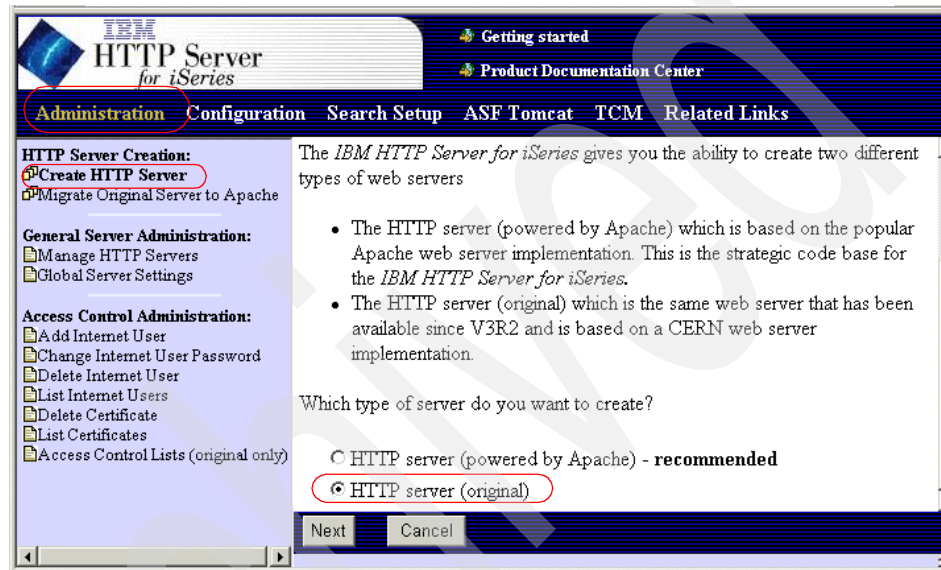


Figure 6-18 Create HTTP server window

3. Select **HTTP server (original)** and click **Next** to continue with the server instance creation wizard.
4. In the next window enter your HTTP server name **WEBCONLDOR** and click **Next**.
5. Select **Create a new original type configuration** and leave the displayed value (**WEBCONLDOR**) and click **Next**. The server root is displayed in the following window.
6. The next wizard question prompts you for the IP address and port the server should listen on for requests. In our scenario we left the IP address at **A11** addresses and the port at **80**. According to our implementation scenario shown in Figure 6-15 on page 256, the test server listened on port **8800** and the production server on port **80**. Click **Next**.
7. Specify the document directory. This directory is used by the new instance to serve a document. In this scenario, we left it as the default because we are going to delete the directives anyway and load our configuration from the

LDAP directory. Click **Next** to display the configuration summary, as shown in Figure 6-19.

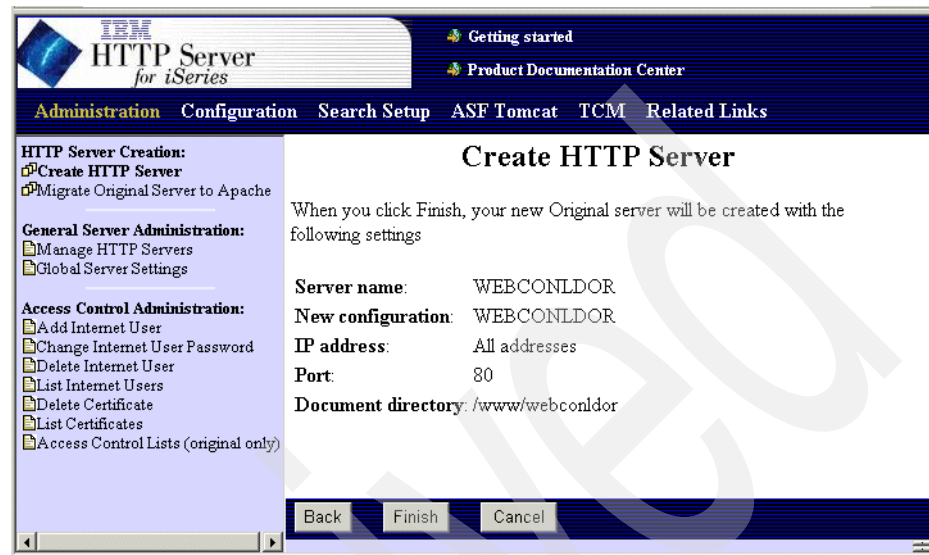


Figure 6-19 Create HTTP Server summary window

8. Click **Finish** and the wizard creates the HTTP server instance according to your input.
9. After the instance is created, a new window is displayed allowing you to manage or configure the instance. Click **Configure** to continue. The server instance configuration window is displayed. Make sure that the name of your newly created instance and the configuration member is displayed in the Configuration for server drop-down list.
10. In the Forms for configuration pane expand **Request Processing** and select **Request routing**.

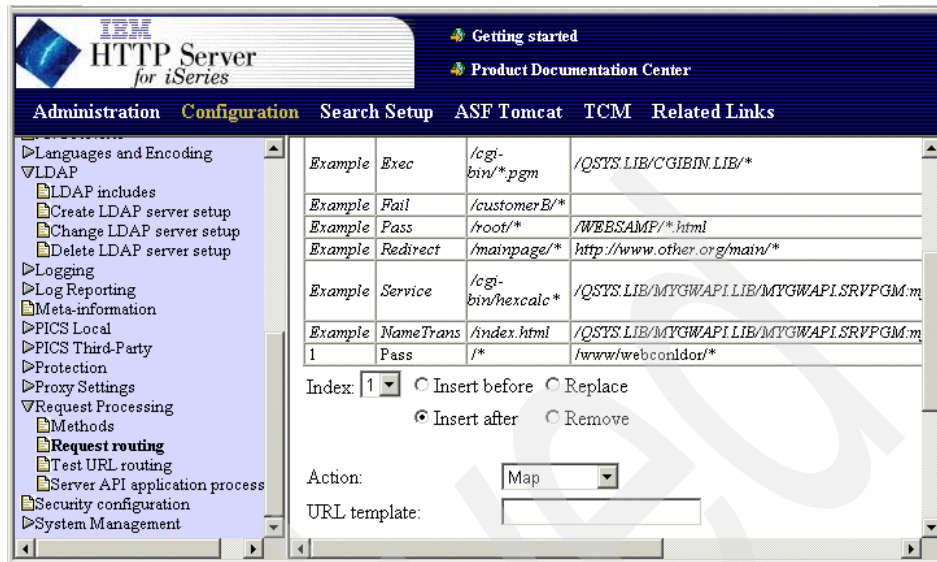


Figure 6-20 Request routing window

- Specify the suffix of the Pass action that was added by the wizard, select **Remove**, and click **Apply**. This will remove the Pass directive from the configuration. Remember, we do not need the directive, as we load all directives from an LDAP directory.

**Note:** This is the time to remove other directives you might not use.

- In the navigation pane expand **LDAP** and click **Create LDAP server setup**. This starts a series of configuration steps to create an LDAPInfo section. This section contains information about the LDAP server location, the port, what DN is used to authenticate to the LDAP server, and so forth.

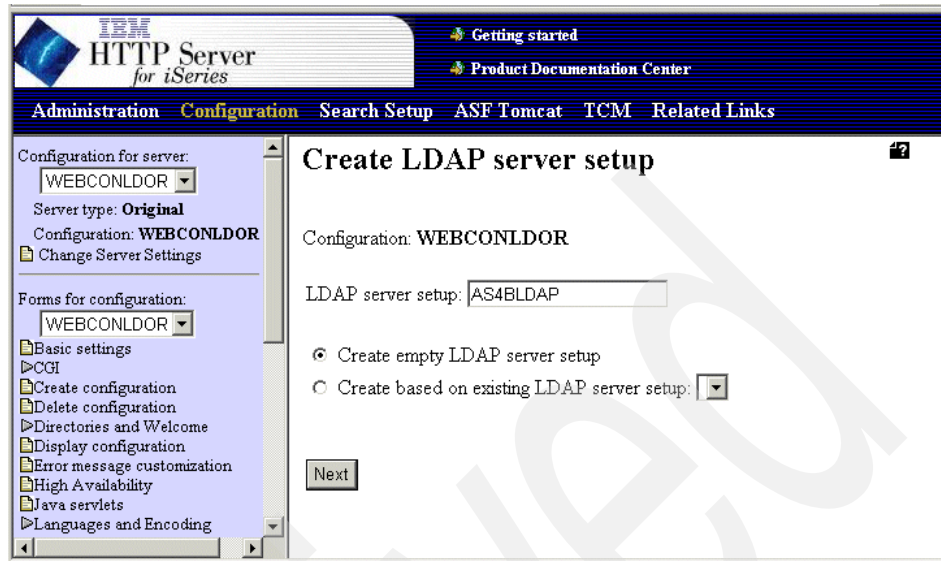


Figure 6-21 Create LDAP server setup window

13. Enter a name for the LDAP server setup. In this scenario we have chosen a name that represents our LDAP server (AS4BLDAP), and select **Create empty LDAP server setup**.
14. Click **Next**.



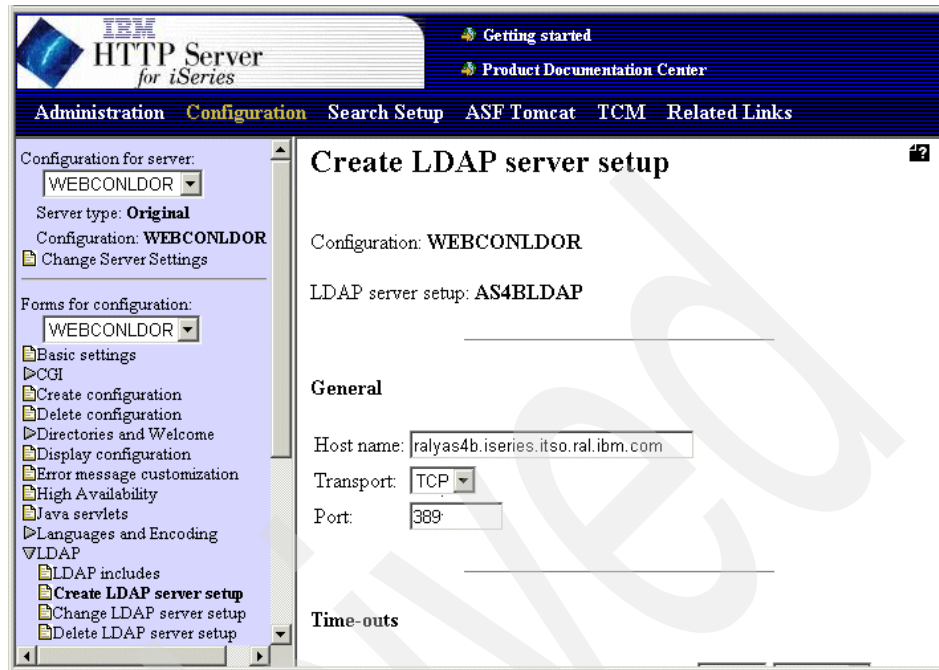


Figure 6-22 Create LDAP server setup window - general information

Enter the information in the General section as follows:

**Host name** ralyas4b.iseries.itso.ral.ibm.com  
This is the host name of your LDAP server.

**Transport** TCP  
Specifies the transport protocol for the connection between the HTTP and LDAP server. In our scenario both servers are on the same system. Therefore, we have left the default value as TCP. In case your connection is through a network, we recommend using SSL to protect the traffic.

**Port** 389  
For a non-secure connection (TCP) use the well-known port 389. For secure connections (SSL), the well-known port is 636.

We left the time-out values as their defaults.

15. Scroll down to the Server connection detail section.

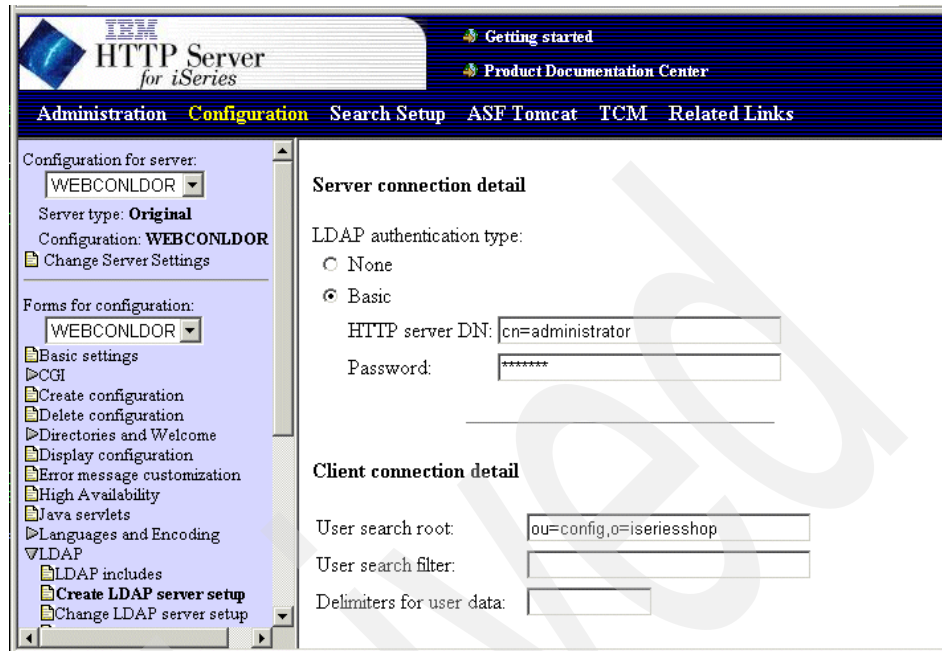


Figure 6-23 Create LDAP server setup window - Server connection detail section

In the Server connection detail section select **Basic** for the LDAP authentication type you want your server to use when connecting to the LDAP server. Basic requires the IBM HTTP server to provide the distinguished name and password of an entry in the LDAP directory that has the authority to access a user entry's userPassword attribute. Typically the administrator is used, but you can also set up a DN for this HTTP server that has the permission to access the attributes required to authenticate a user.

Enter the distinguished name (DN) and password for your IBM HTTP server DN.

<b>HTTP server DN</b>	cn=administrator
<b>Password</b>	Password of DN used

In the Client connection detail section enter the User Search root as follows:

<b>User search root</b>	ou=config,o=iseriesshop
	This is the place in the DIT where the entry is stored that holds the server configuration directives to be included during server startup.

16. Click **Apply** to save the settings.

So far you have configured a basic server setup and added an LDAP server setup. In the remaining steps you configure the LDAPInclude directive.

17. In the navigation pane LDAP section, click **LDAP includes**.

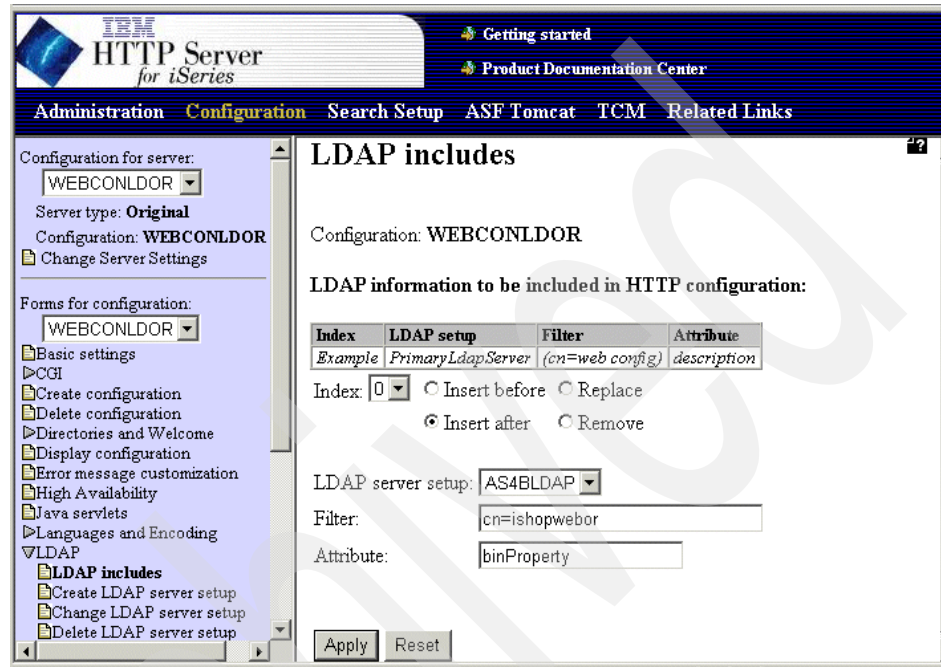


Figure 6-24 LDAP includes window

Adding an entry in the LDAP includes window adds a new LDAPInclude directive to the HTTP server configuration. The HTTP server (Original) allows you to add several LDAPInclude directives, which allows you to build a modular configuration structure and load one or more of this modular configuration blocks. In our scenario we specified the following parameters for the LDAPInclude directive:

**LDAP server setup** AS4BLDAP

This is a drop-down list. You can select a previously created LDAP server setup. In this scenario it is the LDAP server setup we created for the LDAP include configuration.

**Filter**

cn=ishopwebor

This filter specifies the name of the entry in the LDAP directory that holds the server configuration directives to be included.

<b>Attribute</b>	binProperty
	This is the attribute within the directory entry that holds the the server configuration directives to be included.

18. Click **Apply** to add the new LDAPInclude directive to the production server configuration.

### Starting the production server instance

At this point in the implementation phase, the server configuration as exported from the test instance is published to the LDAP server and available to be included at server start-up time. A production server instance has been created that only contains the necessary directives to include server configuration directives from an LDAP directory.

You can now start your production server. During startup, the HTTP server binds to the LDAP server, retrieves and includes the configuration directives stored in the binProperty attribute.

**Note:** When you use the HTTP Administration interface and display the HTTP server configuration of your production server instance, you will not see the directives included from the LDAP directory. If you need to troubleshoot a problem, you may want to use the -vv switch in the instance startup parameter when starting the HTTP server instance. This will cause the server job to create a spoolfile containing trace information. The entire process of binding to the LDAP server, searching the configuration entry, as well as processing the included directives, is listed in the spoolfile.

## 6.4.2 LDAP configuration support for powered by Apache server

In our redbook scenario, the iSeries Shop has set up two HTTP Web servers. In this section, we assume that they have a test HTTP server running where they perform all configuration tests and changes before deploying a new configuration to the production servers.

This section guides you through the following configuration tasks:

- ▶ Export the test server configuration into a file.
- ▶ Create an LDIF file to publish the server configuration to the LDAP server.
- ▶ Publish the server configuration to the LDAP server.
- ▶ Set up a new production server instance and prepare the instance to include directives from an entry in the LDAP directory.
- ▶ Start the production server instance.

Figure 6-25 shows the basic flow for and relationships between objects when implementing the HTTP LDAP include support.

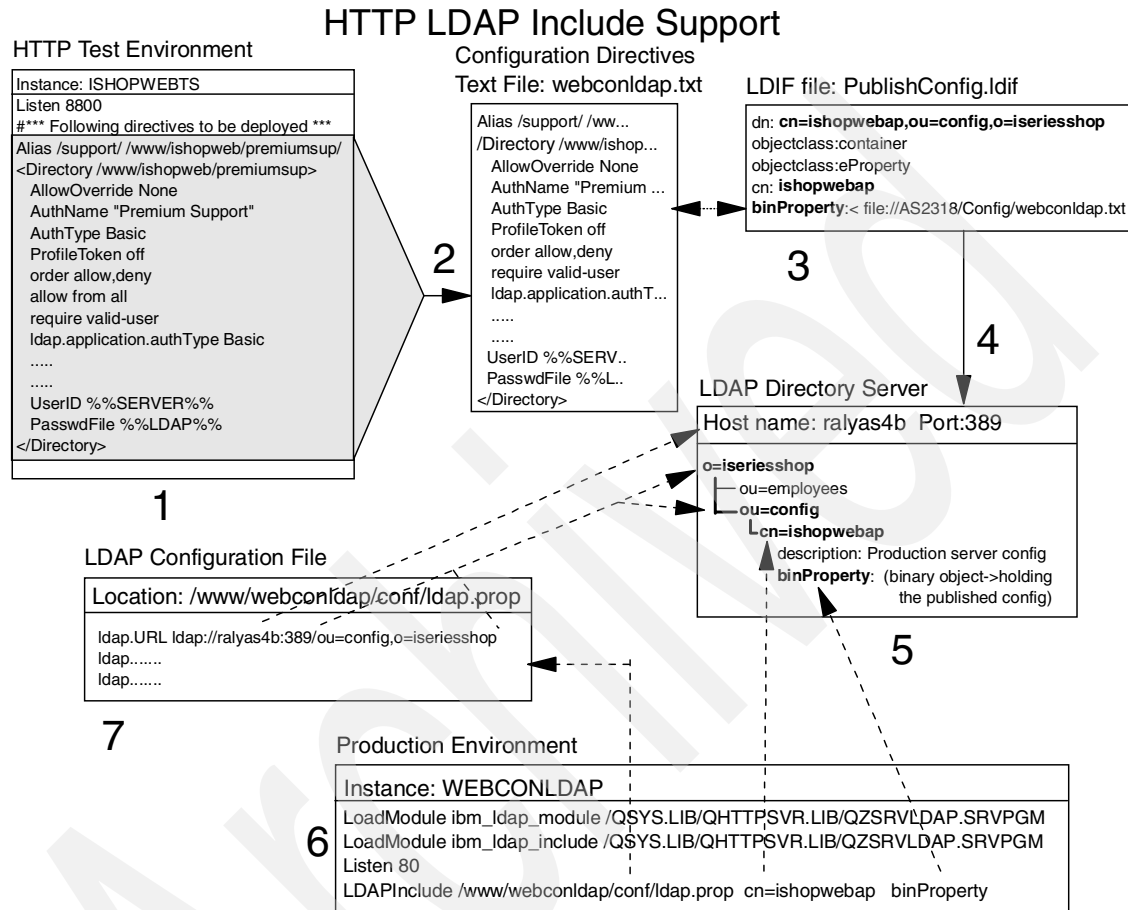


Figure 6-25 HTTP server powered by Apache configuration overview

At a first glance it might seem like a lot of work to implement the LDAP include support. But you will see that most of the steps are for initial set up only and not difficult at all. Once the initial set up is done, deploying configuration changes basically involves just two steps. The following list describes the various components and steps that are involved in implementing the LDAP include support:

1. This is our HTTP test environment where the iSeries Shop Webmaster configures and tests all the HTTP server configurations and applications. The test instance ISHOPWEBTS listens on port 8800 to not interfere with

production server ports. The configuration directives that are to be deployed are separated from the test instance unique directives by a comment line.

2. Upon successful completion of the configuration and testing, you copy and paste the server directives into an ordinary text file. In this scenario we used Notepad as the editor. We named the file webconldap.txt and placed it on our server RALYAS4B into the IFS directory /AS2318/Config.
3. In this step of the implementation, an LDIF file needs to be created. This file, in our scenario named PublishConfig.ldif, will be used to publish the configuration file webconldap.txt into the LDAP directory. The attribute that holds the configuration is named binProperty and is an optional attribute of the eProperty object class. The eProperty object class itself needs an object class as a required attribute, thus the object class container, which has a required attribute of cn (commonName). The cn=ishopwebap combined with the attributes ou=config,o=iseriesshop represent the qualified DN of the directory entry that contains the configuration. The value of the binProperty attribute refers to the IFS path and file name of the file that holds the server configuration directives.
4. When the configuration and LDIF files are created, you are ready to publish your server configuration to the LDAP server. In this case, the underlying directory entries ou=config and o=iseriesshop already exist. The ldapmodify utility can be used to perform the initial creation of the Web server configuration entry as well as for periodic updates.
5. This step in Figure 6-25 on page 271 represents the directory entry of the Web server configuration after it has been published to the directory server RALYAS4B.
6. The Web server instance WEBCONLDAP is our production Web server. Typically the configuration contains only directives that are unique to that server, for example the Listen directive, and the necessary directives to locate and load the additional configuration directives stored in the LDAP directory. The two LoadModule directives allow the HTTP Web server to process LDAP related directives as specified in the ldap.prop file, as well as processing the LDAPInclude directive. As shown in Figure 6-25 on page 271, the LDAPInclude directive has three parameters:
  - The first parameter identifies the path and file name of the LDAP configuration file.
  - In the second parameter you define the LDAP directory entry that contains the attribute that holds the server configuration directives.
  - The last parameter refers to the attribute name within the directory entry that stores the configuration directives.

You may have noticed that the LDAPInclude directive contains information about the directory entry's name and attribute, but no information about where the entry is located within the DIT (ou=config,o=iseriesshop). This information is provided in the LDAP configuration file.

7. The last missing piece in the implementation is the LDAP configuration file. This file contains information about how the HTTP Web server can reach the LDAP server (host name, port, protocol, and so forth). It also specifies the directory search base that is used for locating the directory entry within the DIT.

### Creating the WEBCONLDAP.TXT file

At this implementation step, the Webmaster has tested the Web server configuration on the test server instance and is ready to export it. The steps described in this section show you how to easily export the configuration from the test server instance to a text file. This text file will later be published to the LDAP directory.

1. Open a Web browser window and go to the HTTP Administration page of the system your test instance is running on using the following URL:  
<http://ralyas4b:2001/HTTPAdmin>
2. Select the **Configuration** tab and make sure your test server instance is selected.
3. In the Configuration Files section click **Edit Configuration File**.

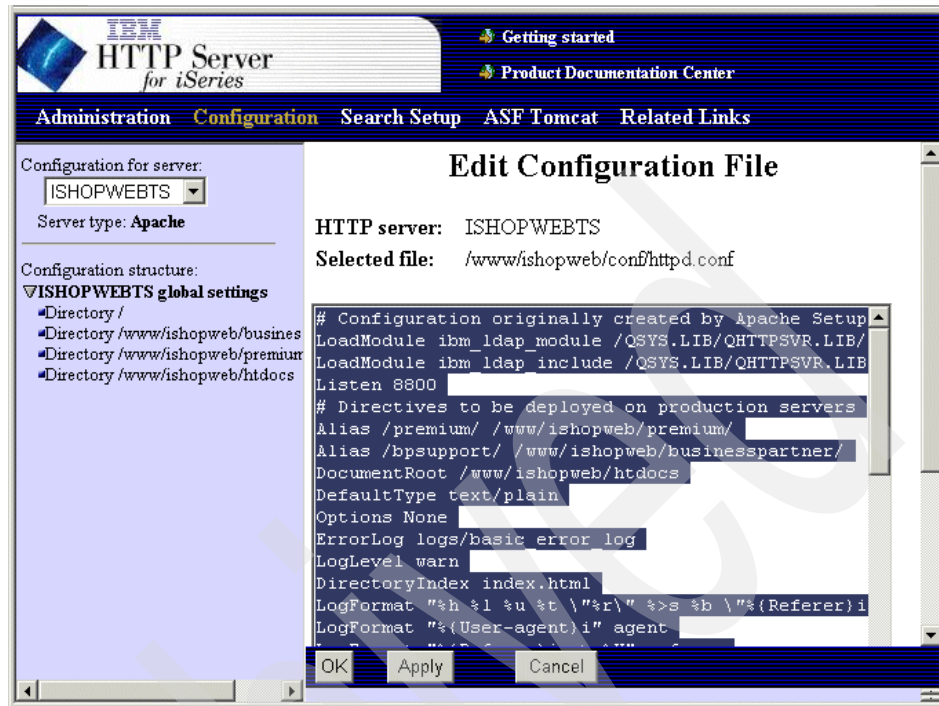
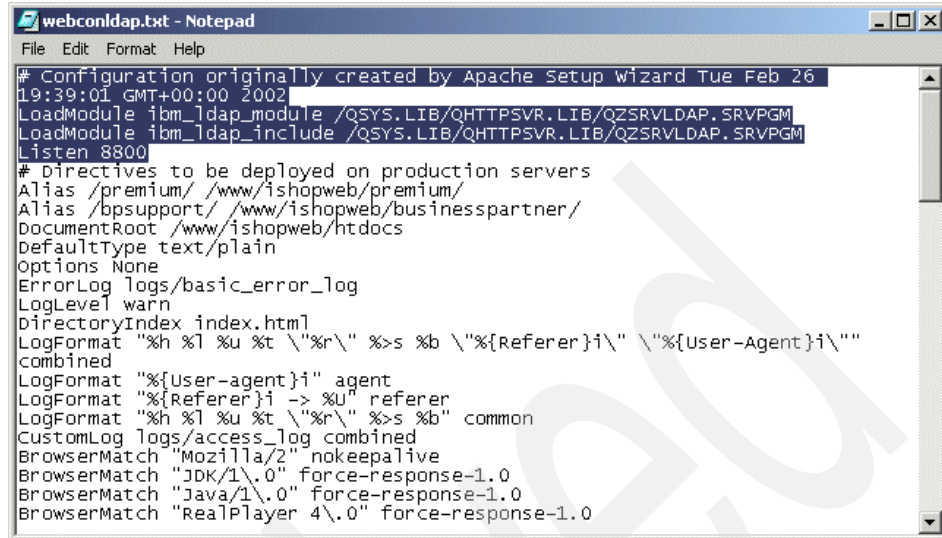


Figure 6-26 Edit Configuration File window

The easiest way to copy your server directives is by using the Edit Configuration File option and then selecting and copying all directives. The Display Configuration File option offers a better view of all directives, but when copying directives, you also get sequence numbers for all directives. Sequence numbers are not needed, and create more work in order for you to get rid of them.

4. Select an area in the Edit window and press Ctrl-A (or an equivalent option in your browser) to select all directives. Then press Ctrl-C to copy the directives into your clipboard.
5. Use a text editor to create an ASCII text file. In this scenario we used the Notepad editor and created a file named webconldap.txt, as shown in Figure 6-25 on page 271. Paste your directives from the clipboard into the Notepad editor window by pressing Ctrl-V (or the equivalent option of your editor). Note that at this time you still have all directives of the test server instance in the editor window.





```
# Configuration originally created by Apache Setup Wizard Tue Feb 26
19:39:01 GMT+00:00 2002
LoadModule ibm_ldap_module /QSYS.LIB/QHTTSPVR.LIB/QZSRVLDP.SRVPGM
LoadModule ibm_ldap_include /QSYS.LIB/QHTTSPVR.LIB/QZSRVLDP.SRVPGM
Listen 8800

# Directives to be deployed on production servers
Alias /premium/ /www/ishopweb/premium/
Alias /bpsupport/ /www/ishopweb/businesspartner/
DocumentRoot /www/ishopweb/htdocs
DefaultType text/plain
Options None
ErrorLog logs/basic_error_log
LogLevel warn
DirectoryIndex index.html
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
LogFormat "%{User-agent}i" agent
LogFormat "%{Referer}i -> %U" referer
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log combined
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "JDK/1\\.0" force-response-1.0
BrowserMatch "Java/1\\.0" force-response-1.0
BrowserMatch "RealPlayer 4\\.0" force-response-1.0
```

Figure 6-27 Editor window

6. Delete the directives you do not want to publish and save the file. In this case we deleted all directives prior to the separator comment line. Since we will publish the configuration via the QShell LDAP utilities, the file is saved to the IFS file system.

The file that contains your HTTP Web server configuration directives is now created.

## Creating an LDIF file

The easiest way to perform the initial publishing of the configuration, and also updates, is the approach of using LDAP Data Interchange Format (LDIF) files. These files contain information following a specific syntax that allow an LDAP server to publish certain information to the directory. In this scenario we create a file that is used to create, and later on updat, the Web server configuration directives in the LDAP directory as described in the next steps.

1. On your PC or 5250 session create a new text file using an editor such as Notepad (PC) or EDTF (OS/400).
2. Enter the following information into the new file:

```
dn: cn=ishopwebap,ou=config,o=iseriesshop
objectclass: container
objectclass: eProperty
cn: ishopwebap
binProperty:< file:///AS2318/Config/webconldap.txt
```

The first entry is the distinguished name (DN) and represents the name and location of the new entry within the directory. The next two entries specify the object classes the new entry belongs to. The cn (commonName) attribute must also be specified as it is a required attribute of the container object class. The binProperty attribute's value specifies that a file is to be published from the location specified. In this scenario, it is the webconldap.txt file that holds the server configuration directives.

3. Save the file. We saved the file under the name PublishConfig.ldif into the IFS directory /AS2318/config.

## Publishing the Web server configuration

Now that you have the Web server configuration directives stored in the text file webconldap.txt and the LDIF file PublishConfig.ldif, you can publish the configuration directives to the LDAP directory. In our scenario we used the ldapmodify utility in the OS/400 QShell to perform the publishing process.

1. From a 5250 session start the QShell by entering the following command:

```
qsh
```

Press Enter. The QShell Command Entry window is shown.

2. Depending on whether you publish the configuration for the first time to or update an existing configuration in the LDAP directory, you need to specify different switches for the **ldapmodify** command.

- First-time configuration

Enter the following command to publish the configuration for the first time:

```
ldapmodify -D "cn=administrator" -w "my5ldap"  
-f "/as2318/config/PublishConfig.ldif" -a
```

The QShell responds for a successful publishing with a message like:

```
$  
> ldapmodify -D "cn=administrator" -w "my5ldap" -f  
"/as2318/config/PublishConfig.ldif" -a  
adding new entry cn=ishopwebap,ou=config,o=iseriesshop
```

```
$
```

- Updating an existing configuration

Enter the following command to update an existing configuration:

```
ldapmodify -D "cn=administrator" -w "my5ldap"  
-f "/as2318/config/PublishConfig.ldif" -r
```

The QShell responds for a successful publishing with a message like:

```
$
```

```
> ldapmodify -D "cn=adminstrator" -w "my5ldap" -f  
"/as2318/config/PublishConfig.ldif" -r  
modifying entry cn=ishopwebap,ou=config,o=iseriesshop  
  
$
```

This command creates or updates an object ishopwebap as specified in the PublishConfig.ldif file and uploads the HTTP configuration from the webconldap.txt file. The major difference between publishing for the first time or updating an existing entry is the -a (add) and -r (replace) command switches. The switches and their meanings, as used in this scenario, are as follows:

<b>-D</b>	The DN used to bind to the LDAP directory. In our scenario is it the user DN (cn=adminstrator) to access the LDAP server.
<b>-w</b>	The password of the user to access the LDAP server.
<b>-f</b>	Reads the entry modification information from an LDIF file instead of the standard input.  /as2318/config/PublishConfig.ldif is the path to the LDIF file used.
<b>-a</b>	Used on the <b>ldapmodify</b> command to indicate that the utility tries to add entries by default. Using this parameter is the same as using <b>ldapadd</b> .
<b>-r</b>	Specifies that <b>ldapmodify</b> tries to replace existing values by default. This parameter is used if you update the existing object.

The Web server configuration is now published and available to be included by one or more HTTP servers in the network.

## Setting up a new production server instance

Perform the following steps to set up the HTTP Server for iSeries (powered by Apache) to use the LDAP support for loading configuration directives:

1. From a Web browser access the HTTP Administration window on your iSeries server by using the following URL:

<http://rallyas4b:2001/HTTPAdmin>

The HTTP Server Administration and configuration main window is displayed. The HTTP server configuration utility requires that the HTTP \*Admin instance is up and running. You can use Operations Navigator (TCP/IP servers) or the following command to start the \*Admin instance:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

2. Select the **Administration** tab and then click **Create HTTP Server** on the navigation pane, as shown in Figure 6-28.

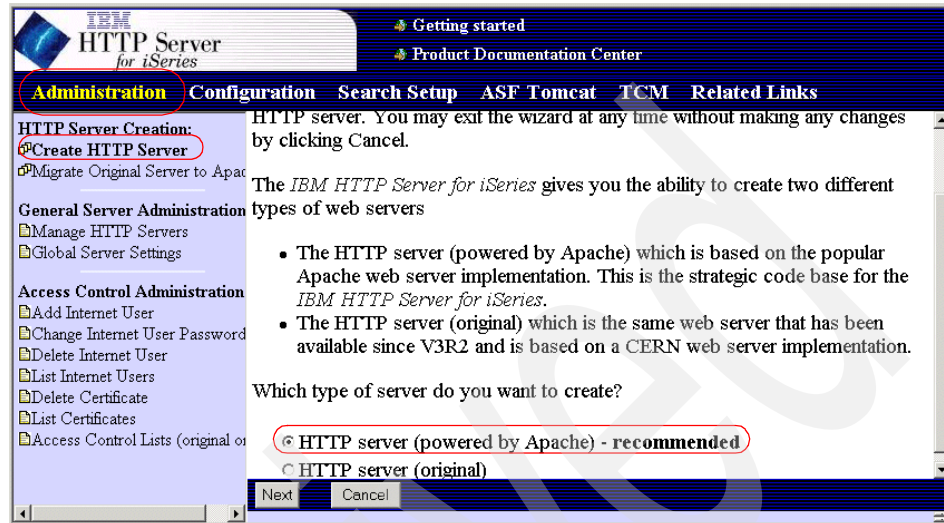


Figure 6-28 Create HTTP server window

3. Select **HTTP server (powered by Apache) - recommended** and click **Next** to continue with the server instance creation wizard.
4. On the next window enter your HTTP server name **WEBCONLDAP** and click **Next**.
5. Select **No** for the questions on whether you want to create this instance based on an existing instance and click **Next**. The server root is displayed in the following window.
6. Specify the IFS path where the server instance root will be. We left it as the default **/www/webconldap**. Click **Next** to specify the document root.
7. Specify the default document root for the new instance. We also left it as the default value **/www/webconldap/htdocs**. Click **Next** to continue.
8. The next wizard question prompts you for the IP address and port the server should listen on for requests. In our scenario we left the IP address as All addresses and the port as 80. According to our implementation scenario, shown in Figure 6-25 on page 271, the test server listened on port 8800 and the production server on port 80. Click **Next**.
9. Select how you want the server to handle logging. In this scenario select **Combined log file**. Click **Next** to display the configuration summary, as shown in Figure 6-29 on page 279.



Figure 6-29 Create HTTP Server summary window

10. Click **Finish** and the wizard creates the HTTP server instance according to your input.
11. After the instance is created, a new window is displayed allowing you to manage or configure the instance. Click **Configure** to continue. The server instance configuration window is displayed. Make sure that the name of your newly-created instance is displayed in the Configuration for server drop-down list.
12. In the Configuration structure pane select the context **Directory/**.

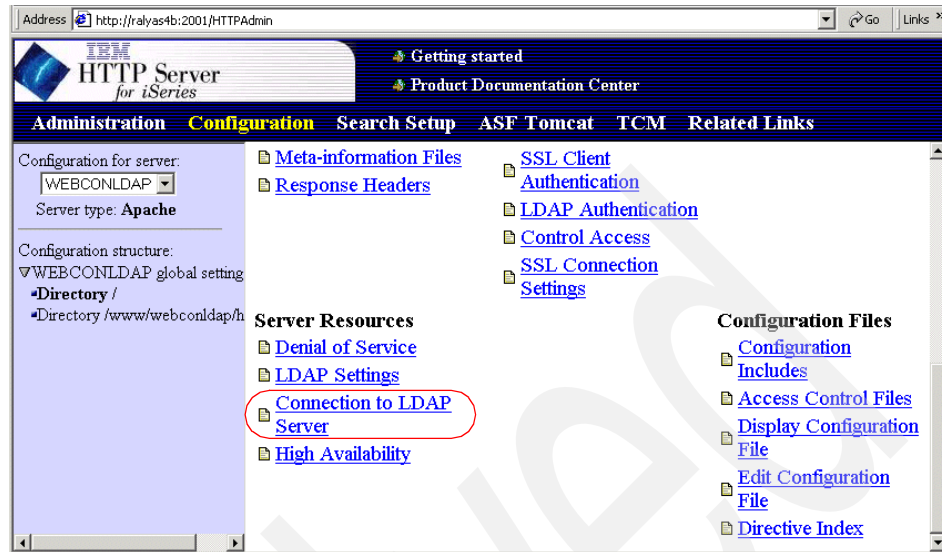


Figure 6-30 Main configuration window

13. Click **Connection to LDAP Server** and the window seen in Figure 6-31 is shown. This option provides various LDAP connection authentication options.

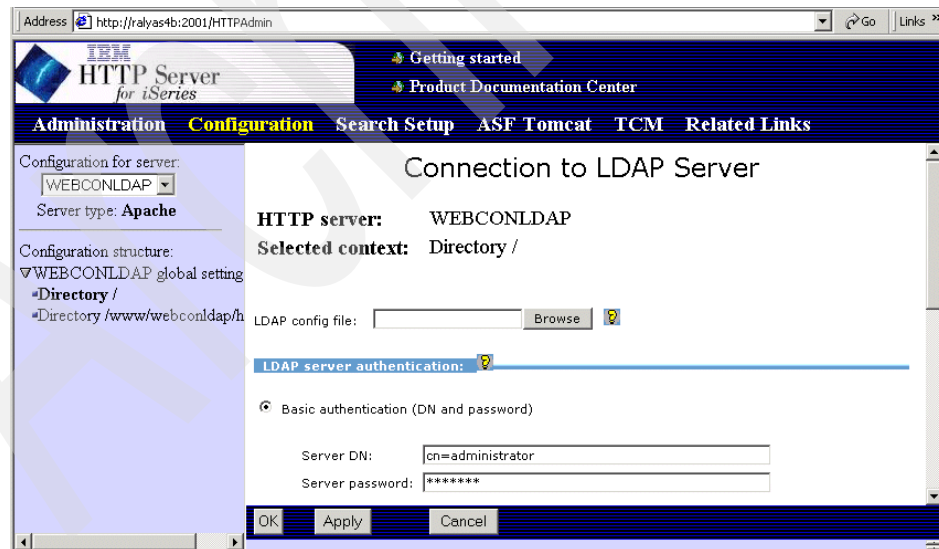


Figure 6-31 Connection to LDAP Server window

14. Scroll down to the LDAP server authentication section and select **Basic authentication (DN and password)**. In this section you provide the credentials the HTTP server uses to authenticate to the LDAP server that holds the configuration. Enter the following values:

**Server DN**                      cn=administrator

**Server password**          Password of DN specified on the Server DN parameter

15. Click **Apply** and then **OK** to continue. This will create the `ldap.application.password.stashFile`  
`/QIBM/UserData/HTTPSVR/LDAP/WEBCONLDAP/1013715647416.stash` directive in the selected directory context. The stash file created contains an encoded form of the password entered and is used by the HTTP server when authenticating to the LDAP server. The stash file is always created in the `/QIBM/UserData/HTTPSVR/LDAP/xyzserver` IFS path where `xyzserver` is the name of your HTTP Web server instance. The stash file name varies. Note that the stash file will not be used within the server configuration itself. It is rather used in the LDAP configuration file (`ldap.prop`), which will be created within the next steps. However, the only way of creating the stash file is through the steps we describe in this section, thus the way is via the GUI. As an alternative to creating the stash file on each production server, you can also transfer the file from your test server to the production server and refer to the transferred file in your remaining configuration steps.
16. Before we continue with the HTTP server configuration you have to create a LDAP configuration file (also referred to as the LDAP properties file). This file contains information about where the LDAP server is, how to access the LDAP server, authentication information, and so forth. If you take a closer look at the directives used in this file, you may notice the resemblance between these directives and the directives in the server configuration. All of these directives can be used by the HTTP server for authentication and most of them for the configuration support. However, the configuration support requires these directives in the LDAP configuration file. The plan for future enhancements for the HTTP Server for iSeries (powered by Apache) is to configure the entire LDAP support via a LDAP configuration file. As seen in Figure 6-25 on page 271, the `LDAPInclude` directive points to the path and file name of the LDAP configuration file that is used when the server starts up. This means that the LDAP configuration file can be anywhere in the IFS, but we recommend storing it in the conf directory of your server root directory `/www/webconldap/`. For your convenience, IBM has provided a sample LDAP configuration file called `ldap.prop`. This file can be used as a template and is stored in the `/QIBM/ProdData/HTTPPA/conf` directory. Copy this file into your HTTP server conf directory `/www/webconldap/conf/`. You can do this via a 5250 session or via Operations Navigator. In our scenario we copied the file

using the following command and used an ordinary ASCII editor to change the ldap.prop file:

```
CPY OBJ('/QIBM/ProdData/HTTP/conf/ldap.prop')
      TODIR('/www/webconldap/conf/')
```

The following directives show directives of an LDAP configuration file:

```
# LDAP HTTP Configuration property file
ldap.URL ldap://rallyas4b:389/ou=config,o=iseriesshop
ldap.transport TCP
ldap.application.authType basic
ldap.application.DN cn=Administrator
ldap.application.password.stashFile
/QIBM/UserData/HTTPSVR/LDAP/WEBCONLDAP/1013715647416.stash
ldap.idleConnection.timeout 600
ldap.waitToRetryConnection.interval 30
ldap.search.timeout 10
ldap.cache.timeout 600
```

Following is an explanation of the most important directives that you need to configure:

#### **ldap.URL**

The ldap.URL directive tells the HTTP server the location of the LDAP server that is being used for authentication or configuration. You specify the host name (rallyas4b) of the LDAP server. This can be the DNS name or the IP address used to identify the host where the LDAP server resides. The port is optional. If not specified, port 389 will be assumed if using TCP/IP connections, and 636 will be used for SSL connections to the LDAP server. The BaseDN (ou=config,o=iseriesshop) provides the starting point for searches of the LDAP directory.

#### **ldap.transport**

The ldap.transport directive is used to specify the transport protocol used to communicate with the LDAP server. The LDAP server can communicate over either TCP/IP or SSL connections.

#### **ldap.application.authType**

The ldap.application.authType directive is used to specify the method used to authenticate the HTTP server



application to the LDAP server. The possible values are None and Basic. If None is selected, the HTTP server connects using anonymous access, if permitted by the LDAP server. If Basic authentication is chosen, the HTTP server is required to identify itself to the LDAP server by using a distinguished name and password. In this scenario we assume that the server configuration directives are not accessible via anonymous access. Therefore we selected the Basic authentication type.

### **ldap.application.DN**

This directive specifies the distinguished name (DN) the HTTP server uses to authenticate to the LDAP server. The DN specified needs to have access to the server configuration. In this scenario the server configuration is published under `cn=ishopwebap,ou=config,o=iserieshop` in attribute `binProperty`.

When using `ldap.application.authType Basic`, the directive `ldap.application.password.stashFile` should be used with `ldap.application.DN`. Unless the LDAP server allows anonymous access, the connection between the HTTP server and the LDAP server will not be made without a valid password.

**ldap.application.password.stashfile** The directive specifies the file that contains the encoded password used by the HTTP server to authenticate to the LDAP server when `ldap.application.authType` is Basic. The configuration tools create, encode, and name the filename as described in step 13 on page 280 to step 15 on page 281. The value of this

directive can be copied from the server configuration by clicking Display Configuration File from the Configuration Files section in the HTTP configuration interface.

**Note:** At the time this redbook was written, the sample ldap.prop file contained a comment telling us to use the `ldapstash` command to create the stash file. However, this command is not shipped with OS/400. We were assured by the Development Team that this comment will be removed from the sample file in the future.

The remaining parameters are for timing specifications and we left them at their default values.

17. When you have finished configuring these directives, save the ldap.prop file to the conf directory of your HTTP server.
18. Now we continue with the HTTP server configuration as shown in Figure 6-32.

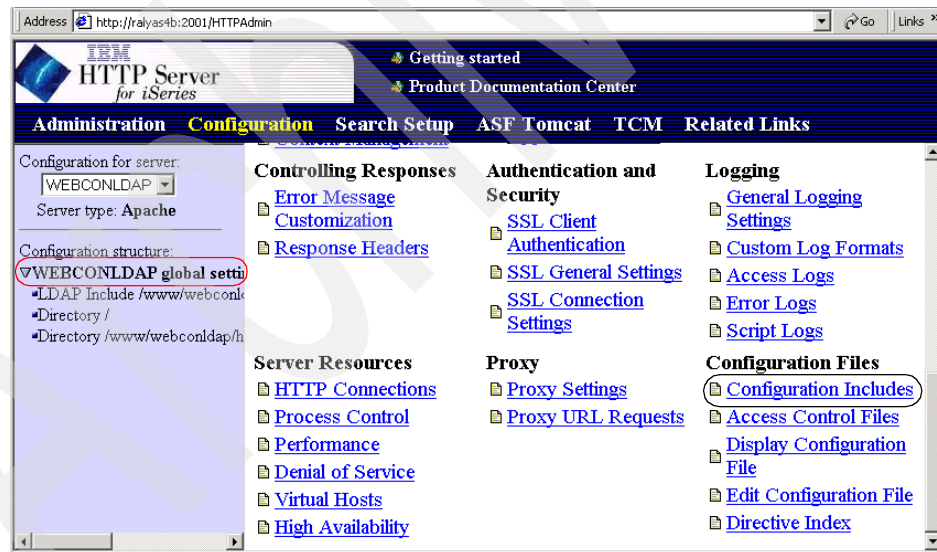


Figure 6-32 Main configuration window

19. From the Configuration structure pane select **WEBCONLDAP global settings**.
20. Click **Configuration includes** in the Configuration Files section to configure your configuration include directive.

21. In the Configuration Includes window scroll down to the Include information stored on an LDAP server section and click **Add**. The window shown in Figure 6-33 is displayed.

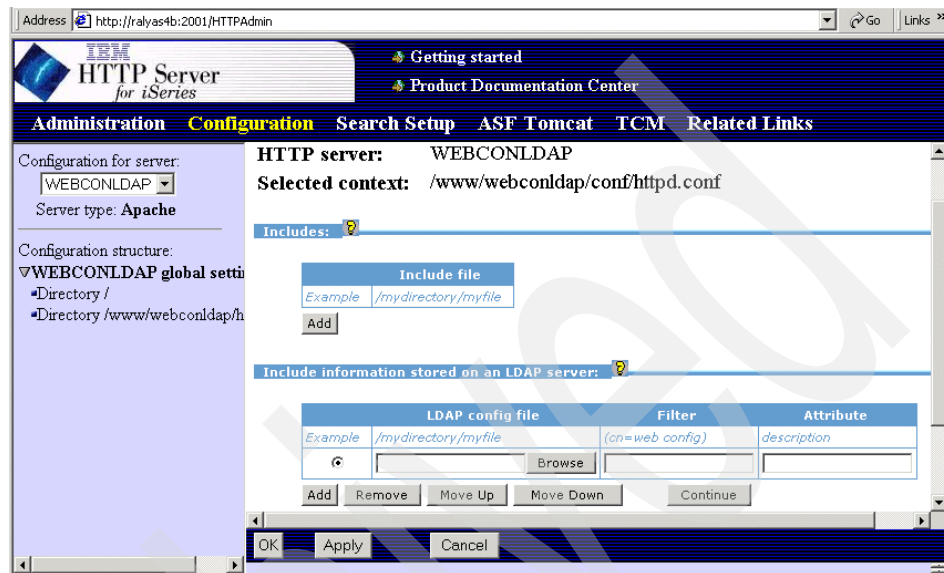


Figure 6-33 Configuration Includes window

22. Click **Browse** on the LDAP config file column. A new File Browse window pops up, as shown in Figure 6-34 on page 286. Select the path where your ldap.prop file is located. In this case it is the path /www.webconldap/conf.

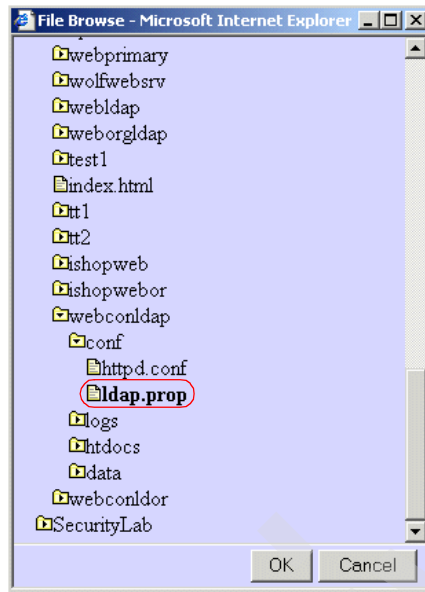


Figure 6-34 Prop File Browse window

23. Select the ldap.prop file and click **OK**. You return to the Configuration Includes window, as shown in Figure 6-35 on page 287, with the LDAP config file value filled in.

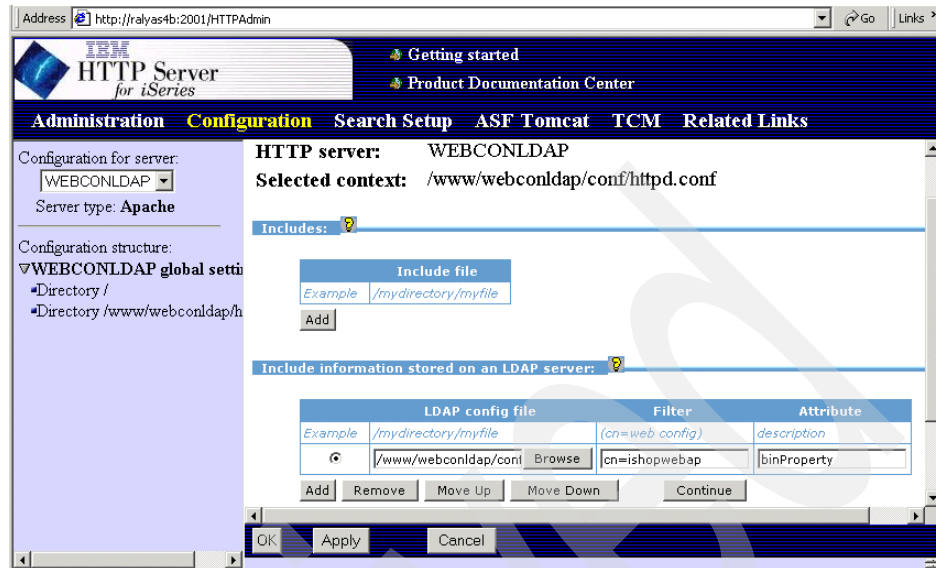


Figure 6-35 Configuration Includes window

The path to the LDAP config file is entered. In the remaining parameters enter the following values:

#### Filter

cn=ishopwebap

This is the object in the LDAP DIT that holds the configuration directives. Note that this is only the relative distinguished name (RDN), not the fully-qualified DN. The search base is configured in the ldap.URL directive in the ldap.prop file.

#### Attribute

binProperty

This value contains the attribute name within the object specified in the Filter parameter that contains the HTTP server configuration directives. Note that the LDAP directory does not exist yet. It will be created later in this chapter.

24. Click **Continue** and the window will reappear with the data you entered shown.
25. Click **OK** to close the Configuration Includes window and return to the main configuration menu. The LDAPInclude directive has been added to the configuration and can be seen in the left pane below WEBCONLDAP global settings.

At this point, the production server instance is created containing the following:

- ▶ LoadModule directives for LDAP modules
- ▶ LDAPInclude directive including a reference to the created ldap.prop file
- ▶ LDAP authentication directives
- ▶ Directives added via the instance creation wizard

As you can imagine, most of the directives added by the instance creation wizard are common to all server instances in a company. This means that they are already configured on the test server instance and you do not need them in the production server since they are loaded from the LDAP directory. The following steps should be done to clean up the production server instance from directives that will be loaded from the LDAP directory.

26. From the HTTP server configuration main menu click **Edit Configuration File** in the Configuration Files section.

27. Remove all directives that are not unique to this particular server instance and not required by the LDAPInclude directive. In our scenario we removed all directives except the LoadModule directives for the LDAP modules, the LDAPInclude directive, and the Listen directive that is different from the one in the test instance. After the clean-up you see a configuration like the one shown in Figure 6-36.

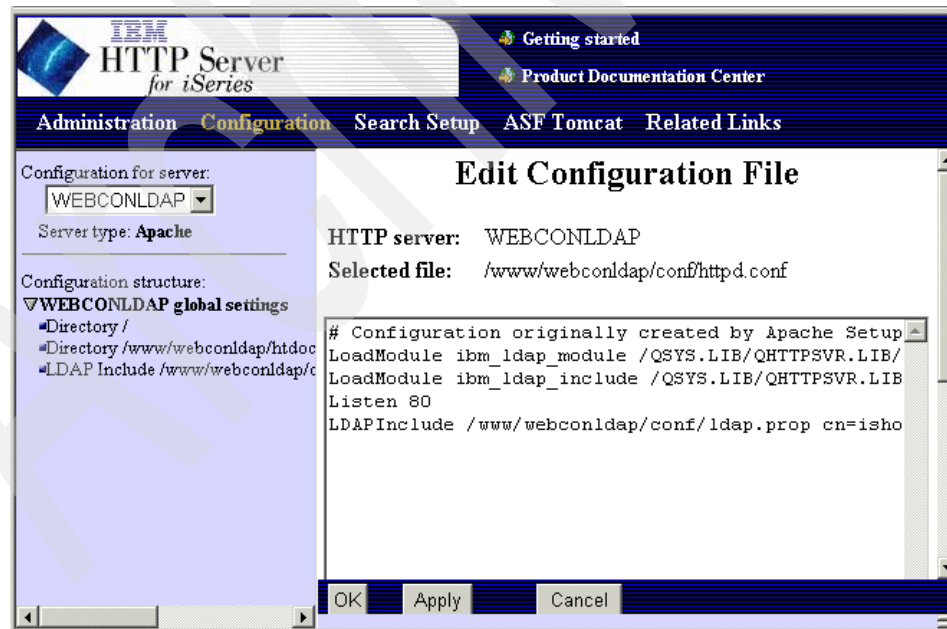


Figure 6-36 Edit Configuration File window

28. After you finish the clean-up, click **OK** to save your changed configuration. Note that you cannot test the configuration at this time because the configuration from the test server has not been published yet.

### Starting the production server instance

At this point in the implementation phase, the server configuration as exported from the test instance is published to the LDAP server and available to be included at server start-up time. A production server instance has been created that only contains the necessary directives to include server configuration directives from an LDAP directory.

You can now start your production server. During start-up, the HTTP server binds to the LDAP server, and retrieves and includes the configuration directives stored in the binProperty attribute.

**Note:** When you use the HTTP Administration interface and display the HTTP server configuration of your production server instance, you will not see the directives included from the LDAP directory. If you need to troubleshoot a problem, you may want to use the -vv switch in the instance startup parameter when starting the HTTP server instance. This will cause the server job to create a spoolfile containing trace information when the server instance is ended. You can also use the DMPUSRTRC command to create a physical file that contains the trace data while the server instance is still running. The entire process of binding to the LDAP server, searching the configuration entry, as well as processing the included directives, is listed in the spoolfile.





## Setting up LDAP on Domino server for iSeries

LDAP on Domino for iSeries is one part of the Domino Directory services, which is a fundamental component of the architecture that delivers Domino's scalability, reliability, and standards-based flexibility.

Lotus Domino Release 5 provides a broad range of Domino-specific and standards-based Directory Services. These services combine to greatly simplify user and systems management, act as a container for a consistent set of data across an organization, and support Web-based and distributed applications across the corporate network.

Domino Directory services have evolved throughout the years, with continuous improvements and innovations, so that Domino Release 5 provides quite a flexible directory infrastructure. This evolution enables existing customers to continue to grow and exploit Domino Directory as their general-purpose directory, even as their enterprise grows in size and sophistication. The evolution also enables Domino and Domino-based applications to fit smoothly into the multi-directory environment present in many organizations today.

This chapter describes the functionality of LDAP on Domino for iSeries, the prerequisites for the installation, and the installation and configuration of the Directory services.

## 7.1 Domino Directory implementation and components

The directory architecture in Domino consists of several components, all based on the main Domino Directory as a central point, the public names and address book, and the names.nsf. When used to its full extent, the Domino Directory Service provides a robust capability equally capable of acting as either the center of an enterprise directory infrastructure or as a peer directory in a multi-directory environment.

The other components of the Domino Directory Services include:

- ▶ **Directory Catalog:** An aggregation of directories located on either the server or client.
- ▶ **Directory Assistance:** Provides access to federated directories that can include secondary Domino directories or third-party LDAP directories.
- ▶ **Domino LDAP server task:** Runs on the Domino server and provides LDAP Version 3-compliant access to Domino and third-party directories for both clients and applications.

### 7.1.1 The role of Directory Services in Domino

The Domino Directory services consist of a primary Domino Directory and any combination of the following components:

- ▶ One or more secondary Domino directories
- ▶ A server Directory Catalog
- ▶ One or more user Directory Catalogs
- ▶ A Directory Assistance database
- ▶ The Domino LDAP service

The following roles and functionality are supported by the Domino Directory services:

- ▶ The primary Domino Directory as a domain configuration store and centralized point of domain management

All certificates, connections, cross certificates, server configurations, and domain documents are maintained in the primary Domino Directory. This enables easy domain administration since the administrators only need to update the details in the Domino Directory in a single Domino Directory in the domain. Replication then ensures and the details are pushed to all the servers in the domain.

- ▶ Mail address lookup and resolution service using the Domino Directory services

Type-ahead functionality and name resolution are provided by any combination of primary and secondary Domino directories, server and user Directory Catalogs, and Directory Assistance, which include LDAP directories for referrals. The solution is focused on easier and quicker mail addressing. Depending on the scale of the solution this may also include dedicated directory servers to allow load balancing and a minimized impact on normal mail servers.

- ▶ User authentication and authorization using the Domino Directory services

The default Notes user authentication procedure via Notes remote procedure call (NRPC) is certificate-based. It is also possible to switch off certificate-based authentication for Notes users, which enables Notes users to remain anonymous until the user attempts to perform an operation that is not allowed to be performed by anonymous users. For non-Notes clients, there are three levels of security and authentication:

- Anonymous access
- Name and password authentication
- Certificate-based authentication

For more information, see Section 2.6 in *Getting the Most from Your Domino Directory*, SG24-5986.

- ▶ Domino Directory services as a user information store and central point of information management

Depending on the requirements, this may be more than just managing the primary Domino Directory as a centralized information store. In the primary Domino Directory, this includes management of people and group information, and it can also include resource information. If Domino is used as the hub in an enterprise directory solution, this may also include using Domino as the centralized point for managing secondary directories.

- ▶ Referral services to other directories from the primary LDAP directory

You use Directory Assistance to refer LDAP Lotus Notes clients that connect to the Domino LDAP service to another LDAP directory. This only happens if the search in the Domino LDAP service's primary Domino Directory and all secondary Domino directories configured in Directory Assistance are not successful. For more information on the referral services, see Section 7.1.4, "Directory Assistance" on page 297.

Domino's Directory Services represent a unique solution for each deployment or environment, since each organization will use a concatenation of Domino Directories, Directory Catalogs and Directory Assistance with one or multiple LDAP directories, according to their needs.

## 7.1.2 The Domino Directory

The Domino Directory is the central store for directory information used by the Domino servers and by clients accessing applications and services hosted on those servers within a Domino domain. The first server in a domain creates the first instance of the Domino Directory as it is initially installed. Each subsequent server in the same domain creates a replica copy of this same directory as part of its installation process. These copies remain synchronized through periodic replication across the domain, providing a robust and distributed directory architecture.

The information maintained in this primary domain directory can be broadly divided into two categories. The first category is information associated with the configuration of the Domino servers, the domain they reside in, the services they provide, and how they interrelate. The second category is information about the people and groups who access those servers and use those services and applications provided. The configuration information is primarily of interest to Domino administrators, who will maintain and update it to ensure smooth operation of the environment. The people/group information is of much wider interest and can be made accessible to a wide range of users, developers, and applications. These applications can be either hosted on Domino servers or can be legacy applications hosted on a wide variety of platforms and systems.

From the perspective of the Domino server and applications, both types of information are stored in a series of documents, which are organized in views and displayed using forms, with individual items of information contained in fields. For a list of the document types, how to configure and use them, and some real world examples, see Appendix C, “Domino Directory forms” in *Getting the Most from Your Domino Directory*, SG24-5986. The folders and views are provided in a format that makes sense to the users of the directory. The normal user uses the people and groups views only. The rest of the default views, including hidden views, are used by administrators and the Domino server.

**Important:** Do not make any changes to the hidden views in the Domino Directory. They are needed in their default format by the Domino server and clients.

**Tip:** Be very careful when adding additional views to the Domino Directory. Every view adds to the size of the directory and workload to each server, because each server has to maintain the view indexes.

As an example, the IBM UK address book has a default size of 398 MB with 33,000 registered users. The default People view is 15 MB in size and the \$Users view is 71 MB. So you can see how quickly the directory can grow with additional views.

The Domino Directory is also an LDAP Version 3-accessible directory store. This means that the same information can be viewed as being held in an LDAP schema and consists of a series of object classes and attributes accessible via LDAP search and manageable through LDAP add, delete, modify, and rename operations and LDAP-compliant tools. The Domino LDAP schema includes standard schema elements, such as Person, OrgPerson, and inetOrgPerson, and can be extended using the Domino designer. The schema can be published using either LDIF commands or into a predesignated Domino database.

### 7.1.3 Directory Catalog

The Domino Directory Catalog is a specialized database populated with entries from one or more Domino directories. The information contained in the Directory Catalog is controlled by the Domino administrator using the configuration document in the database. The entries are created by the Directory Catalog task, a server task running in the background on the Domino server.

The primary purpose of the Directory Catalog is to provide a lightweight, quick access store of directory information primarily for use by mobile and disconnected users. It achieves this goal through a combination of three specific methodologies. These include selection of the information to be included, aggregation of the information into consolidated documents, and elimination of most indexed views.

There are two different types of Directory Catalogs that you can deploy, user Directory Catalogs and server Directory Catalogs. They both have specific roles in the Domino Directory services.

#### User Directory Catalog

The User Directory Catalog is also called the *mobile Directory Catalog*, which can be used by Notes users as local replicas to enable them to do quick mail addressing to anyone in the organization, even in disconnected mode.

The functionality and advantages of the user Directory Catalog are:

- ▶ Notes users use local replicas of a user (mobile) Directory Catalog to enable them to do quick mail addressing to anyone in the organization, even in disconnected mode.
- ▶ Users can use encrypted mail in disconnected mode. When sending an encrypted memo, it is flagged for encryption. At the next server connection, when the Notes client sends the mail item, the client looks up the public key and encrypts the mail on the fly. This is referred to as “just in time” or “on the fly” encryption.
- ▶ Group names can be included in the catalog, so users can address mail to groups. When mail is sent, the group lookup happens on the server and the router includes all the group users in the recipient list.
- ▶ By using the LDAP protocol, users can search in the Directory Catalog the same way that they search a personal address book.
- ▶ Users can use the address assistant dialog box to open and scroll through the names in the Directory Catalog.
- ▶ Network traffic is reduced because most of the name resolution occurs locally on the workstation, rather than on a server.
- ▶ If the soundex is enabled and you are not sure how to spell a person’s name or surname, you can just guess it as close as possible. The Directory Catalog will work through all the entries and will return a list of possible matches.

## **Server Directory Catalog**

The functionality and advantages of the server Directory Catalog are:

- ▶ Notes users can do very quick name searches in secondary Domino directories on the server.
- ▶ If a server Directory Catalog is enabled on the user’s mail server, Notes clients without user Directory Catalogs can use the server Directory Catalog to address mail and browse directory entries.
- ▶ The mail router can look up addresses more quickly in a server Directory Catalog instead of using Directory Assistance to look up the addresses in multiple, individual secondary Domino directories.
- ▶ If you set up Directory Assistance, the LDAP service can use the Directory Catalog and Directory Assistance together to process LDAP searches, providing the functionality of both.
- ▶ The server Directory Catalog can be the central point for all organization level directory access, for applications, LDAP clients, and other components in the Domino environment.

- ▶ The server Directory Catalog is also used in the Web client authentication process. If the client exists in a Directory Catalog and not in the primary Domino Directory, the server uses the information available through Directory Assistance to very quickly access the complete entry in the secondary Domino Directory. This is possible because each entry in the Directory Catalog includes the replica ID of the Domino Directory from which the entry was derived, and the unique ID associated with a replicated document.
- ▶ If the soundex is enabled and you are not sure how to spell a person's name or surname, you can just guess it as closely as possible. The Directory Catalog will work through all the entries and return a list of possible matches.
- ▶ When the mail router uses the Directory Catalog, it performs an exhaustive lookup of all entries in the Directory Catalog, even if the Exhaustive lookup router configuration option is disabled, because the router can do the exhaustive lookup quickly in one database.

The Domino administrator can configure the Directory Catalogs to suit the organization's needs. A typical setup is for an organization to use more than one Directory Catalog. The configurations of a user Directory Catalog and a server Directory Catalog are slightly different because of their different roles. To find out more about how to configure the Directory Catalog, see Section 4.4.2 in *Getting the Most from Your Domino Directory*, SG24-5986.

### 7.1.4 Directory Assistance

Domino Directory Assistance is the third major component of Domino Directory services. Its primary purpose is to provide the mechanism to federate one or more secondary directories, making them transparently accessible to directory users. These secondary directories can be either Domino Directories, the IBM SecureWay Directory Services or third-party LDAP-compliant directories such as Netscape's iPlanet or Novell's NDS. Directory Assistance supports the following functionality:

- ▶ Finds entries in secondary Domino directories and LDAP directories on behalf of Notes users for mail addressing.
- ▶ Finds entries in secondary Domino directories on behalf of LDAP clients.
- ▶ Refers LDAP clients to LDAP directories.
- ▶ Uses name and password security to authenticate Web clients registered in secondary Domino directories.
- ▶ Uses x.509 certificates to authenticate Web clients registered in secondary Domino directories.
- ▶ Authenticates Web clients registered in LDAP directories.

- ▶ Expands groups for authorization of Web users in a single selected LDAP directory.
- ▶ Provides failover to another replica of a secondary Domino Directory.
- ▶ Uses naming rules to efficiently search secondary Domino directories.
- ▶ Provides support for Recipient name type ahead addressing.
- ▶ Works in conjunction with a Directory Catalog on a server.

Directory Assistance provides the administrator with a great deal of flexibility in establishing parameters for the use of the federated directories. These controls include:

- ▶ Determining which of the LDAP directories will be used for group expansion for authentication
- ▶ Identifying name matching rules that can direct a search to a particular directory for resolution
- ▶ Determining which directories can be trusted to provide legitimate credentials for authentication processing
- ▶ Providing its own credentials for performing authenticated binds to secondary directories

All of these characteristics are determined by settings on Directory Assistance documents maintained in a Directory Assistance database on the Domino server.

When discussing Directory Assistance, there are two terms that need some explanation. They are very similar, but have definite differences. They are defined in RFC2251 paragraphs 4.1.11 and 4.5.3:

- ▶ A referral is returned when the target host does not contain the requested entry itself, but does have knowledge of a host that may contain that entry. The search result returned to the client contains one or more referrals to that other directory or directories. The client then uses the referral to look in the new target directory.
  - To return a referral, the Domino LDAP service never connects to and searches the LDAP directory.
  - Instead, the service uses information in the Directory Assistance document to return a referral.
  - As a result, the client only needs to make one call to either the master or the replica.
  - The referral includes the URL host name for the LDAP directory server, the base distinguished name configured for the directory, and the port the LDAP directory uses.



- A continuation reference is returned when the target host holds some, but not all, of the subtree that is targeted by the search request and has knowledge of a host that may contain additional results. In this case, the search result returned to the client contains that portion of the response the original host does hold and a continuation reference to the other directory or directories that may hold additional information relevant to the request. To successfully complete the search, the client must be able to follow the continuation reference.

A referral is returned when a server is contacted by an LDAP client using a base object that is not contained by the server. Search references are a mechanism for partitioning the directory information tree (DIT) among multiple servers, and allowing a search that has started on one server to be continued over one or more platforms.

Domino Directory services support providing referrals in response to queries. Directory Assistance is the mechanism used to define the referrals provided to the client. Domino Directory services in Release 5.x do not provide continuation references. The ability to follow continuation references was added to the Notes client in Release 5.0.5.

### 7.1.5 Different ways to do directory searches

Within the rich Directory Services environment provided by Domino, it is useful to understand the order in which directories are searched and the various behaviors that are followed when addresses are resolved and identities are authenticated. While there is a consistent overall pattern, the specific behaviors differ depending on what components are present and how various parameters are set. This section attempts to outline the general pattern, what parameters affect that pattern, and to provide some specific information on mail address resolution and authentication.

Before that discussion begins, there is one area that must be covered that may help clarify a number of user issues. This is the difference between the type ahead/type down functionality and the name resolution process invoked by pressing F9 or initiating a mail send.

The first point is that these are two distinct and separate functions with different design points and different behavior patterns. They are not intended to work the same and they do not provide identical results.

Type ahead/type down is a client-based function designed to provide a user addressing a message with a level of assistance in rapidly addressing a message. It is more limited in capability than F9 or mail send. For example, type ahead does not search LDAP directories. If a user Directory Catalog is

configured on the client, type ahead will not extend its search to the server when looking for names or providing a list of potentially ambiguous names. F9 is designed to invoke the same NameLookUp code and name resolution processes that will be used by the router when attempting to transmit the message. It is a more robust and extensive functionality designed to eliminate mail address resolution issues to the maximum extent possible. It searches LDAP directories identified in Directory Assistance and searches both client and server-based directories. It generally returns a longer list of potentially ambiguous names than type ahead, from a wider range of sources. The rest of this section focuses primarily on how this more extensive process functions.

Users should be made aware of the differences and should be educated that while type ahead provides an excellent tool to assist in addressing messages, if there is any question in their minds, they should press F9 to invoke a more complete search of all directory sources.

### **Directory search order**

The order in which directories are searched, especially when you are using a combination of Domino directories, Directory Catalogs, and Directory Assistance, can be difficult to sort out. Domino uses slightly different procedures depending upon the combination of directories you are using.

At a high level, the search sequence is always the same regardless of whether the search is initiated by a Notes client, a Web client, or another LDAP process. The general search order is outlined in the following list. The first two items apply specifically to Notes clients, while the last three apply to all clients. If any of the directory components are not present, that step in the lookup sequence will just drop away. Later in this section two specific cases are outlined in more detail.

1. The personal address book on the client machine
2. The user Directory Catalog on the client machine, if it is included in the notes.ini file in the names statement
3. The primary Domino Directory on the user's mail server or on the directory server, if that has been configured
4. The Directory Catalog on the user's mail server or on the directory server, if that has been configured
5. Directories listed in the server's Directory Assistance database that are not in the Directory Catalog

If the name entered in the mail addressing is a common name, not a hierarchical name, all directories in the Directory Assistance database will be searched according to their search order.

If the name entered is hierarchical, only the directories with rules that explicitly match the name, from most specific to least specific, will be searched. If two rules are equally specific, they will be searched in the search order specified in Directory Assistance.

## Settings affecting the search sequence

There are several settings that impact the search sequence used by F9 and the mail send function, as well as type ahead. These settings can be found in the location document, in the User Preferences dialogue, and in the Configuration settings document for the server. It is worthwhile to take a minute to review them.

On the mail tab of the location document there are two settings that impact type ahead and one that impacts the name lookup function invoked by F9 or mail send.

### ► Recipient name type ahead:

This setting provides three choices: Disabled, Local Only, and Local then Server.

The first, Disabled, is self-explanatory.

Local Only enables type ahead to search any address book listed in the Local address book field on the Mail and News panel of the User Preferences dialogue. This list can include the personal address book and one or more Directory Catalogs. The Directory Catalogs are searched in the order entered in the field.

Local then Server enables type ahead to extend its search to directories, Directory Catalogs, and secondary Domino directories configured in Directory Assistance that are located on the Domino server. By default, the server is the user's mail server unless a directory server is specified on the Servers tab of the location document.

**Note:** If the user configures a local copy of a Directory Catalog, then the type ahead functionality will ignore the Local then Server setting and restrict its operation to the client.

### ► Activate recipient name type ahead:

This setting provides two choices: On Each Character and On Delimiter.

- If On Each Character is selected, then type ahead attempts to find a match as each character is entered (there is a minimal delay before the search is started allowing additional characters to be entered). Type ahead presents the first match it finds in the first directory it finds that match and continues to refine those matches as each character is entered. If a Directory Catalog is configured, type ahead matches the

character sequence to the sort order for that Directory Catalog. For example, if the characters entered are Jo and the Directory Catalog is sorted by Last Name, it might present Jones, Jim, while if the Directory Catalog were sorted by Full Name it might present Joseph Smith.

There are two points to keep in mind here. First, if type ahead completes a name and you then enter a delimiter (you type Keith, it returns Keith Smith and you press comma), it only searches for ambiguous names that match Keith Smith; it does not present all the Keiths, even though that is all you physically entered. Second, if you get into the family of names, it is frequently easier to use type down to make the actual selection. Type down allows you to use the up and down arrow keys to scroll through the names in the selected directory that are just before and just after the one displayed.

- If On Delimiter is selected, type ahead is not activated until a delimiter such as a comma or carriage return is entered. This gives the user more control over the point at which name resolution is initiated. Also, the presented ambiguous name dialog will contain any name that is ambiguous based on the entry made to that point.

► Recipient name lookup:

This setting impacts the behavior of the name lookup function that is invoked when F9 or mail send is used. There are two choices here – Stop after first match and Exhaustively search all address books. Stop after first match causes the name lookup function to restrict its name resolution activities to be restricted to the directory in which it finds the first match for the value entered. For example, if the user enters John Smith and presses F9, and the first John Smith is found in the primary domain directory, then any further operations, such as searching for ambiguous names, are only applied to that primary domain directory. John Smiths that may occur in the Directory Catalog or in secondary directories accessed via Directory Assistance are ignored. If Exhaustively search all address books is selected, then all accessible address books are searched during name lookup operations.

**Note:** Type ahead is hard-coded to follow the same behavior as Stop after first match.

On the Mail and News panel of the User Preferences dialogue, the user can specify which local address books will be used by type ahead by including them in the Local address books field. As noted above, if the user configures a local copy of a Directory Catalog, then the type ahead functionality will ignore the Local then Server setting and restrict its operation to the client.

The third location where the address book lookup can be specified is in the Configuration Settings document for the server. The administration can impact the behavior of the router during its name resolution process. By default, the router has the option Exhaustive lookup disabled. This setting can be enabled, but it has a definite performance impact on the router. You can enable it via the Router/SMTP basics tab on a configuration settings document, as shown in Figure 7-1.

If enabled, when a recipient's address is looked up by the router, a search is performed across all available address books to insure that the recipient is unique among all Notes domains. This option can be used if there is the possibility that two users with the same address exist in multiple directories. Using this option increases the cost of delivering all messages to all recipients if there are multiple directories.

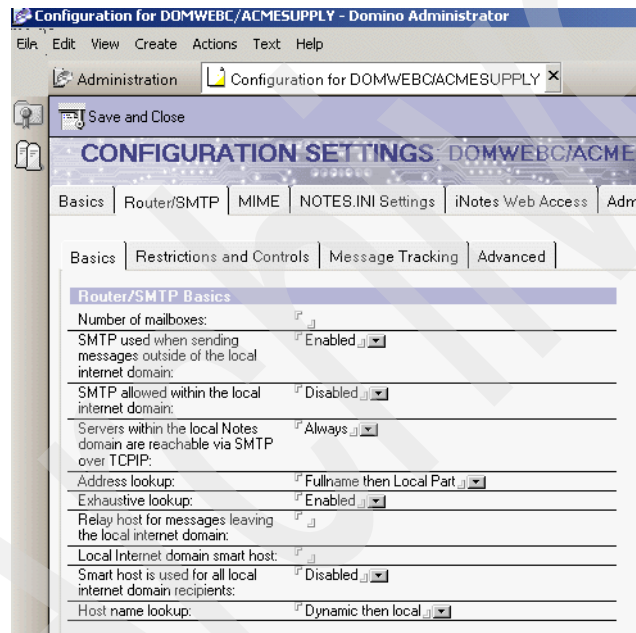


Figure 7-1 Configuring exhaustive lookup

**Note:** When the router uses the Directory Catalog, it performs an exhaustive lookup of all entries in the Directory Catalog, even if Exhaustive lookup is disabled. This is used as a default setting because the router can do the exhaustive lookup quickly in one database.

## **Search sequence for address verification in an LDAP directory**

If a mail item is addressed to a user that exists in an LDAP directory, Directory Assistance accesses the LDAP directory to verify the addresses. Address verification occurs only when the Notes user presses F9 or before the Notes client sends the mail. Domino does not use type ahead addressing to resolve the addresses of users in LDAP directories.

A server running the LDAP service searches directories in the same general order described earlier. However, there are some nuances at each step that it could be helpful to understand. The following list outlines the search sequence in more detail:

1. The primary Domino Directory on the server
2. Directory Catalog on the server  
If the LDAP user searches for an attribute that maps to a field that is not configured in the Directory Catalog, and a secondary Domino Directory is configured in the Directory Assistance database as well as in the Directory Catalog, the search continues to the complete entries in the secondary Domino Directory itself.
3. Domino directories defined in the server's Directory Assistance database that are not included in the Directory Catalog  
If an LDAP user specifies a search base, only Domino directories with assigned naming rules that correspond to the search base are searched.
4. If the search is not successful in any Domino Directory, the server can pick an LDAP directory enabled for LDAP clients in the Directory Assistance database to refer clients to and the clients can then connect to the directory server themselves.
  - If an LDAP user specifies a search base, the server picks an LDAP directory enabled for LDAP users with a naming rule that matches the specified search base.
  - If there is no such directory, the server does not return a referral.

## **Search sequence for authentication for a Web client**

As with address verification in an LDAP directory, the server follows the same general search order when a Web user gets authenticated by a Domino Web server. The following list outlines the specifics of this process:

1. The primary Domino Directory on the server

## 2. Directory Catalog on the server

- If the server finds the name, it refers to the Directory Assistance database to determine if the Domino Directory from which the name came is configured with a naming rule trusted for authentication that matches the username.
  - If the name is found to be trusted, the server looks up the HTTP password or x.509 certificate in the secondary Domino Directory that the Directory Catalog entry has been derived from.
  - If an HTTP password is stored in the Directory Catalog itself, the server looks up the password in the Directory Catalog rather than in the Domino Directory. But it does not authenticate the client if it does not find a trusted matching naming rule for the Domino Directory in the Directory Assistance database.
3. All other directories defined in Directory Assistance that are not included in the Directory Catalog and that have a naming rule that is trusted for authentication that matches the Web username

If there is more than one directory assigned in a trusted naming rule that matches the username, the directory with the most specific matching rule is searched first.

For further information about the search functionality in Domino, see Section 2.5 in *Getting the Most from your Domino Directory*, SG24-5986.

### 7.1.6 Authentication for LDAP clients

Name and password authentication between a Domino server and any LDAP client is a very simple process. The client submits a name and password to the server, which checks the primary directory for a record with the same name and password.

Domino uses SASL automatically if SSL with client authentication is set up on the server and if the LDAP client supports the protocol.

For SSL client authentication, the Notes or Internet client must have:

- ▶ An Internet certificate issued from a Domino or third-party certificate authority
- ▶ A trusted root certificate for a Domino or third party certificate authority (Notes clients only)
- ▶ A cross-certificate for the Domino or third party certificate authority created from the trusted root certificate

The trusted root certificate is not necessary for Notes clients after you create the cross-certificate.

- Software, such as a Web browser or a Notes workstation, that supports the use of SSL

## 7.2 LDAP on Domino in the redbook scenario

This section shows you the role of the LDAP task on Domino for iSeries within our redbook scenario. In this stage, the iSeries Shop expands its business by acquiring the additional company ACME Supply.

### 7.2.1 Scenario characteristics

The iSeries Shop business is expanding into new product areas. They found ACME Supply, a company that is offering exactly the kind of business portfolio that fits into the growth plans of iSeries Shop.

ACME Supply is using Lotus Domino for iSeries Release 5.0.9 as a mail server. Their server is DOMWEBC in the ACMESupply domain. Both companies will use e-mail as the major communication platform. Since there are no plans at the moment to merge the complete ACME Supply company into the iSeries Shop, both companies will keep their own mail systems, including their own user directories. The problem now is, how can an employee of one company look up contact information, such as the e-mail address or phone number, of an employee of the other company.

A direct online connection to the other company's directory would be the best way to connect both directories, but due to network infrastructure constraints and since the information in the directories will not change very often, they decided to provide the required information directly in the other companies directory.

The role of ACME Supply in our redbook scenario is shown in Figure 7-2 on page 307.



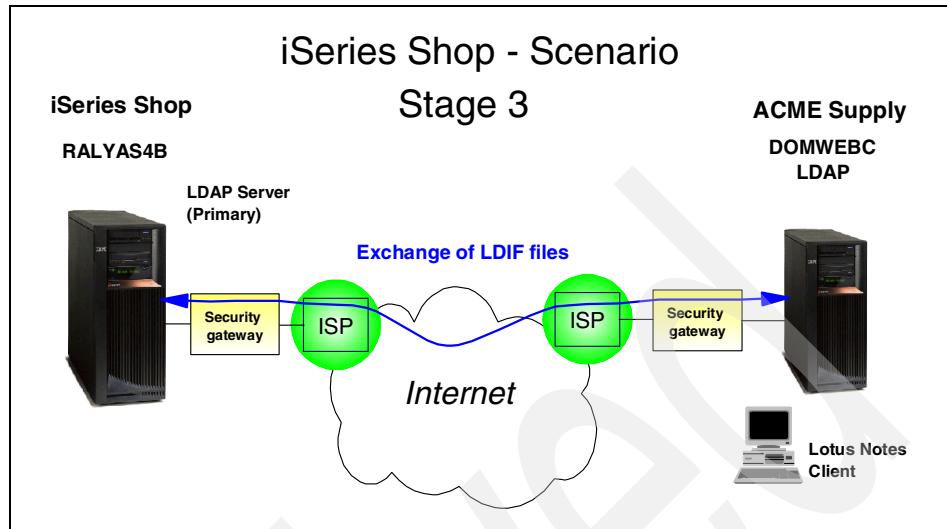


Figure 7-2 Redbook scenario stage 3: ACME Supply

## 7.2.2 Scenario objectives

The objectives of this stage of the scenario are:

- ▶ Configure LDAP on DOMWEBC to serve as an LDAP server.
- ▶ Show the export and import of directory information using LDIF files.

## 7.2.3 Installation prerequisites

This section describes the installation prerequisites for using the Domino LDAP on iSeries used in this scenario.

- ▶ OS/400 prerequisites:
  - 5722-SS1 - OS/400 V5R1
  - 5722-XE1 - Client Access Express for Windows
  - 5722-TC1 - TCP/IP Connectivity Utilities
  - 5769-LNT - Lotus Domino for iSeries Release 5.0.9
- ▶ Workstation prerequisites:
  - Microsoft Windows 95, Microsoft Windows 98, Microsoft Windows NT Version 4.0, or Microsoft Windows 2000 Professional
  - Lotus Domino Administrator Release 5.0.9
  - IBM SecureWay Directory Management Tool (DMT), which includes the ldapsearch, ldapadd, ldapmodify, and ldapdelete command utilities

For detailed information on installation and configuration of a Domino server on iSeries, please read *Lotus Domino for AS/400 R5: Implementation*, SG24-5592.

## 7.3 Configuring LDAP for Domino

The Domino Directory is the basis of information used within the Domino server. It is configured by setting up the Domino server itself.

The additional features of the Domino Directory Services as described in Section 7.1, “Domino Directory implementation and components” on page 292, can be set up separately according to the organizational needs of your enterprise.

Since the Directory Catalog will not be used in the scenario of this book, we do not describe it here. For detailed information how to set up a Directory Catalog for the Domino server, please refer to *Domino Administrator Help* (*help\help5\_admin.nsf*) and Section 4.4 of *Getting the Most from Your Domino Directory*, SG24-5986.

### 7.3.1 Setting up the Domino server for LDAP

Setting up the LDAP service to run on Domino only takes a few minutes. There are two ways to set up the LDAP service for a Domino server.

#### Configuring LDAP services for a new Domino server

To configure a new Domino for iSeries server, please refer to *Lotus Domino for AS/400 R5: Implementation*, SG24-5592. The following steps only describe the part of the configuration necessary to configure LDAP on Domino for iSeries:

1. On a 5250 emulation connected to the iSeries server where you want to configure the domino server, type **cfgdomsvr** on the command line and press Enter.
2. Type in all necessary parameters specifying your new Domino server.
3. For the parameter Directory Services type \*LDAP (see Figure 7-3 on page 309).

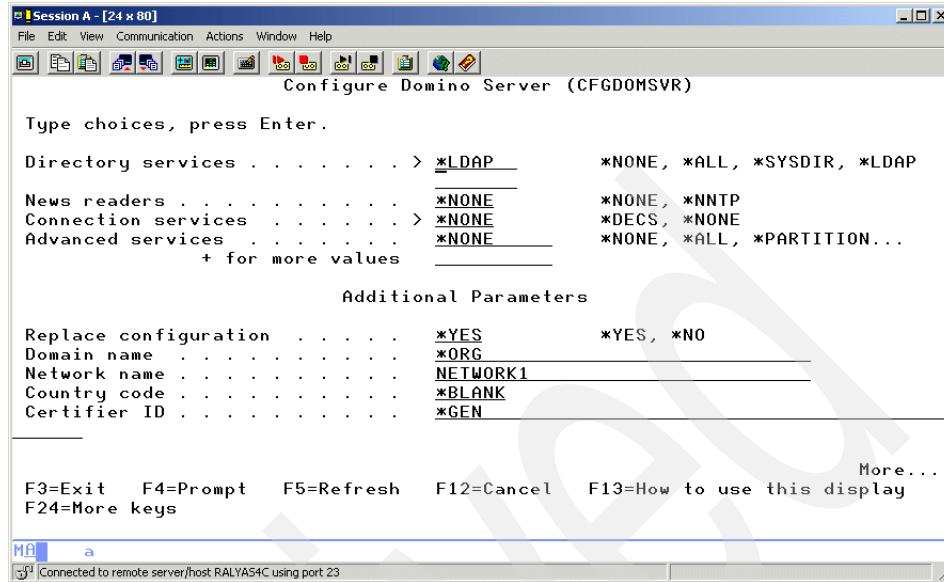


Figure 7-3 CFGDOMSVR for LDAP

4. Press Enter to set up the new server.
5. After the configuration has finished, you might start the Domino server; the LDAP task is automatically loaded and runs on your server.

## Configuring LDAP service on an existing Domino server

If you are adding LDAP services to an existing Domino server, all you need to do to get the service up and running are the following steps:

1. Open the Domino server console.
2. Type `load LDAP` and press Enter. This starts the LDAP task on the Domino server.
3. Wait for the following messages to appear in the Domino console:

```
LDAP Schema: Started loading...
LDAP Schema: Finished loading
```

4. In addition, you should modify the `ServerTasks=` line in `notes.ini` by adding LDAP to the list of tasks:

```
ServerTasks=LDAP,Replica,Router,Update,Stats,AMgr,Adminp,Sched,Cal Conn,...
```

This ensures that the LDAP service is automatically loaded at server startup.

**Note:** If you are adding the LDAP service to an existing partitioned server, you need to add the variable LDAPAddress to your notes.ini file, with the IP address for the partition that is running the LDAP server.

You can verify at any time if the LDAP task is running on your Domino server by typing **show task** on the Domino server console. To see if LDAP is running correctly, check to see if one of the output lines displayed shows the following line:

LDAP Server                      Listen for connect requests on TCP Port:389

The LDAP task is running and listening on port 389. It serves any LDAP requests connecting through this port.

### 7.3.2 Changes to the LDAP setup on Domino

To change any of the LDAP server settings, like the port selection, and to check if the authentication options are set properly, complete these steps:

1. Start your Domino Administrator client.
2. Select the **Configuration** tab.
3. Expand the **Server** section in the left-hand pane and click **Current Server Document**.
4. Select **Edit Server** to edit this server document.
5. Select the **Ports** tab in the server document.
6. Select the **Internet Ports** subtab.
7. Select the **Directory (LDAP)** sub-sub-tab and check for the following settings, as given in Figure 7-4 on page 312:
  - TCP/IP port number: This should be set to the port number your LDAP server is listening to. In our scenario this is the default port 389.
  - TCP/IP port status: Choose one of the following options:
    - Enabled (default) to allow LDAP clients to connect to the server over TCP/IP without using SSL.
    - Redirect to SSL to direct LDAP clients connecting without using SSL to use SSL instead. The LDAP service returns a message to LDAP clients indicating that they must connect over SSL.
    - Disabled to prevent LDAP clients that do not use SSL from connecting.

In our scenario we leave it as the default Enabled.

Authentication options:

- Name & password: If the TCP/IP port status field is set to Enabled, choose Yes (default) to allow LDAP clients to use name-and-password authentication to connect to the LDAP service using the TCP/IP port.

Choose No to prevent LDAP clients from using name-and-password authentication.

In our scenario, we select **Yes**.

- Anonymous: If the TCP/IP port status field is set to Enabled, choose Yes (default) to allow LDAP clients to connect to the LDAP service anonymously using the TCP/IP port.

Choose No to prevent LDAP clients from connecting anonymously.

If you allow anonymous connections, you can configure which fields anonymous LDAP clients can search. To configure the fields anonymous LDAP users can search, refer to *Domino Administrator Help* (*help\help5\_admin.nsf*) and Section 4.2.2 in *Getting the Most from Your Domino Directory*, SG24-5986.

In our scenario, we select **Yes**, to allow LDAP clients like the users of the iSeries Shop an anonymous access to the LDAP service.

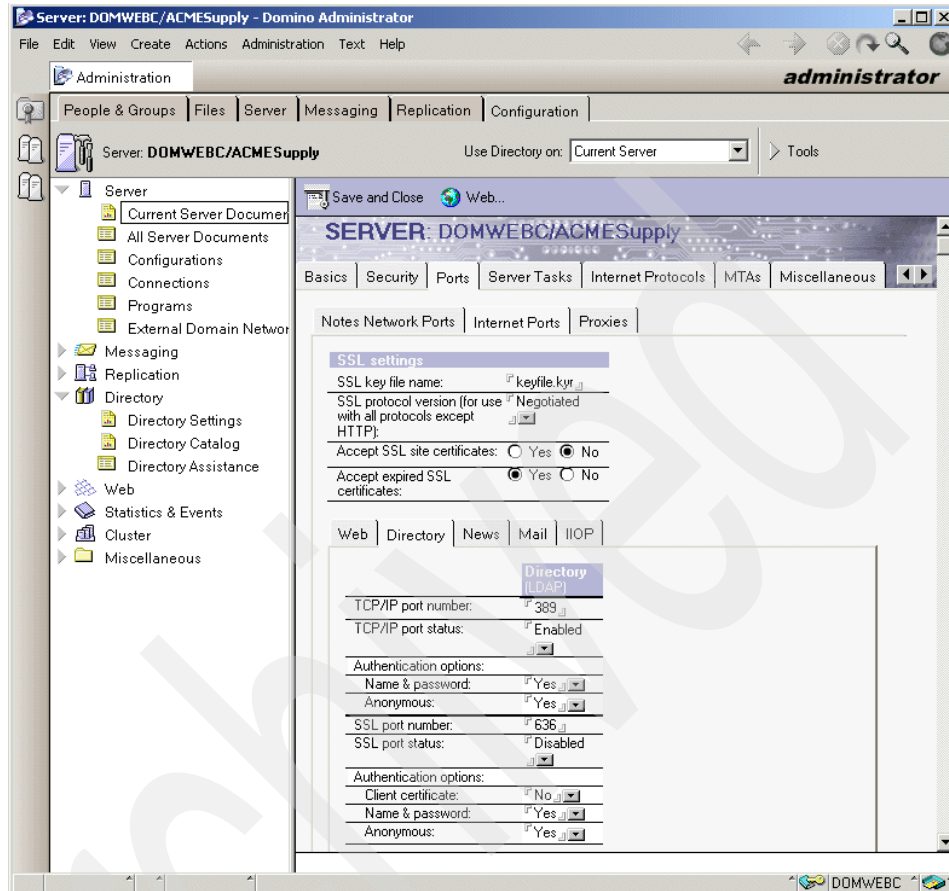


Figure 7-4 Checking the LDAP settings in server doc

8. Click **Save and Close** to save the changes in the server document.
9. End and restart the Domino server to let the changes take effect.

As this is running via a unsecured port, you should only use this setting for testing purposes. If you start this in your production environment, this setting should be changed to run using SSL. See Section 8.10.2, “Enabling Domino SSL with SSO” on page 406.

### 7.3.3 Configuration changes to the Domino LDAP services

It might be necessary to make configuration changes to your Domino LDAP server for such different reasons like:

- ▶ To control which fields anonymous LDAP users can search
- ▶ To enable LDAP write access to the Domino Directory
- ▶ To set time-out and maximum number of entries returned to a search

For more information about these possible configuration changes, please refer to *Domino Administrator Help* (*help/help5\_admin.nsf*) and Section 4.2.2 in *Getting the Most from Your Domino Directory*, SG24-5986.

## 7.4 Using LDIF to exchange directory information

This section describes how LDAP directory data can be exchanged by use of LDIF files. The exchange is done between an IBM Secureway LDAP directory on an iSeries server and an Domino for iSeries server on another iSeries server. In our scenario, this would be the iSeries Shop LDAP server and the ACME Supply Domino server. This data exchange will allow the users of both LDAP servers to look up the e-mail addresses and telephone numbers of the imported users of the other server without having an online connection between these two companies at the moment.

### 7.4.1 Scenario objectives

The objectives of this scenario are:

- ▶ Provide employee information from iSeries Shop employees to ACME Supply, and vice versa.
- ▶ Only the following information should be exchanged between the two companies:
  - Name (first name and last name)
  - Telephone number
  - Mail address

The LDIF files are ASCII files. Therefore, the transfer of these files should be either:

- ▶ Transfer by FTP using the ASCII format transfer option
- ▶ Transfer by attaching the files to an e-mail

## 7.4.2 Exporting LDIF data from the Domino Directory

The Information we want to export from the Domino Directory are all registered users. A registered user in the Domino Directory has an object class of person, and we will only export the following attributes of the object class into the LDIF file to meet the objectives of this scenario:

- ▶ objectclass
- ▶ dn
- ▶ cn
- ▶ sn
- ▶ givenname
- ▶ telephoneNumber
- ▶ mai

The Domino Directory data of ACME Supply will be exported into an LDIF file using the standard LDAP ldapsearch utility.

To export the user data of the Domino Directory, complete these steps:

1. Start a command prompt from your workstation. We used a Windows 2000 command prompt.
2. Enter the following command to export the data into the file domwebc.LDIF:

```
ldapsearch -h domwebc -O 0 -L -D "cn=ACME Admin,cn=ACMESupply" -w my5ldap  
(objectClass=person) objectclass dn cn sn givenname telephoneNumber mail >  
domwebc.LDIF
```

Here, note that:

- The option -L prints all found entries in LDIF format.
- The option -O with the value 0 lets the ldapsearch use a maximum number of zero referrals to follow in the search sequence.

The output for one user would look like this:

```
dn: CN=Claudia Steffens,O=ACMESupply  
objectclass: top  
objectclass: person  
objectclass: organizationalPerson  
objectclass: inetOrgPerson  
objectclass: dominoPerson  
cn: Claudia Steffens  
sn: Steffens  
givenname: Claudia  
telephonenumber: 124-6545  
mail: CSteffens@ralyas4c.iseries.itso.ral.ibm.com
```

This output can now be used by any other LDAP server to import the Domino user information into the directory.



### 7.4.3 Importing LDIF data into an iSeries LDAP

This section shows how to import the ACME supply data into the LDAP directory of iSeries Shop. For a detailed description of importing directory data into an iSeries LDAP server, see Section 5.4, “Exporting and importing information via Operations Navigator” on page 187.

Prior to the import of the LDIF data, you have to decide about the directory structure, where you want to import the Domino users into. In our scenario, we decided to enter a new organization (o=acmesupply) for these users. To enter this new organization, you have to:

1. Add a new suffix o=acmesupply to the LDAP server of the iSeries Shop company by using the Operations Navigator.
2. Add acmesupply as an organization to the LDAP server of the iSeries Shop company by using the DMT.

For detailed information on how to add a new organization into the IBM SecureWay LDAP server in steps 1 and 2, see Section 5.2.4, “Adding organization entries using the DMT” on page 165.

**Important:** When exporting directory information in Domino, all object classes of the exported entries are also written to the LDIF file. This includes the class `dominoPerson`. However, this object class is not part of the standard IBM SecureWay directory schema. To successfully import the entries in the LDIF file, you can either replace the object class by using the replace function of an editor for the LDIF file or by extending your schema. For the iSeries import, we chose to extend the schema by adding the `dominoPerson` class as a sort of dummy class since we do not need the Domino-specific attributes in our server. For details on how to extend the OS/400 directory schema, refer to Appendix B, “Extending your directory schema” on page 521.

There are two ways to import the data into the LDAP server of the iSeries Shop company, using the `ldapadd/ldapmodify` utility or Operations Navigator. Because the iSeries Shop needs its LDAP directory 24 hours a day 7 days a week, Operations Navigator would be the only alternative way to add the data, since the LDAP server has to be stopped to import the data via Operations Navigator. The planned weekly updates of the directory information on the LDAP server of the iSeries Shop would then only be done with the `ldapmodify` utility. For more information about the `ldapmodify`, see Section 5.3, “Using LDAP utilities to manage the directory” on page 176.

**Tip:** For the weekly updates it would make sense to run the **ldapmodify** command twice, the first time with the parameter **-r -c** to force the changes of already-existing data in the first step:

```
ldapmodify -r -c -h ralyas4b -D "cn=administrator" -w my5ldap -f domwebc.ldif
```

And then in the second step use the parameters **-a -c** to add any new entries into the LDAP directory, for example:

```
ldapmodify -a -c -h ralyas4b -D "cn=administrator" -w my5ldap -f domwebc.ldif
```

The important switches in the scenario are:

- r Specifies that the default behavior of the **ldapmodify** command is to replace existing entries with the ones found in the LDIF file.
- a The **ldapmodify** utility tries to add all entries found in the LDIF file.
- c This switch specifies that the **ldapmodify** command continues in case of an error. For example, when using **ldapmodify** to update the **ACMESupply** entries using the LDIF file, some entries might already be there while others represent new employees and therefore have to be added. Without the **-c** switch, replacing entries will fail as soon as one entry is found in the LDIF file that is not found in the LDAP directory. When adding entries the command fails when an entry is already in the directory. Therefore, we recommend that you use the **-c** switch. In this scenario, it causes the **ldapmodify** command to first replace all entries and then add new entries without stopping on an error.

## Importing with **ldapadd**

This section shows how to use the **ldapadd** utility to do the initial import of the LDIF file into the directory. Perform the following steps to import the LDIF data into the iSeries Shop LDAP server:

1. To import LDIF data with the LDAP utilities, the LDAP server must be up and running.
2. Start a command prompt from your workstation.
3. Make sure the LDIF file is in the same directory where you start the **ldapadd** command. In our scenario, this is C:\. See Section 7.4.2, "Exporting LDIF data from the Domino Directory" on page 314.
4. Enter the following command to import the data from the file **domwebc.LDIF** to the iSeries Shop LDAP server **ralyas4b**:

```
ldapadd -h ralyas4b -D "cn=administrator" -w my5ldap -f domwebc.LDIF
```

Then press Enter.

5. The output for one entry to be imported into the iSeries Shop LDAP would look like this:

```
adding new entry CN=Claudia Steffens,O=ACMESupply
```

The users are now imported to the organization O=ACMESupply in the iSeries Shop LDAP server ralyas4b.

To check if all users are imported successfully, start the DMT and check if the users are displayed in your directory tree.

## 7.4.4 Exporting LDIF data from iSeries LDAP

This section shows the export of iSeries Shop LDAP directory data to LDIF files. The iSeries Shop decided to only export the user data of their own employees to ACME Supply.

Here again are two possible ways to export the data from the IBM SecureWay LDAP server. You can use either the **ldapsearch** command or Operations Navigator. Since the Operations Navigator approach exports all attributes of an entry and we only want the ACME Supply company to get names, e-mail addresses, and telephone numbers, we cannot use Operations Navigator.

### Export using ldapsearch

The export of directory data using the **ldapsearch** command is similar to the one used to export the data from the Domino Directory. See Section 7.4.2, "Exporting LDIF data from the Domino Directory" on page 314.

To export the user data of the IBM SecureWay LDAP directory of iSeries Shop with the server name ralyas4b, complete these steps:

1. Start a command prompt on your workstation.
2. Enter the following command to export the data into the file iSeriesShop.LDIF:

```
ldapsearch -h ralyas4b -o 0 -L -D "cn=adminstrator" -w my5ldap -b  
ou=employees,o=iserieshop (objectClass=person) objectclass dn cn sn  
givenname telephoneNumber mail > iSeriesShop.LDIF
```

Here, note that:

- The option -L prints all found entries in LDIF format.
- The option -O with the value 0 lets ldapsearch use a maximum number of zero referrals to follow in the search sequence.

- -b ou=employees,o=iseriesshop defines the base DN for the users to be exported.

The output for one user would look like this:

```
dn: cn=Thomas W Barlen,ou=employees,o=iseriesshop
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
objectclass: publisher
objectclass: ePerson
cn: Thomas W Barlen
cn: Thomas W. Barlen
cn: BARLEN
sn: Barlen
givenname: Thomas
telephonenumber: 919-301-4564
mail: barlen@ISERIES.ITSO.RAL.IBM.COM
```

The LDIF file can now be imported in any other LDAP directory like the Domino LDAP directory on DOMWEBC of ACME Supply.

### 7.4.5 Importing LDIF data into the Domino Directory

The import of LDAP directory data from the iSeries Shop into the ACME Supply Domino LDAP server will be done using the Domino Administrator.

**Note:** From a Domino Administrator point of view, it would make sense if we separate the imported data from the primary Domino Directory data and setup, for example, a secondary directory only for the imported data. This directory could then be addressed by a Directory Assistance document as a referral. Since this setup would exceed the scope of this book, we only show the import of LDIF data into the primary Domino Directory.

#### Important change to the LDIF file

Manual changes have to be made in the LDIF file exported by the IBM SecureWay LDAP server of the iSeries Shop.

- The employee entries in the iSeries LDAP directory published via SDD have the object classes publisher and ePerson associated.
  - The object class publisher is an object class that is populated to the IBM SecureWay LDAP server by publishing System Distribution Directory entries into the LDAP directory.

- The object class `ePerson` is needed in the iSeries LDAP directory to provide the fax number (attribute=`facsimileTelephoneNumber`), since this is not included in any other used object class in our scenario.

These two object classes, `publisher` and `ePerson`, are not used in the Domino LDAP schema. Since the extension of the Domino LDAP schema would exceed the scope of this book, we have chosen to manually modify the LDIF file prior to the import.

- The schema of the IBM SecureWay LDAP server stores the mail address of the user in the attribute `mail`. To achieve our scenario objective, which is to allow the users of the ACME Supply Domino server the lookup of e-mail addresses of the imported iSeries Shop users, and due to the fact that these imported users should not be registered Notes mail users, we have to change the mail attribute in the LDIF import file to the attribute `mailaddress`. If you do not change the mail attribute to `mailaddress`, a Notes user will be created for each imported entry.

You have to perform the following steps to modify the LDIF file:

1. Open the import file using the Notepad of your workstation. In our scenario, this would be the LDIF file exported by the `ldapsearch` command in “Export using `ldapsearch`” on page 317, with the path `C:\iSeriesShop.LDIF`.
2. Click **Edit**, and in the drop-down menu, select **Replace** to open the Replace window. Remember that we used the Windows Notepad to perform the replace function. If you use another editor, the instructions may be different.
3. Type in the following values as shown in Figure 7-5 on page 320:
  - For **Find what**, enter `objectclass: publisher`.
  - For **Replace with**, enter `#objectclass: publisher`.

A line with a `#` as the first character within an LDIF file is not imported into any LDAP directory because it represents a comment line.

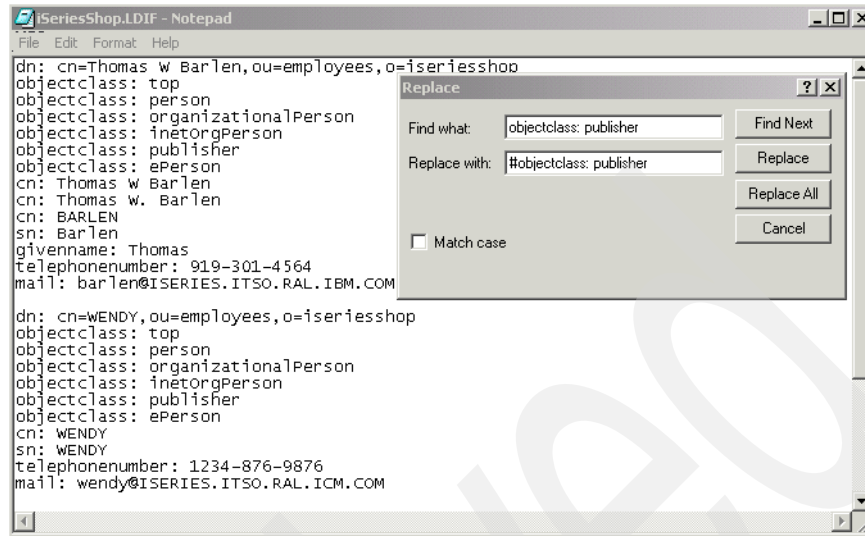


Figure 7-5 Change of LDIF file in Notepad

Click **Replace all** to comment out the line `objectclass publisher`.

4. Since the **Replace** window will still be displayed, perform step 3 again for the line `objectclass: ePerson`.
5. Since the **Replace** window will still be displayed, perform step 3 again, this time for the mail attribute.
  - For **Find what**, enter: `mail`.
  - For **Replace with**, enter: `mailaddress`.

Click **Replace all** to change the name of the mail attribute from `mail` to `mailaddress`.

6. Click **Cancel** to exit the Replace window.
7. Click **File -> Save** to save the changes to the LDIF file.

The LDIF file is now ready to be imported by the Domino migration utility.

## Import using the Domino Administrator

To import the LDIF file into the Domino Directory of ACME Supply, you have to complete the following steps:

1. Start your Domino Administrator Client on your workstation.

**Note:** We are using the Lotus Domino Administrator Release 5.0.6a instead of Release 5.0.9 due to an open Lotus software problem SPR #KCHL4X8AAL regarding the LDIF import of users.

2. Make sure that you have selected your server as the current server you are working on. In our scenario, this would be DOMWEBC/ACMESupply. See Figure 7-6 for an example of steps 2 through 5.

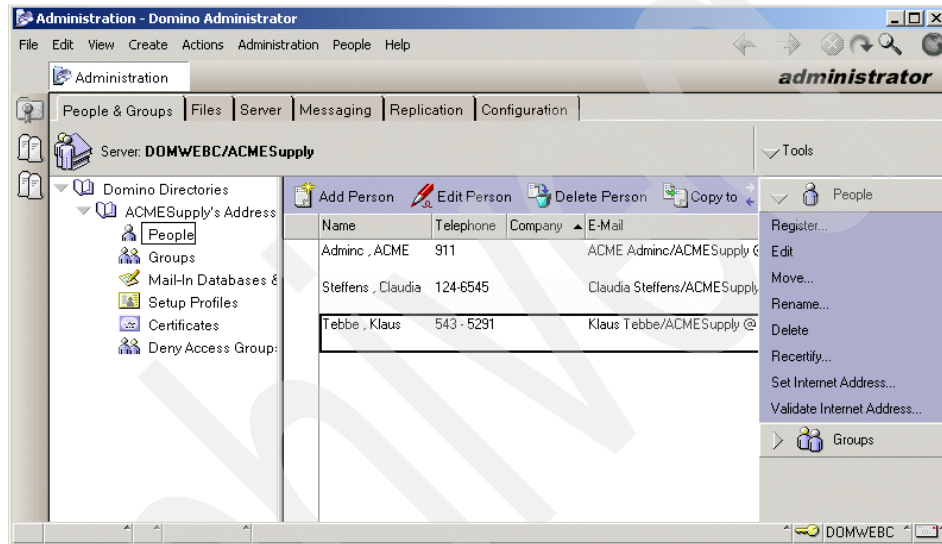
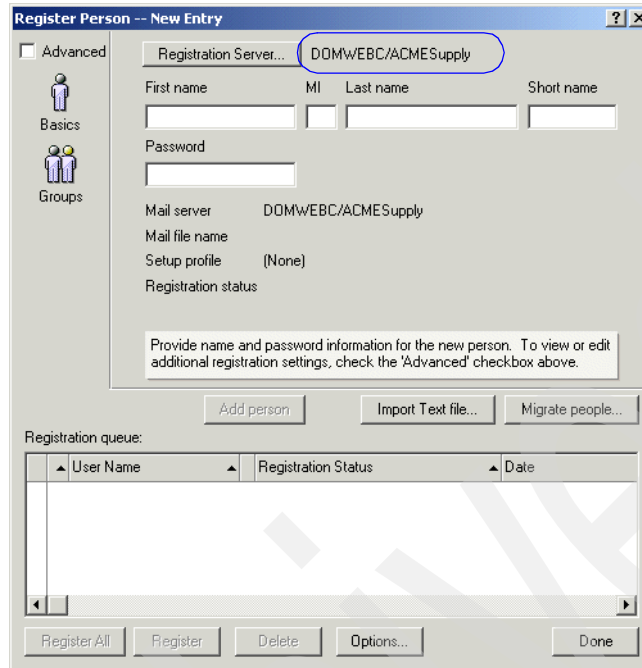


Figure 7-6 Domino Administrator People view

3. Select **People & Groups** to show the people and groups section of the address book.
4. Select **Domino Directories** -> **ACMESupply's Address Book** -> **People** to show the person entries of the address book.
5. Expand the **Tools** drop-down menu and then expand the **People** menu on the right side of the window.
6. Click **Register** to start the import process.
7. When prompted, enter the correct password for the certifier ID of your server /ACMESupply in the Enter Password window and click **OK**.
8. The Register Person -- New Entry dialog box appears. Make sure that the Registration Server field shows the server on which you want to import the LDIF file. In our scenario, this would be DOMWEBC/ACMESupply. See Figure 7-7 on page 322.



The dialog box is titled "Register Person -- New Entry". It has a sidebar on the left with icons for "Basics" (selected) and "Groups". The "Basics" tab contains the following fields:

- Registration Server...**: A dropdown menu with "DOMWEBC/ACMESupply" selected.
- First name**, **MI**, **Last name**, and **Short name**: Four text input fields.
- Password**: A text input field.
- Mail server**: A text input field with "DOMWEBC/ACMESupply" entered.
- Mail file name**: A text input field.
- Setup profile**: A dropdown menu with "(None)" selected.
- Registration status**: A text input field.

Below these fields is a message box: "Provide name and password information for the new person. To view or edit additional registration settings, check the 'Advanced' checkbox above." Below the message box are three buttons: "Add person", "Import Text file...", and "Migrate people...".

At the bottom, there is a "Registration queue:" section with a table:

User Name	Registration Status	Date

Below the table are five buttons: "Register All", "Register", "Delete", "Options...", and "Done".

Figure 7-7 Register Person dialog box - New Entry

9. Click **Migrate People** to open the People and Groups Migration window.
10. In the Foreign directory source field, select **LDIF Entries**. This opens the Select LDIF File window where you have to specify the path and file name of the LDIF file you want to import in the LDIF file to import field. In our scenario, this would be the LDIF file exported by the **ldapsearch** command in "Export using ldapsearch" on page 317 with the path C:\iSeriesShop.LDIF. See Figure 7-8 on page 323.

**Note:** The LDIF Entries option only appears when you have installed the appropriate option during client installation in the Migration Tools section.



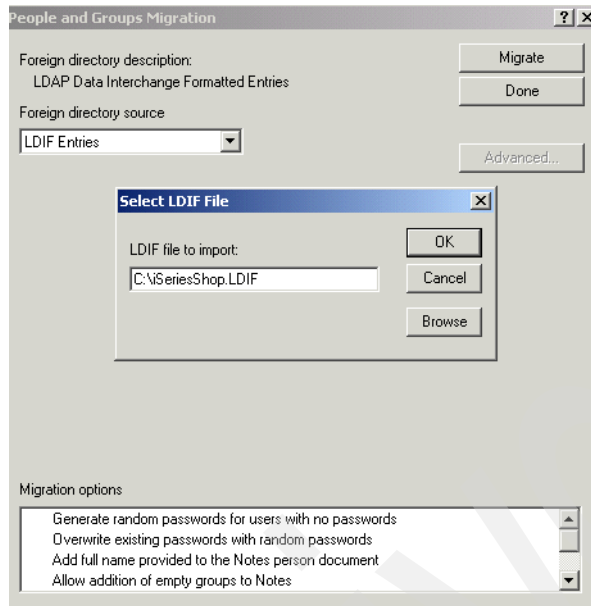


Figure 7-8 People and Groups migration / Select LDIF file

Click **OK** to select the specified LDIF file for the import.

11. Note the message All LDIF person entries are ready to be migrated! in the People and Groups Migration window. Note at the bottom of this window the default Migration options is set for this migration.

Since we are not registering the users migrated from the file in the Domino Directory, we are not creating Notes IDs or mail files for them so check that in the Migration options field. The option **Add full name provided to the Notes person document** is selected. This ensures that the LDAP DN is added to the person document as the primary username. Leave the other options to their defaults.

Click **Migrate** to read in the LDIF file into the Domino Administration import utility.

12. The Domino Administration window appears and tells you how many users of the import file were successfully queued for the registration, were queued with errors, or were not queued. Click **OK** to close this window.

13. Click **Done** to return to the Register Person dialog box.

All the final steps to register the users in the Domino Directory are now done in this dialog box. Please do the following steps:

- a. Note in the Registration queue status box all the users available for registration in the Domino Directory. Select all users in the left most column.
- b. You may want to check the advanced person registration options by selecting **Options**. We do not change any options at this time and keep the defaults.
- c. Select the **Advanced** check box in the upper left corner of the window to display the advanced registration options.
- d. Select the **Mail** pane.
- e. From the **Mail** pane, select **Other internet** in the Mail system box. See Figure 7-9 for an example.

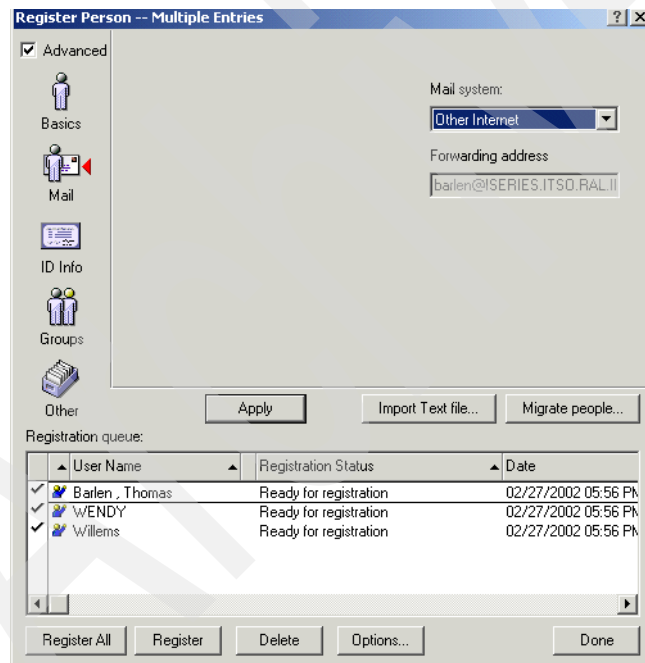


Figure 7-9 Import LDIF users mail pane selection

- f. Click **Apply** to select this mail setting for all users to be registered.
- g. Select the **ID Info** pane and clear the check boxes in the section for the Location for storing user ID since we do not want to create any user ID for the imported users.

- h. Click **Apply** to select the ID info settings for all users to be registered.
- i. In the Registration queue status box, make sure that the users you have selected in step 13 on page 324 are still selected. See Figure 7-10 for the example of our scenario.

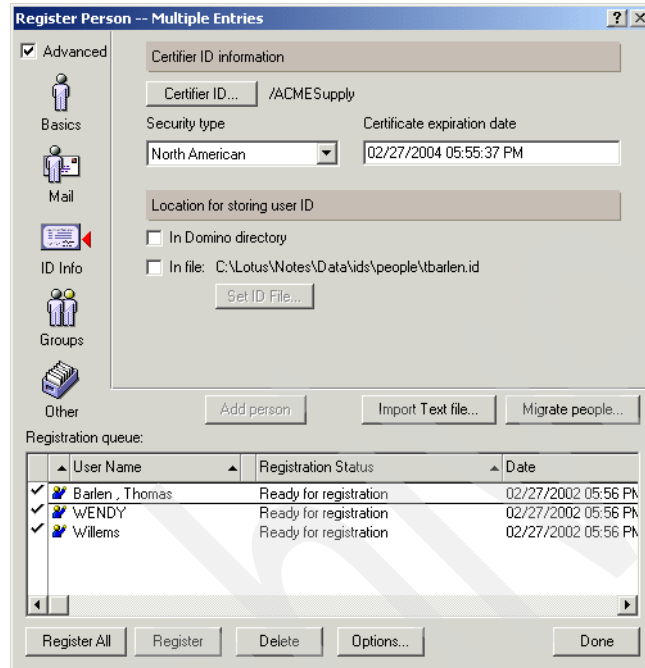


Figure 7-10 Import LDIF user ID info pane

- j. Click **Register All** to finally register all users in the Domino Directory. The Domino Administrator now registers the selected users in the Domino Directory.
- k. The final message box appears and shows that all people were registered successfully (Figure 7-11).

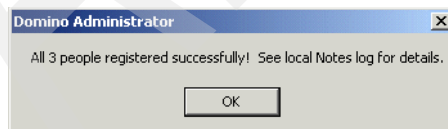


Figure 7-11 Person registered successfully

Click **OK** and **Done** to leave the Register Person dialog box.

The LDAP users of the iSeries Shop directory are now imported as users to the Domino Directory of ACME Supply and can be used for e-mail addressing and looking up telephone numbers.

## 7.5 Connecting directories: The alternative

After removing the network constraints as described in Section 7.2.1, “Scenario characteristics” on page 306, ACME Supply can enhance their data exchange with the iSeries Shop by providing an online connection. This eliminates the manual data exchange including the data import and export. However, it involves an additional administration effort on both LDAP servers in restricting access to directory information that is not supposed to be seen by the other company.

The alternative to the more secure, but time consuming, process of LDIF data file interchange between the two companies would be the setup of LDAP referrals in both LDAP servers of iSeries Shop and ACME Supply and using the online connection between the two companies. See Figure 7-12 for an example of this setup.

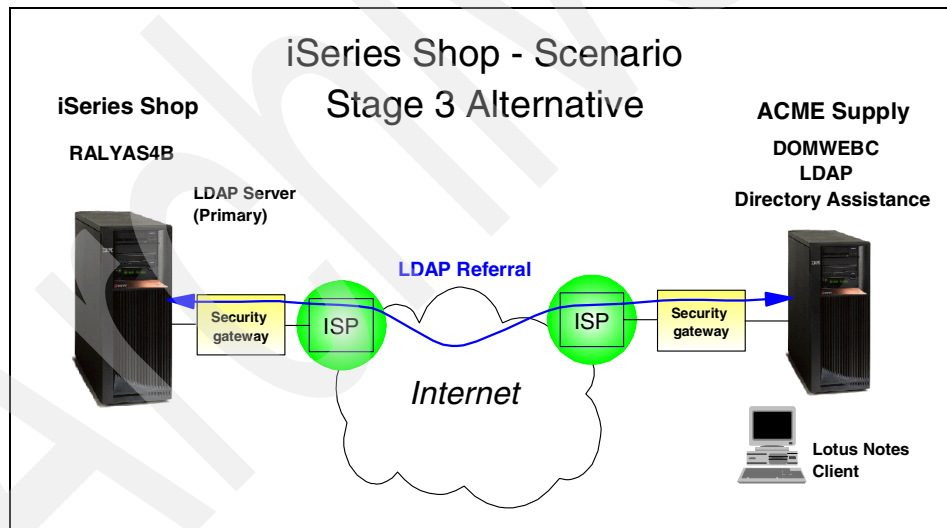


Figure 7-12 Scenario stage 3: The alternative

### 7.5.1 Scenario objectives

The scenario objectives are:

- Setup referrals in the IBM SecureWay LDAP directory (see Section 4.8, “Setting up directory referrals” on page 122).

- ▶ Setup Directory Assistance in the Domino for iSeries server to show the use of LDAP referrals.
- ▶ Secure the directory access on iSeries Directory services to specific data by setting up ACL, as described in Section 5.7, “Controlling access to directory entries” on page 198.

## 7.5.2 Creating the Directory Assistance database

The first step in setting up Directory Assistance is to create the Directory Assistance database. To do this, please follow these steps:

1. Either from the Notes Client or the Domino Administrator pull-down menu, select **File -> Database -> New**.
2. From the New Database window, enter the following values, as shown in Figure 7-13:
  - Server: Select your Domino server, either by typing its full name in the window or by selecting it from the drop-down menu. In our example, the Domino server we used was DOMWEBC/ACMESupply.
  - Title: Type an appropriate title for the database. In our example, we used ACME Supply Directory Assistance.
  - File name: This can be any filename you like. In our example, this was ACMEDA.nsf.
  - Click **Template Server** and choose an appropriate server that has the Directory Assistance database template on it. In our example, the same server where we create the database is DOMWEBC/ACMESupply.
  - From the list of templates, choose the **Directory Assistance** template (da50.ntf).
  - Make sure **Inherit future design changes** is selected.

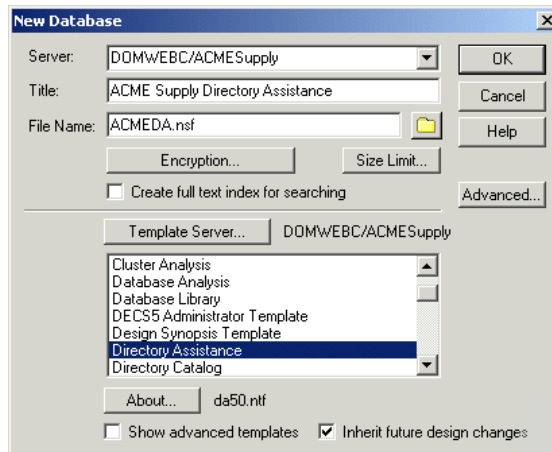


Figure 7-13 Creating the Directory Assistance database

- Click **OK**. Domino creates the database with your name in the ACL as the Manager.

3. When the Directory Assistance about document appears, press Esc.

You are now ready to add new Directory Assistance documents to the database.

### 7.5.3 Setting up directory referrals

In Domino terminology, a referral to another directory is called *Directory Assistance*. Therefore, setting up referrals for directory lookup purposes in the Domino means to create Directory Assistance documents in the Directory Assistance database.

You can use the Directory Assistance database to set up referrals to secondary Domino directories and to an external LDAP server. Since we only discuss the set up for external LDAP servers in this redbook, refer to Section 4.3.1 in *Getting the Most from Your Domino Directory*, SG24-5986, for setting up rules for Domino Directories.

To create referrals for external LDAP servers in the Directory Assistance database, please do the following steps:

1. Open the Directory Assistance database from your Notes Client.
2. Select **Add Directory Assistance** as shown in Figure 7-14 on page 329.

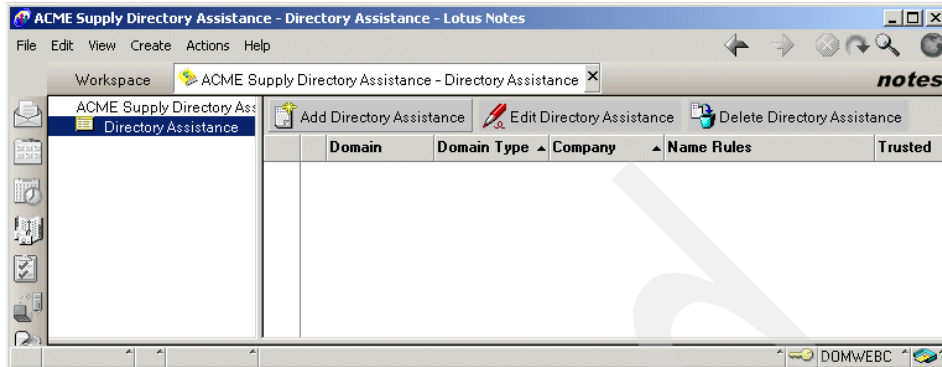


Figure 7-14 Adding the Directory Assistance document

3. In the Directory Assistance document, select the **Basics** tab and enter the basic definitions for this document. These are mostly used by the Domino administrator to organize the entries in the Directory Assistance database. Note that some of the parameters do not show up when not all options are selected as listed. For the values used in our example, see Figure 7-15 on page 330.
  - Domain Type: Choose the type of directory. Since we are setting up an referral to an external LDAP server, this should be LDAP.
  - Domain Name: Domain Name is an identifier that must be assigned to each Directory Assistance rule set. It must be unique. It does not have to be the same name as the actual LDAP server you are referring to, but you might want to use that as a guide. In our example, we use IBM Secureway.
  - Company name: The name of the company associated with this directory. Multiple Directory Assistance documents can use the same company name, in our example we used iSeriesShop.
  - Search order: This is a number that represents the order the directory is searched, relative to other directories in the Directory Assistance database. In this scenario, we left this field blank.
  - Group expansion: Select **No**, because this allows Directory Assistance to verify Web user membership in a group in this LDAP directory when a group is included in a database ACL that a Web user is attempting to access on a Domino Web server. You can choose this option even if you do not use the directory to authenticate the Web users. We do not use this function in this scenario.
  - Enabled: Select **Yes** to enable Directory Assistance for this directory.

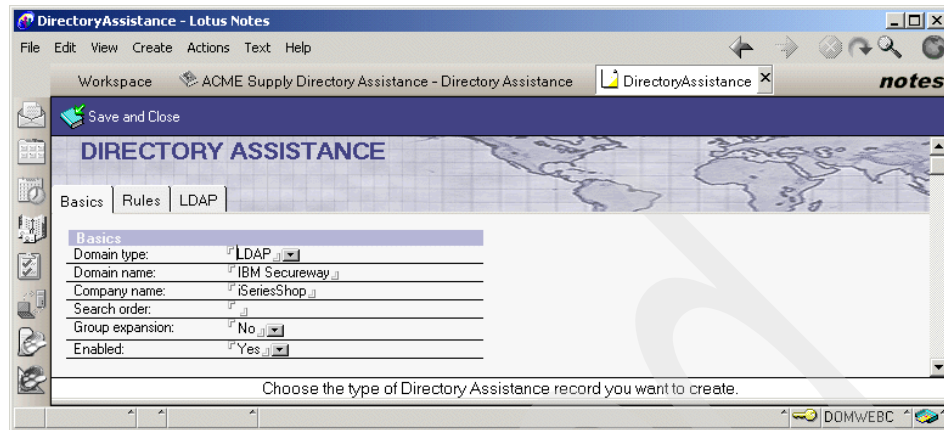


Figure 7-15 Creating the Directory Assistance document: Basics tab

4. The rules that define when Directory Assistance should consult another directory are specified on the second tab (Rules). Once you have finished filling out the Basics tab, click **Rules** and describe what scenario applies to this directory. See Figure 7-16 on page 331.

- Rules: If you want this directory to be searched regardless of the name given for a match, you can put in a rule with all asterisks. More likely, a given directory will hold information for people in given organizations, so you can fill in the organization field. If you want to include a specific OU, you can list it in the relevant OU field(s), you can use a wildcard (\*) character to include all OUs, or you can leave the field blank to prevent matching entries with an OU in that position.

This means that when an LDAP or Notes client specifies a search base that corresponds to one of these rules, the Domino LDAP service refers the clients to this LDAP server.

In our scenario, we leave the rules at default with all asterisks (\*), so that the referral searches for all entries in the addressed LDAP directory.

- Enabled: You can specify whether that rule is enabled. This is an easy mechanism to quickly enable/disable matching rules. To enable the rule, ensure that Enable is set to **Yes**.
- Trusted for Credentials: Here you can configure the rule, regardless of whether it can be used for authentication to the server. A rule that is not trusted can still be used for mail addressing from a Notes client, if selected in the LDAP tab of this document, from LDAP clients as well, but it will not govern authentication.

In our case, we do not use the LDAP directory for authentication. Therefore, Trusted for Credentials is set to **No**.



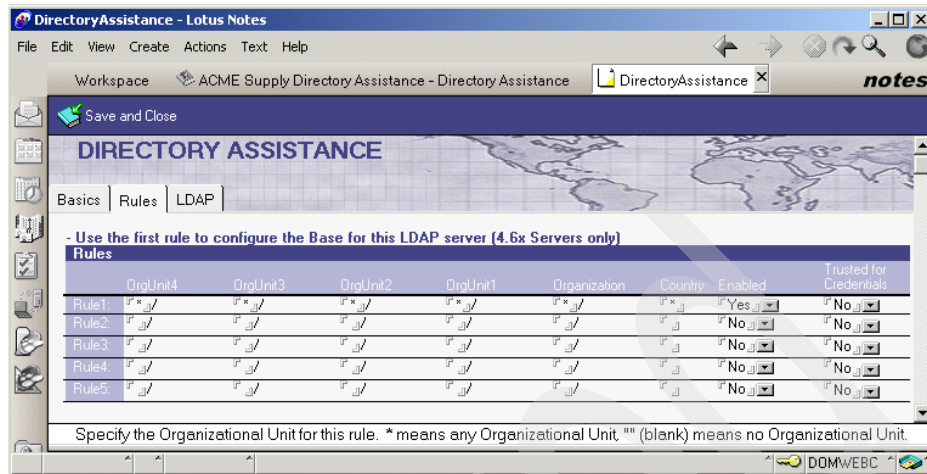


Figure 7-16 Creating the Directory Assistance document: Rules tab

5. Select the **LDAP** tab where you can specify the LDAP-specific parameters for the rules defined in this Directory Assistance document. For the values used in our scenario, see Figure 7-17 on page 333.

- Host name: Specify the fully-qualified host name for the LDAP server. The Domino LDAP service includes this information in the referrals. In our example, we use RALYAS4B.ISERIES.ITSO.RAL.IBM.COM.
- Optional Authentication Credential: To allow the LDAP directory server to authenticate the Domino server, enter a distinguished name in the Username field and a password in the Password field. Enter the distinguished name in LDAP format, for example, cn=admin. The name and password must correspond to a valid name and password in the directory on the LDAP directory server. In case of two different companies, you are likely to get an individual DN specific to your company rather than the administrator of the directory. This DN would only allow access to the information the other company wants you to see.

If you do not enter a name and password, the Domino server attempts to connect to the LDAP directory server anonymously. In our example, we use:

- Username: The distinguished name (DN) for the LDAP server this document is referring to. Here we use cn=admin.
- Password: The password for the DN entered above. Here we use my5ldap.

Lotus recommends that you encrypt the fields containing the username/password combination so that they will not be stored in the clear. For more information on how to encrypt fields in documents, refer to *Domino Administrator Help (help\help5\_admin.nsf)*.

- Base DN for search: Some LDAP directories require you to specify a search base. A search base defines where in the directory naming tree searches should start. In our scenario, we use `ou=employees,o=iseriesshop`.
- Perform LDAP search for: In this entry, you can decide which client can use this directory referral.

Notes clients for mail lookup using function key F9 and Web clients for authentication to Domino servers are selected by default.

If you want to allow LDAP clients to use the external directory by way of a referral you should check that box as well.

In our case, we allow Notes and LDAP clients the use of this referral, so we select **Notes Clients/Web Authentication** and **LDAP Clients**.

- Channel Encryption: To secure the channel between the Domino server and the external LDAP server, we recommend that you enable SSL. This encrypts the packets sent across the wire between the two servers and prevents anyone from capturing directory data.

In our scenario, we do not select SSL at this time, so we select **None**. If you want to use SSL, find more information on how to enable SSL for the LDAP connection in Section 8.10.2, “Enabling Domino SSL with SSO” on page 406.

- Port: The port number that is used to connect to the LDAP server. In our scenario, we use the standard port for LDAP = 389. In case of a secure connection, the port should be changed to 636.
- Timeout: This is the maximum number of seconds before queries to the external directory terminate. We left it as the default of 60 seconds.
- Maximum number of entries returned: The maximum number of names that the LDAP directory server will return for the name searched. If the LDAP directory server also has a maximum setting, the lower setting takes precedence. If the server's maximum time-out is exceeded, it only returns the number of names found to that point. We left it as the default of 100 entries.
- Dereference alias on search: An indicator as to how alias objects (as defined in X.501) are to be handled in searching. Possible values are:
  - Never: Do not dereference aliases in searching or in locating the base object of the search. This is the default and recommended value since

dereferencing aliases can be expensive on some LDAP server implementations.

- InSearching: Dereference aliases in subordinates of the base object in searching, but not in locating the base object of the search.
- FindingBase: Dereference aliases in locating the base object of the search, but not when searching subordinates of the base object.
- Always: Dereference aliases both in searching and in locating the base object of the search.

In our scenario, we set this to Always.

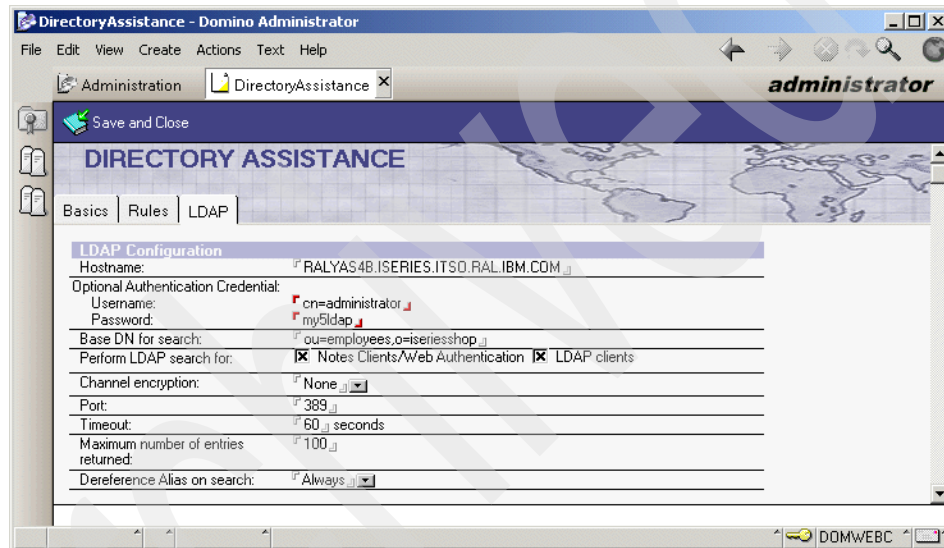


Figure 7-17 Creating the Directory Assistance document: LDAP tab

6. Click **Save and Close** to save this document to the Directory Assistance database.

## 7.5.4 Deploying Directory Assistance and referrals in your domain

Once you have set up the document for your Directory Assistance, you need to make the Domino server aware of the Directory Assistance database. To set that up, you need to add its file name to the server's Server Document in the Domino Directory. To do this, follow these steps:

1. Start your Domino Administrator client.
2. Select the **Configuration** tab.

- Expand the **Server** section in the left-hand pane and click **Current Server Document**.
- Select **Edit Server** to edit this server document.
- In the **Basics** tab section, you see a field for the file name of the Directory Assistance database. Here you should enter the path and file name for the Directory Assistance database. The default is blank. The entry should be relative to the data directory. In our scenario, we enter the name `ACMEDA.nsf`. See Figure 7-18.

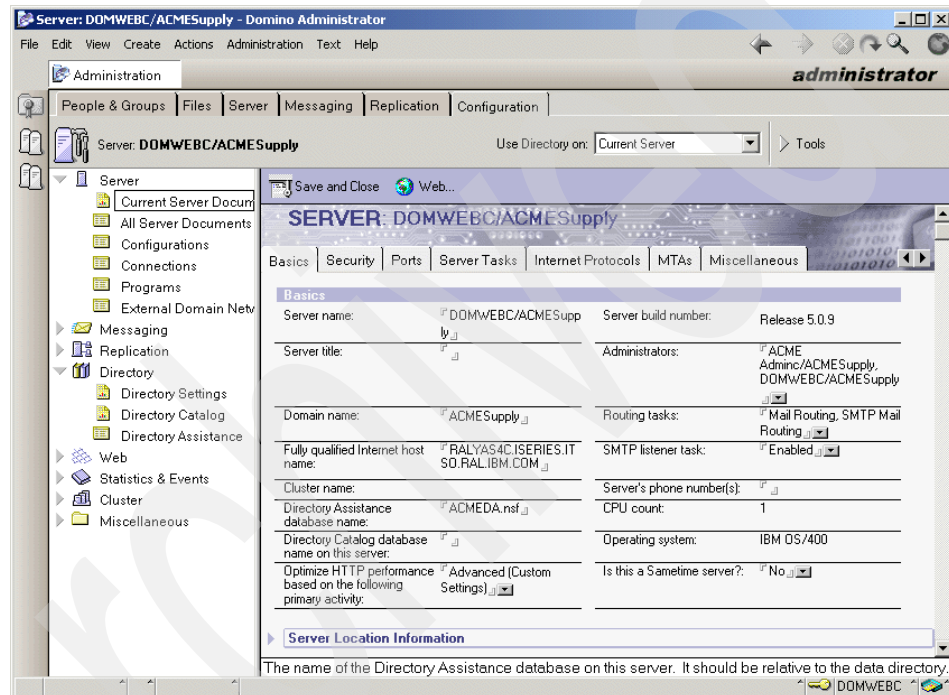


Figure 7-18 Adding the Directory Assistance database name to a server document

- Click **Save and Close** to save the changes in the server document.
- End and restart the Domino server to let the changes take effect.

The Domino server is now able to refer its Notes clients to a secondary LDAP server via Directory Assistance.

Since this is running via an unsecured port, you should only use this setting for testing purposes. If you start this in your production environment, this setting should be changed to run using SSL. See Section 8.10.2, “Enabling Domino SSL with SSO” on page 406.



## Part 3

# Practical scenarios

*Archived*



# Single Sign-On with Domino and WebSphere 4.0

You can use Single Sign-On (SSO) to allow users to log on once per session rather than requiring them to log onto each resource or application separately. The applications could be on the same or different physical servers. Our discussion of SSO implies that a user will not be prompted for authentication credentials more than once during a session.

This chapter describes the following:

- ▶ Enabling WebSphere applications security
- ▶ Protecting a WebSphere resource
- ▶ Enabling Domino security
- ▶ Configuring Domino to use iSeries LDAP
- ▶ Protecting a Domino resource
- ▶ Configuring Single Sign-On
- ▶ Enabling SSL with Single Sign-On

## 8.1 Scenario characteristics

In stage 4 of our scenario, the iSeries Shop merged with iSeries Automotive, Inc., a company that offers similar products and is very successful in the market. iSeries Automotive is, of course, also using an iSeries server to run their business applications. Both companies are interested in leveraging each other's customer base to expand their business. The new company is now responsible for all major marketing campaigns. As part of this effort, they deploy new Web applications that are running under WebSphere Application Server and Lotus Domino on their iSeries server. All business partners and customers need to register to use the new applications.

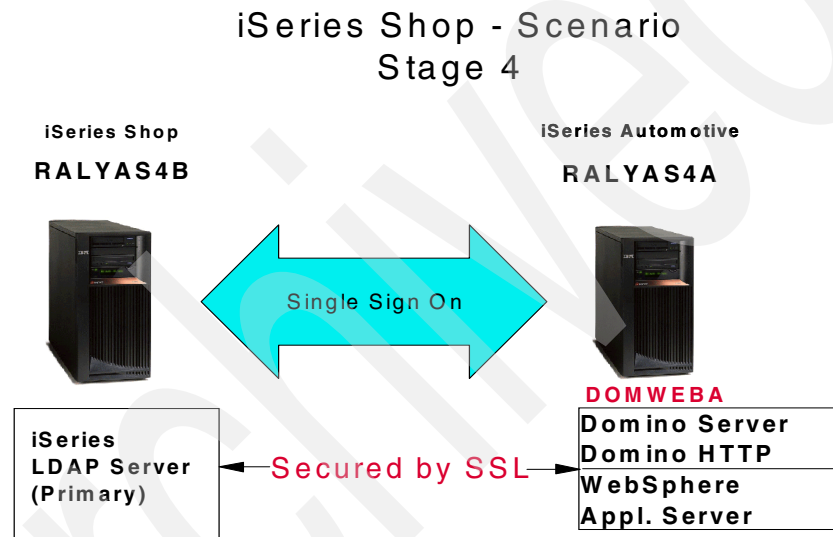


Figure 8-1 Stage 4 Single Sign-On, secured by SSL

## 8.2 Scenario objectives

The objectives of this stage of the scenario are as follows:

- ▶ Minimize user authentication requests for customers who access different applications.
- ▶ iSeries Automotive Lotus Notes user should also be able to look up e-mail addresses from the LDAP directory on system RALYAS4B.



- ▶ LDAP traffic between application servers and LDAP directory should be encrypted.

To meet the objectives, the company implemented the following solution:

- ▶ The Secure Socket Layer (SSL) protocol is used for the LDAP directory server and client to protect the LDAP traffic between DOMWEBA and RALYAS4B.

How to enable SSL for the OS/400 LDAP directory server is described in Section 4.9.4, “Enabling SSL for the LDAP server” on page 128.

How to enable SSL for the Domino Directory and SSO is explained in “Enabling SSL with SSO” on page 402.

- ▶ Since iSeries Shop customers are already registered in the LDAP directory, the WebSphere Application Server and Domino server on iSeries Automotive’s server are set up to authenticate users against the existing LDAP directory. This also requires that common information about customers registered through the new registration process are published to the existing LDAP directory on RALYAS4B. Information that is relevant to the CRM application only, for example, newsletter preferences, is stored in the CRM database and not published to the LDAP directory. The iSeries Shop’s LDAP directory server on RALYAS4B did a once off import from the iSeries Automotive DOMWEBA’s LDAP server via LDIF. From now on Human Resources will manage new Notes users. The Web application registration process uses APIs to publish user information into the LDAP directory on system RALYAS4B.

Refer to Chapter 7, “Setting up LDAP on Domino server for iSeries” on page 291 for information on how to export Domino LDAP to the OS/400 LDAP directory server.

- ▶ To minimize user authentication requests for customers who access different applications, iSeries Automotive implemented Single Sign-On (SSO) on their WebSphere Application Server and Domino server. Using SSO, both application servers share authentication information. For example, if a customer signs on to a Domino application, the WebSphere Application Server can reuse the authentication information and allow access without displaying a user authentication prompt again. This chapter shows you how to set up SSO for WebSphere Application Server 4.0 and Lotus Domino 5.0.9.

## 8.3 Prerequisites

This section covers what you need to install and configure on your iSeries server and PC client in order to use the examples covered in this book.

### 8.3.1 Workstation requirements

This section describes the requirements for the administrative workstation used to install, configure, and manage an integrated Domino and WebSphere environment on the iSeries server. The following lists the required software:

- ▶ Web browser: Netscape 4.75 or Internet Explorer 5.5
- ▶ Lotus Domino Administrator client Release 5.09
- ▶ WebSphere administrative console

The WebSphere administrative console provides a Java graphical user interface (GUI). Since the iSeries server does not support native GUI devices, it cannot run directly on the iSeries server. You must use a Windows or AIX workstation to run the WebSphere administrative console locally.

For details on installing and configuring the WebSphere administrative console to manage a WebSphere Application Server on the iSeries server, please refer to the following URL:

<http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/docs/doc.htm>

### 8.3.2 iSeries requirements

This section describes the requirements for the iSeries server used to install, configure, and manage an integrated Domino and WebSphere environment as documented in this chapter.

The versions of software used in this redbook include:

- ▶ 5769-LNT - Lotus Domino for iSeries R5.0.9
- ▶ 5733-WA4 - IBM WebSphere Advanced Edition V4.0
- ▶ 5722-SS1 - OS/400 V5R1
- ▶ 5722-XE1 - Client Access Express

This redbook assumes that the iSeries server already has TCP/IP configured, and the administrative PC client workstation being used can ping the iSeries server. You must also have \*SECOFR special authority in order to complete some steps in this book.

**Note:** The installation and configuration of each of the products listed is not within the scope of this book. For detailed information, refer to the respective product documentation.

## 8.4 Task summary

This is a summary of all the configuration tasks. Although we could enable SSO in fewer steps, we decided to activate and test security on WebSphere Application Server and Domino individually to make sure that each component works properly on its own before setting up SSO.

- ▶ Enable WebSphere Application Server console security.
- ▶ Protect a WebSphere resource using Application Assembly Tool.
- ▶ Verify that WebSphere Application Server authentication was configured correctly.
- ▶ Enable Domino authentication by restricting access to Domino database(s).
- ▶ Configure Domino to use OS/400 LDAP.
- ▶ Verify that Domino authentication was configured correctly.
- ▶ Configure WebSphere Application Server to use Single Sign-On (SSO).
- ▶ Configure Domino to use Single Sign-On.
- ▶ Verify that Single Sign-On has been configured correctly.
- ▶ Enable WebSphere SSL.
- ▶ Enable Domino SSL.
- ▶ Verify that Single Sign-On works with SSL.

## 8.5 Single Sign-On security concepts

Single Sign-On (SSO) is a description of a user experience; there are multiple technical approaches to ensure the experience meets the user's expectation. Among the major components in an SSO context are:

- ▶ Single authentication directory

Usually an LDAP server, this keeps a central record of a user's credentials (user id/password, distinguished name, and other attributes), which can be used to validate the user's identity to authenticate them. Generally, it is assumed that the user will have a single entry in the directory and therefore a single user id/password or certificate, but this is not absolutely necessary.

This is desirable to ensure that a user has the same identity (user ID) and authorization credentials (password or certificate) in all applications. However, simply using a single authentication directory does not in itself provide SSO; a user visiting multiple applications could receive authentication challenges from all of them. The advantage of a central directory is that the user should not have to remember multiple sets of user IDs and passwords. Naturally, it is also possible to simply store duplicate information in multiple directories, but it is often difficult to synchronize these.

► Persistent authentication

This is often misunderstood to be an example of Single Sign-On. For example, when a Web browser user is challenged by a Web server 401 return code (access denied to resource), the Web browser will present a dialog box to enter a user ID and password rather than display the forbidden access message. Once the user enters their user ID and password, this is presented to the Web server; if accepted, the browser will automatically present it to the same server and all servers in the same realm until the browser session ends. A realm is generally the host name and file system directory accessed, but the Web server can define an arbitrary string to group servers into an extended realm. Although this may superficially appear to be a form of SSO, the user is actually authenticated again for every server access. The limited definition of *realms* (groups of servers or directories within a real or virtual server), plus the fact that the user ID and password are sent base64-encoded but unencrypted, makes this technique difficult to scale and manage. (The communications can, of course, be encrypted by using SSL.) However, the very persistence of the authentication may be a problem in some environments since the browser will cache the user ID and password until the browser is shut down. There is no concept of a “time-out” and another user could use an authenticated user’s credentials if the first user did not shut down the browser after use. As well, the user cannot force a logout from a site; instead they have to remember to shut down their browser. Similarly, if a client has a certificate (usually, an x.509v3 certificate) which is acceptable to all servers they access, there will be the illusion of SSO, but the certificate will be re-authenticated if a user’s session ends and is later re-started. This would occur if a user established an SSL session with a server, then visited a second server and finally returned to the first server. It does have the advantage that authentication is in and case transparent to the user (except that initial access to the certificate will likely be password protected by the browser, and the browser may prompt to select a certificate to use each time the server challenges). Notes access to Domino servers follows this model; the user is only prompted for a password to open their Notes ID file; their certificate(s) are presented for authentication without user prompting to each server the user accesses.

- Persistent authentication sessions

It is also possible for the first server the user contacts to perform authentication against the user's security credentials and then create an "authentication token." Generally, this is a "cookie" stored and presented by the user's Web browser. When the user again contacts the server or any other server that "trusts" it, the token is presented as proof of authentication. The second server can either accept the authentication or re-authenticate, based on the contents of the token. Generally, the token is issued with a limited lifetime so that its validity will expire after a period of time, either of inactivity or simply from its creation. The token often also carries state information or a pointer to state information stored by the application. State information could include the user's last location, contents of a shopping cart, application selections, and the like. This is the type of SSO we will be illustrating in this section. The trust relationship will exist between Domino and one or more WebSphere servers. Trust in this case is assured by encrypting the authentication credentials in the cookie using a "shared secret": a common (LTPA) key. Only servers with the shared LTPA key can participate in the relationship. (LTPA is the WebSphere abbreviation of Lightweight Third-Party Authentication.) In addition, the token is created with a limited lifetime, so it will expire in a fixed period after creation. No state information will be stored in the cookie.

- Reverse proxy or access management applications

These are servers that intercept user requests and pass them on to an application server, retrieving and sending any necessary authentication information on the user's behalf. This technique allows coexistence of applications with inconsistent authentication and state management implementations. We will not illustrate this technique in our examples.

The authentication mechanisms on the platforms need not be the same; all that is necessary is that each platform trusts the other to authenticate. For our example, we will implement Domino session-based authentication using a custom login prompt; for WebSphere, we will use browser basic authentication. However, both Domino and WebSphere will create a cookie acceptable to the other, so the user will only see one authentication challenge in a session and which one depends on whether the user first accesses Domino or WebSphere.

## 8.6 Enabling WebSphere Application Server authentication

Once security has been configured on the WebSphere Application Server the user can only start a Web application after supplying a valid username and password. The username and password must exist in the LDAP directory.

## 8.6.1 WebSphere Application Server authentication concepts

The IBM WebSphere Application Server Version 4.0 security system provides a number of features that you can use to secure your applications, including the following:

- ▶ Authentication policies and services

Authentication is the process of verifying that a user (or process) really is who they say they are. This is usually done with some sort of user ID and password lookup scheme, or a certificate. You can indicate how you want WebSphere Application Server to verify the identity of users who try to access your resources. You can choose a supported directory service, the operating system registry, or a custom registry to verify the identity of users and groups.

- ▶ Authorization policies and services

Authorization is the process of determining if a user has rights to use a secured resource in some way. For example, the right to invoke a method on an EJB or access a particular HTML page, servlet, or JSP. You can specify policies that give different users differing levels of access to your resources. If you define authorization policies, WebSphere Application Server will enforce them for you.

- ▶ Delegation policies

Delegation allows an intermediary to do work initiated by a client under an identity based on the associated delegation policy. Therefore, enforcement of delegation policies affects the identity under which the intermediary performs downstream calls made to complete the current request. When making downstream requests, the intermediary uses the client's credentials by default, but other choices are also possible. The result is that the downstream resources do not know the identity of the intermediary; they see the identity under which the intermediary is operating. There are three possibilities for the identity under which the intermediary operates when making the downstream requests:

- The client's identity (default)
- Its own identity
- An identity specified by configuration

- ▶ Trust policies

Decisions on who or what to trust also help make up an application security policy. Ultimately, in a security policy, something must be judged to be trustworthy, be it a user registry that contains usernames and passwords or a Certificate Authority that issues certificates.

- ▶ **Single Sign-On support**

WebSphere Application Server supports third-party authentication, a mechanism for achieving Single Sign-On across the Internet domain that contains your resources. You can use Single Sign-On to allow users to log on once per session rather than requiring them to log on to each resource or application separately.

- ▶ **WebSphere security components**

As shown in Figure 8-2 on page 346, security for WebSphere Application Server is managed as a collaborative effort by a number of components.

As can be seen in Figure 8-2 on page 346, the security server resides within the WebSphere administrative server. It supports both the administrative aspects of WebSphere security and the runtime aspects. At runtime, the security server is part of the security application and provides the authentication service. The security server will consult with the user registry (an LDAP server, for example) to authenticate a user and to obtain a credential that can be used in the security context to represent the user identity. In this respect, the security server is a trusted third party for security policy and control. The WebSphere security collaborators executing in each application server call on the security server to provide authentication services (including Token services when the LTPA authentication mechanism is used). The security runtime components acquire global security configuration, such as the authentication mechanism and user registry, from the security application. Because the security application is coupled with the WebSphere administrative server, the security configuration information resides in the WebSphere administrative repository (for example, a DB2 database).

## **Secure Sockets Layer (SSL)**

Secure Sockets Layer (SSL) is a public-key network security protocol that can perform message encryption, client authentication, and server authentication. When global security is enabled, SSL is used to encrypt all IIOP communications with application servers and administrative servers, by default. SSL can be used to encrypt HTTP communications between Web browser and Web server, and Web server and application server. WebSphere also supports LDAPS (LDAP over SSL) between the security server and the LDAP server.

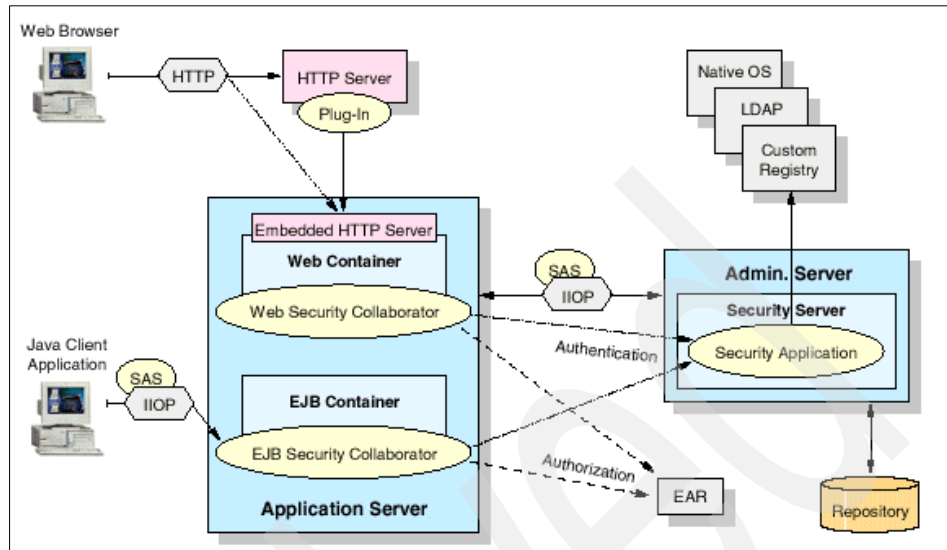


Figure 8-2 WebSphere Advanced Edition security architecture

## Authentication mechanisms

WebSphere Application Server authenticates users by using one of several authentication mechanisms. An authentication mechanism validates the authentication information against a user registry. WebSphere supports two mechanisms for authentication:

- ▶ **Local operating system:** The underlying operating system is used to authenticate the user. Local operating system supports the basic and form authentication methods.
- ▶ **Lightweight Third Party Authentication (LTPA):** LTPA can use a trusted third-party Lightweight Directory Access Protocol (LDAP) server, or a custom user registry to authenticate the user. LTPA supports the basic, form, and client certificate authentication methods.



**Important:** If WebSphere security is to be enabled when running the administrative server as a non-root user, then the local operating system cannot be used as the authentication mechanism. You have to use LTPA in connection with LDAP.

## WebSphere authorization model

Authorization information is used to determine if a caller has the necessary privilege to access a resource. WebSphere Application Server uses the Java 2 Enterprise Edition (J2EE) authorization model. In this model, authorization information is defined as follows:

1. Creates roles and permissions during assembly of the application, so permission to execute methods is granted to one or more security roles.
2. Assigns subjects (such as users or groups) to roles during deployment of the application, so real users are assigned to the security roles.

At execution time, WebSphere Application Server authorizes incoming requests based on the user's identification information and the mapping of the user or group to roles. If the user belongs to any role that has permission to execute a method, the request is authorized. If the user does not belong to any role that has permission, the request is denied.

## Creating roles and permissions

During the assembly of an application, permission to execute methods is granted to one or more security roles. The application assembler defines a list of method permissions for each role, which is stored in the deployment descriptor for the application. WebSphere also allows roles to be assigned to special subjects that are not defined by J2EE. A special subject is a WebSphere-defined entity that is independent of the user registry. It is used to generically represent a class of users or groups in the registry.

- ▶ *AllAuthenticatedUsers* is a special subject that permits all authenticated users to access protected methods. As long as the user can authenticate successfully, the user is permitted access to the protected resource.
- ▶ *Everyone* is a special subject that permits unrestricted access to a protected resource. Users do not have to authenticate to get access, as this special subject allows access to protected methods as if the resources are unprotected.
- ▶ *DenyAllRole* is a special role that is assigned by default to a partially protected resource. For instance, if an enterprise bean has four methods and only three are explicitly protected, the fourth method is associated with the *DenyAllRole*. This role denies everyone access to the methods it is

associated with. The DenyAllRole is never mapped to any users or groups; it is always empty.

## Assigning subjects to roles

During the deployment of an application, users or groups of users, called subjects, are assigned to the roles previously defined by the application assembler. The application assembler does not need to understand the individual methods, as the application assembler grants method permissions to the roles. When a user is assigned to a role, the user gets all the method permissions that are granted to that role. If a user is assigned to more than one role, the permissions granted are the union of the permissions granted to each role. Additionally, if the authentication mechanism supports the grouping of users, these groups can be assigned to roles. Assigning a group to a role has the same effect as assigning each individual user to the role. A "best practice" during deployment is to assign groups, rather than individual users, to roles for the following reasons:

- ▶ It improves performance during the authorization check. There are typically far fewer groups than users.
- ▶ Using group membership to control resource access provides better flexibility.
- ▶ Users can be added to and deleted from groups in the user repository. This is preferred to adding and removing users for WebSphere roles, as the enterprise application must be stopped and restarted for such changes to take effect, which can be very disruptive in a production environment.

## WebSphere security and the operating environment

WebSphere Application Server security sits on top of your operating system security and the security features provided by other components, including the Java language. As shown in Figure 8-3 on page 349, the types of security involved include:

- ▶ Operating system security is used to secure sensitive files in the WebSphere product installation and to authenticate users using the operating system user registry.
- ▶ Java language security provided through the Java Virtual Machine (JVM) used by WebSphere and the Java security classes.
- ▶ CORBA security for inter-application calls between secure ORBs invoked over the Secure Association Service (SAS) layer.
- ▶ J2EE security using the security collaborator to enforce J2EE-based security policies and support J2EE security APIs.
- ▶ WebSphere security relies on and enhances all of the above. It enforces security policies and services in a unified manner for Web and EJB resources.

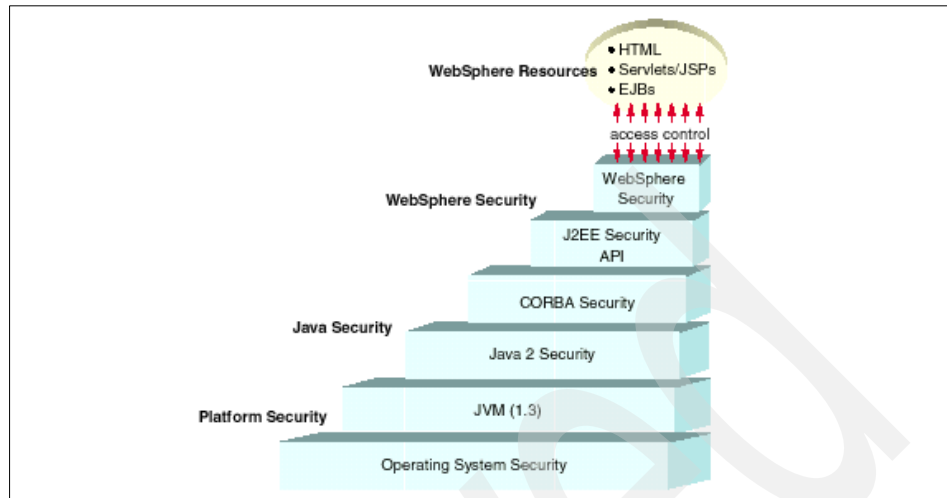


Figure 8-3 WebSphere security layers

For more information on WebSphere security topics see the *IBM WebSphere V4.0 Advanced Edition Handbook*, SG24-6176.

## 8.6.2 Configuring WebSphere Application Server security

In this section we show you how to set the global security settings to secure the WebSphere administrative server.

From the WebSphere administrative console perform the following steps:

1. Start the Security Center by selecting **Console** from the tool bar, then click **Security Center** from the drop-down menu. See Figure 8-4 on page 350.

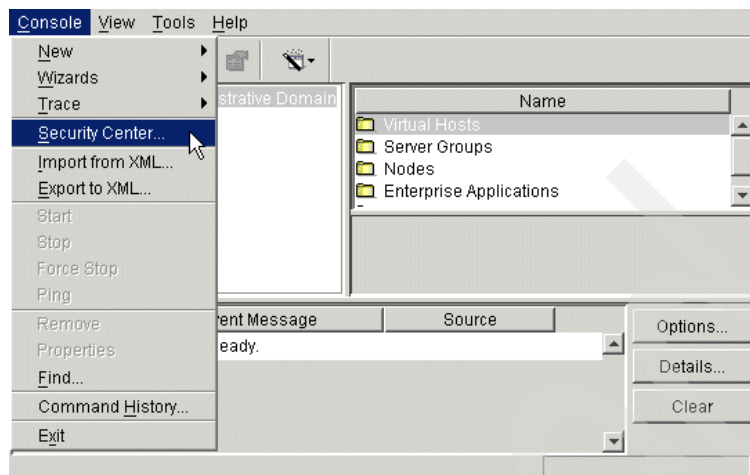


Figure 8-4 Starting Security Center

2. When the Security Center starts, check the **Enable Security** option, as shown in Figure 8-5.

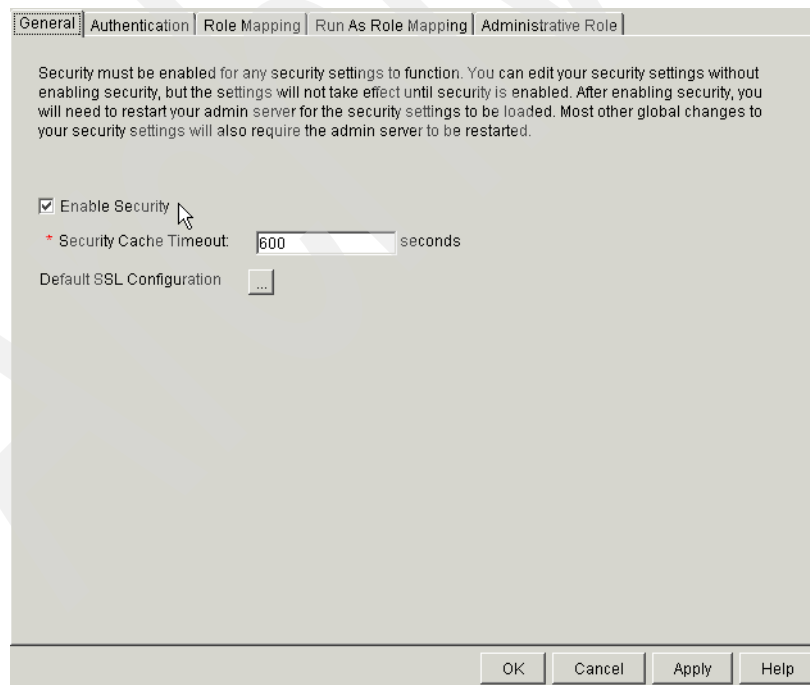


Figure 8-5 Enable WebSphere security

3. Select the **Authentication** tab and then select **Lightweight Third Party Authentication (LTPA)**.

The screenshot shows the 'Authentication' tab of the Security Center configuration window. The 'Authentication Mechanism' is set to 'Lightweight Third Party Authentication (LTPA)'. The 'LTPA Settings' section includes a 'Token Expiration' of 120 minutes, an unchecked 'Enable Single Sign On (SSO)' checkbox, a 'Domain' field, an unchecked 'Limit to SSL connections only' checkbox, and an unchecked 'Enable Web trust association' checkbox. There are buttons for 'Generate Keys...', 'Import Key...', and 'Export Key...'. Below this, the 'LDAP' radio button is selected, and the 'LDAP Settings' section contains fields for 'Security Server ID', 'Security Server Password', 'Host', 'Directory Type' (set to 'Netscape'), 'Port', 'Base Distinguished Name', 'Bind Distinguished Name', and 'Bind Password'. There are also 'Advanced...' and 'SSL Configuration' buttons. At the bottom are 'OK', 'Cancel', 'Apply', and 'Help' buttons.

Figure 8-6 Selecting LTPA authentication

4. Check the **LDAP** option to display LDAP settings and fill in the following parameters as also shown Figure 8-7 on page 353:

**Security server ID**

Set the LDAP user ID you want to use to access the administrative server. In our example it would be the user ID (uid) in the LDAP directory WENDY, but if are using the Domino LDAP it would be the short name of the administrator aadmina. If you do not add more users via the Administrative Role tab of the Security Center, the user ID entered for this parameter will be the only one allowed to access the WebSphere administrative console.

**Security server password**

This is the valid password for the Security server ID. This field is case-sensitive.

**Host**

This must be the fully qualified host name on which the LDAP server is running, in our

scenario this is ralyas4b.iseries.itso.ral.ibm.com but if using Domino as the LDAP server it would be the fully-qualified name of the Domino server for example

**domweba.iseries.itso.ral.ibm.com**

**Directory type**

This value must be set for the type of LDAP server you are using. In our scenario we selected SecureWay for the IBM SecureWay LDAP directory. Remember, the iSeries LDAP Directory Services server is an implementation of IBM's SecureWay Directory. In case you want to use Domino LDAP then you would select Domino 5.0.

**Port**

This is the port on which the LDAP directory server listens on. You may leave this field blank for the default non-SSL port of the LDAP directory (port 389).

**Base distinguished name**

This is the distinguished name (DN) of the directory in which searches begin within the LDAP directory. In our scenario, for a user with a DN of cn=Wendy Thomson, ou=people, o=iseriesshop, you could specify a base DN of ou=people, o=iseriesshop. The base DN is the place in the directory where users you want to use for authentication are published. This is a required field for all LDAP directories except the Domino Directory.

**Attention:** If you are using the Domino Directory, and you specify a base distinguished name, you can not grant permissions to individual Web users for resources managed by your WebSphere Application Server.

**Bind distinguished name**

This is the DN of the user who is capable of performing searches on the directory. In most cases, this field is not required since all users are usually authorized to search an LDAP directory. However, if the LDAP directory contents are protected from all LDAP users, you need to specify the DN of an authorized user, such as the administrator of the directory (for example, cn=Administrator).

## Bind Password

This is the valid password for the user specified as the bind distinguished name. This is required only if you specify a value for Bind Distinguished Name. This field is case-sensitive.

The screenshot shows the 'Security Center' window with the 'Authentication' tab selected. The 'Authentication Mechanism' is set to 'Lightweight Third Party Authentication (LTPA)'. Under 'LTPA Settings', 'Token Expiration' is 120 minutes, 'Enable Single Sign On (SSO)' is unchecked, 'Domain' is 'ISERIES.ITSO.RAL.IBM.COM', 'Limit to SSL connections only' is unchecked, and 'Enable Web trust association' is unchecked. There are buttons for 'Generate Keys...', 'Import Key...', and 'Export Key...'. Below this, 'LDAP' is selected as the authentication method. Under 'LDAP Settings', 'Security Server ID' is 'WENDY', 'Security Server Password' is masked with asterisks, 'Host' is 'RALYAS4B.ISERIES', 'Directory Type' is 'SecureWay', 'Port' is empty, 'Base Distinguished Name' is 'ou=people,o=iserie', 'Bind Distinguished Name' is empty, and 'Bind Password' is empty. There are buttons for 'Advanced...' and 'SSL Configuration'.

Figure 8-7 Selecting the LDAP settings

5. Once you have entered all the LDAP settings click **OK** to continue.

**Tip:** Via the Authentication tab you can also configure whether the WebSphere Application Server will connect to the LDAP server using an SSL connection. The Advanced button allows you to modify the LDAP search filters. For example, the default for the search filter specifies that the username entered in a browser's authentication prompt is checked against a directory entry that has the username in the uid attribute. In case your unique user identifier is an e-mail address, you would need to modify the search filter to search the username (e-mail address) in the mail attribute of an LDAP directory entry.

6. Click **OK** on the information message dialog window warning that changes will not take effect until the administrative server is restarted.

7. Right-click your node in our example it is RALYAS4A, to restart the WebSphere Application Server, as shown in Figure 8-8.

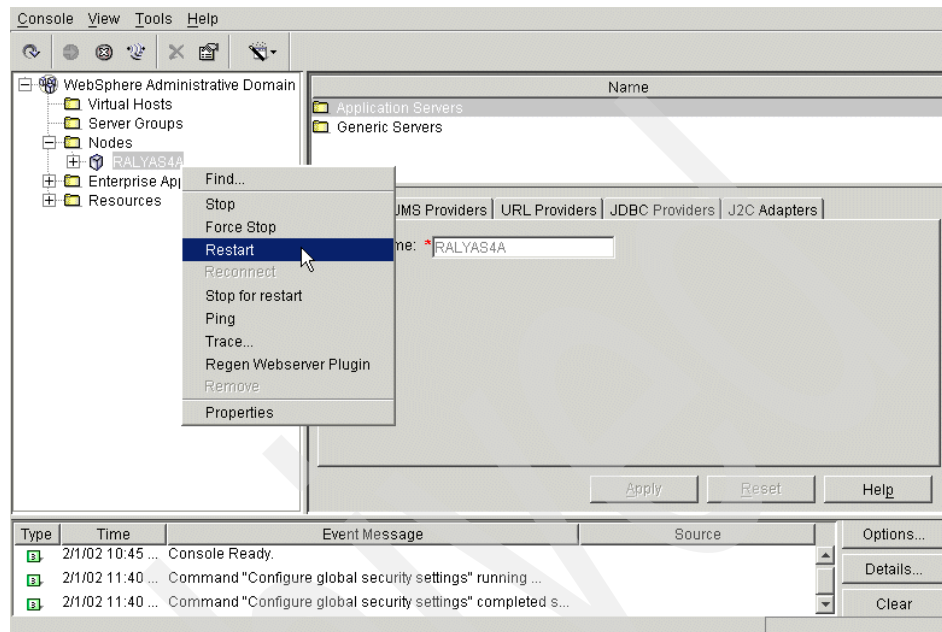


Figure 8-8 Restart Node

8. Click **Yes** on the confirmation dialog box.

At this point you have only enabled security on the WebSphere administration console. Next time you start the console you are prompted for a user id and password. This will be the user id and password you entered in step 4 on page 351. In the next section we describe how to protect the WebSphere application resource. In our scenario it is a servlet called SimpleServlet.

### 8.6.3 Protecting WebSphere resources

This section describes how to protect the WebSphere servlet called SimpleServlet. If your company requirements are to secure a complete enterprise application then you should refer to *IBM WebSphere V4.0 Advanced Edition Handbook*, SG24-6176, for more information.

#### Configure security roles

A security role is a logical grouping of principals. Access to operations is controlled by granting access to a role.



To add a security role perform the following steps:

1. From the WebSphere administrative console, start the Application Assembly Tool (AAT) by selecting **Tools -> Application Assembly Tool** from the main menu.

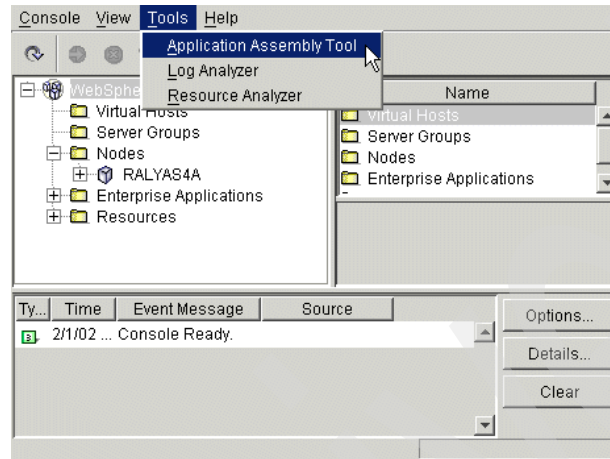


Figure 8-9 Starting Application Assembly Tool

2. In this scenario the Web application already exists and is packaged in an Enterprise Application Archive (EAR). To open the archive, select the **Existing** tab in the Welcome to Application Assembly Tool window.
3. Select the Enterprise Application Archive you want to secure, in our example it is J:\DomWas\DomApp.ear and then click **OK**.
4. Right-click **Security Roles** and select **New** from the drop-down menu, as shown in Figure 8-10 on page 356.

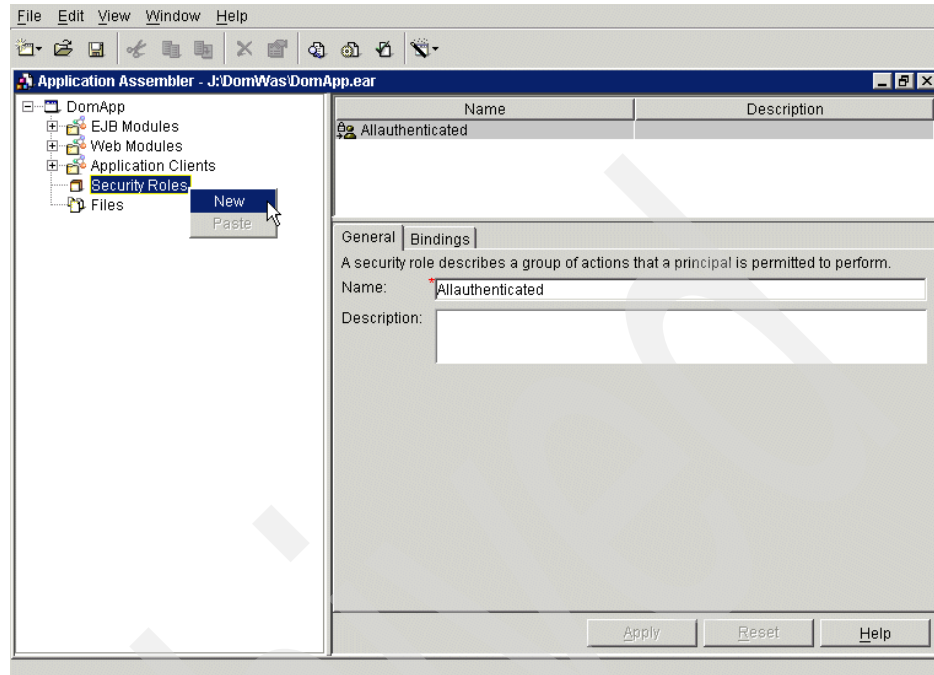


Figure 8-10 Adding a new application security role

5. In the New security role window enter the following parameters under the **General** tab. See Figure 8-11 on page 357.

<b>Name</b>	This can be any name. In this scenario we used Allauthenticated.
<b>Description</b>	Optionally add a description of the security role.

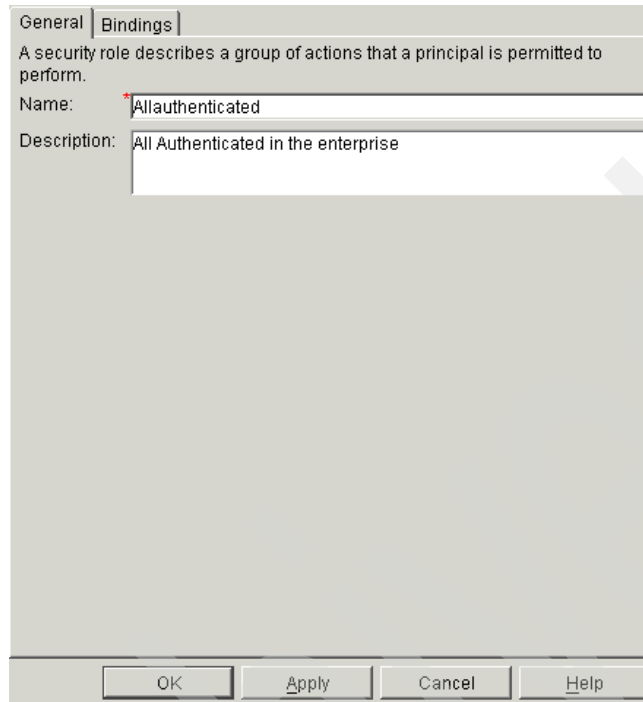


Figure 8-11 Security Roles Properties - General tab

6. Click the **Bindings** tab to set binding parameters as follows:
  - Under Groups click **Add...** to grant the security role to individual user groups. In this scenario we did not grant permissions to individual user groups.
  - Under Users click **Add...** to grant the security role to individual users. In this scenario we did not grant permissions to individual users.
  - Under Special Subjects click **Add....** to grant the security role to one of two special categories of users:
    - Choosing **All authenticated users** grants the security role to any user who can authenticate by using a valid user ID and password. In this scenario, we selected this option specifying that the users stored in the LDAP directory on server RALYAS4B are granted permission.
    - Choosing **Everyone** grants the security role to all users, including those who did not authenticate by using a valid user ID and password. In other words, a method on an enterprise bean or a URI is unprotected if any of the required roles for that method are granted to the special subject Everyone.

We granted the new security role to **All authenticated users**, as shown in Figure 8-12.

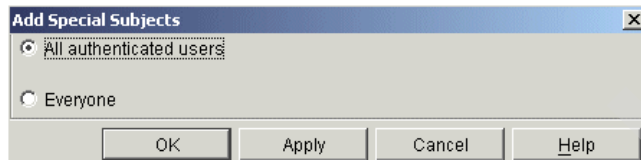


Figure 8-12 Granting a security role to a special subject

Our new security role bindings information is shown in Figure 8-13.

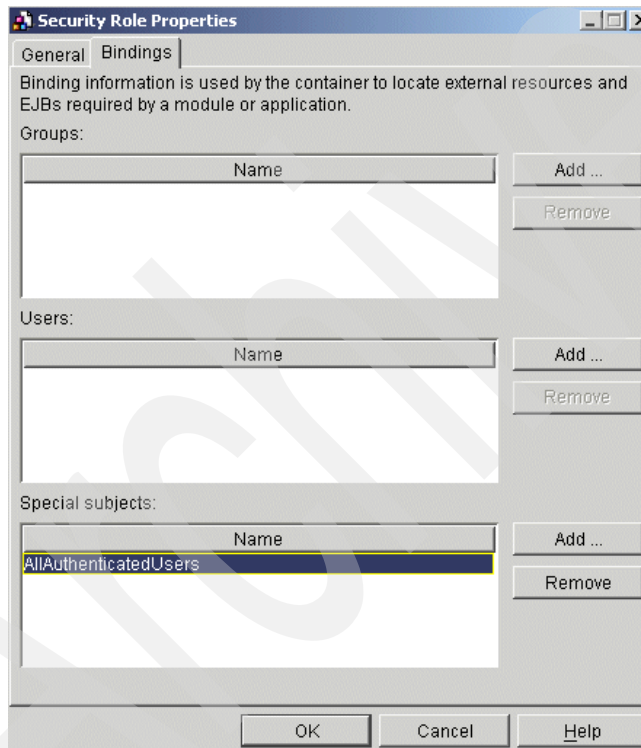


Figure 8-13 Security Roles Properties - Bindings tab

7. Click **OK** to create the new security role.

## Configuring Web module security

These tasks are needed to secure Web applications. Once again we use the Application Assembly Tools to:

- ▶ Configure security constraints to specify how content should be protected.
- ▶ Use login configuration to select and configure each of the supported authentication mechanisms:
  - Basic authentication
  - Form authentication
  - Client certificate authentication

### *Configuring security constraints*

Security constraints specify how Web content is to be protected. These properties associate security constraints with one or more Web resource collections. A constraint consists of a Web resource collection, an authorization constraint, and a user data constraint. It is assumed that the Application Assembly Tool within the enterprise application archive is still opened.

To add security constraints perform the following steps:

1. Expand **Web Modules** by clicking the + sign next to it.
2. Expand the Web module you want to work with. In this scenario it is the DomWebApp Web module.
3. Right-click **Security Constraints** and select **New** from the drop-down menu, as shown in Figure 8-14.

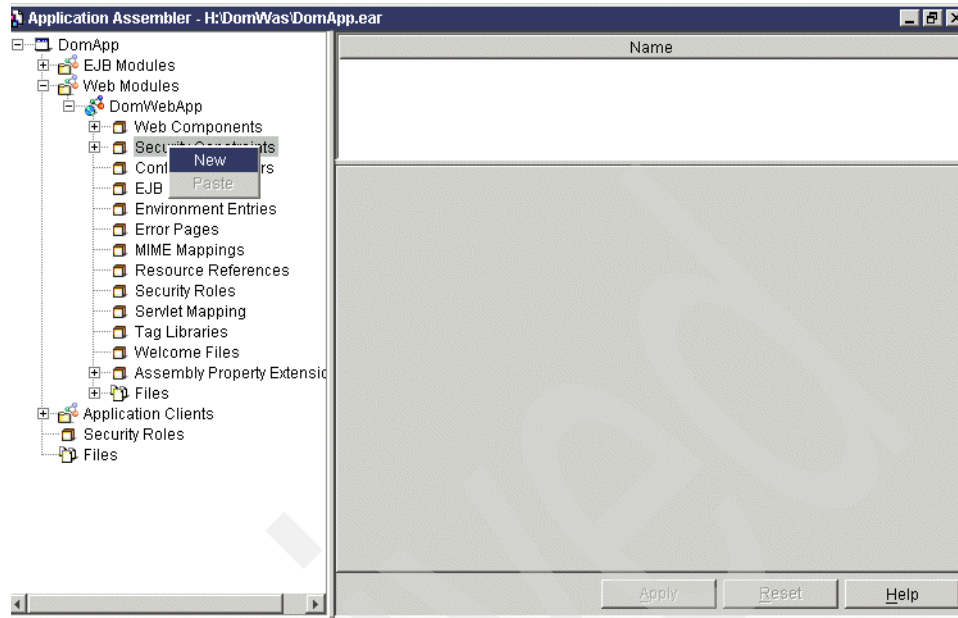


Figure 8-14 Add a new Web module security constraints

4. In the New Security Constraints window enter the following parameters:
  - a. Set the Security constraint name.
  - b. Click **Add...** under the Authorization Constraints section and select the roles that are permitted access to this resource collection. In this case it is the security role Allauthenticated as defined in step 5 on page 356. After you selected the role click **OK** to return to the New Security Constraint window.
  - c. In the User Data Constraint section, select the Transport guarantee. In this scenario we selected **None**.

Transport guarantee indicates how data communicated between the client and the server is to be protected. The options are:

- *None* means that the application does not require any transport guarantees.
- *Integral* means that the application requires that the data sent between the client and the server must be sent in such a way that it cannot be changed in transit.
- *Confidential* means that the application requires that the data must be transmitted in a way that prevents other entities from observing the contents of the transmission.

In most cases, Integral or Confidential indicates that the use of SSL is required.

The new security constraints properties are shown in Figure 8-15.

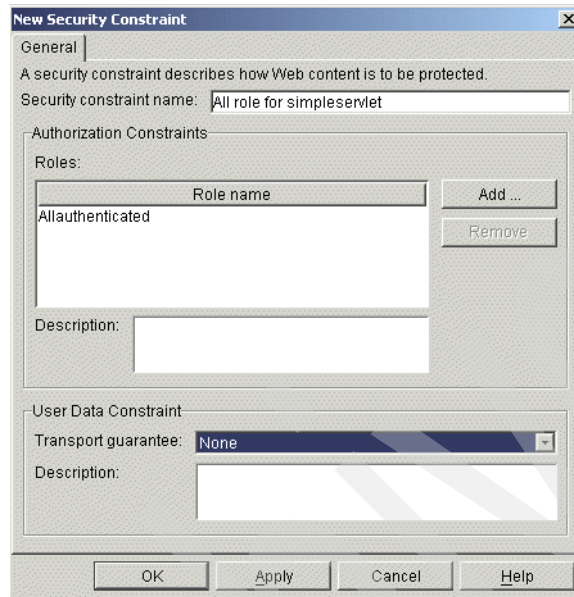


Figure 8-15 New Security Constraint properties

5. Click **OK** to create the new security constraint.

You have just created a new security constraint and now you can associate a Web resource with this security constraint.

1. Expand **Security Constraints**.
2. Expand the new security constraint you just created.
3. Right-click **Web Resource Collections** and select **New** from the pop-up menu, as shown in Figure 8-16 on page 362.

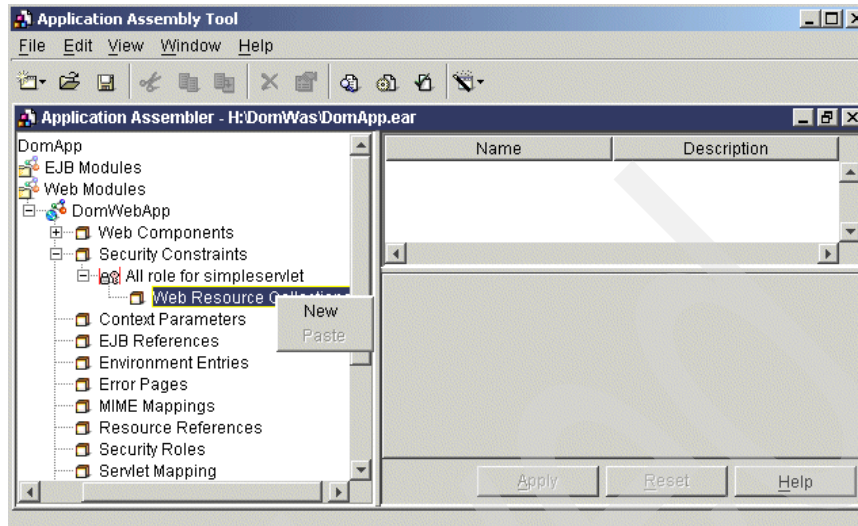


Figure 8-16 Adding new Web Resource Collection to security constraint

4. In the New Web Resource Collection Properties window:
  - a. Set the **Web Resource Name**.
  - b. Optionally add a description of the Web resource collection.
  - c. Under HTTP methods, click **Add...** and select the HTTP methods to which the security constraint applies, as shown in Figure 8-17. After you select the HTTP method click **OK** to add the selection to the new constraint and return to the New Web Resource Collection window.



Figure 8-17 Adding a HTTP method to a Web resource collection

If no HTTP methods are specified, then the security constraint applies to all HTTP methods. The valid values are GET, POST, PUT, DELETE, HEAD, OPTIONS, and TRACE.

- d. Under URLs, click **Add...** and set the URL pattern for resources in the Web application, as shown in Figure 8-18 on page 363.



**Important:** All requests that contain a request path that matches the URL pattern are subject to the security constraint. Do not include the context root for the Web module in the URL pattern.

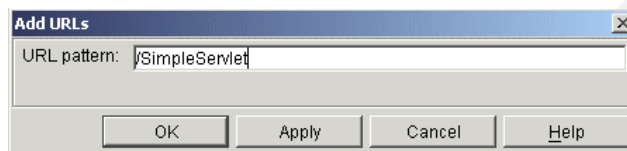


Figure 8-18 Adding a URL pattern for a Web resource collection

Our new Web resource collection properties are shown in Figure 8-19.

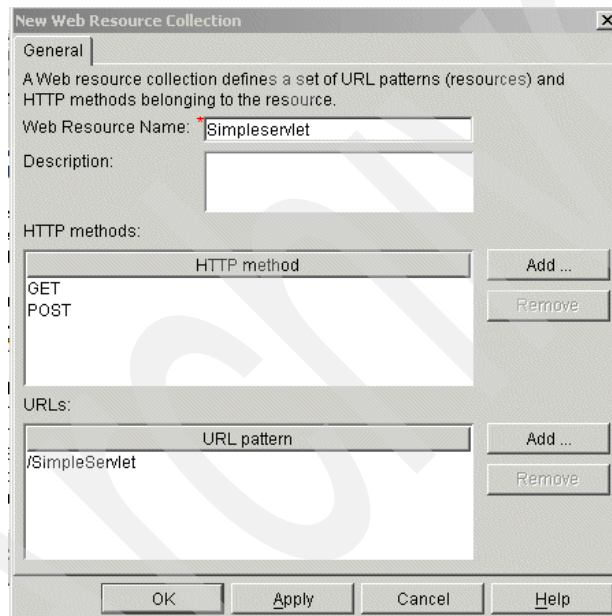


Figure 8-19 New web Resource Collection properties

5. Click **OK** to create the new Web resource collection.

## Login configuration

To gain access to Web resources protected by an authorization constraint, a user must authenticate using the configured mechanism. Web applications can authenticate a user to a Web server using one of the following authentication methods:

- ▶ *Basic* authentication uses the model where the Web server requests the Web client to provide a username and a password for authentication, in each realm. HTTP basic authentication is not a secure protocol because the user password is transmitted with a simple Base64 encoding and the target server is not authenticated.
- ▶ *Digest* authentication transmits the password in encrypted form. Digest authentication is not supported by WebSphere Version 4.0.
- ▶ *Form* authentication allows the Web developer to control the appearance of login windows using custom HTML login forms.
- ▶ *Client certificate* authentication uses HTTPS (HTTP over SSL) and requires the user to possess a public key certificate.

Not all login configurations are supported by all of the WebSphere global security authentication mechanisms. The authentication mechanisms are the local operating system and LTPA with an LDAP directory or a custom user registry. Table 8-1 shows which authentication mechanisms are capable of supporting the various authentication methods.

Table 8-1 Supported authentication methods

Authentication methods	Local OS	LTPA (LDAP, Custom)
Basic	Yes	Yes
Form	Yes	Yes
Client certificate	Not supported	Yes
Digest	Not supported	Not supported

The Login configuration properties are used to select and configure the authentication method that should be used.

### Configuring basic authentication

To configure basic authentication for a Web module do the following:

1. In AAT, right-click your Web module and select **Properties** from the pop-up menu, as shown in Figure 8-20 on page 365. In this scenario this would be the DomWebApp module.

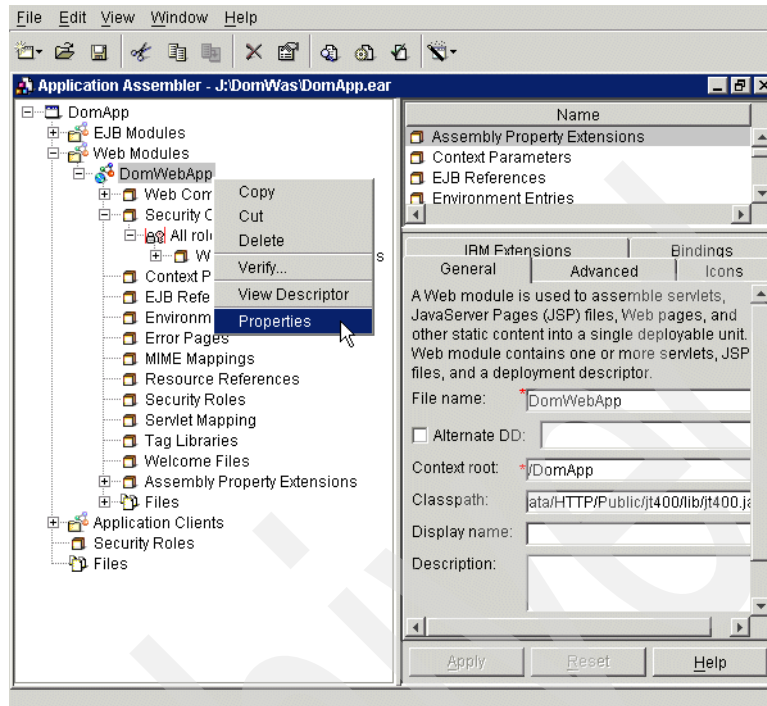


Figure 8-20 Selecting Web module properties

2. In the Web Module Properties window click the **Advanced** tab to set login configuration properties:
  - a. Check the **Login configuration** box.
  - b. Set the Authentication method to **Basic**.
  - c. Enter a realm name. In this scenario we entered DomApp.

A HTTP realm is a string that allows URIs to be grouped together. For example, if a user accesses a secured resource on a Web server within the finance realm, subsequent access to the same or different resources within the same realm does not result in a repeat prompt for a user ID and password.

Our scenario login configuration settings are shown in Figure 8-21 on page 366.

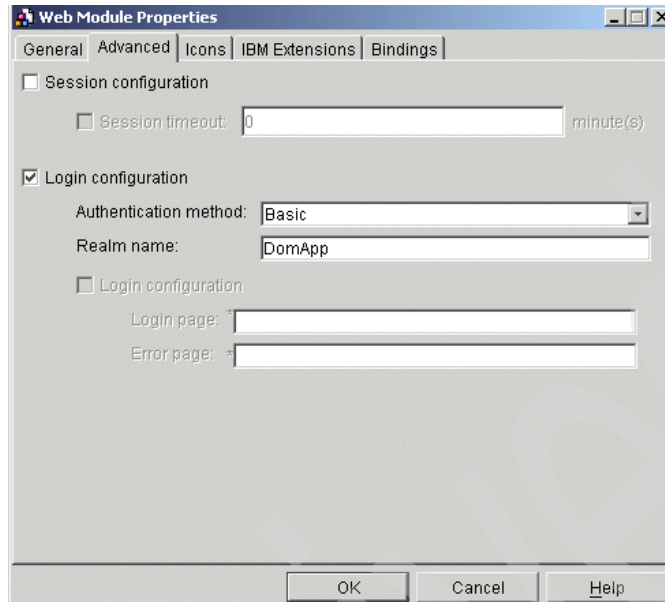


Figure 8-21 Web Module Properties for basic authentication

3. Click **OK** to update the login configuration settings.

When an unauthenticated user attempts to access a protected resource in the deployed Web module, the basic authentication method uses the basic HTTP authentication challenge to request the username and password.

This completes all the resource security configuration that we require for the redbook scenario. After the configuration has been completed, you need to save the configuration and re-deploy the application to the application server. These tasks are outside the scope of this book. For more information on WebSphere security and application deployment, refer to the *IBM WebSphere V4.0 Advanced Edition Handbook*, SG24-6176.

#### 8.6.4 Verifying WebSphere Application Server authentication

Now that your servlet has been secured on the WebSphere Application Server, you can test it by opening the URL through a Web browser.

1. Open your browser and enter the URL. In our scenario it looks like the following:

<http://rallyas4a.iseries.itso.ra1.ibm.com/DomApp/SimpleServlet>

If you have successfully completed the WebSphere Application Server security setup you should be prompted to sign on with the basic login, as shown in Figure 8-22.

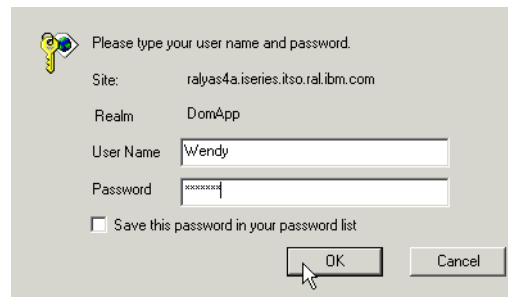


Figure 8-22 SimpleServlet login

2. Enter your username and password, then click **OK**. According to the configuration, this username corresponds to a uid attribute of an entry in the LDAP directory on server RALYAS4B.
3. If you have set your browser preferences to prompt you before accepting cookies, then a cookie will be displayed, as shown in Figure 8-23.

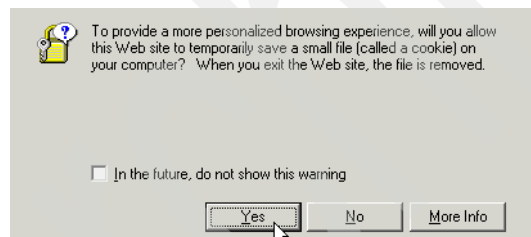


Figure 8-23 SimpleServlet cookie

4. Click **OK** to accept the cookie. As previously described, the SSO function with LTPA uses cookies to store encrypted information about the session on the user's PC.

The SimpleServlet Web page appears.

## 8.7 Configure Domino HTTP server

As Domino does not currently support the IBM HTTP Server for iSeries (powered by Apache) you need to configure the Domino HTTP server for Single Sign-On. The next release of Domino will support the powered by Apache server.

In this section you learn how to configure the Domino HTTP server for use with the WebSphere Application Server. To do this, perform the following steps:

1. Update the Domino server document.
  - a. From the Domino administrator, connect to the appropriate Domino server, select your current server document and click **Edit Server** to edit the server document for your Domino server.
  - b. Within in the Domino server document, select the **Internet Protocols** tab and then the **HTTP** sub-tab.
  - c. In the middle of the right column, for the value of the DSAPI filter file names field, enter:  
`/QSYS.LIB/QEJBADV4.LIB/LIBDOMINOH.SRVPGM`

Figure 8-24 shows the DSAPI filter setting in the Domino server document.

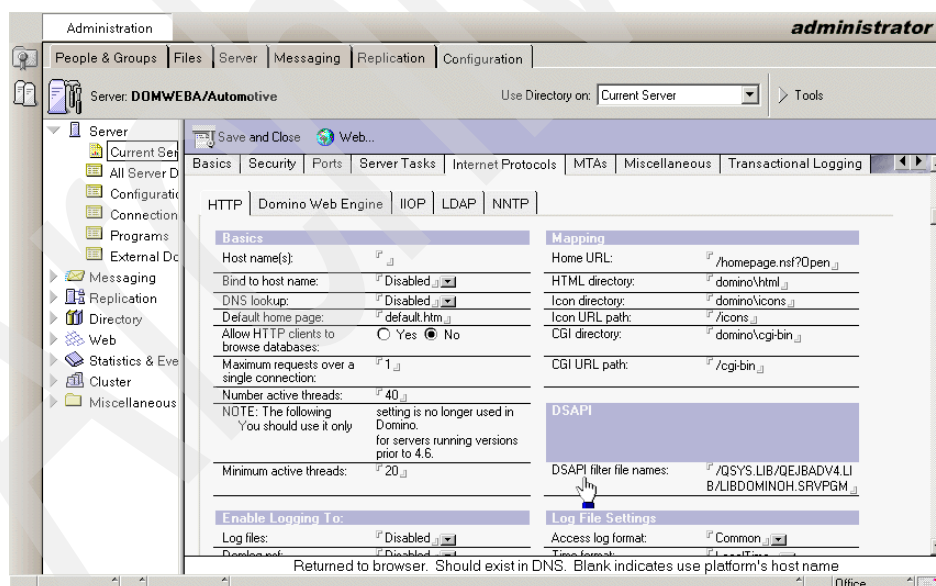


Figure 8-24 Adding the DSAPI filter

2. Click **Save and Close** to save and exit the Domino server document. The DSAPI plug-in will be loaded when the Domino HTTP server task is next loaded.

The filter file (LIBDOMINOH.SRVPGM) specified here is the WebSphere DSAPI plug-in for Domino for iSeries. It passes requests to the WebSphere Application Server. This behavior is transparent to the Web browser.

The Domino server Application Programming Interface (DSAPI) is a C API that lets you write your own extensions to the Domino HTTP server. These extensions or filters let you customize the authentication of Web users.

3. You now need to update the Domino server's NOTES.INI file. Add the following line to the Domino server's NOTES.INI file:

```
WebSphereInit=/qibm/UserData/WebASAdv4/default/config/plugin-cfg.xml
```

For information on how to edit the Domino for iSeries server NOTES.INI file refer to the redbook *Lotus Domino for AS/400 R5: Implementation*, SG24-5592.

**Note:** The line above is for the WebSphere Application Server's default instance. For a non-default instance, modify the line to reference the directory of the WebSphere Application Server instance. For example, if you have an instance named WAS01, you should change the line as follows:

```
WebSphereInit=/qibm/UserData/WebASAdv4/WAS01/config/plugin-cfg.xml
```

The plugin-cfg.xml file includes some configuration information for the WebSphere DSAPI plug-in. This information allows the plug-in to connect the Domino HTTP server and WebSphere together.

4. Restart the Domino server's HTTP task. Enter the following command from the Domino server console to restart the Domino server's HTTP task:  

```
tell http restart
```
5. Update OS/400 authorities. You need to grant the user profile QNOTES the authority needed to be able to create the necessary WebSphere Application Server log files. Enter the following Change Authority (CHGAUT) command from an OS/400 command line:

```
CHGAUT OBJ('/QIBM/UserData/WebASAdv4/default/logs')USER(QNOTES)DTAAUT(*RWX)
```

**Note:** The change authority command above is for the WebSphere Application Server's default instance. For a non-default instance, modify the OBJ parameter to reference the directory of the WebSphere Application Server instance. For example, if you have a instance named WAS01, you should change the command as follows:

```
CHGAUT OBJ('/QIBM/UserData/WebASAdv4/WAS01/logs') USER(QNOTES) DTAAUT(*RWX)
```

## 8.8 Enabling Domino authentication

Domino security is integrated into the product and is always active. In this section the Domino application is configured so that a Web browser can only open the database after supplying a valid username and password. The username and password that are used must exist in the LDAP directory.

### 8.8.1 Domino authentication concepts

The implementation of access control is hierarchical, as described in the following list:

- ▶ **Server access**

For Web browsers, this is implemented by simply not allowing anonymous access to any database on a Domino server and prohibiting browsing the Domino file system from a browser. For a Notes client, there are two server access lists:

- **Deny access:** Lists users or groups who will explicitly be denied access to the server.
- **Allow access:** Lists groups and users allowed access (provided they are not on the Deny Access list).

- ▶ **Database access**

This is implemented by an access control list (ACL) in the database. The ACL lists users (and groups containing users) and their access rights on a scale of None, Depositor, Reader, Author, Editor, Designer and Manager. For those levels allowing writing or alteration of database documents (records), users can also be independently controlled from creating or deleting documents. Generally, Web users would have Reader or Author access; the higher levels are applicable to a Notes client. These levels can be further refined by the use of roles. Roles allow creating of subsets of users in the ACL to be granted access rights to the database or individual documents (records) in the database, as well as application-defined rights.



► Document access

A secondary ACL within a document (a Readers or Authors field) allows refinement of the database access. In other words, it is possible to further restrict access to documents within a database, but it is not possible to increase a user's access rights in this way. Thus, if a user is in the Authors field for a document (implying the right to change the document's contents), but is only a Reader in the database ACL, they will not be granted update access to the document.

► Section level access

It is possible to prevent a user from opening and viewing a section of a document. For Notes clients this is not a security feature since they have other methods to view the hidden fields in the section; however, for a browser this is effective since the contents of a section a user does not have access to are not sent to their Web browser. Further levels of access control are available, but these are all that typically are relevant to a Web-based Domino application. For our purposes, we will simply prevent unauthenticated (Anonymous) access to Domino databases on our test server, but allow general authenticated access to databases. This is why during Domino installation we set the default access control on all Domino databases to be None for unauthenticated users.

Domino security implementation is resource based; the objects being protected carry access control lists defining the rights of users to access and alter the object. Thus the security model does not depend on context. However, there is no central repository of user rights since these are stored with each protected resource.

## 8.8.2 Configure Domino to use iSeries LDAP

For Domino to use the OS/400 LDAP server instead of its own LDAP, you must create a Directory Assistance database that points to the OS/400 LDAP server. To create a Directory Assistance database in Domino, perform the following steps:

1. From the Domino Administrator client pull-down menu, select **File -> Database -> New**.
2. From the New Database window, enter the following values, as shown in Figure 8-25 on page 372:

<b>Server</b>	Select your Domino server. In this scenario the Domino server is DOMWEBBA/Automotive.
<b>Title</b>	This can be any title you like. We entered Directory Assistance.

**File name** This can be any file name you like. In this scenario we used Director.nsf.

3. Click **Template Server** and select your Domino server. In our scenario this is DOMWEBA/Automotive.
4. Select the Directory Assistance template (DA50.NTF).

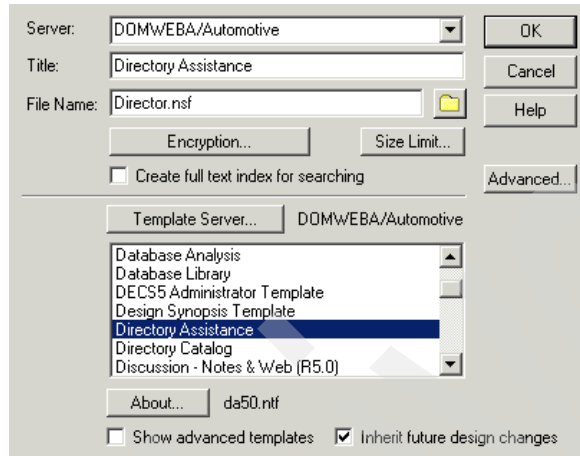


Figure 8-25 Create Directory Assistance database

5. Make sure Inherit future design changes is selected and click **OK**.
6. When the Directory Assistance About document appears, press **Esc**.
7. You are now in the Directory Assistance database. Select **Add Directory Assistance**, as shown in Figure 8-26.

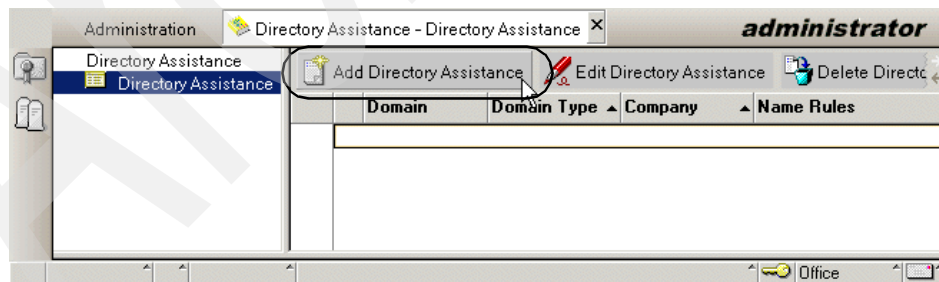


Figure 8-26 Add Domino Directory database

8. In the Directory Assistance document, select the **Basics** tab and enter the following values, as shown in Figure 8-27. Note that some of the parameters do not show up when not all options are selected as listed:

<b>Domain type</b>	Choose the type of Directory, in our case this is LDAP.
<b>Domain name</b>	Enter the name of the Domain the record describes. Domain names must be unique; in our scenario we entered SecureWay.
<b>Company name</b>	Enter the name of the company that this directory is associated with; in our example we used IBM.
<b>Search order</b>	This would be a number representing the order in which the directory is searched, relative to other directories in the Directory Assistance database. In this scenario we left this field blank.
<b>Group expansion</b>	Select <b>Yes</b> ; this allows Directory Assistance to verify Web user membership in a group in this LDAP directory when a group is included in a database ACL that a Web user is attempting to access on a Domino Web server.
<b>Nested group expansion</b>	Select <b>Yes</b> . If this field is set to No the Domino server will <i>not</i> expand the nested groups at the remote LDAP to improve performance.
<b>Enabled</b>	Select <b>Yes</b> to enable Directory Assistance for this directory.

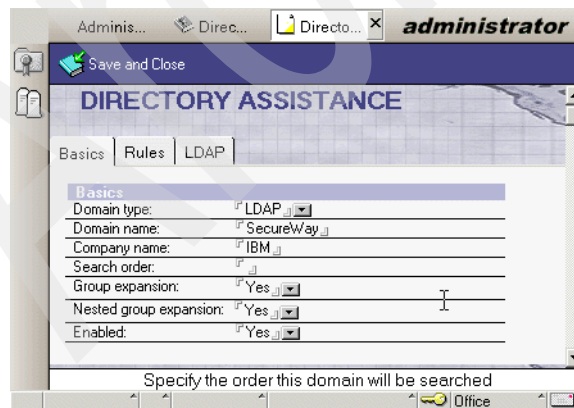


Figure 8-27 Directory Assistance: Basic tab

9. Click the **Rules** tab and fill in the fields for Rule 1 with the following values, as shown in Figure 8-28:

**OrgUnit 4-1, organization, country** OrgUnit4 through to OrgUnit1 has a default of asterisk (\*), meaning any organizational unit will be accepted. Blank would mean no organizational unit, or you could specify an organizational unit. This is the same for the Organization and Country fields. In our example we left these fields as their defaults.

**Enabled**

To enable the rule ensure that Enable is set to **Yes**.

**Trusted for credentials**

Specifies whether security information should be retrieved from this domain for users matching this rule. Ensure that Trusted for Credentials is set to **Yes**.

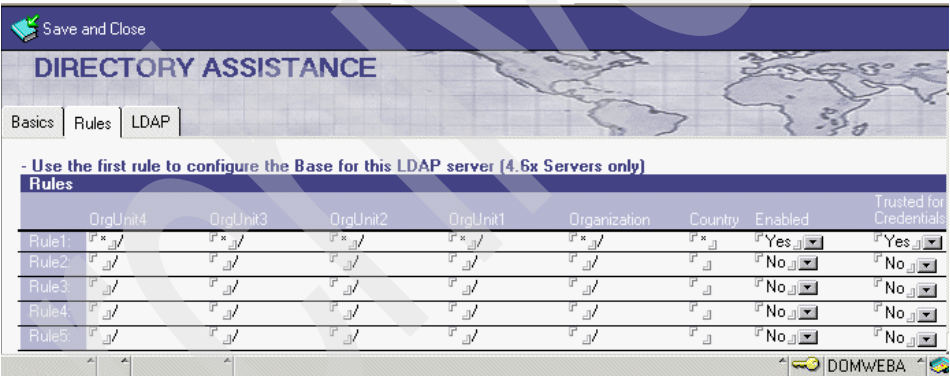


Figure 8-28 Directory Assistance: Rules tab

10. Select the **LDAP** tab and fill in the fields with the following values, as shown in Figure 8-29 on page 376:

**Host name** Your fully-qualified LDAP server host name. For this scenario it is the host name of the iSeries server RALYAS4B.ISERIES.ITSO.RAL.IBM.COM.

Optional Authentication Credentials

**Username** Your OS/400 LDAP administrator. In our example this is cn=administrator.

**Credentials password**

Your OS/400 LDAP administrator's password. In our example this is my51dap.

**Base DN for search**

The distinguished name of the entry that holds the published user information. The base DN specifies where to search for user authentication information within a directory information tree. In this scenario it is ou=people,o=iseriesshop.

**Perform LDAP search for**

For mail recipients lookup for Notes clients and Web Authentication for Notes clients and other Web clients, select

**NotesClients/WebAuthentication.**

If you select LDAP clients, the directory entry will be passed to LDAP clients as a referral URL. For our scenario we only need to select **NotesClients/WebAuthentication.**

**Channel encryption**

If set to SSL, the Domino server will use SSL, to connect to the LDAP server. In our example we do not use SSL and therefore set the value to **None.**

**Port**

The port to use when connecting to the LDAP server. In our example we are using the default port of 389 for non-SSL connections.

**Timeout**

This is the maximum number of seconds before a search is terminated. In our example it was left as the default of 60 seconds.

**Maximum number of entries returned**

The maximum number of entries a single search can return. We left it as the default of 100 entries.

**Dereference alias on search**

How alias objects should be handled in searching. In our example it was left as the default of Always.

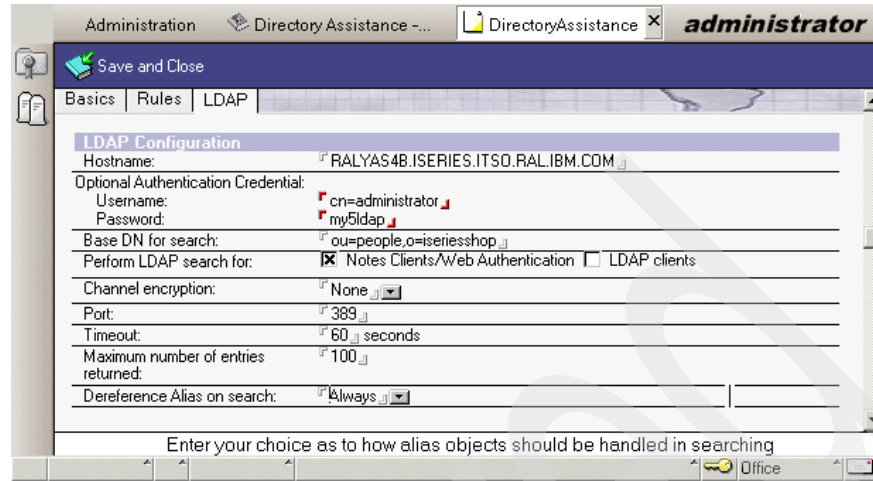


Figure 8-29 Directory Assistance: LDAP tab

11. Click **Save and Close** to save and exit this document.

To change the Domino server to use iSeries LDAP you must update the Domino server document.

1. From Domino Administrator client, select the **Configuration** tab.
2. Select your current Domino server document and click **Edit Server**.
3. In the server document select the **Basic** tab.
4. In the Directory Assistance database name field, type the filename of your new Director Assistance database that you created in step 2 on page 371. In this scenario it was Director.nsf, as shown in Figure 8-30.

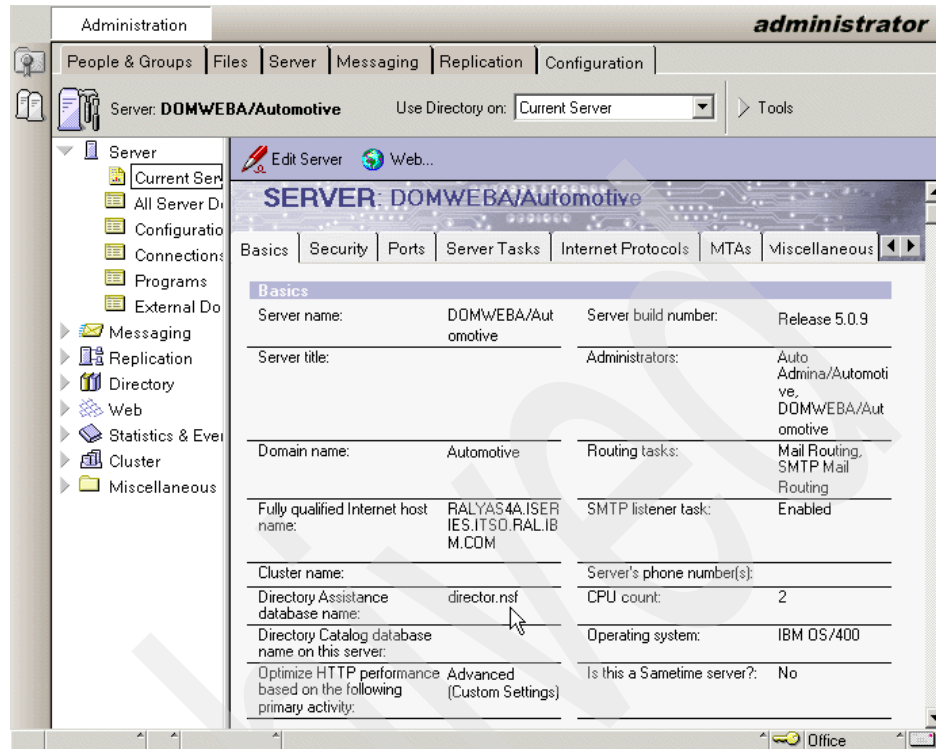


Figure 8-30 Adding Directory Assistance database name to Domino server

5. Select the **Ports** tab, then select the **Internet Ports** sub-tab, then select the **Directory** sub-sub-tab and fill in the following values, as shown in Figure 8-31 on page 378:

**TCP/IP port number**

Set to the correct LDAP port for your iSeries LDAP server. In our example we left it as the default LDAP port of 389.

**TCP/IP port status**

Ensure this is set to Enabled.

**Authentication options**

Name & password field should be set to Yes.

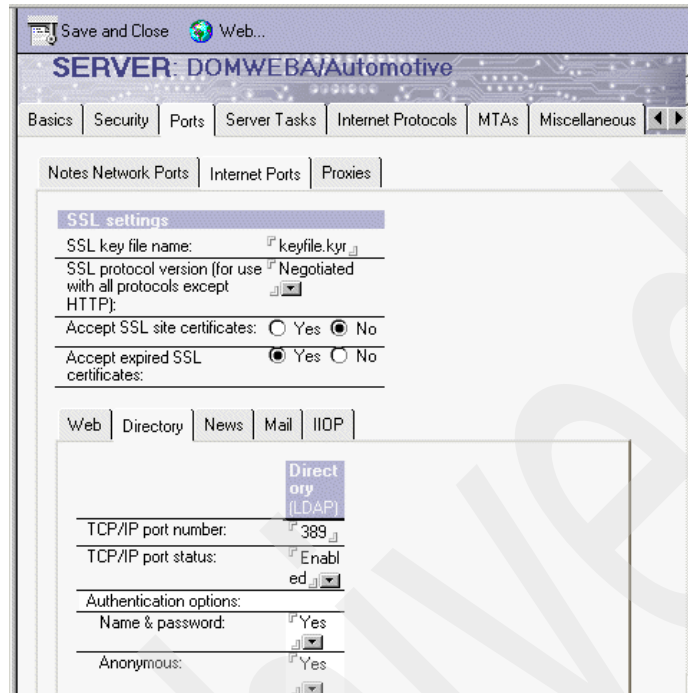


Figure 8-31 Changing the Directory port

6. Click **Save and Close** to save and exit the Domino server document.

### 8.8.3 Activating Domino Database security

You may now need to add new users from your OS/400 LDAP directory to your Domino database ACLs. This task is required to authorize authenticated users to access your application database and can be done by performing the following steps:

1. From your Domino Administrator client, select the **Files** tab.
2. Right-click on the Domino database you want to protect. In our scenario this is domwaslab.nsf. From the drop-down menu select **Access Control -> Manage**.
3. From the Access Control List window, click **Add** to add a new user to the database ACL. Type Anonymous into the Person, server or group field and click **OK**.
4. For access, select **No Access**. Deselect **Read public documents** and **Write public documents**. Click **OK**.



5. Change Default access to No Access. Deselect **Read public documents** and **Write public documents**. Click **OK**.
6. From the Access Control List window, click **Add** to add a new user to the database ACL.
7. In the Add User window, enter the information for your OS/400 LDAP user in the following format: `cn=wendy/ou=people/o=iseriesshop`, then click **OK**.

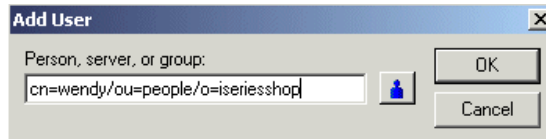


Figure 8-32 Adding a user in LDAP format

**Note:** The syntax here is different from LDAP syntax because a slash (/) is used instead of a comma (,). When a user is added, the format appears different from what you entered, that is, the hierarchy identifiers (cn=, o=, and ou=) disappear. This is expected. In case your directory tree follows a structure like `cn=thomas/cn=users/o=iseriesshop`, the user information in the ACL remains as entered, because the structure does not follow the typical organization-based Domino structure.

8. Ensure that the new ACL member has the correct authority for the database, then click **OK** to exit the ACL. In our example we gave WENDY manager access to the domwaslab database, as shown in Figure 8-33.

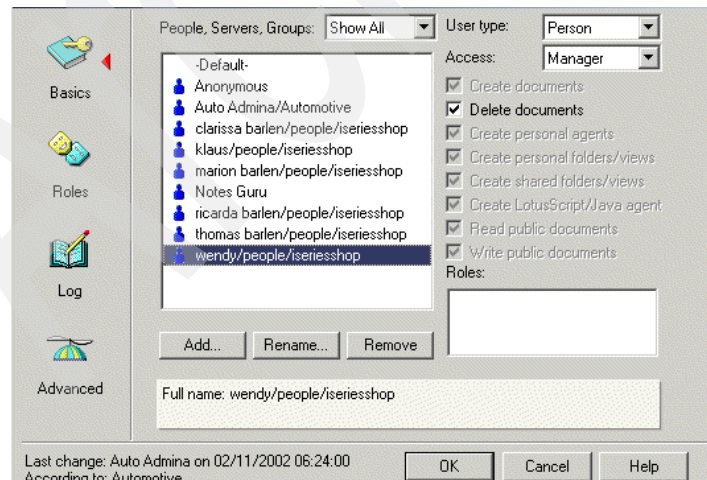


Figure 8-33 New iSeries LDAP user added to ACL

9. For these changes to take effect, the Domino server must be stopped and restarted. To do this, from your 5250 session, enter the OS/400 CL command **WRKDOMSVR DOMWEBA**.
10. From the Work with Domino Servers window, enter option 8 (Work console) next to your Domino server.
11. On the Work with Domino Console window, enter the following command:  
`restart server`
12. Press F5 until you see the Domino server restarted successfully.

### 8.8.4 Verifying Domino authentication

Now that your database has been secured on the Domino server, you can test it by opening the URL through a Web browser. We strongly recommend that you do not skip the verification. The test will show you whether your configuration and authentication setup works before proceeding to the Single Sign-On configuration. At this stage, your WebSphere application and Domino application still require to separate authentications.

1. Open your browser and enter the URL to access your Domino Web application. For the redbook scenario we used the following URL:  
<http://ralyas4a.iseries.itso.ral.ibm.com/domwaslab.nsf/loanapp?openform>
2. If you have successfully completed the Domino security section you should be prompted to sign on with the Domino application login, as shown in Figure 8-34.



Figure 8-34 Domino application login

3. Enter your username and password as stored in the LDAP directory and specified in the Domino database ACL, then click **Login**.

4. If you have set your browser preferences to prompt you before accepting cookies, then a cookie will be displayed, as shown in Figure 8-35.

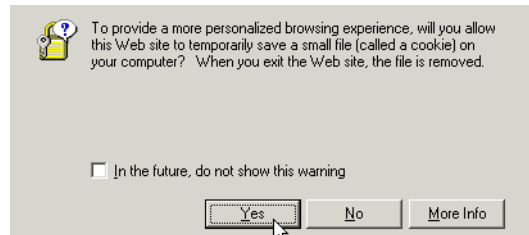


Figure 8-35 Domino Application cookie

5. Click **OK** to accept the cookie.

The Domino Web page appears.

## 8.9 Configuring Single Sign-On

For many Web applications, security is a main concern. Both WebSphere Application Server and Domino provide strong support for securing application access and data. Both products implement security mechanisms that involve determining and verifying user identity (authentication) and allowing access to protected resources to only designated users (authorization). However, the WebSphere and Domino security architectures are different.

Providing security for an application supported across both products requires establishing user identity and access controls in both products. One aspect of the Domino and WebSphere security models, the directory (referred to as the *user registry* in WebSphere), can be made common. This is possible since both Domino and WebSphere support the Lightweight Directory Access Protocol (LDAP) for directory access. The Domino Directory component provides for LDAP access, and WebSphere can be set up to use an LDAP-accessible directory as its user registry. Once a common user registry has been created by configuring Domino and WebSphere to rely on the same LDAP server for authentication, users will find it tedious to enter the same user ID and password multiple times, depending on whether they need to access a Domino object (a Domino database, a view, or a document) or a WebSphere resource, like an HTML file, a servlet, a Java Server Page (JSP) or an Enterprise JavaBean (EJB). To solve the problem of signing on multiple times, a new function called Single Sign-On (SSO) can be implemented in Lotus Domino as well as on the WebSphere Application Server.

This section contains information on configuring Single Sign-On (SSO) for Lotus Domino and the WebSphere Application Server.

### 8.9.1 SSO prerequisites

Prerequisites associated with SSO support for Domino servers and WebSphere Application Servers are as follows:

- ▶ All servers must be configured for the same DNS domain. For example, if the DNS domain is specified to be mycompany.com, then SSO will be effective with any Domino or WebSphere Application Server that serves the mycompany.com domain such as a.mycompany.com and b.mycompany.com.
- ▶ All servers must share the same user registry, accessible using LDAP. A Domino Directory (configured for LDAP access) or other LDAP directory can be used for the user registry.
- ▶ The LDAP directory product must be supported by WebSphere Application Server. This includes both Domino and all IBM SecureWay LDAP directory servers, which includes iSeries Directory Services.
- ▶ All users must be defined in a single LDAP directory. Connecting more than one directory using LDAP referrals is not supported. Using multiple Domino Directory Assistance documents to access multiple directories is also not supported.
- ▶ The users' Web browsers must have HTTP cookies enabled since the authentication information that is generated by the server is transported to the Web browser in a cookie. The cookie is then used to propagate the user's authentication information to other servers, relieving the user from entering the authentication information for every request to a different server.
- ▶ Domino Release 5.0.6a for AS/400 (or later) and Domino Release 5.0.5 (or later) for other platforms are supported.
- ▶ A Notes client Release 5.0.5 (or later) is required for configuration of the Domino server for SSO.
- ▶ Authentication can be shared across multiple Domino domains.
- ▶ WebSphere Application Server Version 3.5.1 (or later) for all platforms is supported.
- ▶ Any HTTP Web server supported by WebSphere Application Server.

**Restriction:** At the time this redbook was written, Domino did not support the IBM HTTP Server for iSeries (powered by Apache).

- ▶ Authentication can be shared across multiple WebSphere administrative domains.

- ▶ Both the WebSphere Application Server Standard and Advanced Editions for Version 3.5 as well as Single Server and Advanced Edition for Version 4.0 are supported. However, this chapter covers only WebSphere Application Server Advanced Edition 4.0. Since configuration is the same for both editions, a distinction between the editions will not be made in the information provided here. For more information on configuring Single Sign-On for WebSphere Application Server Version 3.5 and Domino, refer to *Domino and WebSphere Integration on the IBM sServer iSeries Server*, SG24-6223.
- ▶ Basic authentication (user ID and password) using either basic or custom challenge types is supported.
- ▶ Permissions for either all authenticated users or groups of users is supported.

To enable SSO between Domino servers and WebSphere Application Servers, you must first configure SSO for WebSphere and then configure SSO for Domino. The necessary configuration steps are covered in the following sections.

## 8.9.2 Setting up SSO for WebSphere Application Server

Enabling Single Sign-On (SSO) is only a very small part of the whole process to activate WebSphere authentication. In Section 8.6, “Enabling WebSphere Application Server authentication” on page 343 we stepped through how to enable security for the WebSphere Application Server using the OS/400 LDAP server or the Domino LDAP support.

To prepare to use the Single Sign-On (SSO) abilities for WebSphere and Lotus Domino for iSeries, you must update the WebSphere Global Security configuration again with Single Sign-On enabled, and re-generate and export the LTPA keys to be used when you configure Lotus Domino for iSeries for SSO.

To enable SSO in WebSphere perform the following steps:

1. Start the WebSphere administrative console.
2. On the administrative console, select **Security Center** from the console File menu.
3. Select the **General** tab if it is not already selected. On this window check the following, as shown in Figure 8-36:
  - a. The Enable Security check box should already be checked, as this was done in Section 8.6.2, “Configuring WebSphere Application Server security” on page 349.
  - b. Verify that the Security Cache Timeout field is set to a reasonable value for your application. When the timeout is reached, the WebSphere Application Server clears the security cache and rebuilds the security data. If the value is set too low, the extra processing overhead can be unacceptable. If the

value is set too high, you create a security risk by caching security data for a long period of time. The default value is 600 seconds.

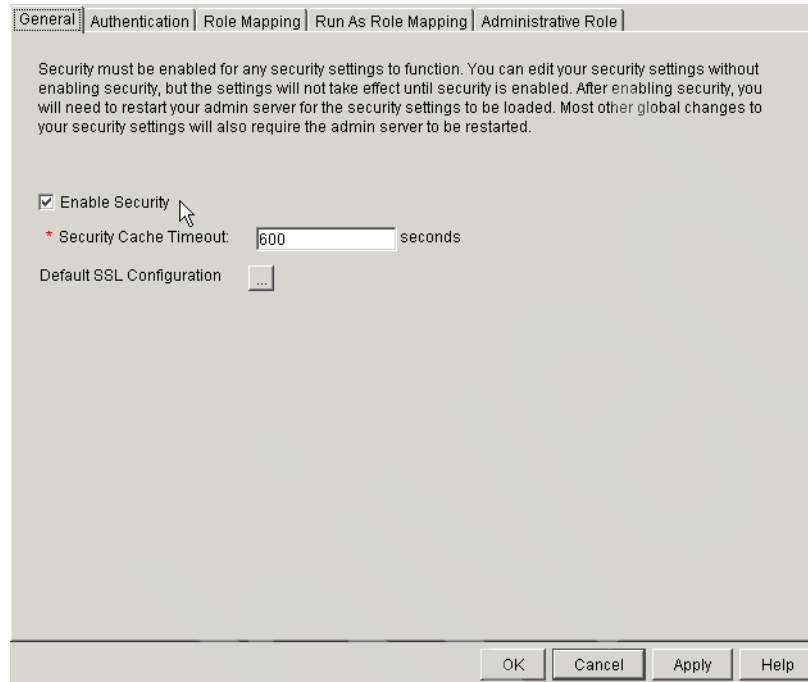


Figure 8-36 Enable WebSphere security

4. Click the **Authentication** tab. In this window perform the following:
  - a. Ensure that the Authentication Mechanism field is set to Lightweight Third Party Authentication (LTPA), to use an LDAP directory as the user registry. This should have been done in Section 8.6.3, "Protecting WebSphere resources" on page 354.
  - b. Check the Enable Single Sign On (SSO) box to enable SSO and allow authentication information to be placed in HTTP cookies.
  - c. Set the Domain field to the domain portion of your fully-qualified DNS name for the system running your WebSphere Application Server administrative domain. In our scenario, the system's host name is RALYAS4B.ISERIES.ITSO.RAL.IBM.COM, so the domain to enter would be ISERIES.ITSO.RAL.IBM.COM.

- d. You also need to configure the LTPA keys to be used by the administrative domain that you are configuring. Perform the following steps:
    - i. Click **Generate Keys...** to generate keys for LTPA.
    - ii. When prompted, type the LTPA password to be associated with these LTPA keys, in our scenario we used **my5ldap**. Click **OK** to save the LTPA keys. You must use this password when importing these keys into Domino.
5. The LDAP section was previously configured in the Section 8.6.3, “Protecting WebSphere resources” on page 354. Check to ensure these values are correct. See Figure 8-37 on page 387.

**Security server ID**

The user ID of the administrator for the WebSphere administrative domain. Use the short name or user ID for a user already defined in the LDAP directory. Do not specify a distinguished name by using `cn=` or `uid=` before the value. This field is not case-sensitive. When you start the WebSphere Application Server administrative console, you are prompted to log in with an administrative account. You must enter the exact same value that you specify in this field.

**Security server password**

The password corresponding to the Security Server ID field. This field is case-sensitive.

**Directory type**

The type of LDAP server you are using. For example, you can select SecureWay for IBM SecureWay LDAP directory or Domino 5.0 for Domino Release 5.05 (or later) from the list.

**Host**

The fully qualified DNS name of the system on which the LDAP directory runs, for example RALYAS4B.ISERIES.ITSO.RAL.IBM.COM.

**Port**

The port on which the LDAP directory server listens. By default, an LDAP directory server using an unsecured connection listens on port 389. If your server meets this description, you can leave this field blank.

**Base distinguished name**

The distinguished name (DN) of the directory in which searches begin within the LDAP directory. In this scenario, users are stored in the directory tree under `ou=people,o=iserieshop`. This is the DN you have to enter for this parameter. This field is

not case-sensitive. This field is required for all LDAP directories except the Domino Directory. If you are using the Domino Directory and you specify a base distinguished name, you will not be able to grant permissions to individual Web users for resources managed by your WebSphere Application Server.

**Bind distinguished name**

The DN of the user who is capable of performing searches on the directory. In most cases, this field is not required; typically, all users are authorized to search an LDAP directory. However, if the LDAP directory contents are restricted to certain users, you need to specify the DN of an authorized user, for example, an administrator, cn=administrator.

**Bind password**

The password corresponding to the Bind Distinguished Name field. This value is required only if you specified a value for the Bind Distinguished Name field. This field is case-sensitive.



The screenshot shows the 'Authentication' tab in the WebSphere Administrative Console. The 'Authentication Mechanism' is set to 'Lightweight Third Party Authentication (LTPA)'. Under 'LTPA Settings', 'Token Expiration' is 120 minutes, 'Enable Single Sign On (SSO)' is checked, and the 'Domain' is 'ISERIES.ITSO.RALIBM.COM'. There are buttons for 'Generate Keys...', 'Import Key...', and 'Export Key...'. Below this, 'LDAP' is selected as the 'Custom User Registry'. Under 'LDAP Settings', 'Security Server ID' is 'WENDY', 'Security Server Password' is masked with asterisks, 'Host' is 'RALYAS4B.ISERIES', 'Directory Type' is 'SecureWay', 'Port' is empty, 'Base Distinguished Name' is 'ou=people,o=iserie', and 'Bind Distinguished Name' and 'Bind Password' are also empty. There are buttons for 'Advanced...' and 'SSL Configuration'. At the bottom are 'OK', 'Cancel', 'Apply', and 'Help' buttons.

Figure 8-37 Enabling Single Sign-On

6. Click **OK** to save the security settings.
7. Click **OK** to acknowledge the information dialog box that warns that changes do not take effect until the administrative server is restarted.

You now need to stop and restart the administrative server. Whenever changes are made to the global security settings, the WebSphere Application Server administrative server must be stopped and restarted for the changes to take effect.

8. On the administrative console, expand the **WebSphere Administrative Domain** and then **Nodes**.
9. Right-click the node representing your administrative server from the drop-down menu and select **Restart**.

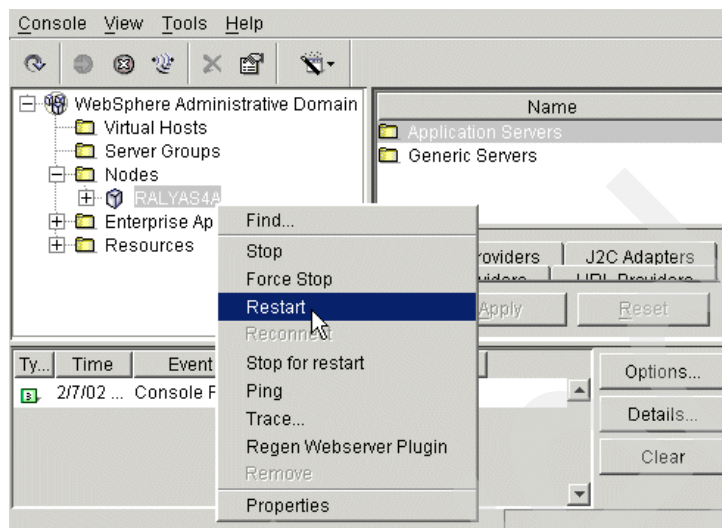


Figure 8-38 Restart WebSphere to enable security changes

10. Click **Yes** on the confirmation dialog box.
11. Monitor the administrative server task (or job) to ensure that the server stops and restarts. Monitor the server task (or job) to determine when the server is running. As you watch the server job, notice that it starts, stops, and then starts again. This is normal behavior after global security settings have been changed.
12. Start the administrative console. Specify the user ID and password by using the exact same values that you specified for the Security Server ID and Security Server Password fields in the Global Security Settings.

To complete the security configuration for SSO, you need to export the LTPA keys to a file. Do this for just one WebSphere administrative server if you are configuring SSO for use with multiple WebSphere Application Server administrative domains. This file is subsequently used during the configuration of additional administrative domains and during the configuration of SSO for Domino.

13. On the administrative console, select **Security Center** from the console's File menu.
14. Select the **Authentication** tab.
15. Click **Export Keys...** to export the LTPA keys to a file.
16. When prompted, specify the name and location of the file to contain the LTPA keys. You can use any file name and extension.

**Attention:** Make note of the name and extension you specify; you have to use this file when you configure SSO for Domino.

17. Click **Save** to save the file.
18. Click **Cancel** to close the security console. (This procedure has not changed any global security setting, so there are no new settings to save and no server to restart.)

### 8.9.3 Setting up SSO for Domino

Configuring SSO for Domino is accomplished by selecting a new Multi-server option in the Domino server document for session-based authentication, along with creating a new domain-wide configuration document in the Domino Directory called the Web SSO Configuration document. The Web SSO Configuration document, which should be replicated to all Domino servers participating in the SSO domain, is encrypted for participating Domino servers and contains a shared secret used by Domino servers for authenticating user credentials.

#### Creating the Domino Web SSO Configuration document

To create the Domino Web SSO Configuration document, use a Lotus Domino Administrator client Release 5.0.5 (or later) and perform the following steps:

1. From the Domino Administrator client, open a connection to your Domino server (DOMWEBA), select the **Configuration** tab under Server, then select the **All Server Documents** view.
2. Click the **Web...** action button and select **Create Web SSO Configuration** to create the document, as shown in Figure 8-39 on page 390.

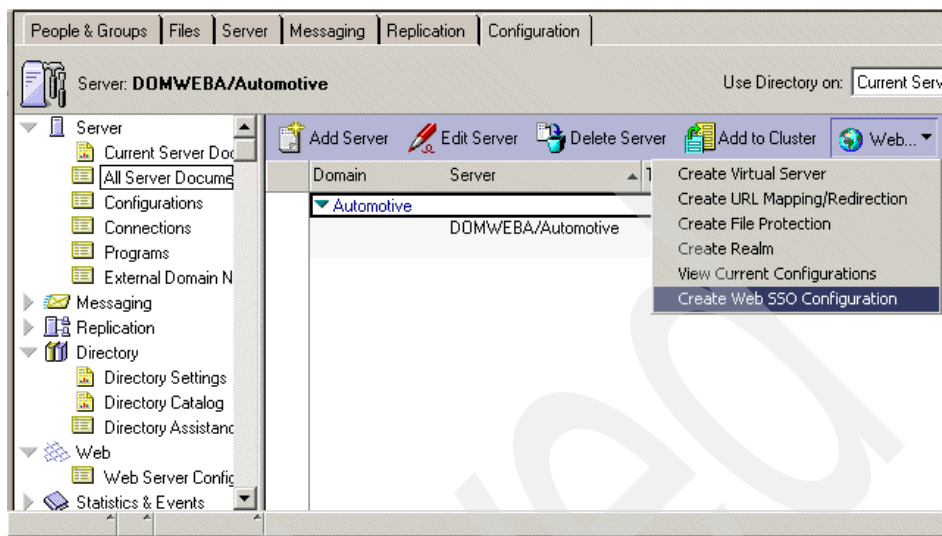


Figure 8-39 Create Web SSO Configuration

3. In the Web SSO Configuration document, click the **Keys...** action button and select **Import WebSphere LTPA Keys** to import the LTPA keys from a file, as shown in Figure 8-40.

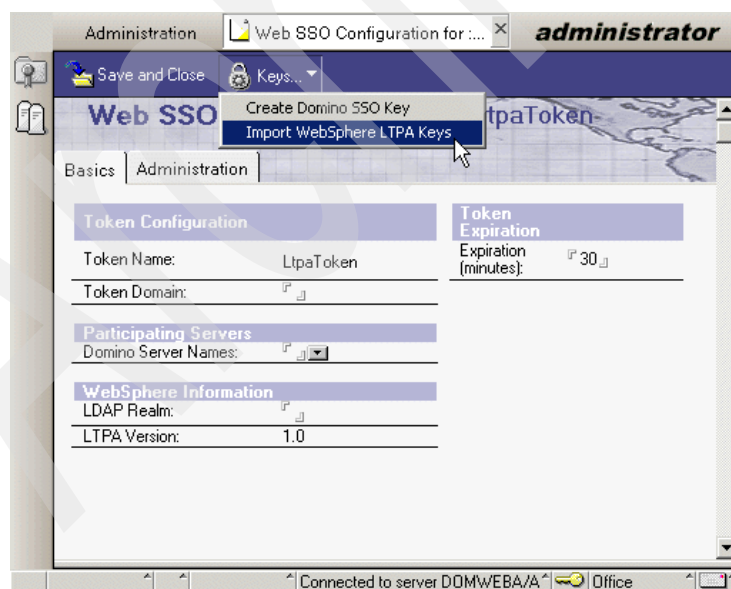


Figure 8-40 Importing WebSphere LTPA keys into Domino

4. Enter the path to the LTPA keys file that was exported earlier in step 15 on page 388 and click **OK** to import the file.
5. When prompted, enter the password that you used earlier when generating the LTPA keys and click **OK** to continue the key import.
6. Click **OK** on the Successfully imported WebSphere LTPA keys message.
7. The Web SSO Configuration document should automatically be updated to reflect the information from the LTPA keys file you just imported.
8. Verify and, if necessary, update the fields in the Web SSO Configuration document as follows:

**Token domain**

This is the DNS domain portion of the fully-qualified Internet name of your system. Because all servers participating in SSO must be in the same DNS domain, this value must be the same as the domain value specified when configuring the WebSphere Application Server. In our example the domain was ISERIES.ITSO.RAL.IBM.COM.

**Domino server names**

These are the Domino servers that will be participating in SSO. This document will be encrypted for the creator of the document, the members of the Owners and Administrators fields, and the Domino servers specified in this field. You must specify a fully-qualified Domino server name here (for example, MyDominoServer/MyOu). The Domino server name that you specify here must also match the name of the home/mail server currently in the active Location document on your Lotus Notes client. In our scenario the Domino server is DOMWEBA/Automotive.

**LDAP realm**

The fully-qualified host name of the LDAP server. This field is initialized from the information provided in the LTPA keys file. You only need to change this value if an LDAP server port value was specified for the WebSphere administrative domain. If a port is specified, a backslash (\) must be inserted in the value before the colon. For example, replace mymachine.mydomain.ibm.com:389 with mymachine.mydomain.ibm.com \:389. The port was not changed in our example, the LDAP Realm is therefore RALYAS4B.ISERIES.ITSO.RAL.IBM.COM.

**Token expiration**

This is the number of minutes a token can exist before expiring. A token does not expire based on

inactivity. Rather, it is valid for only the number of minutes specified from the time of issue. In this scenario we left this field as the default of 30 minutes.

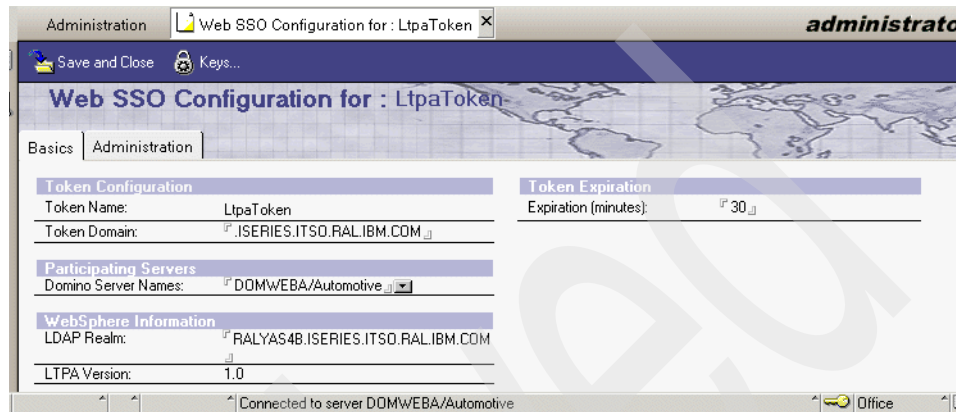


Figure 8-41 Web SSO Configuration document

9. Click **Save and Close** to close the Web SSO Configuration document. It now appears in the Web Configurations view.

## Configuring the Domino server document

This is the last step required is to update the Domino server document for SSO. To do this perform the following steps:

1. In the Domino Administrator, click the **Configuration** tab and then click **Current Server Document** under Servers.
2. Click **Edit Server** to edit the Domino server document, as shown in Figure 8-42 on page 393.

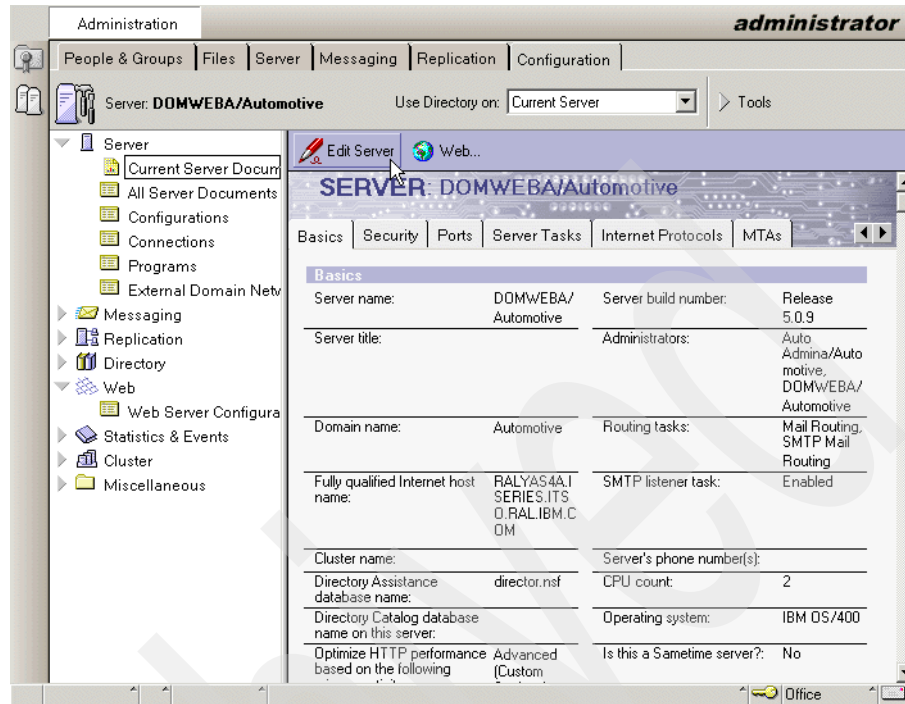


Figure 8-42 Edit Domino server information for SSO

3. Select the **Ports** tab, then the **Internet Ports** sub-tab, and then select the **Web** sub-sub-tab, as shown in Figure 8-43 on page 394.

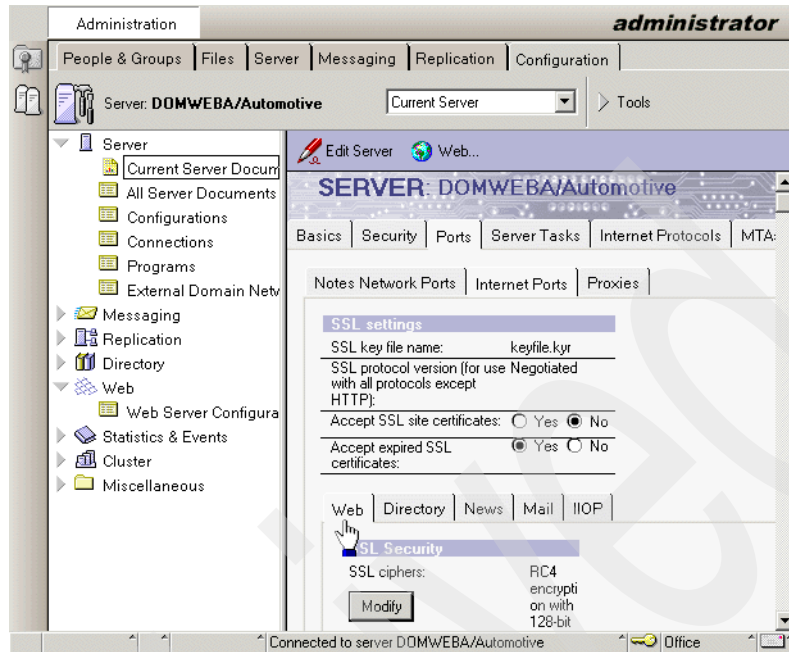


Figure 8-43 Ports, Internet Ports, Web tabs

4. Verify that the Name & password field in the TCP/IP Authentication options section is still set to **Yes**. This enables basic authentication for Web users.



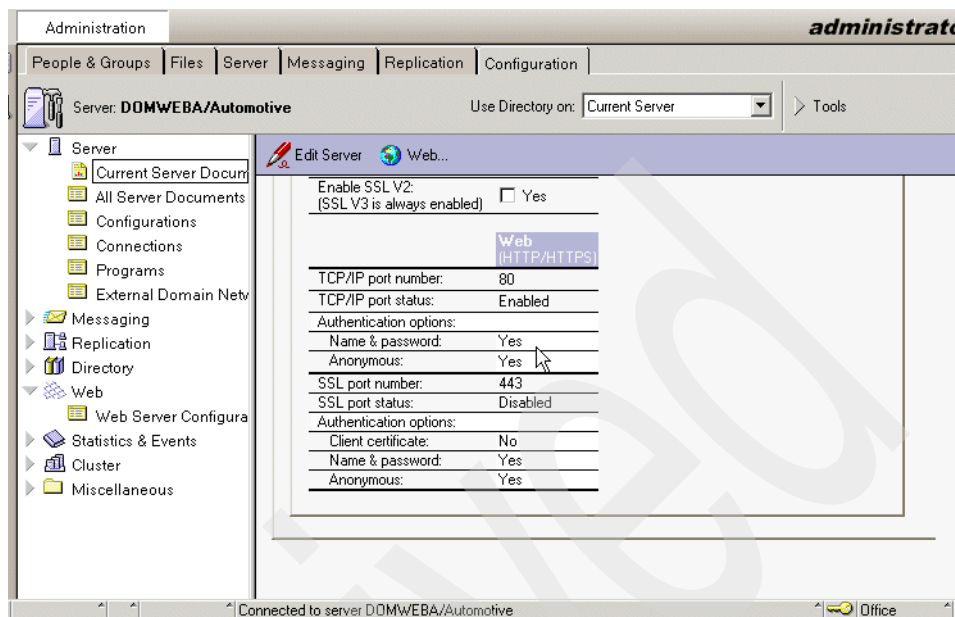


Figure 8-44 Enabling basic authentication for Web users

5. Select the **Internet Protocols** tab, and then select the **Domino Web Engine** sub-tab.
6. In the **HTTP Sessions** section, click the drop-down menu in the Session authentication field and select **Multi-server**. This enables SSO for Domino, as shown in Figure 8-45 on page 396.

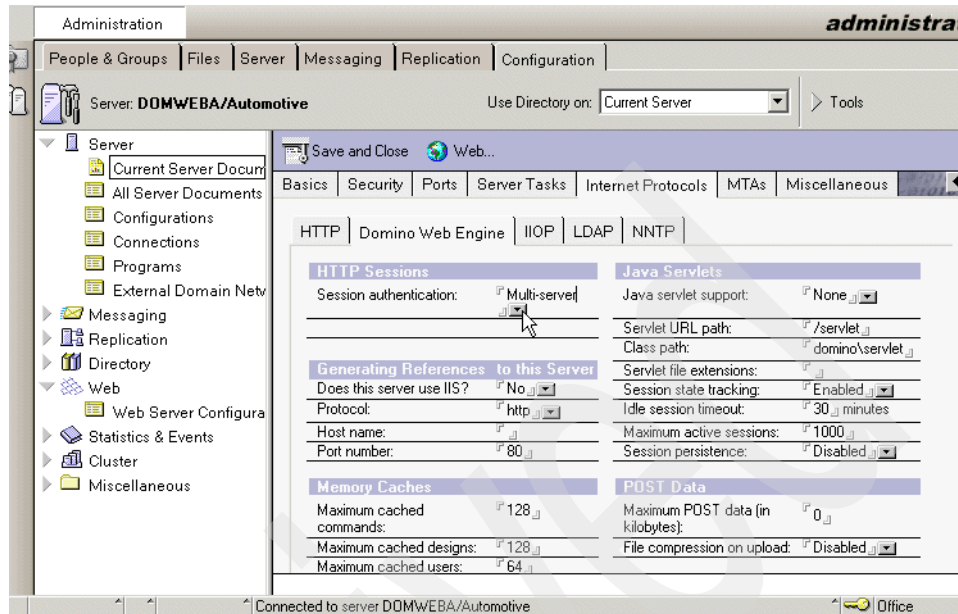


Figure 8-45 Multi-server enables SSO for Domino

7. Click **Save and Close** to close the Domino server document.
8. For these changes to take effect, the Domino server must be stopped and restarted. To do this, enter the following command from a 5250 command prompt and press Enter:  

```
WRKDOMSVR DOMWEBA
```

Where DOMWEBA is your Domino server name.
9. From the Work with Domino Servers screen, enter option 6 (End server) next to your Domino server.
10. Refresh the screen to verify that the Domino server is ended by pressing F5 until the status is \*ENDED.
11. Enter option 1 (Start server) next to your Domino server to start it again.

**Note:** If you are running the Domino HTTP sever, you will see the following message when the HTTP server task starts:

HTTP:Successfully loaded Web SSO configuration

If a Domino server enabled for SSO cannot find a Web SSO Configuration document or is not included in the Domino Server Names field (and thus cannot decrypt the document), then the following message should appear on your Domino server's console:

HTTP:Error Loading Web SSO configuration.Reverting to single-server session authentication.

## 8.9.4 Verifying Single Sign-On

At this point you are ready to verify that SSO for WebSphere and Domino is configured and working correctly. In this section we provide examples of using SSO between Domino and WebSphere Application Servers. Before proceeding with your verification of SSO between WebSphere and Domino, you should have already verified the following:

- ▶ The LDAP directory contains at least one user defined for testing purposes.
- ▶ The WebSphere administrative console can be started for the WebSphere administrative domains that you are using. You should be authenticated to the administrative domain using the security name associated with a user defined in the LDAP directory.
- ▶ At least one user in the LDAP directory must be authorized to access at least one Domino resource, such as the Domino Directory. In this case, the user wendy is authorized to access the Domino Web application.
- ▶ At least one user in the LDAP directory must be authorized to access at least one WebSphere Application Server resource, such as the SimpleServlet servlet as used in the scenario application.
- ▶ From a Web browser that is configured to not accept HTTP cookies, you should be able to access resources, such as a servlet, that are protected by each of the WebSphere Application Servers. You should be prompted for a user ID and password.
- ▶ You should be able to access a resource, such as a Domino database, that is protected by each of the Domino servers. You should be prompted for a user ID and password.

After performing the verifications listed above, you are now ready to verify that SSO is working correctly.

## SimpleServlet to Domino application SSO test

We performed the following tasks to test the WebSphere application (SimpleServlet) to Domino application SSO:

1. Configure the Web browser to accept HTTP cookies. If you are using Internet Explorer, the type of cookies that need to be enabled are per-session (not stored).
2. For testing purposes only, configure the Web browser to prompt before accepting HTTP cookies. This will provide you with visual confirmation that the Domino and WebSphere applications are generating and returning HTTP cookies to your Web browser after you authenticate.
3. From the Web browser, specify the URL for a resource protected by the Domino server. For example, open a database that defines no access for an anonymous user. This will verify that the token generated by the Domino server is accepted by WebSphere Application Servers. Make sure to enter the fully-qualified TCP/IP host name for the URL. For example, enter `http://myhost.mycompany.com/names.nsf` and not just `http://myhost/names.nsf`. In our example the URL would be:  
  
<http://ralyas4a.iseries.itso.ral.ibm.com/domwaslab.nsf/loanapp?openform>
4. When prompted for a user ID and password, make sure that you specify a user ID that is authorized to resources for both the Domino and WebSphere Application Servers. The format of the name that you specify depends on the level of restriction Domino is using for Web users and whether a Domino Directory or another LDAP directory is being used. Refer to the Controlling the level of authentication for Web clients document in Domino 5 Administrative Help for details on the options for specifying a username for basic authentication. The level of restriction Domino uses for Web users is set in the Web server authentication field found on the Security tab of the Server document. If you are using the default configuration settings, you should specify the user's shortname or user ID. Accept the HTTP cookie when prompted. You should now have access to the resource. In our example we gave Wendy access to the Domino database. We used the user ID and we are using the iSeries LDAP server. So we could sign on with user ID of wendy and the password of my51dap, as shown in Figure 8-46 on page 399.

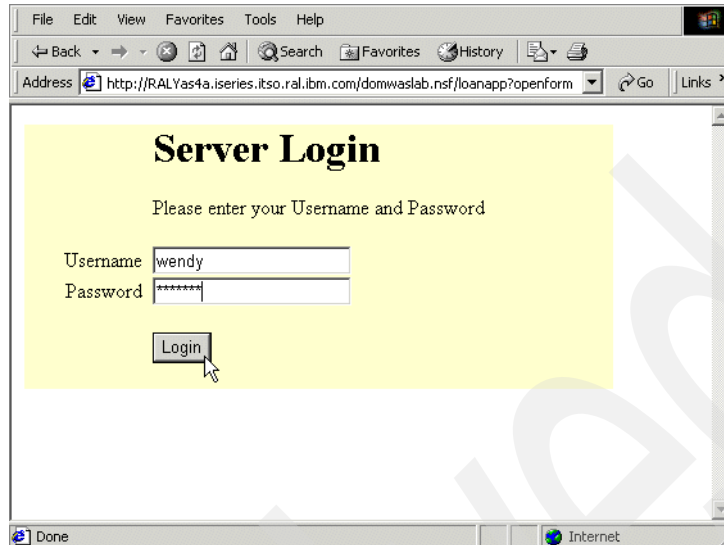


Figure 8-46 Sign on to Domino application

5. You are prompted to accept a cookie that is generated as a result of SSO. Click **OK**.
6. From the same Web browser session, access a resource protected by a WebSphere Application Server. You should have access to the resource without being prompted for a user ID and password. Make sure to enter the fully-qualified TCP/IP host name for the URL. For example, enter `http://myhost.mycompany.com/webapp/examples/showCfg` and not `http://myhost/webapp/examples/showCfg`. In our example from the Domino application we can click on the **Submit** button to go to the WebSphere application, as shown in Figure 8-47 on page 400.

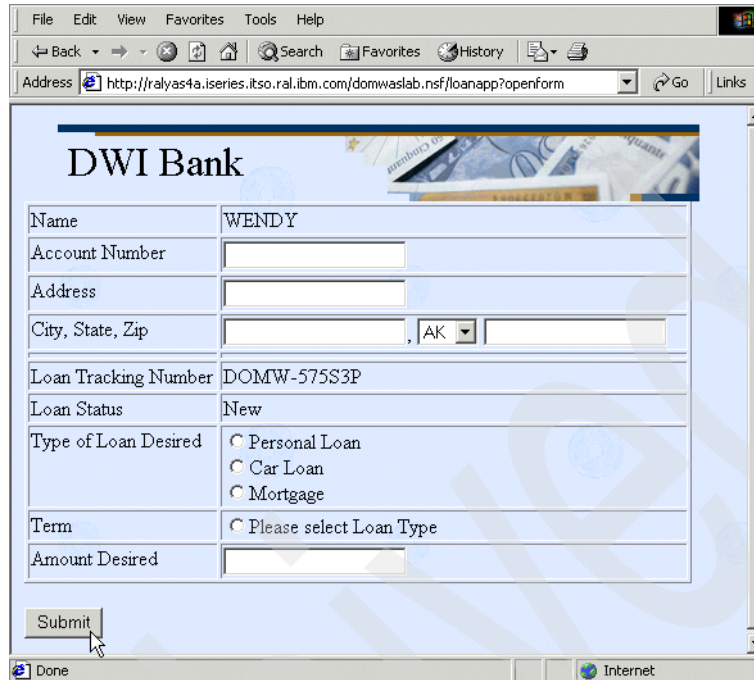


Figure 8-47 Domino application

7. From the second Domino application window, click **Return to Main Menu** to access the WebSphere SimpleServlet, as seen in Figure 8-48.

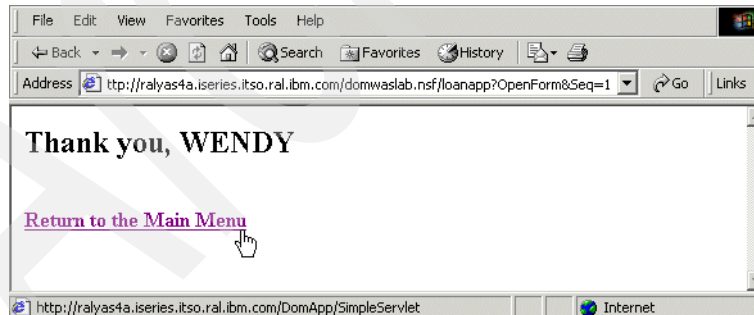


Figure 8-48 Link to WebSphere SimpleServlet

8. You are now prompted to accept a second cookie. This is the original cookie that appeared before enabling SSO. Click **OK**.

You should now see SimpleServlet, as shown in Figure 8-49. You should not be prompted to log into the WebSphere application.

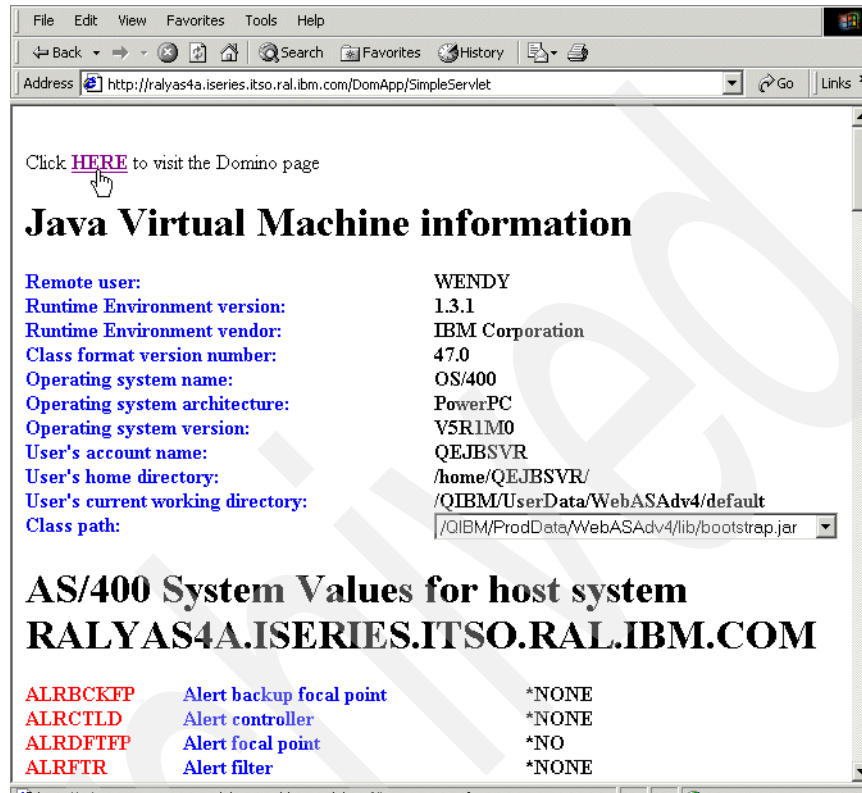


Figure 8-49 WebSphere SimpleServlet

- Restart your Web browser session and perform the SSO verification steps mentioned above, but this time access a resource protected by a WebSphere Application Server first. In our example this would be:

<http://ralyas4a.ISERIES.ITSO.RAL.IBM.COM:80/DomApp/SimpleServlet>

When the WebSphere application of SimpleServlet is displayed, click **HERE** to access the Domino application.

This will verify that the token generated by the WebSphere Application Server is accepted by the Domino server(s). When prompted for a user ID and password, use the user's short name or user ID since this is the default WebSphere Application Server naming convention for users.

**Attention:** The Domino application database and the servlet used in this chapter to demonstrate the Single Sign-On process are available for download. Refer to Appendix F, “Additional material” on page 557 for further information on how to obtain these objects.

## 8.10 Enabling SSL with SSO

In the previous sections we have dealt with non-SSL security settings for Single Sign-On. In this section we show you how to enable SSL for both WebSphere and Domino, while still using SSO.

All previous sections of this chapter need to be completed before trying to configure SSL.

For information on how to enable SSL for the iSeries LDAP Directory Services refer to Section 4.9.4, “Enabling SSL for the LDAP server” on page 128.

### 8.10.1 Enabling WebSphere SSL with SSO

You must first enable SSL in WebSphere, as you need to regenerate the LTPA keys and export them for Domino to import.

To enable SSL do the following steps:

1. Start the WebSphere administrative console.
2. On the administrative console, select **Security Center** from the console File menu.
3. Click the **Authentication** tab. In this window perform the following:
  - a. Check **Limit to SSL connections only**. This permits you to use a connection with SSL for Single Sign-On (SSO).

**Important:** The LTPA keys need to be re-generated but first you must turn on SSL. Complete the remaining steps before re-generating the keys.

- b. In the LDAP settings section specify the port on which the LDAP directory server listens. By default it is 389, which is using an unsecured connection. This needs to be changed to 636 the well-known port for secure LDAP connections.

See Figure 8-50 on page 403 for our example.



General | **Authentication** | Role Mapping | Run As Role Mapping | Administrative Role

Authentication Mechanism: ☐ Local Operating System  
☒ Lightweight Third Party Authentication (LTPA)

LTPA Settings

\* Token Expiration: 120 minutes

☒ Enable Single Sign On (SSO)

\* Domain: ISERIES.ITSO.RALIBM.COM

☒ Limit to SSL connections only

☐ Enable Web trust association

Generate Keys... Import Key... Export Key...

☒ LDAP ☐ Custom User Registry

LDAP Settings

\* Security Server ID: WENDY Port: 636

\* Security Server Password: \*\*\*\*\* Base Distinguished Name: ppl,o=iserieshop

\* Host: ITSO.RALIBM.COM Bind Distinguished Name:

Directory Type: SecureWay Bind Password:

Advanced... SSL Configuration

OK Cancel Apply Help

Figure 8-50 Changing authentication to enable SSL

- c. Click **SSL Configuration**.
- d. In the pop-up window under the **General** tab, check **Enable SSL**, then click **OK**. See Figure 8-51 on page 404.

**Note:** In our example we used the sample Key file name of:

/QIBM/UserData/WebASAdv4/default/etc/DummyServerKeyFile.jks

And Trusted file name of:

/QIBM/UserData/WebASAdv4/default/etc/DummyServerTrustFile.jks

It is strongly advised that prior to enabling WebSphere Global Security in a production environment, that the demo keyring is replaced with either a new self-signed certificate key pair or a certificate signed by a third-party Certificate Authority (CA). The latter CA option, while valid, is perhaps not of any extra benefit as the trust association offered by the CA is of minimal concern to WebSphere internally. See the *IBM WebSphere V4.0 Advanced Edition Security*, SG24-6520, for more details on how to replace the demo keyring.

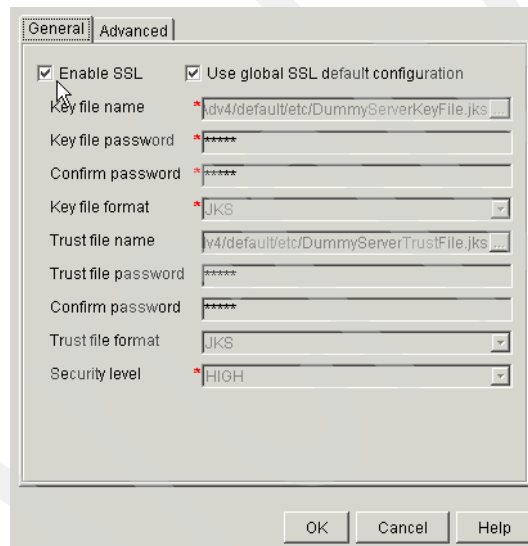


Figure 8-51 Enable SSL for WebSphere

- e. Click **OK** to save the settings.
- f. You now need to regenerate the LTPA keys to be used by the administrative domain that you are configuring. Perform the following steps:
  - i. Click **Generate Keys...** to generate keys for LTPA.

- ii. When prompted, type the LTPA password to be associated with these LTPA keys, in our scenario we used my5ldap. Click **OK** to save the LTPA keys. You must use this password when importing these keys into Domino.
4. Click **OK** to save the security settings.
5. Click **OK** to acknowledge the information dialog box that warns that changes do not take effect until the administrative server is restarted.

You now need to stop and restart the administrative server. Whenever changes are made to the global security settings, the WebSphere Application Server administrative server must be stopped and restarted for the changes to take effect.

6. On the administrative console, expand the **WebSphere Administrative Domain** and then **Nodes**.
7. Right-click the node representing your administrative server from the drop-down menu and select **Restart**.
8. Click **Yes** on the confirmation dialog box.
9. Monitor the administrative server task (or job) to ensure that the server stops and restarts. Monitor the server task (or job) to determine when the server is running. As you watch the server job, notice that it starts, stops, and then starts again. This is normal behavior after global security settings have been changed.
10. Start the administrative console. Specify the user ID and password by using exactly the same values that you specified for the Security Server ID and Security Server Password fields in the Global Security Settings.

To complete the security configuration for SSO, you need to export the LTPA keys to a file. This file is subsequently used during the configuration of additional administrative domains and during the configuration of SSO for Domino.

11. On the administrative console, select **Security Center** from the console's File menu.
12. Select the **Authentication** tab.
13. Click **Export Keys...** to export the LTPA keys to a file.
14. When prompted, specify the name and location of the file to contain the LTPA keys. You can use any file name and extension.

**Attention:** Make note of the name and extension you specify; you must use this file when you configure SSO for Domino.

15. Click **Save** to save the file.

16. Click **Cancel** to close the security console. (This procedure has not changed any global security setting, so there are no new settings to save and no server to restart.)

SSL has now been enabled for WebSphere.

## 8.10.2 Enabling Domino SSL with SSO

Setting up Domino to use SSL is the step that enables the server and client access to the Domino server using a secure connection to ensure privacy and authentication via the network.

You can set up SSL on a protocol-to-protocol basis, for example, not only for mail protocols like SMTP and/or POP3, but also for HTTP and LDAP. This chapter will show the necessary steps to set up the Domino server to use SSL for HTTP and LDAP.

This chapter will not cover the basics of SSL, like the Certificate Authority (CA) or the contents of a digital certificate or a keyring file. It will be assumed that the reader is familiar with the security concept of SSL.

Before starting this section make sure you have successfully completed Chapter 8.10.1, “Enabling WebSphere SSL with SSO” on page 402, as you need the LTPA keys that were generated and exported from WebSphere in the configuration of SSL in Domino.

### The server Certificate Administration database

The Domino server Certificate Admin database is the key point for managing and using certificates within the Domino server. The server Certificate Admin application running in this database will enable you to do the following:

- ▶ Create the server certificate and keyring file, which holds your server certificate.
- ▶ Send a request to a CA to sign the server certificate.
- ▶ View requests that you submitted to CAs.
- ▶ Add a CA's certificate as a trusted root to the server certificate.
- ▶ View information about certificates in the keyring file.
- ▶ Control client access to the server by adding or removing trusted root certificates from the keyring file.
- ▶ Create a self-certified certificate for testing purposes.

There are two possibilities for the source of the digital certificate you are using in your environment:

- ▶ The certificate is generated by an internal Certificate Authority (CA) within your enterprise, such as the iSeries Digital Certificate Manager or the Domino CA.
- ▶ The certificate is obtained from a third-party CA like VeriSign.

The decision of which CA you are use depends on the purpose and where you are going to use SSL, either only for internal communication or also for external users to connect and communicate with your enterprise.

No matter which CA you are going to use, the server Certificate Admin database hosts the application that is responsible for the certificate management in your Domino environment.

For the initial setup of the server Certificate Admin application you have to perform the following steps:

- ▶ Check the server Certificate Administration application database.  
Domino automatically creates the database during server setup. The file name of the database is CERTSRV.NSF.  
If the database is not available after you start the Domino server, create the database by using the server Certificate Admin template (CSRV50.NTF). You will find the template on your Domino server by selecting the **Show advanced templates** button.
- ▶ Set up the Domino server as a Domino Web server.  
We assume that the Domino server has already been set up as a Web server. For a detailed description of how to set up the Domino server as a Web server, please refer to *Lotus Domino for AS/400 R5: Implementation*, SG24-5592, Chapter 9.

### Create a server keyring file

Before you request a certificate from a CA, you must create a keyring file to store the certificates. A keyring file is a binary file that is password-protected and stored in the server's data directory. When you create a server keyring file, Domino generates an unsigned server certificate and automatically includes several trusted root certificates. The unsigned server certificate is not valid until the CA signs it.

Domino also creates a stash file (.STH) using the same name as the keyring file, but with the file extension .STH. Domino uses the stash file to store the keyring file password for unattended access to the server keyring file.

To create the keyring file, please perform the following steps:

1. Start your Nnotes client and log in as the administrator.
2. Open the server Certificate Administration database by clicking **File -> Database -> Open** and select your Domino server.
3. Select **Server Certificate Admin** from the list of available databases and press Enter. This will open the main window for the server Certificate Administration database (see Figure 8-52).

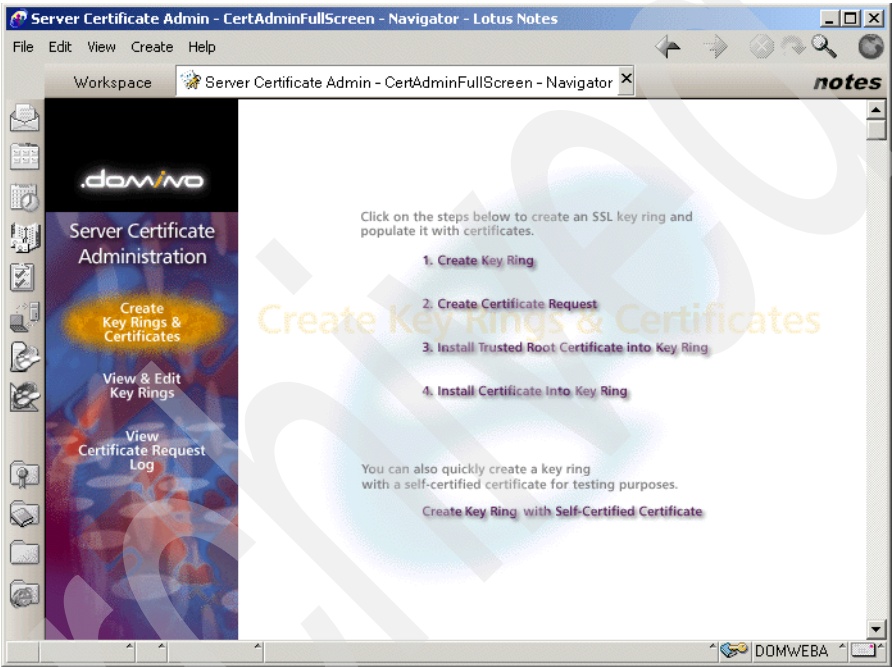


Figure 8-52 Server Certificate Administration database

4. Click on **1. Create Keyring**. You will see a new create keyring document. Complete the following fields as shown in Table 8-2.

Table 8-2 Input fields for server Keyring document

Field	Enter
Keyring File Name	The default is keyfile.kyr. You can choose your own file name, but please keep in mind to use the extension kyr and that the value matches the name entered later in your Domino server Document.
keyring Password	A password for the Keyring file.

Field	Enter
Confirm Password	The password you specify in the Keyring Password field.
Key Size	The size of public/private key pairs in bits, either 512 or 1024. The larger the key size, the greater the key strength.
Common Name	The server's common name is the TCP/IP fully-qualified domain name, in our example domweba.iseries.itso.ral.ibm.com.
Organization	The name of the organization where the certificate owner works. In our scenario Automotive.
Organizational Unit	(Optional) The division or department where the certificate owner works.
City or Locality	(Optional) The city or town where the certificate owner lives.
State or Province	Three or more characters that represent the state or province where the certificate owner lives, such as North Carolina.
Country	A two-character representation of the country where the certificate owner lives -- for example, US for United States.

See Figure 8-53 on page 410 for the values used in our scenario to create the Keyring document.

Workspace Server Certificate Admin - CertAdminFullScreen - Navigator Create Key Ring

### Create Key Ring

The first step in setting up SSL on a server is to create the key ring.  
When the key ring is created, a public/private keypair is automatically generated and stored in the key ring.

Key Ring Information	Quick Help
Key Ring File Name: <input type="text" value="keyfile.kyr"/> Key Ring Password: <input type="password" value=""/> Confirm Password: <input type="password" value=""/>	Specify the name and password for the key ring file.  <b>Note:</b> You'll be referring to the key ring information you enter here in subsequent steps as you create and install certificates into the key ring.
Key Size	
Key Size: <input type="text" value="512"/>	Key Size is the size of the public/private key pair in bits. The larger the key size, the greater the encryption strength.  <b>Note:</b> This Edition of Domino provides the ability to generate RSA keys at both 1024 bits and 512 bits, in accordance with export regulations worldwide.
Distinguished Name	
Common Name: <input type="text" value="domweba1series.itso.ral.ibm.com"/> Organization: <input type="text" value="Automotive"/> Organizational Unit: <input type="text" value=""/> City or Locality: <input type="text" value=""/> State or Province: <input type="text" value="North Carolina"/> Country: <input type="text" value="US"/>	The Distinguished Name is the information about your site that will appear in any certificates you create.  <b>Note:</b> Make sure the Common Name matches the URL of your site. Some browsers check the Common Name and the site URL, and do not allow a connection if they don't match.

Create Key Ring

Figure 8-53 Create Keyring document

5. Click **Create Key Ring** for the creation of the keyring file. You will see a confirmation message like Figure 8-54 on page 411.



**Note:** The keyring file and the stash (.STH) file are created in the Notes data directory of your client machine used to create the keyring.

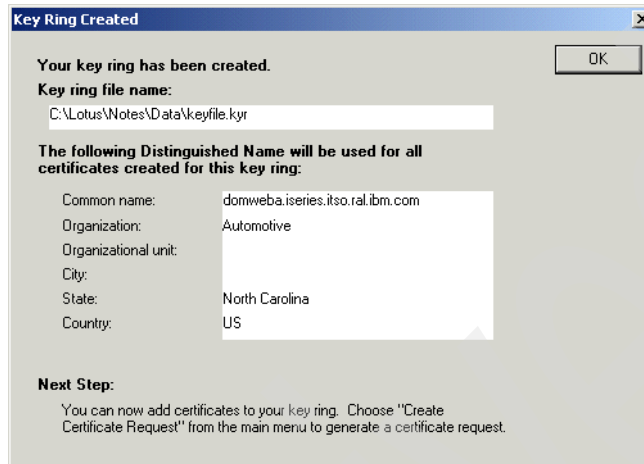


Figure 8-54 Server Keyring creation message

6. Click **OK** to end the initial steps of the server Certificate Administration database set up.

### Transferring the keyring file to your iSeries Domino server

The Keyring file that you have created is still in your Notes client data directory. Before you can use it on your iSeries Domino server, you have to transfer this file and its stash file to your Domino data directory by using a copy command on a mapped OS/400 drive or FTP.

**Important:** You have to transfer both keyfile.kyr and keyfile.sth files. The SSL will not start if you do not include the keyfile.sth file. If you FTP the files, make sure that you set the transfer type as binary and change the owner to QNOTES. Make sure the QNOTES user profile has read (\*R) authority after you copy them.

You must ensure that the keyring password in the stash file is protected. The keyring file password is altered in the stash file so that it cannot be recognized by a casual observer, but it is not encrypted. You should not allow unauthorized persons access to either the stash file or the keyring file. In the normal course of operation, only the server itself should have access to those files; however, administrators may also need permission to remove or replace the files. As with all Web server resources, managing proper file permissions and protections is vital to the security of the system.

## Request an SSL server certificate

Once you set up the server Certificate Administration database and created the keyring file, you are now ready to request a server certificate from a third party CA, like VeriSign.

**Note:** When you request an SSL server certificate, you use standard Public-Key Cryptography Standards (PKCS) format. This is an industry standard format that many CA's, including Domino, understand. Before you request a certificate from a third-party CA, make sure the CA uses the PKCS format, not some other format, such as Privacy-Enhanced Mail (PEM). If you are unsure of the format required by a third-party CA, check with the CA.

To request a server certificate from a third party CA, perform the following steps:

1. From the server Certificate Administration main menu (Figure 8-52 on page 408) click **2. Create Certificate Request**.
2. In the first field specify the full path of the keyring file that you have created during the setup of the server certificate administration (see "Create a server keyring file" on page 407).

In our example this would be J:\domweba\data\keyfile.kyr.

3. Select **Yes** if you want to log the request in the server Certificate Admin application.
4. For the methods you have two possibilities:
  - Select **Send to CA by e-mail** if your CA supports the PKCS format sent via e-mail and enter the CA's e-mail address, your e-mail address, phone number, and location.

- Select **Paste into form on CA's site** if your CA does not support the PKCS format requests sent via e-mail.

As we are requesting a certificate from VeriSign, we have to select the **Paste into form on CA's site** method, as VeriSign does not use PKCS format for requests sent by e-mail.

5. Click **Create Certificate Request**.
6. Type in the password for your keyring file and click **OK**.
7. The Certificate Request Created confirmation window (see Figure 8-55) appears.

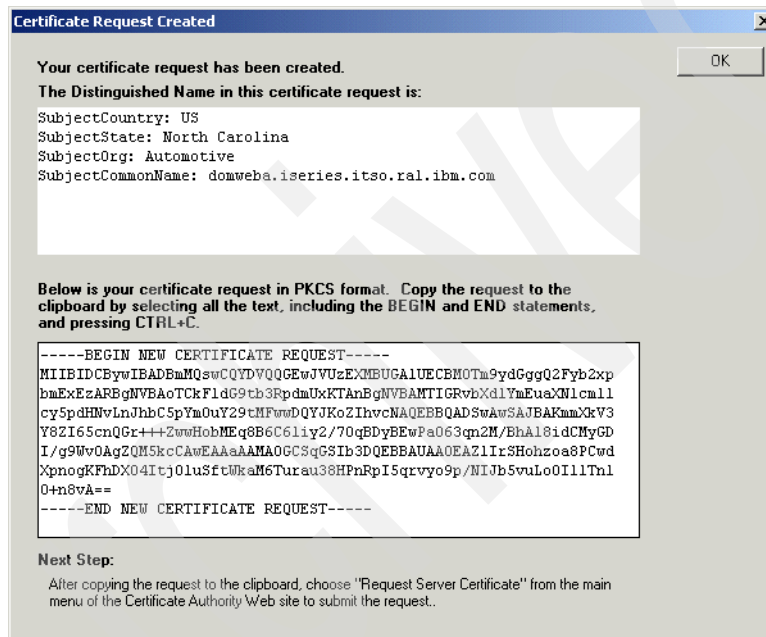


Figure 8-55 Certificate Request Created confirmation

You should now copy the server certificate to the clipboard of your workstation. Make sure that you copy everything from the first character on the first line of the window to the last character of the last line, including the BEGIN NEW CERTIFICATE REQUEST and END NEW CERTIFICATE REQUEST lines.

**Tip:** You might want to save the server certificate request in a text file to process it at a later date using an ASCII text editor such as NotePad. Do not use a word processor such as Word that inserts formatting or control characters.

8. Click **OK** to end this dialog with the Certification Admin application.
9. Go to the Certificate Authority's Web site and follow the instructions to create a request for a certificate. At the end of the process you normally get a pickup ID that you can use later to pick up the certificate that has been signed or approved by the Certificate Authority.

### **Merge the CA certificate as a trusted root into keyring file**

It is recommended that you install or merge the CA Trusted Root certificate into your keyring file.

The following steps are necessary for all Trusted Root CA certificates you are using or accepting signed certificates from. Please view the default trusted roots in the keyring file prior to these steps to make sure the third-party CA's certificate is not already included.

**Note:** In our scenario the VeriSign Test Certificate Authority is one of the default trusted root certificates that is already included in the keyring file of the domino server. To show the installation of the CA certificate as a trusted root into the keyring file we deleted the certificate prior to these steps.

Please follow these steps to merge the CA Trusted Root certificate into your server's keyring file.

1. Follow the instructions from the Certificate Authority to obtain the Trusted Root certificate. Usually you get the certificate as a file from the CA, or it is available for you to copy to a clipboard.

**Note:** *Do not* proceed to the next step until you have received the CA certificate from the Certificate Authority.

In our scenario we obtained the Trusted Root certificate from VeriSign as a file copy.

2. From the server Certificate Administration main menu (Figure 8-52 on page 408) click **3. Install Trusted Root Certificate into keyring.**

Workspace Server Certificate Admin - CertAdminFullScreen - Navigator Merge Trusted Root Certificate Into Key Ring

### Install Trusted Root Certificate

Use this form to install the Certificate Authority Trusted Root certificate into the server key ring. If you haven't already done so, first obtain the Certificate Authority Trusted Root certificate by choosing "Accept This Authority In Your Server" from the main menu of Certificate Authority Web site. **Note:** This step of installing the Certificate Authority Trusted Root certificate into your server key ring is recommended before installing certificates signed by this Certificate Authority into the key ring.

Key Ring Information		Quick Help
Key Ring File Name	j:\domweba\data\keyfile.kyr	Specify the key ring file.
Certificate Information		
Certificate Label	Verisign Trusted Root Certificate	The identifier you'll see for this certificate when you choose "View & Edit Key Ring" from the main menu.
Certificate Source	<input checked="" type="radio"/> File <input type="radio"/> Clipboard	The source of the certificate can be from a file or from the clipboard.
File Name	j:\domweba\data\VeriSignTestCA.cer	The name of the file containing the CA's Trusted Root certificate.
File Format	<input type="radio"/> Base 64 encoding <input checked="" type="radio"/> Binary file format	Base 64 encoding is most common. Binary format is used by some CA's (e.g., CAs based on the Microsoft CA Server).

Merge Trusted Root Certificate into Key Ring

Figure 8-56 Install Trusted Root Certificate

3. Enter the full path and name of the keyring file that will store this certificate in the field keyring File Name (see "Transferring the keyring file to your iSeries Domino server" on page 411). In our scenario this would be j:\domweba\data\keyfile.kyr.
4. Enter as the Certificate Label an easy to remember name to identify the certificate. In our Scenario this would be Verisign Trusted Root Certificate.
5. Do one of the following:
  - If you copied the Trusted Root certificate to the clipboard, select **Clipboard** in the Certificate Source field and paste the clipboard contents into the Certificate from Clipboard field.
  - If you received the Trusted Root certificate in a file from the CA, like we did in our example from VeriSign, select **File** in the Certificate Source field.
    - Enter the name and path of your received and detached Trusted Root certificate file name in the File Name field, in our scenario this would be j:\domweba\data\VeriSignTestCA.cer.
    - Change the File Format to Binary.
6. Click **Merge Trusted Root Certificate into keyring**.
7. Type in the password for your keyring file and click **OK**.

**Note:** As we have removed the VeriSign Test Certificate Authority as a trusted root prior to these steps to show the installation of a Trusted Root certificate, there is one confirmation screen missing here. When you install a new certificate as a Trusted Root Certificate you will see an additional confirmation screen similar to Figure 8-59 on page 418.

8. A confirmation screen appears confirming that the Certificate is received into the keyring and is designated as a trusted root (see Figure 8-57). Click **OK**.

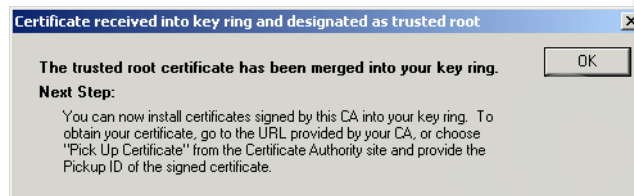


Figure 8-57 Merge Trusted Root Certificate completion

You are now able to use the CA's Trusted Root certificate to add the approved server certificate of your CA in your keyring file.

**Important:** If you are following our test scenario, using the Test certificate, it is important that you remove the installed trusted Root certificate, once you have finished testing. Note that a test CA does not verify any data in a certificate signing request. Therefore, you should never establish a trust relationship with test CAs in a production environment.

## Add the approved server certificate to the server keyring file

This is the last of the three steps to certify your server to a Certificate Authority.

**Important:** Perform this task only after you have received the approved or signed) certificate for your server from the Certificate Authority. Read the instructions that come from the CA regarding the procedures to pick up the certificate.

1. Please follow these steps to add the approved server certificate into your server keyring file. Receive the signed server certificate for your server from your CA, either:
  - As a file, and store it in the Notes data directory of your server.
  - As a copy in the clipboard, out of a note, like in our scenario in the e-mail received from VeriSign.

2. From the server Certificate Administration main menu (Figure 8-52) click **4. Install Certificate Into keyring.**

The Certificate Authority will notify when your signed certificate is ready. The specifics depend on the Certificate Authority, but typically you will receive an e-mail specifying a URL where you can pick up the certificate. Once you have obtained the signed certificate, this form lets you install it into your key ring. **Note:** Before installing this certificate, it is recommended that you install the certificate of the signing Certificate Authority in your key ring as a Trusted Root. If you haven't already done so, choose "Accept This Authority In Your Server" from the main menu of the Certificate Authority Web site to obtain the CA certificate.

Key Ring Information	Quick Help
Key Ring File Name <input type="text" value="J:\domweba\data\keyfile.kyr"/>	Specify the key ring file.

Certificate Information	Quick Help
Certificate Source <input type="radio"/> File <input checked="" type="radio"/> Clipboard	The source of the certificate can be from a file or from the clipboard.
Certificate from Clipboard: <pre>-----BEGIN CERTIFICATE----- MIICCTCCABMCEAI//Bshp+kUFSkdLaeIdgwDQYJKoZIhvcNAQEEBQAwgaksFjAU BgNVBAAoTDVZ1cm1TaWduLCBjb2MxRzBFBG9uVBAAsTPnd3dy52ZXJpc21mbi5jb20v cmVub3NpdG9yeS9UZXR0Q1BTIE1uY29ycC4gQnkqUmVnLiBMaWFiLiBMVEQuMUYw RAYDVQQLZz1Gb3IgVnVyaVNPZ24gYXV0aG9yaXplZCB0ZXN0aW5nIG9ubHkuIE5v IGFzc3VyYW5jZXNpZG9uVnV0aG9yaXplZCB0ZXN0aW5nIG9ubHkuIE5v NTk1OVowZjELMAkGA1UEBhMCVVMxPzAVBgNVBAGTDk5vcmRoIENhcm9saW5hMRhw EQYDVQKFApBdXRvbW90aXZ1MScwJwYDVQQDFCBkb213ZWJhLmlzZXJpZCZMaXZl by5yYUwuaWJtLmNvbTBcMAOGCSqGSIb3DQEBAQUAA0sAMEgCQQCppl5Fd2PGS0uX J0Bq/vvmcMB6GzEKvAegupYstvw+zzggQ8gRMD2jut6p9jPwYQNFInQjMhgyP4Pvr9 AIGUD0ZHAGMBAAEwDQYJKoZIhvcNAQEEBQADQQA4mw1zcTdbpIt4e84ndw3DQB0 3Lkx1iU+Z9yLw6Lv5t09RhQXP5j4cam061rfI+cFhwbbNTEnQ1MIZS0+M+ -----END CERTIFICATE-----</pre>	Paste the clipboard contents into this field.  <b>Note:</b> The pasted certificate must include the "Begin Certificate" and "End Certificate" lines.

Figure 8-58 Install Certificate into keyring

3. Enter the full path and name of the keyring file that will store this certificate in the field keyring File Name (see "Transferring the keyring file to your iSeries Domino server" on page 411). In our scenario this would be J:\domweba\data\keyfile.kyr.
4. Do one of the following:
  - If you received the approved server certificate in a file from the CA, select **File** in the Certificate Source field.
    - Enter the name and path of your received and detached server certificate file name in the File Name field.
    - Choose the file format as described in the installation notes from your CA.
  - If you copied the approved server certificate to the clipboard, choose **Clipboard** in the Certificate Source field and paste the clipboard contents into the Certificate from Clipboard field.

5. Click **Merge Certificate into keyring**.
6. Type in the password for your keyring file and click **OK**.
7. You will see a confirmation window as shown in Figure 8-59.

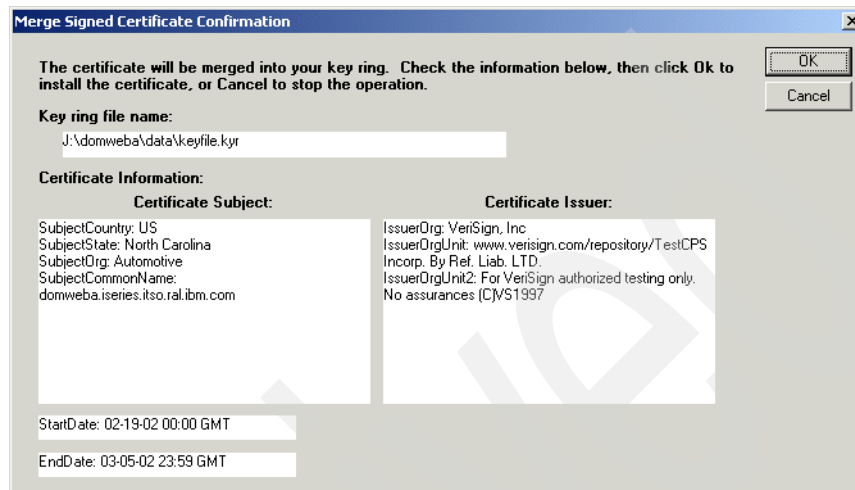


Figure 8-59 Merge Signed Certificate Information

8. Check the information given in the confirmation window and if everything is correct click **OK** to install the certificate into your keyring file.
9. A second confirmation window appears confirming that the certificate has been merged into your keyring, as shown in Figure 8-60.

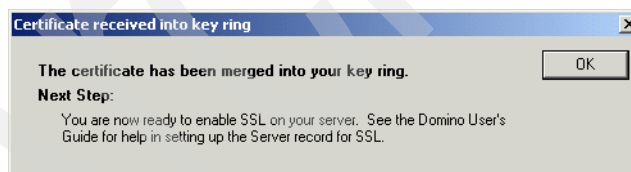


Figure 8-60 Certificate Merged into key ring

10. Click **OK** to finish the installation process for the signed server certificate.

You are now ready to enable SSL on your server.

## Changing the Domino Web SSO Configuration document

To change the Domino Web SSO Configuration document, use a Lotus Domino Administrator client Release 5.0.5 (or later) and perform the following steps:



1. From the Domino Administrator client, ensure that the correct Domino server is selected and you are on the Configuration tab, then expand **Web** and select **Web Server Configuration**, as shown in Figure 8-61.

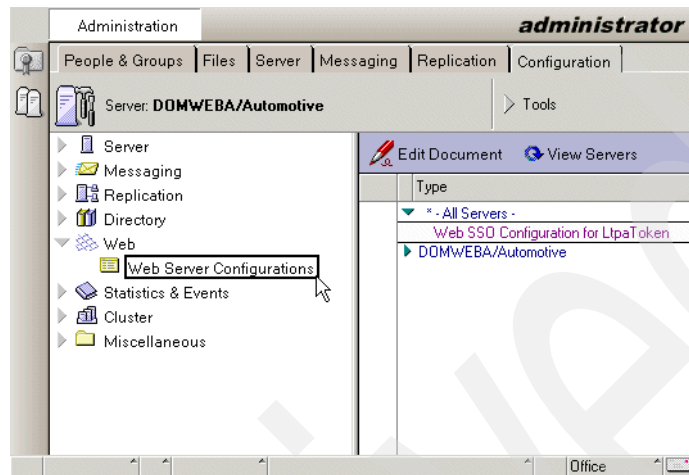


Figure 8-61 Domino Web Server Configuration

2. Under All Servers you have the Web SSO Configuration for LTPA token, that you created earlier in this chapter. Highlight this document and click **Edit Document**.
3. Click the **Web...** action button and select **Import WebSphere LTPA Keys** to import the regenerated keys from WebSphere, as shown in Figure 8-62 on page 420.

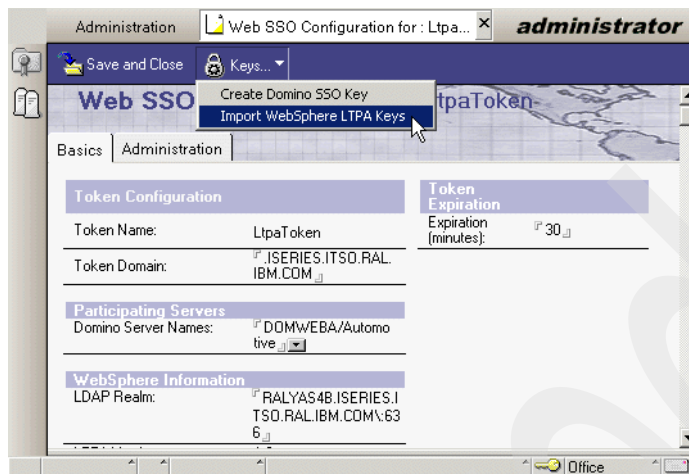


Figure 8-62 Importing WebSphere Keys to Web SSO Configuration

4. Enter the path to the LTPA keys file and click **OK** to import the file.
5. When prompted, enter the password that you used earlier when generating the LTPA keys and click **OK** to continue the key import.
6. Click **OK** on the Successfully imported WebSphere LTPA keys message.

The Web SSO Configuration document should automatically be updated to reflect the information from the LTPA keys file you just imported.

7. The only field that should have changed is the LDAP Realm. This should now include the secured port number at the end of the fully-qualified LDAP server name. In our example this would be RALYAS4B.ISERIES.ITSO.RAL.IBM.COM:636. You need to change this value as the LDAP server port value was specified for the WebSphere administrative domain. If a port is specified, a backslash (\) must be inserted in the value before the colon. In our example, replace RALYAS4B.ISERIES.ITSO.RAL.IBM.COM:636 with RALYAS4B.ISERIES.ITSO.RAL.IBM.COM \:636, as shown in Figure 8-63 on page 421.

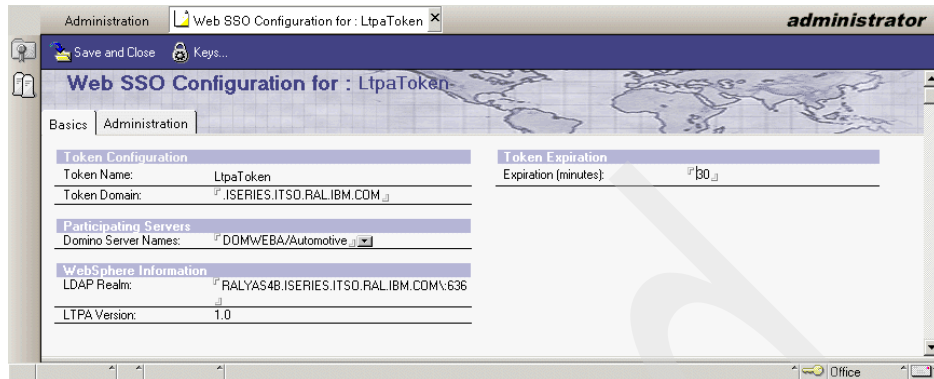


Figure 8-63 Web SSO LDAP Realm

8. Click **Save and Close** to close the Web SSO Configuration document.

## Change Directory Assistance document to enable SSL

Earlier in this chapter you created a Directory Assistance document but did not enable SSL. You are now going to change that document to enable SSL by doing the following:

1. Using the Domino Administrator client, select the **Configuration** tab, expand **Directory**, and click **Directory Assistance**.
2. Select the Directory Assistance document (Secureway) and click **Edit Directory Assistance**.
3. Select the **LDAP** tab and fill in or change the fields with the following values, as shown in Figure 8-64:

### Channel encryption

Change this from None to **SSL**. If set to SSL the Domino server will use SSL to connect to the LDAP server.

### Port

Change the port from the non-SSL 389 to the SSL port of 636. This is the port to use when connecting to the LDAP server.

### Accept expired SSL certificates

Choose **No** to enforce certificate expiration checking.

### SSL protocol version

Choose **Negotiated** to allow SSL to determine handshake and protocol.

**Verify server name with remote...** Select **Enable** to require that the subject line of the remote server's certificate and the LDAP directory server host name match.

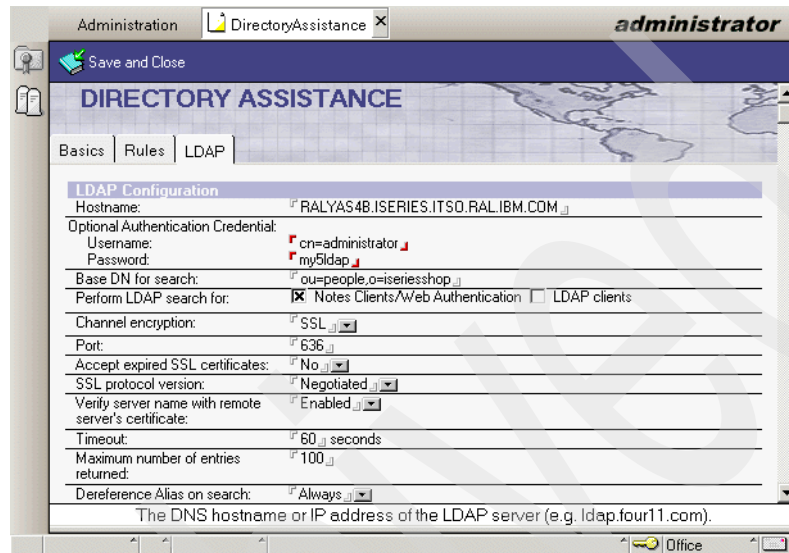


Figure 8-64 Directory Assistance: LDAP tab

4. Click **Save and Close** to save and exit this document.

### Enable SSL for LDAP in the Domino server document

We now describe how to activate SSL for LDAP on your Domino server, after all the previous steps have been done successfully.

**Important:** Please ensure that both files, keyfile.kyr and the keyfile.sth, are stored in your servers data directory and that QNOTES is the owner of both files and has read authority to them (refer to “Create a server keyring file” on page 407 for more information).

1. Start your Domino Administrator client and click the **Configuration** tab.
2. Expand **Server** and click **Current Server Document**.
3. Click **Edit Server**.
4. Click **Ports -> Internet Ports** and then **Directory**.

5. In the SSL settings section, specify the following values, as shown in Figure 8-65 on page 424:
  - Enter the SSL key file name you have used in the previous steps. In our scenario this would be `keyfile.kyr`. You can only enter a file name, no path information would be accepted here and the extension of the file has to be `kyr`.
  - Set the SSL protocol version (all protocols except HTTP) to **Negotiated**. Negotiated allows SSL to determine handshake and protocol. This would not restrict the clients and servers to using a specific SSL protocol and will allow them to freely negotiate what SSL level to use to secure the channel.
  - Set the Accept SSL site certificates to **No**, as we will accept certificates from all trusted root certificates.
  - Set the Accept expired SSL certificates to **Yes** to enable SSL to accept remote certificates that have expired. In a highly-sensitive environment you should not accept expired certificates and, thus, set this value to **No**.
6. In the Directory (LDAP) section specify the following values:
  - The TCP/IP port status should be set to **disabled**, as we will now use only the SSL port for LDAP.

The SSL section should be changed to the following values:

- The SSL port number specifies which port is used to connect to the LDAP server. In our scenario we use the standard port for SSL = 636.
- Set the SSL port status to **Enabled** to activate the port for SSL use.
- The Authentication options should be set according to the security policy you have for your environment. In our scenario we kept the default values. These are:
  - Set the Client Certificate to **No** to not allow X.509 certificates. Select **Yes** if you want to allow the use of X.509 client certificates for authentication. In our scenario we left the default as **No**.
  - We did not want to allow the use of Name & password for authentication, so we set this value to **No**.
  - We allow anonymous connections to the LDAP server, so this value is set to **Yes**.

The screenshot shows the Domino Server Configuration window for a server named 'DOMWEBAutomotive'. The 'Internet Ports' tab is selected, and the 'Web' sub-tab is active. The 'SSL settings' section is expanded, showing the following configuration:

- SSL key file name: keyfile.kyr
- SSL protocol version (for use with all protocols except HTTP): Negotiated
- Accept SSL site certificates: ☐ Yes ☒ No
- Accept expired SSL certificates: ☒ Yes ☐ No

Below the SSL settings, the 'Directory (LDAP)' section is expanded, showing the following configuration:

- TCP/IP port number: 389
- TCP/IP port status: Disabled
- Authentication options:
  - Name & password: Yes
  - Anonymous: Yes
- SSL port number: 636
- SSL port status: Enabled
- Authentication options:
  - Client certificate: No
  - Name & password: No
  - Anonymous: Yes

Figure 8-65 Configure Server Document to use SSL for LDAP

7. Click **Save and Close** to save these settings in the server document.

## Enable SSL for HTTP in the Domino server document

Here we will set up the Domino HTTP server to use SSL only.

1. Start your Domino Administrator client and open your Current server Document.
2. Click **Edit Server**.
3. Click **Ports, Internet Ports** and then **Web**.
4. Leave the SSL Security setting to its default value.
5. In the Web (HTTP / HTTPS) section specify the following values as shown in Figure 8-66 on page 425:
  - Set the TCP/IP port status to **Disabled** to stop the HTTP server from listening on the TCP/IP port number 80.
  - Check that the SSL port number is set to the default value of 443.
  - Set the SSL port status to **Enabled**.

- For the Authentication options select the following:
  - Set the Client Certificate to No to not allow X.509 certificates. Select Yes if you want to allow the use of X.509 client certificates for authentication. In our scenario we left the default as **No**.
  - As we use Name & password for authentication, set this value to **Yes**.
  - We do allow Anonymous connections to the LDAP server, so this value should be set to **No**.

The screenshot shows the Domino Administrator's configuration window for 'SERVER: DOMWEBAutomotive'. The 'Security' tab is selected, displaying the following settings:

**SSL settings**

- SSL key file name:
- SSL protocol version (for use with all protocols except HTTP):
- Accept SSL site certificates: ☐ Yes ☒ No
- Accept expired SSL certificates: ☒ Yes ☐ No

**SSL Security**

SSL ciphers:  RC4 encryption with 128-bit key and MD5 MAC, RC4 encryption with 128-bit key and SHA-1 MAC, Triple DES encryption with 168-bit key and SHA-1 MAC, DES encryption with 56-bit key and SHA-1 MAC, RC4 encryption with 40-bit key and MD5 MAC, RC2 encryption with 40-bit key and MD5 MAC

Enable SSL V2: ☐ Yes (SSL V3 is always enabled)

**Web (HTTP/HTTPS)**

- TCP/IP port number:
- TCP/IP port status:
- Authentication options:
  - Name & password:
  - Anonymous:
- SSL port number:
- SSL port status:
- Authentication options:
  - Client certificate:
  - Name & password:
  - Anonymous:

Figure 8-66 Configure Domino HTTP for SSL

6. Click **Save and Close** to close the Domino server document.
7. For these changes to take effect, the Domino server must be stopped and restarted. To do this, enter the following command from the Domino server console:

```
restart server
```

Press Enter.

The server will do a controlled shutdown and will restart within 10 seconds.

### 8.10.3 Testing SSO between Domino and WebSphere using SSL

We performed the following tasks to test the WebSphere application (SimpleServlet) to the Domino application SSO:

1. You will need to take the appropriate action in your applications to specify that HTTPS instead of HTTP is used for the secured connection.
2. Configure the Web browser to accept HTTP cookies. If you are using Internet Explorer, the type of cookies that need to be enabled are per-session (not stored).
3. For testing purposes only, configure the Web browser to prompt before accepting HTTP cookies. This will provide you with visual confirmation that the Domino and WebSphere application are generating and returning HTTP cookies to your Web browser after you authenticate.
4. From the Web browser, specify the URL for a resource protected by the Domino server. For example, open a database that defines no access for an anonymous user. This will verify that the token generated by the Domino server is accepted by WebSphere Application Servers. Make sure to enter the fully-qualified TCP/IP host name for the URL. For example, enter `https://myhost.mycompany.com/names.nsf` and not just `https://myhost/names.nsf`. In our example the URL would be:  
<https://ralyas4a.iseries.itso.ral.ibm.com/domwaslab.nsf/loanapp?openform>
5. You will be alerted that you are about to view pages over a secured connection, as shown in Figure 8-67 on page 427. Click **OK**.



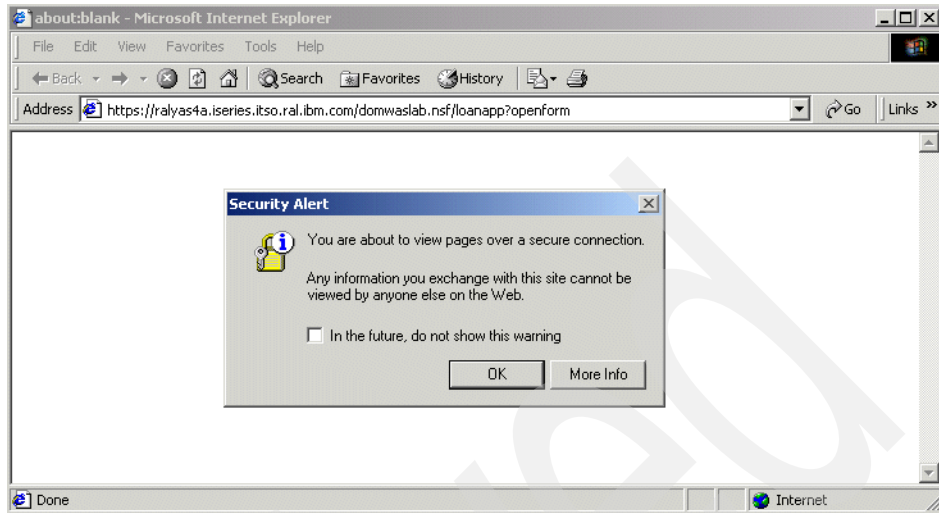


Figure 8-67 Security Alert

6. You see the warning Security Alert if there is a problem with the certificates. If you receive this alert you should click **View Certificates** to view the details of the problem. Once you have ascertained what the problem is you can click **Yes** to proceed.



Figure 8-68 Security Alert dialog box

**Note:** A closed padlock appears in your web browser, indicating that you have a secured connection. In Internet Explorer it looks like Figure 8-69. If you click on the padlock it gives you details of your certificates.



Figure 8-69 Internet Explorer Lock

7. When prompted for a user ID and password, make sure that you specify a user ID that is authorized for resources for both the Domino and WebSphere Application Servers. If you are using the default configuration settings, you should specify the user's shortname or user ID. Accept the HTTP cookie when prompted. You should now have access to the resource. In our example we gave Wendy access to the Domino database. We used the user ID and we are using the iSeries LDAP server. So we could sign on with the user ID of wendy and password of my51dap, as shown in Figure 8-70.

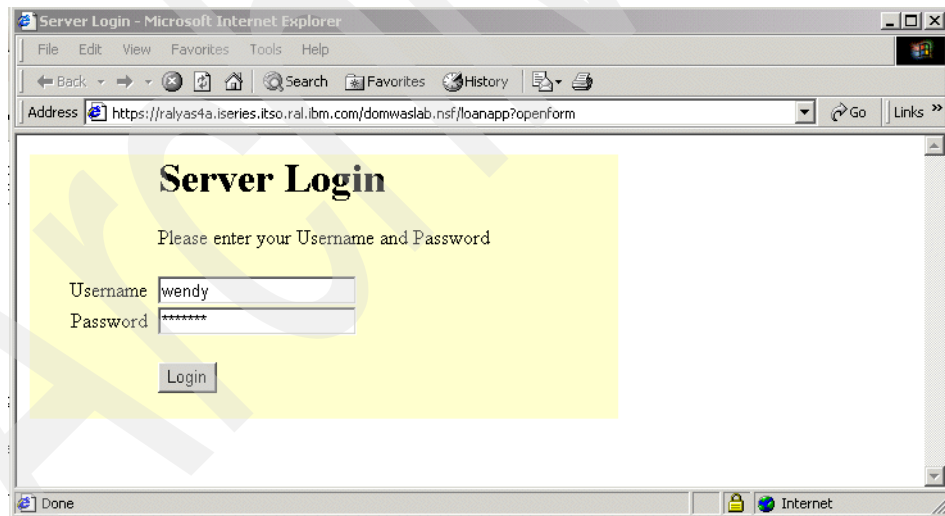


Figure 8-70 Sign on to Domino application

8. You are prompted to accept a cookie that is generated as a result of SSO. Click **OK**.

9. From the same Web browser session, access a resource protected by a WebSphere Application Server. You should have access to the resource without being prompted for a user ID and password. Make sure to enter the fully-qualified TCP/IP host name for the URL. For example, enter `https://myhost.mycompany.com/webapp/examples/showCfg` and not `https://myhost/webapp/examples/showCfg`. In our example from the Domino application we can click the **Submit** button to go to the WebSphere application as shown in Figure 8-71.

The screenshot shows a web browser window with the address bar displaying `https://ralyas4a.iseries.itso.ral.ibm.com/donwaslab.nsf/loanapp?openform`. The browser's status bar at the bottom shows 'Done' and 'Internet'. The web page has a header with the text 'DWI Bank' and a background image of a banknote. Below the header is a form with the following fields and values:

Name	wendy
Account Number	<input type="text"/>
Address	<input type="text"/>
City, State, Zip	<input type="text"/> , AK <input type="text"/>
Loan Tracking Number	DOMW-57KSVW
Loan Status	New
Type of Loan Desired	<input type="radio"/> Personal Loan <input type="radio"/> Car Loan <input type="radio"/> Mortgage
Term	<input type="radio"/> Please select Loan Type
Amount Desired	<input type="text"/>

At the bottom of the form is a 'Submit' button.

Figure 8-71 Domino application

10. From the second Domino application window, click **Return to Main Menu** to access the WebSphere SimpleServlet as seen in Figure 8-72 on page 430.



Figure 8-72 Link to WebSphere SimpleServlet

11. You are now prompted to accept a second cookie. This is the original cookie that appeared before enabling SSO. Click **OK**.

You should now see the SimpleServlet as shown in Figure 8-73. You should not be prompted to log into the WebSphere application.

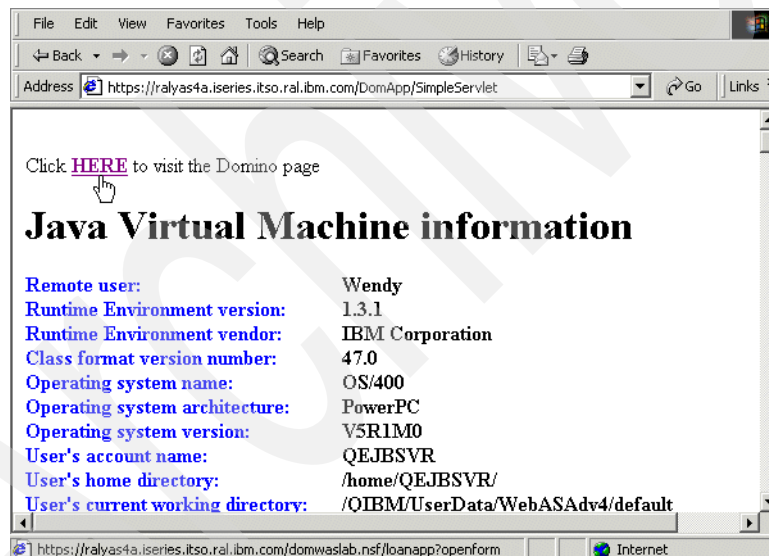


Figure 8-73 WebSphere SimpleServlet

12. Restart your Web browser session and perform the SSO verification steps mentioned above, but this time access a resource protected by a WebSphere Application Server first. In our example this would be:

<https://ralyas4a.ISERIES.ITSO.RAL.IBM.COM:80/DomApp/SimpleServlet>

When the WebSphere application of SimpleServlet is displayed, click **HERE** to access the Domino application.

This will verify that the token generated by the WebSphere Application Server is accepted by the Domino server(s). When prompted for a user ID and password, use the user's short name or user ID since this is the default WebSphere Application Server naming convention for users.

**Attention:** The Domino application database and the servlet used in this chapter to demonstrate the Single Sign-On process are available for download. Refer to Appendix F, “Additional material” on page 557 for further information on how to obtain these objects.

Archived



## LDAP directory: The enterprise directory for mail clients

Everybody knows the hassle of keeping individual address books for different mail clients or other applications. For example, some users might use Outlook as their mail client while others might use Netscape Messenger or Lotus Notes. Maintaining address books for each software product requires some effort and many companies cannot afford this luxury. One solution would be to maintain a single company-wide or cross-company directory that contains information about all employees, contractors, customers, and so forth. Then, whatever mail client is used to send an e-mail to one of these recipients, the recipient's e-mail address can be easily retrieved from the central directory. An LDAP directory is the right choice, since most of the currently available mail clients support LDAP search capabilities.

As part of our redbook example scenario, we will configure in this chapter a single LDAP directory that holds information about various groups of people. Since these groups belong to different organizations and companies, they do not use the same mail clients. This scenario allows us to show how various clients are configured to search for e-mail addresses in the LDAP directory. We cover Netscape, Outlook, and Lotus Notes.

This chapter describes the following:

- ▶ Configuring Notes mail clients to use LDAP directories
- ▶ Tips on searching for mail recipients using Lotus Notes
- ▶ Configuring Netscape Messenger to use LDAP directories
- ▶ How to search for mail recipients using Netscape Messenger
- ▶ Configuring Outlook mail clients to use LDAP directories
- ▶ How to search for mail recipients using Outlook



## 9.1 Scenario characteristics

The iSeries Shop and iSeries Automotive have both realized that their IT applications, as well as employees, benefit from using LDAP Directory Services. However, as the companies have merged, a number of different mail systems are still being used. iSeries Shop is using POP3 clients of Outlook Express and Netscape Messenger, iSeries Automotive and Acme Supply are using Lotus Notes.

All employees need access to all e-mail addresses on the different mail systems.

## 9.2 Scenario objectives

The objectives are:

- ▶ Allow employees from iSeries Shop to view e-mail addresses in the LDAP directory on RALYAS4B and the Domino LDAP directory of iSeries Automotive on RALYAS4A. The e-mail addresses of the users from Acme Supply have been imported from the Domino LDAP on RALYAS4C using LDIF into the LDAP directory on RALYAS4B.
- ▶ Allow employees from iSeries Automotive to view e-mail address of iSeries Shop in the LDAP directory on RALYAS4B and their own Domino LDAP directory on RALYAS4A. The e-mail addresses of the users from Acme Supply have been imported using LDIF into the LDAP directory on RALYAS4B.
- ▶ Acme Supply employees have no direct access to the LDAP directories of iSeries Shop or iSeries Automotive. But the LDAP entries from iSeries Shop will be export to them on a weekly basis.

The following list describes the solution to meet the stated objectives:

- ▶ Configure Domino servers/clients to search for LDAP entries when composing e-mails.
- ▶ Configure POP3 clients, Outlook, and Netscape Messenger to search for LDAP entries when composing e-mails.

## 9.3 Scenario network and system environment

Stage 5 scenario and system environment.

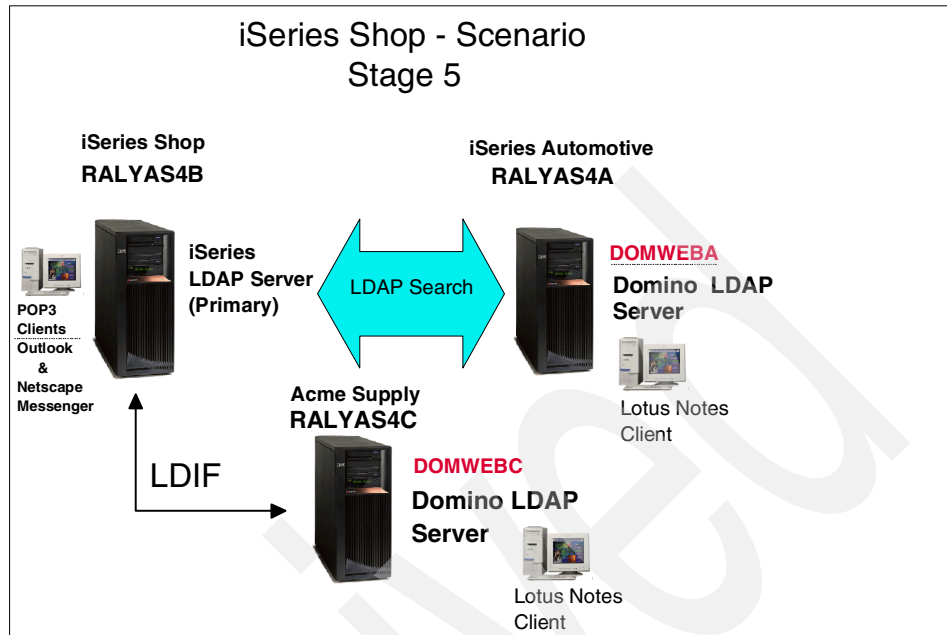


Figure 9-1 Redbook scenario - stage 5

## 9.4 Scenario prerequisites

The LDAP directories need to be set up for iSeries Shop on RALYAS4B, iSeries Automotive on RALYAS4A, and Acme Supply on RALYAS4C. Security needs to be enabled to ensure a secure connection to the LDAP directories. For more details on setting up iSeries LDAP and security, see Chapter 4, “OS/400 LDAP Directory Services” on page 55. For more details on setting up Domino LDAP see, Chapter 7, “Setting up LDAP on Domino server for iSeries” on page 291.

LDIF needs to be configured to import entries from the Acme Supply LDAP to iSeries Shop and export to entries from iSeries Shop LDAP into Acme Supply LDAP. See “Exporting and importing information via Operations Navigator” on page 187.

## 9.5 Task summary

The task list summary is as follows:

- Configuring Notes mail clients to use LDAP directories

- ▶ Tips on searching for mail recipients using Lotus Notes
- ▶ Configuring Netscape Messenger to use LDAP directories
- ▶ How to search for mail recipients using Netscape Messenger
- ▶ Configuring Outlook mail clients to use LDAP directories
- ▶ How to search for mail recipients using Outlook

## 9.6 Configure Notes mail clients to use LDAP

In this section we configure iSeries Automotive Lotus Notes mail clients to search for e-mail addresses of users in the primary LDAP directory on the iSeries Shop server.

There are two ways of doing this:

- ▶ Configure on the server for all users
- ▶ Configure on the client for one user

### 9.6.1 Configure LDAP address Look-up on the server for all users

To configure all Lotus Notes mail clients to look up addresses in one or more LDAP directories, you need to set up one or more Directory Assistance documents. This topic has been covered in “Creating the Directory Assistance database” on page 327.

### 9.6.2 Configure LDAP address look-up on a Notes client

To configure one Notes mail client to look up e-mail addresses of users on the iSeries Shop LDAP directory do the following:

1. From your Lotus Notes Local Address Book, click **Create**, from the drop-down menu select **Account**, as shown in Figure 9-2 on page 438.

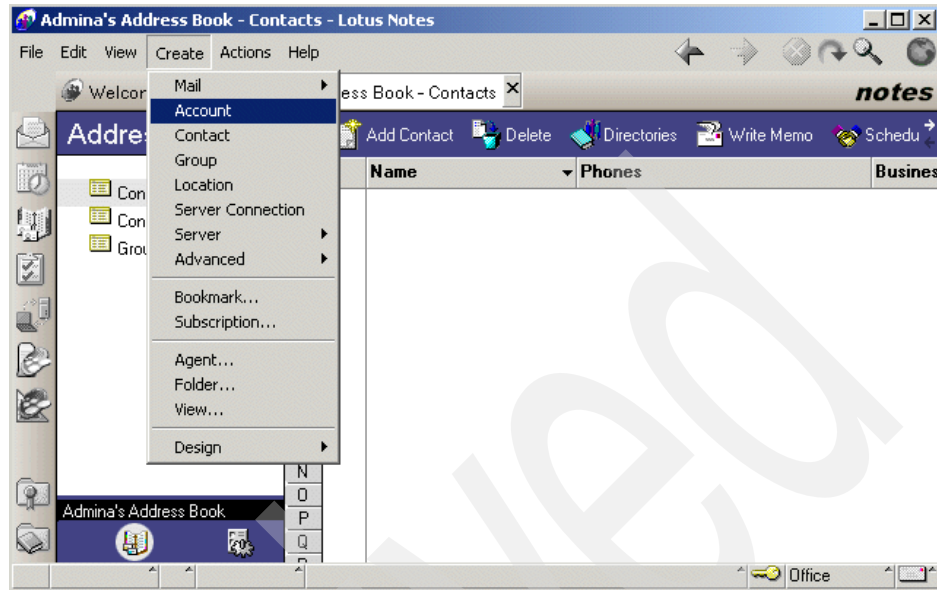


Figure 9-2 Create an Account in Local Address Book

2. On the Basic tab fill in the fields as follows:

**Account name** The name for this account. In our scenario we used iSeries Shop LDAP.

**Account server name** The DNS server name or IP address for the server you are connecting to when using this account. In our scenario we want to connect to the iSeries Shop LDAP directory, so the name would be alyas4b.iseries.itso.ral.ibm.com.

**Login name** Your login or username. You do not require a username for Internet LDAP directories. This would only be required if certain information you want to query is protected and requires a user to authenticate to the server.

**Password** The password associated with the login name. Not required.

**Protocol** The protocol required for this account. Select **LDAP**.

**SSL** If SSL is required change this field to **Enabled**.

Select the locations you want to use this account for. An asterisk (\*) means all locations. In our example we left this as the default of asterisk (\*). See Figure 9-3 for New Account Basic tab.

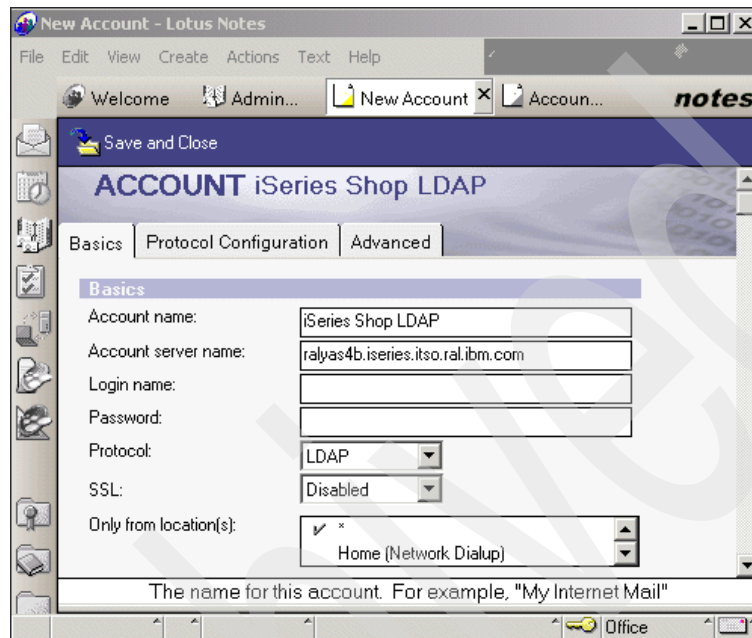


Figure 9-3 New Account - Basic tab

3. Click the **Protocol Configuration** tab and fill in the fields as follows:

**Search timeout**

Maximum length of time in seconds to wait for search results to be returned. In our example we left this as the default of 60 seconds.

**Maximum entries to return**

The maximum number of matches to return as results for a search. In our example we left this as the default of 100 entries. Depending on network bandwidth and directory size, you may want to reduce or increase the number.

**Search Base**

The base of the directory tree from which to start a search. Specify where in the server's directory tree to begin searches (for example, c=us,o=acme means country is U.S. and organization is Acme). To search the entire directory

tree, leave this field blank, unless the server requires an entry in this field. Some servers, such as Domino, do not require a search base. The Domino directory will be searched from top to bottom. However, many other servers, such as the IBM SecureWay Directory server, require a search base. In any case, it is always recommended to specify the exact search base to get the results as expected. In our example we used the search base of ou=employees,o=iserieshop.

**Check names when sending mail** Select **Yes** to check names against this server when sending mail.

**Use simple search filter** This specifies to use the simple form of search filter when querying the LDAP server. This may produce better search results. In our example it was not required.

See Figure 9-4 for the New Account Protocol Configuration tab.

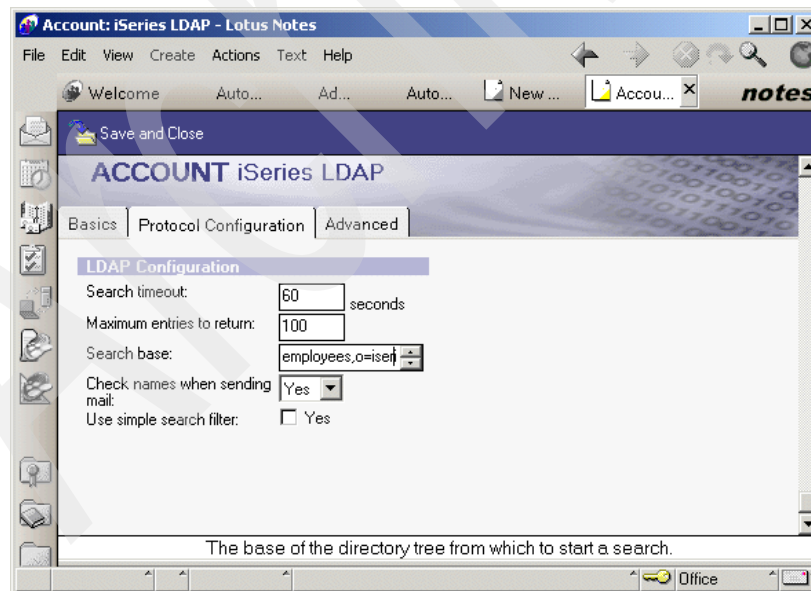


Figure 9-4 New Account - Protocol Configuration tab

4. Click the **Advanced** tab and fill in the field as follows:
 

**Port number** (Optional) Change the port number of the LDAP server if necessary. In our example we left this as the default of 389. The port 389 is the default port for non-secure LDAP connections.
5. Click **Save and Close** to create the new account.
6. To view all accounts, from the Tool bar select, **View -> Advanced -> Accounts**.

### 9.6.3 Searching for e-mail addresses in an LDAP directory

To look up entries in the LDAP directory when sending a new e-mail from Notes, do the following:

1. Type at least one character of the first or last name. If the e-mail address is not found in the Domino directory, as shown in Figure 9-5, then press F9 to do a look-up in the LDAP directory and the entry will be found as shown in Figure 9-6 on page 442.

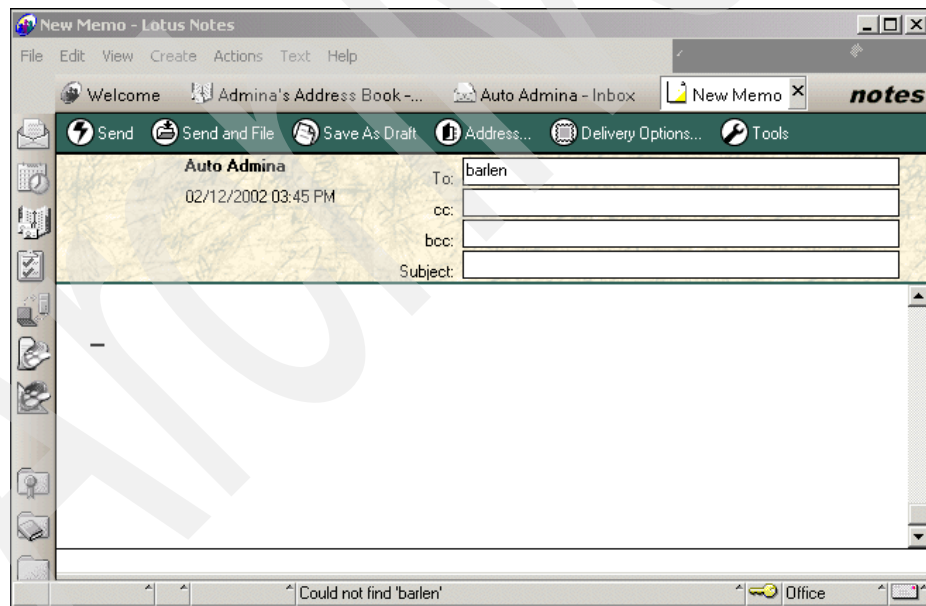


Figure 9-5 E-mail address not found in Domino directory

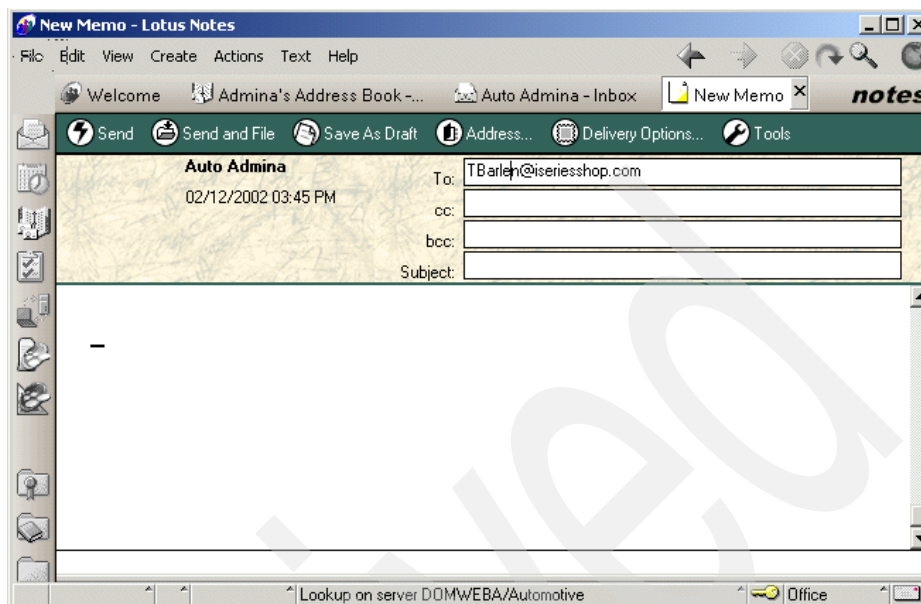


Figure 9-6 E-mail address found on LDAP server

## Tips on e-mail searches

While testing we found that your search results will vary depending on a number of factors. These include the following:

- ▶ Type Ahead does not search LDAP directories. It searches the Local Address Book, and if search criteria is not found it then searches the Domino dDirectory.
- ▶ F9 key:
  - After at least one search character, for example, L and F9, it searches the local account referral to the LDAP directory. If the search criteria is not found, it returns an error. It does not search Directory Assistance referrals to the LDAP directory.
  - After at least one search character and an asterisk (\*), for example, L\* and F9, it searches the Directory Assistance referral to the LDAP directory. If the search criteria is not found it then searches the local account referral to the LDAP directory.
- ▶ If you use the Address Assistance, it searches the local Account referral to the LDAP directory only.



In our example we have the following configuration:

<b>Directory assistance</b>	ou=people,o=iseriesshop
<b>Local account</b>	ou=employees,o=iseriesshop

There are two users with the first characters of Le in the LDAP directory with DNs of cn=Lee,ou=people,o=iseriesshop and cn=Leen,ou=employees,o=iseriesshop. There are no users in the local address book or Domino Directory starting with Le. When entering a new e-mail to Leen you could use Le + F9 or the Address Assistance. If entering a new e-mail to Lee you would use Le\* + F9.

The checking process is as follows:

- ▶ Personal Address Book on client
- ▶ Primary Domino directory of users mail server
- ▶ Local account referral to LDAP directory
- ▶ Directory Assistance database referral to LDAP

We found that you get the best search results when using the Address Assistant and select the directory you want to search in. For more information see Section 7.1.5, “Different ways to do directory searches” on page 299.

## 9.7 Configure Netscape Messenger to use LDAP

In this section we configure Netscape Messenger clients to use the primary LDAP directory on the iSeries Shop server to look up e-mail addresses of people in this directory. We also add the LDAP directory of iSeries Automotive Domino server to look up e-mail addresses of the Domino users.

You can add a directory server or modify the information for a selected server. You can then search the server for addresses. To add a new LDAP directory do the following:

1. Choose **New Directory** from the File menu of the Address Book window. To display the Address Book window, click the **Address Book** icon on the component bar. You see the Directory Server Property dialog box.
2. On the **General** tab of the Directory Server dialog box, fill in the following fields:

<b>Description</b>	Enter the name of the directory service, such as iSeries Shop.
<b>LDAP server</b>	Enter the Internet address of the server, such as ralyas4b.iseries.itso.ral.ibm.com.

**Search root**

Enter a code that restricts searching to a specific area. For example, us restricts the search to America only. Search root also specifies the organization to search on within the directory (for instance, o=iseriesshop,c=US). In our example we used ou=employees,o=iseriesshop.

**Port number**

Enter the port number that the LDAP server allows connections on. The default is 389.

**Maximum number of hits**

Enter the maximum number of search results to return. In our example we left the default as 100.

**Save password**

Check this to have Messenger remember the password for this directory. This is only required if your LDAP server requires you to authenticate.

**Secure**

Check this if the server uses a secure connection.

Our example is shown in Figure 9-7.

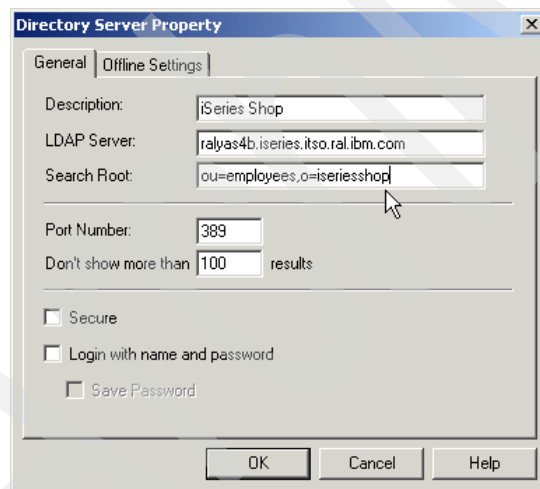


Figure 9-7 Setting up new directory account for Netscape

3. Click **OK**. We repeated the same steps for the Automotive directory on the Domino server (DOMWEBBA) on RALYAS4A.

To check e-mail addresses in an LDAP directory while composing Netscape Messenger e-mails we found the best approach is the following:

1. Enter at least one character for the recipients name; Netscape only searches your local address book for entries.
2. To search for a LDAP directory entry, select the **Address book** icon.
3. From the Select Address window, select the LDAP directory you want to search. In this scenario we used our new entry iSeries Shop.
4. Enter in the search characters of the recipient you want to find.
5. When the required name is found, select it, then select **To**, **Cc** or **Bcc** to specify the type of recipient.
6. Click **OK** to return to the Composition window; the e-mail address will be displayed.

## 9.8 Configure Outlook mail clients to use LDAP

In this section we configure Outlook mail clients to use the primary LDAP directory on iSeries Shop to look up e-mail addresses of people in this directory. We also add the LDAP directory of the iSeries Automotive Domino server to look up e-mail addresses of the Domino users.

1. From the Outlook Express client, click on **Address Book**.
2. From the Outlook Express Address Book, click **Tools** and from the drop-down menu select **Accounts...** as shown in Figure 9-8 on page 446.

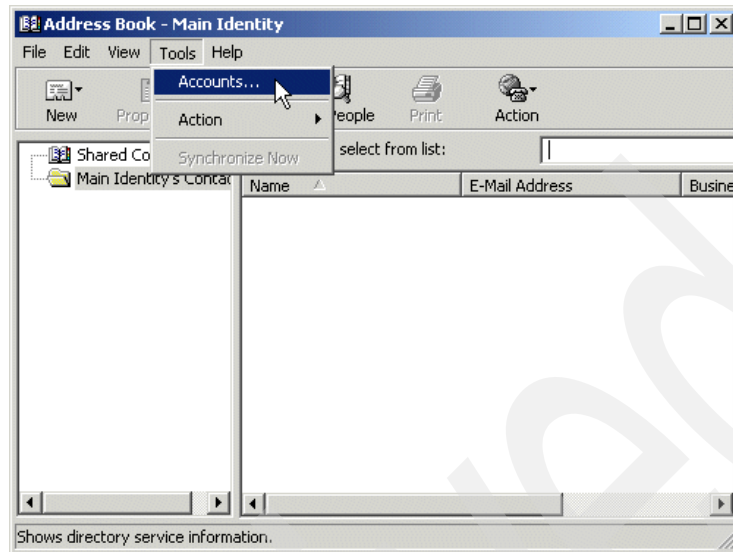


Figure 9-8 Create Outlook accounts

3. From Internet Accounts click **Add**, as shown in Figure 9-9.

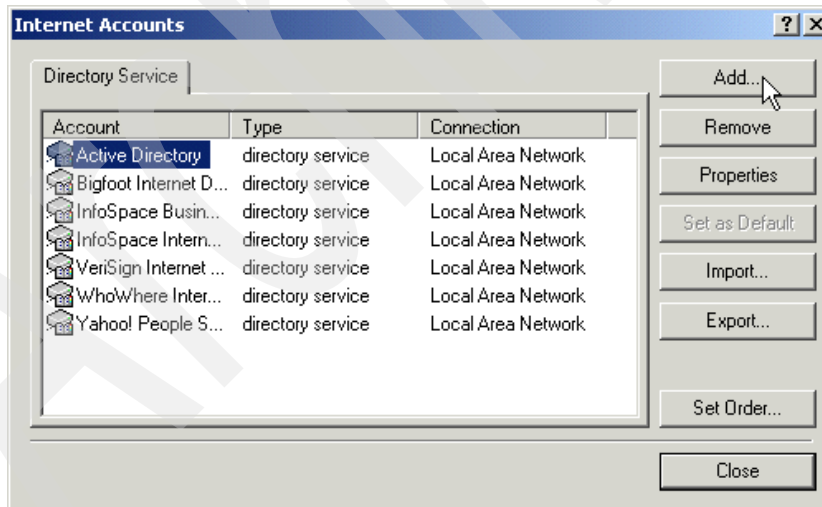


Figure 9-9 Adding new directory service for Outlook

4. In the Internet (LDAP) Directory field, enter the fully-qualified name of your LDAP directory. As shown in Figure 9-10, the value used in our example is:  
RALYAS4B.ISERIES.ITS0.RAL.IBM.COM

We did not need to check the My LDAP server requires me to log on field as we do not require users to authenticate with this LDAP server, but this will depend on your environment.

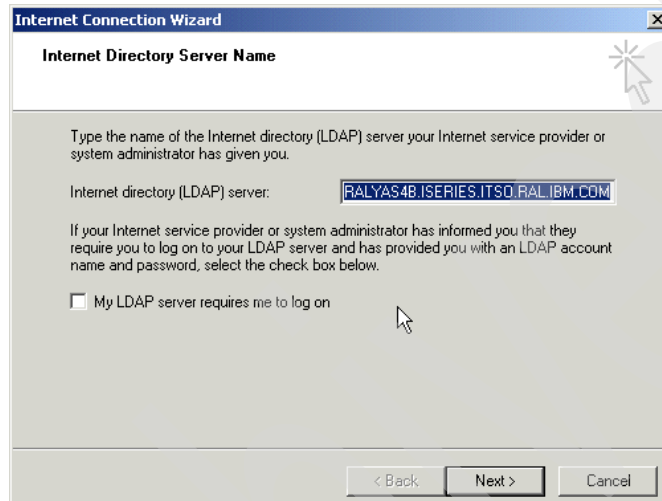


Figure 9-10 Internet Connection Wizard for new directory service

5. Click **Next** to continue.
6. In the Check E-mail Addresses window, check **Yes**, as we do want to check this directory service for e-mail addresses, as shown in Figure 9-11 on page 448.

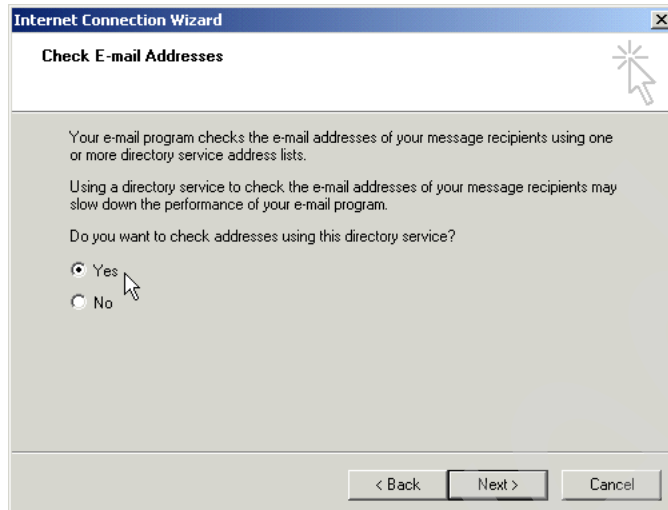


Figure 9-11 Check e-mail addresses in this directory

7. Click **Next** to continue.
8. In the Congratulations window, click **Finish**.
9. You will now see the new account in the directory service list, as shown in Figure 9-12.

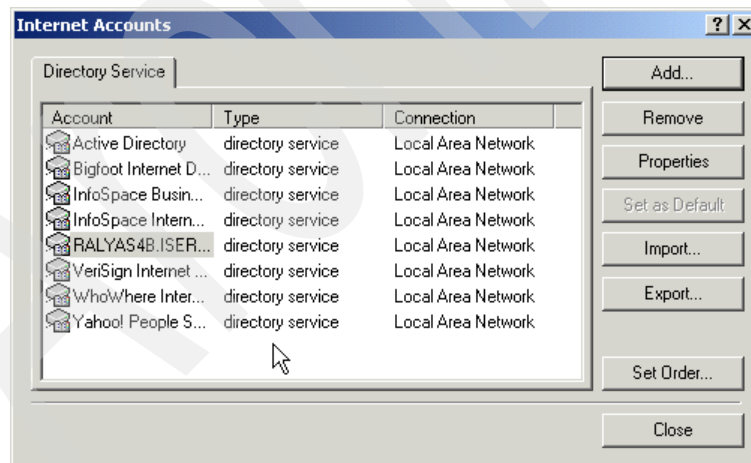


Figure 9-12 New directory service account

10. Select your new directory service account and click **Properties**.

11. In the first field you can change the name of the directory service. In our example we changed this to iSeries Shop, as shown in Figure 9-13.

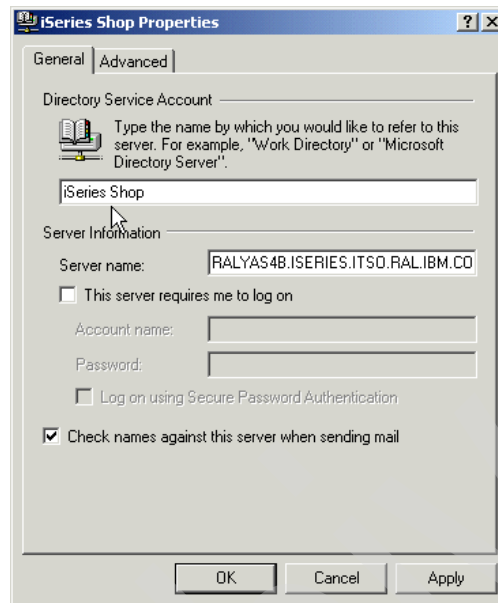


Figure 9-13 Change properties of Outlook account

12. Click the **Advanced** tab, then change the Search base field to the base DN of your LDAP directory. In our example this is ou=employees,o=iseriesshop, as shown in Figure 9-14 on page 450.

**Note:** If using a Domino LDAP directory, the search base is not required, but strongly recommended.

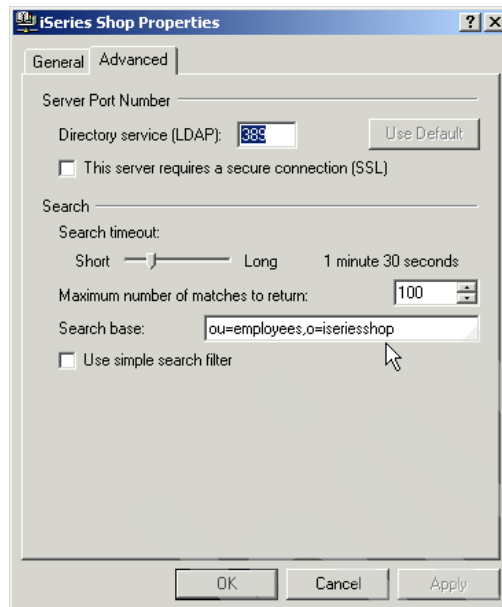


Figure 9-14 Change search base for iSeries Shop

13. Click **OK** to save and close.
14. You can set the order in which directory services are searched by clicking the **Set Order...** button in the Internet Accounts window, as shown in Figure 9-15 on page 451.



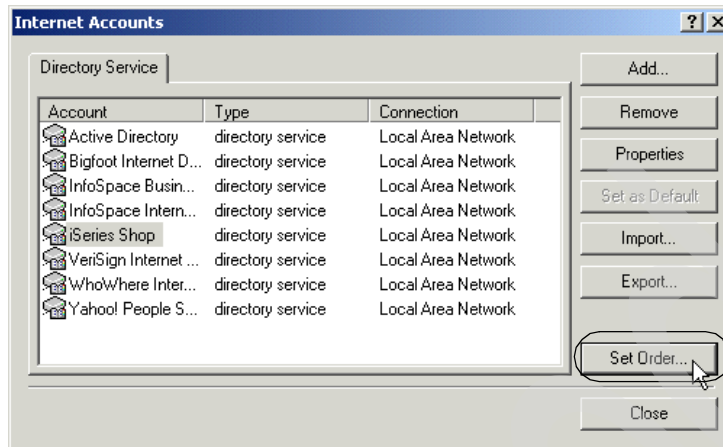


Figure 9-15 Set LDAP directory search order

15. Select the directory you wish to move and then use the **Move Up** or **Move Down** button. In our example we want to move the iSeries Shop to the top by selecting move up, as shown in Figure 9-16.

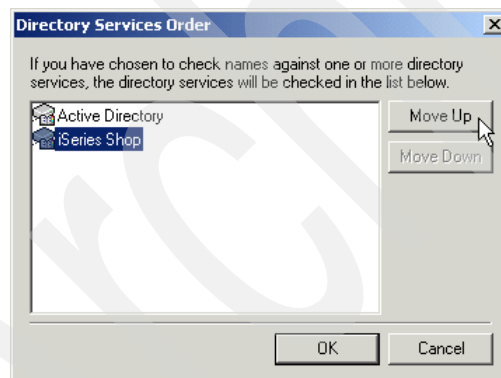


Figure 9-16 Move directory up to search it first

16. Click **OK** to continue.
17. Repeat the above steps to add additional LDAP directories.

In our example we added another two LDAP directories, for the Domino LDAP on iSeries Automotive and Acme Supplies. These entries are shown in Figure 9-17 on page 452.

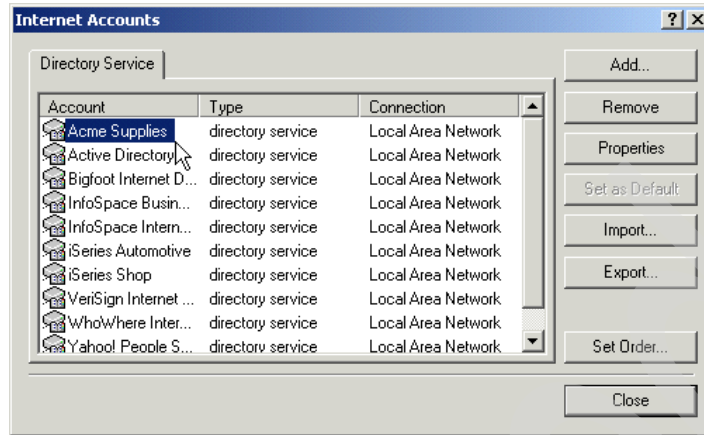


Figure 9-17 All the new directory service accounts

18. Close the Internet Accounts window, then from your Address Book select **Find People**.
19. Click the drop-down bar in the Look in field and select the LDAP directory you want to search. In our example we selected the iSeries Shop, as shown in Figure 9-18.

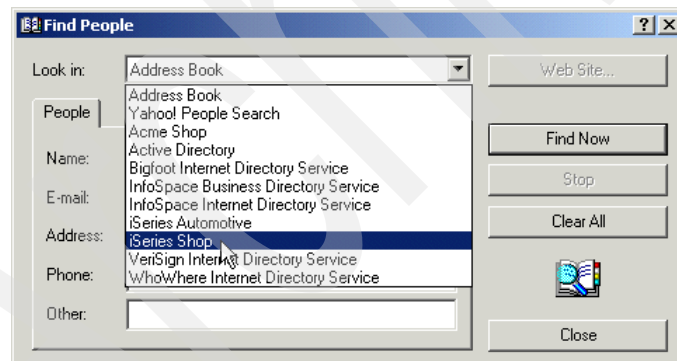


Figure 9-18 Select Directory to search for a e-mail address

20. In the Name field enter at least one character of the recipients first or last name, then click **Find Now**. In our example we entered w as the search criteria and it returned three names from the iSeries Shop LDAP directory, as shown in Figure 9-19 on page 453.

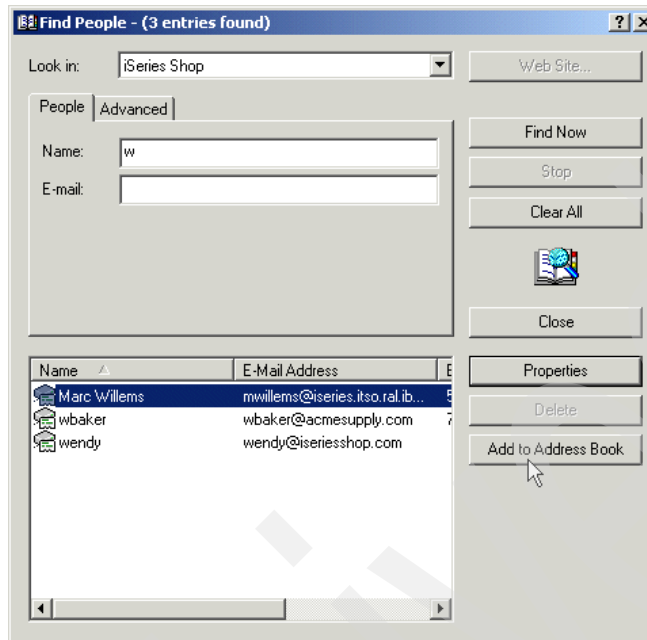


Figure 9-19 Search results from iSeries Shop

You could now select one or more entries to add to your address book.

To check an e-mail address in a LDAP directory while writing an Outlook Express e-mail, we found the best approach to this is as the following:

1. Click the **Address Book** icon before typing the user's e-mail address.
2. In the Select Recipients window, click **Find...**
3. In the Find People window, select the LDAP directory you require, type at least one character, then click **Find Now**.
4. Once it has found the user(s) based on your search criteria, you can select the name required, then click **To**, **Cc**, or **Bcc**.

**Restriction:** While testing we found that **Check Name** will only check for names in your local address book. It will only go on to list names in the first LDAP directory if there are no names in your local address book that meet your search criteria. Using the above method will ensure that you get all recipients from the selected LDAP directory that meet your search criteria.





## Part 4

# Developing directory- enabled applications



## Introduction

As you have seen throughout this book, OS/400 and many other applications are already LDAP-enabled. They utilize LDAP directories for various purposes, such as storing and retrieving configuration information, user authentication, and so forth. It is a logical step to take full advantage of directories in user applications too. This chapter gives you an idea of how to leverage LDAP directories in self-written applications and introduces the various programming interfaces and methods to directory-enable your applications. For example, your company sets up an enterprise address book via an LDAP directory and applications, where e-mail clients already leverage the directory to look up e-mail addresses or phone numbers. The information in the directory can, of course, be used for various other purposes, such as the payroll application to retrieve employee addresses, or a print management application that provides to each application the printer name and queue of the closest printer for an employee. You may also want to allow employees to update their own information in the directory. All these uses require some sort of programming you have to create unless you bought a software that does exactly what you need.

Due to the variety of different interfaces to access and work with an LDAP directory, you can directory-enable virtually all applications whether they are written in C, RPG, COBOL, or Java, to name a few. You can write applications to be used on user workstations to access directory data, or you enable your applications on the iSeries server to utilize Directory Services. In this case, the iSeries server application acts as an LDAP client. The LDAP directory server may or may not run on the iSeries server.

## 10.1 Application programming interfaces

OS/400 provides a series of Application Programming Interfaces (APIs) that allow you to search a directory or perform operations, such as additions, modifications, or deletions of directory entries. All these APIs are available for use in the Integrated Language Environment (ILE), such as ILE C, ILE RPG, and so forth. This redbook contains some examples of how to use APIs in an RPG application to search for a specific directory entry or add a new entry into a directory. These sample applications are covered in Chapter 11, “Using APIs to directory-enable your applications” on page 461.

Another source of excellent information about Directory Services APIs can be found in the document *LDAP API Listing* found in the iSeries Information Center by clicking **Programming -> CL and APIs -> APIs -> APIs by category -> Directory Services**.

## 10.2 Java applications

Java applications, also, whether they are running as a stand-alone application, a servlet, or in any other form, can utilize information stored in an LDAP-accessible directory. There are various packages available that allow you to work with data in a directory in any conceivable way. The interface that allows you to directory-enable your Java application is called the Java Naming and Directory Interface (JNDI). The JNDI is part of JavaSoft's platform application program interfaces (API). With JNDI you can connect seamlessly to multiple naming and directory services. You can build powerful and portable directory-enabled Java applications by using this interface. JavaSoft developed the JNDI specification with leading industry partners, including IBM, SunSoft, Novell, Netscape, and Hewlett-Packard Co.

In Chapter 12, “Using the JNDI to search and update the directory” on page 477 we provide a sample application that allows you to search a directory, display the results on a user's browser, and update certain attributes based on the authenticated user. This sample application gives you a fairly good understanding of what you have to cope with when working with LDAP directories in your Java application. Fortunately, Java is a open language and, as such, is used by many programmers who also share their experience on the Web. For whatever problem you have, we recommend to searching the Internet.

For more information on the JNDI interface and how to use it, refer to:

<http://java.sun.com/products/jndi/docs.html>



## 10.3 IBM SecureWay Directory Client SDK

The LDAP APIs, as well as other directory management tools, are also available through the IBM SecureWay Directory Client SDK for Windows 95, 98, and NT. The SDK includes the following:

- ▶ C header files, library files, online documentation, and sample programs
- ▶ Java Naming and Directory Interface (JNDI) LDAP service provider (IBMJNDI)
- ▶ LDAP command line utilities
- ▶ Directory content management graphical user interface
- ▶ Directory Management Tool (DMT)

The SDK can be found in OS/400 Version 5 Release 1 in the following IFS directory:

`\QIBM\ProdData\OS400\DirSrv\UserTools\Windows`

This directory contains a file `setup.exe` that when extracted, installs the SDK with the included tools on the PC workstation. You can easily access this directory by mapping the predefined share `QDIRSRV` on your iSeries server. To be able to map a shared directory, the AS/400 NetServer has to be started via Operations Navigator or by using the OS/400 command **STRTCPSVR SERVER(\*NETSVR)**.

More information about the IBM SecureWay Directory Client SDK can be found at:

<http://www.ibm.com/software/network/directory/library/>



# Using APIs to directory-enable your applications

Whether you want to write new applications using one of the Integrated Language Environment (ILE) programming languages or modernize existing applications on your iSeries server, you certainly should consider directory-enabling them. OS/400 Directory Services provides a rich set of application programming interfaces (APIs) that allow you to search and update entries in your LDAP directory.

This chapter gives you some examples of how to use some of these APIs for searching and updating the directory.

## 11.1 Overview

The set of LDAP APIs are designed to provide a suite of functions that can be used to develop directory-enabled applications. Directory enabled applications will typically connect to one or more directories and perform various directory-related operations, such as:

- ▶ Adding entries
- ▶ Searching the directory and obtaining the resulting list of entries
- ▶ Deleting entries
- ▶ Modifying entries
- ▶ Renaming entries

The type of information that is managed in the directory depends on the nature of the application. Directories are often used to provide public access to information about people, including:

- ▶ Phone numbers
- ▶ E-mail addresses
- ▶ Fax numbers
- ▶ Mailing addresses

Increasingly, directories are being used to manage and publish other types of information, including:

- ▶ Configuration information
- ▶ Public key certificates (managed by Certification Authorities)
- ▶ Access control information
- ▶ Locating information (how to find a service)

The LDAP APIs provide for both synchronous and asynchronous access to a directory. Asynchronous access makes it easy for your application to do other work while waiting for the results of a potentially lengthy directory operation to be returned by the server.

The LDAP APIs on the iSeries server are grouped into the following categories.

- ▶ APIs, such as the Change Directory Server Attributes (QgldChgDirSvrA) or the Configure Directory Server (QgldCfgDirSvr) APIs are provided with OS/400 Directory Services to configure or retrieve LDAP server attributes. This can be very useful when distributing an application that sets up Directory Services. The setup would then be done via a program using the APIs rather than performing the changes using Operations Navigator.

- ▶ A second set of APIs, the Export LDIF file (QgldExportLdif) and Import LDIF File (QgldImportLdif) APIs allow you to export the entire or a subtree of your OS/400 LDAP directory or to import directory data from a LDAP Data Interchange Format (LDIF) file. These APIs can be used instead of the export and import options from the Directory configuration menu in Operations Navigator.
- ▶ The third group are publishing APIs, such as the Synchronize System Distribution Directory to LDAP (QGLDSSDD) API. This API publishes System Distribution Directory (SDD) entries to an LDAP directory and keeps the LDAP directory synchronized with changes made in the System Distribution Directory. Another example is the Publish Directory Object (QgldPubDirObj) API.
- ▶ The last and largest group of APIs are the ones that are discussed in more detail in the remainder of the chapter. These APIs, such as the `ldap_init()`, `ldap_simple_bind()`, or `ldap_search()` APIs, are used in LDAP client applications to access information on the LDAP directory server. These APIs are also referred to as C APIs due to the fact that almost all programming examples and API descriptions are written for the C programming language. However, you can also use these APIs in any other ILE programming language on your iSeries server. In general, all APIs that perform an action on the directory, such as adding, modifying, and removing an entry, as well as binding to or searching the directory, are available in two flavors. These flavors allow for synchronous and asynchronous processing. When using the synchronous method (API names ending with `_s`, such as `ldap_search_s()`), your program waits for the API request to complete before continuing. Asynchronous access makes it easy for your application to do other work while waiting for the results of a potentially lengthy directory operation to be returned by the server.

## 11.2 Programming techniques for using APIs and C functions in ILE RPG

Almost all C functions and all APIs are either procedures in service programs or independent programs. ILE RPG has several ways of calling external routines such as programs or procedures in service programs. Calls can be made without any preamble, by using the `CALL`, or `CALLB`, operation codes. However it is recommended, and sometime necessary, to first define a prototype and then use the `CALLP` operation code, or simply name the procedure in an `EVAL` statement. Naming the procedure in an `EVAL` statement is the most common and

convenient method. It also allows procedures to return a value - as many C functions do. For more information on calling procedures from ILE RPG read the *ILE RPG Programmer's Guide* or *ILE RPG Reference* found in the iSeries Information Center by clicking **Programming -> RPG -> ILE**.

## 11.2.1 Prototypes

Briefly, a prototype is a description of a procedure. It documents the types of parameters the procedure uses and the value, if any, that the procedure returns. For example, see Figure 11-1.

```
* Prototype for QExample, which.....
D Example      pr          10i 0 extproc('QsyExample')
D $C1          15
D $N1          10i 0
      .....
* Call example procedure to get result
C      Eval    Result = Example(25: 'ABC')
      .....
```

Figure 11-1 Prototype example

Prototypes must be placed in the data definitions. This prototype defines the external procedure `QsyExample` as returning a 10 digit integer and requiring two parameters. Within the RPG program the name `Example` will be used to call the procedure.

Prototypes have several advantages:

- ▶ They allow the compiler to verify that the parameters used on your procedure calls are of the correct type and size.
- ▶ They allow you to call a procedure by name within an Eval statement
- ▶ They allow you to use constants on procedure calls such as 123 or ABC. The compiler automatically converts these to the appropriate data types.
- ▶ They are the only way to obtain procedure return values as in the example in Figure 11-1.

For more information on prototypes in ILE RPG programs read the *ILE RPG Programmer's Guide* or *ILE RPG Reference* found in the iSeries Information Center by clicking **Programming -> RPG -> ILE**.

When programming in C, the prototypes for all the C functions, the associated data structures, and useful constants, are readily available. They are held in files such as QSYSINC/H member LDAP. C program source normally includes instructions for the relevant files to be copied in by the compiler. Unfortunately such files are not generally available for ILE RPG. This can make getting started somewhat slow as you have to define the prototypes and data structures needed for the APIs and C functions you want to use.

When defining prototypes and their associated data structures for use in ILE RPG, it is best to put them in a separate file of related functions. In your programs you can use /copy file-name,member-name at the start of the data definition section so that the definitions are copied into your source by the compiler. This keeps your definitions in one place, ensuring consistency and saving retyping.

For more information about defining prototypes and how to pass parameters and variables, as well as process error structures, refer to the following redbooks:

- ▶ *IBM eServer iSeries Wired Network OS/400 V5R1 DCM and Cryptographic*, SG24-6168
- ▶ *Who Knew You Could Do That with RPG IV? A Sorcerer's Guide to System Access*, SG24-5402

## 11.3 Where to find API and C function documentation

All API documentation can be found in the Information Center by clicking **Programming -> CL and APIs -> APIs** then **Alphabetical list of APIs, APIs by category**, or **APIs by description**. When using the APIs by category option, extend the Directory Services API section to see the list of available LDAP APIs.

If you know the name or description of the API you want, the **APIs by description** option is good. You can use your browser's search function, Ctrl F normally, to search the list for the API name or for words describing the API.

C functions are also documented in the Information Center. Read *ILE C for AS/400 Run-Time Library Reference* or *IBM Open Class Library Reference* found in the iSeries Information Center by clicking **Programming -> C and C++ -> ILE**.

Functions intended for use in C programs will almost always specify the file, or files, holding the required prototype and data structure definitions. There will be one or more lines such as `#include <ldap.h>`. The example specifies the member ldap in the default file and H. QSYSINC is the library holding these files.

These files contain data structure and constant definitions that can be used when defining your prototype files for ILE RPG. For the LDAP APIs you have the `ldap` and `ldapsl` members in file `H` in library `QSYSINC` that contain the described information. The `ldapsl` member contains the SSL return codes.

There is also the file `QRPGLESRC` in library `QSYSINC`. This holds RPG structure definitions and function prototypes, for example, for the Directory Publishing APIs. These APIs are contained in the service program `QSYS/QGLDPAPI`. They can be useful for obtaining the data structures you need.

## 11.4 API flow when searching a directory

This section provides you with an overview of what APIs can be used to perform a search operation in a LDAP directory. This example, based on LDAP Version 3 APIs, shows one way of searching the directory using a non-secure session in synchronous mode. There are also APIs available to initiate an SSL session to the LDAP server. The APIs are documented in the order they have to be used. Figure 11-2 on page 467 shows an overview of the APIs and the order in which they are used.



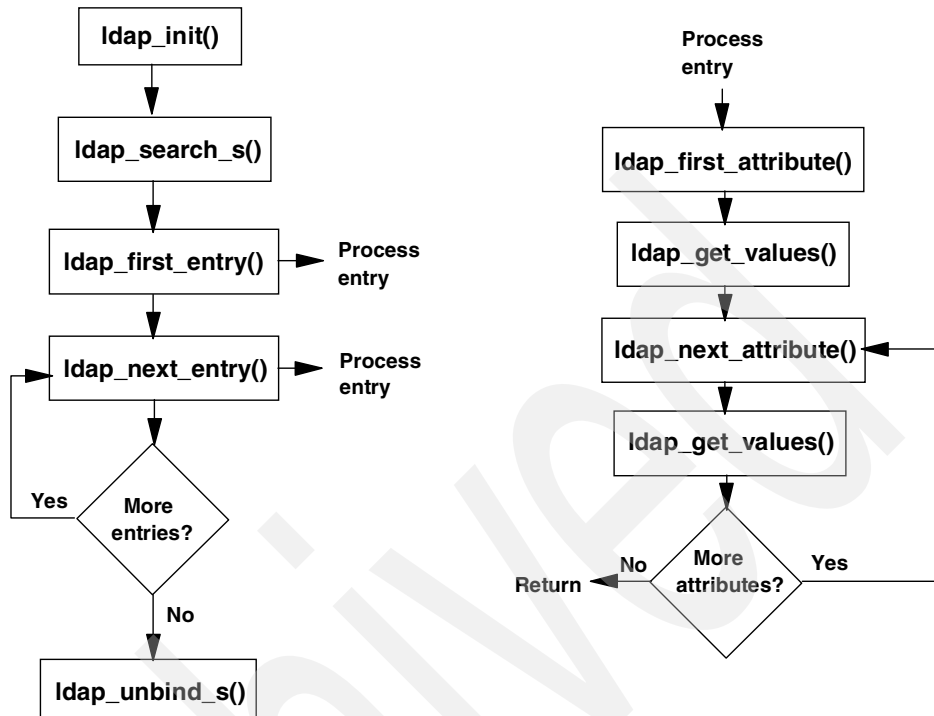


Figure 11-2 API overview for searching a directory

To better understand why you need to process entries, attributes, and values, take a look at Figure 11-3 on page 468.

Entry 1: cn=John McMeeking,ou=employees,o=iseriesshop

Attribute: sn	Value: McMeeking
Attribute: givenName	Value: John
Attribute: mail	Value: john@iseries.itso.ral.ibm.com
Attribute: telephoneNumber	Value: 919-411-1111
	919-111-0000
	919-111-0002

Entry 2: cn=Thomas Barlen,ou=employees,o=iseriesshop

Attribute: sn	Value: Barlen
Attribute: givenName	Value: Thomas
Attribute: roomNumber	Value: E307
Attribute: mail	Value: tbarlen@iseries.itso.ral.ibm.com
Attribute: telephoneNumber	Value: 919-311-2222
	919-222-0000

Entry n: cn=.....

*Figure 11-3 Example of directory entries*

As you can see in Figure 11-3, you may have multiple entries in a directory. Each entry contains one or more attributes. Depending on the attribute definition, a single attribute may contain one or more values. Likewise you have to process your search results. In the example given in Figure 11-2 on page 467 the APIs are processed in a certain order. The following information explains the purpose of each API used in our example.

**Important:** For all LDAP API descriptions in this example, the iSeries Information Center contains details for each API including parameter and return code explanations. However, you will notice that the product documentation refers in many cases to constants for parameter values. These constants are defined in the C header include member LDAP in file H in library QSYSINC. Unless you defined the constants in your prototype files, you need to enter the values for these constants rather than their names.

## ldap\_init()

The `ldap_init()` API initializes a session with an LDAP server. The server is not actually contacted until an operation is performed that requires it, allowing various options to be set after initialization, but before actually contacting the host. It allocates an LDAP structure that is used to identify the connection and maintain per-connection information. As input parameters you have to provide

the host and port of the LDAP server you want to connect to. The `ldap_init()` function returns a pointer to an LDAP structure, which should be passed to subsequent calls to other LDAP functions such as `ldap_simple_bind_s()` and `ldap_search_s()`.

**Note:** Note that we continue with the `ldap_search_s()` API without binding to the directory. In LDAP V3, when not binding explicitly using one of the bind APIs, such as `ldap_simple_bind_s()`, anonymous access is granted. If you want to search for entries or attributes that are not accessible by anonymous users, you need to include a bind API between the `ldap_init()` and the `ldap_search_s()` API.

### **ldap\_search\_s()**

The `ldap_search_s()` API is used to perform an LDAP search operation. `ldap_search_s()` is a synchronous request. This API requires as the input parameter `ld` the value that was returned by the `ldap_init()` API. The remaining input parameters define the search base, scope of search, search filter, attributes to be returned, and whether you want to have only attribute names or also their values returned. Entries returned from the search (if any) are contained in the `res` parameter. When an LDAP operation completes and the result is obtained as described, a list of `LDAPMessage` structures is returned. This is referred to as the search result chain. A pointer to the first of these structures is returned by `ldap_search_s()` API. However, the results cannot be used in the form returned. They have to be parsed by the corresponding APIs to process the returned entries, their attributes, and the attribute values as depicted in Figure 11-2 on page 467.

### **ldap\_first\_entry()**

In the search example, this API is used to parse results for the first entry received from the synchronous LDAP search function `ldap_search_s()`. As an input parameter you need to provide the LDAP structure that was returned by the `ldap_init()` API and the result `LDAPMessage` structure returned by the `ldap_search_s()` API. The latter value is the pointer to the first entry returned by the search function. The return value is the pointer to the first entry of the search results and is required as an input parameter for the `ldap_first_attribute()` API.

### **ldap\_first\_attribute()**

The `ldap_first_attribute()` API returns the first attribute in an entry. `ldap_first_attribute()` takes the LDAP structure returned by the `ldap_init()` API and an entry returned by `ldap_first_entry()` or `ldap_next_entry()`. In addition it has an output parameter that contains a pointer to an opaque data structure for data encoded with Basic Encoding Rules (BER). This pointer is used in subsequent calls to the `ldap_next_attribute()` API to keep track of the current position. It returns a pointer to a buffer containing the first attribute type in the entry.

### **ldap\_get\_values()**

The `ldap_get_values()` API is used to retrieve attribute values from an LDAP entry as returned by `ldap_first_entry()` or `ldap_next_entry()`. The input parameters for the `ldap_get_values()` API are the pointer to the entry as returned by the `ldap_first_entry()` or `ldap_next_entry()` APIs and the pointer to the buffer containing the attribute type as returned by the `ldap_first_attribute()` or `ldap_next_attribute()` APIs. The `ldap_get_values()` returns a NULL-terminated array of the attribute's values. Remember that an attribute value can contain more than one value, as shown for the `telephoneNumber` attribute in Figure 11-3 on page 468.

### **ldap\_next\_attribute()**

Once you have processed the value(s) of the first attribute, you can use a loop to process the remaining attributes of the current entry. The `ldap_next_attribute()` API takes the LDAP structure returned by the `ldap_init()` API and the entry returned by `ldap_first_entry()` or `ldap_next_entry()`. In addition it has an input/output parameter that contains the pointer that is used to keep track of the current position. For the first time the `ldap_next_attribute()` API is called, the pointer is the one returned by the `ldap_first_attribute()` API. It returns a pointer to a buffer containing the next attribute type in the entry. Processing continues with the `ldap_get_values()` API until a NULL value is received indicating that no more attributes are available in the current entry.

### **ldap\_get\_values()**

The `ldap_get_values()` API is now used to retrieve attribute values from the subsequent attributes returned by the `ldap_next_attribute()` API. The parameter and return values are as described in “`ldap_get_values()`” on page 470.

### **ldap\_next\_entry()**

After the attributes and values of the first entry have been processed, you can process the next entry from the search results using the `ldap_next_entry()` API. As input parameters you need to provide the LDAP structure that was returned by the `ldap_init()` API. The second parameter is the pointer to the entry as returned by the `ldap_first_entry()` or for subsequent calls to the `ldap_next_entry()`

API by the `ldap_next_entry()` API. The return value is the pointer to the next entry of the search results and is required as an input parameter for the `ldap_first_attribute()` and `ldap_next_attribute()` APIs. You can now process the attributes and their values of the next entry as described in “`ldap_first_attribute()`” on page 470. A return value of `NULL` indicates that no more entries are in the search results to be processed.

### **`ldap_unbind_s()`**

After all your entries have been processed you need to unbind from the LDAP server using, as in this example, the `ldap_unbind_s()` API. The API is used to end the connection to the LDAP server and free the resources contained in the LDAP structure that was created by the `ldap_init()` API.

**Note:** Several of the APIs mentioned in this section allocate memory and resources. It is strongly recommended to use APIs, such as `ldap_memfree()`, `ldap_msgfree()`, and `ldap_control_free()`, to free up the allocated resources. The Information Center descriptions for each API contain the names of the APIs to be called to free up allocated resources.

## **11.5 API flow when updating a directory entry**

This section provides an overview of what APIs can be used to perform an update of attributes for an existing entry in the LDAP directory. This example, based on LDAP Version 3 APIs, shows how to perform the update using a non-secure session in synchronous mode. There are also APIs available to initiate an SSL session to the LDAP server. The APIs are documented in the order they have to be used. Figure 11-4 on page 472 shows an overview of the APIs and the order in which they are used.

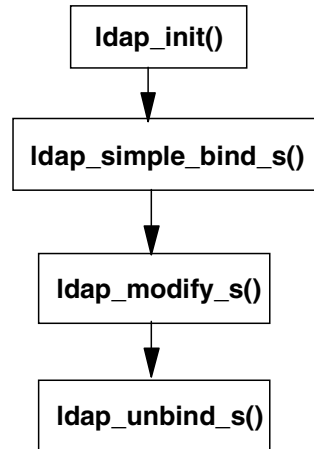


Figure 11-4 API overview for updating a directory entry

In the example shown in Figure 11-4, we want to perform an update of the existing entry with the DN `cn=John McMeeking,ou=employees,o=iseriesshop`. Figure 11-5 shows the current attributes and the new attributes after the update has been performed.

### **Before update**

**Entry:** `cn=John McMeeking,ou=employees,o=iseriesshop`

<b>Attribute:</b> <code>sn</code>	<b>Value:</b> <code>McMeeking</code>
<b>Attribute:</b> <code>givenName</code>	<b>Value:</b> <code>John</code>
<b>Attribute:</b> <code>mail</code>	<b>Value:</b> <code>john@iseries.itso.ral.ibm.com</code>
<b>Attribute:</b> <code>telephoneNumber</code>	<b>Value:</b> <code>919-411-1111</code>
	<code>919-111-0000</code>
	<code>919-111-0002</code>

### **After update**

**Entry:** `cn=John McMeeking,ou=employees,o=iseriesshop`

<b>Attribute:</b> <code>sn</code>	<b>Value:</b> <code>McMeeking</code>
<b>Attribute:</b> <code>givenName</code>	<b>Value:</b> <code>John</code>
<b>Attribute:</b> <code>telephoneNumber</code>	<b>Value:</b> <code>919-555-1111</code>
	<code>919-555-0000</code>
	<code>919-555-0002</code>
<b>Attribute:</b> <code>roomNumber</code>	<b>Value:</b> <code>D333</code>

Figure 11-5 Directory update

As shown in Figure 11-5 on page 472, the last name (sn) and first name (givenName) attributes remain unchanged. The telephone numbers are changed, the e-mail (mail) address has been deleted, and the office number (roomNumber) was added. The following descriptions contain an overview of each API that is involved in the update process. For a detailed explanation about the APIs including their parameters and return values, refer to the *LDAP API Listing* found in the iSeries Information Center by clicking **Programming -> CL and APIs -> APIs -> APIs by category -> Directory Services** or the IBM SecureWay Directory Client SDK Programming Reference at:

<http://www.ibm.com/software/network/directory/library/>

### **ldap\_init()**

The `ldap_init()` API initializes a session with an LDAP server. The server is not actually contacted until an operation is performed that requires it, allowing various options to be set after initialization, but before actually contacting the host. It allocates an LDAP structure that is used to identify the connection and maintain per-connection information. As input parameters you have to provide the host and port of the LDAP server you want to connect to. The `ldap_init()` function returns a pointer to an LDAP structure, which should be passed to subsequent calls to other LDAP functions such as `ldap_simple_bind_s()` and `ldap_modify_s()`.

### **ldap\_simple\_bind\_s()**

The `ldap_simple_bind_s()` function is used to authenticate a distinguished name (DN) to a directory server. There are other APIs available to authenticate users with a different authentication method. With LDAP Version 3 you can skip the bind API allowing you to connect anonymously to the directory server. However, for updating directory entries it is very unlikely that anonymous users can perform any modification. The `ldap_simple_bind_s()` API requires as the input parameter `ld` the LDAP structure as returned by the `ldap_init()` API. In addition you need to provide the distinguished name (DN) of the entry you want to bind to the LDAP server and its password. The DN used has to have the authorities to perform the intended changes. The return code of this API indicates a successful bind or another error code. Error codes are described in the *LDAP API Listing* found in the iSeries Information Center by clicking **Programming -> CL and APIs -> APIs -> APIs by category -> Directory Services** under LDAP Client API Error Conditions.

### **ldap\_modify\_s()**

The `ldap_modify_s()` API is a synchronous API that can be used to add, replace, and delete attributes from an existing entry. It takes several input parameters. The first one is the LDAP structure as returned by the `ldap_init()` API. The second parameter is the DN of the entry to be changed. It is not the DN used for the

ldap\_simple\_bind\_s() API unless you authenticate with the DN you want to change. The third parameter, named mods, is more complex. It is a NULL-terminated array of modifications to be performed to the entry. Each element of the mods array is a pointer to an LDAPMod structure. In regards to the changes described in Figure 11-5 on page 472 you need to create three LDAPMod structure elements.

Element 1 (removes the e-mail address):

<b>mod_op</b>	Set to 0x01 (LDAP_MOD_DELETE).
<b>mod_type</b>	Specifies the name of the attribute. In this case mail.
<b>mod_vals</b>	The mod_vals field specifies a pointer to a NULL-terminated array of values to add, modify, or delete. Since this element is supposed to delete the mail attribute, mod_vals is set to NULL. You can also set the pointer to point to a specific value to be removed.

Element 2 (changes the telephone numbers):

<b>mod_op</b>	Set to 0x02 (LDAP_MOD_REPLACE).
<b>mod_type</b>	Specifies the name of the attribute. In this case it is telephoneNumber.
<b>mod_vals</b>	The mod_vals field specifies a pointer to a NULL-terminated array of values to add, modify, or delete. In this case the pointer points to an array with four elements. The first three elements contain the individual telephone numbers. The fourth element is a null pointer.

Element 3 (adds the office number):

<b>mod_op</b>	Set to 0x00 (LDAP_MOD_ADD).
<b>mod_type</b>	Specifies the name of the attribute. In this case it is roomNumber.
<b>mod_vals</b>	The mod_vals field specifies a pointer to a NULL-terminated array of values to add, modify, or delete. In this case the pointer points to an array with two elements. The first element contains the office number and the second a null pointer.

You do not need to add an LDAPMod element for last name (sn) and first name (givenName) attributes, as they remain unchanged. All modifications are performed in the order in which they are listed.

The return value of the ldap\_modify\_s() API indicates whether the modification was successful or not.



## **ldap\_unbind\_s()**

After the entry has been updated you need to unbind from the LDAP server using, as in this example, the `ldap_unbind_s()` API. This API is used to end the connection to the LDAP server and free the resources contained in the LDAP structure that was created by the `ldap_init()` API.

Archived



## Using the JNDI to search and update the directory

Any Java application, whether it is a servlet, a server application, or a client application, can be directory-enabled. You can exploit LDAP directory information, for example, for automatically addressing payment slips, retrieving user information at a user help desk, and performing your own application authentication. You can even serialize Java objects, such as GUI elements, into an LDAP directory and dynamically load them by all Java applications. The advantage of this method is, for example, that corporate-wide GUI design requirements can be deployed and changed very easily without recompiling programs or even touching the Java programs. The Java package that allows you to directory-enable your applications is the Java Naming and Directory Interface (JNDI) developed by Sun Microsystems, Inc. There are also other Java LDAP clients available, for example the Java LDAP client from OpenLDAP (<http://www.openldap.org>). This client is written directly to the LDAP protocol. Use of asynchronous operations, as required to use the iSeries transaction support, is an example of a function that is not available using JNDI. This chapter shows you, based on a sample application, how to use the JNDI interface. However, it does not provide a complete description of the package and its included classes. For the most current information, as well as a comprehensive *Tips for LDAP Users* section, refer to the following Web page:

<http://java.sun.com/products/jndi/>

## 12.1 The JNDI

JNDI, defined by Sun Microsystems, Inc., provides naming and directory functionality to Java programs. JNDI is an API independent of any specific directory service implementation. It enables seamless access to directory objects through multiple naming facilities.

The definition prevents, by design, the appearance of any implementation-specific artifacts in the API. The unified API is designed to cover the commoncase. Providing this unified interface does not imply that access of unique features of a particular service, such as LDAP, is precluded; additional classes can be added to access service-unique features. JNDI can be used by a wide range of Java programs running on servers and traditional clients. JNDI can also accommodate a thin client by specifying a service provider that provides a proxy-style protocol where access to specific naming and directory services is relegated to a server. Security is dealt with by individual service providers; however, security-related problems can be returned to the client.

As discussed above, JNDI provides a generalized naming and directory service interface. For example, JNDI could be used to retrieve files from a file system. In this case, a file system acting as a naming service could return the file that is bound to a particular file name. JNDI could also be used to access an LDAP directory, performing searches and retrieving attributes.

JNDI provides an API that applications use to access a naming and directory service. The naming and directory service could be provided by any of a variety of servers, such as LDAP, NDS, or a file system. JNDI provides a Service Provider Interface (SPI) that enables access to the particular underlying directory service.

JNDI provides classes that implement a naming interface for applications, such as the file system example, that only look up names and access objects bound to names. JNDI also provides a directory interface that extends the naming interface. The directory interface adds functionality to access attributes and schema.

In JNDI terminology, a name is made up of individual components called atomic names that correspond to RDNs in LDAP. A sequence of atomic names is a compound name. An LDAP DN is a compound name. Since the underlying naming and directory services can have different name syntaxes, the SPI provides an implementation of a NameParser that can break a name into its component parts. For example, LDAP RDNs are separated by commas; DNS

names are separated by periods, and so on. Composite names are compound names that span different name spaces. For example, an LDAP URL can contain both a DNS and an LDAP name, as, for instance, in `ldap://ldap.mycompany.com/cn=John%20Smith,ou=employees,o=iseriesshop`.

Names are interpreted within a context. A context can be thought of as a particular node in the Directory Information Tree (DIT). If the current context is `o=iseriesshop`, then the atomic name `ou=employees` refers to the child node in the DIT with the DN `ou=employees,o=iseriesshop`. The node `ou=employees,o=iseriesshop` is also called a subcontext of `o=iseriesshop`. A name space is traversed from context to subcontext like a file system is traversed from a directory to the directory subtree.

The `DirContext` interface extends the `Context` interface by adding operations specific to a directory service such as accessing attributes and searching. An application must establish an initial directory context as a starting point from which to do searches or traverse the DIT. The initial directory context is usually the name of an LDAP server.

JNDI does provide a mechanism for using extended operations and extended responses, and provides some implementations of these, for example, the `StartTLS` operation. Searches use a search filter as defined in *The String Representation of LDAP Search Filters* (RFC 2254). A `SearchControls` object passed to the search method can be set to control search characteristics such as the scope of the search, the number of entries returned, the time limit, and so on. Also, the entire schema name space can be browsed, and object and attribute schema definitions can be retrieved.

When a directory context is established, it is passed to an environment that contains preferences and controls to access the directory service. The environment specifies the SPI to use, the security level for binding to the server, and so on. The environment is a `Hashtable` or `Properties` list of (key, value) pairs. The environment settings could be coded in the application, retrieved from the system properties, or retrieved from a file. Table 12-1 lists some of the important environment properties. Different SPIs may support other environment properties and interpret or support values differently.

Table 12-1 JNDI directory context environment properties

Environment property	Description
<code>java.naming.factory.initial</code>	Contains the class name of the initial context factory. The property value should be the fully-qualified class name of the factory class that is being used to create an initial context.

Environment property	Description
java.naming.provider.url	LDAP URL that specifies the LDAP server.
java.naming.ldap.version	Specifies if server supports LDAP Version 2 or 3.
java.naming.referral	Specifies if referrals should be followed, ignored, or thrown an exception.
java.naming.security.authentication	Authentication method used to bind to LDAP server: none, simple, or strong.
java.naming.security.principal	Identity of user to authenticate.
java.naming.security.credentials	Password or other security credential.
java.naming.security.protocol	Specifies whether the connection to the LDAP server is secure (SSL).

## 12.2 Scenario characteristics and objectives

As described in the redbook scenario in Section 3.1.4, “Stage 5 - The enterprise directory” on page 48, LDAP Directory Services have become an integral part of the iSeries Shop application infrastructure. Several applications already use the information stored in the company’s LDAP directory for authentication, configuration, e-mail addressing, and so forth. In the past, all employee information, such as their e-mail addresses, phone numbers, or office numbers, was maintained by a member of the Human Resources Department (HR). Due to vacation and other reasons, the update requests were not always processed right away, causing directory information to not be up-to-date.

A decision has been made to allow employees to update their personal information in the company’s directory. However, an employee should only be able to change the following information (the directory schema attribute name is listed in parenthesis):

- ▶ Description of the employee’s responsibilities (description)
- ▶ Telephone number (telephoneNumber)
- ▶ Secretary information (secretary)
- ▶ Middle name (middleName)
- ▶ Job title (title)
- ▶ Office number (roomNumber)

All the remaining information of an employee's entry should not be changeable by the employee. In addition to the information changeable by the employee, the manager of the employee should be allowed to change the following information:

- ▶ Employee's user ID (uid)
- ▶ Department title (departmentNumber)
- ▶ Employee's first name (givenName)

In addition to the attributes that can be changed by the employee and his manager, members of the HR department are allowed to change the remaining attributes:

- ▶ Employee's last name (sn)
- ▶ Employee's manager information (manager)
- ▶ Employee's serial number (employeeNumber)
- ▶ Employee's worker type (employeeType)
- ▶ Employee's e-mail address (mail)  
(Note that the e-mail address is used as a unique identifier for an employee.)

As directory maintenance should not involve a new application on the client side, the IT Department has decided to provide a Web browser interface to keep employee's directory information current. The application runs as a servlet under the WebSphere Application Server on their iSeries server RALYAS4B.

## 12.3 Sample application environment

The sample application, as described in this chapter, requires the following software components:

- ▶ 5722-SS1 - OS/400 operating system Version 5 Release 1 including option 30  
PTF level C1302510
- ▶ 5722-DG1 - HTTP Server for iSeries
- ▶ 5722-WDS - WebSphere Development ToolSet
- ▶ 5722-JV1 - Java Developer Kit (option 5 JDK 1.3)
- ▶ 5733-WA4 - WebSphere Application Server Advanced Edition 4.0
- ▶ IBM WebSphere Studio Application Developer (for application development on a workstation)

## 12.4 Application overview

Figure 12-1 shows an overview of the various components and their relationships as used in the sample application.

**Important:** The sample application as described in this chapter is written to help you understand the various tasks involved to use the JNDI to search and update an LDAP-based directory within a Java application. It is written to provide a proof of concept, but does not cover and act on all possible exceptions, nor is it fully tested under all possible circumstances. However, it is a working application that can be used as an example and be extended to build a complete application.

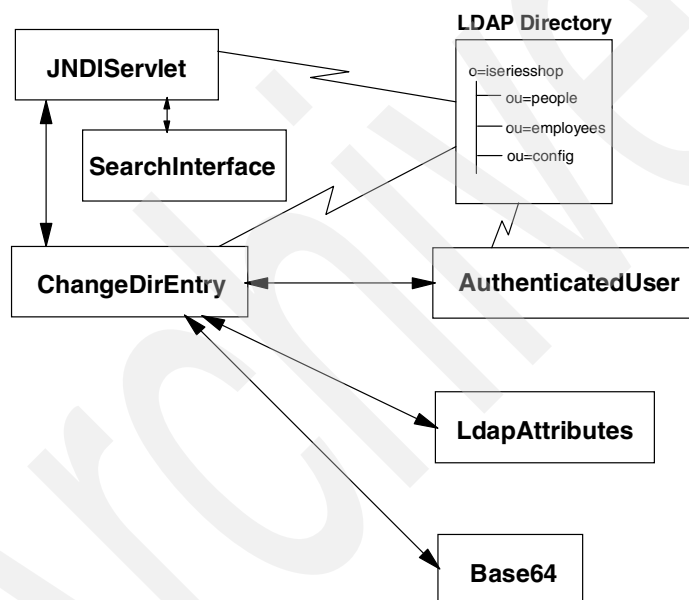


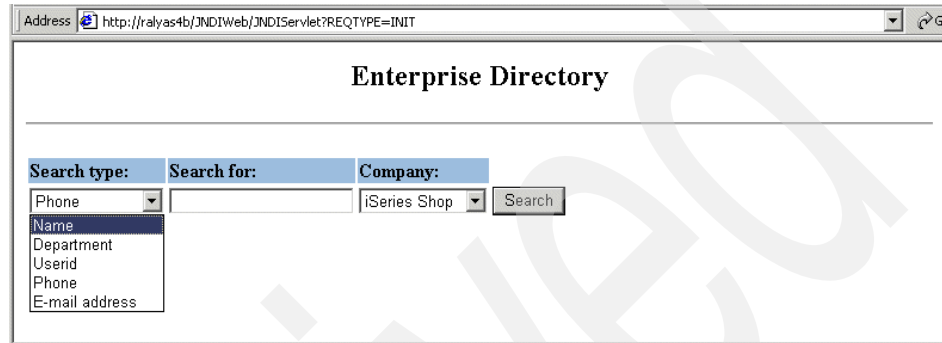
Figure 12-1 Application component overview

The directory browsing and update sample application consists of several servlets and helper classes.



## 12.4.1 JNDIServlet servlet

The JNDIServlet is a servlet called via a URL entered in the browser by a Web user. Based on the request type (parameter REQTYPE) in the URL request, an initial search or an output of the search results are performed. The initial window, as shown in Figure 12-2, is generated through a JavaServer Page (JSP), SearchInterface.jsp.



The screenshot shows a web browser window with the address bar containing the URL `http://italyas4b/JNDIWeb/JNDIServlet?REQTYPE=INIT`. The main content area is titled "Enterprise Directory". Below the title, there is a search interface with three main sections: "Search type:", "Search for:", and "Company:". The "Search type:" section has a dropdown menu currently showing "Phone", with a list of other options: "Name", "Department", "Userid", "Phone", and "E-mail address". The "Search for:" section is an empty text input field. The "Company:" section has a dropdown menu currently showing "iSeries Shop". To the right of these sections is a "Search" button.

Figure 12-2 Sample application initial browser window with sample selections

The user has the choice of searching based on different criteria, such as a person's name, phone number, and e-mail address. The user can select whether he wants to search for employees of iSeries Shop or ACME Supply.

The search, as implemented in the application, always searches with a wildcard character at the end of the search value. For example, if you enter the characters `ba`, all entries that start with the characters `ba` do match and are displayed on the result page. However, names that contain the characters `ba` in the middle or at the end of the value are not a match. If you want to search for all entries with a telephone number that contains the numbers `43` in any position, whether at the beginning, in the middle, or at the end, you could enter `*43` and all results that match the criteria are displayed, as shown in Figure 12-3 on page 484.

Remember that the wildcard character in this application is automatically placed at the end of the search string. In this example, the search string as used in the LDAP search filter is `*43*`.

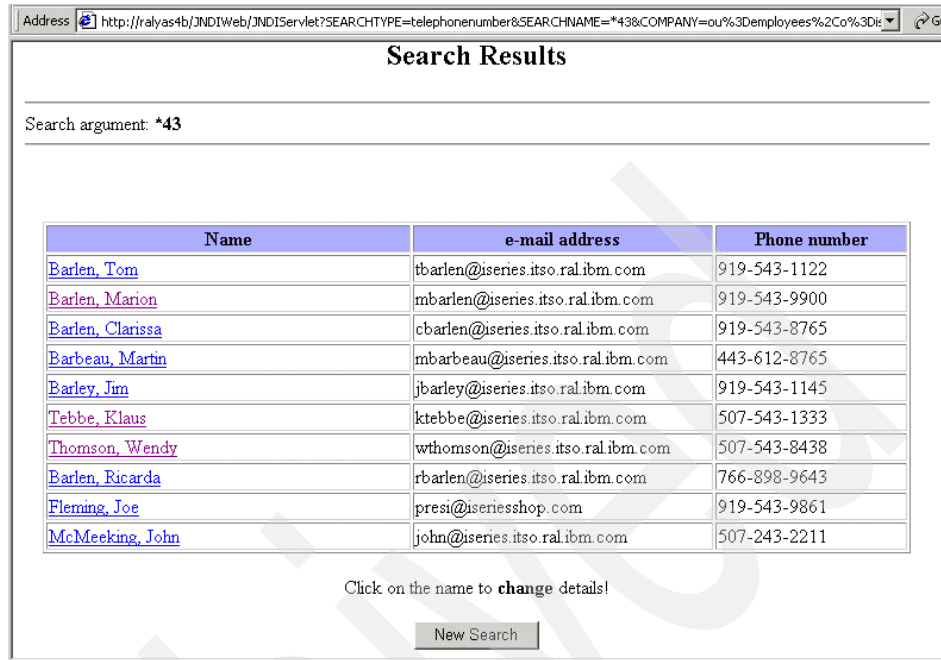


Figure 12-3 Sample application Search Results window

From the list of displayed search results, the user can now click the name of a person to change the entry's details. Typically, you would first display the details and then offer a button to change an entry. However, the purpose of the sample application is to show how to use the JNDI to search and update the directory, and this is what the application does.

The user also has the chance to submit a new search request by clicking the **New Search** button. When the user clicks on a name, the change request will be initiated by calling the `ChangeDirEntry` servlet.

## SearchInterface JSP

The `SearchInterface` JSP is used to generate the initial search browser interface. It is sent to the browser via a request from the `JNDIServlet` when receiving the request type `INIT`.

## 12.4.2 ChangeDirEntry servlet

The ChangeDirEntry servlet is initiated by the JNDIServlet when a user clicks on a person's name in the search result window. When a change is initiated, the e-mail address of the person's entry to be changed and the request type are passed to the servlet. The first task of the ChangeDirEntry servlet is to prompt the user for a username and password. The authentication challenge is initiated by the HTTP header response (www-authenticate). Typically you need to authenticate a user by a unique identifier. In a company this could be the employee number or an e-mail address. When dealing with external users the employee number is not usable anymore. First and last names might also occur more than once. Therefore, we have decided to use the e-mail address of a user as an unique identifier.

Once the user entered the username (e-mail address) and password as shown in Figure 12-4, the servlet decodes the browser input using the Base64 helper class and searches for a person with the provided e-mail address.



Figure 12-4 Browser authentication challenge

The DN of the found entry along with the password is then passed on to the AuthenticatedUser class. Once the user is authenticated and the user type (for example, manager or HR member) is determined, an output method is used to display the attributes of the entry the user has selected to change. The AuthenticatedUser class method hasChangeAuthority() determines whether the attribute to be displayed can be changed by the authenticated user or not. Changeable attributes are displayed as input fields, non-changeable fields as plain text. To prevent the user from returning changes for attributes she is not authorized to in a Get or Post request, all attributes that a user is authorized to change are written into a hashtable. When the changes are submitted, the application performs the changes only for those attributes that are stored in the hashtable. Figure 12-5 on page 486 shows an example of changing details for a directory entry where the authenticated user is the manager of the person's entry.

Address <jeDirEntry?REQTYPE=INITCHANGE&DETAILSFOR=jbarley@iseries.itso.ral.ibm.com&COMPANY=ou=employees,o=iseriesshop> Go

### Details for Barley, Jim

Attribute	Value
Surname	Barley
First name	Jim
Telephone number	919-543-1145
e-mail address	jbarley@iseries.itso.ral.ibm.com
User ID	jbarley
Description	Jim the server sales specialist
Office	507
Department	Sales Support Server
Employee number	+43312
Employee type	Supplemental
Manager	cn=thomas barlen,ou=employees,o=iseriesshop

User authorization is: Authenticated user is manager of the entry to be changed

*Figure 12-5 Changing a directory entry example*

The message at the bottom of the window was added for informational purposes and tells the user what permission he or she has on the selected entry. As described in Section 12.2, “Scenario characteristics and objectives” on page 480, a manager is allowed to change only certain attributes. In this example, the authenticated user is the manager Jim Barley and can perform changes on the attributes displayed as input fields. As previously mentioned, the e-mail address user identifier is converted by the servlet into a DN. To be a manager of an entry, this DN has to be listed in the Manager attribute of the entry to be changed. The button Save Changes allows the user to save the changes to the directory. In case the authenticated user has no authority to perform any changes, the New Search button is displayed instead of the Save Changes button. When the user clicks the Save Changes button, the servlet receives all attributes via the request data stream and saves the changes for the attributes listed in the hashtable as mentioned before. Note that all changes on the directory are performed under the directory’s administrator DN. More information about the permissions of an authenticated user is provided in Section 12.4.3, “AuthenticatedUser class” on page 487, and Section 12.4.4, “LdapAttributes class” on page 487.

### 12.4.3 AuthenticatedUser class

The `AuthenticatedUser` class plays an important role in the sample application. It is instantiated by the `ChangeDirEntry` servlet and takes the DN and password of the user to be authenticated as well as the e-mail address of the entry to be changed.

The first task of this class when instantiated is to set the authentication status. One of the instance variables (`authStatus`) determines whether the authentication using the provided credentials was successful. The class method `isAuthenticated()` returns the authentication status. A second method, called by the `ChangeDirEntry` servlet, `getUserType()`, returns the type of the authenticated user. There are four user types:

- 0 This type indicates that the authenticated user has no authority to change any attribute of the selected entry.
- 1 Type 1 is set when the DN of the authenticated user matches the DN of the entry to be changed. In that case, the user has permissions to change certain attributes of his own entry as described in Section 12.2, “Scenario characteristics and objectives” on page 480.
- 2 This user type specifies that the authenticated user is the manager of the person’s entry to be changed. To be authenticated as a manager for a particular entry, the authenticated user’s DN has to be a member of access group `cn=managers,ou=employees,o=iserieshop` and must be listed in the manager attribute of the entry to be changed.
- 3 This type defines the highest authority. It represents a member of the Human Resources Department (HR). To be considered a member of HR, the authenticated user’s DN has to be a member of the access group `cn=HumanResources,ou=employees,o=iserieshop`. This user type takes precedence over all other types. If an authenticated user is a member of HR, she can change all attributes of an entry.

### 12.4.4 LdapAttributes class

The `LdapAttributes` class serves also as a helper class. Its main purpose is to define what attributes of a directory entry are handled by the application and what authority an individual user has on a particular attribute. What does this mean? Well, a person’s entry might inherit attributes from object classes `inetOrgPerson`, `ePerson`, `organizationalPerson`, and so on. Entries using these

object classes may contain, for example, a user's certificate in binary form used for e-mail encryption. It would not make sense to return the certificate to the application that serves as a telephone directory because there is no practical use for displaying binary data in a browser window. For performance reasons it is also a wise decision to specify in an LDAP search request only the attributes you are interested in. Imagine a person's entry that contains images, audio files, certificates, and so forth. Without limiting the attributes to be returned in a search result you would get a lot of data you do not need at all.

A hashtable `attributeTable` holds information about all attributes that are processed by the sample application. A hashtable entry's name is the name of the attribute and its value (string array) and has the following format:

```
departmentNumber = {"Department", "no", "yes", "yes"}
```

For example, the value for attribute `departmentNumber` consists of four values. The first one provides a descriptive name for the attribute itself and is used by the `JNDIServlet` and `ChangeDirEntry` servlet when displaying attributes to the user. The second value identifies the permission of a user who changes his own directory entry. In this case the user has no authority to change the department he is working in. The third value represents the manager permission for the `departmentNumber` attribute and the last value the permission of a HR Department member. You may notice the relationship between the position of the permission values and the user type as discussed in Section 12.4.3, "AuthenticatedUser class" on page 487. The `hasChangeAuthority()` method (called by `ChangeDirEntry`) takes an attribute name and the user type as input parameters. The user type value matches the position of the permission value within the array. For example, when an `AuthenticatedUser` class returns a user type of 2 (manager) and the `hasChangeAuthority` method is called, the hashtable entry of the attribute name passed to the method is looked up. Then the position as provided by the user type is retrieved from the array that is stored as the hashtable value (remember, the first element in an array starts with 0). If the user has change permissions to the attribute, a true value is returned; otherwise a false value is returned. As mentioned before, all directory changes are performed under the directory administrator.

You might be thinking, why do not they use attribute level permissions to check whether an authenticated user has the proper authority to change an entry's attribute value. Well, security of directory entries, including their attributes, is up to the implementation of the directory on the specific platform. This means that there is currently no standardized way for the JNDI to determine the attribute permissions. Let us assume that you would control all permissions by attribute level permissions and you would perform all changes under the authority of the authenticated user. You would still specify all the attributes you are interested in in your directory search request. Since the JNDI has no way of determining whether the authenticated user has the proper authority to change a certain

attribute before trying the change, an exception would be thrown when the authenticated user has no change authority to an attribute. Unfortunately, the exception does not contain information about the attribute in question. You would just know that you have some authority problem with one of the attributes. Writing an application using this method makes it very hard to handle the error situation properly. Therefore we have chosen to implement our own security mechanism within the application.

Note that we used a hashtable in the `LdapAttributes` class to store all the attribute properties for our security mechanism. A more elegant way to store this kind of information would be to use XML documents. However, this sample application focuses on the JNDI interface and XML is outside the scope of this redbook.

### 12.4.5 Base64 class

The class as used in the sample application was downloaded as free software from the the following Web page:

<http://ostermiller.org/utis/>

The Base64 class is one of the utilities provided in the `utis` package and allows you to encode to and decode from Base64 encoded data. Within the sample application the `ChangeDirServlet` challenges the user to authenticate username (e-mail address) and password. The credentials are returned by the browser in Base64 encoded form. The `ChangeDirEntry` servlet passes the encoded data stream to the `decode` method of the Base64 class, which in turn returns the decoded clear text username and password. The clear text credentials are then used by the `AuthenticatedUser` class to authenticate the user with the LDAP server and to determine the user type.

**Important:** Base64 encoding does not provide encryption. That means that somebody who looks at encoded data cannot easily determine the clear text, but with commonly available utilities, such as the Base64 class, one can decode the data. To provide proper encryption, we recommend performing the user authentication via a secure session (SSL).

### 12.4.6 Obtaining the sample application code

The source for the sample application can be downloaded from the Web. Refer to Appendix F, “Additional material” on page 557 for more information on how to download the application code from the Internet. The `README.HTML` file that is part of the Web material contains details on how to deploy the application.

## 12.5 Searching the directory

Throughout the entire sample, application searches are performed for various tasks including searching for people in the directory, determining the entry's DN for a given e-mail address, and checking for a user's membership in groups. All the searches have to be processed in a certain way. This section explains what you need to do to search a directory using the JNDI interface. The sample application imports the following JNDI packages:

```
javax.naming.*
javax.naming.directory.*
```

### 12.5.1 Creating the directory context

A context can be thought of as a bind, in terms of API calls. The context specifies to which LDAP server to connect, what DN and password to use for the bind, what authentication method to use, and so forth. An instance of the `javax.naming.directory.InitialDirContext` class needs to be created. There are several constructors for this class. However, to properly initialize the context, you need to provide an environment as the parameter. This environment contains the information as mentioned before. Example 12-1 shows the code as used in the `JNDIServlet` servlet to create the environment and the context.

*Example 12-1 Creating a context*

---

```
protected static Hashtable env = new Hashtable();
...
...
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    /** Creating the search controls and context environment for the LDAP
    //Directory (DirContext)
    env.put(Context.INITIAL_CONTEXT_FACTORY,
    "com.sun.jndi.ldap.LdapCtxFactory");
    env.put(Context.PROVIDER_URL, "ldap://" + ldapServerHostname + ":" +
    + ldapServerPort + "/" + request.getParameter("COMPANY"));
    ...
    ...
    try {
        // Create directory context for initial search of people within the LDAP
        // directory
        DirContext ctx = new InitialDirContext(env);
```

---



First you need to define the hashtable env. The next step adds the following entries to the hashtable:

**Context.INITIAL\_CONTEXT\_FACTORY** A constant that stores the name of the environment property for specifying the initial context factory to be used. The value of the property has to be the fully qualified class name of the factory class that will create an initial context.

**Context.PROVIDER\_URL** This constant holds information about the LDAP server address and its port as well as the search base for directory searches. In this case, the server address and port are provided via variables. These variables are defined at the beginning of the JNDIServlet servlet and are used throughout all servlets and classes. You could also use a properties file to provide the information. The last part of the constant represents the search base. In this case, the user interface allows searches for employees of iSeries Shop and ACME Supply. Depending on the selection made on the browser, the search base is filled with either  
ou=employees,o=iseriesshop or  
o=acmesupply.

The JNDIServlet searches for employee information based on criteria provided through the browser interface, as described in Section 12.4.1, “JNDIServlet servlet” on page 483. The search is performed using anonymous access to the LDAP server. Therefore, the environment does not contain any information about user credentials or authentication methods.

In the last step of the example, as shown in Example 12-1 on page 490 the directory context is created using the env hashtable as a parameter for the constructor. The name of the context is later being used as a “handle” to the connection. The context can be used for search, get, and update operations.

## 12.5.2 Performing the search

After the context has been created you can perform a search operation using the new context. The directory context provides several `search()` methods. The methods differ by the type and number of parameters they require, but they all have one thing in common, they return an enumeration of objects. The returned objects are instances of the `SearchResult` class. Example 12-2 shows the `JNDIServlet` code to set up the search parameter and perform the actual search operation.

*Example 12-2 Searching the directory*

---

```
protected static SearchControls ctls = new SearchControls();
...
try {
...
// Defining search controls
    ctls.setSearchScope(SearchControls.SUBTREE_SCOPE);
    ctls.setCountLimit(100);
    ctls.setReturningAttributes(LdapAttributes.getDefinedAttributes());
...
...
// Create search filter based on HTML FORM input and add wildcard character
// at the end of value e.g. (sn=barlen*)
    String searchFilter = "(" + request.getParameter("SEARCHTYPE")
        + "=" + request.getParameter("SEARCHNAME") + "*)";
// Search for objects that have those matching attributes
    NamingEnumeration answer = ctx.search("", searchFilter, ctls);
...
} catch .....
```

---

The sample application uses the following search method:

```
public NamingEnumeration search(String name, String filter,
SearchControls cons) throws NamingException
```

The first parameter of the selected `search()` method specifies the name of the context or object to search. In this example, the parameter is blank. This means that the search is performed under the search base as specified in the context constant `Context.PROVIDER_URL`. You could use this parameter to specify that the search operation begins at a lower level in the directory tree. For example, when we look up employees within iSeries Shop, the search base in the context environment is set to `ou=employees,o=iseriesshop`. If the directory tree had multiple subcontexts for each organization, you could narrow down your search to employees within a specific organization by specifying a parameter for the search. For example, for the manufacturing organizational unit you enter `ou=manufacturing`. The search base for this particular search would then be `ou=manufacturing,ou=employees,o=iseriesshop`.

The second parameter represents the search filter. The filter specifies the search criteria. The JNDIServlet uses a variable that is defined before the search operation. As mentioned in Section 12.4.1, “JNDIServlet servlet” on page 483, the search is performed using a wildcard character in the last position. The string variable `searchFilter` constructs a search filter using the input from the browser via the HTTP servlet request as shown in Figure 12-2 on page 483. For example, if the user selects to search for the name (attribute `sn`) and enters a search value of `ba`, the following search filter is created:

```
(sn=ba*)
```

Using this search filter, all entries that have a last name that starts with the characters `ba` are returned as search results.

More examples and information on search filters can be found in Section 5.6.2, “Searching the directory from a browser” on page 195.

The third parameter defines the search controls. Search controls are used to control the behavior of a search operation. In this example, a search controls object `ctrls` is created. Three search control properties are set. The first one defines the scope of the search operation. A `SUBTREE_SCOPE` specifies that the search operations start at the search base defined in the context environment and the first parameter of the `search()` method and searches through all subcontexts. You could also limit the search to just the context as defined in the search base. The second property defines the maximum number of search results that are returned. A value that is used in many applications is 100. You should always set this value to avoid all sorts of problems. Imagine you have a directory with 200,000 entries, and somebody would search for entries that have a certain e-mail address, and this person would specify the following search value `*@*`. Assuming that all entries contain an e-mail address, the search would return all 200,000 entries because an e-mail address always contains the `@` sign. In this case, the client application might not be designed or have the required resources to handle all the responses. The last property used in the sample application specifies the attributes that will be returned as part of the search. The attributes need to be provided in a string array. The `SearchControls` class documentation contains further information on all available properties you can define.

The search method returns, for each directory entry found, a separate instance of `SearchResult` in a `NamingEnumeration` object.

### 12.5.3 Processing the search results

At this point in the application, the search has found entries that match the search filter. The search results are stored in a NamingEnumeration object answer. Each element in the NamingEnumeration object is an instance of the SearchResult class and contains all attributes returned by the search. You need to process your search results in a nested approach, as shown in Example 12-3.

*Example 12-3 Processing the search results*

---

```
//If search results were found process header
SearchResult sr;
if (answer.hasMore()) {
    out.println("<FORM action='/JNDIWeb/JNDIServlet' enctype='text/plain'>");
    out.println("<BR><BR><CENTER><TABLE border='1' width='729'><TBODY><TR>");
    out.println("<TH bgcolor='#aaaaff'>Name</TH>");
    out.println("<TH width='248' bgcolor='#aaaaff'>e-mail address</TH>");
    out.println("<TH width='157' bgcolor='#aaaaff'>Phone number</TH></TR>");

    while (answer.hasMore()) {
        sr = (SearchResult) answer.next();
        Attributes ans = sr.getAttributes();
        Attribute name = (Attribute) ans.get("sn");
        Attribute givenName = (Attribute) ans.get("givenName");
        Attribute email = (Attribute) ans.get("mail");
        Attribute phone = (Attribute) ans.get("telephonenumber");

        outputResultRow(out, request.getParameter("COMPANY"),
            ((String) name.get() + ", " + (String) givenName.get()),
            (String) email.get(), (String) phone.get());
    }
    out.println("</TBODY></TABLE><BR>");
    outputSearchResultButtons(out);
    out.println("</CENTER></FORM>");

    ctx.close();

} else { // end if block (whether search results were found)
    errorMessage = ERR_NO_DATA_FOUND;
    response.sendRedirect("JNDIServlet?REQTYPE=INIT");
    break;
} // end else

...

```

---

As shown in Example 12-3 on page 494 the program checks first if the search returned at least one search result by using the `hasMore()` method of the `NamingEnumeration` class. If no results were returned, a `No data found` message is returned to the user; otherwise processing continues creating the HTML header elements.

The *while* loop processes as long as there are more elements in the `NamingEnumeration` object `answer`. Within the while loop, the `NamingEnumeration next()` method is used to retrieve the next element and cast it to a `SearchResult` object `sr`. The `sr` object holds all attributes of the directory entry in the object class `Attributes`. Using the `getAttributes()` method of the `SearchResult` class, the attributes are retrieved from the search result and stored in the `Attributes` `ans` object. As shown in Figure 12-3, the only attributes that are listed in the browser interface are the last name, first name, e-mail address, and phone number. Each of these attributes is retrieved via the `get()` method from the `ans` object, casted into and stored in an `Attribute` class object. The call to the `outputResultRow()` method of the `JNDIServlet` generates a row in the output table. To be able to display the value of the attributes, the `Attribute get()` method is used to retrieve the value of each attribute and cast it into a string object.

After all entries have been processed, the HTML code is completed and the directory context `ctx` is closed.

## 12.6 Changing a directory entry

Adding, modifying, or deleting attributes or entire entries also requires creating a context. However, there are different methods for creating or deleting entire directory entries or, in LDAP terms, subcontexts, and for adding, modifying, or deleting attributes for an individual entry. The context's `createSubcontext()` method is used to create a new entry and the `destroySubcontext()` method to remove or delete an entry. The `modifyAttributes()` method is used to add, modify, and delete attributes for a directory entry or subcontext. The sample application shows only the more complex task of modifying attributes. The `ChangeDirEntry` servlet first reads all attributes of an entry to be changed and later saves the changes into the directory. Saving in this place is a misnomer. Modifying existing entries means actually replacing them. This section describes the important parts of the code for creating the context, getting all attributes of the entry to be changed, and performing the update on the selected entry.

## 12.6.1 Creating the directory context

Also for the update of attributes, an instance of the `javax.naming.directory.InitialDirContext` class needs to be created. In addition to the information as mentioned before when searching the directory, the environment for updating attributes contains authentication information. Typically only read, search, and compare operations are allowed when using anonymous access. Therefore, you need to provide information about the authentication method to be used as well as the credentials of the user DN to be used to perform the update. Example 12-4 shows the code as used in the `ChangeDirEntry` servlet to create the environment and the context.

*Example 12-4 Creating the directory context for updates*

---

```
// If user that performs the change is successfully authenticated, proceed with
// change request
if (authUser.isAuthenticated()) {
    Hashtable env1 = new Hashtable();
    env1.put(Context.SECURITY_PRINCIPAL, JNDIServlet.adminDn);
    env1.put(Context.SECURITY_CREDENTIALS, JNDIServlet.adminPw);
    env1.put(Context.INITIAL_CONTEXT_FACTORY,
        "com.sun.jndi.ldap.LdapCtxFactory");
    // environment PROVIDER_URL needs to be updated to remote search base,
    //since we search for a fully
    // qualified DN from the root
    env1.put(Context.PROVIDER_URL, "ldap://" + JNDIServlet.ldapServerHostname
        + ":" + JNDIServlet.ldapServerPort + "/");
    definedAttributes = LdapAttributes.getDefinedAttributes();

    try {
        ctx = new InitialDirContext(env1);
    } catch (Exception e) {
        System.out.println(
            "Exception creating directory context within constructor. Details: "
            + e.toString());
    }
    ....
    ....
}
```

---

First you need to define the hashtable `env1`. The next step adds the following entries to the hashtable:

### **Context.INITIAL\_CONTEXT\_FACTORY**

A constant that stores the name of the environment property for specifying the initial context factory to be used. The value of the property has to be the fully-qualified

class name of the factory class that will create an initial context.

#### **Context.PROVIDER\_URL**

This constant holds information about the LDAP server address and its port as well as the search base for directory searches. In this case, the server address and port are provided via variables. These variables are defined at the beginning of the JNDIServlet servlet and are used throughout all servlets and classes. You could also use a properties file to provide the information. The last part of the constant represents the search base. In this case, there is no search base specified as we already know the fully-qualified DN of the entry to be changed.

#### **Context.SECURITY\_AUTHENTICATION**

This constant specifies what kind of authentication is being used when binding to the directory server. The possible values depend on the service provider that is used. In this case, we used the Sun JNDI interface, which supports the authentication mechanisms none, simple, and strong. Other service providers, such as IBM's JNDI might also support SASL or other values. The simple authentication method uses DN's and passwords in clear text for authentication.

#### **Context.SECURITY\_PRINCIPAL**

This constant defines the DN to be used for authentication. As mentioned before, the sample applications perform all changes using the directory administrator DN. The ChangeDirEntry servlet retrieves the value of the administrator DN `cn=administrator` from the variable `adminDn` that is defined in the JNDIServlet.

## Context.SECURITY\_CREDENTIALS

The value of this constant represents the password of the DN specified as the principal (user) for the connection. The `ChangeDirEntry` servlet retrieves the value of for the password from the variable `adminPw` that is defined in the `JNDIServlet`.

In the last step of the example as shown in Example 12-4 on page 496 the directory context is created using the `env1` hashtable as a parameter for the constructor. The name of the context is later being used as a “handle” to the connection. The context can be used for search, get, and update operations.

### 12.6.2 Getting the attributes

After the context has been created, the `ChangeDirEntry` servlet uses the `getAttributes()` method of the `DirContext` interface to retrieve all relevant attributes for the directory entry to be changed, as shown in Example 12-5.

*Example 12-5 Getting all attributes*

---

```
protected Attributes attrDnToBeChanged;
...
...
// retrieve all attributes of the entry that was selected to be changed
try {
    attrDnToBeChanged =
        ctx.getAttributes(authUser.getDnToBeChanged(), definedAttributes);
    boolean up = outputEntry(out, authUser.getUserType(),
        authUser.getDnToBeChanged());
} catch (Exception e) {
    System.out.println(
        "Exception search for DN to be changed. Details: " + e.toString());
}
...
...
```

---

The `attrDnToBeChanged` instance of the `Attributes` interface is created to hold all attributes of the directory entry to be changed. The `getAttributes()` method's first parameter specifies the DN from which to retrieve the attributes. In this case, the `authUser` instance of the `AuthenticatedUser` class holds this information. The second parameter defines the attributes to be returned by the get operation.



These are the attributes as defined in the `LdapAttributes` class of the sample application. The `outputEntry()` method of the `ChangeDirEntry` servlet generates the HTML code to allow or disallow updates for a particular attribute, as shown in code extract in Example 12-6.

*Example 12-6 Generating HTML output*

---

```
try {
    // determine whether this user has change or read authority
    if (LdapAttributes.hasChangeAuthority(definedAttributes[i], userType)) {
        valueTable = "<INPUT size='60' type='text' maxlength='70' name='"
            + definedAttributes[i] + "' value='" + attr.get() + "'>";
        // put all attributes the authenticated user is authorized to change in the
        // hashtable for later use
        changeAuthAttr.put(definedAttributes[i], definedAttributes[i]);

    } else
        valueTable = (String) attr.get();
    out.println("<TR><TH align=left> " +
        LdapAttributes.getAttributeDescription(definedAttributes[i])
        + "</TH>");
    out.println("<TH align=left > " + valueTable + "</TH></TR>");
} catch (Exception e) {
    ...
    ...
}
```

---

The `if` statement of the `outputEntry()` method checks whether the authenticated user has the permission to update the attribute, as described in Section 12.4.4, “`LdapAttributes` class” on page 487. If the condition is true, an input field HTML tag is created for the processed attribute. If the condition is false, the text is displayed as read only. Note the `changeAuthAttr` hashtable; all attributes for which the authenticated user has write permission are added to the hashtable.

### 12.6.3 Performing the modification

The last part of the sample application performs the actual update of the attributes as shown in Example 12-7.

*Example 12-7 Processing the Save button*

---

```
// Process request type SAVE_DETAIL. Called by ChangeDirEntry to save the
// changes
if (request.getParameter("REQTYPE").equals("SAVE_DETAIL")) {
    // enum to hold all attribute names the user is authorized to change. This
    // prevents the user to inject changes in the Get request for read only
    // attributes (no permission to change)
    Enumeration changeAuthAttrEnum = changeAuthAttr.elements();
    // define an attribute set that holds all changable attributes with its
    // values
```

```

BasicAttributes changeAttr = new BasicAttributes();
Attribute dummyAttr;
// dummy attribute as required by BasicAttributes put method
String tempString;
// temp variable holding the next element to be processed
// process all attributes
while (changeAuthAttrEnum.hasMoreElements()) {
    tempString = (String) changeAuthAttrEnum.nextElement();
    dummyAttr = changeAttr.put(tempString,
        request.getParameter(tempString));
}
try {
    // Now we try to save the changes to the directory. Only attributes that
    // the authenticated user is authorized to change are sent
    ctx.modifyAttributes(authUser.getDnToBeChanged(),
        DirContext.REPLACE_ATTRIBUTE, //always try to replace the attributes
        changeAttr);
} catch (Exception e) {
    ...
    ...

```

---

First, an Enumeration object `changeAuthAttrEnum` is created. Using the `Hashtable` class method `elements()`, the attributes for which the authenticated user has write permissions and that were previously stored in the hashtable `changeAuthAttr` are stored in the Enumeration object. Using the approach of keeping track of attributes the authenticated user can change, we can prevent malicious unauthorized change requests injected in the get request data stream. We can prevent this because we only process the attributes that are listed in the hashtable. All other attributes, whether changed or not are not, considered for update.

The `BasicAttributes` instance `changeAttr` is a new object that will hold the changed attribute names along with the attribute values. This object needs to be created as a necessary parameter for updating the directory entry.

The `dummyAttr` object is only required to satisfy the `put()` method of the `BasicAttributes` class.

The `tempString` object is a work variable that holds the attribute name of the currently processed attribute. The `put()` method of the `BasicAttributes` object `changeAttr` requires as the first parameter the name of the attribute to be added and as a second parameter the attribute value. The value is retrieved from the HTTP servlet request.

And finally we are ready to perform the update on the directory entry. The `DirContext modifyAttributes()` method allows us to update the directory entry. From the available `modifyAttributes()` methods we selected the following:

```
public void modifyAttributes(String name, int mod_op, Attributes attrs) throws NamingException
```

This first parameter is a string representation of the directory entry's DN to be changed. The second parameter specifies the modification operation. In this example, all attributes will be replaced. Other choices are adding (`ADD_ATTRIBUTE`) or removing (`REMOVE_ATTRIBUTE`) an attribute. The last parameter specifies the object that holds the attributes to be modified. In this case it is the `changeAttr` object.





## Part 5

# Appendixes



## Problem determination

In this appendix we provide some hints and tips that will help you debug directory-related errors. Most of the problems documented were encountered while this redbook was written. However, this is not a complete list of all available tools and tips for isolating and solving directory-related problems.

# OS/400 Directory Services hints and tips

This section provides some hints and tips for isolating directory problems on the iSeries server.

## Traces

To debug OS/400 directory problems, you should always start with checking the QDIRSRV, QGLDPUBA, and QGLDPUBE joblogs. If these joblogs do not reveal any clue what a problem might be, you can start a trace for the Directory Services as to using the following command:

```
TRCTCPAPP APP(*DIRSRV) SET(*ON)
```

After the problem you want to debug has occurred you can stop the trace using the \*OFF option for the SET parameter. A spoolfile is generated containing debug information. The information in the spoolfile is intended for IBM service representatives and IBM development.

## Symptoms

Following are some problems and their solutions, which we encountered when writing this redbook.

- ▶ Symptom: The SSD entries are not published to the LDAP directory server.
  - a. The publishing API that causes the System Distribution Directory to get published is called from the QGLDPUBA job. If this job is not running, it will need to be started in order for LDAP to synchronize with the directory entries. This job and the QGLDPUBE job are normally started with the QSYSWRK subsystem; however, if they are ended, they can be restarted with the following commands:

```
QSYS/SBMJOB CMD(CALL PGM(QSYS/QGLDPUBA)) JOBD(QSYS/QGLDPUBA)
USER(*JOB) SYSLIBL(*SYSVAL) CURLIB(*USRPRF) INLLIBL(*JOB)
MSGQ(*NONE)

QSYS/SBMJOB CMD(CALL PGM(QSYS/QGLDPUBE)) JOBD(QSYS/QGLDPUBE)
USER(*JOB) SYSLIBL(*SYSVAL) CURLIB(*USRPRF) INLLIBL(*JOB)
MSGQ(*NONE)
```
  - b. To manually start SDD publishing, type the following command:

```
CALL PGM(QSYS/QGLDSSDD) PARM(*ALL 'cn=administrator' 'password' 0 0 0)
```



This should be where cn=adminimator is the name that you configured for the directory administrator, and password is the password that you selected for the administrator DN.

**Important:** This API call should only be performed when advised by an IBM service representative.

- c. The Allow Directory Updates check box (in the directory server properties) is not checked. This write protects LDAP, and if there are changes to the SDD they will not get updated.
- Symptom: LDAP replica out of sync with the master.

The LDAP master job QDIRSRV in QSYSWRK queues the updates for the LDAP replica. If the master LDAP server encounters a problem when it is trying to replicate, it will stop replicating. If that happens, the replica will be incomplete.

To solve the problem, check the LDAP directory server joblog (QDIRSRV) on the master and replica site for errors indicating a replication problem.

Figure A-1 shows an example where the joblog on the replica indicates that the master tried to add an entry to the replica that already exists.

```
Additional Message Information

Message ID . . . . . : GLD0401      Severity . . . . . : 10
Message type . . . . . : Diagnostic
Date sent . . . . . : 03/27/02      Time sent . . . . . : 22:39:23

Message . . . . . : Entry already exists.
Cause . . . . . : The Directory Services server 012914/QDIRSRV/QDIRSRV
could
not add the entry 'CN=WENDY,OU=EMPLOYEES,O=ISERIESSHOP' because an entry
already exists with that distinguished name.
Recovery . . . . . : Change the distinguished name or delete the entry with
that name and try the operation again.

Bottom
Press Enter to continue.

F3=Exit F6=Print F9=Display message details F12=Cancel
F21=Select assistance level
```

Figure A-1 Replication problem example

In this case the master keeps on trying to add the new entry to the replica. Of course, it will never succeed because the entry already exists on the replica. These cases should never happen, but in case they do occur, you can use the corresponding LDAP utilities from a workstation or the QShell to solve the problem. The following command could be used to solve the problem as indicated in Figure A-1 on page 507.

```
ldapdelete -h rchasm80 -D cn=thomas -w my5ldap  
cn=wendy,ou=employees,o=iseriesshop
```

In the given example, the replica resides on system RCHASM80. So we need to delete the problem entry from the replica so that the master can add the entry again and proceed with other updates. The -D parameter in this case represents the replication DN as specified on the Replication tab of the Directory Services properties on the replica. You cannot use the administrator DN to perform this task.

As an alternative to manually correcting individual replication problems, you can also perform the initial population of the replica again as described in Section 4.7, "Setting up directory replication" on page 106.

- Symptom: Updates to the LDAP directory are not sent to the iSeries SDD.  
This behavior is correct, as changes to LDAP directory entries are not synchronized back to the System Distribution Directory (SDD).

For further problem determination information refer to *Troubleshooting Directory Services* found in the iSeries Information Center by clicking **Networking -> TCP/IP -> Directory Services -> Troubleshooting Directory Services**.

## Directory Management Tool (DMT) hints and tips

- Symptom: The DMT receives error messages during startup.  
If you start DMT and you receive an error message about not being able to authenticate with your LDAP server (not localhost), as shown in Figure A-2 on page 509, try editing the DMT configuration file (dmt.conf) again, ensuring that there are no extra hidden characters or spaces at the end of each line. Refer to Section 5.2.2, "Connecting to the LDAP server" on page 152 for more information on editing the dmt.conf file.



Figure A-2 Error on DMT startup

- Symptom: Object class violations occur when adding new entries.

One of the most common errors when using the DMT or the LDAP utilities to add or modify directory entries are object class violations. Figure A-3 shows an example of an object class violation when trying to add an entry.

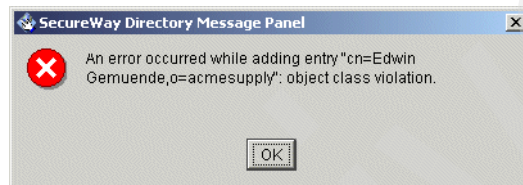


Figure A-3 Object class violation error

In almost all cases the problem is that a required attribute was not specified or values entered did not meet the syntax rules for a particular attribute. There is no simple way of determining which attribute causes the trouble. You need to check the object class and attribute definitions in the directory schema to find out what attributes are required or optional and what syntax is required for the individual attributes. When this error occurs during adding a new entry, we recommend adding the entry with *only* the required attributes. If this works, you can edit the entry and try adding your optional attributes. If the error reoccurs, try adding one attribute after the other until you find the troublemaker.

- Symptom: Java class not found errors occur when using SSL with the DMT.

The DMT as shipped with OS/400 Version 5 Release 1 does not support SSL connections. If you try to set up SSL, and even configured everything correctly, the startup will fail with Java errors because the classes needed for SSL are not included in the JAR files. You need to obtain and install the full version of the tool as described in Section 5.2.3, “Setting up the DMT for using SSL” on page 158.

## Domino problem determination hints and tips

The following are some debug statements for the Domino server regarding Directory Assistance and LDAP.

This first statement `DEBUG_OUTFILE=debug.txt` will have to be set in order to capture any of the following output of the debug statements:

To debug Directory Assistance (DA) you can add the following statements to the `notes.ini` file:

- ▶ `DEBUG_NAMELOOKUP=1`
- ▶ `DEBUG_DIRECTORY_ASSISTANCE=1`

As these two debug statements will give a lot of output, and not all of it will pertain to name lookups performed by the router and mailer processes. It is recommended that these parameters be run separately to aid in easier viewing of the captured data.

For more information about the output of these two statements, please see the the Lotus Knowledgebase Technote No. 178043.

Additionally, the parameter `DEBUG_DIRECTORY_ASSISTANCE=1` dumps the contents of a possibly configured LDAP server referral in the Directory Assistance database. This output looks like the following in the debug out file:

```
DirAssist: Dumping contents of LDAPServer struct for
RALYAS4B.ISERIES.ITSO.RAL.IBM.COM ...
```

```
DirAssist:   LDAPPort:           389
DirAssist:   LDAPTimeout:        60
DirAssist:   LDAPMaxEntries:     100
DirAssist:   LDAPDerefAlias:     3
DirAssist:   LDAPBase:           ou=employees,o=iseriesshop
DirAssist:   LDAPSearchURLForNotesClient: 1
DirAssist:   LDAPSearchURLForLDAPClient:  1
DirAssist:   LDAPConnectionType: Normal
DirAssist:   LDAPChaseReferrals:  0
```

## Debugging Domino LDAP

To debug the LDAP task of the Domino server you can use the statement `LDAPDebug` in the `notes.ini` file to gather more debug information about LDAP running on the Domino server. Unfortunately, the `LDAPDebug` parameter has limited documentation, so this will only be a short description of the possible output of this debug statement.

For a better handling of the LDAPDebug output, please use the  
DEBUG\_OUTFILE=debug.txt statement in the notes.ini.

LDAPDebug can accept many different values. LDAPDebug=65535 is the setting that enables all debug fields. Note that there are not 65534 other settings from which to choose; the decimal number corresponds to a group of hexadecimal numbers that LDAPDebug uses internally. The fields that will be explained in the following appear with a setting of 65535.

#### 1. Startup sequence

The first lines in the following output declare that the LDAP subsystem is initializing and list the directories it is serving (through Directory Assistance).

```
03/22/2002 08:57:31 AM LDAP Server: Started
03/22/2002 08:57:32 AM LDAP Server: Serving Directory
/domwebc/data/names.nsf in the Internet Domain
03/22/2002 08:57:33 AM LDAP Server: Serving Directory
/domwebc/data/iSeriesS.nsf in the Internet Domain
03/22/2002 08:57:33 AM LDAP Server: Serving Directory
RALYAS4B.ISERIES.ITSO.RAL.IBM.COM in the Internet Domain
```

From the above entries, we can determine precisely which directories, if any, are being served by the LDAP server. Secondary directories are listed here also; if no secondary directories are listed, either they do not exist or the server is not recognizing them properly in the Directory Assistance database.

The next entries display configuration options for LDAP, as set in the LDAP tab of the Configuration document and the PORTS tab of the Server document. These entries are valuable in determining if the LDAP server is properly configured. The "CIServ" lines refer to the "base class" that the Domino code uses to start all LDAP, HTTP, IMAP, POP3, and SMTP sessions; in this case, CIServ is used to spawn the LDAP Listener task.

```
03/22/2002 08:57:33 AM LDAP Server: Maximum entries returned = Unlimited
03/22/2002 08:57:33 AM LDAP Server: Time limit for search = Unlimited
seconds
03/22/2002 08:57:33 AM LDAP Server: Minimum characters needed for wild
card = 1
03/22/2002 08:57:33 AM LDAP Server: WARNING: Authenticated Users do not
need SSL
03/22/2002 08:57:33 AM LDAP Server: Anonymous access allowed
03/22/2002 08:57:33 AM LDAP Schema: Started loading...
03/22/2002 08:57:39 AM LDAP Schema: Finished loading
03/22/2002 08:57:39.08 AM ~014730:03561:00002-0000019F| LDAP CIServ
CreateListenerTask> Listen on Port 389
03/22/2002 08:57:39.09 AM ~014730:03561:00002-0000019F| LDAP CIServ
CreateListenerTask> Listen on SSL Port 636
03/22/2002 08:57:39.10 AM ~014730:03561:00003-000001A0| LDAP CIServ
ListenerTask> Listener task (Multi-Endpoint) started
```

**Note:** Starting from this point the example output will be modified for better reading and understanding. The timestamp and thread information at the beginning of each ldapdebug output line will be deleted:

```
03/22/2002 08:57:39.10 AM -014730:03561:00003-000001A0| LDAP CIServ  
ListenerTask> Listener task (Multi-Endpoint) started
```

The modified output will look like:

```
LDAP CIServ ListenerTask> Listener task (Multi-Endpoint) started
```

From these entries, we can inspect detailed server configurations. If a user cannot successfully return a query, confirm that Anonymous access allowed is displayed in the log output. Also, if a site is attempting to implement SSL, the above WARNING... entry will be replaced with a confirmation notice that SSL is required. In our case we enabled both SSL and non-SSL access to the LDAP server. If the site is attempting to use a customized schema, read the entries starting with LDAP Schema to determine if the server is having problems interpreting the schema database.

## 2. Connect accept

The following two entries signal a connection attempt and give the IP address of the querying entity:

```
LDAP CIServ Listen> Connection Accepted on Port 389 for Session 0C620001  
LDAP Server: 9.167.40.71 connected
```

These lines are positive proof of an LDAP query's successful connection to the Domino system. As in the case with all other Internet transactions, if these entries do not appear, something is wrong with the communications path that the LDAP query is traveling. If this is the case, perform troubleshooting steps, such as checking firewalls and any other tasks that are running.

## 3. Basic Encoding Rules (BER) Information

Basic Encoding Rules (BER) is the language used to formulate LDAP queries, much like HTML is the language used to formulate Web pages. For the most part, BER is completely transparent to both the user and to the analyst, and is not used for troubleshooting purposes. The following entries appear as the Domino server reads the information from the query and then aligns it for processing.

```
LDAP> BERGetTag State  
LDAP> BERGetLeadingLengthByte State  
LDAP> BERGetNext State  
LDAP> Bind State  
LDAP>   Version: 3  
LDAP>   DN:  
LDAP>   Method: 0x80 (Simple)
```

```

LDAP> Successful bind for Anonymous User
LDAP> Return Result State
LDAP> InitForSearch
LDAP> StateReturnResult returning resultCode 0 (Success)
LDAP> SendBufferFree
LDAP> BERGetTag State
LDAP> BERGetLeadingLengthByte State
LDAP> BERGetNext State
LDAP> Search State

```

If an error occurs at this point, there may be a problem in the way in which the client is communicating with the server. Several corporations implement their own LDAP clients or solutions, and often program them in-house. For the most part, the BER language is completely cross-platform and universal, and should not be a major sticking point in implementing a client. However, if an error does occur, it may be wise to investigate precisely which LDAP client is being used.

#### 4. Filter Information

This section of filter information, not normally used for analyst troubleshooting purposes, shows the individual functions called by Domino to process the LDAP filter used in the query. The filter itself is shown later in the LDAPDebug output.

```

LDAP> begin LDAPget_filter
LDAP> SUBSTRINGS
LDAP> begin get_substring_filter
LDAP> ANY
LDAP> end get_substring_filter
LDAP> end LDAPget_filter 0

```

#### 5. Query information

This section of query information has several subsections. First, we see the information that the query passes to the server in terms of the subject, the time and size limits, the method of the query, and the attributes that the query is seeking.

```

LDAP> Scope: SUBTREE
LDAP> Dereference Aliases: 0
LDAP> TimeLimit: 0
LDAP> SizeLimit: 100
LDAP> Attributes to return:
LDAP> cn
LDAP> mail
LDAP> o
LDAP> telephonenumber

```

```
LDAP> 1
LDAP> nickname
LDAP> sn
LDAP> givenname
```

Attributes are extremely important regarding how the LDAP query is answered. For example, the above attributes were requested by a Netscape client. When a Notes R5 client requests a query from a server, however, the attributes requested are listed as *\*ALL\**, which the server responds to with all of the fields that the LDAP query asked for, plus other information, such as the Notes public certificate.

Variation in client behavior can be a major problem for sites that are attempting to ensure that LDAP queries return a standard suite of information. Many corporations do not realize that different clients (and for that matter different servers) return different attributes in their default states. If a site relies on a particular attribute that is asked for by default in one client, it must realize that that attribute may not be asked for by default from another client, and must plan accordingly. This section can be invaluable in showing precisely what is being asked for by any particular search.

After the attributes, the query passes the base and the filter used for the search, which can have a profound impact on the query. The base specifies where in the LDAP directory tree to search; the filter is the Boolean listing of the instructions that the query has given the LDAP server. In our example we used no base and the simple search *cn=\*Klaus\**.

```
LDAP> Base:
LDAP> Filter: (cn=Klaus*)
```

A common problem that can occur while making LDAP queries is having an incorrect base. For example, a user may be defined in the LDAP directory tree under the organizational unit */France/World*, but if the LDAP query submitted for the user's name has a base of */Sweden/World*, the query will be unsuccessful.

## 6. Search/retrieval information

After the LDAP query has been processed, the server launches the search for information through all available directories. The *\$LDAPCN* (common name), *\$LDAPS* (surname), and *\$LDAPG* (given name) views are opened, and based on the construction of the filter, information is passed to these views and searched. In our example we only searched for Klaus, and therefore the output looks like this:

```
LDAP> Searching in database /domwebc/data/names.nsf ...
LDAP> Type of search: View Search
LDAP> ... Opening ($LDAPCN)
LDAP> ... Searching entries for a filter 'cn = Klaus*' in ($LDAPCN)
LDAP> ... Opening ($LDAPS)
```



At this point, the actual melding of LDAP and Domino Directory technology takes place. All of the information passed to the Domino server through the LDAP search filter is broken down and passed to these views, which contain all of the unique name elements in the Domino Directory converted into an LDAP-recognizable form. If there is any corruption or problem with these views, normally an error will be displayed here in the log output.

If a result is matched, the server pulls all the available information on the entry that has been requested from the query. Even though the query may have listed dozens of fields, the Domino server will return only that information that (1) has a meaningful value, and (2) has been set to be available to the access level of the querier. (Anonymous queries have specific fields that are available; this availability can be modified by the Domino Administrator.) Next, the server sends the entry to the querier.

```
LDAP> GetSearchEntry State
LDAP> Found matching entry, Note ID: 3670
LDAP> => acl_get: entry (CN=Klaus Tebbe/O=ACMESupply) attr
(InternetAddress)
LDAP> <= acl_get: no match
LDAP> <= acl: denied by default (no matching to)
LDAP> => acl_get: entry (CN=Klaus Tebbe/O=ACMESupply) attr (MailAddress)
LDAP> <= acl_get: backend acl #2
LDAP> <= acl: matched by clause #1 access granted
0 LDAP> => acl_get: entry (CN=Klaus Tebbe/O=ACMESupply) attr (o)
LDAP> <= acl_get: no match
LDAP> <= acl: denied by default (no matching to)
LDAP> => acl_get: entry (CN=Klaus Tebbe/O=ACMESupply) attr
(OfficePhoneNumber)
LDAP> <= acl_get: no match
LDAP> <= acl: denied by default (no matching to)
LDAP> => acl_get: entry (CN=Klaus Tebbe/O=ACMESupply) attr (OfficeCity)
LDAP> <= acl_get: no match
LDAP> <= acl: denied by default (no matching to)
LDAP> => acl_get: entry (CN=Klaus Tebbe/O=ACMESupply) attr (nickname)
LDAP> <= acl_get: no match
LDAP> <= acl: denied by default (no matching to)
LDAP> => acl_get: entry (CN=Klaus Tebbe/O=ACMESupply) attr (LastName)
03/ LDAP> <= acl_get: backend acl #2
LDAP> <= acl: matched by clause #1 access granted
LDAP> => acl_get: entry (CN=Klaus Tebbe/O=ACMESupply) attr (FirstName)
03 LDAP> <= acl_get: backend acl #2
LDAP> <= acl: matched by clause #1 access granted
LDAP> Entry:
LDAP> dn: CN=Klaus Tebbe,O=ACMESupply
LDAP> cn: Klaus Tebbe
LDAP> mail: ktebbe$ralyas4c.iseries.itso.ral.ibm.com
LDAP> sn: Tebbe
LDAP> givenname: Klaus
```

```

LDAP> SendSearchEntry, sending entry CN=Klaus Tebbe,O=ACMESupply
LDAP> => acl_get: entry (CN=Klaus Tebbe,O=ACMESupply) attr (entry)
LDAP> <= acl_get: backend acl #3
LDAP> <= acl: matched by clause #1 access granted
LDAP> <= acl: matched by clause #1 access granted
LDAP> <= acl: matched by clause #1 access granted
LDAP> <= acl: matched by clause #1 access granted

```

The above entries include all of the information that will be sent to the client making the above LDAP query. This section of search/retrieval information provides a very good way to learn precisely which attributes are being returned and to check their validity.

## 7. BER output information

The BER output information section, not normally used for troubleshooting by analysts, displays the actual BER language stream that Domino sends to the requesting LDAP client. This language is similar to the user-readable LDAP distinguished name, but has differences specific to the BER language.

```

30 81 93 02 01 02 64 81 8D 04 1B 43 4E 3D 4B 6C 'O.....d....CN=Kl'
61 75 73 20 54 65 62 62 65 2C 4F 3D 41 43 4D 45 'aus Tebbe,O=ACME'
53 75 70 70 6C 79 30 6E 30 13 04 02 63 6E 31 0D 'SupplyOn0...cn1.'
04 0B 4B 6C 61 75 73 20 54 65 62 62 65 30 32 04 '..Klaus Tebbe02.'
04 6D 61 69 6C 31 2A 04 28 6B 74 65 62 62 65 40 'mail1*(ktebbe$'
72 61 6C 79 61 73 34 63 2E 69 73 65 72 69 65 73 'rallyas4c.iseries'
2E 69 74 73 6F 2E 72 61 6C 2E 69 62 6D 2E 63 6F 'itso.ral.ibm.co'
6D 30 0D 04 02 73 6E 31 07 04 05 54 65 62 62 65 'm0...sn1...Tebbe'
30 14 04 09 67 69 76 65 6E 6E 61 6D 65 31 07 04 'O...givenname1..'
05 4B 6C 61 75 73 'Klaus'

```

## 8. Wrap-up information

Finally, the LDAP server (1) records a successful transaction, and (2) unbinds itself from the LDAP query connection, thereby disconnecting from the remote computer.

```

LDAP> SendBufferFree
LDAP> GetSearchEntry State
LDAP> Entry:
LDAP> Search State
LDAP> Search State
LDAP> Searching in database /domwebc/data/iSeriesS.nsf ...
LDAP> Type of search: View Search
LDAP> ... Opening ($LDAPCN)
LDAP> ... Searching entries for a filter 'cn = Klaus*' in ($LDAPCN)
LDAP> ... Opening ($LDAPS)
LDAP> GetSearchEntry State
LDAP> Entry:
LDAP> Search State
LDAP> Search State

```

```
LDAP> Hits:1
LDAP> Return Result State
LDAP> InitForSearch
LDAP> StateReturnResult returning resultCode 0 (Success)
LDAP> SendBufferFree
LDAP> BERGetTag State
LDAP> BERGetLeadingLengthByte State
LDAP> BERGetNext State
LDAP> UnBind State
03/22/2002 09:26:05 AM LDAP Server: 9.167.40.71 disconnected
```

## Processing referrals

Make sure you have set up the LDAPReferrals entry on the notes.ini file in the following format:

```
LDAPReferrals=n
```

Otherwise, your LDAP searches will receive only one referral. When you specify LDAPReferrals=5 the LDAP server will send up to five referral's information back to the LDAP client. For more information on this, please refer to the Lotus Domino Release 5 product documentation.

## Adding users to an access control list (ACL)

When securing a database with the ACL in Domino the user needs to be entered from the information in your LDAP directory using the following format:

```
cn=wendy/ou=people/o=iseriesshop
```

The syntax here is different from LDAP syntax because a slash (/) is used instead of a comma (,). When a user is added, the format appears different from what you entered; that is, the hierarchy identifiers (cn=, o=, and ou=) disappear. This is expected. In case your directory tree follows a structure like cn=thomas/cn=users/o=iseriesshop, the user information in the ACL remains as entered, because the structure does not follow the typical organization-based Domino structure.

## Domino WebSphere SSO hints and tips

This section describes some problems we encountered when implementing the redbook scenario.

- ▶ When setting up security in WebSphere under the Authentication tab you can also configure whether the WebSphere Application Server will connect to the LDAP server using an SSL connection. The Advanced button allows you to

modify the LDAP search filters. For example, the default for the search filter specifies that the username entered in a browser's authentication prompt is checked against a directory entry that has the username in the uid attribute. In case your unique user identifier is an e-mail address, you would need to modify the search filter to search the username (e-mail address) in the mail attribute of an LDAP directory entry.

- ▶ Any time changes are made in the WebSphere security console the node should be restarted. Especially when the LTPA keys are generated or regenerated. The process is to generate the LTPA keys, then restart the node. Once the node has restarted then you can export the LTPA keys.
- ▶ The WebSphere Version 4.0, Domino Server Application Programming Interface (DSAPI) plug-in for Domino for iSeries is: LIBDOMINOH.SRVPGM
- ▶ The path that needs to be added to the notes.ini file for the WebSphere V4.0 plugin is:

WebSphereInit=/qibm/UserData/WebASAdv4/default/config/plugin-cfg.xml

## IBM HTTP Server for iSeries LDAP hints and tips

This section provides some tips about the LDAP support of the IBM HTTP Server for iSeries.

### Traces

Whether you use the authentication or configuration support of the HTTP Server for iSeries (original) and (powered by Apache), the following command can be used to trace the HTTP server instance:

```
TRCTCPAPP APP(*HTTP) SET(*ON) HTTPSVR(<instance_name>)
```

After the problem you want to debug occurred, stop the trace using the \*OFF option for the SET parameter. Two spoolfiles are created that contain debug information. This is especially useful when you use the configuration support. The dynamically loaded configuration cannot be displayed. The only way to see what directives are loaded from the LDAP directory is by starting a trace and checking the debug statements during the startup of the server instance.

## Configuration support

In some documentation for the HTTP Server for iSeries (original) we read about publishing the HTTP configuration directives into the description attribute of an LDAP entry. Publishing the directives works without any error, but when it comes to using the information the trouble starts. The server instance usually crashed during startup while complaining about wrong or missing directives. When we displayed the attribute values, there were indeed some directives missing.

It turned out that the description attribute cannot be used for this purpose for the following reason. Many attributes do not allow duplicate values. In the case of server directives, there are configuration blocks of directives that always end with the same character. For example, a named protection setup always ends with the character `}`. Since duplicate entries are not allowed, the directory server removes all `}` directives except one rendering the entire configuration unusable. The same goes for the powered by Apache server where directory container always end with the directive `</Directory>`. Even if you do not have duplicate directives, the description attribute does not work. The LDAP standard allows attributes and their values to be returned in any order causing the server directives to be returned out of sequence.

To use the configuration support, you need to store the configuration directives you want to publish into a text file. This text file needs to then be published into an attribute that can hold binary data. You can use the `binProperty` attribute as described in Section 6.4, “Configuration support” on page 254.



## Extending your directory schema

The schema of an LDAP directory defines what object classes are allowed where in the directory, what attributes they must contain, what attributes are optional, and the syntax of each attribute.

For example, a schema could define a person object class. The person schema might require that a person have a surname attribute that is a character string, specify that a person entry can optionally have a telephoneNumber attribute that is a string of numbers with spaces and hyphens, and so on.

This Appendix will show the necessary steps to extend your LDAP schema on the IBM SecureWay LDAP server with an object class needed for the import of Domino for iSeries directory information.

As the extension of a directory schema always depends on the environment and specific situation, there are no general procedures that fit all scenarios. For more information on the IBM SecureWay Directory schema as implemented in iSeries Directory Services, refer to:

<http://www-1.ibm.com/servers/eserver/series/ldap/schema/>

For information about what a schema is refer to Chapter 1, "Directory concepts" on page 3.

## Considerations when extending the schema

The schema used in the IBM LDAP directory on iSeries supports the standard LDAP schema as defined in RFC 2256 (A Summary of the X.500(96) User Schema for use with LDAPv3). This schema possibly meets most of the directory requirements of an enterprise, but there might be the need for an extension of it as new applications or other needs arise after the directory services have been established.

Any given schema can be adapted to the specific needs of your business, but there are a few rules that should be kept in mind:

- ▶ Never delete any element of your standard schema, as this would effect the compatibility of your schema to other LDAP services (servers and clients).
- ▶ Never change the standard schema of your directory by modifying the already existing elements, as this would also effect the compatibility like above.
- ▶ Changing your directory schema should always be an extension of your schema by adding additional elements.
- ▶ When making the extended schema available to other directory-enabled client applications, make sure that you register your Object Identifiers (OIDs). For more information about OIDs and where to find more information about registrations, refer to Chapter 1, “Directory concepts” on page 3.

Therefore you should think twice before you extend or modify your existing schema. And always save your current schema before you start this process.

## Finding the schema definition files on the iSeries server

iSeries Directory Services includes the IBM SecureWay Directory schema. The schema is stored in the Integrated File System (IFS) in the following directory:

`/QIBM/UserData/OS400/DirSrv`

This directory contains all the files that comprise the directory schema. Before and after extending your schema you should back up the IFS directory.



## The redbook scenario - extending the schema

The two companies iSeries Shop and ACME Supply would like to enable the users of each other's company to look up user information like telephone numbers and e-mail addresses. As they cannot set up an online connection between these two companies at this time, they have to exchange their directory information using LDIF files. As iSeries Shop uses iSeries Directory Services (IBM SecureWay Directory) and ACME Supply uses the Domino Directory, we have to adapt the schema information to import the data into the other directory.

We would like to add the person information of the Domino users of ACME Supply to the LDAP directory of the iSeries Shop. As the output of the `ldapsearch` from the domino server includes the object class `dominoPerson`, which is in fact an object class that only exists in the Domino LDAP schema and not in the LDAP schema of the IBM SecureWay LDAP directory on iSeries, we have to add this object class to the schema of the IBM SecureWay LDAP directory on iSeries.

We will only show the extension of the IBM SecureWay LDAP directory schema here. The extension of the Domino Directory is beyond the scope of this book. For more information about extending the Domino directory schema, see *Getting the Most from Your Domino Directory*, SG24-5986.

### Scenario objectives

The objectives of this scenario are:

- ▶ Check for the necessary elements and their definitions to be added into the schema.
- ▶ Add the new elements to the schema of the iSeries LDAP server to allow the import of Domino user data exported from the Domino Directory into LDIF files. In this scenario, the `dominoPerson` object class in the iSeries Directory schema serves just the purpose of importing data in the format as they were exported from the Domino Directory.

### Scenario prerequisites

For the IBM SecureWay LDAP server on iSeries see Chapter 4.2, "Installation prerequisites" on page 62.

For the Domino for iSeries see Chapter 7.2.3, "Installation prerequisites" on page 307.

## Modifying your directory schema

In many cases extending a schema is required when importing directory information that contain object classes or attributes that are not in the directory schema of the importing directory server. You should then follow a two-step process for extending the schema:

1. Obtain the schema definition for the object classes or attributes to be added.
2. Extend the schema on the destination directory server using the definitions obtained from the source server.

### Obtain the schema definition

First you should decide which element you would like to add to the schema. You should set up a description of this new element, describing the functionality of the element.

In our scenario we would like to add the object class `dominoPerson` to the IBM SecureWay LDAP schema. To check for the definition of this object class in the Domino schema you should do the following:

1. On the Domino server console type in `tel1 LDAP exportschema` and press Enter. The LDAP task will export the schema into a database on the server called `schema50.nsf`.
2. Open the database `schema50.nsf` with your Notes client and search for the Domino object class `dominoPerson`.
3. Select the `dominoPerson` object class to display its definition, as shown in Figure B-1.

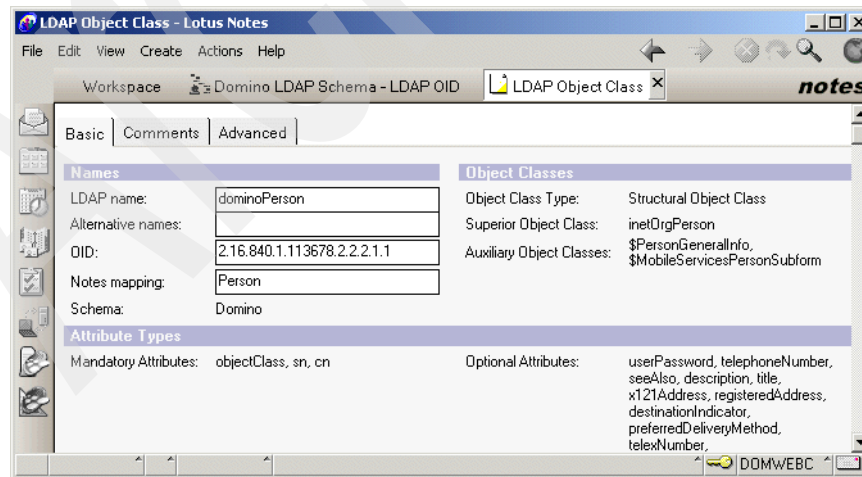


Figure B-1 Domino LDAP object class `dominoPerson`

4. Note the required definitions needed for the generation in the IBM SecureWay Directory, such as:

<b>LDAP name</b>	dominoPerson
<b>OID</b>	2.16.840.1.113678.2.2.2.1.1  This OID is the registered OID for the Lotus Domino Directory Service - LDAP Object Class in the US.
<b>Object class type</b>	Structural Object Class
<b>Superior object class</b>	inetOrgPerson
<b>Mandatory attributes</b>	objectClass, sn, cn

5. For the other object class elements, such as Alternative names, Notes mapping, Auxiliary Object Classes, and the Optional Attributes you have to decide which ones you would like to include for your new object class on the iSeries Directory schema. In our scenario we will not include any of these attributes.

## Extending the iSeries Directory schema

Add the new element to the IBM SecureWay LDAP directory server.

For adding the object class dominoPerson to the schema of the IBM LDAP directory on iSeries, you have to do the following:

1. Start the IBM SecureWay Directory Management Tool and add your LDAP server to the Management Tool.
2. Expand **Schema** and then **Object classes**.
3. Click **Add Object Class** to enter the new object class definition on the General tab, as shown in Figure B-2 on page 526.

<b>Object class name</b>	dominoPerson.
<b>Description</b>	Enter a self-explanatory description of the new object class.
<b>OID</b>	2.16.840.1.113678.2.2.2.1.1.

**OID:** Make sure that you enter the OID after you entered all other fields in the three input windows, because when you enter it in the beginning it will change to the character value dominoPerson-OID.

<b>Superior object class</b>	inetOrgPerson
<b>Object class type</b>	Structural

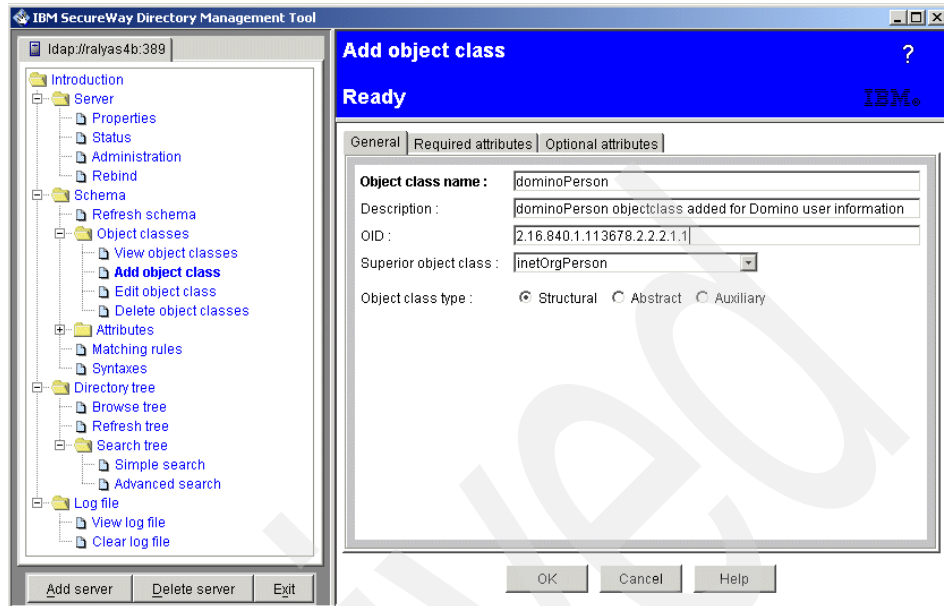


Figure B-2 Add object class window - General tab

4. Check on the Required attributes tab for the already Inherited attributes cn, objectclass, sn. No more attributes will be added here in our scenario.
5. In the Optional attributes window enter any optional attributes if needed. In our scenario we will not enter any optional attributes here.
6. Click **OK** to add the new object class to the directory schema.

To check if the change to the schema was successful, do the following in the DMT.

1. Expand **Schema -> Object classes -> View Object classes**.
2. Click **Search** and enter the object class name, in this case, dominoPerson.
3. Press **OK** and the DMT scrolls to the object class.

## **OS/400 and Domino LDAP history**

The purpose of this apendix is to give you an overview of when different directory functions were intorduced to OS/400 Directory Services and Lotus Domino Directory support. It also gives you a brief summary of future enhancement plans in this area.

## OS/400 Directory Services by release level

Table C-1 summarizes various Directory Services-related functions as they were introduced by release level.

*Table C-1 OS/400 directory service related functions by release*

Release	Function
V3R1	SDD support for POP3 and cc:Mail clients and SMTP addressing.
V4R2	SDD/Domino Directory synchronisation in Domino for AS/400.
V4R3	LDAP server Version 2 (uses DB2). OS/400 LDAP client including ILE application access. CA/400 LDAP client Java client in JNDI CL utilities for server access Windows and 5250 Qshell search, modify, add, and delete utilities Configuration via Operations Navigator. SDD-to-LDAP synchronization. Supports network hardware and software inventory in LDAP. Schemas (person, organisationalPerson, inetOrgPerson object classes). Schemas can be extended.
V4R4	Automated SDD-to-LDAP synchronization via QGLDSSDD API. Prevent SDD records from being published. New ePerson object class supported. QgldChgDirSrvA and QgldRtvDirSrvA APIs have been changed, affecting compatibility. For a further details on Version 4 Release 4 enhancements and changes refer to: <a href="http://www-1.ibm.com/servers/eserver/series/ldap/ldapv4r4.htm">http://www-1.ibm.com/servers/eserver/series/ldap/ldapv4r4.htm</a>
V4R5	LDAP Version 3 support Changes in Operations Navigator (including configuration, ACLs, and NLS data support for UTF-8. Enhanced schema checking options. Dynamic/extensible schema. SASL authentication support. New client APIs. New server change log. LDIF Version 1 supported. UTC and Generalized Time attribute type support. For a further details on Version 4 Release 5 enhancements and changes refer to: <a href="http://www-1.ibm.com/servers/eserver/series/ldap/ldapv4r5.htm">http://www-1.ibm.com/servers/eserver/series/ldap/ldapv4r5.htm</a>

Release	Function
V5R1	<p>IBM SecureWay Version 3.2 support.</p> <p>Printers added to hardware and software inventory.</p> <p>New LDAP APIs.</p> <p>Network authentication via Kerberos Version 5 APIs.</p> <p>The LDAP directory server supports event notification. This allows clients to register with the LDAP server to be notified when a specified event occurs.</p> <p>New transaction support allows users to group LDAP directory operations into units.</p> <p>Access control lists now use an attribute level permissions model. This model allows one to specify authorities for specific attributes rather than specifying authorities by access class.</p> <p>OS/400 security auditing can be used.</p> <p>OS/400 Directory Services included in base operating system. APIs are available from the QSYS library. In Version 5 Release 1, the only reason you need option 32 are the APIs in the library QDIRSRV to support backwards compatibility for Operations Navigator (prior versions).</p> <p>A default configuration is provided allowing you to start Directory Services without prior manual configuration. This simplifies configuration work for inexperienced users.</p>

## The future

Several enhancements are planned to be included in future releases. They include:

- ▶ IBM SecureWay Directory Version 3.2.2 support. New OS/400 releases will be updated to support the newest IBM SecureWay Directory versions.
- ▶ System Projected Backend for user profiles. For example, you can access user profiles using LDAP access.
- ▶ Enterprise Identity Mapping.
- ▶ LDAP administrator access using user profile.
- ▶ Directory Services API calls have to be changed from the QDIRSRV library to QSYS. OS/400 option 32 (Directory Services) removed from the operating system as Directory Services will be completely moved to the base operating system.

# Lotus Domino Directory by release level

The Lotus Domino product has added a lot of new directory functions in recent releases and is firmly expected to continue doing so in the future. This appendix aims to summarize that function by Domino software release level including information for future releases where Lotus has made this available. This list is for guidance and is not intended to be definitive. Reference should be made to Lotus documentation where appropriate.

## Notes server pre-Release 4.6

Almost since the inception of Lotus Notes the Notes Name and Address Book (as the directory was known) has played a major part in the management and security of Notes data. The main directory database is known as `names.nsf`. It contains user registration and e-mail addressing information plus server-specific data. By a process of replication this information is shared across multiple servers.

The Lotus Notes server has since been renamed to Domino and the server's Name and Address Book renamed to the Domino Directory. The title of Name and Address Book is still retained for the user's personal name and address book.

For AS/400, releases previous to Release 4.6 had run under a specific version of OS/2 on the File Server Input Output Processor (FSIOP) and subsequently the Integrated PC Server for AS/400 (IPCS). This was supported from OS/400 Version 3 Release 2 and Version 3 Release 7.

## Domino Release 4.6

Release 4.6 was the release at which the Lotus Notes server was renamed as Domino and the term Lotus Notes was retained to refer to the standard client software. This was to reflect the addition of the Web browser as a client to the Domino server.

Beginning with Domino Release 4.6 we see the first LDAP function in the directory. LDAP Version 2 and the IETF standard person/group schema were supported. However, there was no way to modify the directory structure.



Release 4.6 was also the first release to run natively under OS/400 at OS/400 Version 4 Release 2. The specific version of Lotus Domino for AS/400 included a directory synchronization function for two-way synchronization between the OS/400 System Distribution Directory (SDD) and a Domino directory running on the same physical server. There were, and still are, lots of platform-specific features that make the AS/400, and now the iSeries, the best platform on which to run Domino. However, it is not the purpose of the appendix item to detail them.

For more information refer to <http://www.lotus.com>.

## Domino Release 5

With the major launch of Domino Release 5 came support for LDAP Version 3 including writes from other LDAP clients and servers. The ability to extend the schema and to add new auxiliary structured objects became available. Object class was also now supported in Domino Designer but inheritance was not.

A new Directory Catalog function was announced that provides a compressed subset of directory data for use off-line in Notes. A directory catalog consolidates entries for users, groups, mail-in databases, and resources from one or more Domino directories into a single, lightweight, quick-access database. You can use a user setup profile to create a replica of a directory catalog on Notes clients, known as a mobile directory catalog, so Notes users can quickly address mail to anyone in your organization even when disconnected from the network. Type-ahead addressing searches the mobile directory catalog rather than Domino directories on a server, which reduces network traffic. You can also create a directory catalog for server use so that servers can use one database to search for names in multiple Domino directories. Typically, you create two directory catalogs: A mobile directory catalog and a server directory catalog. A directory catalog can combine entries from many Domino directories and still be very small. For example, several Domino directories that together contain more than 350,000 users and total 3 GB in size, when combined in a directory Catalog, are likely to be only about 50 MB. In general, each entry in the directory catalog is slightly more than 100 bytes. By default, entries in the directory catalog include those fields that are required to resolve mail addresses. You can include other fields that users in your organization access frequently.

Directory Assistance was another new feature introduced at Release that helps manage name lookups in organizations that use multiple Domino directories and/or third-party LDAP directories. A Directory Assistance database associates each Domino Directory/LDAP directory with specific hierarchical names so that when looking up a hierarchical name, Domino first searches the directory that contains names in that hierarchy. Directory Assistance is a feature that enables users to locate information in a directory that is not the primary Domino Directory for that server. Directory Assistance also enables you to authenticate Web clients

by using a directory that is not the primary Domino Directory on the server to which the clients connect. You can configure Directory Assistance for secondary Domino directories and for LDAP directories, for example, for third-party LDAP directories.

## Domino Release 5.0.1

At Release 5.0.1 LDAP schema publication was introduced. Domino automatically builds a database called Domino LDAP schema (SCHEMA50.NSF) in the Domino data directory. This database provides information about the Domino LDAP schema attributes, object classes, and syntax. If you extend the directory schema, you can use the server command **te11 1dap reloadschema** to update the schema in memory and then update SCHEMA50.NSF with your directory changes. If you do not use this command, these two updates are done automatically when the designer task runs or when you quit and restart the LDAP service.

The LDAP search utility (LDAPSEARCH.EXE) also became available.

## Domino Release 5.0.2

At Release 5.0.2 LDAP schema enforcement was introduced. If you enforce schema checking, the LDAP service only carries out LDAP add or modify operations if the information in the operations conform to the directory schema. Enforcing schema checking helps administrators control the content of the directory and helps lay the groundwork for future interoperability with other directories.

To better comply with LDAP standards, the LDAP service also provided enhancements in areas such as street and postal address attribute mapping, new mappings for LDAP name attributes, and changes to object class definitions, plus many LDAP schema changes. The Directory Assistance (DA50.NTF) template now creates a modifiable Group Expansion field in the Directory Assistance document. This enhancement allows administrators to control nested group searches in LDAP queries during Web authentication.

## Domino Release 5.0.3

An option in an LDAP directory Assistance document called Optional Authentication Credential enables a Domino server to present a name and password when connecting to an LDAP directory server. If you do not specify a name and password, the connection occurs anonymously.

## **Domino Release 5.0.4**

At Release 5.0.4 Directory Assistance can now fail over to an alternate replica of a secondary directory without a server restart.

Also, Directory Catalogs came to support the use of selection formulae to use only a subset of documents from secondary directories. Server Directory Catalogs now support full functionality for Release 4.6 clients.

LDAP services were enhanced by providing support of more name formats for members of external LDAP directory groups, providing more flexibility in the storage of telephone numbers, returning the base entry on subtree searches, and LDAP recognition of group membership changes in a Domino Directory. Additional enhancements include the removal of case-sensitivity when searching the mail attribute, as well as returning better results when doing wildcard searches on the mail attribute.

## **Domino Release 5.0.5**

The Extended Directory Catalog combines the advantages of the Domino Directory and the standard Directory Catalog by aggregating entries from multiple Domino directories into a single directory database. This is similar to the construction of a standard Directory Catalog but retains the full set of indexed views and other features of the Domino Directory. This enables the enterprise to maintain a single consolidated server-based directory structure that responds rapidly to a variety of search patterns and can contribute to enhanced mail router performance.

## **Domino Release 5.0.9**

At Release 5.0.9 Off-line Directory Catalogs can be utilized for DOLS-enabled applications for such things as directory lookups and addressing. The feature is available through new settings within the DOLS configuration document.

## **Domino Release 6.0**

Domino Release 6.0 is the name of the release of Domino that was formally announced at US Lotusphere in January 2002 as this redbook was being written, hence the points below are based on informal customer presentations from Lotus rather than detailed technical manuals. For more details please refer to Lotus documentation. This release was formerly known as Domino Rnext.

In general, Domino 6.0 aims to make Domino easy to integrate in a multi-directory environment. It provides an option to move from a distributed directory architecture to the use of Domino as the central directory. At Domino 6 you can store a complete directory on one central server (actually two for availability; failover of directory servers is automatic), and can store smaller configuration directories with Domino-specific data on the other servers in the domain. The centralized directory information is available to all users. There is now no need to replicate the directory across all servers in a domain.

In terms of LDAP support, there is a new schema.nsf to maintain, extend, and publish in an automated process. To do this there is no need for Domino Designer (use Notes or a browser). There is support for object class inheritance (which can come from schema.nsf, names.nsf, and Ischeadmin server of the directory, hence a domain-wide schema can be replicated to every server. There is built-in support for more standard schemas, improved performance of LDAP access and an LDAP upgrade service to migrate person and group entries from LDAP into the directory. There is support of arbitrary distinguished names, new LDAP configuration settings, activity logging for the LDAP service and multiple values in the Hostname field in the LDAP directory Assistance document for LDAP server failover. You can now authenticate Internet clients for IMAP, POP3, LDAP, and NNTP clients, as well as HTTP clients, using a secondary directory (Domino or LDAP).

To provide better integration when working with Windows 2000's Active Directory, a new ADSync process lets you register, synchronize properties and passwords, and rename and delete users and groups in the Domino Directory when you perform such actions in Active Directory.

There is significantly enhanced support for the ASP/Hosting environment. You can now create a multiple organization Directory. This reduces the complexity of server administration. The Domino Administrator works with one server, yet each organization on that server functions as if hosted by its own server, for example its own HTTP application and file locations. Organization-specific authentication controls can be applied and a template allows granular configuration control for each hosted organization. Domino Administrator allows an ASP/hoster to register a new organization, creating the hosted configuration, producing a new certificate, creating a subdirectory, and implementing the security mechanisms (database ACLs, ACL files, and extended ACLs) automatically. Each document in the directory is controlled by an extended ACL (xACL) to allow or disallow access. The existing database ACLs and the new ACL file feature ensure that organization-private databases remain secure. File protection documents for the Domino Web server provide additional access control for files accessed via HTTP. Multiple organizations hosted by the logical server can also access shared databases. ASPs can provide IMAP, POP3, LDAP, SMTP, HTTP, SSL, IIOP, and DOLS.

To provide more granular control, the database server utility programs such as compact, fixup, updall, and design, now allow a directory to be specified. Scalability features for a hosted environment include:

- ▶ Support for a configuration-only directory to improve server performance.
- ▶ Qualified name lookups per organization in the Domino directory to provide improved name lookup performance for any size directory.
- ▶ Support for the use of a network sprayer to provide load balancing or failover capabilities.

Changes at a higher level to the Domino software release rollout plans mean that the scope for individual server platform teams to make major functional changes to their version of the product will now be restricted. This has resulted in the withdrawal of the SDD/Domino Directory Synchronization capability in Domino 6 for AS/400. See Appendix E, “The BlueNotes Product Suite” on page 549, for possible alternative solutions.

## Domino futures

These major updates to directory functions with Domino 6 represent the first stage of the implementation of a major new strategy from Lotus. This strategy has three key elements.

The first is to target the Enterprise Directory market. This is the market for using a single directory platform as the core of an infrastructure. Domino is already a serious player here and these enhancements represent a major advance in positioning.

The second is to provide integration in a multi-directory world, either as a primary or a secondary player. This will be achieved primarily via LDAP standards, and also via DSML Version 2 in the future.

The third is to extend Notes and Domino as directory-enabled applications. This raises some interesting possibilities including running a Notes client infrastructure independently of Domino, and using instead some other LDAP-enabled directory such as SecureWay.

These elements all go beyond Domino 6. Some minor issues within the current strategy are also still to be addressed. For many customers the provision of directory synchronization with LDAP would be a prime requirement and this, we are assured, is firmly in the plans.



## Directory futures and trends

Scores of millions of people access directories every day. Directories are now starting to reach the point of ubiquity. They are appearing in server operating systems, workgroup solutions, Web browsers, and the network infrastructure. We expect this trend to continue. But what else might we expect or hope to see? This appendix identifies some current trends and suggests a few potential new ones. The views expressed here for information and guidance are not necessarily those of the IBM Corporation.

## Dominance of LDAP

LDAP directories continue to dominate the Directory Services arena. Few people would plan to implement a new non-LDAP directory today. There is no sign of an end to this trend, nor do we expect to see one.

Directory products continue to enhance their LDAP capabilities. SecureWay and Domino are in the vanguard of this trend and you should feel comfortable with building an infrastructure upon them secure in the knowledge of their LDAP support. The support of the major application and operating system vendors for LDAP means that it is a de facto standard as well as a published one.

The LDAP standard itself continues to advance, having published LDAP Version 2 as RFCs in 1995, and LDAP Version 3 in 1997, the two current IETF working groups dedicated to furthering the scope of LDAP-related directory functionality. The LDAP Extension (LDAPEXT) working group defines and standardizes extensions to the LDAP Version 3 protocol and extensions to the use of LDAP on the Internet. The planned extensions include the following areas:

- ▶ Access controls
- ▶ Server-side sorting of search results/paged retrieval of search results
- ▶ Language tags
- ▶ Dynamic directories
- ▶ Referral and knowledge reference maintenance
- ▶ LDAP server discovery
- ▶ LDAP APIs
- ▶ CLDAP
- ▶ Signed directory information

Other areas such as deployment and schema definition and review are, or will be, handled by other groups if and when they turn out to be necessary for the deployment of LDAP and feasible for the group to tackle.

The second group, LDUP (LDAP Duplication/Replication/Update Protocols), addresses the issues of replication of data across servers running different implementations, which is becoming an important part of providing a distributed directory service as LDAP Version 3 becomes more widely deployed. The LDAP Version 3 community to date has focused on standardizing the client/server access protocol, and so the LDUP group is chartered to standardize LDAP Version 3-based replication. It is anticipated that emerging consensus on the LDUP work will be taken up by vendors over the next two years.



## Common data definition: DSML

Directories are increasingly storing metadata about available Web services, what they do, what they require for input, how to execute them, what the results will be, who wrote them and how to pay for them. Combined with the power of eXtensible Markup Language (XML), the Internet's common language for e-business, this information enables whole new classes of individually tailored applications for e-commerce. The definition of the XML schema for describing directory structure and data is Directory Services Markup Language (DSML).

Applications consume DSML documents as they would XML because DSML is a subset of XML. Applications can transmit DSML documents to other DSML-enabled applications on the Internet. This process effectively extends LDAP across firewalls to any Internet transport protocol, such as HTTP, FTP, or SMTP, which is a major benefit for business-to-business (B2B) efforts.

DSML is also a major step forward in facilitating interoperability between different vendors' products, by describing their contents in XML. DSML-compliant directories can publish schema information as an XML document which can then be shared by other directories or applications. For example, account information can be maintained across multiple business partners, regardless of the underlying directory structure on each partner's site.

By leveraging the XML/DSML standards, applications can be enabled to react quickly to the needs of business, while leveraging a solid foundation of interoperability with back-end systems. The combination of LDAP and XML provides data in a way that allows easy integration within both new and existing applications; while LDAP provides a means for accessing directory information, DSML provides the means for reading and understanding directory content in XML. So DSML provides a standard for creating XML documents from the information that LDAP delivers.

A group of commercial vendors came together in the DSML Forum to work with the standards bodies to enhance and refine the DSML standard, and to continue development of DSML. At the Lotusphere conference in January 2002 IBM's Lotus division committed to DSML Version 2 as a key standard for the future of the Domino Directory. iSeries Directory Services does not support DSML at this time.

## Directory-enabled applications

A directory-enabled application is an application that uses a directory service to improve its functionality, ease of use, and administration. Today, many applications make use of information that can be stored in a directory. For example, consider a group calendar application that is used to schedule meetings of company personnel in different conference rooms.

In the worst case, the calendar application does not use a directory service at all. If this were the case, a user trying to schedule a meeting would have to remember the room number of every conference room that might be appropriate for the meeting. Is the room big enough, does it have the necessary audio and video equipment, and so on? The user would also have to remember the names and e-mail addresses of every attendee that needs to receive a meeting notice. Such an application would obviously be cumbersome to use.

If conference room information (size, location, special equipment, and so on) and personnel information (name, e-mail address, phone number, and so on) can be accessed from a directory service, the application will be much easier to use. Also, the functionality of the application can be improved. For example, a list of all available conference rooms meeting the size and equipment requirements can be presented to the user. Some types of applications that can benefit from being directory-enabled include line-of-business, document management, middleware (message queuing, distributed transaction processing), traditional client-server applications, and infrastructure products (storage management, systems management).

The IBM SecureWay Directory and Client SDK supports two ways of directory-enabling applications: JNDI and the C API.

The Java Naming and Directory Interface (JNDI) was defined by Sun Microsystems, Inc. It provides naming and directory functionality to Java programs. JNDI is an API independent of any specific directory service implementation. It enables seamless access to directory objects through multiple naming facilities.

The C LDAP API is an API library for C language applications. As the JNDI, it can be used for establishing connections, performing searches, and parsing the results.

Microsoft offers its own API, called Active Directory Services Interface (ADSI).

Today, several leading ERP companies, such as SAP, Baan, and J. D. Edwards, have (or are in the process of) directory-enabled their applications to bring the focus of their development efforts to the business solutions instead of the underlying infrastructure. This is a trend that should, in the short term, spread to other vendors' as well as customers' applications.

## Directory-enabled networks

The Directory Enabled Network (DEN) specification is designed to provide the building blocks for more intelligent networks by mapping users to services, and mapping business criteria to the delivery of network services. This will enable applications and services to transparently leverage network infrastructure on behalf of the user; empower end-to-end services; and support distributed network-wide service creation, provisioning, and management.

DEN defines a directory as a centralized repository that defines the relationship of users and applications to network services. DEN builds intelligent networks and networked applications that are managed holistically by associating users and applications to network services and according to a consistent and rational set of policies. This will result in a generation of cross-domain network applications and services that are intelligent and self-managing.

DEN is part of the Distributed Management Task Force's (DMTF) (refer to <http://www.dmtf.org> for more information) Common Information Model (CIM) standard to model functionality and management of network elements. DEN enables a company to manage its network as a single system and provides interoperability, data sharing, and transparency of the data source for cross-domain solutions. This is the main point of DEN: Managing the network as a whole, from the hosts to the physical components. The ultimate goal of DEN is a self-managing network; it behaves according to a defined set of rules (policies) to ensure less deployment and maintenance costs.

## Metadirectories

The term metadirectory does not have a precise meaning but is used to describe a variety of ways of maintaining directory coexistence. It is an alternative to a single, shared directory solution. An enterprise directory service will often be multiple directory servers each responsible for a portion of the directory objects and entries. It is highly unlikely that an organization will have one single software directory server to provide its enterprise directory service.

As the number of Web servers, application servers, access servers and so on, in an intranet grows explosively, a concern arises: If you manage user IDs and passwords individually, users will have many combinations of user IDs and passwords.

Consequently, the users tend to set passwords that are easy to remember, for example, a girlfriend's name, favorite singer's name, birthday, and so on. This undoubtedly makes your systems or applications vulnerable.

For this reason, LDAP directories and metadirectories are one of the most effective solutions to this problem (other important almost-metadirectory applications include directory synchronization solutions, global address book solutions, and some e-commerce applications). The reason is that you can centralize the location of user IDs and passwords for systems or applications by using LDAP. Some Web servers and application servers already have the ability to access LDAP directories for authentication. Therefore, even if you have a large number of Web servers and application servers, users can access every server using only one set of user IDs and passwords extracted from the LDAP directory. This makes it easier for users to create a strong password that is harder to guess, because they need to remember only one.

Because user IDs and passwords are very critical information, it is desirable that this sensitive information be encrypted. LDAP provides a number of secure authentication mechanisms including Secure Sockets Layer support. Therefore, you can encrypt user IDs and passwords stored in the LDAP directory through the SSL communication channel between the LDAP clients and LDAP servers and consequently prevent hackers from guessing passwords. IBM WebSphere, for example, can retrieve user IDs and passwords from the LDAP directory.

This trend will eventually lead to the complete elimination of user IDs and passwords. By using certificates instead of ID/password pairs, companies can better secure their networks, because the use of certificates makes brute-force attacks much harder (and thus, expensive). There are no easy-to-guess passwords, no social engineering (guessing passwords based on personal information), and no dictionary attacks. These certificates can even be stored on smart cards for added physical security and better mobility.

Several companies implement these solutions today, at least for network sign-on. Coupled with the growth of directory-enabled applications, we expect ever-increasing support for these solutions.

## Directories and databases

In Chapter 2, “Planning your directory” on page 17, we explored several differences and similarities between directories and databases. These may lead to the suspicion that a directory is no more than a limited-function database. This is indeed partly true, since one of the important design goals of a directory service is that it can be accessed and used from relatively small and simple applications. In fact, certain vendor products, such as IBM's SecureWay Directory, use a relational database under the cover to implement the functions. LDAP as a standard does not specify the underlying database. Each vendor chooses their implementation based on their strengths. IBM being in the large server market and addressing scalability requirements has chosen a relational database as the underlying database.

Proposals are being discussed in the standards bodies to add some functions to future versions of LDAP that currently are specific to databases, such as support for transactional updates.

Depending on the future growth and development of these technologies (directories and databases), we may see them merging. This, in turn, would mean the definitive incorporation of directories as simple database applications tuned for read, instead of write, performance.

## Directories and database tools

In parallel with this trend towards database functionality we also expect to see those tools that have traditionally provided access to databases for reporting and graphing, via ODBC for instance, to extend their range to cover LDAP. One such example is IBM's former BNDM for AS/400 product, now renamed BlueNotes Direct Messaging, which performs the letter-merge function of a word processing tool but can also do so based on Domino Directory and can use the directory to route the output to e-mail, fax, print, or Web posting according to customer preferences. A natural extension of this is that the directory will need to store such preferences as they are captured in customer enrolments.

## Directories and operating system integration

The major directory vendors are increasingly tying their directory service into the operating systems upon which they run. The most recent convert to this approach is Microsoft, which is making Active Directory a key part of Windows 2000, as opposed to having separate directories for e-mail and other products.

In the case of the iSeries the SecureWay directory was always a free, optional part of OS/400. At Version 5 Release 1 it became fully integrated with the operating system.

Some may argue that this diminishes the importance of the server operating system, in the same way that the universality of the Web browser diminished the importance of the desktop operating system. Application developers no longer need to tie themselves to specific operating systems. To an extent this is true, but then of course for highly manageable, scalable, and reliable operating systems like OS/400 this may be an opportunity. The majority of new applications will be able to run on the customer's choice of the best platform without this being dictated by the application, and the large installed base of existing native applications can readily be network-enabled without a migration.

## More directory content types

To date we have seen directories largely used as repositories of information about people. Increasingly we are seeing the addition of printers, servers, and even file system links, to the directory. We expect this trend to accelerate.

## Remote printing

Consider this scenario. One of the authors of the presentation to accompany this redbook is creating text while working in Europe and is using the international A4 paper size. He is to present it at a conference in the US where hard copy handouts are required and letter format is preferred but his office does not stock US paper sizes. The handouts are due two days before the conference. Should he e-mail the presentation to the conference center, hope that they use compatible presentation software, and ask them to print a copy before photocopying? Should he order US formatted paper in Europe and then hope that it arrives either in time to print one copy for further duplication on-site or in time to make multiple copies and to travel early with them?

No. Instead, he simply reformats the presentation to letter size, accesses the Web site of the conference center's copy offices, uses their LDAP directory to locate a suitable printer and prints directly to that printer across the Internet. IBM's latest InfoPrinters even have an inbuilt Web server that enables him to respond to printer messages when addressed in this way.

Now, why would the copy office make such printer information public? Because they are in the print business and they can charge for the work. A similar scenario applies when a travelling manager wishes to send copies to his head office before travelling to a meeting.

We expect these remote printing scenarios to grow strongly in the near future.

## Directories in Knowledge Management

Knowledge Management (KM) is all about finding the right expert with the right knowledge in the right place at the right time. The three cornerstones of KM are known as People, Places and Things. Lotus says "People are the experts. They need a Place - both physical and virtual - that fosters collaboration and learning. The knowledge they bring to bear is made up of content, processes or rules - Things."

Clearly, a clean, up-to-date, secure, and accessible list of people, both inside and outside of the organization, must be at the heart of KM. The directory is the ideal place for this. As an example, the Lotus Discovery Server's Expertise Locator imports a list of people and attributes from Domino Directory or LDAP into its Profile Repository. One of the key steps in preparing for KM is clearly to clean up your directory.

As KM grows in importance as an application so, we believe, will directories and the role they play in KM.

## Directory and taxonomies

One of the core components of Knowledge Management is a taxonomy. Think of this as a set of hierarchical definitions of your organization and its attributes.

A taxonomy for IBM may have a geographically-based hierarchy based on tree-structure entries like World/Americas/USA/Minnesota/Rochester. It may have a parallel product-based hierarchy based on entries like Products/eserver/series/270/IOPs. It may have an organizationally-based hierarchy with on entries like Corporate/Marketing/Industry/Pharmaceutical. There would also, of course, be a management reporting hierarchy, which may or may not map to the organizational hierarchy.

In the past, the only hierarchical structure in directories has been the namespace design, but as we can see from the IBM example, one tree is not enough to define an entire organization.

Lotus Domino has shown the way here with its Corporate Hierarchy Information fields in the directory's Person documents, which provide for four six-level hierarchical structures. These provide for users to "drill down" through the directory and find users by department or job title without necessarily even knowing their names.

The BlueNotes Directory Taxonomy for Domino product provides a way for the administrator to build the taxonomy and automatically apply it to these fields (see Appendix E, “The BlueNotes Product Suite” on page 549 for details). The implementation of this Domino function has begun most strongly amongst customers in Japan, but as directory sizes continue to grow and as content is added we can expect to see this trend of requiring more intelligent views of the data to spread to all geographical areas.

## **Pervasive access**

Now think of directory users with a smaller device *footprint* - that of a mobile cell phone or palmtop device. These users are on the ideal platform to require the fast lookup of a telephone number. We expect this form of access to grow and we think that the taxonomy-based view of the directory will enable better navigation than scrolling through thousands of surnames.

## **Separating applications from infrastructure**

At Lotusphere 2002 the Lotus Directory developers announced a strategic and very interesting future intention for their product. They intend for Domino Directory to be used by non-Lotus applications as at present and also for Lotus applications, and potential such as Notes, to be used with non-Domino LDAP-enabled directories.

This means that the applications become separated even further from the infrastructure. We expect other vendors to follow this lead from Lotus.

## **Extranet/Internet directories**

Most directories today are wholly or largely contained within the organization. As you have seen from the scenario in this redbook, it is now both feasible and, in many cases, desirable to extend this to a broader group of enterprises. Firms like IBM already make available their entire world-wide e-mail directory in the Web via LDAP (see <http://whois.ibm.com>) and many more do so every day. This we expect to be the major trend in the near future. It will become commonplace for people to use the Web and LDAP, rather than a call center or switchboard, to find an appropriate member of staff in your organization.



## Shared organizational directories

Rather than just publish directories as above, the scenario in this redbook has also shown how the directories from one organization can be replicated to another. Taking this trend a stage further we might expect whole communities, for instance IBM and its 95,000 business partners, to share directory information securely in this way. One advantage of this is that the business partners could add further information to the IBMer's entries about their relationship with them, for instance, who is the contact for a particular product or marketing program. They could also work with an off-line copy of the directory for mobile workers.

Other sectors where shared semi-private databases of people-based information would be useful include government (particularly the security services), missing persons bureaux, medical records, and credit control.

## Reuniting friends

The recent success of the Friends Reunited Web site (<http://www.friendsreunited.co.uk>) that allows you to find out what your old school or workplace friends are now doing illustrates the popularity of public sites to look for people. This trend could reach the stage where you can identify the majority of Web users in the world, albeit through separate sites at this stage.

## Directories and genealogy

Some of the largest non-governmental people-based databases in the world include those providing genealogical information. These are of two main kinds. The first consists of simple listings such as church burials or immigration records where the people entries are otherwise unrelated. The second is more structured sets of family trees where the genealogy (ancestors, siblings, descendants) is part of the database structure. In the brief time available to the researchers of this redbook we found only limited use of LDAP in such databases, but it would make the searching for relatives, ancestors, and descendants much easier. Conversely, the current LDAP standards do not appear to support genealogy-based schemas. Both developments would seem to be predictable.

## One world-wide directory network

Taking these three last items to their logical conclusion, we may well reach the point where a single world-wide directory image emerges, at least for people-based entries. Such an image need not and would not contain all the data from all the constituent directories because most of the searches would be handled by referrals, but we could reasonably expect to see at least the ability to perform one world-wide search to find a person's e-mail address.

# The BlueNotes Product Suite

The BlueNotes suite of products provide complete office productivity. They provide integration between servers (including IBM eServer iSeries, xSeries (and other Intel platforms), pSeries (and other UNIX platforms)) and Lotus Domino and Notes. They improve efficiency and productivity in an office workplace environment.

Various components of the suite can be used for:

- ▶ Messaging migration
- ▶ Server consolidation
- ▶ Customer Relationship Management (CRM)
- ▶ Knowledge Management (KM)
- ▶ Collaborative e-business
- ▶ Mobile and pervasive
- ▶ Application development tool

The BlueNotes Suite is developed by Typex and marketed by IBM and Typex.

The product suite consists of:

- ▶ Directory products
- ▶ MailStore
- ▶ Document Warehouse
- ▶ Direct Messaging
- ▶ Task manager
- ▶ Office portal

## Directory products

The BlueNotes Directory Products have two optional modules, which provide directory integration and enhancement of the Domino Directory.

The Directory Synchronisation for the OS/400 module adds selected OS/400 user profiles to the System Distribution Directory; exports them to the IBM SecureWay directory, which is LDAP-enabled; and then makes them available to the Domino Directory, hence providing a replacement for the Domino DirSynch function withdrawn in Domino 6 for OS/400.

The Domino Taxonomy module provides users with hierarchical views of the Domino Directory when selecting addresses. A four-category, seven-level enterprise taxonomy can be defined for people-related information based on directory fields such as Department, Location and Internet address. The taxonomy database then updates corporate hierarchy information in the directory's person documents.

The same taxonomy is also available to categorize data in other Notes applications including BlueNotes Document Warehouse and MailStore for Domino.

## Role in directory projects

Clearly, this product in the suite is the one with the most impact in directory projects. The close affinity with OS/400's SDD, SecureWay, and Domino means that it is used to obtain and maintain a common, cross-platform set of directory-related data and then to provide enhanced Domino user access to that data.

What is also unique is the ability to use the structure and content Directory data as the basis for other Notes and Domino applications.

Future enhancements will focus on extracting further data from OS/400 and LDAP to populate Domino.

## MailStore

BlueNotes MailStore for Domino enables the automatic filing of sent and received Lotus Notes e-mail into a shared database with manual and automatic categorization options including Enterprise Taxonomy.

For instance, mail received from a customer can be filed under the category for customers and the sub-category for that organization. This enables the interaction with a customer to be tracked and shared in a CRM-like function in much the same way that a lawyer keeps files of customer correspondence.

MailStore for Domino can benefit users and server administrators. It can be used in any of the following scenarios:

- ▶ Messaging Migration
- ▶ Customer Relationship Management (CRM)
- ▶ Knowledge Management (KM)

## Role in directory projects

MailStore might at first seem unrelated to directories, but consider the scenario where you have categorized your directory by company, department, and location. If you receive and file an e-mail from a customer or supplier, would not it be nice to be able later to search for that e-mail based on those fields? For example, “Can I see all mails from a supplier in a given month?” With MailStore you can.

## Document Warehouse

BlueNotes Document Warehouse provides a Domino-based, cross-platform index of documents in a wide variety of repositories.

It is a three-tier client/server solution that creates a Lotus Notes or Domino-based document index as an object server to the document repository, including support of existing document libraries. Documents and other objects can be accessed from either a Notes or Domino environment.

It works like a Data Warehouse, but for documents. It collects and maintains a document server index, replicates indices to one or more Domino servers, and adds Domino value to the index for both Notes and Web interfaces. It provides a consolidated view across multiple document servers, supplying *logical folder* views based on index fields and offers a full text search facility on index data. In summary, making it easier to find a document with the added benefit of providing a link or *shortcut* back to the document.

Multiple views of documents, for example by author, path, size, or date revised, could include:

- ▶ Word processing
- ▶ Spreadsheets

- ▶ Faxes
- ▶ Images
- ▶ OV/400 documents
- ▶ Other objects that can be stored on multiple servers in a network

The Notes client enables migration of file system documents to Notes attachment. From there it is possible to readily migrate both the document profile and its content to domino.doc.

The Document Warehouse has benefits for users, developers, and server administrators. It can be used for:

- ▶ Messaging migration
- ▶ Server consolidation
- ▶ Customer Relationship Management (CRM)
- ▶ Knowledge Management (KM)
- ▶ Collaborative e-business
- ▶ Mobile and pervasive
- ▶ Application development tool

### **Document Warehouse for MultiPlatforms**

BlueNotes Document Warehouse for MultiPlatforms is used for indexing all legacy, 32-bit server platforms supporting the DOS **DIR** command including Windows NT, 2000, OS/2 Warp, NetWare, and Linux servers. (Intel clients due in development.)

The function and benefits are as described above for the Document Warehouse product. It can be used to migrate files alongside of an e-mail migration, for example, Microsoft Exchange. This version has a batch index build facility and points-based licensing for flexibility.

### **Document Warehouse for OS/400**

Document Warehouse for OS/400 works in much the same way as the MultiPlatform version, but with several key enhancements:

- ▶ Scheduled, incremental document index build ensures high fidelity of the index.
- ▶ Additional information from OS/400's QDLS (Document Library Services) supports shared folder objects.
- ▶ Automated launch of predefined editors and viewers, for example, for RFT and FFT documents and images.
- ▶ Additional security for index details of personal documents.

Document Warehouse for OS/400 was previously marketed by IBM as BNDW for AS/400. It is now available as IBM 5620-FHM.

### **OfficeVision/400 Conversion Tool**

BlueNotes OfficeVision/400 Conversion Tool is an enhancement to Document Warehouse for OS/400. It provides:

- ▶ Additional document description fields, including Document Class, Keywords, and OS/400 Document Type.
- ▶ Access to filed OV/400 e-mail documents, the capture of, and access to, existing OV/400 e-mail information and its conversion, where appropriate, to Notes attachments.
- ▶ Online user listings of OV/400 calendar entries.

This product was previously available as a feature of the IBM OfficeVision to Lotus Notes Migration Tools for AS/400. It is now available as IBM 5620-FHN.

### **Document Warehouse for Domino**

Document Warehouse for Domino provides a cross-database and cross-server index of Notes documents in multiple Notes databases. These might include a document library, a discussion database, e-mail, domino.doc, and a CRM application.

### **Document Warehouse for the Web**

Document Warehouse for the Web provides team-based Web bookmarks. It maintains a Notes database of bookmarked Web URLs, indexed by Domino, not the browser, and hence shareable.

It allows for multiple categorizations, including categorization by the Enterprise Taxonomy of BlueNotes Directory Services. It has securable views by group membership.

It is used to provide shared bookmarks of external sites such as all competitors, suppliers, customers, technical support sites, and so forth.

## **Role in Directory Projects**

The positioning of Document Warehouse in Directory projects is best seen not as a tool to enhance directory data but as a way to enhance a document store using the structure of the directory via a taxonomy.



# Direct Messaging

BlueNotes Direct Messaging advanced data merge feature enables mass messaging from CRM and other customer databases to generate output such as messages, letters, reports, and statements.

BlueNotes Direct Messaging benefits include:

- ▶ **Enhanced Performance:** Data merges are performed at a rate of approximately 185,000 per hour.
- ▶ **Ease of use:** Forms can be created either by end users or by developers with the ability for users to make last-minute amendments.
- ▶ **Versatility:** Output can now be generated in multiple dynamic ways, for example, fax, e-mail, print, or published on the Web, depending on recipient preferences.

Supported data sources include ODBC plus the new alternative of Domino databases, including the Domino Directory and the entire range of Notes-based CRM applications.

The Print Engine function provides Domino with the network print function, which it otherwise lacks by using the Notes Services approach pioneered by IBM's Fax for Domino. This also supports Advanced Function Printing (AFP).

Documentation and sample forms are provided for developers to incorporate the Direct Messaging function into their applications including Web-based statements for Electronic Bill Presentation and Payment (EBPP).

Direct Messaging also contains a relational database to Notes export function known as DataPump. This has been described as "DECS on steroids." Importing data to Notes is now fast and easy.

Direct Messaging was previously known as BlueNotes Direct Marketing and was also marketed by IBM as BNDM.

Direct Messaging can benefit users and developers. It can be used for:

- ▶ Messaging Migration
- ▶ Customer Relationship Management (CRM)
- ▶ Collaborative e-business
- ▶ Mobile and pervasive
- ▶ Application development tool

## Role in directory projects

Direct Messaging is used in directory projects in two key ways. Firstly, with the DataPump function a Domino Directory can readily be populated from a relational or Notes database, or indeed the reverse, for example. when a CRM database requires directory information. Secondly, when a targeted message is to be sent to customers, prospects, or suppliers the Directory field contents can be used for record selection purposes.

## More information

Besides the products listed in this appendix, the BlueNotes product suite contains several other products. These products are not described here because they are not related to directory functions.

For more information about products in the BlueNotes Suite visit <http://www.bluenotes.com>, e-mail to [info@typex.com](mailto:info@typex.com), or refer to appropriate to the IBM product documentation.

## Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

### 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:

<http://www.redbooks.ibm.com/redbooks/SG246193>

Alternatively, you can go to the IBM Redbooks Web site at:

[ibm.com/redbooks](http://ibm.com/redbooks)

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG246193.

## Using the Web material

The additional Web material that accompanies this redbook includes the following files:

<i>File name</i>	<i>Description</i>
<b>JNDI.zip</b>	Contains the Java application code from Chapter 12, “Using the JNDI to search and update the directory” on page 477.
<b>SSO.zip</b>	Contains the servlet and Domino application that is used in Chapter 8, “Single Sign-On with Domino and WebSphere 4.0” on page 337 to test the Single Sign-On.
<b>README.HTML</b>	The readme file contains further details about how to install and use the provided sample applications.

## System requirements for downloading the Web material

The following system configuration is recommended:

<b>Hard disk space</b>	10 MB.
<b>Operating system</b>	One operating system that supports a browser and can unzip the downloaded files.

## How to use the Web material

The readme.html file available from the download page contains further details about the individual files and how to install and use the sample applications.

# Abbreviations and acronyms

<b>AAT</b>	Application Assembly Tool	<b>EJB</b>	Enterprise JavaBea
<b>ACL</b>	Access control list	<b>FSIOP</b>	File Server Input Output Processor
<b>AD</b>	Active Directory	<b>HTML</b>	Hypertext Markup Language
<b>AES</b>	Advanced Encryption Standard	<b>HTTP</b>	Hypertext Transfer Protocol
<b>AFP</b>	Advanced Function Printing	<b>IAB</b>	Internet Architecture Board
<b>API</b>	Application Programming Interface	<b>IANA</b>	Internet Assigned Number Authority
<b>AS/400</b>	IBM Application System/400	<b>IBM</b>	International Business Machines Corporation
<b>ASN</b>	Abstract Syntax Notation	<b>IETF</b>	Internet Engineering Task Force
<b>ASP</b>	Application Service Provider	<b>IFS</b>	Integrated File System
<b>BER</b>	Basic Encoding Rules	<b>IIOF</b>	Internet Inter-Orb Protocol
<b>CA</b>	Certificate Authority	<b>ILE</b>	Integrated Language Environment
<b>CRAM-MD5</b>	Challenge-Response Authentication Mechanism - Message Digest 5	<b>IMAP</b>	Internet Message Access Protocol
<b>CRL</b>	Certificate Revocation List	<b>ISO</b>	International Standards Organization
<b>CRM</b>	Customer Relationship Management	<b>ITSO</b>	International Technical Support Organization
<b>DAP</b>	Directory Access Protocol	<b>J2EE</b>	Java 2 Enterprise Edition
<b>DCM</b>	Digital Certificate Manager	<b>JDBC</b>	Java Database Connectivity
<b>DEN</b>	Directory-Enabled Networks	<b>JNDI</b>	Java Naming and Directory Interface
<b>DES</b>	Data Encryption Standard	<b>JSP</b>	Java Server Page
<b>DIT</b>	Directory Information Tree	<b>JVM</b>	Java Virtual Machine
<b>DMT</b>	Directory Management Tool	<b>KM</b>	Knowledge Management
<b>DMT</b>	Directory Management Tool	<b>LAN</b>	Local Area Network
<b>DN</b>	Distinguished Name	<b>LDAP</b>	Lightweight Directory Access Protocol
<b>DNS</b>	Domain Name Services	<b>LDIF</b>	LDAP Data Interchange Format
<b>DOLS</b>	Domino Online Services		
<b>DSAPI</b>	Domino server Application Programming Interface		
<b>DSML</b>	Directory Services Markup Language		
<b>DSS</b>	Directory & Security Services		

<b>LDUP</b>	LDAP Duplication/Replication/ Update Protocols	<b>TLS</b>	Transport Layer Security
		<b>UMICH</b>	University of Michigan
<b>LTPA</b>	Lightweight Third-Party Authenticatio	<b>URL</b>	Uniform Resource Locator
		<b>WAN</b>	Wide Area Network
<b>MD5</b>	Message Digest 5		
<b>MIME</b>	Multipurpose Internet Mail Extensions		
<b>MMC</b>	Microsoft Management Console		
<b>NNTP</b>	Network News Transport Protocol		
<b>NRPC</b>	Notes remote procedure call		
<b>OID</b>	Object Identifier		
<b>OS/400</b>	Operating System/400		
<b>OSF</b>	Open Software Foundation		
<b>OSI</b>	Open Systems Interconnection		
<b>PKCS</b>	Public-Key Cryptography Standards		
<b>PKI</b>	Public Key Infrastructure		
<b>POP</b>	Post Office Protocol		
<b>RDN</b>	Relative Distinguished Name		
<b>RFC</b>	Request For Comments		
<b>RPC</b>	Remote procedure call		
<b>SAS</b>	Secure Association Service		
<b>SASL</b>	Simple Authentication and Security Layer		
<b>SDD</b>	System Distribution Directory		
<b>SDK</b>	Software Development Kit		
<b>SHA</b>	Secure Hash Algorithm		
<b>SMTP</b>	Simple Mail Transfer Protocol		
<b>SPI</b>	Service Provider Interface		
<b>SQL</b>	Structured Query Language		
<b>SSL</b>	Secure Sockets Layer		
<b>SSO</b>	Single Sign-On		
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol		

# 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 563.

- ▶ *S/400 Internet Security: Developing a Digital Certificate Infrastructure*, SG24-5659
- ▶ *Domino and WebSphere Integration on the IBM eSeries iSeries Server*, SG24-6223
- ▶ *Getting the Most from Your Domino Directory*, SG24-5986
- ▶ *IBM eServer iSeries Wired Network OS/400 V5R1 DCM and Cryptographic*, SG24-6168
- ▶ *IBM WebSphere V4.0 Advanced Edition Handbook*, SG24-6176
- ▶ *LDAP Implementation Cookbook*, SG24-5110
- ▶ *Lotus Domino for AS/400 R5: Implementation*, SG24-5592
- ▶ *Understanding LDAP*, SG24-4986
- ▶ *Using LDAP for Directory Integration: A Look at IBM SecureWay Directory, Active Directory, and Domino*, SG24-6163
- ▶ *WebSphere Advanced Edition: Security*, SG24-6520
- ▶ *WebSphere Edge Server: Working with Web Traffic Express & Network Dispatcher*, SG24-6172

## Other resources

These publications are also relevant as further information sources:

- ▶ *e-Directories - Enterprise Software, Solutions, and Services*, Addison Wesley, ISBN 0-201-70039-5
- ▶ *Implementing Directory Services*, Mc Graw Hill, ISBN 0-07-134408-X
- ▶ *iSeries Security Reference*, SC41-5302

- ▶ *Understanding and Deploying LDAP Directory Services, 1999, Howes Smith & Good, New Riders Publishing, ISBN 1-57870-070-1*

## Referenced Web sites

These Web sites are also relevant as further information sources:

- ▶ ANSI Organization home page where OIDs are registered for the US  
<http://www.ansi.org>
- ▶ Article about LDAP configuration support with IBM HTTP Server for iSeries (Original)  
<http://www-1.ibm.com/servers/eserver/series/software/http/services/ldapinfo.html>
- ▶ Base64 class free software  
<http://ostermiller.org/utills/>
- ▶ Distributed Management Task Force organization  
<http://www.dmtf.org>
- ▶ IBM SecureWay Directory Online Library
- ▶ <http://www.ibm.com/software/network/directory/library/>
- ▶ IBM SecureWay Directory Schema  
<http://www-1.ibm.com/servers/eserver/series/ldap/schema>
- ▶ IBM SecureWay Directory Web site  
<http://www.ibm.com/software/network/directory>
- ▶ Information about Tivoli Policy Director  
<http://www.tivoli.com>
- ▶ iSeries server LDAP Frequently Asked Questions  
<http://www.iseries.ibm.com/ldap/ldapfaq.htm>
- ▶ LDIF RFC 2849  
<http://www.rfc-editor.org/rfcsearch.html>
- ▶ Lotus Software, a division of IBM Software Group  
<http://www.lotus.com>
- ▶ Object classes and attributes Object Identifier (OID)  
<http://www.alvestrand.no/objectid/index.html>



- ▶ Online documentation for IBM WebSphere Application Server for iSeries  
<http://www.ibm.com/servers/eserver/iseriess/software/websphere/wsappserver/docs/doc.htm>
- ▶ Online documentation for Sun Microsystems' Java Naming and Directory Interface  
<http://java.sun.com/products/jndi/docs.html>
- ▶ Typex BlueNotes family  
<http://www.bluenotes.com>
- ▶ Version 4 Release 4 enhancements and changes  
<http://www-1.ibm.com/servers/eserver/iseriess/ldap/ldapv4r4.htm>

## How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

## IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.



# Index

## Symbols

\*ALLOBJ 58, 64, 129, 232  
\*AUDIT 58, 64  
\*AUDLVL 146  
\*IOSYSCFG 58, 64, 232  
\*OBJAUD 146  
\*SECADM 129  
\*SECOFR 340  
\*SYSTEM certificate store 127, 138

## A

-a switch 176  
Access class  
    Critical 199  
    Normal 199  
    Sensitive 199  
access class level permissions 199  
access classes 211  
access control 47  
Access Control List 150  
Access Control List (ACL) 13, 31, 198, 370, 517  
Access Control List (ACL) group 201  
Account name 438  
Account referral 442  
ACL 31, 71, 198, 217  
ACL Entry Details window 221  
ACL Group members 203  
ACL groups 199, 215  
Active Directory 92, 103  
AD 92  
ADD\_ATTRIBUTE 501  
ADDDIRE command 92, 121  
Address look-up 437  
administration 31  
administrative overhead 26  
Administrator DN 67  
administrator DN 67, 108  
ADSI Edit 93  
ADSI Edit Microsoft Management Console (MMC)  
92  
Allow directory updates 111  
Anonymous access 183, 311  
anonymous access 198  
anonymous connections 208  
Anonymous users 208  
ANSI 14  
API 7, 55, 75, 176, 195, 198, 458, 462, 478  
API information 465  
application infrastructure 480  
application-independent directory 9  
ASCII text file 259  
Assign New Certificate 132, 139  
attribute 23, 26  
Attribute authorities 219  
attribute level 199  
Attribute level permission 199, 217  
attribute name 167  
attribute permissions 208, 217  
attribute properties 174  
attribute value 500  
Attributes 13, 173, 182, 270, 495  
attributes 13, 126, 257, 473  
Audit entry type 146  
auditing settings 145  
Auditing support 143  
Auditing tab 146  
Authentication 29, 42, 230, 304  
authentication 29, 46, 116, 128, 232, 480, 497  
authentication challenge 237  
authentication dialog window 237  
Authentication mechanisms 346  
Authentication method 116, 480  
Authentication policies 344  
Authentication type 153, 163  
    CRAM MD5 154  
    Kerberos 154  
    None 153  
    SASL External 153  
    Simple 153  
authorities 473  
Authority 206  
Authority Failure Code 147  
Authorization 31  
Authorization policies 344  
Availability 33, 41  
availability 35

## B

- b switch 181–183
- base operating system 62
- Base64 485
- Base64 encoded 489
- Basic Authentication 30, 245
- basic authentication 230
- bind 156, 480
- bind distinguished name 177
- binProperty 257, 261, 270, 519
- BlueNotes
  - Direct Messaging 555
  - Directory Products 551
  - Document Warehouse 552
- BlueNotes Directory Synchronisation for OS/400 40
- BlueNotes Directory Taxonomy 49
- BM Universal Database (UDB) 55
- Branching the Directory Tree 26
- Browser LDAP search
  - attributes 196
  - dn 196
  - filter 196
  - hostname 196
  - port 196
  - scope 196

## C

- CA Trust 132, 139, 143
- CCITT 8
- centralized approach 33
- Certificate Authority (CA) 30, 129
- Certificate name 153, 163
- Certificate Revocation List (RCL) 36
- Certificate Store 129
- CFGDOMSVR command 308
- CFGTCP command 76
- Change Directory Server Attributes (QgldChgDirSvrA) API 462
- Change log 175, 185
  - log directory changes 186
  - maximum entries 186
- ChangeDirEntry servlet 484
- changeLogEntry object class 186
- CHGDIRE command 92
- CHGSYSDIRA command 91
- CHGSYSVAL command 144
- cipher 134

- Clarissa 182
- classifications 199
- classified attributes 208
- classNotFound errors 153
- Client Access Express 218
- client applications 140
- Client authentication 132
- client authentication 143
- client certificate 154–155
- Client Encryption 63, 128
- Client errors 59
- cn(common name) 79
- cn=administrator 84, 91, 182, 237
- CN=ANYBODY 153, 208
- cn=changelog 186
- common directory 10
- Composite names 479
- configuration directives 257
- Configuration includes 284
- Configuration Wizard 58, 64
- Configure Directory Server (QgldCfgDirSvr) API 462
- constant definitions 466
- container 261
- context 479
- Context interface 479
- Context.INITIAL\_CONTEXT\_FACTORY 491, 496
- Context.PROVIDER\_URL 491, 497
- Context.SECURITY\_AUTHENTICATION 497
- Context.SECURITY\_CREDENTIALS 498
- Context.SECURITY\_PRINCIPAL 497
- Controlling access 198
- CORBA 348
- Create LDAP server setup 265
- credentials 79
- CRM 49
- CRTJRN command 144
- CRTJRNRCV command 144
- CRTUSRPRF command 120
- Crypt algorithm 226
- Cryptographic Access Provider 63, 127

## D

- D switch 177, 181–183
- DAP 8
- Data elements 22
- data structure 465
- DB2 19

- DCM 132, 138
- default referral 123
- Default server profile 246
- Defining Directory Content 18
- Defining Directory Requirements 18
- Delegation policies 344
- Delimiter 238
- DEN 17
- departmentNumber 80
- DES algorithm 226
- Design 18
- Digital Certificate Manager 63, 127–129
- Digital Certificate Manger 110
- digital signature 30
- DirContext interface 479, 498
- directories 6
- Directory Access Protocol (DAP) 8
- directory administrator 154
- Directory Assistance 292–293, 297, 304, 327, 421, 510
- Directory Catalog 292, 295
- directory client SDK 158
- Directory components 11
- Directory concepts 5
- directory context 495
- directory implementation 23
- Directory implementations 17
- Directory Information Tree (DIT) 11, 27, 49, 165, 479
- Directory Management Tool (DMT) 48, 57, 63, 150, 508
  - GSKit 159
  - SSL 158
- Directory service 7, 21
- Directory Services 41, 55, 58, 63, 67, 76, 81, 128, 138, 149
- Directory Services client 127
- Directory Services publishing 127
- Directory Services server 127, 130
- directory subtree 479
- directory-enabled applications 20
- Directory-Enabled Networks (DEN) 17
- distinguished name 26, 28, 79–80, 237, 473
- Distinguished Name (DN) 11, 29, 84, 261
- distributed approach 33
- DIT 11, 36, 49, 126, 165, 258, 479
- DMT 57, 73, 150, 154, 201
- DMT binding 166
- dmt.conf 156, 508

- DN 26, 79, 84, 122
- DNS 33, 478
- DNS switchover 34
- Document protection 238
- Domino application login 380
- Domino authentication 370
- Domino Directory 37, 45, 291, 294, 339, 440, 442
- Domino LDAP server task 292
- DSAPI filter file names 368

## E

- Edit Authority window 208, 221
- editor 259
- EDTF command 191
- Edwin 184
- elements() 500
- e-mail systems 22
- Enable attribute-level permissions 218
- encoded form 489
- encrypt e-mail 31
- encryption algorithms 134
- Entry RDN 166, 168
- Entry Type 146
- Entry type 166, 168
- ePerson 79, 171, 237
- eProperty object class 257
- error code 473
- Excluding entries from being published 91
- Explicit ACL 208, 216
- Explicit directory referrals 125
- Export File 112, 188
- Export LDIF File (QgldExportLdif) API 463
- Exporting directory data 188
- extend your directory schema 170
- extended realm 342
- Extending the schema 522
- EZ-Setup 64

## F

- f switch 177, 181
- facsimileTelephoneNumber 80
- filter 23
- firewall 123
- FTP 21, 187

## G

- general-purpose databases 7

- getAttributes() 498
- GetMask 243
- givenName 80
- group 242
- Group authentication 251
- GroupMemberAttrs 243
- GroupNameFilter 242–243
- groupOfNames 242
- GroupSearchBase 242–243
- GSKit V5 158
- GUI design 477

## H

- h switch 181
- hash 154
- Hashtable 479
- hashtable 491
- HTTP 21
- HTTP \*Admin instance 129, 233, 244
- HTTP cookies 382
- HTTP header response 485
- HTTP password 305
- HTTP server DN 237, 268
- HTTP Server for iSeries
  - authentication support 244
  - configuration support 230, 254
  - group authentication 240
  - LDAP server setup 234
- HTTP servlet request 493
- HTTP Web server 41

## I

- IBM HTTP Server for iSeries 42, 198, 229
- IBM Key Manager 159
- IBM SecureWay Directory 11, 13, 55, 149
- IBM SecureWay Directory Client Software Development Kit (SDK) 57
- IBM SecureWay Directory Management Tool (DMT) 57
- IBM SecureWay Directory Version 3.2.2 158
- IBM WebSphere Advanced Edition V4.0 340
- IBM WebSphere Application Server 11
- IBM WebSphere Host On-Demand 11
- Identity 480
- IETF 24
- IFS path 257
- ILE 198
- ILE RPG 7

- IMAP4 30
- Import File 113, 193
- Import LDIF File (QgldImportLdif) API 463
- Importing directory data 192
- inetOrgPerson 79, 169, 171
- Infrastructure 18, 32
- inherit 216
- Inherited attributes 526
- initial load 111
- Initial replication 111
- initial request 122
- InitialDirContext class 490, 496
- Integrated Language Environment (ILE) 55, 195, 458
- Internet Engineering Task Force 24
- iSeries server 338

## J

- Java Developer Kit 481
- Java Naming and Directory Interface (JNDI) 150, 195, 458
- JDNI 198
- JNDI 195, 477
  - BasicAttributes class 500
  - context 490
  - createSubcontext() method 495
  - destroySubcontext() method 495
  - environment 490
  - getAttributes() method 495
  - java.naming.factory.initial 479
  - java.naming ldap.version 480
  - java.naming.provider.url 480
  - java.naming.referral 480
  - java.naming.security.authentication 480
  - java.naming.security.credentials 480
  - java.naming.security.principal 480
  - java.naming.security.protocol 480
  - modifyAttributes() method 495, 501
  - NamingEnumeration class 494
  - packages 490
  - Search base 491
  - search control 493
  - search() method 492
  - SearchResult class 493
- Journal 61
- journal entry 146
- journal receiver 61

## K

Kerberos 30, 78, 109, 116, 154  
key database 155  
Key Database File 159  
Key Ring File 411, 414  
Keyclass file name 155, 164  
Keyclass file password 155, 164  
keyring class file 154

## L

-L switch 183  
LAN 34  
LDAP 3, 6–7, 17, 24, 33, 39, 57, 106, 229  
LDAP Authentication 247  
LDAP client 57  
LDAP client installation 156  
LDAP Data Interchange Format (LDIF) 36, 44, 150, 177, 187, 260, 275, 313, 463  
LDAP directory 42, 75, 165, 230, 338  
LDAP directory schemas 23  
LDAP Directory server 71  
LDAP directory servers 30  
LDAP mapping 79  
LDAP Realm 391  
LDAP requests 59  
LDAP schema 24  
LDAP server 122  
LDAP utilities 55, 150, 176, 260  
    ldapadd 63, 176, 316  
    ldapdelete 63, 184  
    ldapmodify 63, 176, 316  
    ldapmodrdn 63, 185  
    ldapsearch 63, 126, 181, 314  
LDAP utility switches 179  
ldap.application.authType 282  
ldap.application.DN 253, 283  
ldap.application.password.stashfile 253, 283  
ldap.group.memberAttributes 253  
ldap.group.name 253  
ldap.group.name.filter 253  
ldap.group.url 253  
ldap.prop file 272, 286  
ldap.transport 282  
ldap.URL 282  
ldap.url 253–254  
ldap.user.name.filter 253–254  
ldap\_control\_free() 471  
ldap\_first\_entry() 469  
ldap\_get\_values() 470  
ldap\_init() 469, 473  
ldap\_int() 463  
ldap\_memfree() 471  
LDAP\_MOD\_ADD 474  
LDAP\_MOD\_DELETE 474  
LDAP\_MOD\_REPLACE 474  
ldap\_modify\_s() 473  
ldap\_msgfree() 471  
ldap\_next\_attribute() 470  
ldap\_next\_entry() 470  
ldap\_search() 463  
ldap\_search\_s() 469  
ldap\_simple\_bind() 463  
ldap\_simple\_bind\_s() 469, 473  
ldap\_unbind\_s() 471, 475  
ldapadd 57, 125, 176  
ldapdelete 126, 176  
LDAP-enabled application 24  
LDAP-enabled applications 57  
LDAP-enabled directory 19  
LDAPInclude directive 258, 269, 272, 281  
LDAPMessage structure 469  
LDAPMod structure 474  
ldapmodify 57, 126, 145, 176  
ldapmodrdn 176  
LDAPRequire 253–254  
ldapsearch 57, 126, 145, 176  
LDIF 36, 44, 114, 150, 177, 187, 260  
LDIF file 113, 125  
LDIF format 181, 183  
library card catalog 5  
Lightweight Directory Access Protocol (LDAP) 7  
Lightweight Third Party Authentication (LTPA) 346, 351, 384  
Lightweight Third-Party Authentication 343  
Load balancing 34, 41  
load balancing 231  
Load sharing 230  
Lotus Domino 34, 43, 291, 338  
Lotus Domino Administrator client 340  
Lotus Domino for iSeries 340  
Lotus Notes 433  
LTPA 343  
LTPA authentication mechanism 345

## M

management tasks 156

Managing an LDAP directory 149  
Marion 484  
Master 42, 106, 126, 135, 507  
master 34, 106  
Maximum entries to return 439  
Maximum Number of Hits 444  
MD5 226  
Membership 241  
Message Digest 5 226  
Microsoft Active Directory 92  
Microsoft Internet Explorer 4  
Microsoft Management Console (MMC) 92–93  
MMC 92

## N

NameParser 478  
Namespace 18, 25  
naming structure 27  
NamingEnumeration object 493  
NDS 478  
Netscape Messenger 433  
Netscape Web browser 72  
NETSTAT command 138  
Network Infrastructure 18  
network traffic constraints 123  
Non-secure port 142, 236  
Notes remote procedure call (NRPC) 293  
Novell NetWare 22

## O

Object authorities 219  
object class 12, 24, 169  
object class referral 125  
Object classes 13, 172, 525  
Object Identifier (OID) 13, 522  
object permissions 217  
Objects and Attributes tab 221  
OID 14, 173  
one-time authentication 47  
one-way hash 226  
Operations Navigator 63, 108, 133, 149, 201, 217  
Operations Type 146  
optional attribute 169  
Optional attributes 526  
ORB 348  
Organization 50, 165  
Organizational unit 12, 50  
organizational unit 168

organizationalPerson 171  
OS/400 19, 57  
OS/400 Directory Server 34  
OS/400 Directory Services 41, 55, 58–59, 64, 76, 149, 170  
OS/400 LDAP Directory Services 62  
OS/400 QShell command interface 195  
OSI stack 8  
Outlook Express 435  
OV/400 40

## P

Parent DN 166, 168  
Partitioning 33  
partitioning 25, 32  
PasswdFile 243, 253  
Password 84, 99, 109, 237, 268  
password 230, 237  
performance 35, 41  
Permissions  
    Compare 211  
    Read 211  
    Search 211  
    Write 211  
permissions 199, 347  
Person 171  
POP 41  
POP3 30  
Port 153, 163  
Post Office Protocol (POP) 40  
Print shares 96  
Propagate ACL to lower level objects 216  
Properties 77, 82, 102, 108, 133  
Protection realm 239  
prototype 465  
Public Key Infrastructure (PKI) 47  
Publish print share information 98  
publishing API 60  
publishing client 138  
Publishing printer information 92  
Publishing system information 75, 78  
Publishing the SDD 42  
Publishing the System Distribution Directory 79  
put() 500

## Q

QASYDIJ4 147  
QAUDCTL system value 144



QAUDJRN 145  
 QAUDLVL 146  
 QDIRSRV 59, 507  
 Qdirsrv file share 151  
 QDIRSRV library 58  
 QDIRSRV2 61  
 QGLDPUBA 59, 506  
 QGLDPUBE 59, 506  
 QGLDSSDD 91  
 QIBM\_GLD\_DIRSRV\_CLIENT 127, 143  
 QIBM\_GLD\_DIRSRV\_PUBLISHING 78, 127  
 QIBM\_GLD\_DIRSRV\_SERVER 127, 130  
 QREPL 92  
 QRETSVRSEC 226  
 QRETSVRSEC system value 154  
 QShell 177, 212, 261  
 QSQJRN 61  
 QSYSINC/H 465  
 QUSAPIBD binding directory 58  
 QUSRDIRCL 58, 61, 75, 185  
 QUSRDIRDB 58, 75  
 QUSRSYS 61

## R

-r switch 176, 180  
 RDN 12, 27, 63, 151  
 realm 342  
 Recipient name lookup 302  
 recovery 35  
 Redbook scenario 39  
 Redbooks Web site 563  
     Contact us xvi  
 reference 123  
 referral entry 126  
 referral responses 123  
 Referrals 35, 122, 517  
 referrals 35  
 registeredAddress 80  
 registry 347  
 Relational Database 19  
 relational database 21  
 Relative Distinguished Name 151  
 Relative Distinguished Name (RDN) 12  
 Relative DN for new group 203  
 remote procedure call 10  
 REMOVE\_ATTRIBUTE 501  
 Replica 42, 106, 126, 507  
 replica 34, 107

replica server 116  
 Replication 18, 33–34, 36, 106, 135  
 replication 25, 32, 107  
 Required attributes 526  
 Restoring Directory Services 61  
 restrict access 198  
 Retain server security data 154  
 Ricarda 484  
 roles 347  
 roomNumber 80  
 round-robin 33  
 RPGLESRC 466  
 RSA Security 129

## S

-s switch 183  
 SASL 153  
 Saving Directory Services 61  
 Schema 13, 18, 23, 170, 172, 211, 521, 525  
 schema 24, 27, 78, 156, 169, 478  
 schema design 23  
 schema-checking 13  
 scope 196  
 SDD 22, 27, 37, 40, 55, 66, 81, 165  
 Search base 181, 257, 375, 439  
 search base 196  
 search criteria 196  
 Search filter 249, 269, 440, 483  
 search filter 196, 237, 493  
 search request 123, 183  
 Search Root 444  
 Search sequence 301  
 Search timeout 439  
 SearchControls object 479  
 SearchResult class 494  
 Secondary Domino directory 293  
 Secure Association Service (SAS) 348  
 Secure Hash Algorithm 226  
 Secure port 142, 236  
 Secure Socket Layer 339  
 Secure Sockets Layer (SSL) 110  
 SecureWay Directory Client SDK 459  
 SecureWay Directory Management Tool (DMT) 57  
 SecureWay V3.2 106  
 Securing Directory Entries 18  
 Security 28, 126  
 Security Center 383  
 Security roles 354

- Select location for new group 203
- selected subtree 189
- server applications 130
- Server authentication 132
- server certificate 30
- server certificates 129
- Server DN 250, 281
- Server name 163
- Server password 250, 281
- Service Provider Interface (SPI) 478
- SHA 226
- Signer Certificates 162
- Simple Authentication and Security Layer (SASL) 30
- Single authentication directory 341
- single directory 230
- Single Sign-On 47, 128
- Single Sign-On (SSO) 46, 337, 368
  - concepts 341
  - persistent authentication 342
  - SSL 402
- sn(surname) 79
- snap-in 92
- Social Security numbers 28
- Sparky 184
- special authorities 64, 129, 232
- specialized database 5
- SPI 478
- SQL 6
- SSL 30, 45, 78, 116, 128, 139, 153, 236, 339, 345
  - cipher suites 134
  - handshake 134
- SSL client authentication 305
- SSL handshake 163
- SSL-enabled 134
- SSLight key database class 161
- SSO 46
- STRTCPSVR command 129, 233, 244
- subcontext 479, 493
- subset of attributes 188
- subtree 36, 113, 183
- SUBTREE\_SCOPE 493
- suffi 76
- Suffix 66, 108, 166, 186
- suffix 25, 68, 155
- Superior class 171
- Synchronize System Distribution Directory 91
- Synchronize System Distribution Directory to LDAP (QGLDSSDD) API 463

- Syntax 521
- System Distribution Directory (SDD) 22, 26, 37, 40, 59, 508
- System Distribution Directory to LDAP mapping 79
- System-generated 67

## T

- taxonom 27
- TCP/IP hostname 116
- telephone directory 5
- telephoneNumber 80
- Testing the LDAP replication 120
- Thomas 484
- Time-outs 236
- Tivoli Policy Director 11, 47
- top 166, 171
- Topology 33
- Topology Design 18
- topology design 36
- transactions 6
- Transmission Control Protocol/Internet Protocol (TCP/IP) 57
- transport protocol 236
- TRCTCPAPP command 506, 518
- Trust policies 344
- trusted CA 154
- trusted root certificate 155
- Type Ahead 442

## U

- UDB 55
- uid 79
- Update Certificate Assignment 131, 139
- Update interval 117
- URL 195, 480
- URL request 483
- Use user entries in LDAP server 246
- User DN 154
- user profile 67, 232
- user registry 347
- User search root 268
- UserNameFilter 242–243
- userPassword 154, 174
- userPassword attribute 126, 225, 237, 250
- userPassword attribute protection
  - Crypt 226
  - MD5 226
  - None 226

- SHA 226
- UserSearchBase 242–243
- UserTools 151
- UTF-8 147
- utilities 126

## **V**

- Verify Connection 117, 137
- verifying SSL 132
- VeriSign 129
- Verisign Test CA Root Certificate 162

## **W**

- w switch 177, 181–183
- WAN 34
- Web browser 340
- Web module security 359
- Web SSO Configuration 389, 419
- Web user authentication 230
- WebSphere Administrative Console 340
- WebSphere Application Server 50, 338
  - authentication concepts 344
  - security 349
- WebSphere Application Server Advanced Edition 4.0 481
- WebSphere authorization model 347
- WebSphere Development ToolSet 481
- WebSphere Edge Server 33, 41
- WebSphere Global Security 383
- WebSphere security components 345
- well-known CA 129
- well-known port 236
- wildcard character 483, 493
- Windows 55
- Windows 2000 92
- Windows LDAP client 63, 150, 176
- Wizard 64, 81
- write permission 499
- www-authenticate 485

## **X**

- X.500 8, 20, 24
- X.501 27
- X.509 297





**Redbooks**

# Implementation and Practical Use of LDAP on the IBM <sup>®</sup>server iSeries Server







# Implementation and Practical Use of LDAP

on the IBM  iSeries Server



**Redbooks**

**Use LDAP for Single Sign-On with WebSphere and Domino**

**Experience the power of LDAP with the HTTP Server for iSeries**

**Learn how to set up and manage your directory**

This redbook will help system administrators and programmers to understand the concepts of directories. It also explains the major steps in planning and deploying a directory. You will learn how to install and configure OS/400 Directory Services with all its features provided with OS/400 Version 5 Release 1. The implementation topics also teach you how to improve availability and scalability by exploiting directory replication and referral services. In a world where security is a key factor in establishing a reliable IT infrastructure, you also gain the skills to enable your LDAP directory server and client applications to securely communicate via the SSL protocol.

The redbook continues with detailed information about how to successfully manage your directory. This includes various tools, such as the IBM SecureWay Directory Management Tool and LDAP utilities. Detailed information is provided on how to secure your directory entries and their attributes using the Version 5 Release 1 attribute-level permission enhancements.

Based on a practical scenario that spans across the entire redbook, you will learn how to leverage OS/400 LDAP Directory Services to authenticate users and share configuration information with the IBM HTTP Server for iSeries. You discover how Lotus Domino and WebSphere Application Server 4.0 exploit the OS/400 LDAP directory for Single Sign-On. Using the directory as an enterprise directory, you gain the knowledge to configure various e-mail client applications to look up e-mail addresses from a single directory. The redbook also describes how to directory-enable your applications using OS/400 APIs and the Java Naming and Directory Interface (JNDI).

## 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](http://ibm.com/redbooks)