LDAP-UX Client Services B.05.00 Administrator's Guide

HP-UX 11i v2 and v3

HP Part Number: J4269-90086
Published: June 2010
Edition: 1.0

The following table lists the publication history of this document. Check for more recent updates of the document at:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX LDAP-UX Integration Software**.

Table 1 Publishing history details

Document Manufacturing Part Number	Operating Systems Supported	Supported Product Versions	Publication Date
J4269-90086 Edition 1.0	11i v2 and v3	B.05.00	June 2010
J4269-90083	11i v1, v2, and v3	B.04.15	October 2007
J4269-90075	11i v1, v2, and v3	B.04.15	August 2007
J4269-90073	11i v1, v2, and v3	B.04.10	April 2007
J4269-90067	11i v1, v2, and v3	B.04.10	December 2006
J4269-90053	11i v1, v2, and v3	B.04.00	June 2006
J4269–90051	11i v1 and v2	B.04.00	August 2005
J4269-90048	11i v1 and v2	B.04.00	July 2005
J4269-90040	11.0, 11i v1 and v2	B.03.30	September 2004
J4269-90038	11.0, 11i v1	B.03.30	July 2004
J4269-90030	11.0, 11i v1 and v2	B.03.20	October 2003
J4269-90016	11.0, 11i	B.03.00	September 2002

Table of Contents

1 Introduction....15
1.1 Overview of LDAP-UX Client Services....15
1.1.1 How LDAP-UX Client Services works....16

2 Installing and configuring LDAP-UX Client Services....21
2.1 Before you begin: general installation and configuration considerations....21
2.2 Choosing the method of installation: guided or customized....22
2.3 Guided installation (autosetup)....23
2.3.1 What autosetup does....25
2.3.2 Principles of the LDAP-UX domain....27
2.3.2.1 Directory information tree (DIT)....28
2.3.2.2 Information model....29
2.3.2.2.1 Managed objects and how they are defined....29
2.3.2.2.2 Domain entity classification schema....31
2.3.2.3 Security framework....33
2.3.2.3.1 Proxy users....33
2.3.2.3.2 Access control rights....34
2.3.2.3.3 SSL/TLS and CA/server certificates....35
2.3.3 Domains in LDAP-UX environments....36
2.3.4 Administrators and managers in the LDAP-UX directory server environment....38
2.3.5 Using the guided installation autosetup command—syntax and options....38
2.3.5.1 autosetup options....39
2.3.5.2 autosetup environment variables....41
2.3.5.3 autosetup command examples....43
2.3.6 Guided installation steps: New Directory Server Installation mode....44
2.3.6.1 Interactively running New Directory Server Installation mode....45
2.3.6.2 Automating New Directory Server Installation mode....49
2.3.6.3 Post-installation steps for New Directory Server Installation mode....50
2.3.7 Guided installation steps: Existing Directory Server Installation mode....50
2.3.7.1 Interactively running Existing Directory Server Installation mode....51
2.3.7.2 Automating Existing Directory Server Installation mode....53
2.3.7.3 Post-installation steps for Existing Directory Server Installation mode....53
2.3.8 Guided installation steps: Existing LDAP-UX Domain Installation mode....53
2.3.8.1 Interactively running Existing LDAP-UX Domain Installation mode....54
2.3.8.2 Automating Existing LDAP-UX Domain Installation mode....56
2.3.8.3 Post-installation steps for Existing LDAP-UX Domain Installation mode....56
2.4 Customized installation (setup)....56
2.4.1 Summary of customized installation and configuration steps....57
2.4.2 Planning for your customized installation and configuration....59
2.4.3 Installing LDAP-UX Client Services on a client....64
2.4.4 Configuring your directory....65
2.4.5 Configuring the LDAP-UX Client Services....68
2.4.5.1 Quick configuration....69
2.4.5.2 Custom configuration....73
2.4.5.3 Remapping attributes for services....76
2.4.6 Configuring the LDAP-UX Client Services with SSL or TLS support....79
2.4.6.1 Configuration parameters....79
2.4.6.2 Configuring the LDAP-UX client to use SSL or TLS....79
2.4.6.2.1 Steps to create certificate database files using the certutil utility....80
2.4.6.2.2 Adjusting the peer certificate policy....81

2.4.6.3 SSL/TLS ciphers....82
2.4.7 Configuring LDAP-UX Client Services with NIS publickey support....84
2.4.7.1 HP-UX Enhanced Publickey-LDAP software requirement....84
2.4.7.2 Extending the NIS publickey schema into your directory....84
2.4.7.3 Admin Proxy user....85
2.4.7.3.1 Configuring an Admin Proxy user by using ldap_proxy_config....85
2.4.7.3.2 Password for an Admin Proxy user....85
2.4.7.4 Setting ACI for key management....85
2.4.7.4.1 Setting ACI for an Admin Proxy user....85
2.4.7.4.2 Setting ACI for a user....87
2.4.7.5 Configuring serviceAuthenticationMethod....87
2.4.7.5.1 Authentication methods....87
2.4.7.5.2 Procedures used for configuring serviceAuthenticationMethod....87
2.4.7.6 Configuring Name Service Switch (NSS)....89
2.5 Post-installation configuration tasks....89
2.5.1 Importing name service data into your directory....90
2.5.1.1 Ensure user and group numbers do not collide with those created by a guided New Directory Server mode installation....90
2.5.1.2 Steps to importing name service data into your directory....91
2.5.2 Verifying the LDAP-UX Client Services....92
2.5.3 Enabling AutoFS support....95
2.5.3.1 Automount schemas....95
2.5.3.1.1 New automount schema....95
2.5.3.1.2 The nisObject automount schema....96
2.5.3.2 Attribute mappings....97
2.5.3.3 Configuring NSS....98
2.5.3.4 AutoFS migration scripts....98
2.5.3.4.1 Environment variables....98
2.5.3.4.2 General syntax for migration scripts....99
2.5.3.4.3 The migrate_automount.pl script....99
2.5.3.4.4 The migrate_nis_automount.pl script....101
2.5.3.4.5 The migrate_nisp_autofs.pl script....102
2.5.4 Enabling offline credential caching for authentication when the directory server is unavailable....102
2.5.4.1 How the offline cache works....103
2.5.4.2 Configuring the offline cache....103
2.5.5 Enabling integrated Compat Mode to control name services and user logins....104
2.5.5.1 Overview....104
2.5.5.2 Netgroups in LDAP....105
2.5.5.3 Configuring integrated “compat” mode....105
2.5.5.3.1 Limitations....106
2.5.6 Controlling user access to the system through LDAP....106
2.5.6.1 Using the disable_uid_range flag to prevent access to the local system by unwanted users....106
2.5.6.2 Using the deny_local option to prevent access to the local system by unwanted users....107
2.5.6.3 Configuring PAM_LDAP authentication to ignore specific users....109
2.5.7 Configuring subsequent client systems....112
2.5.8 Downloading the profile periodically....113
2.5.9 Using the r-command for PAM_LDAP....113

3 LDAP Printer configurator support....115
3.1 Overview....115
3.1.1 Definitions....115

3.1.1.1 Printer services....115
3.1.1.2 Printing protocol....115
3.1.1.3 LP printer types....115
3.2 How the LDAP printer configurator works....115
3.3 Printer configuration parameters....117
3.4 Printer schema....117
3.4.1 An example....118
3.5 Managing the LP printer configuration....118
3.6 Limitations of the printer configurator....120

4 Dynamic group support....121
4.1 Overview....121
4.2 Specifying an LDAP URL for a dynamic group....121
4.2.1 Creating an HP-UX POSIX dynamic group....121
4.2.1.1 Step 1: Creating a dynamic group....121
4.2.1.2 Step 2: Adding POSIX attributes to a dynamic group....122
4.2.1.2.1 Adding attributes to a dynamic group using ldapmodify....122
4.2.2 Changing an HP-UX POSIX static group to a dynamic group....124
4.3 Multiple group attribute mappings....124
4.3.1 Examples....125
4.3.2 Group attribute mappings....125
4.4 Number of group members returned....127
4.5 Number of groups returned for a specific user....127
4.6 Performance impact for dynamic groups....128
4.6.1 Enabling/disabling enable_dynamic_getgroupsbymember....128
4.7 Configuring dynamic group caches....128

5 Administering LDAP-UX Client Services....129
5.1 Using the LDAP-UX client daemon....129
5.1.1 Overview....129
5.1.2 ldapclientd....129
5.1.2.1 Starting the client....129
5.1.2.2 Controlling the client....129
5.1.2.3 Client daemon performance....130
5.1.2.4 Command options....130
5.1.2.5 Diagnostics....130
5.1.2.6 Warnings....131
5.1.3 ldapclientd.conf....131
5.1.3.1 Missing settings....131
5.1.3.2 Configuration file syntax....131
5.1.3.2.1 Section details....132
5.1.3.3 Configuration file....137
5.2 Integrating with Trusted Mode....138
5.2.1 Overview....138
5.2.2 Features and limitations....138
5.2.2.1 Auditing....138
5.2.2.2 Password and account policies....139
5.2.2.3 PAM configuration file....139
5.2.2.4 Others....139
5.2.3 Configuration parameter....139
5.3 PAM_AUTHZ login authorization....140
5.3.1 Policy and access rules....140
5.3.2 How login authorization works....140

5.3.3 PAM_AUTHZ supports security policy enforcement....142
5.3.3.1 Authentication using LDAP....142
5.3.3.2 Authentication with secure shell (ssh) and r-commands....142
5.3.4 Policy file....143
5.3.5 Policy validator....144
5.3.5.1 An example of access rule evaluation....144
5.3.6 Dynamic variable support....145
5.3.7 Constructing an access rule in the access policy file....146
5.3.7.1 Fields in an access rule....146
5.3.8 Static list access rule....150
5.3.9 Dynamic variable access rule....151
5.3.9.1 Supported functions for dynamic variables....151
5.3.9.2 Examples....152
5.3.10 Security policy enforcement with secure shell (ssh) or r-commands....153
5.3.10.1 Security policy enforcement access rule....153
5.3.10.1.1 An example of access rules....154
5.3.10.2 Setting access permissions for global policy attributes....154
5.3.10.3 Configuring the PAM configuration file....155
5.3.10.4 Evaluating the directory server security policy....155
5.3.10.5 PAM return codes....155
5.3.10.6 Directory server security policies....156
5.4 Adding a directory replica....158
5.5 Managing users and groups....159
5.5.1 LDAP user and group command-line tools....159
5.5.2 Listing users....161
5.5.3 Listing groups....162
5.5.4 Adding a user or a group....163
5.5.4.1 Adding users....164
5.5.4.2 Examples of adding a user....164
5.5.4.3 Examples of adding a group....166
5.5.4.4 Modifying defaults in /etc/opt/ldapux/ldapug.conf....167
5.5.5 Modifying a user....168
5.5.6 Modifying a group....169
5.5.7 Deleting a user or a group....170
5.5.7.1 Examples....170
5.5.8 Examining the LDAP-UX configuration....171
5.5.8.1 Checking if LDAP-UX is configured....171
5.5.8.2 Listing available templates....172
5.5.8.3 Discovering required attributes....172
5.5.8.4 Displaying configuration defaults....172
5.5.8.5 Displaying the LDAP-UX profile's DN....173
5.5.8.6 Displaying default search base....173
5.5.8.7 Displaying recommended attributes....173
5.5.8.8 Displaying attribute mapping for a specific name service....174
5.6 Managing hosts in an LDAP-UX domain....174
5.6.1 Adding a host....174
5.6.2 Modifying a host....176
5.6.3 Deleting a host....176
5.6.4 Managing IP addresses....177
5.6.5 Managing hosts in groups....178
5.6.6 Classifying hosts....179
5.6.7 Managing process access rights (proxy_is_restricted)....180
5.7 Displaying the proxy user's DN....182
5.8 Verifying the proxy user....182
5.9 Creating a new proxy user....182

5.9.1 Example....182
5.10 Displaying the current profile....182
5.11 Creating a new configuration profile....183
5.12 Modifying a configuration profile....183
5.13 Specifying a different profile for client use....183
5.14 Changing from anonymous access to proxy access....183
5.15 Changing from proxy access to anonymous access....184
5.16 Performance considerations....185
5.16.1 Minimizing enumeration requests....185
5.17 Client daemon performance....185
5.17.1 ldapclientd caching....185
5.17.2 ldapclientd persistent connections....188
5.18 Troubleshooting....189
5.18.1 Enabling and disabling LDAP-UX logging....189
5.18.2 Enabling and disabling PAM logging....189
5.18.3 Directory server log files....190
5.18.4 User cannot log on to client system....190

6 Managing ssh host keys with LDAP-UX....193
6.1 Overview....193
6.1.1 How it works....193
6.1.2 Secure framework....194
6.1.3 Permissions....196
6.1.4 Distributed management (manage from any host)....196
6.2 Setting up the key management domain....196
6.2.1 Host repository....197
6.2.2 Data Location....197
6.2.3 Trust....197
6.2.4 Validating directory server identity....198
6.2.5 Authentication and access control....198
6.2.6 Administrative users....199
6.3 Managing keys in the directory server....200
6.3.1 Configuring ssh and sshd to use LDAP-managed keys....201
6.3.2 Adding keys for HP-UX hosts....201
6.3.3 Adding keys for non-HP-UX hosts or devices....203
6.3.4 Adding keys in a batch....203
6.3.5 Changing keys for HP-UX hosts....204
6.3.6 Changing key size....204
6.3.7 Changing keys for non-HP-UX hosts....205
6.3.8 Revoking or removing keys....206
6.4 Managing key age....206
6.4.1 Setting advisory key expiration dates....207
6.4.2 Key Auditing....207
6.5 Centrally managing ssh configuration....207
6.5.1 Overriding central configuration....209
6.6 Distributing Keys to Non-HP-UX hosts....210

7 Command and tool reference....211
7.1 The LDAP-UX Client Services components....211
7.2 Client management tools....214
7.2.1 create_profile_entry tool....214
7.2.1.1 Syntax....214
7.2.2 create_profile_cache tool....214

7.2.2.1 Syntax....214
7.2.2.2 Examples....214
7.2.3 create_profile_schema tool....215
7.2.3.1 Syntax....215
7.2.4 display_profile_cache tool....215
7.2.4.1 Syntax....215
7.2.4.2 Examples....215
7.2.5 get_profile_entry tool....215
7.2.5.1 Syntax....215
7.2.5.2 Examples....215
7.2.6 ldap_proxy_config tool....216
7.2.6.1 Syntax....216
7.2.6.2 Examples....217
7.3 LDAP user and group management tools....219
7.3.1 Environment variables....219
7.3.2 Return value formats....220
7.3.3 Common return codes....220
7.3.4 ldapuglist tool....223
7.3.4.1 Synopsis....223
7.3.4.2 Options....223
7.3.4.3 Arguments....224
7.3.4.4 Output Format....227
7.3.4.5 Special considerations for output format....228
7.3.4.5.1 Multi-values attributes....228
7.3.4.5.2 Non-POSIX accounts and groups....228
7.3.4.5.3 Encoding of the DN....229
7.3.4.5.4 Passwords....229
7.3.4.6 Specific return codes for ldapuglist....229
7.3.4.7 Limitations....230
7.3.4.8 Examples....230
7.3.5 ldapugadd tool....232
7.3.5.1 Syntax translation....232
7.3.5.2 Synopsis....233
7.3.5.3 Options....233
7.3.5.4 Arguments....234
7.3.5.4.1 Arguments applicable to -D....234
7.3.5.4.2 Arguments applicable to -t passwd....235
7.3.5.4.3 Arguments applicable to -t group....240
7.3.5.5 LDAP UG tool configuration file....241
7.3.5.6 Template files....242
7.3.5.6.1 Template file naming....242
7.3.5.6.2 Default template files....243
7.3.5.6.3 Defining template files....244
7.3.5.6.4 Multi-valued attributes in template files....245
7.3.5.7 Security considerations....246
7.3.5.8 Specific return codes for ldapugadd....246
7.3.5.9 Limitations....247
7.3.5.10 Examples....247
7.3.6 ldapugmod tool....250
7.3.6.1 Synopsis....250
7.3.6.2 Options....250
7.3.6.3 Arguments....251
7.3.6.3.1 Options applicable to -t passwd....253
7.3.6.3.2 Options applicable to -t group....255
7.3.6.4 Warnings....256

7.3.6.5 Specific return codes for ldapugmod....258
7.3.6.6 Security considerations....259
7.3.6.7 Limitations....260
7.3.6.8 Examples....260
7.3.7 ldapugdel tool....261
7.3.7.1 Removing attributes only....261
7.3.7.2 Synopsis....261
7.3.7.3 Options....261
7.3.7.4 Arguments....262
7.3.7.5 Specific return codes for ldapugdel....264
7.3.7.6 Security considerations....265
7.3.7.7 Limitations....265
7.3.7.8 Examples....265
7.3.8 ldaphostmgr tool....266
7.3.8.1 Synopsis....266
7.3.8.2 Options and Arguments....267
7.3.8.3 Object Classes....274
7.3.8.4 Binding to the Directory Server....274
7.3.8.5 Security Considerations....275
7.3.8.6 Usage Notes....275
7.3.8.7 Errors and Warnings....276
7.3.8.8 External Influences....277
7.3.8.8.1 Environment Variables....277
7.3.8.8.2 LDAP-UX Profile....277
7.3.8.9 Limitations....277
7.3.8.10 Examples....277
7.3.8.11 See Also....277
7.3.9 ldaphostlist tool....277
7.3.9.1 Synopsis....278
7.3.9.2 Options and Arguments....278
7.3.9.3 Output Format....282
7.3.9.4 Special Considerations for Output Format....283
7.3.9.5 Binding to the Directory Server....283
7.3.9.6 Errors and Warnings....284
7.3.9.7 External influences....284
7.3.9.7.1 Environment Variables....284
7.3.9.7.2 LDAP-UX Configuration....284
7.3.9.8 Security Considerations....284
7.3.9.9 LDAP-UX Profile....285
7.3.9.10 Limitations....285
7.3.9.11 Examples....285
7.3.9.12 See Also....285
7.3.10 ldapcfinfo tool....286
7.3.10.1 Synopsis....286
7.3.10.2 Options....286
7.3.10.3 Specific return codes for ldapcfinfo....288
7.3.10.4 Examples....289
7.4 LDAP directory tools....292
7.4.1 ldapentry....292
7.4.1.1 Syntax....293
7.4.1.2 Examples....293
7.4.2 ldappasswd....294
7.4.2.1 Syntax....294
7.4.2.2 Examples....294
7.4.3 ldapsearch....295

7.4.3.1 Syntax....295
7.4.3.2 ldapsearch options....295
7.4.4 ldapmodify....296
7.4.4.1 Syntax....296
7.4.4.2 ldapmodify options....296
7.4.5 ldapdelete....297
7.4.5.1 Syntax....297
7.4.5.2 ldapdelete options....297
7.5 Schema extension utility....298
7.5.1 Overview....298
7.5.1.1 Benefits of the schema extension tool....298
7.5.2 How the schema extension utility works....298
7.5.2.1 Operations performed by the schema extension utility....299
7.5.2.2 DTD and XML files used by ldapschema....299
7.5.3 ldapschema (schema extension) tool....301
7.5.3.1 Syntax for ldapschema....301
7.5.3.1.1 Required command options....301
7.5.3.1.2 Additional options (optional)....302
7.5.3.2 Security....303
7.5.3.3 Environment variables....304
7.5.3.4 Examples....304
7.5.3.4.1 An example for querying the schema status....304
7.5.3.4.2 An example for extending the new schema into the directory server....304
7.5.4 Schema definition file....305
7.5.4.1 Sample RFC3712.xml file....306
7.5.4.2 Defining attribute types....307
7.5.4.3 Attribute type definition requirements....308
7.5.4.4 Defining object classes....309
7.5.4.5 Object class definition requirements....310
7.5.4.6 Predefined schema definition files....310
7.5.5 Defining directory-specific information....311
7.5.5.1 Example of defining directory-specific information in the attribute type definition....311
7.5.5.2 Example of defining directory-specific information in the object class definition....312
7.5.6 LDAP directory server definition file....313
7.5.6.1 Example of the directory server definition file....313
7.5.6.2 Defining matching rules....314
7.5.6.3 Defining LDAP syntaxes....314
7.5.7 Mapping unsupported matching rules and LDAP syntaxes....315
7.5.7.1 Examples of alternate matching rules and syntaxes in /etc/opt/ldapux/map-rules.xml....315
7.5.8 Return values from ldapschema....317
7.5.8.1 Schema status messages....317
7.5.8.2 Attribute type status messages....319
7.5.8.3 Object class status messages....322
7.5.8.4 Matching rules status messages....323
7.5.8.5 LDAP syntax status messages....324
7.6 Name service migration scripts....326
7.6.1 Naming context....326
7.6.2 Migrating all your files....326
7.6.3 Migrating individual files....327
7.6.3.1 Migration scripts....327
7.6.3.2 Environment variables....328
7.6.3.3 General syntax for perl migration scripts....328
7.6.4 Examples....328
7.7 Unsupported contributed tools and scripts....330

7.7.1 beq (search) tool....330
7.7.1.1 Syntax....330
7.7.1.2 Examples....330
7.7.2 certutil (certificate database) tool....332
7.7.3 uid2dn (display user's DN) tool....332
7.7.3.1 Syntax....333
7.7.3.2 Examples....333
7.7.4 get_attr_map.pl (get attributemap from profile) tool....333
7.7.4.1 Syntax....333
7.7.4.2 Examples....333

8 User tasks....335
8.1 Modifying passwords....335
8.2 Modifying personal information....336

9 Mozilla LDAP C SDK....337
9.1 Overview....337
9.2 The Mozilla LDAP C SDK file components....337
9.3 Legacy versions of the LDAP SDK....340

10 Support and other resources....343
10.1 Contacting HP....343
10.2 New and changed information in this edition....343
10.3 Related information....345
10.4 Typographic conventions....346

A Configuration worksheet....347

B LDAP-UX Client Services object classes....349
B.1 Profile attributes....349

C Sample /etc/pam.ldap.trusted file configured by setup....353

D Sample /etc/pam.conf file for security policy enforcement....357

E Samples of LDAP-UX configuration files created or modified by autosetup....359
E.1 NSS configuration file after autosetup configuration....359
E.2 PAM configuration file after autosetup configuration....359
E.3 ldapux_client.conf file after autosetup configuration....361
E.4 ldapclientd.conf file after autosetup configuration....363

Glossary....367

Index....369

List of Figures

1-1 A simplified NIS environment....16
1-2 A simplified LDAP-UX Client Services environment....16
1-3 `ldapclientd` and the LDAP-UX Client Services environment....17
1-4 Local start-up file and the configuration profile....19
2-1 Directory information tree (DIT)....28
2-2 LDAP-UX Domain subtrees and ACIs....29
2-3 Example directory structure....61
3-1 Printer configurator architecture....117
5-1 PAM_AUTHZ environment....141
6-1 ssh host key management infrastructure....194
6-2 ssh host key management trust framework....195
8-1 Cannot change passwords on replica servers....335
8-2 Changing passwords on master server with ldappasswd....335
8-3 Sample passwd command wrapper....336

List of Tables

1 Publishing history details....2
1-1 Examples of commands and subsystems that use PAM and NSS....17
2-1 New attributes....32
2-2 New object classes....32
2-3 New Directory Server Installation mode configuration values....48
2-4 Configuration parameter default values....71
2-5 Supported ciphers....83
2-6 Enhanced Publickey-LDAP software requirement....84
2-7 Attribute mappings....97
2-8 Migration scripts....98
4-1 Attribute mappings....126
5-1 Field syntax in an access rule....146
5-2 Global security attributes....156
5-3 Security policy status attributes....156
5-4 Benefits and side-effects for caching....186
7-1 LDAP-UX Client Services components....211
7-2 LDAP-UX Client Services libraries on the HP-UX 11i v2 or v3 PA-RISC machine....213
7-3 LDAP-UX Client Services libraries on an HP-UX 11i v2 or v3 Integrity server machine....213
7-4 Common return codes....220
7-5 Return codes for ldapuglist....229
7-6 Return codes for ldapugadd....246
7-7 Return codes for ldapugmod....258
7-8 Return codes for ldapugdel....265
7-9 Return codes for ldapcfinfo....288
7-10 Supported directory servers....301
7-11 Reserved LDAPv3 directory servers....301
7-12 Directory specific information....311
7-13 Default naming context....326
7-14 Migration scripts....327
9-1 Mozilla LDAP C SDK file components on the PA-RISC machine....337
9-2 Mozilla LDAP C SDK file components on an Integrity server machine....339
9-3 Mozilla LDAP C SDK API header files....340
A-1 LDAP-UX Client Services configuration worksheet....347
A-2 LDAP-UX Client Services configuration worksheet explanation....347

List of Examples

2-1 Sample host entry....30
2-2 Sample user entry....30
2-3 Sample group entry....30
2-4 Sample configuration profile....30
2-5 autosetup: interactive mode with verbose set at the highest level....43
2-6 autosetup: passing two parameters directly in the command line along with a password file....44
2-7 autosetup command for silent mode....44
2-8 autosetup: silent mode....44
6-1 Creating an administrator that has the rights to manage ssh public keys....199
6-2 Extending administrator accounts with posixAttributes....200

1 Introduction

This document describes how to install and configure the LDAP-UX Client Services product on HP-UX platforms. This document is intended for system and network administrators responsible for installing, configuring, and managing the LDAP-UX Client Services. Administrators are expected to have knowledge of the LDAP-UX Client Services Integration product.

NOTE: The document printing date and part number indicate the document's current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The document part number will change when extensive changes are made.

Document updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

You can check for updates of this and related documents at the following website:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX LDAP-UX Integration Software**.

LDAP-UX Client Services simplifies HP-UX system administration by consolidating account and configuration information into a central LDAP directory. The directory can be used as a single source repository for HP-UX authentication, authorization, and user data/account management. The product uses the Lightweight Directory Access Protocol (LDAP) to centralize user, group, and network information management in the LDAP directory.

The LDAP directory can reside on any LDAP-capable directory server, with tier one support provided for the HP-UX Directory Server (HPDS) and Red Hat Directory Server (RHDS), as well as Windows Server 2003 R2 and 2008. A directory server helps globalize authentication and authorization as well as management of users, accounts, and network information, across multiple systems in a large enterprise environment. The Windows Active Directory server integrates the respective HP-UX management functionality with Windows clients.

Information provided in this manual outlines the installation and administration tasks of LDAP directories based on LDAP-UX Client Services and supporting the HP-UX Directory Server 8.1 (or later) and the Red Hat Directory Server 8.0.

For information on the integration of LDAP-UX Client Services with Windows Active Directory, see *LDAP-UX Client Services B.05.00 with Microsoft Windows Active Directory Server Administrator's Guide* at:

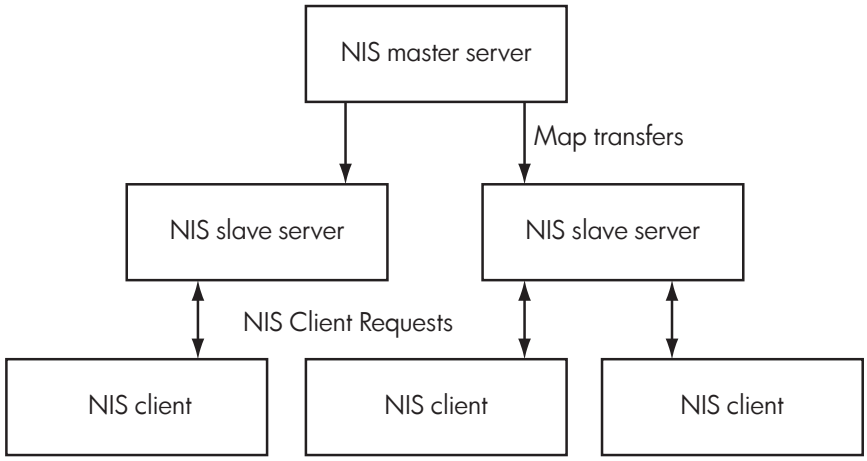
http://www.hp.com/go/hpux-security-docs

Click **HP-UX LDAP-UX Integration Software**.

1.1 Overview of LDAP-UX Client Services

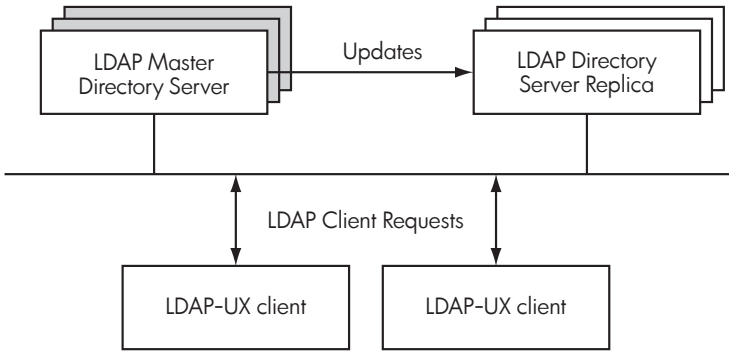
Traditionally, HP-UX account and configuration information is stored in text files, for example, `/etc/passwd` and `/etc/group`. Network Information Service (NIS) was developed to ease system administration by sharing this information across systems on the network. With NIS, account and configuration information resides on NIS servers. NIS client systems retrieve this shared configuration information across the network from NIS servers, and store the retrieved information, as shown in Figure 1-1.

Figure 1-1 A simplified NIS environment



LDAP-UX Client Services improves on this configuration information sharing. HP-UX account and configuration information is stored in an LDAP directory, not on the local client system. Client systems retrieve this shared configuration information across the network from the LDAP directory, as shown below. LDAP adds greater security, scalability, interoperability with other applications and platforms, and less network traffic from replica updates.

Figure 1-2 A simplified LDAP-UX Client Services environment



LDAP-UX Client Services supports the following name service data: passwd, groups, hosts, rpc, services, networks, protocols, NIS publickeys, automount, netgroup. For any additional supported services, see the *LDAP-UX Integration B.05.00 Release Notes*.

1.1.1 How LDAP-UX Client Services works

LDAP-UX Client Services works by providing back-end services for the authentication mechanism provided in the Pluggable Authentication Module (PAM), and and by providing a back-end database for the naming services provided by the Name Service Switch (NSS).

The PAM configuration file `/etc/pam.conf` defines the security mechanisms that are used for authenticating users. Its default values provide the customary operation of the system under both standard HP-UX and trusted systems. It also provides support for controls on individual users. The NSS configuration file `/etc/nsswitch.conf` defines LDAP support for the specified services.

For more information about PAM, see the *pam*(3) and *pam.conf*(4) manpages, and the *Managing Systems and Workgroups: A Guide for HP-UX System Administrators* document at the following location:

http://www.hp.com/go/hpux-core-docs (click **HP-UX 11i v2**)

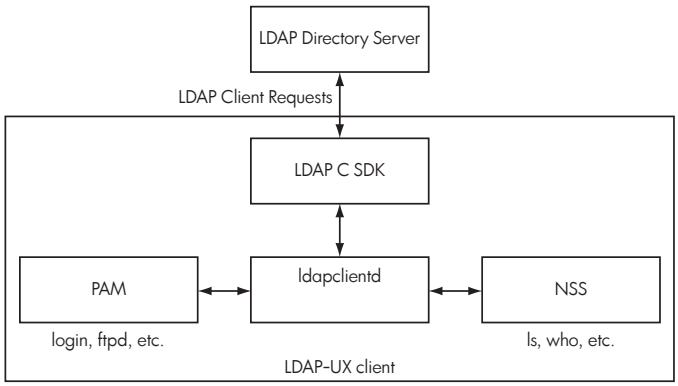
For information on NSS, see the *switch*(4) manpage and the "Configuring the Name Service Switch" chapter in *NFS Services Administrator's Guide*, available at the following location:

www.hp.com/go/hpux-networking-docs

These extensible mechanisms allow new authentication methods and new name services to be installed and used without changing the underlying HP-UX commands. In addition, by supporting the PAM architecture, the HP-UX client is fully integrated into the LDAP environment. The PAM_LDAP library allows the HP-UX system to use the LDAP directory as a trusted server for authentication as well as for centralized password and account policy management. This allows passwords to be stored in any syntax and to remain hidden from view (preventing a decryption attack on the passwords). Because passwords can be stored in any syntax, HP-UX can share passwords with other LDAP-enabled applications, and passwords on LDAP accounts are not subject to an 8-character limitation.

As shown in Figure 1-3, the client daemon `ldapclientd` is the nucleus of the product. It enables LDAP-UX clients to work with LDAP directory servers, and it supports all NSS backend services for LDAP and data enumeration. It also supports PAM_LDAP for authentication and password change.

Figure 1-3 `ldapclientd` and the LDAP-UX Client Services environment



With LDAP-UX Client Services, and `ldapclientd` in particular, HP-UX commands and subsystems can access name service information transparently from the LDAP directory. Table 1-1 (page 17) shows some examples of commands and subsystems that use PAM and NSS. In addition, the `getpwent` and `getgrent` family of system calls obtain user and group information from the directory (for more information, see the *getpwent*(3C) and *getgrent*(3C) manpages).

Table 1-1 Examples of commands and subsystems that use PAM and NSS

Commands that use NSS	Commands that use PAM and NSS
`finger`[1]	`dtlogin`
`grget`[1]	`ftp`
`groups`[1]	`login`
`id`	`passwd`
`listuserslistusers`[1]	`rlogin`
`logins`[1]	`remsh`
`logname`	`su`
`ls`	`telnet`
`newgrp`[1]	
`nslookup`	
`nsquery`[2]	
`pwget` [1]	

Table 1-1 Examples of commands and subsystems that use PAM and NSS *(continued)*

Commands that use NSS	Commands that use PAM and NSS
`who`	
`whoami`	

1 These commands enumerate the entire passwd or group database, which may reduce network and directory server performance for large databases.

2 `nsquery` is a contributed tool included with the ONC/NFS product. For more information, see the *nsquery*(1) manpage.

After you install and configure an LDAP directory and migrate your name service data into it, HP-UX client systems locate the directory from a start-up file. As shown in Figure 1-4, the start-up file tells the client system how to download a configuration profile from the LDAP directory. The configuration profile is a directory entry containing configuration information common to many clients. Storing it in the directory allows you to maintain it in one place and share it among many clients rather than storing it redundantly across clients. Because the configuration information is stored in the directory, each client simply needs to know where its profile is, which is indicated by the start-up file. Each client downloads the configuration profile from the specified directory.

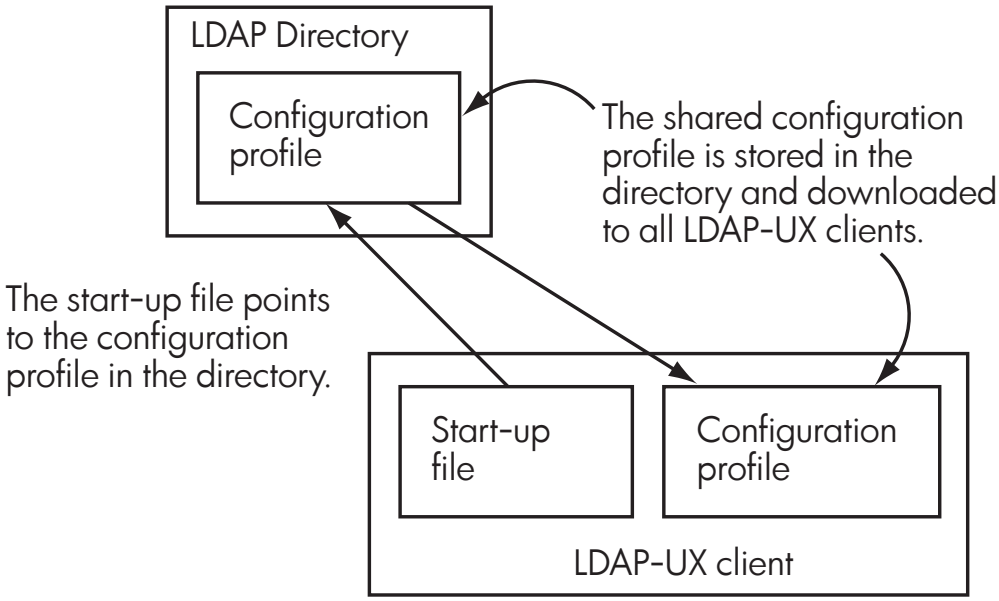
The configuration profile is an entry in the directory containing details on how clients are to access the directory, such as:

- Where and how clients should search the directory for user, group, and other name service information.
- How clients should bind to the directory: anonymously or as a proxy user. Anonymous access is simplest and used most often because most data in the directory server is not considered confidential. However, sometimes directory administrators do not allow anonymous access, in which case a proxy user is created to represent the OS and its users. With a proxy user, the OS can be granted access to the data in the directory server. This identity (user ID and password) is stored in the `/etc/opt/ldapux/pcred` file. Additionally, in some instances, administrators may wish to define an administrator proxy credential. This credential is used to represent administrators of the HP-UX OS, and is often used when NIS public keys are managed in the directory server. The administrator credential (user ID and password) is stored in the `/etc/opt/ldapux/acred` file.

NOTE: The user credentials are stored in the `pcred` and `acred` files, including the password. While these credentials are not visible as plain text, the `pcred` and `acred` files are not encrypted. Access must be restricted to these files.

- Other configuration parameters such as search time limits.

Figure 1-4 Local start-up file and the configuration profile



2 Installing and configuring LDAP-UX Client Services

This chapter describes the decisions you need to make and the steps to install the HP-UX Directory Server or Redhat Directory Server and to configure LDAP-UX Client Services.

2.1 Before you begin: general installation and configuration considerations

Consider the following as you plan your installation and configuration.

- You can choose either of two methods for installing and configuring LDAP-UX Client Services on your HP-UX system:
  - Guided installation — a simple, quick, and automated procedure that sets up a basic installation of the software, which you can customize afterward as needed
  - Customized installation — a screen-based procedure that allows you to customize the software

  The benefits of each type of installation are described in more detail in Section 2.2 (page 22).
- For a customized installation, use the configuration worksheet in "Configuration worksheet" (page 347) to record your decisions and other information you will need later for configuration. For a guided installation, the instructions in this manual advise you to record the significant information. You can do so in any way that is convenient. The information to record is much less in comparison to a customized installation.
- For the latest information, see the *LDAP-UX Integration B.05.00 Release Notes* at:

  http://www.hp.com/go/hpux-security-docs

  Click **HP-UX LDAP-UX Integration Software**.
- The functionality of LDAP-UX requires that a valid LDAP directory server be installed and configured on your host or on another host within your network: either HP-UX Directory Server 8.1 (or later) or Red Hat Directory Server 8.0. If you are using the guided installation (`autosetup`), and have installed HP-UX directory server, the guided installation can configure a directory server instance for you.

NOTE: In this manual, most discussion assume HP-UX Directory Server is the server installed in your environment. HP recommends the HP-UX Directory Server.

You can obtain the latest version of the HP-UX Directory Server (or Red Hat Directory Server) from your local HP sales office or at the following website:

http://www.hp.com/go/softwaredepot

Documentation for the HP-UX Directory Server and Red Hat Directory Server is available at http://www.hp.com/go/hpux-security-docs.

- For advice on how to set up and configure your directory to work with HP-UX, see the white paper *Preparing Your Directory for HP-UX Integration*, available at the following website:

  http://www.hp.com/go/hpux-security-docs

  Click **HP-UX LDAP-UX Integration Software**.

NOTE: This white paper was published before HP-UX Directory Server 8.1 was introduced. However, much of the information continues to be relevant and helpful.

- Most examples in this chapter are based on the HP-UX Directory Server and assume you have some knowledge of this directory and its tools, such as the Directory Server Console and `ldapsearch`. If you have another directory, consult your directory's documentation for specific information.

- For details on how to integrate LDAP-UX Client Services with the Windows Server 2003 R2/2008 Active Directory, see the *LDAP-UX Client Services B.05.00 with Microsoft Windows Active Directory Server Administrator's Guide* at:

  http://www.hp.com/go/hpux-security-docs

  Click **HP-UX LDAP-UX Integration Software**.

- For illustrative purposes, the examples use a base DN of `o=hp.com`.

2.2 Choosing the method of installation: guided or customized

LDAP-UX Client Services releases prior to B.05.00 only provided one installation option, the customized installation using the `setup` program, which is a traditional screen-based program that requires that you run several procedures to set up and configure a new directory server instance after installing the directory server product bundle. This option allows an experienced administrator to customize the software. LDAP-UX Client Services B.05.00 introduces the guided installation using the `autosetup` program, which greatly simplifies the installation and configuration process. This is a simple, quick, and automated procedure that gets you started with a basic implementation of the software, requiring little input other than identifying administrator-level entities. These entities automatically perform privileged configuration tasks for you. The guided installation allows you to install and configure a new instance of an LDAP directory server automatically and configured for use with LDAP-UX. The `autosetup` script creates and configures the new directory server instance with Secure Socket Layer (SSL)/Transport Layer Security (TLS) services enabled. You can customize the software afterward.

Both the `setup` and `autosetup` programs are available in `/opt/ldapux/config`.

The guided installation (`autosetup`) is most advantageous if:

- You prefer simplicity, ease, and quickness of installation.
- You prefer an installation that enables immediate use of LDAP-UX, with minimal input required; `autosetup` automatically provides default values for many parameters that must be provided manually during a customized installation (you can customize parameters later, if desirable).
- You are installing and configuring LDAP-UX for the first time in an environment that has no LDAP directory server instance;`autosetup` detects whether a directory server instance already exists, and if one is not found, the script can set up the directory server for you (if you use the custom installation in an environment that lacks an LDAP directory server, you are responsible for setting up the directory server yourself).
- You want HP-UX host management automatically enabled in the directory server (for more information about host management, see Section 5.6 (page 174)).
- You want secure shell (ssh) host key management automatically enabled (ssh key management is supported in non-Windows environments only); for more information about managing ssh host keys, see "Managing ssh host keys with LDAP-UX" (page 193).

You can also use `autosetup` to install LDAP-UX Client Services into a single Windows domain that has been configured with SSL support. For information about installing and configuring LDAP-UX Client Services into a Windows domain, see the *LDAP-UX Client Services B.05.00 with Microsoft Windows Active Directory Server Administrator's Guide*.

The customized installation (`setup`) is advantageous if:

- You are more experienced and familiar with the product, and you want to manually customize the software during the installation.
- You are installing into an environment that already includes an LDAP directory server, and user and group data has already been installed on that directory server. The guided installation makes assumptions about the location of user, group, and host data that is stored in the directory server (for more information, see "Principles of the LDAP-UX domain" (page 27)). The customized installation allows you to define data location and customized

attribute mapping to specifically match the schema model defined in the existing directory server.

- You want to install the HP-UX host into multiple-domain Windows environment. Guided installation only supports installation into a single windows domain.
- You cannot modify the directory server's schema. In this case, you can deploy using a local-only profile. The local-only profile can also be useful for small deployments and testing purposes. For more information, see Section 2.4.5.1 (page 69).
- You require integration with HP-UX Trusted Mode. The `autosetup` script will not properly configure LDAP-UX on host using Trusted Mode.

2.3 Guided installation (autosetup)

The guided installation greatly simplifies installation of LDAP-UX, and it gives you the option of creating an HP-UX Directory Server instance. Setting up an HP-UX client with LDAP-based security can be accomplished in a matter of moments. The information required for installation is kept to an absolute minimum. For example, the only information required when installing and configuring LDAP-UX into an existing directory server environment is the name of the directory server or the name of the LDAP-UX or Windows domain being joined, as well as the credentials of a user who is permitted to either create a new domain or join an existing one. (The LDAP-UX domain is created by LDAP-UX 5.0 or later installations; it is the collection of users, groups and hosts that can be managed in the LDAP directory server, as defined by the LDAP-UX configuration profile. For more information, see Section 2.3.2 (page 27).) When creating a new directory server, the guided installation can automatically discover the name of the local host and generate the name of the new directory server instance based on the DNS domain. While the guided installation (`autosetup`) is intended to be an interactive utility, you can use command-line options to specify input required by the utility and, in some scenarios, make it completely automated. The command-line options are described in detail in Section 2.3.5 (page 38).

While one of the strengths of LDAP-UX is its ability to integrate into any environment using a variety of configuration options, the guided installation configures LDAP-UX with the most commonly-used installation settings that support a trusted management framework. To assure that the associated directory server is trusted in the security management space for HP-UX, the guided installation requires that the directory server be enabled for SSL support. The guided installation can automatically provision a new HP-UX Directory Server instance with SSL enabled, if one is needed.

The guided installation supports three basic installation scenarios:

- **Installing LDAP-UX to create a new directory server (New Directory Server Installation mode)**: In this scenario, the guided installation creates and provisions a new SSL-enabled instance of an HP-UX Directory Server on the local host, and then configures LDAP-UX to connect to that directory server. (It sets up the PAM configuration file `/etc/pam.conf` and the NSS configuration file `/etc/nsswitch.conf`; samples of these files are included in "Samples of LDAP-UX configuration files created or modified by autosetup" (page 359).) The guided installation prompts for a directory server administration domain name, or if one already exists, the host name and port number of the directory server that manages the existing server administration domain (this directory server is also referred to as the Configuration Directory Server or configuration directory).

NOTE: The directory server administration domain is the domain used for managing the directory servers themselves. In contrast, the LDAP-UX domain is the domain used for managing the data stored by the directory server. It consists of the collection of users, groups and hosts that can be managed in the LDAP directory server. For more information for the variety of domains discussed in this manual, see Section 2.3.3 (page 36).

The guided installation also prompts for the initial credentials used for managing the elements of the directory server and the data managed by that directory server. It configures the

directory server to suit managing an LDAP-UX domain. For more information about the LDAP-UX domain, see Section 2.3.2 (page 27).

In this scenario, the guided installation:

- Configures the directory server with an LDAP-UX schema used for managing users, groups, and hosts. This includes definition of the database indexes based on that schema.
- Defines the initial framework for the directory information tree.
- Defines access control rights for directory server and LDAP-UX domain administration.
- Creates an LDAP-UX configuration profile (based on RFC 4876) that can be used for configuring additional clients. This file defines the LDAP-UX domain contents. For information about this RFC, see:

  http://www.ietf.org/rfc/rfc4876.txt

  For more information about RFCs in general, see the following website:

  http://www.ietf.org/rfc.html

- Provisions HP-UX host information into the directory server, to be used for proxied authentication and ssh key management.
- Creates a certificate authority (CA) and server certificate along with a CA package depot that can be pre-installed on HP-UX clients to be managed in the LDAP-UX domain.

The creating and provisioning of a new directory server instance is supported only with Red Hat Directory Server 8.0 and HP-UX Directory Server 8.1 or later. The guided installation will not create instances of earlier versions of Red Hat Directory Server or Netscape Directory Server.

Instructions for installing LDAP-UX for the first time in an environment without a directory server are described in Section 2.3.6 (page 44).

- **Installing LDAP-UX into an existing directory server environment (Existing Directory Server Installation mode)**: In this scenario, instead of creating a new directory server instance, the guided Installation discovers information about your existing directory server and directory information tree. The existing directory server must be HP-UX Directory Server 8.1 or later, or Red Hat Directory Server 8.0. The guided installation then configures LDAP-UX accordingly. The guided installation requires that the existing directory information tree follow the structure defined in Figure 2-1 (page 28), unless being installed into a Windows domain.

  If the directory server hosts a Windows domain, the guided installation configures the LDAP-UX profile to follow the standard layout and attributes defined for an ADS domain. For a non-ADS domain, the guided installation creates an LDAP-UX configuration profile based on the existing directory information tree, with the defaults defined for an LDAP-UX domain shown in Figure 2-1 (page 28). The guided installation provisions information about the current host into the directory server. (For more information about the directory information tree in an LDAP-UX domain, see Section 2.3.2.1 (page 28).)

  In this scenario, the guided installation prompts for several parameters, depending on the exact circumstances. You will be prompted for the existing directory server's host name (and optionally the port), as well as the bind DN and password of a user who has sufficient privileges to add the local HP-UX host to the LDAP-UX domain. When you specify a remote host where the existing directory server is located, the guided installation cannot validate the identity of the directory server unless a valid domain (CA certificate) or server certificate exists on the local host. If one does not exist, you are given the option of having the guided installation download and install the CA or server certificate (without trust) or, if the server was created by `autosetup`, you can download (from the server to your host) a certificate depot that installs the CA certificate for the LDAP-UX domain.

Instructions for installing LDAP-UX for the first time in an existing directory server environment are described in "Guided installation steps: Existing Directory Server Installation mode" (page 50).

- **Installing LDAP-UX into an existing LDAP-UX domain (Existing LDAP-UX Domain Installation mode)**: In this scenario, LDAP-UX has already been configured in the environment. You can then use the guided installation to join the HP-UX host to an existing LDAP-UX domain or to a Windows ADS domain. The guided installation simply downloads the existing domain configuration (LDAP-UX configuration profile) and registers the host in the domain.

  In this scenario, the guided installation prompts you for similar input as does the preceding scenario, and if you have not pre-installed the CA certificate, you will also be asked if you want to trust the directory server.

  Instructions for installing LDAP-UX into an existing LDAP-UX domain are described in "Guided installation steps: Existing LDAP-UX Domain Installation mode" (page 53).

NOTE: You can install LDAP-UX into an existing LDAP B.04.*xx* environment; however, the hosts search descriptor `serviceSearchDescriptor` in the LDAP-UX configuration profile will likely define an incorrect location for host entries (it should be `ou=hosts`). Host tools expect the correct location for host entries to be defined in the configuration profile. If the location is incorrect, the `ldaphostmgr` tool will add hosts to an incorrect location in the directory tree.

The guided installation (with LDAP-UX B.05.00 or later) configures the profile with the correct location for host entries. If you are installing LDAP-UX into an LDAP-UX environment that has not been set up by the guided installation, ensure that the correct location is specified in the profile (normally, that is `ou=hosts`). To determine the location configured for hosts in the LDAP-UX configuration profile, you can use the following command:

```
/opt/ldapux/bin/ldapcfinfo -t hosts -b
```

If you need to modify the configuration profile, you can modify the `serviceSearchDescriptor` attribute for the `hosts` service. For information about how to modify the LDAP-UX configuration profile, see "Modifying a configuration profile" (page 183).

In all three scenarios, you configure LDAP-UX on the local host for the first time. Scenario 1 introduces the LDAP-UX domain to your organization, creates a directory server and a new LDAP-UX configuration profile, configures your local HP-UX host and joins the host to the LDAP-UX domain. Scenario 2 introduces the LDAP-UX domain to your organization using an existing directory server, creates a new LDAP-UX configuration profile, configures your local HP-UX host and joins the host to the LDAP-UX domain. Scenario 3 configures your local HP-UX host based on an existing directory servers' LDAP-UX configuration profile and joins the host to the existing LDAP-UX domain.

If no valid directory server software is installed on the local system, the guided installation prompts you for the name of an existing remote directory server or Windows ADS domain. If the specified directory server or Windows domain is not found, the guided installation aborts.

2.3.1 What autosetup does

As mentioned, the guided installation (`autosetup`) greatly simplifies the configuration process. The procedure performs numerous activities automatically, with minimal input required from whoever runs the script, including the following:

1. Automatically detects existing directory servers by querying the DNS server of the DNS domain for any registered directory servers, and then tries to connect to the directory server with a search request. If multiple SRV resource records are returned, `autosetup` stops

searching once it makes a successful connection. If a directory server cannot be found by DNS, you will be prompted for the host name and port number for an existing directory server in your environment or asked if you wish to create a new directory server instance on the local host.

2. If you choose to create a new directory server instance on the local host, `autosetup` will create an HP-UX Directory Server instance on the local machine. This directory server instance will be configured with SSL and populated with a framework to support the LDAP-UX domain. For information about the LDAP-UX domain created by `autosetup`, see "Principles of the LDAP-UX domain" (page 27).
3. To guarantee confidentiality and data integrity, `autosetup` uses the StartTLS extended operation on a regular LDAP connection with simple authentication (bind DN and password).
4. To trust the certificate presented by the server, `autosetup` determines whether the local HP-UX host has a certificate database that includes the Certificate Authority (CA) certificate that issues the server certificate.
5. If the CA certificate has not already been pre-installed, to create certificate and key database files (`cert8.db` and `key3.db`), `autosetup` obtains the server certificate from the directory server, and then downloads all the trusted CA certificates published in the directory server. The `autosetup` script places in the `cert8.db` database file the one CA certificate that signed the SSL server certificate of the directory server. The `cert8.db` file stores public keys, while the `key3.db` file stores private keys. A warning message will be displayed to indicate that an un-trusted method is being used to obtain the CA certificate.
6. Because a configuration profile can be shared by LDAP-UX clients, `autosetup` checks for an existing profile entry in the directory server, using a standard profile path (`ou=services,ou=configuration`). If the default profile entry exists, `autosetup` downloads it into an LDIF file (`/etc/opt/ldapux/ldapux_profile.ldif`) and creates a binary profile file (`/etc/opt/ldapux/ldapux_profile.bin`) based on the LDIF file.
7. If the default profile entry does not exist, `autosetup` checks for any other profile entries that might be saved. If any are found, you are prompted to select a configuration profile to download or to create a default profile entry.
8. Before adding the profile entry, `autosetup` determines whether the schema defined in RFC 4876 exists in the directory server. If the schema does not exist, then the script extends the directory server schema. Additionally, `autosetup` will extend the directory server with additional LDAP-UX 5.0 schema and the ssh public key management schema.
9. Creates the start-up file (`/etc/opt/ldapux/ldapux_client.conf`) on the LDAP-UX client system, enabled for TLS support (`enable_startTLS` is set to `1`). A sample of the file is included in Section E.3 (page 361).
10. Creates a new computer account/host entry in the directory server that represents the current HP-UX host. If a host entry already exists with the same name, an `autosetup` prompt asks if the existing entry should be deleted and replaced.
11. Configures the host entry as a proxy user. It stores the encrypted proxy user information in the `/etc/opt/ldapux/pcred` file. The proxy file contains two lines, the proxy user DN on the first line, and the password on the second line.
12. Configures the NSS and PAM_LDAP by modifying the `/etc/pam.conf` and `/etc/nsswitch.conf` files; samples of these files are included in "Samples of LDAP-UX configuration files created or modified by autosetup" (page 359).
13. Modifies the LDAP-UX client daemon configuration file `/etc/opt/ldapux/ldapclientd.conf` to:
    - Enable the LDAP-UX client daemon `ldapclientd` to launch automatically whenever the system is rebooted (`[StartOnBoot]` is defined with `enable=yes`).
    - Set `iproxy_is_restricted=yes` in the `[general]` section, which indicates that the host entry created in step 10 is not privileged. This setting enables additional capabilities provided by the `ldapuglist` and `ldaphostlist` tools.

A sample of the `ldapclientd.conf` file is included in Section E.4 (page 363).

14. Starts the LDAP-UX client daemon (`ldapclientd`) and the central configuration service daemon (`ldapconfd`).

2.3.2 Principles of the LDAP-UX domain

When used for installing LDAP-UX in a non-Windows environment for the first time, the guided installation defines the management framework for, and actually creates, an LDAP-UX domain. An LDAP-UX domain is a collection of users, groups and hosts that can be managed in the LDAP directory server, using the user and host management tools described elsewhere in this document (see Section 5.6 (page 174)).

NOTE: This section does not apply to guided installations of LDAP-UX into a Windows ADS domain. An LDAP-UX domain is not a Windows domain. A Windows ADS domain already defines a directory information tree, information model, and security policy. The LDAP-UX domain defines similar elements.

An LDAP-UX domain is defined by an LDAP-UX configuration profile. All hosts configured to point to the same LDAP-UX configuration profile are considered part of that same domain. The configuration profile follows the standard defined by RFC 4876; as such, it can be used to define the same domain for platforms aside from HP-UX. (For more information about configuring the profile, see Section 5.12 (page 183).) While the guided installation defines this configuration profile automatically, any configuration profile can be considered the basis of an LDAP-UX domain.

The guided installation uses the host management tools to automatically provision into the directory server any relevant information about HP-UX hosts contained in the domain. Creating host entries in the directory server serves at least two purposes:

- As part of the secured framework described in Section 2.3.2.3 (page 33), the guided installation assures that data is protected from anonymous access (anonymous access is defined when a new HP-UX Directory Server instance is created.) Directory server data is available only to known clients. When the OS is acting on behalf of its users, it needs a proxy identity to represent the users of the host. The host entry is used to represent that proxy identity.
- As part of the new ssh key management feature, when the guided installation creates the new host entry it also uploads the host's ssh public keys. This simplifies management of ssh keys in the directory server. With HP Secure Shell A.05.50 or higher, the host entry can be used to assure trust between hosts managed in the domain. For more information about managing ssh key management, see "Managing ssh host keys with LDAP-UX" (page 193).

To assure that the LDAP directory server can be trusted as a secure repository for host users and groups, the identity of the directory server must be validated. Being SSL-enabled (as is required), the directory server can provide that validation with SSL certificates. In addition, through SSL encryption, it can assure that private information such as user passwords are not intercepted while they are in transit.

NOTE: SSL/TLS protocols support a variety of different cryptographic algorithms (ciphers) for use in authentication operations between server and client, certificate transmissions, and session key establishment. If a cipher is found to be flawed and subject to attack, administrators of HP-UX and the directory server would need to know about their vulnerability. Ciphers can be disbled in the directory server. For information about SSL/TLS ciphers and which ones are supported by LDAP-UX, see Section 2.4.6.3 (page 82).

When a new directory server instance is created, the guided installation defines the management framework for the LDAP-UX domain. This framework consists of the following major components:

- **Directory information tree (DIT)**: Defines the hierarchical structure in which different objects in the domain are stored, as described in Section 2.3.2.1 (page 28).

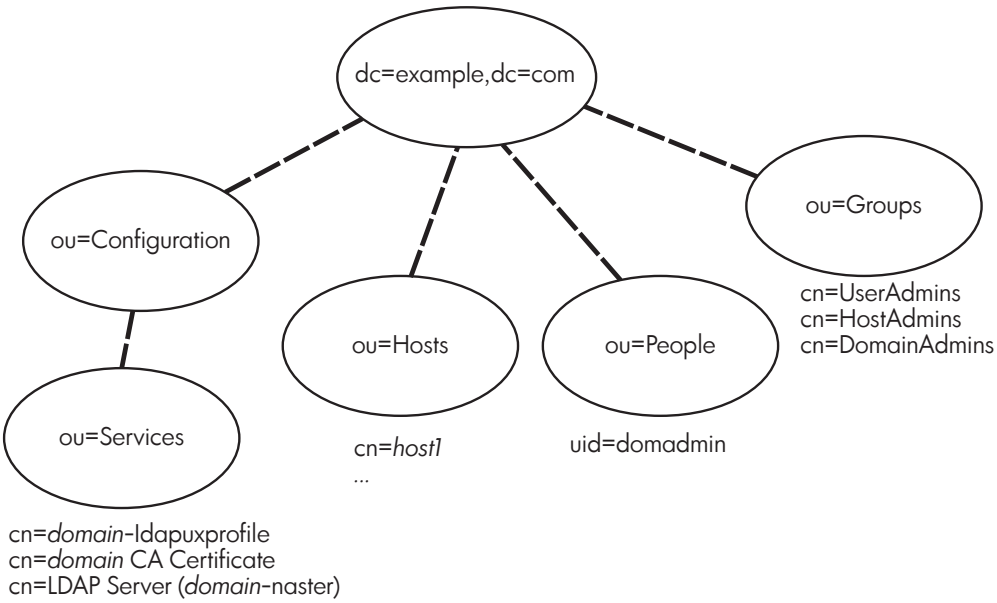
- **Information model**: Defines the types of objects managed in the directory server and the attributes and object classes that represent them, as described in Section 2.3.2.2 (page 29).
- **Security framework**: Defines rights to access and modify data in the DIT, including the definition of three management groups, the Access Control Instructions (ACIs) that grant permissions to each group to manage different objects in the DIT, and general access policies such as which attributes are considered public and private. Details are provided in Section 2.3.2.3 (page 33).

2.3.2.1 Directory information tree (DIT)

When the guided installation creates a new HP-UX Directory Server instance, it creates the foundation for a directory information tree, which is a name space that stores the users, groups, hosts, and configuration in the LDAP-UX domain. This tree can be expanded or altered, as long as appropriate updates are made to the LDAP-UX configuration profile.

To build the DIT, the guided installation creates the root suffix based on the discovered or specified DNS domain. The guided installation uses the domain component syntax to define the root suffix DN, as defined by RFC 2247. Below that, it defines the organizational units to act as containers for the users, groups, hosts, and configuration, as shown in Figure 2-1 (page 28).

Figure 2-1 Directory information tree (DIT)



The subtrees created in the DIT (and shown in Figure 2-1 (page 28)) are:

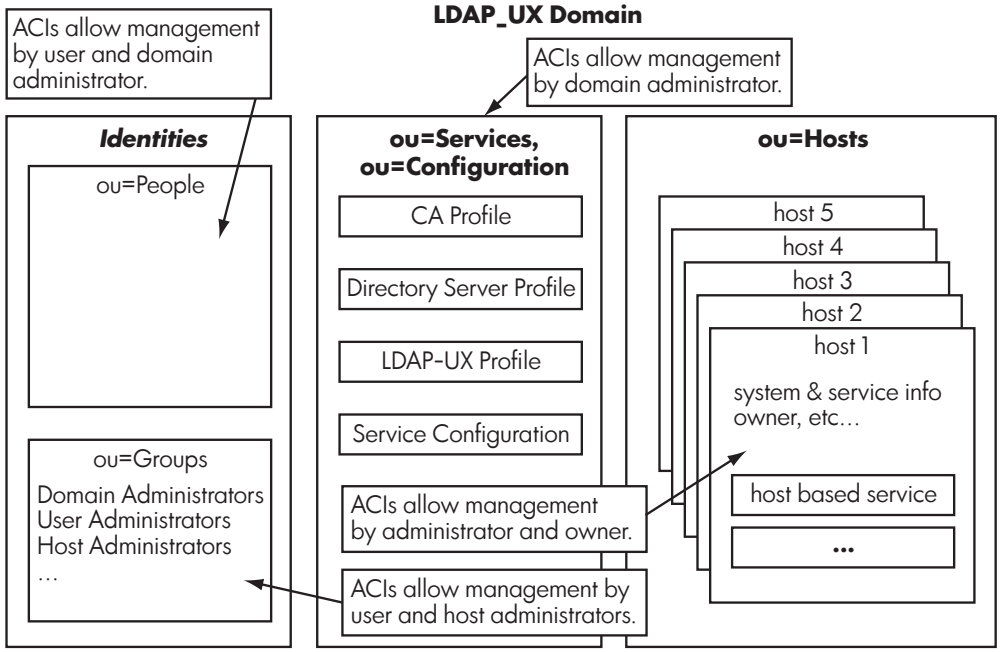
- `ou=People`: Stores all users managed in the LDAP-UX domain. Utilities, such as the LDAP user/group management tools (see Section 7.3 (page 219)) and `ldapentry` (see Section 7.4.1 (page 292)) can be used to manage users and accounts under this subtree. The `ou=People` subtree will be populated with one user, the Domain Administrator. By default, the LDAP-UX Domain Administrator is named `domadmin`. The guided Installation allows this name to be changed.
- `ou=Groups`: Stores all groups managed in the domain. The LDAP user/group management tools and `ldapentry` can also be used to manage these groups. This subtree will be populated with the initial management groups, `cn=UserAdmins`, `cn=HostAdmins`, and `cn=DomainAdmins`. Members of these groups will be granted privileges to manage their related data. For more information about privileges and security in general, see Section 2.3.2.3 (page 33)"Security Framework".
- `ou=Hosts`: Registers information about hosts and devices associated with the LDAP-UX domain. The LDAP host tools `ldaphostmgr` and `ldaphostlist` (see Section 7.3 (page 219)), or `ldapentry` can be used to manage hosts and devices under this subtree. When the guided

installation configures LDAP-UX, it initializes this subtree with the local host's information. Any additional hosts that use the guided installation to configure LDAP-UX will be added under this subtree (joined to the LDAP-UX domain).

- `ou=Configuration,ou=Services`: Stores centrally managed configuration information for LDAP-enabled applications, or information about services available in the domain. The `ldapentry` tool can be used to manage items under this subtree. This subtree will be populated with the LDAP-UX configuration profile and will register the HP-UX Directory Server instance and the CA certificate used in the LDAP-UX domain.

Access control instructions (ACIs) are created (using the `aci` attribute) at the root suffix as well as in the `ou=Hosts`, `ou=People`, and `ou=Groups` subtrees. These ACIs grant administration privileges to the members of the initial groups defined in the `ou=Groups` subtree. Figure 2-2 (page 29) shows the function of the ACIs for each subtree. For more information about access control in the LDAP-UX domain, see Section 2.3.2.3 (page 33).

Figure 2-2 LDAP-UX Domain subtrees and ACIs



2.3.2.2 Information model

As mentioned previously, within the various subtrees defined in the LDAP-UX domain, various types of objects can be managed, including users, groups, and hosts. Management of these objects is based primarily on existing standards (defined by RFCs 2307, 2798 and 4519) and extended schema defined for LDAP-UX. Most manageable information registered for users, groups, and hosts is defined in the RFCs. LDAP-UX includes two additional schemas named ssh_schema and ldapux50.

Information about the manageable objects and how they are defined in the LDAP-UX configuration profile is included in Section 2.3.2.2.1 (page 29). Information about the schema used by LDAP-UX is included in Section 2.3.2.2.2 (page 31).

2.3.2.2.1 Managed objects and how they are defined

For the configuration objects, the LDAP-UX configuration profile created by the guided installation uses the schema defined by RFC 4876. For service objects, the directory server and CA server entries are described by the ldapux50 schema and RFC 4523.

The following examples show entries created for hosts, users, and groups, displayed in LDIF format.

Example 2-1 Sample host entry

```
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
objectClass: top
objectClass: device
objectClass: ldapPublicKey
objectClass: iphost
objectClass: domainEntity
sshPublicKey: ssh-rsa AAAAB3Nza...
sshPublicKey: ssh-dss AAAAB3Nza...
sshPublicKey: 1024 35 140898...
owner: uid=domadmin,ou=people,dc=mydomain,dc=example,dc=com
ipHostNumber: 16.92.96.116
cn: hptem079
```

Example 2-2 Sample user entry

```
dn: uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com
uid: domadmin
givenName: Domain
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
objectClass: posixaccount
sn: Administrator
cn: Domain Administrator
homeDirectory: /home/domadmin
loginShell: /usr/bin/sh
uidNumber: 1095
gidNumber: 1187
```

Example 2-3 Sample group entry

```
dn: cn=HostAdmins,ou=Groups,dc=mydomain,dc=example,dc=com
description: Administrators that are allowed to manage host attributes
objectClass: top
objectClass: groupofuniquenames
objectClass: posixgroup
owner: uid=domadmin,ou=people,dc=mydomain,dc=example,dc=com
uniqueMember: uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com
cn: HostAdmins
gidNumber: 1872
```

When LDAP-UX creates the configuration profile, attributes from RFC 2307 define most of the information model used for users, groups, and hosts. The configuration profile is created mostly with defaults, meaning that the search filters and attributes are based on RFC 2307 recommendations. However, the profile includes a few exceptions that help improve interoperability with other LDAP-enabled applications. The following is a sample profile generated by the guided installation. A summary of the enhancements made to improve interoperability follows the example.

Example 2-4 Sample configuration profile

```
dn: cn=mydomain-ldapuxProfile,ou=Services,ou=Configuration,
    dc=mydomain,dc=example,dc=com
objectClass: top
objectClass: DUAConfigprofile
```

```
objectClass: configurableService
cn: cup-ldapuxProfile
preferredServerList: 192.168.10.20:389
profileTTL: 14400
defaultSearchBase: dc=domain,dc=example,dc=com
bindTimeLimit: 5
authenticationMethod: tls:simple
credentialLevel: proxy
attributeMap: passwd:userpassword=*NULL*
attributeMap: shadow:userpassword=*NULL*
attributeMap: group:userpassword=*NULL*
attributeMap: group:memberUid=uniqueMember member memberUid
attributeMap: passwd:gecos=cn l telephoneNumber
serviceSearchDescriptor: passwd:ou=People,
serviceSearchDescriptor: shadow:ou=People,
serviceSearchDescriptor: group:ou=Groups,
serviceSearchDescriptor: pam:ou=People,
serviceSearchDescriptor: rpc:ou=Services,
serviceSearchDescriptor: protocols:ou=Services,
serviceSearchDescriptor: networks:ou=Services,
serviceSearchDescriptor: hosts:ou=Hosts,
serviceSearchDescriptor: services:ou=Services,
serviceSearchDescriptor: printers:ou=Services,
serviceSearchDescriptor: automount:ou=Services,
serviceSearchDescriptor: netgroup:ou=Groups,
```

The guided installation enhances the configuration profile to improve interoperability with other LDAP-enabled applications in the following ways:

- Most all LDAP-enabled applications use the DN-based membership syntax, defined by the X.500 standards. So, instead of using the `memberUid` attribute as the sole, primary attribute for defining group membership, the guided installation uses the `uniqueMember`, `member`, and `memberUid` attributes by default. In addition, when new members are added to a group (using the LDAP user/group management tools), LDAP-UX uses the `uniqueMember` attribute to define that membership (based on the ordering found in `attributeMap`, which lists a mapping from RFC 2307 attributes to alternate attributes).
- Instead of using the `gecos` attribute to define account details, the `cn` (common name), `l` (location), and `telephoneNumber` attributes are mapped to fill the GECOS field. This eliminates the need to define the `gecos` attribute in the directory server.
- To use common authentication with other LDAP-enabled applications, the `userPassword` attribute is defined as NULL. This means it is not visible to applications on the HP-UX host. But instead, applications use the standardized Pluggable Authentication Module (PAM) framework to perform authentication.

2.3.2.2.2 Domain entity classification schema

The guided installation (and LDAP-UX B.05.00 or later) provides new schema that can be used to manage information about users, groups, hosts, and services in your network. As indicated in Table 2-1 (page 32) and Table 2-2 (page 32), LDAP-UX only uses some of the newly added schema directly by default . The tables describe the full list of new attributes and object classes, and explain how the schema are used. The recommended uses are merely advisory. Each organization can customize usage to suit its unique needs. Table 2-1 (page 32) describes the new attributes.

Table 2-1 New attributes

Attribute name	Description and use
`entityModel`	Describes the model associated with the object. The `ldaphostmgr` tool (with the `-I` option specified) uses this attribute to record the hardware model of the HP-UX host.
`entityVersion`	Represents the version of the associated entity. The `ldaphostmgr` tool (with the `-I` option specified) uses this attribute to record the version of the HP-UX OS on a host.
`entityUsage`	Describes the object's designated usage.
`entityRole`	Represents a role associated with the object. The `ldaphostmgr` tool (with the `-r` option specified) will define this attribute.
`entityFunction`	Represents the function of the associated object.
`entitySecurityLevel`	Represents the security level of the associated object.
`entityType`	Represents the type of the associated object.

`serviceConfigParam`	Defines a configuration parameter for the associated service. Suggested format: *`service-name`*[/*`subsystem`*[/...]]:*`service-specific-configuration-parameter`*
	For example:
	`serviceConfigParam:`
	`ssh/client/ssh_config:strictHostKeyChecking yes`

`serviceType`	Describes the type of service supported by the entity.
`servicePort`	Describes the port of service supported by the entity, typically a TCP socket number.
`entityDomain`	Represents a locally assigned name associated with a management collection. The name may be a translated representation of the `associatedDomain` attribute (RFC 4524) or a name provisioned from an organization-defined procedure. The `entityDomain` value is expected to be unique within the larger management space or at least within the `associatedDomain`.
`cfgGlobalPolicyDN`	Global policy DN to support central configuration service. If this attribute is defined in the configuration profile, the central configuration service (`ldapconfd`) will search the specified entry and download the configuration specified in the `serviceConfigParam` attributes. As of release B.05.00, only HP Secure Shell has configuration handlers to support centralized configuration management. For more information about the central configuration service, see Chapter 6 (page 193).
`sshPublicKey`	Defines an ssh public key for the associated object.

Table 2-2 (page 32) describes the new object classes.

Table 2-2 New object classes

Column Head	Column Head
`networkService`	Contains attributes that describe configurable service objects. It typically extends the iPService objectclass.
`domainEntity`	An object class used to classify objects being managed, such as users, hosts, etc.
`configurableService`	A subset of the networkService object class, it is used to indicate that at least some services provided by the object can be centrally configured.
`ldapPublicKey`	An auxiliary object class that indicates the object has an associated ssh public key.

The general intent of the object classes and attributes described by the schema is to provide a way to help group and classify objects within an organization. For example, suppose an

organization wishes to register printers in the directory server, and the organization has a policy that "Top Secret" documents may only be printed on a restricted set of printers. The organization could have all printers that are eligible for printing "Top Secret" documents created with the `entitySecurityLevel` attribute value set to "Top Secret". The conventions used with the attributes in the preceding table, such as defining acceptable value sets, are entirely up to the policies of the organization. The "Top Secret" printer is just one example of such a convention.

While the usage of the schema as defined in this section is entirely optional and up to the organization, when classifying objects, organizations should consider standardizing key strings. For example, a limited set of key strings should be defined for an object's role, enabling the use of LDAP search filters to easily identify all objects of the same role. For information about how to use the above schema for identifying classes of systems, see "Classifying hosts" (page 179).

Most of the attributes described in Table 2-1 (page 32) are multi-valued. For example, you can define more than one role for an object by specifying the `entityRole` attribute with multiple values.

For more information about the schema described in this section, see the `/etc/opt/ldapux/schema/ldapux50.xml and /etc/opt/ldapux/schema/ssh_schema.xml` files.

2.3.2.3 Security framework

When a new directory server instance is created, the guided installation defines a simple access control framework that provides basic protection for data stored in that directory server.

In most directory server deployments, the general policy is to manage public data while maintaining high data integrity. The guided installation creates access control instructions that instantiate such a policy. Most attributes in the directory server are considered public to the organization in which it is deployed. The guided installation establishes controls such that those attributes can be managed by a limited (privileged) set of individuals. There are exceptions to this policy. For example, the user's password (stored in the `userPassword` attribute) is private. Another exception is that, to protect data integrity, most attributes are modifiable only by a limited set of privileged administrators. However, some attributes may be modified by users themselves. For example, a user may modify his own `userPassword`. Because the base access policy assumes most information is public, at least within the organization, HP recommends that this access control policy be reviewed before you store private or confidential data in the directory server. You can modify the ACIs to suit your organization's unique privacy and integrity requirements. A review of the ACIs created by the guided installation are described in the following subsections. For more specific information about the access control instructions and how to modify them, see the *HP-UX Directory Server administrator guide*.

2.3.2.3.1 Proxy users

One of the primary principles of the base security policy is that, although most information is considered public, the scope of users considered public is limited to those who can authenticate to the directory server. This means that information managed in the directory server subtree is visible only to users who can bind and authenticate to the directory server. This policy is enforced by the following ACI:

```
dn: dc=mydomain,dc=example,dc=com
aci: (targetattr!="userPassword || nisSecretKey")(version 3.0; acl "[ALL:READ:
  NOT-PRIVATTRS] Enable proxied access"; allow (read, search, compare) userdn
  ="ldap:///all";)
```

Basically, this ACI states that if the name of the attribute (any attribute defining managed information) is neither `userPassword` nor `nisSecretKey`, then it is visible to anyone who can bind to the directory server (`ldap:///all`).

To assure that HP-UX hosts can retrieve user, group, and other information from the directory server, the HP-UX OS must bind to the directory server on behalf of the users using the OS. To do this, a proxy entry must be created in the directory server that represents the host and its OS.

This is known as the proxy user. The customized installation requires that you create the proxy user manually. The guided installation automatically creates an entry in the directory server. This user (the host entry) is created with a randomly-generated password. The information is recorded in the `/etc/opt/ldapux/pcred` file.

2.3.2.3.2 Access control rights

To assure that administration rights are limited to specific individuals, access control instructions are placed in the directory server to allow for administrator modification, owner modification, and user self-modification:

- **Administration groups access control rights**: These allow for three levels of administration. Three types of administration groups are created to allow management of data in the directory server:
  - `UserAdmins` allows its members to create, modify, and remove user accounts. This includes the ability to adjust user attributes, including passwords, account numbers, and so forth. Members of this group can also manage groups, including creating, modifying and deleting groups as well as adding and removing group members. The rights for `UserAdmins` are granted with the following ACIs:

```
dn: ou=People,dc=mydomain,dc=example,dc=com
aci: (targetattr = "objectclass || cn || manager || gidNumber || givenName ||
 homeDirectory || homePhone || memberUid || memberURL || memberOf ||  ou || s
 n || uid || uidNumber || uniqueMember || userPassword || userCertificate") (
 target = "ldap:///ou=People,dc=mydomain,dc=example,dc=com")(version 3.0;acl
 "[USERADMIN:ALL:USERATTRS] Allow changes to User attributes by User Administ
 rators";allow (all)(groupdn = "ldap:///cn=UserAdmins,ou=Groups,dc=mydomain,d
 c=example,dc=com");)

dn: ou=Groups,dc=mydomain,dc=example,dc=com
aci: (targetattr = "cn || objectclass || member || uniqueMember || memberUid |
 | gidNumber ")(version 3.0;acl "[USERADMIN:WRITE:USERGROUPATTRS] Allow User
 Administrator Rights to modify group membership";allow (write) (groupdn = "l
 dap:///cn=UserAdmins,ou=Groups,dc=mydomain,dc=example,dc=com");)
```

  - `HostAdmins` allows its members to create, modify, and remove host accounts. This includes the ability to adjust host attributes, including passwords, host names, IP addresses, and so forth. Members of this group can also manage groups, including creating, modifying and deleting groups as well as adding and removing members from these groups. The rights for `HostAdmins` are granted with the following ACIs:

```
dn: ou=Groups,dc=mydomain,dc=example,dc=com
aci: (targetattr = "cn || objectclass || member || uniqueMember") (version 3.0
 ;acl "[HOSTADMIN:WRITE:HOSTGROUPATTRS] Allow Host Administrator Rights to mo
 dify group membership";allow (write) (groupdn = "ldap:///cn=HostAdmins,ou=Gr
 oups,dc=mydomain,dc=example,dc=com");)

dn: ou=Hosts,dc=mydomain,dc=example,dc=com
aci: (targetattr = "objectclass || cn || owner || host || ipHostNumber || ipNe
 tmaskNumber || ipNetworkNumber || ipProtocolNumber || ipServicePort || ipSer
 viceProtocol || sshPublicKey || oncRpcNumber || userPassword || userCertific
 ate" )(version 3.0;acl "[HOSTADMIN:ALL:HOSTATTRS]: Allow changes to host att
 ributes by Host Administrators";allow (all) (groupdn = "ldap:///cn=HostAdmin
 s,ou=Groups,dc=mydomain,dc=example,dc=com");)
```

  - `DomainAdmins` allows its members to have complete control of data managed under the root suffix of the directory server. In other words, members can manage data used by the local host's OS and stored in the LDAP-UX domain. More specifically, this is the data defined by the LDAP-UX configuration profile. Any member of this group is considered a Domain Administrator. By default, the name of the Domain Administrator created by the guided installation is `domadmin`. The rights for `DomainAdmins` are granted with the following ACI:

```
dn: dc=mydomain,dc=example,dc=com
aci: (targetattr = "*")(version 3.0;acl "[DOMAINADMIN:ALL:ALLATTRS]: Allow changes
 by Domain Administrators";allow (all) (groupdn = "ldap:///cn=DomainAdmins
 ,ou=Groups,dc=mydomain,dc=example,dc=com");)
```

- **Owners access control rights**: LDAP-UX 5.0 simplifies demarcating ownership of items in the directory server. Owners are considered any users or members of a group that have a DN in the `owner` attribute of the target entry. Currently, only one type of owner exists: owners of hosts. The rights of these owners are granted with the following ACI:

```
dn: ou=Hosts,dc=mydomain,dc=example,dc=com
aci: (targetattr = "sshPublicKey || ipHostNumber")(version 3.0;acl "[OWNER:ALL
 :HOSTOWNERATTRS]: Allow owner modification of host information";allow (all)
 userattr = "owner#USERDN";)
```

  Based on this ACI, an owner of a host may change a host's IP address or `sshPublicKey`. Modifications for other attributes would require that of a Host or Domain Administrator.

- **Self (user) access control rights**: To enable users to change their own passwords, some rights must be granted to every user. These rights are granted through the following self-modify ACI:

```
dn: dc=mydomain,dc=example,dc=com
aci: (targetattr="carLicense || preferredLanguage || nisSecretKey || nisPublic
  Key || sshPublicKey || userCertificate || userPassword || userSMIMECertific
 ate || facsimileTelephoneNumber || homePhone || homePostalAddress || mobile
 || pager")(version 3.0; acl "[SELF:WRITE:SELFWRITEATTRS] Enable self write f
 or common attributes"; allow (write) userdn="ldap:///self";)
```

  As shown in this example, additional attributes (besides the user password) may be specified to give users control of the associated entities, such as the car license (`carLicense`), preferred language (`preferredLangage`), and so forth.

2.3.2.3.3 SSL/TLS and CA/server certificates

To assure the integrity of data that the directory server delivers to the HP-UX client, some means must be established to validate the identity of the directory server. In addition, the data must be protected in transit between the directory server and the HP-UX client. This is especially critical when the directory server performs authentication for the HP-UX client, as the password of the account being verified is transmitted to the directory server (when SIMPLE authentication is used). To validate the identity of the directory server and encrypt data in transit, the guided installation creates a CA certificate and a server certificate on the HP-UX host where the directory server instance is created. These certificates serve to automatically enable SSL/TLS on the directory server.

To simplify distribution of the CA certificate, the guided installation automatically creates a depot file that can be pre-distributed to other HP-UX clients in the domain before configuring LDAP-UX on them. This process pre-establishes trust with the directory server. During the `autosetup` procedure, you will see a message similar to the following, where *`mydomain.example.com`* is the name of the LDAP-UX domain:

```
================================================================================
NOTE: A CA certificate for the "mydomain.example.com" domain has been created.
This certificate can be pre-installed on HP-UX clients or included as part
of an HP-UX Ignite image.  Installing this CA certificate on host will
pre-establish trust with this directory server.  The depot file for this
CA certificate is found at : /tmp/ca-mydomain.example.com.depot
================================================================================
```

The depot contains one product that, when installed, will install the CA certificate for the LDAP-UX domain on the host. For each domain, a CA certificate should be created, and the product created will be named as follows:

```
# swlist -d -s /tmp/ca-cup.hp.com.depot
# Initializing...
# Contacting target "hpt079"...
#
# Target:  hpt079:/tmp/ca-cup.hp.com.depot
#
```

```
#
# No Bundle(s) on hpt079:/tmp/ca-cup.hp.com.depot
# Product(s):
#
  LDAPUX-MYDOMAIN-CA      A.01.00     LDAP-UX mydomain.example.com domain CA Certificate
```

NOTE: SSL/TLS protocols support a variety of different cryptographic algorithms (ciphers) for use in authentication operations between server and client, certificate transmissions, and session key establishment. If a cipher is found to be flawed and subject to attack, administrators of HP-UX and the directory server would need to know about their vulnerability. Ciphers can be disbled in the directory server. For information about SSL/TLS ciphers and which ones are supported by LDAP-UX, see Section 2.4.6.3 (page 82).

Some organizations may wish to pre-distribute this certificate product by pre-installing it on an Ignite-UX image or on other media that can be used to distribute and install new instances of HP-UX.

As part of generating the server certificate, the guided installation creates a `pin.txt` file to hold the password it uses for retrieving the server certificate's private key. The guided installation requires access to the private key to automatically start up the newly-created directory server. The private key validates the directory server's identity.

The private key is stored in the `/etc/opt/dirsvr/slapd-domain-instanceName/key3.db` file. The `pin.txt` file that holds the private key password is stored in the same directory. (The `instanceName` of the first directory server created on a host will always be *`domain-name-prefix`*`-master`, where *`domain-name-prefix`* is the prefix of the DNS domain name.)

WARNING! The root user, or any user that can bypass file system access controls, can read the `pin.txt` file. Any user that has access to the `pin.txt`, `cert8.db`, and `key3.db` files can use them to impersonate a directory server. Therefore, ensure that you restrict access to the accounts of the root user and users that can bypass file system access restrictions.

For security purposes, you can consider removing the `pin.txt` file and requiring that the private key password be manually entered whenever the directory server is restarted. However, requiring manual password entry at every start-up can have drawbacks. For example, consider the impact for server availability after a reboot or power failure.

The CA certificate generated when the guided installation creates the first directory server (the master instance) is stored in the `/etc/opt/dirsvr/slapd-`*`domain`*`-master/cacert.pk12` file. The password to protect that file is stored in `/etc/opt/dirsvr/slapd-`*`domain`*`-master/pk12-passwd.txt`.

WARNING! Any user that can access the `pk12-passwd.txt` file and the `cacert.pk12` file can create a new directory server with sufficient trust to be considered part of the LDAP-UX domain. Such a user can control what data is visible to the HP-UX hosts. Any host with a server certificate signed by the CA certificate will be considered a trusted directory server. Be sure to restrict access to privileged accounts that can bypass file access restrictions on the local host.

2.3.3 Domains in LDAP-UX environments

The LDAP-UX domain is one of several types of domains discussed in this manual. The following list helps you understand the significance of each domain.

- **LDAP-UX domain** — the realm of users, groups, and hosts defined by the LDAP-UX configuration profile and managed by the LDAP directory server. All hosts configured to point to the same LDAP-UX configuration profile are considered part of that domain. The guided installation creates a LDAP-UX domain when setting up a new directory server environment. In an existing LDAP-UX (B.05.00 or later) environment, the guided installation

joins an HP-UX OS instance into an existing LDAP-UX domain. The guided installation can provision information about hosts in the domain into the directory server. The LDAP-UX domain serves as a focal point for managing hosts, securing data, and in non-Windows AD environments, for simplifying management of ssh host keys.

The guided installation uses the LDAP-UX domain name to define the suffix of the directory tree. For example, if the local host is a member of the `AccountingDept.acme.com` domain, the directory server instance is named `AccountingDept-master` by default. The directory server suffix becomes `dc=AccountingDept,dc=acme,dc=com`. For more information about the LDAP-UX domain, see "Principles of the LDAP-UX domain" (page 27).

- **DNS (Domain Name System) domain** — identifies a specific realm of administrative autonomy, authority, or control in a namespace. DNS assigns a name server to maintain the domain namespace and provide translation services between names and associated Internet Protocol (IP) addresses. The domain name space consists of a tree of domain names.

  The HP-UX host system managed by LDAP-UX may participate in a DNS domain. The DNS domain is often used to register directory servers. The guided installation looks for existing directory servers in the local host's DNS domain. When creating a new directory server, it discovers the DNS domain's name and generates the directory server instance name and suffix from the local host's DNS name.

  LDAP-UX can also be used for host-name resolution similar to DNS.

- **Windows Server domain** — a logical collection of users, groups, and computers running versions of the Microsoft Windows operating system that share a central directory database. This central database (known as Active Directory starting with Windows 2000, and as Active Directory Domain Services starting with Windows Server2003 R2), contains the user accounts and security information for the resources in that domain. Each person who uses computers within a domain receives his or her own unique account, or user name. This account can then be assigned access to resources within the domain. In a domain, the directory resides on computers that are configured as "domain controllers." A domain controller (DC) is a server that manages all security-related aspects in user and domain interactions; it responds to all security authentication requests (logging in, checking permissions, and so forth) within the domain. Each DC has a copy of the Active Directory; changes on one computer are synchronized (converged) among all the DC computers by multi-master replication. Servers joined to the Active Directory that are not domain controllers are called Member Servers.

  LDAP-UX Client Services for Microsoft Windows Active Directory allows integration of user account information into a Microsoft Windows 2003 R2/2008 Active Directory Server.

- **NIS (Network Information Service) domain** — defines the system of programs and data files that HP-UX machines use to collect, collate, and share specific information about machines, users, file systems, and network parameters throughout a network of computers. Traditionally, HP-UX account and configuration information is stored in text files, for example, `/etc/passwd` and `/etc/group`. NIS was developed to ease system administration by sharing this information across systems on the network. With NIS, account and configuration information resides on NIS servers. NIS client systems retrieve this shared configuration information across the network from NIS servers, and store the retrieved information (see Figure 1-1 (page 16)).

  The NIS/LDAP Gateway (ypldapd) is a product bundled with LDAP-UX Integration. This product will allow the directory server to act as a repository for an NIS domain and provide a means to allow for a transition from an NIS domain to a domain managed fully in an LDAP directory server. The LDAP-UX Client Services product improves on this configuration information sharing. HP-UX account and configuration information is stored in an LDAP directory or Windows Active Directory instead of on the local client system. Client systems retrieve this shared configuration information across the network from the LDAP directory (see Figure 1-2 (page 16)). LDAP adds greater security, scalability, interoperability with other applications and platforms, and less network traffic from replica updates.

- **Administration domain (Admin domain)** — for HP-UX Directory Server, a container entry for server groups, with each server group containing directory server instances that are managed by the same Configuration Directory Server. This domain is administered by the Configuration Administrator. Using the hpds-idm-console, the Configuration Administrator can view and manage all the HP-UX directory server instances in this domain. The Configuration Directory Server (configuration directory) is used by the hpds-idm-console to discover and manage information about this domain.

2.3.4 Administrators and managers in the LDAP-UX directory server environment

A variety of administrators and managers may be created and involved in the LDAP-UX environment:

- **Directory Manager** — a unique, powerful user established when a directory server is created. The Directory Manager is the "super user" who typically has the responsibility of repairing and recovering from errors in configuration. The Directory Manager is a special entry that does not have to conform to directory server access control policies. The Directory Manager can correct problems that affect users who do not have access control privileges to do so. There is no directory entry for the Directory Manager user; it is used only for authentication. You cannot create an actual Directory Server entry that uses the same distinguished name (DN) as the Directory Manager DN.

  The LDAP-UX guided installation establishes the Directory Manager for a newly-created directory server as `cn=Directory Manager`, and requests that you set up a password for this user.

- **Configuration Administrator** (also known as the Directory Administrator) — a user responsible for managing the directory servers in the directory server administration domain. This user is the "super user" that manages all Directory Server and Administration Server instances through the Directory Server Console. The default Directory Administrator user name is `admin`. Every Directory Server is configured to grant this user administrative access, thus allowing this user to perform configuration changes.

  Some important differences between the Configuration Administrator and the Directory Manager:

  — The Configuration Administrator cannot create top-level entries for a new suffix through an add operation, neither by adding an entry with the Directory Server Console nor by using the `ldapadd` tool.

  — Password policies do not apply to the Directory Manager but do apply to the Configuration Administrator. However, you can define a separate password policy for the Configuration Administrator with similar rights as the Directory Manager.

  — Size, time, and lookthrough limits do not apply to the Directory Manager but do apply to the Configuration Administrator. However, you can define resource limits for the Configuration Administrator similar to those of the Directory Manager.

- **LDAP-UX Domain Administrator** — a user responsible for managing all data in the LDAP-UX domain. This administrator can add a new HP-UX host to the LDAP-UX domain, create a new administration domain, and manage all HP-UX OS instances in that domain. This user also has privileges to log in to any HP-UX host that is a member of the LDAP-UX domain. The default account name is `domadmin`. An LDAP-UX Domain Administrator is any user who is a member of the `DomainAdmins` group. A subset of the Domain Administrator's privileges are available to users defined as members of the `UserAdmins` and `HostAdmins` groups.

2.3.5 Using the guided installation autosetup command—syntax and options

You can run the `autosetup` script interactively, responding to prompts to provide information. You can pass parameters in the command line to reduce the need for providing input during the

installation. In some cases, you can run the script in silent mode, which requires no user interaction during the installation.

To run the script interactively, simply enter the `autosetup` command as is. The script prompts you for the minimal information required. To reduce user interaction during the installation, you can pass parameters by specifying options in the command line. In addition to these options, you can define environment variables with pre-defined parameter settings; ultimately, this enables you to run the installation without any manual intervention required. This section describes the command-line options and environment variables.

The syntax for the `autosetup` command line is:

`autosetup` [*option1 option1-value* [*option2 option2-value*] ...]

The options are described in Section 2.3.5.1 (page 39).

For detailed information about how to perform the guided installation and how `autosetup` configures the LDAP-UX environment, see the following sections:

- "Guided installation steps: New Directory Server Installation mode" (page 44)
- "Guided installation steps: Existing Directory Server Installation mode" (page 50)
- "Guided installation steps: Existing LDAP-UX Domain Installation mode" (page 53)

NOTE: When configuring and setting up LDAP-UX, you will likely be prompted for credentials of an administrator. If you are asked to enter the credentials (password) of a user, make sure that the connection between your client and the HP-UX system (where you are running `autosetup`) is secured and not subject to network eavesdropping. One option to protect such communication may be to use the ssh protocol when connecting to the HP-UX host being configured.

2.3.5.1 autosetup options

The following options can be specified on the command line.

`-C` *`computer_acct_container`*
: Valid only for LDAP-UX installations with Windows Server ADS, this is the computer account container. It is ignored if specified with an LDAP-UX installation for a directory server environment.

`-D` *`privileged_user_DN`*
: When creating a new directory server, or setting up a new LDAP-UX environment with an existing directory server, this typically specifies the Directory Manager's distinguished name (DN) (for the latter scenario, it can be at minimum any user who has sufficient privileges to update the schema on the directory server). When configuring LDAP-UX in an existing LDAP-UX domain, this can be the DN of any member of the `DomainAdmins` or `HostAdmins` groups — host administrators that have privileges to add hosts to the domain (but more limited privileges than the Directory Manager) — or any user given sufficient privileges by the directory server administrator.

  This is the bind DN that LDAP-UX clients use for accessing the HP-UX Directory Server. The default is `cn=Directory Manager` when creating a new LDAP-UX domain, and `uid=domadmin,ou=people,`*`domainBaseDN`* when joining an existing one. An example of a setting for this variable is `uid=domadmin,ou=people,dc=document,dc=hp,dc=com`.

`-b` *`search_base`*	Specifies the base DN for which search operations should be performed for an existing LDAP-UX domain, or the base DN used when creating a new LDAP-UX domain; for example, `dc=lab,dc=acme,dc=com`. Typically, set the base DN to the directory's suffix value. Because the directory suffix is equal to the root, or topmost, entry in the directory, this causes all searches to begin from the directory's root entry. If not specified, the default when creating a new directory server instance will be generated using the local DNS domain name. With an existing directory server, the default is discovered from the list of default naming contexts on the existing directory. This option only applies for LDAP-UX installations in a directory server environment. If specified with an LDAP-UX Windows Server AD installation, this variable is ignored.
`-h` *`host_name`*	Specifies the host name (or IP address), and optionally the port number, of an existing directory server that contains an existing LDAP-UX domain or of a directory server that is to be configured for LDAP-UX support. The default host name is the fully-qualified DNS domain name of the host; the default port is 389. Specify in one of the following formats: ***`host-name`*** ***`host-name:port`*** ***`ip-address`*** ***`ip-address:port`***
`-j` *`password_filename`*	Specifies a file that includes the password for the user specified with the `-D` option. Specifying this file enables `autosetup` to run without prompting you for the password; thus, it enables you to perform the guided installation in silent mode.
`-p` *`ds_port_id`*	Specifies the number of the port that clients use for accessing the directory server. This is the port clients access to search the directory server for entries. The default is port number 389. You need not specify the port if it has already been specified with the host name.
`-s` *`ds_sslport_id`*	Specifies the number of the directory server SSL port for accessing the directory server when SSL options are used. The default is SSL port 636.
`-v` *`n`*	Specifies verbose level for debugging purposes, with *`n`* specifying one of the following: 0 (turns off verbose mode), 1, 2, or 3 (specifies the highest level of verbosity).
`-x` *`domain_name`*	When configuring LDAP-UX in a directory server environment, this option specifies the LDAP-UX domain name; for example, `accounting.acme.com`. The specified name will be used to define the suffix of the new directory server's directory tree. If you do not specify the domain name, this tool obtains it from DNS by executing `nslookup` on the host name of the local system.
`-q`	Specifies that `autosetup` be invoked in silent (quiet) mode. Silent mode runs without user intervention. Instead

of prompting for input for parameters such as the Configuration Administrator or Domain Administrator, it uses the default values and any values specified with command-line options or environment variables. If values are not given for any required parameters that do not have defaults, silent mode will abort. For this reason, silent mode is not valid when installing LDAP-UX on a host for the first time (creating a new directory server requires user intervention).

NOTE: The following parameters cannot be specified with options in New Directory Server Installation mode, but they can be set with the indicated environment variables. For more information about the environment variables, see Section 2.3.5.2 (page 41).

Short name of the directory server Configuration Administrator

Use `DS_ADMIN_NAME`

Password of directory server Configuration Administrator

Use `DS_ADMIN_PASS`

Host name (and optionally, port number) of the Administration Server for the administration domain the directory server will join

Use `DS_ADMIN_SERVER`

Port number of the directory server's Administration Server

Use `DS_ADMIN_PORT`

Instance name of new directory server

Use `DS_INSTANCE_NAME`

Name of the LDAP-UX Domain Administrator

Use `LDAP_DOMAIN_ADMIN`

Password of the LDAP-UX Domain Administrator

Use `LDAP_DOMAIN_ADM_PASSWD`

2.3.5.2 autosetup environment variables

The following environment variables can be used with `autosetup`.

`DS_ADMIN_NAME` — Sets the short name of the directory server Configuration Administrator, responsible for managing the directory servers in the directory server administration domain. The default short name is `admin`. No command option exists for passing this name on the command line. Only used in New Directory Server Installation mode (installing LDAP-UX for the first time).

`DS_ADMIN_PASS` — Sets the password for the directory server Configuration Administrator, responsible for managing the directory servers in the directory server administration domain. Only used in New Directory Server Installation mode (installing LDAP-UX for the first time). No command option exists for passing this password on the command line.

`DS_ADMIN_PORT` — Sets the port number of the directory server's Administration Server, which manages the directory server administration domain. . The default port number is 9830. You need not specify the port if it has already been specified with the host name. No command option exists for passing this port number

	on the command line. Only used in New Directory Server Installation mode (installing LDAP-UX for the first time).
DS_ADMIN_SERVER	Sets the host name (or IP address), and optionally the port number, of the directory server's Administration Server, the server that manages the directory server administration domain that the new directory server will join. Only used in New Directory Server Installation mode (installing LDAP-UX for the first time). The default port is 389. Specify in one of the following formats: ***host-name*** ***host-name:port*** ***ip-address*** ***ip-address:port***
DS_INSTANCE_NAME	Sets the name of the newly created directory server instance. This is the only way to pre-set the instance name. If this variable is not used, `autosetup` automatically generates the instance name in the format *`dns-prefix`*`-master`, where *`dns-prefix`* is the prefix of the local host's DNS domain name. For example, if the local host's DNS domain name is `west.acme.com`, the generated instance name would be `west-master`. Only used in New Directory Server Installation mode (installing LDAP-UX for the first time).
LDAP_BASEDN	Sets the search base DN; for example, `dc=lab,dc=acme,dc=com`. A search operation is performed on the base DN, the DN of the entry and all entries below it in the directory tree. Typically, set the base DN to the directory's suffix value. Because the directory suffix is equal to the root, or topmost, entry in the directory, this causes all searches to begin from the directory's root entry. Equivalent to using the `-b` option in the command line.
LDAP_BINDDN	When creating a new directory server, this specifies the Directory Manager's distinguished name (DN). When installing and configuring LDAP-UX in an existing directory server environment or LDAP-UX domain, this is the DN of any member of the `DomainAdmins` group — host administrators that have privileges to add hosts to the domain (but more limited privileges than the Directory Manager). This is the bind DN that LDAP-UX clients use for accessing the HP-UX Directory Server. The default is `cn=Directory Manager`. An example of a setting for this variable is: `LDAP_BINDDN=uid=domadmin,ou=people,dc=document,dc=hp,dc=com` Equivalent to using the `-D` option in the command line.
LDAP_BINDCRED	Sets the password for the user defined by `LDAP_BINDDN`. Equivalent to using the `-j` option in the command line (except this command-line option specifies a file containing the password).
LDAP_DOMAIN_ADMIN	Sets the name of the LDAP-UX Domain Administrator, who is responsible for managing all data in the LDAP-UX domain and can create a new administration domain and register all directory servers in that domain. If this variable is not defined for use with `autosetup`, the default is `domadmin`. No

	command option exists for passing this name on the command line. This variable only applies for LDAP-UX installations when creating a new directory server environment. If specified with an LDAP-UX Windows Server AD installation, this variable is ignored.
LDAP_DOMAIN_ADM_PASSWD	Sets the password for the LDAP-UX Domain Administrator. No command option exists for passing this password on the command line. This variable only applies for LDAP-UX installations when creating a new directory server environment. If specified with an LDAP-UX Windows Server AD installation, this variable is ignored.
LDAP_DOMAIN	Sets the LDAP-UX domain name; for example, `accounting.acme.com`. Equivalent to using the `-x` option in the command line. If the search base is not specified with the `-b` option and with the `LDAP_BASEDN` variable, the `LDAP_DOMAIN` variable determines the search base. If you do not specify the domain name, this tool obtains it from DNS by executing `nslookup` on the host name of the local system. The `LDAP_DOMAIN` variable only applies for LDAP-UX installations in a directory server environment. If specified with an LDAP-UX Windows Server AD installation, this variable is ignored.
LDAP_HOSTPORT	Sets the host name (or IP address), and optionally the port, of an existing directory server that is to be configured for LDAP-UX support. Equivalent to using the `-h` and `-p` options in the command line. The default host name is the fully-qualified DNS domain name of the host; the default port is 389. Specify in one of the following formats: ***host-name*** ***host-name:port*** ***ip-address*** ***ip-address:port***
LDAP_SSLPORT	Sets the SSL port of the directory server to be created or, if one already exists, to be configured for LDAP-UX support.. The default is 636.

2.3.5.3 autosetup command examples

The following are examples showing how to run `autosetup` with command-line options:

Example 2-5 autosetup: interactive mode with verbose set at the highest level

```
# autosetup -v 3
```

This command runs `autosetup` interactively, with verbose set at the highest level.

Example 2-6 autosetup: passing two parameters directly in the command line along with a password file

```
# autosetup -D "cn=Directory Manager" -j /tmp/jfile -x document.hp.com
```

This command specifies the Directory Manager and a file that includes the password required for the Directory Manager of the directory server being created. The command also specifies the LDAP-UX domain name. When you invoke `autosetup`, you will not be prompted for these parameters.

Example 2-7 autosetup command for silent mode

```
# autosetup -h blipsa01 -D "uid=phil,ou=people,dc=document,dc=hp,dc=com" -j /tmp/jfile -q
```

This command specifies the host name of the directory server that contains the LDAP-UX domain to be joined. In addition, it specifies the bind DN (the user who has privileges to add the host to the LDAP-UX domain) and a file that includes the password for this user.

Example 2-8 autosetup: silent mode

```
# autosetup -q
```

This command invokes silent mode. It can be used in any scenario in which user intervention is not required.

2.3.6 Guided installation steps: New Directory Server Installation mode

This section explains how to install LDAP-UX into a new environment and create a new HP-UX Directory Server for managing the LDAP-UX data. Section 2.3.6.1 (page 45) shows how to perform the guided installation interactively, explaining step-by-step how to respond to each prompt for user input. Section 2.3.6.2 (page 49) shows how to run a completely-automated guided installation by specifying all required user input in the command line.

NOTE: If you are planning a first-time deployment of managing user and group data in the directory server, HP suggests that you devise a strategy to avoid UID number and GID number overlap. Most likely, you will need to continue managing some accounts that are local to the hosts in the LDAP-UX domain. Often the root user, and sometimes application accounts (such as `www` for the `httpd` process) remain managed in the local `/etc/passwd` file. Devise a convention establishing a range for UID numbers and one for GID numbers such that accounts and groups in LDAP do not conflict with those on the local hosts. For example, accounts in LDAP could all have UID numbers greater than 1000, while accounts on local hosts would be restricted to UID numbers less than 1000.

For information about ensuring that user and group numbers to be migrated or imported into a new directory server do not collide with the ones created by the guided installation, see Section 2.5.1.1 (page 90).

NOTE: When configuring and setting up LDAP-UX, you will likely be prompted for credentials of an administrator. If you are asked to enter the credentials (password) of a user, make sure that the connection between your client and the HP-UX system (where you are running `autosetup`) is secured and not subject to network eavesdropping. One option to protect such communication may be to use the ssh protocol when connecting to the HP-UX host being configured.

2.3.6.1 Interactively running New Directory Server Installation mode

To interactively install LDAP-UX and create a new HP-UX Directory Server for your LDAP-UX environment, follow these steps. Before you begin, make sure you have installed the HP-UX Directory Server product on the local host.

1. Log in as root and run the `autosetup` command, as shown in the following example:

   ```
   # /opt/ldapux/config/autosetup
   ```

2. The script detects whether a registered LDAP-protocol directory server instance exists in the local DNS domain. You are creating a new LDAP-UX environment that needs a new directory server, so a directory server is not found, as indicated. The first prompt gives you several options. To run the installation so that it sets up a new directory server, press **Return**, as shown:

   ```
   Scanning DNS domain west.acme.com for any registered LDAP directory servers
       - No directory servers found.

   Please enter the host name and port number of a directory server,
   a Windows domain name, or press Return to create a new directory
   server on this host: Return
   ```

3. The script begins creating a new directory server instance on the local host. It creates the Directory Manager root DN as `cn=Directory Manager` and prompts you to create a password and to re-enter the password to confirm (the password is hidden):

   ```
   The directory server requires a "super-user" ID.  This ID has all
   privileges (is not subject to any access control) on the directory server
   and the name is set as "cn=Directory Manager".  Please enter a password
   for this user.

   Please enter the "cn=Directory Manager" password: [password not displayed] Return
   Please re-enter the "cn=Directory Manager" password: [password not displayed] Return
   ```

   As indicated, the Directory Manager has all privileges and is not subject to directory server access control policies. The Directory Manager is a unique, powerful entry that is typically used to repair and recover from errors in the configuration. The Directory Manager can correct problems that affect users who do not have access control privileges for doing so. There is no directory entry for the Directory Manager user; it is used only for authentication.

You cannot create an actual Directory Server entry that uses the same distinguished name (DN) as the Directory Manager DN. For more information about the Directory Manager and other administrators, see Section 2.3.4 (page 38).

4. The script asks whether you want to manage the new directory server in an existing HP-UX Directory Server administration domain (Admin domain) or whether you want to create a new directory server administration domain. To direct the installation to set up the new directory server in an existing administration domain, specify the host name and optionally the port number of the Administration Server that manages the existing domain, such as `east.acme.com:389`. To create a new directory server administration domain on the local host, press **Return**. In the following example, the user opts to create a new domain. (For more information about this and other domains in the LDAP-UX environment, see Section 2.3.3 (page 36).)

```
HP-UX Directory Server supports management of multiple directory server
instances under one administration domain.  Would you like to manage this
directory server in an existing HP-UX Directory Server administration domain?
If so, enter the host name and optionally the port of the directory server
that manages that topology (for example, acme.bus.com:389).  Or, to create
a new directory server administration domain, simply press Return.
(hostname[:port]|Return): Return
```

5. Enter the short name of the Configuration Administrator required to manage the directory servers and the new directory server administration domain. The default short name is `admin`. In this example, the user takes the default.

```
A Configuration Administrator is required to manage the
directory servers and the administration domain.  This user has
configuration and administration privileges on all directory servers
in the administration domain, but is subject to access control
(unlike the Directory Manager).  Please Enter the short name of the
Configuration Administrator (typically "admin") [admin]: Return
```

   The Configuration Administrator (also known as the Directory Administrator) is the “super user” who has configuration and administration privileges for all directory servers in the managed domain, and so has the power to modify directory server configurations and to create, start, and back up any of the directory servers in the managed domain. The Configuration Administrator has fewer access privileges to the directory server than does the Directory Manager.

6. Create a password for the new Configuration Administrator and re-enter it to confirm:

```
Please enter the password for the Configuration Administrator.
Please enter the "admin" password: [password not displayed] Return
Please re-enter the "admin" password: [password not displayed] Return
```

7. Specify the name of the LDAP-UX domain that is to be created; this is the domain that distinguishes data managed by the new directory server from data managed by other director servers in other domains. The default is the host's DNS domain name. In this example, the default is taken.

```
Please enter the domain name that can be used to distinguish data managed on
this directory server from data managed by other directory servers in other
domains.  This will be used to define the suffix of the directory tree.
Please enter the domain [west.acme.com]: Return
```

8. Specify the name of the LDAP-UX Domain Administrator. The LDAP-UX Domain Administrator is responsible for managing all data in the LDAP-UX domain and can create a new administration domain (created in an earlier step) and register all directory servers in that domain. The default is `domadmin`.

```
An LDAP-UX domain administrator is used to manage all data within
the LDAP-UX domain.  The domain administrator has fewer privileges than the
Directory Manager or Configuration Administrator. This account will be the
primary account used to manage data within the directory server, or its
privileges can later be distributed to other users.  This account should
typically be associated with an individual and may be named as such.  The
account name should be 8 characters or less, since this account can be used on
the HP-UX OS.
Enter Domain Administrator's account name [domadmin]: Return
```

9. Enter the password for the LDAP-UX Domain Administrator and then re-enter it to confirm:

```
Please enter the "domadmin" password: [password not displayed] Return
Please re-enter the "domadmin" password: [password not displayed] Return
```

The installation now begins, followed by other related tasks; `autosetup` displays the progress and results, as in the following example.

NOTE: For future reference, be sure to record the information displayed. To record this information, you can use Table 2-3 (page 48) . The table also describes the parameters that were configured in the preceding example.

```
Creating new directory server instance in local host...

Creating directory server master instance "west-master". Please wait ...

Successfully created master instance with the following parameters:
        Instance name:  west-master
            Host name:  acctl053.west.acme.com
          Server port:  389
           Admin  URL:  http://acctl053.west.acme.com:9830
             SSL port:  636
          Domain name:  west.acme.com
        Domain suffix:  dc=west,dc=acme,dc=com
        Domain  Admin:  domadmin

* Generating a self-signed CA Certificate "WEST CA Certificate" ... completed.
* Generating a server certificate "west-master Certificate" ... completed.
* Enabling SSL on directory server instance west-master ... completed.
* Restarted the Directory Server instance west-master.
* Created directory server subtree.
* Added Domain and Host Administrator user/groups to the directory server.
* Created Domain Administrator account : "domadmin".
* Extended directory server schemas.
* Registered CA and server certificates in directory server.

================================================================================
NOTE: A CA certificate for the "west.acme.com" domain has been created.
This certificate can be pre-installed on HP-UX clients or included as part
of an HP-UX Ignite image.  Installing this CA certificate on host will
pre-establish trust with this directory server.  The depot file for this
CA certificate is found at : /tmp/ca-west.acme.com.depot

================================================================================


Setting up the LDAP-UX client using the newly created directory server.
Loading CA certificate from directory server to local host ... done.
* Extending schemas ... done.
No LDAP-UX Configuration Profile was found.  Creating a new one.

* Downloading profile from DS ... done.
* Configuring ldapux_client.conf ... done.
* Provisioning LDAP-UX Client information into the Directory Server ... done.
* Setting up proxy user ... done.
* Configuring "/etc/nsswitch.conf" and "/etc/pam.conf" to use ldap ... done.
* Starting ldapclientd daemon ... done.
* Starting ldapcconfd ... done.
```

```
LDAP-UX was successfully configured.
```

As indicated in the guided installation log, the guided installation configures LDAP-UX and starts the LDAP-UX daemon (`ldapclientd`) and the central configuration service (`ldapconfd`). For more information about the files configured by autosetup, see "Samples of LDAP-UX configuration files created or modified by autosetup" (page 359). For more information about the central configuration service, see Chapter 6 (page 193).

For more information about the CA and server certificates that are registered in the directory server, see Section 2.3.2.3.3 (page 35).

Table 2-3 (page 48) lists the items configured by the guided installation and gives the values displayed in the example in Section 2.3.6.1 (page 45), in which no parameters were specified in the command line.

Table 2-3 New Directory Server Installation mode configuration values

Parameter	Value in example	Value for your installation
Instance name	`west-master` (if the instance name is not predefined in an environment variable (`DS_INSTANCE_NAME`), as described in Section 2.3.5.2 (page 41)), it is automatically generated as in this example, based on the prefix of the local host's DNS domain name `west.acme.com` (and adding `-master` for the first directory server created in the domain)	
Host name	`acct1053.west.acme.com` (when creating a new directory server, the host name prefix is based on the local computer name (acct1053); the remainder of the host name is based on the DNS name)	
Server port	`389` (default)	
Admin URL	`http://acct1053.west.acme.com:9830` (generated from the computer name and DNS domain name, this is the directory server's Administration Server location, which is used for the Directory Server Console, for example)	
SSL port	`636` (the default port number for LDAP with TLS/SSL)	
Domain name	`west.acme.com` (this is the LDAP-UX domain name; unless specified, it is generated automatically from the DNS domain name)	
Domain suffix	`dc=west,dc=acme,dc=com` (the LDAP-UX domain suffix was generated automatically from the domain name)	
Domain Admin	`domadmin` (the default for the Domain Administrator)	

2.3.6.2 Automating New Directory Server Installation mode

To install LDAP-UX for the first time on a host and create a new directory server, you must run the script interactively to indicate at minimum, when prompted, that you want a new directory server created. You can use command-line options and environment variables to completely automate the rest of the procedure. In the example provided in this section, the following environmental variables are defined for all the parameters needing input. Certain parameters cannot be provided by command-line options.

- `LDAP_BINDDN="cn=Directory Manager"`
- `LDAP_BINDCRED="dmdontforget"`
- `LDAP_DOMAIN_ADMIN="domadmin"`
- `LDAP_DOMAIN="west.acme.com"`
- `LDAP_DOMAIN_ADM_PASSWD="4getmeknot"`
- `DS_ADMIN_NAME="admin"`
- `DS_ADMIN_PASS="4getmenot"`

Running `autosetup` results with the following installation. As shown (in **bold**), user intervention is required only twice after the procedure starts.

```
# ./autosetup
Scanning DNS "west.acme.com" domain for any registered LDAP directory servers...

No directory server found.

Please enter the host name and port number of a directory server
[hostname:port], a Windows domain name, or press Return to create
a new directory server on this host: Return

HP-UX Directory Server supports management of multiple directory server
instances under one administration domain.  Would you like to manage this
directory server in an existing HP-UX Directory Server administration domain?
If so, If so, enter the host name and optionally the port of the directory server
that manages that topology (for example, acme.bus.com:389). Or to create
a new directory server administration domain, simply press Return.
(hostname[:port]|Return): Return

Creating new directory server instance in local host...

Creating directory server master instance "west-master". Please wait ...

Successfully created master instance with the following parameters:
        Instance name:  west-master
            Host name:  acct1053.west.acme.com
          Server port:  389
           Admin  URL:  http://acct1053.acme.com:9830
             SSL port:  636
          Domain name:  west.acme.com
        Domain suffix:  dc=west,dc=acme,dc=com
        Domain  Admin:  domadmin

* Generating self-signed CA certificate "WEST CA Certificate" ... completed.
* Generating server certificate "west-master Certificate" ... completed.
* Enabling SSL on directory server instance west-master ... completed.
* Restarted directory server instance west-master.
* Created directory server subtree.
* Added Domain and Host Administrator user/groups to the directory server.
* Created Domain Administrator account : "domadmin".
* Extended directory server schemas.
* Registered CA and server certificates in directory server.

================================================================================
NOTE: A CA certificate for the "west.acme.com" domain has been created.
This certificate can be pre-installed on HP-UX clients or included as part
of an HP-UX Ignite image.  Installing this CA certificate on host will
pre-establish trust with this directory server.  The depot file for this
CA certificate can be found at : /tmp/ca-west.acme.com.depot
```

```
================================================================================


Setting up the LDAP-UX client using the newly created directory server.
Loading CA certificate from directory server to local host ... done.
* Extending schemas ... done.
No LDAP-UX Configuration Profile was found.  Creating a new one.

* Downloading profile from DS ... done.
* Configuring ldapux_client.conf ... done.
* Provisioning LDAP-UX Client information into the Directory Server ... done.
* Setting up proxy user ... done.
* Configuring "/etc/nsswitch.conf" and "/etc/pam.conf" to use ldap ... done.
* Starting ldapclientd daemon ... done.
* Starting ldapcconfd ... done.

LDAP-UX was successfully configured.
```

2.3.6.3 Post-installation steps for New Directory Server Installation mode

After completing a New Directory Server mode guided installation, perform these steps:

1. The `autosetup` process created a new HP-UX account, the Domain Administrator (also known as `domadmin`). It also created three new groups: `DomainAdmins`, `HostAdmins`, and `UserAdmins`. Ensure that the user and group numbers (UIDs and GIDs) of the information you are importing or migrating does not collide with those numbers that were created by `autosetup`, as explained in Section 2.5.1.1 (page 90).
2. Consider registering the new directory server using an LDAP server record in the host's DNS domain (contact your DNS domain administrator). For more information, refer to RFC 2782.
3. When a new directory server instance is created, `autosetup` generates a CA and server SSL/TLS certificate for this instance. The generated CA certificate can be distributed to other HP-UX clients to pre-establish trust and confidentiality with the directory server just created. The CA certificate has been conveniently packaged in a Software Distributor depot file. The CA product found in this depot will install the CA certificate in the `/etc/opt/ldapux/cert8.db` file on any host where you install the CA product. As a means to pre-establish trust with the directory server, you can simplify distribution of this CA certificate by including the CA product in an Ignite-UX depot. You can view the contents of this depot file with the `swlist -s /tmp/ca-west.acme.com.depot` command.
4. Perform the post-installation configuration tasks documented in Section 2.5 (page 89), as needed.

2.3.7 Guided installation steps: Existing Directory Server Installation mode

This section explains how to install LDAP-UX for the first time on a host that already has a valid directory server. Section 2.3.7.1 (page 51) shows how to perform the guided installation interactively, explaining step-by-step how to respond to each prompt for user input. Section 2.3.7.2 (page 53) shows how to run a completely-automated (silent mode) guided installation.

NOTE: When configuring and setting up LDAP-UX, you will likely be prompted for credentials of an administrator. If you are asked to enter the credentials (password) of a user, make sure that the connection between your client and the HP-UX system (where you are running `autosetup`) is secured and not subject to network eavesdropping. One option to protect such communication may be to use the ssh protocol when connecting to the HP-UX host being configured.

2.3.7.1 Interactively running Existing Directory Server Installation mode

To interactively install LDAP-UX into an environment that already has a valid directory server, follow these steps. This example assumes that you have pre-installed a CA certificate, as described in step 2.

1. Log in as root and run the `autosetup` command, as shown in the following example:

   ```
   # /opt/ldapux/config/autosetup
   ```

2. The `autosetup` script searches for a registered LDAP-protocol directory server in the local DNS domain but does not find one, as indicated in the following example.

NOTE: The script searches for a registered server only if the directory server was not specified with the `-h` option command-line option or `LDAP_HOSTPORT` environment variable. If a registered directory server is found, `autosetup` uses that directory server automatically.

The script gives you the option of entering the host identification of an existing directory server (along with two other options). The installer specifies host name `hpdhcalif` (a directory server already exists, so a new directory server is not needed for serving LDAP-UX clients).

```
Scanning DNS domain "west.hp.com" for any registered LDAP directory servers...
    - No directory servers found.

Please enter the host name and port number of a directory server,
a Windows domain name, or press Return to create a new directory
server on this host [host: hpdhcalif Return
```

NOTE:

Unless you pre-install a CA or server certificate for the directory server, the `autosetup` tool has no means of validating the identity of the directory server. The tool can download and permanently install the CA or server certificate for the server; however, the server could be an impostor. If `autosetup` created the specified server, it created a depot file on that server's host that contains the CA certificate for that server. The depot on the specified host in this example is found at : `/tmp/ca-calif.acme.com.depot`. The depot file can be distributed to your host or any other HP-UX clients to be established in the same LDAP-UX domain. By installing it on your host prior to configuring LDAP-UX, you pre-establish trust with the specified remote server. For more information, see Section 2.3.2.3.3 (page 35).

If the specified server was not created by `autosetup`, you can obtain and pre-install the CA or server certificate directly from the server (in `/etc/opt/ldapux`) and pre-install it on your host, following the instructions in Section 2.4.6.2 (page 79).

If the CA certificate is not installed on your local host at this point of the guided installation, `autosetup` warns you that it cannot validate the identity of the remote server and suggests installing the CA certificate. You can abort so that you can install the CA certificate before proceeding with the rest of the guided installation, or you can continue, trusting the CA certificate that will be installed automatically by `autosetup`.

This example assumes the CA certificate has already been installed; therefore, you will not see the warning and the prompt asking whether to abort or continue.

3. The script then asks for the DN of the directory server user who can add the local host to the directory server's LDAP–UX domain. This is any host administrator with such privileges (a member of the DomainAdmins group). In this example, the DN for the user with such privileges is `uid=domadmin,ou=people,dc=calif,dc=acme,dc=com`. The server's DNS domain in this example is `calif.acme.com`; this will be the name of the LDAP-UX domain configured by `autosetup`. This being the first time adding an HP-UX host to this directory server, LDAP-UX will extend the server's schema.

```
Please enter the DN of a user that has sufficient privilege to add this host
to the "calif.acme.com" domain.  Note also that if this is the first
time adding an HP-UX host to this directory server, LDAP-UX may
also need to extend the server's schema.  Please enter the DN of an
Administrator with these privileges or press Return for the default value.
[uid=domadmin,ou=people,dc=calif,dc=acme,dc=com]: Return
```

4. Enter the password for the user identified in the preceding step (the entered password is not visible):

```
Enter the password for the above user: [password not displayed] Return
```

The installation now begins, followed by other related tasks; `autosetup` displays the progress and results, as in the following example. As indicated, because an existing LDAP-UX configuration profile does not exist, `autosetup` creates a new one. The profile and the associated LDAP-UX domain will be based on the existing directory tree. In addition, `autosetup` provisions information about the local host into the existing directory server.

```
* Extending schemas ... done.
No LDAP-UX Configuration Profile was found.  Creating a new one.

* Downloading profile from DS ... done.
* Configuring ldapux_client.conf ... done.
* Provisioning LDAP-UX Client information into the Directory Server ... done.
* Setting up proxy user ... done.
* Configuring "/etc/nsswitch.conf" and "/etc/pam.conf" to use ldap ... done.
* Starting ldapclientd daemon ... done.
* Starting ldapcconfd ... done.
```

```
LDAP-UX was successfully configured.
```

NOTE: For more information about the configuration files created or modified by `autosetup`, see "Samples of LDAP-UX configuration files created or modified by autosetup" (page 359). You can display details about the LDAP-UX Client Services configuration by using the `/opt/ldapux/config/display_profile_cache` command. For more information about the use of this command, see Section 7.2.4 (page 215).

2.3.7.2 Automating Existing Directory Server Installation mode

For this mode of installation, you can run `autosetup` in silent mode as well as provide pre-set values for parameters in the command line or with environment variables. As discussed in Section 2.3.7.1 (page 51), you must pre-establish trust with the remote directory server by installing the CA certificate prior to running `autosetup`. To perform this installation without user interaction, you need to specify the following in command-line options or environment variables:

The bind DN (the DN of the directory server user who can add the local host to the directory server's LDAP-UX domain): use either the `-D` option or the `LDAP_BINDDN` variable.
The password used with the bind DN: use either `-j` option or the `LDAP_BINDCRED` variable.
The host name of the directory server being joined: use either `-h` option or the `LDAP_HOSTPORT` variable.

In the following example, these parameters are specified in the command line:

```
# ./autosetup -h hpdhcalif -D "uid=domadmin,ou=people,dc=calif,dc=acme,dc=com" -j /tmp/jfile -q

* Extending schemas ... done.
No LDAP-UX Configuration Profile was found.  Creating a new one.

* Downloading profile from DS ... done.
* Configuring ldapux_client.conf ... done.
* Provisioning LDAP-UX Client information into the Directory Server ... done.
* Setting up proxy user ... done.
* Configuring "/etc/nsswitch.conf" and "/etc/pam.conf" to use ldap ... done.
* Starting ldapclientd daemon ... done.
* Starting ldapcconfd ... done.

LDAP-UX was successfully configured.
```

2.3.7.3 Post-installation steps for Existing Directory Server Installation mode

Perform the post-installation configuration tasks documented in Section 2.5 (page 89), as needed.

2.3.8 Guided installation steps: Existing LDAP-UX Domain Installation mode

This section explains how to install LDAP-UX in an environment that has already been configured for LDAP-UX, joining the local host into an existing LDAP-UX domain. In this mode, the guided installation simply downloads the existing domain configuration (the LDAP-UX configuration profile) and registers the host in the LDAP-UX domain. Section 2.3.8.1 (page 54) shows how to perform the guided installation interactively, explaining step-by-step how to respond to each prompt for user input. Section 2.3.8.2 (page 56) shows how to run a completely-automated (silent mode) guided installation.

NOTE: This section assumes you are installing LDAP-UX on a host on which LDAP-UX is not already installed. If you attempt to run `autosetup` on a host on which LDAP-UX (`ldapclientd`) is already running, the procedure aborts. If the LDAP-UX is installed on the host but not running, the procedure proceeds. However, if a previous LDAP-UX configuration profile is found on the system, the procedure warns you that proceeding will overwrite the file and asks if you want to proceed.

You can proceed if your intention is to reconfigure LDAP-UX on the host. You could reconfigure LDAP-UX for any of several reasons, such as:

- You want the host to connect to a different directory server than the one the host was originally configured to connect to.
- The LDAP-UX configuration was corrupted (an error indicates a component is corrupted).
- A directory server user inadvertently deleted the host entry from the directory server. This removes the proxy user required to connect to the directory server; to correct this, re-run `autosetup` to recreate the host entry and re-establish user proxies.

NOTE: When configuring and setting up LDAP-UX, you will likely be prompted for credentials of an administrator. If you are asked to enter the credentials (password) of a user, make sure that the connection between your client and the HP-UX system (where you are running `autosetup`) is secured and not subject to network eavesdropping. One option to protect such communication may be to use the ssh protocol when connecting to the HP-UX host being configured.

2.3.8.1 Interactively running Existing LDAP-UX Domain Installation mode

To interactively install LDAP-UX onto a host that is to join an existing LDAP-UX environment, follow these steps. This example assumes that you pre-install a CA certificate, as described in step 2.

1. Log in as root and run the `autosetup` command, as shown in the following example:

   ```
   # /opt/ldapux/config/autosetup
   ```

2. Install the domain CA certificate product from the depot created when the original directory server instance was created. Securely copy the `/tmp/ca-mydomain.example.com.depot` file to your local host and install the Domain CA product using the following command:

   ```
   swinstall -s /tmp/ca-mydomain.example.com.depot \*
   ```

   If you skip this step, `autosetup` will prompt you whether to trust the directory server.

3. The `autosetup` script searches for a registered directory server in the local DNS domain but does not find one, as indicated in the following example.

NOTE: The script searches for a registered server only if the directory server was not specified with the `-h` option command-line option or `LDAP_HOSTPORT` environment variable. If a registered directory server is found, `autosetup` uses that directory server automatically.

The script gives you the option of entering the host identification of the existing directory server (along with two other options). In this example, the existing directory server is the one created in Section 2.3.6 (page 44). The installer specifies it's host name `acct1053` (a directory server already exists, so a new directory server instance will not be created).

```
Scanning DNS domain "west.hp.com" for any registered LDAP directory servers...
    - No directory servers found.

Please enter the host name and port number of a directory server,
```

```
a Windows domain name, or press Return to create a new directory
server on this host: acct1053 Return
```

NOTE: Unless you pre-install a CA or server certificate for the directory server, the `autosetup` tool has no means of validating the identity of the remote directory server (`acct1053`). The tool can download and permanently install the CA or server certificate for the server; however, the server might be an impostor.

If the specified server was not created by the guided installation, you can obtain the CA or server certificate directly from the server (in `/etc/opt/ldapux`) and pre-install it on your host. For more information, see Section 2.4.6.2 (page 79).

If the CA certificate is not installed on your local host at this point of the guided installation, `autosetup` warns you that it cannot validate the identity of the remote server and suggests installing the CA certificate. You can abort so that you can install the CA certificate before proceeding with the rest of the guided installation, or you can continue, trusting the CA certificate that will be installed automatically by `autosetup`.

This example assumes the CA certificate has already been installed; therefore, you will not see the warning and the prompt asking whether to abort or continue.

4. The script then asks for the DN of the directory server user who can add the local host to the directory server's LDAP-UX domain. This is any host administrator with such privileges (a member of the DomainAdmins group). In this example, the DN for the user with such privileges is `uid=domadmin,ou=people,dc=calif,dc=acme,dc=com`. The server's DNS domain in this example is `calif.acme.com`; this will be the name of the LDAP-UX domain configured by `autosetup`. Because the LDAP-UX domain has already been set up on the directory server, LDAP-UX should not need to extend the server's schema. Instead, the credentials entered at this prompt merely need the privilege to add information about the current HP-UX host to the directory server.

   ```
   Please enter the DN of a user that has sufficient privilege to add this host
   to the "calif.acme.com" domain.  Note also that if this is the first
   time adding an HP-UX host to this directory server, LDAP-UX may
   also need to extend the server's schema.  Please enter the DN of an
   Administrator with these privileges or press Return for the default value.
   [uid=domadmin,ou=people,dc=calif,dc=acme,dc=com]: Return
   ```

5. Enter the password for the user identified in the preceding step (the entered password is not visible):

   ```
   Enter the password for the above user: [password not displayed] Return
   ```

The installation now begins, followed by other related tasks; `autosetup` displays the progress and results, as in the following example. Because an existing LDAP-UX configuration profile does exist, `autosetup` downloads the existing profile from the directory server instead of creating a new one. The profile and the associated LDAP-UX domain will be based on the existing directory tree. In addition, `autosetup` provisions information about the local host into the existing directory server.

```
* Extending schemas ... done.
* Downloading profile from DS ... done.
* Configuring ldapux_client.conf ... done.
* Provisioning LDAP-UX Client information into the Directory Server ... done.
* Setting up proxy user ... done.
* Configuring "/etc/nsswitch.conf" and "/etc/pam.conf" to use ldap ... done.
* Starting ldapclientd daemon ... done.
* Starting ldapcconfd ... done.

LDAP-UX was successfully configured.
```

NOTE: For more information about the configuration files created or modified by `autosetup`, see "Samples of LDAP-UX configuration files created or modified by autosetup" (page 359).

You can display details about the LDAP-UX Client Services configuration by using the `/opt/ldapux/config/display_profile_cache` command. For more information about the use of this command, see Section 7.2.4 (page 215).

2.3.8.2 Automating Existing LDAP-UX Domain Installation mode

For this mode of installation, you can run `autosetup` in silent mode as well as provide pre-set values for parameters in the command line or with environment variables. You must pre-establish trust with the remote directory server by installing the CA certificate prior to running `autosetup` (for more information, see Section 2.3.2.3.3 (page 35)). To perform this installation without user interaction, you need to specify the same command-line options or environment variables as required by the automated Existing Directory Server Installation:

The bind DN (the DN of the directory server user who can add the local host to the directory server's LDAP-UX domain): use either the `-D` option or the `LDAP_BINDDN` variable.
The password used with the bind DN: use either `-j` option or the `LDAP_BINDCRED` variable.
The host name of the directory server being joined: use either `-h` option or the `LDAP_HOSTPORT` variable.

In the following example, these parameters are specified in the command line:

```
# ./autosetup -h acct1053 -D "uid=domadmin,ou=people,dc=calif,dc=acme,dc=com" -j /tmp/jfile -q
* Extending schemas ... done.
* Downloading profile from DS ... done.
* Configuring ldapux_client.conf ... done.
* Provisioning LDAP-UX Client information into the Directory Server ... done.
* Setting up proxy user ... done.
* Configuring "/etc/nsswitch.conf" and "/etc/pam.conf" to use ldap ... done.
* Starting ldapclientd daemon ... done.
* Starting ldapcconfd ... done.
LDAP-UX was successfully configured.
```

2.3.8.3 Post-installation steps for Existing LDAP-UX Domain Installation mode

After completing an Existing LDAP-UX Domain mode guided installation, perform these steps:

- If you installed LDAP-UX into an existing LDAP-UX B.04.*xx* environment, or into an LDAP-UX B.05.*xx* environment that was configured by the customized installation (`setup`), or in any LDAP-UX B.05.*xx* environment that was modified since being created, ensure that the user/group and host management tools can identify the proper locations in the directory server tree for creating new users, groups, and hosts. The tools expect the LDAP-UX configuration profile to indicate the correct location for the host entries. For more information about assuring the management tools properly interface with the configuration profile, see Section 7.3.5.6 (page 242) and Section 5.6.1 (page 174).
- Perform the post-installation configuration tasks documented in Section 2.5 (page 89), as needed.

2.4 Customized installation (setup)

The customized installation requires that you be familiar with the LDAP-UX and directory server environment. This section describes how to perform this installation, tailoring the installation and configuration to the specific needs of your organization and environment.

2.4.1 Summary of customized installation and configuration steps

The following are the steps you take when custom installing and configuring an LDAP-UX Client Services environment:

- Plan your installation (see Section 2.4.2 (page 59)).
- Install LDAP-UX Client Services on each client system (see Section 2.4.3 (page 64)).
- Install and configure an LDAP directory, if not already done (see Section 2.4.4 (page 65)).
- If you want to enable SSL support with LDAP-UX, install and set up the security database files on the LDAP-UX client system (see Section 2.4.6 (page 79)).
- Migrate your name service data to the directory (see Section 2.5.1 (page 90)).
- Run the setup program to configure LDAP-UX Client Services on a client system (see Section 2.4.5 (page 68)). The `setup` program does the following for you:
  - Extends your RHDS/HPDS directory schema with the configuration profile schema, if not already done.
  - Imports the LP printer schema into your LDAP-based directory server if you choose to start the LDAP printer configurator.
  - Imports the NIS publickey schema into your LDAP-based directory if you choose to store the NIS-style public keys of users and hosts in the LDAP directory.
  - Imports the automount schema into your LDAP-based directory server if you choose to store the AutoFS maps in the LDAP directory.
  - Creates a start-up file on the client. This enables each client to download the configuration profile.
  - Creates a centrally-managed configuration profile in the LDAP directory server. This profile defines how HP-UX clients should access the directory server and defines the data model (schema) used to identify users, groups, and other OS services. This profile can be shared across numerous clients and defines what is known as the "LDAP-UX domain". The `setup` program can download an existing configuration profile, create a new one, or define a local-only profile.
  - Downloads the configuration profile from the directory to the client.
  - Starts the product daemon `ldapclientd`, if you choose to start it. Starting with LDAP-UX Client B.03.20 or later, the client daemon must be started to obtain LDAP-UX functionality. With LDAP-UX Client B.03.10 or earlier, running the client daemon is optional.
- To specify LDAP authentication and name service, modify the files `/etc/pam.conf` and `/etc/nsswitch.conf`, respectively, on the client (see Section 2.4.5 (page 68)).
- Optionally, configure the PAM Authorization Service Module (PAM_AUTHZ) to control access rules defined in the `/etc/opt/ldapux/pam_authz.policy` policy file. In addition, verify the user access rights of a subset of users in a large repository needing access, modifying the `/etc/opt/ldapux/pam_authz.policy` and `/etc/pam.conf` files. For command syntax, see the *pam_authz*(5) manpage; for more information about configuring this service, see "PAM_AUTHZ login authorization " (page 140).
- Perform the relevant post-installation tasks described in Section 2.5 (page 89). These include:
  - Importing name service data into your directory (see Section 2.5.1 (page 90))
  - Verifying each client is working properly (see Section 2.5.2 (page 92))
  - Enabling AutoFS support (see "Enabling AutoFS support" (page 95))
  - Enabling offline credential caching for authentication when the directory server is not available (see "Enabling offline credential caching for authentication when the directory server is unavailable" (page 102))
  - Enabling integrated Compat Mode to control name services and user logins (see Section 2.5.5 (page 104))

- Control user access to the system, using any of several methods mentioned in "Controlling user access to the system through LDAP" (page 106)
- Configure subsequent client systems (see the shortcuts mentioned in "Configuring subsequent client systems" (page 112))
- Downloading the profile periodically (see "Downloading the profile periodically" (page 113))
- Enabling the use of `-r` commands with PAM_LDAP (see "Using the r-command for PAM_LDAP" (page 113))

2.4.2 Planning for your customized installation and configuration

Before beginning your installation, you should plan how you will set up and verify your LDAP directory and your LDAP-UX Client Services environment before putting them into production. Consider the following questions. Record your decisions and other information that you will need later in "Configuration worksheet" (page 347).

- How many LDAP-based directory servers and replicas will you need?

  Each client system binds to an LDAP directory server containing your user, group, and other data. Multiple clients can bind to a single directory server or replica server. The answer depends on your environment, the size and configuration of your directory, and how many users and clients you have. Write your directory server host and TCP port number in "Configuration worksheet" (page 347). For more information, see the white paper *Preparing Your LDAP Directory for HP-UX Integration* at:

  http://www.hp.com/go/hpux-security-docs

  Click **HP-UX LDAP-UX Integration Software**.

  In addition, for more information about preparing an HP-UX Directory Server or Red Hat Directory Server, see the appropriate *Deployment Guide* at the website mentioned previously. You can add directory replicas to an existing LDAP-UX Client Services environment as described under Section 5.4 (page 158). You may also want to review the *LDAP-UX Integration Performance and Tuning Guidelines*, also located at the website mentioned previously.

- Where will you get your name service data when migrating it to the directory?

  You can get the data from your files in the `/etc` directory or, if you are using NIS, from the same source files from which you create your NIS maps, or you can get the data from your NIS maps themselves. Write this information in "Configuration worksheet" (page 347).

  For information about how to import your information into the directory, see Section 2.5.1 (page 90). For information about the migration scripts, see Section 7.6 (page 326).

  To add an individual user entry or modify an existing user entry in your directory, you can use the `ldapugadd` or `ldapugmod` command or other directory administration tools such as the `ldapmodify` command or the HP-UX Directory Server Console. For additional contributed tools, see the *LDAP-UX Integration B.05.00 Release Notes*.

NOTE: You should keep a small subset of users in `/etc/passwd`, particularly the root login. This allows administrative users to log in during installation and testing. Also, if the directory is unavailable, you can still log in to the system.

- Where in your directory will you put your name service data?

  Your directory architect needs to decide where in your directory to place your name service information. By default, LDAP-UX Client Services expects user and group data to use the object classes and attributes specified by RFC 2307. By default, the migration scripts create and populate a new subtree that conforms to RFC 2307. Figure 2-3 (page 61) shows a base DN of `ou=unix,o=hp.com`. Write the base DN of your name service data in "Configuration worksheet" (page 347).

  If you prefer to merge your name service data into an existing directory structure, you can map the standard RFC 2307 attributes to alternate attributes. For more information, see "LDAP-UX Client Services object classes" (page 349).

- How will you put your user, group, and other data into your directory?

  LDAP supports group membership defined in the X.500 syntax (using the `member` or `uniquemember` attribute), while still supporting the RFC 2307 syntax (using the `memberuid` attribute). This new group membership syntax increases LDAP-UX integration with LDAP and other LDAP-based applications, and may reduce administration overhead eliminating the need to manage the `memberuid` attribute. In addition, a new performance improvement

has been made through the addition of a new caching daemon that caches passwd, group, and X.500 group membership information retrieved from an LDAP server. This significantly reduces LDAP-UX's response time to applications. To improve performance further, the daemon re-uses connections for LDAP queries and maintains multiple connections to an LDAP server.

The migration scripts provided with LDAP-UX Client Services can build and populate a new directory subtree for your user and group data.

If you merge your data into an existing directory, such as to share user names and passwords with other applications, the migration scripts can create LDIF files of your user data, but you will have to write your own scripts or use other tools to merge the data into your directory. You can add the posixAccount object class to your users already in the directory to leverage your existing directory data.

For information about how to import your information into the directory, see Section 2.5.1 (page 90). For information about the migration scripts, see Section 7.6 (page 326) .

CAUTION: If you place a root login (any account with UID number 0) in the LDAP directory, that user and password will be able to log in as root to any client using LDAP-UX Client Services. Keeping the root user in `/etc/passwd` on each client system allows the root user to be managed locally. This can be especially useful when the network is down, because it allows local access to the system when access to the directory server is unavailable.

It is not recommended that you put the same users both in `/etc/passwd` and in the directory. This could lead to conflicts and unexpected behavior.

Note that LDAP-UX Client Services (version B.05.00 or later) offers offline, long-term credential caching that enables LDAP-UX to authenticate users attempting to log in to the system when credential information is unavailable from the directory server (when the server or network is down, for example). For information about this feature and how to configure it, see Section 2.5.4 (page 102).

NOTE: If you are planning a first-time deployment of managing user and group data in the directory server, HP suggests that you devise a strategy to avoid UID number and GID number overlap. Most likely, you will need to continue managing some accounts local to the hosts. Often the root user, and sometimes application accounts (such as `www` for the `httpd` process) remain managed in the local `/etc/passwd` file. Devise a convention establishing a range for UID numbers and one for GID numbers such that accounts and groups in LDAP do not conflict with those on local hosts. For example, accounts in LDAP could all have UID numbers greater than 1000, while accounts on local hosts would be restricted to UID numbers less than 1000.

For information about ensuring that user and group numbers to be migrated or imported into a new directory server do not collide with the ones created by the guided installation, see Section 2.5.1.1 (page 90).

- How many profiles do you need?

  A configuration profile is a directory entry that contains configuration information shared by a group of clients. The profile contains the information clients need to access user and group data in the directory, for example:
  - Your directory server hosts
  - Where user, group, and other information is in the directory
  - The method clients use to bind to the directory
  - Other configuration parameters such as search time limits

  If these parameters are the same for all your clients, you need only one profile. You need at least one profile per directory server or replica. In general, to simplify maintenance, it is a

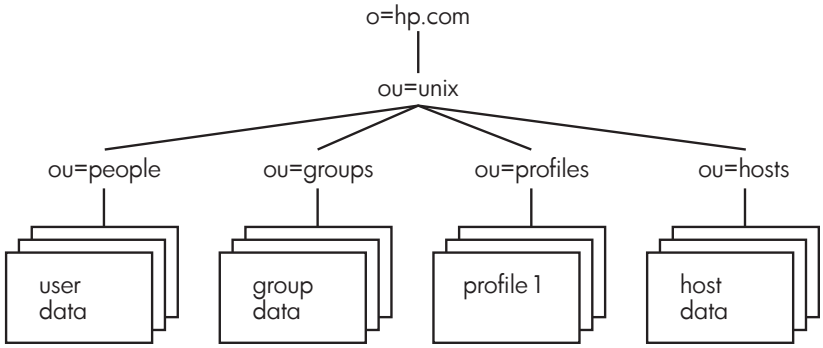
good idea to have as few profiles as necessary. To see what is in a profile and help you decide how many different profiles you need, look at the posixNamingProfile object class in "LDAP-UX Client Services object classes" (page 349).

If you are familiar with NIS, one possibility is to create a separate profile for each NIS domain.

- Where in your directory will you put your profile?

  The profile contains directory access information. It specifies how and where clients can find user and group data in the directory. You can put the profile anywhere you want, as long as the client systems can read it. For example, you might put it near your user data, or you could put it in a separate administrative area. To simplify access permission, put the profile in the same directory as your user and group data. Clients must have access to both the profile and the user and group data. Figure 2-3 shows a configuration profile DN of `cn=profile1,ou=profiles,ou=unix,o=hp.com`.

Figure 2-3 Example directory structure



  Write your configuration profile DN on the worksheet in "Configuration worksheet" (page 347).

- By what method will client systems bind to the directory?

  Clients can bind to the directory anonymously. This is the default and is simplest to administer. If you need to prevent access to your data from anonymous users, or your directory does not support anonymous access, you can use a proxy user. If you configure a proxy user, you can also configure anonymous access to be attempted in the event the proxy user fails.

  Write your client access method and proxy user DN, if needed, on the worksheet in "Configuration worksheet" (page 347).

- How will you increase the security level of the product to prevent an unwanted user from logging in to the system through LDAP? What is the procedure to set up increased login security?

  The default is to allow all users stored in the LDAP directory to log in. To disallow specific users to log in to a local system, you can configure the `disable_uid_range` flag in `/etc/opt/ldapux/ldapux_client.conf` file, as described in Section 2.5.6.1 (page 106).

  You can also use `pam_authz` or the `deny_local` option (in PAM_LDAP) to disable system access for accounts defined in LDAP. For more information, about the PAM_AUTHZ service module, see Section 5.3 (page 140) or the *pam_authz*(5) manpage. For information about the `deny_local` option, see Section 2.5.6.2 (page 107).

- What PAM authentication will you use? How will you set up the PAM configuration file `/etc/pam.conf`? What other authentication do you want to use and in what order? Do

you wish to use the Pam Authorization Service module (PAM_AUTHZ) for user access control?

PAM provides authentication services. You can configure PAM to use LDAP, Kerberos, or other traditional UNIX locations (for example files, NIS, NIS+) as controlled by NSS. For more information about PAM, see the *pam*(3) and *pam.conf*(4) manpages, and the *Managing Systems and Workgroups: A Guide for HP-UX System Administrators* document at the following location:

www.hp.com/go/hpux-core-docs (click **HP-UX 11i v2**)

HP recommends that you use HP-UX file-based authentication first, followed by LDAP or other authentication. The `/etc/pam.ldap` file is an example of this type of configuration. With this configuration, PAM uses traditional authentication first, searching `/etc/passwd` when any user logs in, then attempts to authenticate to the directory if the user is not in `/etc/passwd`. If you have a few users in **/etc/passwd**, in particular the root user, and if the directory is unavailable, you can still log in to the client as a user in `/etc/passwd`.

- Do you want to use TLS (Transport Layer Security) or SSL for secure communication between clients and the directory server?

  LDAP-UX supports SSL or TLS with password as the credential, using either simple bind or DIGEST-MD5 authentication to ensure confidentiality and data integrity between clients and servers. startTLS is a new extension operation of TLS protocol. You can use the StartTLS operation to set the TLS secure connection over a regular (un-encrypted) LDAP port. The secure connection can also be established on an encrypted LDAP port when using SSL. By default, SSL and TLS are disabled. For detailed information, see Section 2.4.6 (page 79).

- What authentication method will you use when you choose to enable TLS?

  You have a choice between SIMPLE (the default), or SASL/GSSAPI, or SASL/DIGEST-MD5. SASL/GSSAPI is only supported for LDAP-UX used with Windows ADS.

- What authentication method will you use if you choose to enable SSL?

  You have a choice between SIMPLE (the default), or SASL/GSSAPI, or SASL/DIGEST-MD5. SASL/GSSAPI is only supported for LDAP-UX used with Windows ADS.

- What authentication method will you use if you choose not to enable SSL and TLS?

  You have a choice between SIMPLE (the default), or SASL/GSSAPI, or SASL/DIGEST-MD5. SASL/ DIGEST-MD5 improves security, preventing snooping over the network during authentication. SASL/GSSAPI is only supported for LDAP-UX used with Windows ADS.

  Using the DIGEST-MD5 authentication may require that the password be stored in clear text in the LDAP directory server.

- Do you want to import the LDAP printer schema (if you choose to start the printer configurator)?

  LDAP-UX Client Services B.03.20 or later provides the integration with the LDAP printer configurator to simplify the LP printer management by updating LP printer configuration automatically on your HP-UX system. A new printer schema, which is based on RFC 3712, is required to start the services.

IMPORTANT: If you attempt to use this new feature, in the `ldapclientd.conf` file, the `start` configuration parameter of the printer services section must be set to `yes`. If the `start` option is enabled, the printer configurator will start when `ldapclientd` is initialized. By default, the `start` parameter is enabled.

- Do you want to import the NIS publickey schema into your LDAP directory if you choose to store and manage NIS publickeys in the LDAP directory.

  LDAP-UX Client Services supports discovery and management of NIS publickeys in an LDAP directory. Both public and private (secret) keys, used by the SecureRPC API can be stored in user and host entries in an LDAP directory server, using the `nisKeyObject` objectclass.

- Do you want to import the automount schema into your LDAP directory server if you choose to store and manage automount maps in the LDAP directory?

  LDAP-UX Client Services supports the automount service under the AutoFS subsystem. This new feature allows you to store or retrieve automount maps in/from an LDAP directory. LDAP-UX Client Services supports the new automount schema based on RFC 2307-bis. The `nisObject` automount schema can also be used if configured through attribute mappings.

  For the detailed information about AutoFS with LDAP support, see Section 2.5.3 (page 95).

- What name services will you use? How will you set up `/etc/nsswitch.conf`? In what order do you want NSS to try services?

  NSS is the Name Service Switch, providing naming services for user names, group names, and other information. You can configure NSS to use files, LDAP, or NIS in any order and with different parameters. For an example `nsswitch.conf` file using files and LDAP, see `/etc/nsswitch.ldap`. For information on NSS, see the *switch*(4) manpage and the "Configuring the Name Service Switch" chapter in *NFS Services Administrator's Guide*, available at the following location:

  http://www.hp.com/go/hpux-core-docs (Click **HP-UX 11i v3**).

  HP recommends that you use files first, followed by LDAP for `passwd`, `group`, and other supported name services. With this configuration, NSS will first check files, and if the name service data is not in the respective files, then check the directory. The `/etc/nsswitch.ldap` file is an example of this configuration.

- Do you need to configure login authorization for a subset of users from a large repository such as an LDAP directory? How will you set up the `/etc/opt/ldapux/pam_authz.policy` and `/etc/pam.conf` files to implement this feature?

  The PAM_AUTHZ service module for PAM provides functionality that allows the administrator to control who can log in to the system. These modules are located at `/usr/lib/security/libpam_authz.1` on a PA-RISC machine and at `libpam_authz.so.1` on the HP Integrity (IA64) server. The PAM_AUTHZ module has been created to provide access control similar to the netgroup filtering feature that is performed by NIS. These modules are located at `/usr/lib/security/libpam_authz.1` on a PA-RISC machine (`libpam_authz.so.1` on the Integrity server machine. Starting with LDAP-UX Client Services B.04.00, PAM_AUTHZ has been enhanced to allow system administrators to configure and customize their local access rules in a local policy file, `/etc/opt/ldapux/pam_authz.policy`. The PAM_AUTHZ module uses these access control rules defined in the local policy file to control the login authorization. PAM_AUTHZ is intended to be used when NIS is not used, such as when the PAM_LDAP or PAM_KERBEROS authentication modules are used. Because PAM_AUTHZ doesn't provide authentication, it doesn't verify if a user account exists.

  If the `/etc/opt/ldapux/pam_authz.policy` file does not exist in the system, PAM_AUTHZ provides access control based on the netgroup information found in the

`/etc/passwd` and `/etc/netgroup` files. If the `/etc/opt/ldapux/pam_authz.policy` file exists in the system, PAM_AUTHZ uses the access rules defined in the policy file to determine who can log in to the system.

For detailed information on this feature and how to configure the `/etc/opt/ldapux/pam_authz.policy` file, see Section 5.3 (page 140) or the *pam_authz*(5) manpage.

- Do you want to configure the `/etc/opt/ldaux/pam_authz.policy` to enforce account and password policies, stored in an LDAP directory server?

  LDAP-UX provides PAM_AUTHZ enhancement to support enforcement of account and password policies, stored in an LDAP directory server. This feature works in conjunction with secure shell (ssh), r-commands (`rlogin`, `rcp`, and so forth) with `rhost` enabled where authentication is not performed by the PAM subsystem, but is performed by the command itself.

  For detailed information on this feature and how to configure the `pam_authz.policy` file, see Section 5.3.10 (page 153).

- How will you communicate with your user community about the change to LDAP?

  For the most part, your user community should be unaffected by the directory. Most HP-UX commands will work as always.

  Check the *Release Notes* for any other limitations and tell your users how they can work around them.

2.4.3 Installing LDAP-UX Client Services on a client

Use `swinstall` to install the LDAP-UX Client Services software, the NativeLdapClient subproduct, on a client system. For more information about the command, see the *swinstall*(1M) manpage. In addition, see the *LDAP-UX Integration B.05.00 Release Notes* for any last-minute changes to this procedure. You do not need to reboot your system after installing the product.

NOTE: Starting with LDAP-UX Client Services B.03.20 or later, system reboot is not required after installing the product.

NOTE: For the HP 9000 and HP Integrity server client systems, you might need to install required patches. For the detailed information about the required patches, see *LDAP-UX Integration B.05.00 Release Notes* at:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX LDAP-UX Integration Software**.

2.4.4 Configuring your directory

This section describes how to configure your directory to work with LDAP-UX Client Services. Examples are given for the HP-UX Directory Server. For information about supported directories, see the *LDAP-UX Integration Release Notes* . If you have a different directory, see the documentation for your directory for details on how to configure it.

For more information, see *Preparing Your LDAP Directory for HP-UX Integration* at:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX LDAP-UX Integration Software**.

1. Install the posix schema (RFC 2307) into your directory.

   With most directory servers, the posix schema is already installed. However, if you need to install this schema, you may use the `/opt/ldapux/bin/ldapschema` tool to install the `/etc/opt/ldapux/schema/rfc2307.xml` schema file.

   For information on the posix schema (RFC 2307), see the following website:

   http://www.ietf.org/rfc.html

   RFC 2307 consists of object classes such as: posixAccount, posixGroup, shadowAccount (deprecated), etc. posixAccount represents a user entry from `/etc/passwd`. posixGroup represents a group entry from `/etc/group`.

2. Restrict write access to certain passwd (posixAccount) attributes of the posix schema.

   CAUTION: Make sure you restrict access to the attributes listed below. Allowing users to change them could be a security risk

   Grant write access of the `uidnumber`, `gidnumber`, `homedirectory`, and `uid` attributes only to directory administrators; disallow write access by all other users. You may want to restrict write access to other attributes in the `passwd` (posixAccount) entry as well.

   With HP-UX Directory Server, you can use the Directory Server Console or `ldapmodify` to set up access control instructions (ACI) so ordinary users cannot change these attributes in their passwd entry in the directory.

   The following access control instruction is by default at the top of the directory tree for an HP-UX Directory Server (version 8.1). This ACI allows a user to change any attribute in their passwd entry:

   ```
   aci: (targetattr = "*") (version 3.0; acl "Allow self entry modification";
    allow (write)userdn = "ldap:///self";)
   ```

   You could modify this example ACI to the following, which prevents ordinary users from changing their `uidnumber`, `gidnumber`, `homedirectory`, and `uid` attributes:

   ```
   aci: (targetattr != "uidnumber || gidnumber || homedirectory || uid") (version
    3.0; acl "Allow self entry modification, except for important posix attributes";
    allow (write)userdn = "ldap:///self";)
   ```

   You may have other attributes you need to protect as well.

   To change an ACI with the Directory Server Console, select the Directory tab, select your directory suffix in the left-hand panel, then select the **Object→Set Access Permissions** menu item. In the dialog box, select the "Allow self entry modification" ACI and click OK. Use the Set Access Permissions dialog box to modify the ACI. For details, see the *HP-UX Directory Server administrator guide*.

3. Restrict write access to certain group (posixGroup) attributes of the posix schema.

   Grant write access of the cn, memberuid, gidnumber, and userPassword attributes only to directory administrators; disallow write access by all other users.

   With the HP-UX Directory Server, you can use the Directory Server Console or `ldapmodify` to set up access control lists (ACL) so ordinary users cannot change these attributes in the posixGroup entry in the directory. For example, the following ACI, placed in the directory

at ou=groups,ou=unix,o=hp.com, allows only the directory administrator to modify entries below ou=groups,ou=unix,o=hp.com:

```
aci: (targetattr = "*")(version 3.0;acl "Disallow modification of group
 entries"; deny (write) (groupdn != "ldap:///ou=Directory Administrators,
 o=hp.com");)
```

4. Grant read access of all attributes of the posix schema.

   Ensure all users have read access to the posix attributes.

   When using PAM_LDAP as your authentication method, users do not need read access to the userPassword attribute since the authentication is handled by the directory itself. Therefore, for better security, you can remove read access to userPassword from ordinary users.

5. Configure anonymous access, if needed. If you do not configure a proxy user, then the attributes of your name service data must be readable anonymously.
6. Create a proxy user in the directory, if needed.

   To create a proxy user with the HP-UX Directory Server, go to the the directory server's main Console, select the Users and Groups tab, and then click on the **Create** button. For example, you might create a user uid=proxyuser,ou=Special Users,o=hp.com.

7. Set access permissions for the proxy user, if configured.

   Give the proxy user created above read permission for the posix account attributes.

   With HP-UX Directory Server, for example, the following ACI gives a proxy user permission to compare, read, and search all posix account attributes except the userPassword attribute:

```
aci: (target="ldap:///o=hp.com")(targetattr!="userpassword")
  version 3.0; acl "Proxy userpassword read rights";
  allow (compare,read,search)
  userdn = "ldap:///uid=proxyuser,ou=Special Users,o=hp.com";)
```

8. The default ACI of Netscape Directory Server 6.11 allows a user to change his own common attributes. But, for Netscape Directory Server 6.21 or later, you need to set ACI that gives a user permission to change his own common attributes. By default, the Netscape Directory Server 6.21 or later provides the following ACI named Enable self write for common attributes that gives a user permission to change his own common attributes:

```
aci: (targetattr = "carLicense ||description ||displayName
 ||facsimileTelephoneNumber ||homePhone ||homePostalAddress ||initials
 ||jpegPhoto ||labeledURL ||mail ||mobile ||pager ||photo ||postOfficeBox
 ||postalAddress ||postalCode ||preferredDeliveryMethod ||preferredLanguage
 ||registeredAddress ||roomNumber ||secretary ||seeAlso ||st ||street
 ||telephoneNumber ||telexNumber ||title ||userCertificate ||userPassword
 ||userSMIMECertificate ||x500UniqueIdentifier")
 (version 3.0; acl "Enable self  write for common attributes"; allow (write)
 (userdn = "ldap:///self"))
```

   You can modify the default ACI and give appropriate access rights to change your own common attributes.

9. Index important attributes for better performance of Directory Server.

   Since many of your directory requests will be for the attributes listed below, you should index these to improve performance. If you don't index, your directory may search sequentially causing a performance bottleneck. As a rule of thumb, databases containing more than 100 entries should be indexed by their key attributes.

   The following attributes are recommended for indexing:

   - cn
   - objectclass
   - memberuid
   - uidnumber

- `gidnumber`
- `uid`
- `ipserviceport`
- `iphostnumber`

To index these entries with HP-UX Directory Server, go to the Directory Server Console's Configuration tab, then the Indexes tab, and click on the **Add Attributes** button.

10. Determine if you need to support enumeration requests. If you do, increase the Look-Through limit, and the Size limitin the Directory Server.

    Enumeration requests are directory queries that request all of a database, for example all users or all groups. Enumeration requests of large directory databases could reduce network and server performance. With large HPDS/RHDS directories and default configurations, enumerations may fail or provide incomplete data, but the default configuration also may prevent performance problems from enumerations.

    If you need to support enumerations with large directory databases, increase the listed parameters as described in *Preparing Your LDAP Directory for LDAP-UX Integration* available at:

    http://www.hp.com/go/hpux-security-docs

    Click **HP-UX LDAP-UX Integration Software**.

    In HP-UX Directory Server, the Look-through limit specifies the maximum number of directory entries to examine before aborting the search operation. The Size limit determines the maximum number of entries to return to any query before aborting.

    For information on these parameters and how to change them, see the *HP-UX Directory Server administrator guide*. See also Section 5.16.1 (page 185).

11. If you want to enable SSL support with LDAP-UX, you need to turn on SSL in your directory server. For detailed information on how to set up and configure your Directory Server to enable SSL communication over LDAP, see the *HP-UX Directory Server administrator guide* at:

    http://www.hp.com/go/hpux-security-docs

    Click **HP-UX Directory Server**.

2.4.5 Configuring the LDAP-UX Client Services

Below is a summary of how to configure LDAP-UX Client Services with HP-UX Directory Server. For a default configuration, see Section 2.4.5.1 (page 69). For a custom configuration, see Section 2.4.5.2 (page 73) for more information.

NOTE: The `setup` program has only been certified with HP-UX Directory Server version 8.1, Red Hat Directory Server 8.0, Windows Server 2003 R2 Active Directory Server, and Windows 2008 Active Directory Server. For more information, see the *LDAP-UX Integration B.05.00 Release Notes*.

NOTE: The LDAP-UX Client Services supports storage of automount maps and NIS publickeys on Red Hat Directory Server 8.0 and HP-UX Directory Server 8.1. For more information, see the *LDAP-UX Integration B.05.00 Release Notes* .

- Run the `setup` program. The `setup` program provides the following assistance:
  - Extends your directory server schema with the configuration profile schema, if not already done

    NOTE: To use a local-only profile, run the `setup` program using the `-l` option . Use the local-only profile for small deployments, testing purposes, and for environments where administrators lack server administrative privileges.

  - Imports the LDAP printer schema into your Directory Server if you choose to start the LDAP printer configurator
  - Imports the NIS publickey schema into your Directory Server if you choose to store the public keys of users and hosts in an LDAP directory
  - Imports the new automount schema into your Directory Server if you choose to store the AutoFS maps in an LDAP directory
  - Provides the option to enable SSL for secure communication between LDAP clients and Directory servers
  - Optionally configures SASL DIGEST-MD5 authentication
  - Creates a configuration profile entry in your directory server from information you provide
  - Updates the local client's start-up file (/etc/opt/ldapux/ldapux_client.conf) with your directory and configuration profile location
  - Downloads the configuration profile from the directory to your local client system
  - Configures a proxy user for the client, if needed
  - Starts the client daemon if you choose to start it

IMPORTANT: Starting with LDAP-UX Client Services B.03.20, the client daemon, `/opt/ldapux/bin/ldapclientd`, must be running for LDAP-UX functions to work. With LDAP-UX Client Services B.03.10 or earlier, running the client daemon, `ldapclientd`, is optional.

NOTE: The LDAP printer configurator can support any Directory Servers that support the LDAP printer schema based on RFC 3712.

However, the LDAP-UX Client Services only supports automatically importing the LDAP printer schema into the Directory Server by running the `setup` program.

If your directory server does not support the LDAP printer schema, you may experience problems when importing the printer schema.

- Configure the Pluggable Authentication Module (PAM) by modifying the PAM configuration file `/etc/pam.conf`. See `/etc/pam.ldap` for a sample.
- Configure the Name Service Switch (NSS) by modifying the file `/etc/nsswitch.conf`. See `/etc/nsswitch.ldap` for a sample.
- Optionally modify the `disable_uid_range` flag in the `/etc/opt/ldapux/ldapux_client.conf` file to disable logins to the local system from specific users, as described in Section 2.5.6.1 (page 106).
- Optionally configure the authorization of one or more subgroups from a large repository such as an LDAP directory server. For the detailed information on how to set up the policy file, `/etc/opt/ldapux/pam_authz.policy`, see Section 5.3.4 (page 143).

After you configure your directory and the first client system, configuring additional client systems is simpler. For more information, see Section 2.5.7 (page 112).

2.4.5.1 Quick configuration

You can quickly configure a HP-UX Directory Server/Rat Hat Directory Server directory and the first client by letting most of the configuration parameters take default values as follows. For a custom configuration, see Section 2.4.5.2 (page 73).

The steps described below assume that you don't use SSL or TLS support with LDAP-UX. If you want to enable SSL support, see Section 2.4.5.2 (page 73).

NOTE: When configuring and setting up LDAP-UX, you will likely be prompted for credentials of an administrator. If you are asked to enter the credentials (password) of a user, make sure that the connection between your client and the HP-UX system (where you are running `setup`) is secured and not subject to network eavesdropping. One option to protect such communication may be to use the ssh protocol when connecting to the HP-UX host being configured.

1. Log in as root and run the `setup` program:

   ```
   cd /opt/ldapux/config
   ./setup
   ```

NOTE: To use a local-only profile, run the `setup` program using the `-l` option . Use the local-only profile for small deployments, testing purposes, and for environments where administrators lack server administrative privileges.

The `setup` program asks you a series of questions and usually provides default answers. Press the Enter key to accept the default, or change the value and press Enter. At any point during `setup`, enter Control-b to back up or Control-c to exit `setup`.

2. Choose the Directory Server as your LDAP directory server (option 1).

3. Enter either the host name or IP address of the directory server where your profile exists, or where you want to create a new profile from “Configuration worksheet” (page 347).
4. Enter the port number of the previously specified directory server that you want to store the profile from “Configuration worksheet” (page 347). The default port number is 389.
5. If the profile schema has already been imported, or if you invoked setup with the `-l` option to use a local-only profile, `setup` skips to the next step.

   Otherwise, you are asked whether you want to import the profile schema with LDAP-UX Client Services object class `DUAConfigProfile`. Enter "yes" to import the profile schema into the LDAP directory server. This must be done once (the schema is shared with subsequently-configured client systems). Enter “no” if you do not want to import the profile schema into the LDAP directory server. The `setup` program skips to the next step if you enter “no”.

NOTE: The LDAP-UX Integration product supports a local-only profile for testing, small LDAP deployments, or operating in environments where the HP-UX administrators lack directory server administrative privileges. The local-only configuration profile is intended not to push back to the directory server.

   See “LDAP-UX Client Services object classes” (page 349) for a detailed description of the `DUAConfigProfile` object class.
6. If the LDAP printer schema has already been extended, `setup` skips this steps. Otherwise, enter "yes" to extend the LP printer schema if you choose to start the printer configurator. The LDAP printer configurator is a feature that simplifies the LP printer management by refreshing LP printer configurations on your client system. A new printer schema, which is based on RFC 3712, is required to start the services.
7. If the NIS publickey schema has already extended, `setup` skips this step. Otherwise, enter "yes" to extend the publickey schema if you choose to store the public keys of users and hosts in the LDAP directory. A publickey schema, which is based on RFC 2307-bis is required to migrate the publickeys in the NIS+ credential table entries on the NIS+ server to the LDAP directory.
8. If the new automount schema has already been imported, `setup` skips to the next step. (The HP-UX Directory Server includes the new automount schema by default.)

   Otherwise, you will be asked whether you want to install the new automount schema which is based on RFC 2307-bis. Enter "yes" if you want to import the new automount schema into the LDAP directory server. Enter "no" if you do not want to import new automount schema into the LDAP directory server.
9. If you are creating a new profile, add all parent entries of the profile DN to the directory (if any). If you attempt to create a new profile and any parent entries of the profile do not already exist in the directory, `setup` will fail. For example, if your profile will be **`cn=profile1,ou=profiles,o=hp,com`**, then **`ou=profiles,o=hp.com`** must exist in the directory or `setup` will fail.
10. Next enter either the DN of a new profile, or the DN of an existing profile you want to use, from “Configuration worksheet” (page 347).

    To display all the profiles in the directory, use a command like the following:

    ```
    ldapsearch -b o=hp.com objectclass=DUAConfigProfile dn
    ```

    If you are using an existing profile, `setup` configures your client, downloads the profile, and exits. In this case, continue with the next step.
11. If you are creating a new profile, enter the DN and password of the directory user who can create a new profile from “Configuration worksheet” (page 347).
12. Next, it will prompt you for the following information:

```
Select authentication method for users to bind/authenticate to
the server
1. SIMPLE
2. SASL DIGEST-MD5
To accept the default shown in brackets, press the Return key.

Authentication method: [1]:
```

Press the return key if you choose to accept SIMPLE authentication method, type 2 if you choose SASL DIGEST-MD5 authentication method for the following prompt:

```
Authentication method: [1]:
```

13. Next enter the host name and port number of the directory where your name service data is, from "Configuration worksheet" (page 347). For high availability, each LDAP-UX client can look for name service data in up to three different directory hosts. You can enter up to three hosts, to be searched in order.
14. Enter the base DN where clients should search for name service data from "Configuration worksheet" (page 347).
15. You can quickly configure a Directory Server and the first client by accepting the remaining default configuration parameters when prompted.

    If you want to use the SASL DIGEST-MD5 authentication method, you need to configure a proxy user with its credential level.

    Using the SASL DIGEST-MD5 authentication, the password must be stored in the clear text in the LDAP directory.

    Table 2-4 shows the configuration parameters and the default values they will be configured with.

    Table 2-4 Configuration parameter default values

Parameter	Default value
Type of client binding	Anonymous
Bind time limit	5 seconds
Search time limit	no limit
Use of referrals	Yes
Profile TTL (Time To Live)	0 - infinite
Use standard RFC 2307 object class attributes for supported services	Yes
Use default search descriptions for supported services	Yes
Authentication method	Simple

    To change any of these default values, see Section 2.4.5.2 (page 73).
16. After entering all the configuration information, `setup` extends the schema, creates a new profile, and configures the client to use the directory.
17. Configure the Pluggable Authentication Module (PAM).

    Save a copy of the file `/etc/pam.conf` and edit the original to specify LDAP authentication and other authentication methods you want to use. See `/etc/pam.ldap` for a sample. You may be able to just copy `/etc/pam.ldap` to `/etc/pam.conf`. For more information about PAM, see the *pam*(3) and *pam.conf*(4) manpages. In addition, see the document *Managing Systems and Workgroups: A Guide for HP-UX System Administrators* at the following location:

    www.hp.com/go/hpux-core-docs (click **HP-UX 11i v2**)

18. Configure the Name Service Switch (NSS).

    Save a copy of the file `/etc/nsswitch.conf` and edit the original to specify the LDAP name service and other name services you want to use. See `/etc/nsswitch.ldap` for a sample. You may be able to just copy `/etc/nsswitch.ldap` to `/etc/nsswitch.conf`. See *nsswitch.conf*(4) for more information.

19. Optionally, configure the Pam Authorization Service module (PAM_AUTHZ).

    LDAP-UX Client Services provides a sample configuration file, `/etc/opt/ldapux/pam_authz.conf.template`. This sample file shows you how to configure the policy file to work with PAM_AUTHZ. You can copy this sample file and edit it using the correct syntax to specify the access rules you wish to authorize or exclude from authorization. For more detailed information on how to configure the policy file. See Section 5.3 (page 140).

    The sample `/etc/pam.conf` file in the *pam.conf*(4) manpage will help show you how to configure the `/etc/pam.conf` file to work with PAM_AUTHZ. For more detailed information about PAM_AUTHZ, see the *pam_authz*(5) manpage.

20. Optionally configure the `disable_uid_range` flag, as described in Section 2.5.6.1 (page 106).

    You can also use `pam_authz` or the `deny_local` option (in PAM_LDAP) to disable system access for accounts defined in LDAP. For more information, about the `pam_authz` service module, see Section 5.3 (page 140) or the *pam_authz*(5) manpage. For information about the `deny_local` option, see Section 2.5.6.2 (page 107).

21. "Verifying the LDAP-UX Client Services" (page 92).
22. Configure subsequent clients by running `setup` on those clients and specifying an existing configuration profile. Or for a simpler process see Section 2.5.7 (page 112).

2.4.5.2 Custom configuration

Running the `setup` program for a quick configuration, as described above, configures your client using default values where possible. If you would like to customize these parameters, proceed as follows.

If you want to use SSL or TLS, you must perform the following tasks before you run the custom configuration. See Section 2.4.6 (page 79) for details.

- Ensure that you have installed the certificate database files, `cert8.db` and `key3.db`, on your client system.
- If you choose to use TLS, set the `enable_startTLS` parameter to `1` in the `/etc/opt/ldapux/lldapux_client.conf` file to enable TLS. To use SSL, set `enable_startTLS` to `0` to disable TLS. By default, TLS is disabled.

NOTE: When configuring and setting up LDAP-UX, you will likely be prompted for credentials of an administrator. If you are asked to enter the credentials (password) of a user, you should make sure that the connection between your client and the HP-UX system (where you are running `setup`) is secured and not subject to network eavesdropping. One option to protect such communication may be to use the ssh protocol when connecting to the HP-UX host being configured.

1. Perform the steps described in Section 2.4.5.1 (page 69).

   However, after step 11, you will be asked whether you want to use SSL or not if the value of the `enable_startTLS` parameter is `0` (disabled) or undefined. Enter "yes" to the following question if you want to use SSL for the secure communication between LDAP clients and the HP-UX Directory Server or Redhat Directory Server. Enter "no" to the following question if you don't want to use SSL. Skip to step 2.

   ```
   Do you want to use SSL (y/n)?
   ```

   Otherwise, if the value of the `enable_startTLS` parameter is `1` (enabled), you will be asked whether you want to use TLS or not. Enter "yes" to the following question if you want to use TLS for the secure communication between LDAP clients and the HP-UX Directory Server or Redhat Directory Server. Enter "no" to the following question if you don't want to use TLS. Skip to step 3.

   ```
   Do you want to use TLS (y/n)?
   ```

2. Next, it will prompt you for selecting the authentication method for users to bind/authenticate to the server.

   You have a choice between SIMPLE (the default), SASL/GSSAPI, or SASL/DIGEST-MD5 if you choose to not enable SSL. However, you have a choice between SIMPLE with SSL (the default), or SASL/GSSAPI or SASL/DIGEST-MD5 with SSL if you choose to enable SSL.

   LDAP-UX supports the SASL/GSSAPI or SASL/DIGEST-MD5 authentication method. SASL/GSSAPI is only supported for LDAP-UX used with Windows ADS.

   If you select SASL DIGEST-MD5, two additional prompts will appear. The first will prompt you for a user mapping (UID, DN, or Other). The second will prompt you for a single realm to use when retrieving user authentication information. If no realm is specified, user information will be retrieved from the first realm the directory server offers.

3. Specify the host name and optional port number where your directory is running. If you choose to use TLS, the default directory port number is 389. If you choose to use SSL, the default directory port number is 636.

   For high availability, each LDAP-UX client can look for user and group information in up to three different directory servers. You are able to specify up to three directory hosts, to be searched in order.

4. Reply "no" when asked if you want to accept the remaining default configuration parameters.

5. Select the client binding you want from "Configuration worksheet" (page 347). This determines the identity that client systems use when binding to the directory to search for user and group information.
6. If you configured a proxy user, enter the DN and password of your proxy user, from "Configuration worksheet" (page 347).

   If you want to use the SASL/DIGEST-MD5 authentication method, you need to configure a proxy user with its credential level.

   Using the SASL/DIGEST-MD5 authentication, the password must be stored in the clear text in the LDAP directory.
7. Enter the maximum time in seconds the client should wait for directory searches before aborting. Enter 0 for no time limit.
8. Enter whether or not you want directory searches to follow referrals. Referrals are a redirection mechanism supported by the LDAP protocol. Please see your directory manuals for more information on referrals.

NOTE: If you want your directory searches to follow referrals, you must allow anonymous access into your directories.

9. Enter the Profile TTL (Time To Live) value. This value defines the time interval between automatic downloads (refreshes) of new configuration profiles from the directory. Automatic refreshing ensures that the client is always configured using the newest configuration profile.

   If you want to disable automatic refresh or manually control when the refresh occurs, enter a value of 0. Section 2.5.8 (page 113).
10. In this step, the `setup` program initiates a dialog where you can remap the standard object class attributes to alternate attributes. You may want to do this if the attributes in your directory do not conform to the object classes defined in RFC 2307.

    You can remap the attributes for any of the supported services: `passwd`, `shadow passwd`, `group`, `PAM`, `netgroup`, `rpc`, `protocols`, `networks`, `hosts`, `services` and `automount`.

NOTE: Make sure that the attribute names are entered correctly to avoid unpredictable results later.

For a description of the standard object classes and attributes, see RFC 2307 at:

http://www.ietf.org/rfc/rfc2307.txt.

At this point, the `setup` program will display the following dialog:

```
LDAP-UX Client Services supports the following services:
1.Password                                7.Networks
2.Shadow passwd                           8.Hosts
3.Group                                   9.Services
4.PAM (Pluggable Authentication Module)10.Printers
5.RPC                                    11.Automount
6 Protocols                              12.Netgroup

Each services uses a standard object class (defined by RFC 2307)
You can remap any of these attributes to alternate attributes.
Do you want to remap any of the standard RFC 2307 attributes?
```

Enter "yes" if you want to remap attributes for any of the supported services. Then go to Section 2.4.5.3 (page 76) for details of the procedures.

Otherwise, if you do not want to remap attributes for any of the supported services, then enter "no" to this prompt to continue to the next step.

11. In this step, the `setup` program initiates a dialog where you can create a custom search descriptor. A custom search descriptor allows you to specify a different search location or filter for retrieving entries for services supported by LDAP-UX Client. Each name service

can have up to three different search descriptors. A custom search descriptor consists of three parts: a search base DN, scope, and filter. The client uses the search descriptors in order until it finds what it is looking for.

NOTE: If your search filters overlap, enumeration requests will result in duplicate entries being returned. For example, if one search filter searched a subset of your organization and a second search filter searched your entire organization, an enumeration request would return duplicate entries.

See the "Minimizing Enumeration Requests" section for more information.

To begin the process to create custom search descriptors, `setup` will prompt you for the following information:

```
LDAP-UX Client Services supports the following services:

1.Password                                  7.Networks
2.Shadow passwd                             8.Hosts
3.Group                                     9.Services
4.PAM (Pluggable Authentication Module)10.Printers
5.RPC                                      11.Automount
6.Protocols                                12.Netgroup

You can create up to three custom search descriptors for each name
service to search different locations in the directory for user
and group information.
Do you want to create custom search descriptors? [No]:
```

Enter 'yes' if you want to create custom search descriptors for any of the supported services. Then enter the number of the service for which you want to create a custom search descriptor.

If, you do not want to create custom search descriptors, enter 'no' to this prompt to continue to the next step.

Creating the nisObject search filter

LDAP-UX Client Services uses the automount search filter for the automount service as default. If you want to create the `nisObject` search filter for the automount service to search a different location in the directory, use the following steps:

1. Type yes for the following question and press the return key:

   ```
   Do you want to create custom search descriptors? [No]:yes
   ```

2. Next, it will take you to the screen which shows you the following information:

   ```
   To accept the default shown in brackets, press the Return key.
   search base [dc=cup,dc=hp,dc=com]:
   search scope (base, one, sub) [sub]
   Search filter [(objectclass=automount)]
   ```

   If you want to create the `nisObject` search filter for the automount service, then type `(objectclass=nisObject)` for the following prompt and press the Return key; otherwise press the return key to accept the default search filter, `objectclass=automount`:

   ```
   Search filter [(objectclass=automount)]: (objectclass=nisObject)
   ```

12. You will be asked whether or not you want to start the client daemon. For LDAP-UX Client B.03.20 or later versions, the client daemon must be started for LDAP-UX functions to work. With LDAP-UX Client B.03.10 or earlier, the client daemon is optional, and should be turned on in order to provide better prformance (response time) and for the X.500 group membership to work.

2.4.5.3 Remapping attributes for services

This section describes detailed procedures on how to perform attribute mappings for automount, dynamic group and X.500 group membership services.

Attribute mappings for automount service

By default, LDAP-UX Client Services uses the RFC 2307-bis automount schema. The `nisObject` automount schema can also be used if configured via attribute mappings.

Use the following steps if you want to remap the automount attributes to the `nisObject` automount attributes:

1. Enter yes for the following question:

   ```
   Do you want to remap any of the standard RFC 2307 attributes? [yes]:
   ```
   yes

2. If you want to select the `automount` service, then enter 11 for the following question and press the return key:

   ```
   Specify the service you want to map? [0]:11
   ```

3. Next, it will take you to the screen which shows you the following information:

   ```
   Current Automount attribute names:
   1.automountMapName ->[automountMapname]
   2.automountKey -> [automountKey]
   3.automountInformation -> [automountInformation]
   Specify the attribute you want to map. [0]:
   ```

   You type 1 for the following question and press the return key:

   ```
   Specify the attribute you want to map. [0]:1
   ```

4. Next, type the attribute `nisMapName` that you want to map to the `automountMapName` attribute for the following question and press the return key:

   `automountMapName` -> `nisMapName`

5. Next, it will take you to the screen which shows you the following information:

   ```
   Current Automount attribute names:
   1.automountMapName ->[nisMapname]
   2.automountKey -> [automountKey]
   3.automountInformation -> [automountInformation]
   Specify the attribute you want to map. [0]:
   ```

   If you want to specify the attribute to map to the `automountKey` attribute, then type 2 for the following question and press the return key:

   ```
   Specify the attribute you want to map. [0]:2
   ```

6. Next, type the attribute `cn` you want to map to the `automountKey` attribute and press the return key:

   `automountKey` -> cn

7. Next, it will take you to the screen which shows you the following information:

   ```
   Current Automount attribute names:
   1.automountMapName ->[nisMapname]
   2.automountKey -> [cn]
   3.automountInformation -> [automountInformation]
   Specify the attribute you want to map. [0]:
   ```

   If you want to specify the attribute to map to the `automountInformation` attribute , then type 3 for the following question and press the return key:

   ```
   Specify the attribute you want to map. [0]:3
   ```

8. Next, type the attribute `nisMapEntry` you want to map to the `automountInformation` attribute and press the return key:

   `automountInformation` -> nisMapEntry

9. Next, it will take you to the screen which shows you the following information:

```
Current Automount attribute names:
1.automountMapName ->[nisMapname]
2.automountKey -> [cn]
3.automountInformation -> [nisMapEntry]
Specify the attribute you want to map. [0]:
```

   You type 0 to exit this menu for the following question:

```
Specify the attribute you want to map. [0]:0
```

Attribute mappings for dynamic group support

If you are configuring dynamic group support, you need to remap the default group member attribute, `memberuid`, to `memberURL` (for HP-UX Directory Server or Redhat Directory Server). For detailed information about dynamic group support, see "Dynamic group support" (page 121).

Use the following steps to remap the `memberuid` attribute to the dynamic group attributes, `memberURL` or `nxsearchFilter`. For example, the following procedures are used to remap `memberuid` to `memberURL`:

1. Type yes for the following question:

```
Do you want to remap any of the stantdard RFC 2307 attributes? [yes]:
yes
```

2. Select the `group` service by entering 3 for the following question and press the return key:

```
Specify the service you want to map? [0]: 3
```

3. Next, it will take you to the screen which shows you the following information:

```
Current Group attribute names:
1.cn ->[cn]
2.gidnumber  -> [gidnumber]
3.memberuid -> [memberuid]
4.userpassword -> [userPassword]
Specify the attribute you want to map. [0]:
```

   If you want to specify the attribute to map to `memberuid`, then type 3 for the following question and press the return key:

```
Specify the attribute you want to map? [0]: 3
```

4. Type the attribute, `memberURL` or `nxsearchFilter`, that you want to map to the `memberuid` attribute and press the return key:

   `memberuid` —> memberURL

5. Next, it will take you to the screen which shows you the following information:

```
Current Group.attribute names:
1.cn ->[cn]
2.gidnumber  -> [gidnumber]
3.memberuid -> [memberURL]
4.userpassword -> [userPassword]
Specify the attribute you want to map. [0]:
```

   You type 0 to exit this menu for the following question:

```
Specify the attribute you want to map. [0]:0
```

Attribute mappings for X.500 group membership support

If you want to configure X.500 group membership support, you should remap the group member attribute to `member` or `uniquemember` instead of using the default attribute, `memberuid`.

Perform the following steps for attribute mappings to set up X.500 group membership:

1. Type yes for the following question:

   ```
   Do you want to remap any of the startdard RFC 2307 attributes? [yes]:
   ```
   yes

2. Select the `group` service by entering 3 for the following question and press the return key:

   ```
   Specify the service you want to map? [0]: 3
   ```

3. Next, it will take you to the screen which shows you the following information:

   ```
   Current Group attribute names:
   1.cn ->[cn]
   2.gidnumber  -> [gidnumber]
   3.memberuid -> [memberuid]
   4.userpassword -> [userPassword]
   Specify the attribute you want to map. [0]:
   ```

   If you want to specify the attribute to map to `memberuid`, then type 3 for the following question and press the return key:

   ```
   Specify the attribute you want to map? [0]: 3
   ```

4. Type the `member` attribute that you want to map to the `memberuid` attribute and press the return key:

   `memberuid` –> member

5. Next, it will take you to the screen which shows you the following information:

   ```
   Current Group.attribute names:
   1.cn ->[cn]
   2.gidnumber  -> [gidnumber]
   3.memberuid -> [member]
   4.userpassword -> [userPassword]
   Specify the attribute you want to map. [0]:
   ```

   You type 0 to exit this menu for the following question:

   ```
   Specify the attribute you want to map. [0]:0
   ```

NOTE: LDAP-UX supports DN-based (X.500 style) membership syntax. This means that you do not need to use the memberUid attribute to define the members of a POSIX group. Instead, you can use either the `member` or `uniqueMember` attribute. LDAP-UX can convert from the DN syntax to the POSIX syntax (an account name).

For HP-UX Directory Server or Redhat Directory Server, the typical member attribute would be either `memberUid`, `member` or `uniqueMember`.

2.4.6 Configuring the LDAP-UX Client Services with SSL or TLS support

The LDAP-UX Client Services supports either SSL (Secure Socket Layer) or TLS (Transport Layer Security) to secure communication between LDAP clients and the LDAP directory server.

With SSL, an encrypted session is established on an encrypted port, 636. The LDAP-UX Client Services supports SSL with a password as the credential, using either simple bind or SASL/GSSAPI, or SASL/DIGEST-MD5 authentication to ensure confidentiality and data integrity between clients and servers. (SASL/GSSAPI is only supported for LDAP-UX used with Windows ADS.) SSL enables LDAP-UX clients to provide a secure way to protect the password over the network. In addition, SSL/TLS can be used to validate the identity of the directory server if the privacy of the server's and CA's private keys can be assured.The directory administrator can choose the authentication mechanism, such as using a simple password stored in the directory server as a hash syntax.

The LDAP-UX Client Services supports SSL communication with Microsoft Windows Server 2003 R2 and 2008 Active Directory Server (ADS), HP-UX Directory Server 8.1 (or later), and Red Hat Directory Server 8.0. For detailed information about how to set up and configure your directory server to enable SSL communication over LDAP, see the appropriate administrator's guide at the following location:

http://www.hp.com/go/hpux-security-docs

Starting with LDAP-UX Client Services B.04.10, LDAP-UX Client Services supports a new extension operation of TLS protocol called startTLS to secure communication between LDAP clients and the LDAP directory server. By default, an encrypted session is established on an un-encrypted port, 389. If an encrypted port is used, it will fail to establish the secure connection. The TLS protocol provides administrators better flexibility for using TLS in their environment by allowing the use of an un-encrypted LDAP port for communication between clients and server. LDAP-UX supports TLS with password as the credential, using either simple bind or SASL/GSSAPI, or SASL/DIGEST-MD5 authentication to ensure confidentiality and data integrity between clients and servers.

The LDAP-UX Client Services supports TLS communication with Microsoft Windows Server 2003 R2 and 2008 Active Directory Server (ADS), HP-UX Directory Server 8.1 (or later), and Red Hat Directory Server 8.0.

2.4.6.1 Configuration parameters

LDAP-UX Client Services provides the following parameter in the `/etc/opt/ldapux/ldapux_client.conf` file to support TLS:

`enable_startTLS` This integer variable controls whether the TLS feature is enabled or disabled. The valid values of this parameter are `1` and `0`. If you choose to use TLS, set this parameter to `1`. To disable TLS, set this variable to `0`. By default, TLS is disabled. If the enable_startTLS parameter is undefined or does not exist, it is processed as the TLS feature is disabled.

If you want to use SSL or TLS, you must perform the following tasks before you run the `setup` program:

- Ensure that the certificate database files `cert8.db` and `key3.db` are on your client system. For more information, see Section 2.4.6.2 (page 79).
- If you choose to use TLS, set the `enable_startTLS` parameter to `1` in the `/etc/opt/ldapux/lldapux_client.conf` file. To use SSL, set `enable_startTLS` to `0`. By default, TLS is disabled.

2.4.6.2 Configuring the LDAP-UX client to use SSL or TLS

You can choose to enable SSL or TLS with LDAP-UX when you run the `setup` program. If you attempt to use SSL or TLS, you must install the Certificate Authority (CA) certificate on your

LDAP-UX Client and configure your LDAP directory server to support SSL or TLS before you run the `setup` program.

NOTE: If you already have the certificate database files `cert8.db` and `key3.db` on your client for your HP-UX applications, you can simply create a symbolic link `/etc/opt/ldapux/cert8.db` that points to `cert8.db`, and `/etc/opt/ldapux/key3.db` that points to `key3.db`.

2.4.6.2.1 Steps to create certificate database files using the certutil utility

The following steps show how you can create the security database files, `cert8.db` and `key3.db` on your client system using the Certificate Database Tool command line utility (`certutil`):

1. Retrieve the certificate. The procedure for this varies, depending on several factors. If your organization is using either a certificate management system internal to the organization, or a third-party certificate authority, you will usually use a web browser to download a Certificate Authority (CA) certificate. The certificate is downloaded in one of two forms: ASCII-encoded PEM form, or binary DER form.

   In PEM form the certificate looks similar to this:

   ```
   --------------- BEGIN CERTIFICATE --------------------------------
   -MIICJjCCAY+gAwIBAgIBJDANBgkghkiG9w0BAQQFADBxMQswCQYDVQQGEwJVUzEL
   MAkga1UECBMCQ2ExEjAQBgNVBAcTCWN1cGVvsG1ubzEPMA0GA1UEChmgAhaUy29T
   MRIwEAYDVQQLEw1RR1NMLUxkYXAxHDAaBgNVBAMTE0N1cnRpzmljYXR1IE1hbmFn
   4I2vvzz2i1Ubq+Ajcf1y8sdafuCmqTgsGUYjy+J1weM061kaWOt0HxmXmrUdmenF
   skyfHyvEGj8b5w6ppgIIA8JOT7z+F0w+/mig=
   --------------- END CERTIFICATE ----------------------------------
   ```

   As an alternative to installing the CA certificate, you can install and trust the LDAP server's own certificate rather than the CA certificate that is issued with the LDAP server's certificate. Because LDAP-UX only accepts connections to the LDAP server for which the server certificate is valid, this alternative establishes a more narrow scope of trust. So, if you plan to connect to multiple LDAP servers, you must install multiple server certificates. Additionally, because server certificates tend to have a validity range shorter than that of CA certificates, you may find yourself needing to update the certificate more often.

2. Use the `rm` command as in the following example to remove the old database files `/etc/opt/ldapux/cert8.db` and `/etc/opt/ldapux/key3.db`:

   ```
   # rm -f /etc/opt/ldapux/cert8.db /etc/opt/ldapux/key3.db
   ```

3. Create new certificate database files, using the command shown in the following example.

   ```
   # /opt/ldapux/contrib/bin/certutil -d /etc/opt/ldapux –N
   ```

   The `certutil` tool will prompt you to enter a password to protect the private key database. If you will not be storing any private keys in the certificate database files, press **Enter** to leave the password empty . LDAP-UX does not require a private key; however, if you plan to use these certificate database files with other applications that make use of a private key, you should set a password.

4. Add the downloaded CA certificate to the certificate database created in the preceding step. If the CA certificate was downloaded in binary DER form, use the following command:

   ```
   # /opt/ldapux/contrib/bin/certutil -d /etc/opt/ldapux -A -n "CA
   cert" -t “CT,,” -i cacert.der
   ```

   If the CA certificate was downloaded in ASCII-encoded PEM form, use the –a (ASCII) option as in the following example:

   ```
   # # /opt/ldapux/contrib/bin/certutil -d /etc/opt/ldapux -A -n "CA
   cert" -t “CT,,” -i cacert.pem -a
   ```

   If the certificate is a server certificate, use the “`P,,`” trust flag:

```
# # /opt/ldapux/contrib/bin/certutil -d /etc/opt/ldapux -A -n "server
cert" -t “P,,” -i servercert.der
```

NOTE: The required `-n` parameter gives the certificate a nickname in the certificate database files. The nickname value is arbitrary. If you plan to connect to multiple LDAP servers that were issued SSL certificates by different certificate authorities, you should use the nickname to help differentiate between the different CA certificates. For example, you might name one `Issuer1 CA cert` and the other `Issuer2 CA cert`.

The `-t` parameter sets the trust bits for the certificate. For CA certificates, use “`CT,,`” to indicate that the certificate is trusted as an issuer of SSL certificates. For server certificates, use “`P,,`” to indicate that the certificate represents a trusted peer.

For more information about using the `certutil` utility, see the following website:

http://www.mozilla.org/projects/security/pki/nss/tools/certutil.html

2.4.6.2.2 Adjusting the peer certificate policy

With SSL/TLS, not only communication between clients (LDAP-UX) and servers (the LDAP directory server) can be protected, but in addition, specific levels of assurance of the identities of the clients and servers can be validated. This section describes how to adjust this validation level.

The `peer_cert_policy` parameter in the `/etc/opt/ldapux/ldapux_client.conf` configuration file is a string variable used to control the validation level. There are three valid options for this parameter described below:

`WEAK` Performs no validation of SSL or TLS certificates. Communication between the client and server can be encrypted, however the client has no assurance that it is communicating with a trusted server.

`CERT` Verifies that the issuers of peer SSL or TLS certificates are trusted. Communication between the client and server can be encrypted and the client has some assurance that it is communicating with a trusted server. In this scenario, it is still possible for the server to have a certificate that has been issued for a different server if methods used to protect private keys of server certificates are not in place. `CERT` is the default mode of operation with LDAP-UX.

`CNCERT` Performs both the CERT check and also verifies that the common name or `subjectAltName` values embedded in the certificate matches the address used to connect to the LDAP server, as described in RFC 4513.

As mentioned above, the default mode of operation for LDAP-UX is `CERT`. Increasing certificate validation level to `CNCERT` requires additional and specific configuration steps. If not properly established, it can interfere with LDAP-UX and proper system operation. Because LDAP-UX can be used for host-name resolution (similar to DNS), LDAP-UX normally stores the IP address of LDAP servers in the configuration profile. This procedure assures that if LDAP-UX is asked to resolve a host name, it can do so without first needing to resolve the host name of the LDAP directory server (which could lead to a catch-22). However, since certificates normally embed the host name or fully qualified host name and LDAP-UX only has the IP address of the host, it is not possible for LDAP-UX to verify the host name on the certificate.

If you want to configure the `CNCERT` validation level with the `peer_cert_policy` parameter, you must manually execute the following configuration steps:

1. Update the `preferredserverlist` setting in the profile to contain the host name of the LDAP server such that it matches the host name specified in the LDAP server’s certificate. See the “Modifying preferredserverList in the LDAP-UX Profile” section for details.

2. Select and execute one of the following steps:
   - Either LDAP-UX must not be used for host-name resolution by removing "ldap" from the "hosts" service in the /etc/nsswitch.conf file.
   - Or the host name and IP address must be provided by some other name resolution service, such as "files" or "dns", and that service must appear before "ldap" in the /etc/nsswitch.conf file for the "hosts" service.

2.4.6.2.2.1 Modifying preferredServerList in the LDAP-UX profile

Use the following steps to modify the value of the preferredServerList attribute in the LDAP-UX configuration profile:

1. Run the following steps to find the name of the LDAP server used on the server certificate. Assuming this certificate has been installed in your local certificate database file, /etc/opt/ldapux/cert8.db:
   - Run the following commands to list all server certificates used by LDAP-UX:

     ```
     cd /etc/opt/ldapux
     certutil -d . -L
     ```

   - Run the following command to select the nickname of the certificate from the above list:

     ```
     certutil -d . -L -n <selected nickname>
     ```

   - Select the first name component of the "Subject:" name. For example, if the "Subject:" string is "CN=ldapserver.example.com, O=Example Corp" then the name component would be "ldapserver.example.com".

NOTE: Depending on how your certificate administrator manages your network, the above server certificate may not be found in your cert8.db file. Instead you may only find certificates for any trusted Certificate Authorities. In this case, contact your certificate administrator for the LDAP server certificate details.

2. In a separate window, use the ldapentry tool to modify the value of the preferredServerList attribute with the LDAP server name found in step 1. See Section 5.12 (page 183) for detailed information on changing LDAP-UX configuration profile settings manually.
3. Examine the "preferedServerList" attribute
4. Use the nslookup tool to verify the IP address specified in the preferred server list matches that of the name of the host name found in step 1 above.

   For example, if the preferredserverlist attribute value is 192.168.1.1:636 and "Subject" is CN=ldapserver.example.com,O=Example Corp, then

   ```
   $ nslookup 192.168.1.1
   Name Server:  dns-resolver.example.com
   Address:  192.169.1.254

   Trying DNS
   Name:  ldapserver.example.com
   Address:  192.168.1.1
   ```

2.4.6.3 SSL/TLS ciphers

The SSL/TLS protocols support a variety of different cryptographic algorithms called ciphers for use in operations such as authenticating the server and client to each other, transmitting certificates, and establishing session keys. When an LDAP client connects to an LDAP directory server, the server usually picks the strongest cipher supported by both client and server. Clients

and servers may support different cipher suites, or sets of ciphers, depending on a variety of factors. The ciphers currently supported by LDAP-UX are listed in Table 2-5 (page 83).

Table 2-5 Supported ciphers

Version	Key exchange	Encryption	Key length	Message authentication
SSL3 and TLS	RSA (A public-key algorithm for both encryption and authentication)	RC4 (Rivest encryption)	128	MD5 (Message Digest algorithm)
SSL3 and TLS	RSA	3DES (Data Encryption Standard applied three times)	168	SHA1 (Secure Hash Algorithm)
SSL3 and TLS	RSA	DES (Data Encryption Standard)	56	SHA1
SSL3 and TLS	RSA	RC4	40	MD5
SSL3 and TLS	RSA	RC2	40	MD5
TLS	RSA (1024–bit public key)	RC4	56	SHA1
TLS	RSA (1024–bit public key)	DES	56	SHA1

If vulnerabilities are discovered in cipher systems, administrators can use this list to determine whether the cited vulnerabilities might affect their systems. If a cipher with a known vulnerability is indeed being used, the appropriate administrator can disable the cipher in the central directory server (not in LDAP-UX). For information about managing available ciphers for use with HP-UX Directory Server, see the *HP-UX Directory Server administrator guide*.

2.4.7 Configuring LDAP-UX Client Services with NIS publickey support

LDAP-UX Client Services supports discovery and management of NIS publickeys in an LDAP directory. Both public and secret keys, used by the Secure RPC API can be stored in user and host entries in an LDAP directory server, using the`nisKeyObject` objectclass. Support for discovery of keys in an LDAP directory server is provided through the `getpublickey()` and `getsecretkey()` APIs. You can use `chkey` and `newkey` commands to manage user and host keys in an LDAP server. The `chkey -s ldap` command is used to change user's secure RPC public key and secret key in an LDAP directory. The `newkey -u <username> -s ldap` command is used to add new keys for users to an LDAP directory while the `newkey -h <hostname> -s ldap` command is used to create new keys for machines to an LDAP directory.

For detailed information on the `newkey` and`chkey` commands, see the *newkey*(1M),*chkey*(1), *getpublickey*(3N), *getsecretkey*(), and *publickey*(4) manpages.

2.4.7.1 HP-UX Enhanced Publickey-LDAP software requirement

Support for NIS publickey through LDAP requires functionality enhancement in LDAP-UX Client Services and an enhancement in the ONC product. ONC with publickey LDAP support is available through the HP-UX Enhanced Publickey-LDAP Software Pack (SPK) web release.

To enable the publickey LDAP support, you must install the appropriate Enhanced Publickey-LDAP software bundle listed in Table 2-6 (for HP-UX 11i v2 only; no patch is required for HP-UX 11i v3) and LDAP-UX Client Services B.04.00 or later on your client systems. The software bundle contains all the required patches plus the enablement product for this new feature. For detailed information, see the *ONC with Publickey LDAP Support Software Pack Release Notes* at the following website:

http://www.hp.com/go/hpux-networking-docs (click **HP-UX 11i v2 Networking Software**)

Navigate to `NFS Services`.

Table 2-6 Enhanced Publickey-LDAP software requirement

Operating System Supported	Software Bundle Version	Release Date
`HP-UX 11i v2`	`Enhkey B.11.23.01`	October, 2006

You can download the Enhanced Publickey-LDAP software bundle from the following Software Depot website:

- Go to http://www.hp.com/go/softwaredepot.
- Click on **Enhancement releases and patch bundles**.
- Select the link:
  - **HP-UX Software Pack (Optional HP-UX 11i v2 Core Enhancements)**
- Select the link:
  - **PublicKey-LDAP** (for HP-UX 11i v2)
- Select and download the following software bundle, place it to on your client system (/tmp):
  - `Enhkey_B.11.23.01_HP-UX_B.11.23_IA_PA depot` for HP-UX 11i v2
- Use `swinstall` to install the software bundle:
  - `swinstall -x autoreboot=true -x reinstall=false -s /tmp/ENHKEY_B.11.23.01_HP-UX_B.11.23_IA_PA.depot` for HP-UX 11i v2

2.4.7.2 Extending the NIS publickey schema into your directory

The NIS publickey schema is not loaded in the HP-UX Directory Server or Redhat Directory Server. If you are installing LDAP-UX B.04.00 or later on your client system, the `setup` program will extend the publickey schema into your Directory Server. If you previously configured LDAP-UX B.03.30 or earlier version, and now update the product to version B.04.00 or later, you

must re-run the `setup` program to extend the publickey schema into your LDAP directory. You do not need to re-run the `setup` program for the subsequent client systems. For detailed information on how to run the `setup` program to extend the publickey schema into an LDAP directory, see Section 2.4.5.1 (page 69).

2.4.7.3 Admin Proxy user

A special type of proxy user, known as an Admin Proxy has been added to LDAP-UX to support management of NIS publickey information in an LDAP directory server. The Admin Proxy represents the HP-UX administrator's rights in the directory server and typically is used to represent root's privileges extended to the directory server. Only an Admin Proxy user is allowed to use the `newkey` tool to add host and user keys into the LDAP directory server, or to use the `chkey` tool to modify host keys in the LDAP directory server.

2.4.7.3.1 Configuring an Admin Proxy user by using ldap_proxy_config

You need to use a new `ldap_proxy_config` tool option `-A` to configure an Admin Proxy user. You must specify the `-A` option along with other options to perform operations applying to an Admin Proxy user. For example, you can use the `ldap_proxy_config -A -i` command to create an Admin Proxy user. See Section 7.2.6 (page 216) for details.

2.4.7.3.2 Password for an Admin Proxy user

To protect user secret keys in the LDAP directory, the secret keys are encrypted using the user's password. This process is used in NIS as well as NIS+ environments. The host's secret key must also be encrypted. Since the host itself does not have its own password, root's password is used to encrypt the host's secret key. The `chkey` or `newkey` command prompts for root's password when changing or adding a key for a host. For this reason, you may wish to configure the Admin Proxy user in the LDAP directory to have the same password as the root user on the master host. Although it is not required that the Admin Proxy user and root user share the same password, it allows you to avoid storing the Admin Proxy user's password in the administrator's credential file `/etc/opt/ldapux/acred` (this file as well as the `pcred` file are not encrypted). In this case, when you run the `ldap_proxy_config -A -i` command to configure the Admin Proxy user, you enter only Admin Proxy user's DN without the password. LDAP-UX will use the root's password given to the `chkey` and `newkey` commands as the Admin Proxy user's password to perform public key operations. However, the `ldap_proxy_config -A -v` command will not be able to validate the Admin Proxy user because no password is available to `ldap_proxy_config`. As a result, the message `"No password is provided. Validation is not performed"` will be displayed.

2.4.7.4 Setting ACI for key management

Before storing public keys in an LDAP server, LDAP administrators may wish to update their LDAP access controls such that users can manage their own keys, and the Admin Proxy user can manage host keys. This section describes how you set up access control instructions (ACI) for an Admin Proxy user or a user.

2.4.7.4.1 Setting ACI for an Admin Proxy user

With the HP-UX Directory Server, you can use the Directory Server Console or the `ldapmodify` command to set up an ACI, which gives an Admin Proxy user permissions to manage host and user keys in the LDAP directory.

An Example

The following ACI gives the permissions for the Admin Proxy user `uid=keyadmin` to read, write, and compare `nissecretkey` and `nispublickey` attributes for hosts and users:

```
dn:dc=org,dc=hp,dc=com
```

```
aci:(targetattr ="objectclass||nispublickey||nissecretkey")
 (version 3.0;acl "Allow keyadmin to change key pairs";
 allow (read,write,compare)
 userdn="ldap:///uid=keyadmin,ou=people,dc=org,dc=hp,dc=com";)
```

2.4.7.4.2 Setting ACI for a user

With the HP-UX Directory Server, you need to set up an ACI which gives a user permission to change his own `nissecretkey` and `nispublickey` attributes. To set up ACI for a user, use the Directory Server Console or `ldapmodify`.

An Example

The following ACI gives a user permission to change his own `nissecretkey` and `nispublickey` attributes for user keys:

```
dn:ou=People,dc=org,dc=hp,dc=com
aci:(targetattr ="nissecretkey||nispublickey")(version 3.0;
 acl "Allow key self modification";allow (write)
 (userdn = "ldap:///self");)
```

2.4.7.5 Configuring serviceAuthenticationMethod

`serviceAuthenticationMethod` is a newly supported attribute of the configuration profile, `/opt/ldapux/ldapux_profile.ldif`. It's function is the same as `authenticationMethod`, but it allows authentication configuration for specific name services. The `serviceAuthenticationMethod` attribute is created to resolve issues that may arise when the default authentication method is not considered secure enough for specific name services. For example, if the default `authenticationMethod` is configured as `NONE` then the `newkey` and `chkey` commands would not know how to properly bind to the directory server when changing or adding key pairs. LDAP-UX only supports the `serviceAuthenticationMethod` attribute for the `keyserv` service, since the `keyserv` service is the only one that currently needs modification of privileges in the directory server.

To perform `newkey` and `chkey` operations, LDAP-UX binds the Admin Proxy user to the LDAP directory using the authentication method specified in `serviceAuthenticationMethod`. LDAP-UX only supports `serviceAuthenticationMethod` for `keyserv`. Any other services configured in `serviceAuthenticationMethod` will be ignored.

Configuring `serviceAuthenticationMethod` is optional. If you do not configure `serviceAuthenticationMethod`, LDAP-UX binds the Admin Proxy user to the LDAP directory using the authentication method specified for the proxy user.

2.4.7.5.1 Authentication methods

LDAP-UX Client Services supports the following authentication methods for the `keyserv` service:

- simple with SSL enabled
- SASL/GSSAPI or SASL/DIGEST-MD5 with SSL enabled
- simple with SSL disabled
- SASL/GSSAPI or SASL/DIGEST-MD5 with SSL disabled

NOTE: SASL/GSSAPI is only supported for LDAP-UX used with Windows ADS.

SSL settings for both `authenticationMethod` and `serviceAuthenticationMethod` must be set the same. It is not supported to have SSL enabled for `authenticationMethod` and SSL disabled for `serviceAuthenticationMethod`, or vice versa.

2.4.7.5.2 Procedures used for configuring serviceAuthenticationMethod

Use the following steps on one of LDAP-UX client sytems to configure the `serviceAuthenticationMethod` attribute in the `/etc/opt/ldapux/ldapux_profile.ldif` file:

1. Log in as `root`.
2. Use the `ldapentry` tool to modify the profile entry in the LDAP directory server to include `serviceAuthenticationMethod`. To do this, `ldapentry` requires the profile DN. You

can find the profile DN from PROFILE_ENTRY_DN in /etc/opt/ldapux/ldapux_client.conf after you finish running the setup program. The following example edits the profile entry "cn=ldapuxprofile,dc=org,dc=hp,dc=com":

For example:

```
cd /opt/ldapux/bin
./ldapentry -m "cn=ldapuxprofile,dc=org,dc=hp,dc=com"
```

After you enter the prompts for "Directory login:" and "password:", ldapentry will bring up an editor window with the profile entry. You can add the serviceAuthenticationMethod attribute.

The value of the serviceAuthenticatioMethod entry depends on the authentication method you configure. The following shows the possible values of the serviceAuthenticationMethod attribute:

- For SASL /DIGEST-MD5 using the Distinguish Name (DN) to generate the DIGEST-MD5 hash, the data in the entry is:

  ```
  serviceAuthenticationMethod:keyserv:sasl/digest-md5:username=dn
  ```

- For SASL /DIGEST-MD5 using the uid attribute to generate the DIGEST-MD5 hash, the data in the entry is:

  ```
  serviceAuthenticationMethod:keyserv:sasl/digest-md5
  ```

- For SASL/DIGEST-MD5 with SSL enabled using the DN to generate the DIGEST-MD5 hash, the data in the entry is:

  ```
  serviceAuthenticationMethod:keyserv:tls:sasl/digest-md5:username=dn
  ```

- For SASL/DIGEST-MD5 with SSL enabled using the uid attribute to generate the DIGEST-MD5 hash, the data in the entry is:

  ```
  serviceAuthenticationMethod:keyserv:tls:sasl/digest-md5
  ```

- For simple authentication, the data in the entry is:

  ```
  serviceAuthenticationMethod:keyserv:simple
  ```

- For simple with SSL enabled, the data in the entry is:

  ```
  serviceAuthenticationMethod:keyserv:tls:simple
  ```

For more information on ldapentry, see "Command and tool reference" (page 211).

NOTE: If you use TLS for secure communication between LDAP clients and the HP-UX Directory Server or Redhat Directory Server, you need to use the Directory Server Console to manually add the values of the serviceAuthenticationMethod attribute.

3. Go to /opt/ldapux/config:

   ```
   cd /opt/ldapux/config
   ```

4. Use /opt/ldapux/config/get_profile_entry to download the modified LDIF profile:

   ```
   ./get_profile_entry -s nss
   ```

5. Run the /opt/ldapux/config/display_profile_cache tool to check the configuration of the serviceAuthenticationMethod attribute:

   ```
   ./display_profile_cache
   ```

   For example:

   If the serviceAuthenticationMethod:keyserv:sasl/digest-md5 entry is added to the profile entry in the LDAP directory, you can see the following information when you run the display_profile_cache tool:

```
serv-auth: keyserv:sasl/digest-md5
auth opts: username: uid
realm:
```

For subsequent LDAP-UX client systems that share the same profile configuration, use the following steps to download and activate the profile:

1. Log in as `root`.
2. Go to `/opt/ldapux/config`:

   ```
   cd /opt/ldapux/config
   ```

3. Use `/opt/ldapux/config/get_profile_entry` to download the modified LDIF profile:

   ```
   ./get_profile_entry -s nss
   ```

4. Run the `/opt/ldapux/config/display_profile_cache` tool to check the configuration of the `serviceAuthenticationMethod` attribute:

   ```
   ./display_profile_cache
   ```

5. Restart the LDAP-UX client daemon, `ldapclientd`, if you change the authentication method from non-SSL to SSL. Otherwise, skip this step.

2.4.7.6 Configuring Name Service Switch (NSS)

Configure the Name Service Switch (NSS) to enable the LDAP support for NIS-based publickeys.

You can save a copy of `/etc/nsswitch.conf` file and modify the original to add LDAP support to the NIS publickey service. See `/etc/nsswitch.ldap` for a sample.

The following shows the sample file, `/etc/nsswitch.ldap`:

```
passwd:       files ldap
group:        files ldap
hosts:        dns files ldap
networks:     files ldap
protocols:    files ldap
rpc:          files ldap
publickey:    ldap [NOTFOUND=return] files
netgroup:     files ldap
automount:    files ldap
aliases:      files
services:     files ldap
```

2.5 Post-installation configuration tasks

This section includes tasks you can perform after performing your guided or customized installation.

2.5.1 Importing name service data into your directory

To import your name service data into your LDAP Directory, consider the following:

- If you have already imported data into your directory with the NIS/LDAP Gateway product, LDAP-UX Client Services can use that data and you can skip to Section 2.4.5 (page 68).
- If you are using NIS, the migration scripts take your NIS maps and generate LDIF files. These scripts can then import the LDIF files into your directory, creating new entries in the directory. This only works if you are starting with an empty directory or creating an entirely new subtree in your directory for your data.

  If you are not using NIS, the migration scripts can take your user, group, and other data from files, generate LDIF, and import the LDIF into your directory.
- If you integrate the name service data into your directory, the migration scripts may be helpful depending on where you put the data in your directory. You could use them just to generate LDIF, edit the LDIF, then import the LDIF into your directory. For example, you could manually add the posixAccount object class to your existing entries under `ou=People` and add their HP-UX information there.
- If you used the guided installation (`autosetup`) to create a new directory server, ensure that the user and group numbers to be imported or migrated do not collide with those created by `autosetup` (see "Ensure user and group numbers do not collide with those created by a guided New Directory Server mode installation" (page 90)).

2.5.1.1 Ensure user and group numbers do not collide with those created by a guided New Directory Server mode installation

The information in this section is a post-installation task for guided installations only (`autosetup`); it does not pertain to customized installations (`setup`).

If you used the guided installation (`autosetup`) and created a new directory server instance, `autosetup` created one new HP-UX account, the Domain Administrator (also known as `domadmin`). It also created three new groups: `DomainAdmins`, `HostAdmins`, and `UserAdmins`. LDAP-UX assigned a UID number to `domadmin` and GID numbers to the three groups. Once you start to migrate user information into this directory server, you need to ensure that the user and group numbers to be migrated do not collide with those created by `autosetup`. If you already know that some user or group numbers will collide with those created by `autosetup`, you can change the UID or GID numbers now by using the `ldapugmod` tool. To determine the UID numbers and GID numbers that were assigned by `autosetup`, use the `ldapuglist` tool, as shown in the following example. Log in as the `domadmin` user on the local host.

```
brewer (): /opt/ldapux/bin/ldapuglist -n domadmin
dn: uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com
cn: Domain Administrator
uid: domadmin
uidNumber: 123
gidNumber: 220
loginShell: /usr/bin/sh
homeDirectory: /home/domadmin
gecos: Domain Administrator

brewer (): /opt/ldapux/bin/ldapuglist -t group -f "cn=*Admins"
dn: cn=UserAdminss,ou=Groups,dc=mydomain,dc=example,dc=com
cn: UserAdmins
cn: UserAdminss
gidNumber: 1910

dn: cn=HostAdmins,ou=Groups,dc=mydomain,dc=example,dc=com
cn: HostAdmins
gidNumber: 1920
memberUid: domadmin
```

```
dn: cn=DomainAdmins,ou=Groups,dc=mydomain,dc=example,dc=com
cn: Domain Administrators
cn: DomainAdmins
gidNumber: 1900
memberUid: domadmin
```

Use the `ldapugmod` tool to change numbers as needed. In the following example, the `ldapugmod` tool changes the GID number of `DomainAdmins` from `1900` to `1999`.

```
brewer (): ldapugmod -P -t group -g 1999 DomainAdmins
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
ntc9-212 (src/tools): ldapuglist -t group -n DomainAdmins
dn: cn=DomainAdmins,ou=Groups,dc=mydomain,dc=example,dc=com
cn: Domain Administrators
cn: DomainAdmins
gidNumber: 1999
memberUid: domadmin
```

For more information about using the `ldapuglist` and `ldapugmod` tools to list and modify users and groups, see Section 5.5 (page 159).

2.5.1.2 Steps to importing name service data into your directory

Here are the steps for importing your user and group data into your LDAP directory. Modify them as needed.

1. Decide which migration method and scripts you will use. Migration scripts are provided to ease the task of importing your existing name service data into your LDAP directory.

   See Section 7.6 (page 326) for a complete description of the scripts, what they do, and how to use them. Modify the migration scripts, if needed.

2. Back up your directory.
3. Run the migration scripts, using the worksheet in "Configuration worksheet" (page 347).
4. If the method you used above did not already do so, import the LDIF file into your directory.

2.5.2 Verifying the LDAP-UX Client Services

This section describes some simple ways you can verify the installation and configuration of your LDAP-UX Client Services. You may need to do more elaborate and detailed testing, especially if you have a large environment.

If any of the following tests fail, see Section 5.18 (page 189).

1. To test the name service, use the `nsquery`[1] command:

   `nsquery lookup_type lookup_query [lookup_policy]`

   For example, to test the name service switch to resolve a user name lookup, enter:

   `nsquery passwd username ldap`

   where ***`username`*** is the login name of a valid user whose posix account information is in the directory. You should see output something like the following depending on how you have configured `/etc/nsswitch.conf`:

   ```
   Using "ldap" for the passwd policy.
   Searching ldap for jbloggs
   User name: jbloggs
   user Id: 10000
   Group Id: 2000
   Gecos:
   Home Directory: /home/jbloggs
   Shell: /bin/sh
   Switch configuration: Terminates Search
   ```

   This tests the Name Service Switch configuration in `/etc/nsswitch.conf`. If you do not see output like that above, check `/etc/nsswitch.conf` for proper configuration.

2. Use other commands to display information about users in the directory, making sure the output is as expected:

   `pwget -n username`
   `nsquery hosts host_to_find`
   `grget -n groupname`
   `ls -l`

NOTE: While you can use the following commands to verify your configuration, these commands enumerate the entire passwd or group database, which may reduce network and directory server performance for large databases:

`pwget``(with no options)`

`grget``(with no options)`

`listusers`

`logins`

3. Use one of the following expressions of the `ldapcfinfo` command to verify that a particular service is properly configured:

   ```
   ldapcfinfo -t passwd
   ldapcfinfo -t group
   ldapcfinfo -t pam
   ```

   When any of these commands return an error, the command reports what is improperly configured.

4. Use the `beq` search utility to search for the following services: `pwd` (password), `grp` (group), `shd` (shadow password), `srv` (service), `prt` (protocol), `rpc` (RPC), `hst` (host), `net` (network), `ngp` (netgroup), and `grm` (group membership). An example `beq` command using `name` as the search key, `pwd` as the service, and `ldap` as the library in 32-bit mode on an HP-UX 11i v2 or v3 a PA-RISC machine is shown below.

1. `nsquery` is a contributed tool included with the ONC/NFS product. For more information, see the *nsquery*(1) manpage.

```
./beq -k n -s pwd -l /usr/lib/libnss_ldap.1 iuser1
nss_status........ NSS_SUCCESS
pw_name..........(iuser1)
pw_passwd.........(*)
pw_uid............(101)
pw_gid............(21)
pw_age............()
pw_comment........()
pw_gecos..........(gecos data in files)
pw_dir............(/home/iuser1)
pw_shell..........(/usr/bin/sh)
pw_audid..........(0)
pw_audflg.........(0)
```

Use the following `beq` command if you are running 64-bit applications on an HP-UX 11i v2 or v3 HP Integrity server:

```
./beq -k n -s pwd -l /usr/lib/hpux64/libnss_ldap.so.1 iuser1
```

Use the following `beq` command if you are running 32-bit applications on an HP-UX 11i v2 or v3 HP Integrity server:

```
./beq -k n -s pwd -l /usr/lib/hpux32/libnss_ldap.so.1 iuser1
```

For command syntax and examples, see Section 7.7.1 (page 330) .

5. Log in to the client system from another system using rlogin or telnet. Log in as a user in the directory and as a user in /etc/passwd to make sure both work.
6. Optionally, test your PAM_AUTHZ authorization configuration:

   If the PAM_AUTHZ is configured without the `pam_authz.policy` file, verify the following:

   - Log into the client system from another system using `rlogin` or `telnet`. From there log in to the directory as a member from +@netgroup to verify that PAM_AUTHZ authorizes you and is working correctly.
   - Log in as a user to the directory as a member of a `-@netgroup` to be sure that the system will not authorize you to log in.

   If the PAM_AUTHZ module is configured with the `pam_authz.policy` file, verify the following:

   - Log in the client system with a user name that is covered by an `allow` access rule in the policy file. Make sure the user will be allowed to log in.
   - Log in as a user that is covered by `adeny` access rule in the policy file. Make sure the user can not log in to the client system.
7. Open a new `hpterm` window and log in to the client system as a user whose account information is in the directory. (For more information about the `hpterm` command, see the *hpterm*(1X) manpage.) It is important you open a new `hpterm` window or log in from another system, because if login doesn't work, you could be locked out of the system and would have to reboot to single-user mode. Logging in to the client system in this way tests the PAM configuration in `/etc/pam.conf`. If you cannot log in, check that `/etc/pam.conf` is configured properly. In addition, check your directory to make sure the user's account information is accessible by the proxy user or anonymously, as appropriate. Check your profile to make sure it is correct. For troubleshooting information, see Section 5.18 (page 189).
8. To examine files belonging to a user whose account information is in the directory, use the `ls` or `ll` command. Make sure the owner and group of each file are accurate:

   ```
   ll /tmp
   ls -l
   ```

   If any owner or group shows up as a number instead of a user or group name, the name service switch is not functioning properly. Check the file /etc/nsswitch.conf, your directory, and your profile.

If you want to verify that you set up X.500 group membership correctly, follow these steps:

1. Create a valid posix user and group. Add this user as a member of this group using the attribute "member" instead of "memberuid". Here is an example ldif file specifying `xuser2` as a member of the group `xgrpup1`:

```
#cat example_ids.ldif
dn: cn=xgroup1,ou=Groups,o=hp.com]
objectClass: posixGroup
objectClass: groupofnames
objectClass: top
cn: xgroup1
userPassword: {crypt}*
gidNumber: 999
member: uid=xuser2,ou=People,o=hp.com
dn: uid=xuser2,ou=People,o=hp.com
uid: xuser2
cn: xuser2
objectClass: top
objectClass: account
objectClass: posixAccount
userPassword: {crypt}xxxxxxxxxxxx
loginShell: /bin/ksh
uidNumber: 9998
gidNumber: 999
homeDirectory: /home/xuser2
```

2. Make sure that the file `/etc/nsswitch.conf` specifies `ldap` for group service:

```
cat /etc/nsswitch.conf
:
:
group: files ldap
:
:
```

3. Verify:

```
# grget -n xgroup1
xgroup1:*:999: xuser2
```

If xuser2 shows up as a member of xgroup1, then your setup is correct.

2.5.3 Enabling AutoFS support

AutoFS is a client-side service that automatically mounts appropriate file systems when users request access to them. If an automounted file system has been idle for a period of time, AutoFS unmounts it. AutoFS uses name services such as files, NIS, or NIS+ to store and manage AutoFS maps.

LDAP-UX Client Services supports the automount service under the AutoFS subsystem. This feature allows users to store AutoFS maps in an LDAP directory server.

2.5.3.1 Automount schemas

This section describes the following automount schemas:

- new automount schema

  An automount schema is based on RFC 2307-bis. This schema defines new `automountMap` and `automount` structures to represent the AutoFS maps and their entries in the LDAP directory.

- nisObject automount schema

  The `nisObject` automount schema defines`nisMap` and `nisObject` structures to represent the AutoFS maps and their entries in the LDAP directory. There are some limitations that you need to be aware of when using the `nisObject` automount schema.

The LDAP-UX Client Services supports the new automount schema. The `nisObject` automount schema can also be used if configured via attribute mappings. The `autosetup` program installs the correct schema. No adjustments are necessary.

Read subsequent subsections of this chapter for the detailed information about the automount schemas.

2.5.3.1.1 New automount schema

This schema is a new schema defined in RFC 2307-bis. This schema defines new `automountMap` and `automount` structures to represent AutoFS maps and their entries in the LDAP directory. AutoFS maps are stored in the LDAP directory server using structures defined by this schema.

The RFC 2307-bis automount schema is included with HP-UX Directory Server by default. If you are installing LDAP-UX with a different directory server, the `setup` program will import the new automount schema into your directory server. If you previously configured LDAP-UX B.03.30 or an earlier version, and are now updating the product to version B.04.00 or later, you must re-run the `setup` program to import the new automount schema into the LDAP directory. The subsequent client systems do not need to re-run the `setup` program.

2.5.3.1.1.1 Schema

The following shows the RFC 2307-bis automount schema in the LDIF format:

```
objectClasses: ( 1.3.6.1.1.1.2.16
NAME 'automountMap'
DESC 'Automount Map information'
SUP top STRUCTURAL
MUST automountMapName
MAY description
X-ORIGIN 'user defined' )

objectClasses: ( 1.3.6.1.1.1.2.17
NAME 'automount'
DESC 'Automount information'
SUP top STRUCTURAL
MUST ( automountKey $ automountInformation )
MAY description
X-ORIGIN 'user defined' )
```

```
attributeTypes: ( 1.3.6.1.1.1.1.31
NAME 'automountMapName'
DESC 'automount Map Name'
EQUALITY caseExactIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE
X-ORIGIN 'user defined' )

attributeTypes: ( 1.3.6.1.1.1.1.32
NAME 'automountKey'
DESC 'Automount Key value'
EQUALITY caseExactIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE
X-ORIGIN 'user defined' )

attributeTypes: ( 1.3.6.1.1.1.1.33
NAME 'automountInformation'
DESC 'Automount information'
EQUALITY caseExactIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE
X-ORIGIN 'user defined' )
```

For the HP-UX Directory Server, each entry starting with the text "`attributetypes`:" or "`objectclasses`:" must be one continuous line.

2.5.3.1.1.2 An example

The following shows an example of a direct AutoFS map, `auto_direct`, stored in the LDAP directory server using new automount schema:

```
dn:automountMapName=auto_direct,dc=nishpind
objectClass: top
objectClass: automountMap
automountMapName: auto_direct


dn:automountKey=/mnt_direct/test1,\
automountMapname=auto_direct, dc=nishpind
objectClass: top
objectClass: automount
automountInformation:hostA:/tmp
automountKey: /mnt_direct/test1

dn:automountKey=/mnt_direct/test2,\
automountMapname=auto_direct, dc=nishpind
objectClass: top
objectClass: automount
automountInformation:hostB:/tmp
automountKey:/mnt_direct/test2
```

2.5.3.1.2 The nisObject automount schema

The `nisObject` automount schema defines`nisMap` and `nisObject` structures to represent the AutoFS maps and their entries. The AutoFS maps are stored in the LDAP directory server using the `nisMap`and `nisObject` structures.

2.5.3.1.2.1 An example

The following shows an example of a direct AutoFS map, `auto_direct`, stored in the LDAP directory server using the `nisObject` automount schema:

```
dn:nisMapName=auto_direct,dc=nishpind
objectClass: top
objectClass: nisMap
nisMapName: auto_directdn:cn=/mnt_direct/test1,
nisMapName=auto_direct, dc=nishpind
objectClass: top
objectClass: nisObject
```

```
nisMapName: auto_direct
cn: /mnt_direct/test1
nisMapEntry:hostA:/tmp

dn:cn=/mnt_direct/test2, nisMapname=auto_direct, dc=nishpind
objectClass: top
objectClass: nisObject
nisMapName: auto_direct
cn: /mnt_direct/test2
nisMapEntry:hostB:/tmp
```

2.5.3.1.2.2 Limitations

The `nisObject` automount schema contains three attributes, `cn`, `nisMapEntry` and `nisMapName`. `cn` is an attribute that ignores case-matching. Consider the following example:

```
# an indirect map named auto_test
test1    server1:/source
TEST1    server2:/source
```

In the above example, because the `cn` attribute is case-insensitive, the LDAP considers `"cn=TEST1, nisMapName=auto_test"` to be a redefinition of `"cn=test1, nisMapName=auto_test"`.

Using the `nisObject` automount map schema, capital letters are not significant. In other words, if two keys have names that are only different by the use of capital letters, then one of those entries will be rendered inoperable because the other one is the only one that can be retrieved.

NOTE: If you use the `nisObject` automount map schema, do not use any keys that have capital letters and only differ from other keys by those capital letters.

2.5.3.2 Attribute mappings

LDAP-UX Client Services supports attribute mappings between the new RFC 2307-bis automount schema and the `nisObject` automount schema. This feature allows the directory administrators to use the `nisObject` schema if they have already deployed it.

When both new automount schema and `nisObject` schema exist in the LDAP directory server, if you choose to use the `nisObject` automount schema, you must run the `setup` program using the custom configuration to perform the attribute mappings and search filter changes for the automount service. The attribute mappings include the following:

- Remap the new automount attributes to the`nisObject` automount attributes. The attribute mappings are done in step 10 of the Custom Configuration. For detailed information on how to remap the automount attributes, see Section 2.4.5.2 (page 73).

  Table 2-7 (page 97) shows the attribute mappings:

  Table 2-7 Attribute mappings

New Automount Attribute	nisObject Automount Attribute
`automountMapname`	`nisMapname`
`automountKey`	`cn`
`automountInformation`	`nisMapEntry`

- Change the `automount` search filter for the automount service to the `nisObjectsearch` filter. LDAP-UX Client Services uses the `automount` search filter for the automount service as a default. The search filter change can be done in step 11 of the Custom Configuration. If you want to create the `nisObject` search filter for the automount service to search a different location in the LDAP directory server, see Section 2.4.5.2 (page 73) for details.

If you want to perform attribute mappings or search filter changes by using the Custom Configuration, ensure that you do not accept the remaining default configuration parameters in step 4 of the Custom Configuration.

NOTE: You can use the `nisObject` automount schema without attribute mappings and search filter changes if only the `nisObject` automount schema exists in the LDAP directory.

2.5.3.3 Configuring NSS

Configure the Name Service Switch (NSS) to enable the LDAP support for AutoFS.

You can save a copy of `/etc/nsswitch.conf` file and modify the original to add LDAP support to the automount service. See `/etc/nsswitch.ldap` for a sample.

The following shows the sample file, `/etc/nsswitch.ldap`:

```
passwd:       files ldap
group:        files ldap
hosts:        dns files ldap
networks:     files ldap
protocols:    files ldap
rpc:          files ldap
publickey:    ldap [NOTFOUND=return] files
netgroup:     files ldap
automount:    files ldap
aliases:      files
services:     files ldap
```

2.5.3.4 AutoFS migration scripts

This section describes the migration scripts which can be used to migrate your AutoFS maps from files, NIS servers or NIS+ servers to LDIF files. After LDIF files are created, you can use the `ldapmodify` tool to import LDIF files to your LDAP directory server. These migration scripts use the new automount schema defined in RFC 2307-bis to migrate the AutoFS maps to LDIF. You need to import the new automount schema into your LDAP directory server before you use these migration scripts to migrate AutoFS maps.

Table 2-8 (page 98) describes the migration scripts:

Table 2-8 Migration scripts

Migration Script	Description
`migrate_automount.pl`	Migrates AutoFS maps from files to LDIF.
`migrate_nis_automount.pl`	Migrates AutoFS maps from the NIS server to LDIF.
`migrate_nisp_autofs.pl`	Migrates AutoFS maps from NIS+ server to the `nisp_automap.ldif` file.

2.5.3.4.1 Environment variables

When you use the AutoFS migration scripts to migrate AutoFS maps, set the following environment variables:

`LDAP_BASEDN` The base distinguished name of the LDAP directory that the AutoFS maps are to be placed in.

`DOM_ENV` This only applies to the `migrate_nisp_autofs.pl` script. This variable defines the fully qualified name of the NIS+ domain where you want to migrate your data from.

`NIS_DOMAINNAME` This only applies to the `migrate_nis_automount.pl` script. This variable specifies the fully qualified name of the NIS domain where you want to migrate your data from. This variable is optional. If the NIS

domain name is not specified, LDAP-UX uses the value of the `NIS_DOMAIN` parameter configured in the `/etc/rc.conf.d/namesvrs` file.

Examples:

The following command sets the fully qualified name of the NIS+ domain to "`cup.hp.com`":

```
export DOM_ENV="cup.hp.com"
```

The following command sets the fully qualified name of the NIS domain to "`india.hp.com`":

```
export NIS_DOMAINNAME="india.hp.com"
```

The following command sets the base DN to "`dc=cup, dc=hp, dc=com`":

```
export LDAP_BASEDN="dc=cup, dc=hp, dc=com"
```

2.5.3.4.2 General syntax for migration scripts

The migration scripts use the following general syntax:

scriptnameinputfileoutfile

where

scriptname Is the name of the particular script you are using.

inputfile Is the fully qualified file name of the appropriate AutoFS map that you want to migrate. For example, `/etc/auto_master`.

outputfile This only applies to the `migrate_nis_automount.pl` and `migrate_automount.pl` scripts. This is optional and is the name of the file where the LDIF is written. stdout is the default output.

2.5.3.4.3 The migrate_automount.pl script

This script, found in `/opt/ldapux/migrate`, migrates the AutoFS maps from files to LDIF.

2.5.3.4.3.1 Syntax

scriptnameinputfileoutputfile

2.5.3.4.3.2 Examples

The following commands migrate the AutoFS map `/etc/auto_direct` to LDIF and place the results in the `/tmp/auto_direct.ldif` file:

```
export LDAP_BASEDN="dc=nishpind"
migrate_automount.pl /etc/auto_direct /tmp/auto_direct.ldif
```

The following shows the `/etc/auto_direct` file:

```
#local mount point          remote server:directory
/mnt/direct/lab1                   hostA:/tmp
/mnt/direct/lab2                   hostB:/tmp
```

The following shows the `/tmp/auto_direct.ldif` file:

```
dn:automountMapName=auto_direct,dc=nishpind
objectClass: top
objectClass: automountMap
automountMapName: auto_direct

dn:automountKey=/mnt_direct/lab1,\ automountMapname=auto_direct, dc=nishpind
objectClass: top
objectClass: automount
automountInformation:hostA:/tmp
automountKey: /mnt_direct/lab1

dn:automountKey=/mnt_direct/lab2,\
automountMapname=auto_direct, dc=nishpind
objectClass: top
```

```
objectClass: automount
automountInformation:hostB:/tmp
automountKey:/mnt_direct/lab2
```

You can use the `/opt/ldapux/bin/ldapmodify` tool to import the LDIF file `/tmp/auto_direct.ldif` that you just created above into the LDAP directory. For example, the following command imports the `/tmp/auto_direct.ldif` file to the LDAP base DN "`dc=nishpind`" in the LDAP directory server `LDAPSERV1`:

```
/opt/ldapux/bin/ldapmodify -a -h LDAPSERV1 -D "cn=Directory Manager" -w <passwd> -f /tmp/auto_direct.ldif
```

Where options are:

-a Add a new entry into the LDAP directory

-h The LDAP directory host name

-D The Distinguish Name (DN) of the directory manager

-w The password of the directory manager

-f The LDIF file to be imported into the LDAP directory

2.5.3.4.4 The migrate_nis_automount.pl script

This script, found in `/opt/ldapux/migrate`, migrates the AutoFS maps from the NIS server to LDIF.

2.5.3.4.4.1 Syntax

scriptnameinputfileoutputfile

2.5.3.4.4.2 Examples

The following commands migrate the AutoFS map `/etc/auto_indirect` to LDIF and place the results in the `/tmp/auto_indirect.ldif` file:

```
export LDAP_BASEDN="dc=nisserv1"
export NIS_DOMAINNAME="cup.hp.com"
migrate_nis_automount.pl /etc/auto_indirect  /tmp/auto_indirect.ldif
```

The following shows the `/etc/auto_indirect` file:

```
#local mount point          remote server:directory
lab1                            hostA:/tmp
lab2                            hostB:/tmp
```

The following shows the `/tmp/auto_indirect.ldif` file:

```
dn:automountMapName=auto_indirect,dc=nisserv1
objectClass: top
objectClass: automountMap
automountMapName: auto_indirect

dn:automountKey=lab1,\
automountMapname=auto_indirect, dc=nisserv1
objectClass: top
objectClass: automount
automountInformation:hostA:/tmp
automountKey: lab1

dn:automountKey=lab2, \
automountMapname=auto_indirect, dc=nisserv1
objectClass: top
objectClass: automount
automountInformation:hostB:/tmp
automountKey:lab2
```

You can use the `/opt/ldapux/bin/ldapmodify` tool to import the LDIF file `/tmp/auto_indirect.ldif` that you just created above into the LDAP directory. For example, the following command imports the `/tmp/auto_indirect.ldif` file to the LDAP base DN "`dc=nisserv1`" in the LDAP directory server `LDAPSERV1`:

```
/opt/ldapux/bin/ldapmodify -a -h LDAPSERV1 -D "cn=Directory Manager"
-w <passwd> -f /tmp/auto_indirect.ldif
```

2.5.3.4.5 The migrate_nisp_autofs.pl script

This script, found in `/opt/ldapux/migrate/nisplusmigration`, migrates the AutoFS maps from the NIS+ server to the `nisp_automap.ldif` file.

2.5.3.4.5.1 Syntax

scriptnameinputfile

2.5.3.4.5.2 Examples

The following commands migrate the AutoFS map `/etc/auto_indirect` to LDIF and place the results in the `nisp_automap.ldif` file:

```
export LDAP_BASEDN="dc=nishpbnd"
export DOM_ENV ="cup.hp.com"
migrate_nisp_autofs.pl /etc/auto_indirect
```

The following shows the `/etc/auto_indirect` file:

```
#local mount point          remote server:directory
lab1                            hostA:/tmp
lab2                            hostB:/tmp
```

The following shows the `nisp_automap.ldif` file:

```
dn:automountMapName=auto_indirect,dc=nishpbnd
objectClass: top
objectClass: automountMap
automountMapName: auto_indirect

dn:automountKey=lab1, \
automountMapname=auto_indirect, dc=nishpbnd
objectClass: top
objectClass: automount
automountInformation:hostA:/tmp
automountKey: lab1

dn:automountKey=lab2, \
automountMapname=auto_indirect, dc=nishpbnd
objectClass: top
objectClass: automount
automountInformation:hostB:/tmp
automountKey:lab2
```

You can use the `/opt/ldapux/bin/ldapmodify` tool to import the LDIF file `nisp_automap.ldif` that you just created above into the LDAP directory. For example, the following command imports the `nisp_automap.ldif` file to the LDAP base DN "`dc=nishpbnd`" in the LDAP directory server `LDAPSERV1`:

```
/opt/ldapux/bin/ldapmodify -a -h LDAPSERV1 -D "cn=Directory Manager"
 -w <passwd> -f nisp_automap.ldif
```

2.5.4 Enabling offline credential caching for authentication when the directory server is unavailable

If contact with the directory server is lost because of a network problem or server crash, LDAP users cannot log in to the system. This may have a negative impact on the OS and its applications, especially for mission-critical applications. To enable the OS to continue to properly function when connection with the directory server is lost, LDAP-UX Client Services 5.0 (or later) provides an offline credential cache that allows LDAP to continue authenticating users even when contact with all directory servers is lost. You can enable this feature by configuring several parameters available in the LDAP-UX client daemon configuration file, `/etc/opt/ldapux/ldapclientd.conf` file.

NOTE: For information about patches that need to be installed to support offline credential caching, see the *LDAP-UX Integration B.05.00 Release Notes*.

2.5.4.1 How the offline cache works

To support this feature, you can configure LDAP-UX to maintain a secondary (offline) long-term cache that stores previously-discovered user account and group information, including authentication passwords that are hashed using the salted Secure Hash Algorithm (SHA-512). If the directory server becomes unavailable, LDAP-UX resorts to this cache for the information needed to authenticate users. When the directory server becomes available again, LDAP-UX resumes referring to the directory server for authentication information.

While LDAP-UX is in contact with the directory server, if long-term credential caching is enabled, LDAP-UX captures user account and password information during a user's login attempt, and if the login is successful, stores this information in the offline cache. LDAP-UX updates the cache as necessary with new or changed account information as it becomes available during later authentication attempts. It also updates passwords that are successfully changed by users on the local host.

When the directory server is unreachable, LDAP-UX does not allow users to change their passwords (because the password cannot be updated in the directory server).

The offline cache maintains information only for users who have recently logged in to the system while the directory server was available.

The offline credential cache will survive after a reboot. However, data stored in the cache has a configurable expiration date (two weeks, by default) to help ensure that stale user accounts are removed. Because the long-term credential cache expires after a defined period, any user that has not recently used the system (within the expiration period defined by the LDAP-UX administrator) will not be allowed to authenticate, since that user's cached credential may not exist or may have been removed after it expired.

LDAP-UX allows you to enable long-term enumeration, in which case LDAP-UX periodically retrieves and updates all user and group entries in the local on-disk storage for later reference when the directory server is not reachable. You can specify how frequently LDAP-UX should refresh the enumeration data in the cache.

NOTE: Enumeration requests involving large databases could reduce network and server performance. Use this feature only if it is expedient for your environment.

LDAP-UX also allows you to specify:

- How frequently long-term data should be saved to the offline cache
- How much memory to allocate for the offline cache

2.5.4.2 Configuring the offline cache

The following shows the section in `/etc/opt/ldapux/ldapclientd.conf` that includes the offline credential cache variables that you can configure.

```
[longterm_cache]
#enable=no
#
# How long before data is considered stale and not usable.  1,209,600 = 2 weeks
#longterm_expired_interval=1209600
#
# How frequently should save long term data to permanent storage. 900 = 15 min.
#longterm_cache_backup_interval=900
#
# How much memory to allocate for the long term cache, which stores user and
# group information.  This cache is only used by the working set of users and
# groups.  The working set means any user or group being used or displayed on
# the system. If you have numerous large groups with numerous members, this
```

```
# value should be at least twice as large as the combined size of all those
# groups.
#longterm_cache_size=50000000
#
# Should long term caching support enumeration of users and groups.  If
# getpwent() and getgrent() are not required, this can be disabled.
#longterm_enum_enable=no
#
# How frequently should the HP-UX client go to the directory server to refresh
# the enumeration cache. 84600 = once per day.
#longterm_enum_search_interval=86400
```

As shown, offline credential caching is disabled by default. To enable offline credential caching, uncomment the first line of the section (remove the pound sign (#)) and specify `yes` instead of `no` as shown:

```
[longterm_cache]
enable=yes
#
```

Configure the other parameters as noted in the comments included with the configuration file. To keep the default settings, you can leave the lines as they are (without removing the pound signs that precede each line that defines a parameter).

NOTE: Offline credential caching is not supported for Windows ADS users.

Offline credential caching and integrated compat mode cannot be used together. Compat mode is discussed in Section 2.5.5 (page 104).

2.5.5 Enabling integrated Compat Mode to control name services and user logins

LDAP-UX version 5.0 and higher makes available traditional NIS-style Compat (Compatibility) Mode to control the name services that are used to obtain user and group information.

2.5.5.1 Overview

A legacy feature of NIS (the Network Information Service) is the ability to allow local control of network-defined passwd entries. Administrators of NIS clients can select which accounts would be available on the local host by specifying lists of netgroups in the host's `/etc/passwd` file. For additional details, see Appendix C of the *Network Information Service (NIS) Administrator's Guide* . The following example shows how an administrator might limit logins on the local host to members of the `operator` and `webadmin` groups. Within the `/etc/passwd file`, the following entries would be added:

```
...
+@operator::::::
+@webadmin::::::
...
```

While this feature was typically used to control which groups of users could log in to a particular host, it also could be used to obscure or override fields of a user's `passwd` entry. For example, an administrator could force a particular group of users to use a specific login shell by inserting the desired path to the desired login shell in the 7th field of the entry (the login shell is positionally defined as the 7th field in each entry in the `/etc/passwd` file.):

```
...
+@icsuteam::::::/usr/local/bin/supportapp
...
+:x
```

In the previous example, any user that is a member of the `icsuteam` will be forced to run the `supportapp` upon login to the system, regardless of how their personal login shell is defined in the NIS passwd map. The `+:x` as the last line of the `/etc/passwd` file indicates that all remaining accounts managed in the NIS passwd map will be visible on the system, but their passwords will be masked with an `x`, which traditionally would prevent login.

2.5.5.2 Netgroups in LDAP

With LDAP, the ability to use netgroups to control which groups of users are visible on a host, or which fields are masked, is still available. System administrators can enable NIS compat mode by defining the following sequence in the `/etc/nsswitch.conf` file:

```
...
passwd: compat
passwd_compat: files ldap
...
```

The first line indicates that the `passwd` name service should operate in the traditional "compatibility mode," allowing netgroups to be specified in the `/etc/passwd` file. The second line indicates that the files and LDAP repositories should be used as the name service repository for finding the user accounts referenced by those netgroups.

However, use of compat mode with an LDAP repository can greatly impact performance of the name service system. When compat mode is used to mask passwd entries, numerous requests to the directory server must be generated to examine the netgroups to find their members and then search for each individual member. While `ldapclientd` can cache netgroup and passwd entries, the name service subsystem does the actual processing to generate the proper masked results. In this case, while caching does improve performance, it places an extreme load on the CPU from the `ldapclientd` caching daemon, as it resolves the numerous requests from the name service subsystem.

Most deployments use compat mode just to control which users are allowed to log in to the host. In this case, the `libpam_authz` library can be used to control which users can log in to the host, based on the netgroups listed in the `/etc/passwd` file. (For more information about using PAM_AUTHZ login authorization and `libpam_authz` , see Section 5.3 (page 140).) Compat mode can therefore be disabled. However, for deployments that rely on the field-masking feature of compat mode, no alternative was available. In these situations, if a large organization used numerous netgroups with many users, CPU usage of `ldapclientd` could reach maximum limits.

As a means to greatly mitigate the performance impacts of compat-mode field masking, LDAP-UX has integrated compat mode support directly into `ldapclientd`, allowing caching of compat-mode user entries.

2.5.5.3 Configuring integrated "compat" mode

To enable integrated compat mode, you must perform four configuration steps:

1. Disable compat mode in the name service switch. In the `/etc/nsswitch.conf` file, replace the following:

   ```
   ...
   passwd: compat
   passwd_compat: files ldap
   ...
   ```

   with:

   ```
   ...
   passwd: files ldap
   ...
   ```

2. Configure `internal-compat-mode` processing inside `ldapclientd`:

   a. `/etc/opt/ldapux/ldapclientd.conf`

      Search for "`flush_compat_info_time`". This indicates how often `ldapclientd` will refresh its cached copy of the netgroup structures defined in the `/etc/passwd` file. If you make changes to the netgroup list in the `/etc/passwd` file, these changes will not appear until this time period has passed or you have restarted or flushed the caches of `ldapclientd (-F)`. Adjust this value as needed, or leave as the default (1 day).

b. `/etc/opt/ldapux/ldapux_client.conf`

Search for "`enable_compat_mode`". To enable internal compat-mode processing in `ldapclientd`, set this value to `1`.

NOTE: If LDAP-UX has been configured previously on your host, you will need to examine the newly delivered configuration files found under `/opt/ldapux/newconfig/etc/opt/ldapux`. Compare and merge the existing configuration files with those delivered in the `newconfig` subdirectory.

3. Restart `ldapclientd`. Use the following commands:

```
# /opt/ldapux/bin/ldapclientd -k
# /opt/ldapux/bin/ldapclientd
```

4. If you change the netgroup list in the `/etc/passwd` or `/etc/group`, and want to force `ldapclientd` to reflect the updated configuration, force `ldapclientd` to rebuild its cache with the following command:

```
# /opt/ldapux/bin/ldapclientd -f
```

2.5.5.3.1 Limitations

When processing netgroup information for compat mode (that is, `+@<netgroup>`, `-@<netgroup>` in `/etc/passwd`), internal compat-mode processing in `ldapclientd` always searches the LDAP directory first for definition of the netgroup entries and then the `local /etc/netgroup` file. As a result, if the same network group with different group members is configured in both `/etc/netgroup` and the LDAP directory, the members defined in the netgroup stored in the LDAP directory will be used instead of the entries from the local `/etc/netgroup` file.

HP recommends that you do not configure netgroups with the same name in both the `/etc/netgroup` file and the LDAP directory.

Also, long-term offline credential caching and integrated compat mode cannot be used together. Long-term offline credential caching is discussed in Section 2.5.4 (page 102).

2.5.6 Controlling user access to the system through LDAP

By default, all users stored in the LDAP directory are allowed to log in to the local HP-UX client system. LDAP-UX provides several ways to increase the security level to prevent unwanted users from logging in to the local system through LDAP, including the following:

- Using the PAM_AUTHZ service module to control login access, as described elsewhere, inSection 5.3 (page 140)
- Disabling logins to the local system from specified LDAP users by configuring the `disable_uid_range` flag in the local client's start-up file (`/etc/opt/ldapux/ldapux_client.conf`), as described in Section 2.5.6.1 (page 106)
- Preventing unwarranted access to the local system by users defined in the LDAP directory server that have equivalent user names or user identification numbers (UIDs) in the local system `/etc/passwd` file, as described in Section 2.5.6.2 (page 107)
- Using the ignore option to enable specified users to be ignored by PAM_LDAP authentication, as described in "Configuring PAM_LDAP authentication to ignore specific users" (page 109).

2.5.6.1 Using the disable_uid_range flag to prevent access to the local system by unwanted users

To disallow specific users to log in to a local system, you can set the `disable_uid_range` flag in the local client's start-up file`/etc/opt/ldapux/ldapux_client.conf`. The flag is in the [NSS] section of the file. (HP recommends that you do not edit the [profile] section of the file.) The following example shows the portion of the file containing the flag:

```
#
# You can disable specific users so that they are unable to log in
# through the LDAP server by uncommenting the "disable_uid_range"
# flag and adding the UID numbers you want to disable. For example:
#
#   disable_uid_range=0-100,120,300-400
#
# Note:  The list of UID numbers must be on one line and the maximum
# number of ranges is 20. The system will ignore the typos and white spaces.
#
#disable_uid_range=0
```

To enable and configure the flag, first save a copy of the `/etc/opt/ldapux/ldapux_client.conf` file and edit the original. Then uncomment the flag (remove the #) and enter the UID range(s). For example, the flag might look like this:

```
disable_uid_range=0-100, 300-450, 89
```

Another common example would be to disable root access, in which case the flag would look like this: `disable_uid_range=0`.

NOTE:

- White spaces between numbers are ignored.
- Only one line of the list is accepted; however, the line can be wrapped.
- The maximum number of ranges is 20.

When the `disable_uid_range` is turned on, the disabled UIDs will not be displayed when you run commands such as `pwget`, `listusers`, and `logins`.

NOTE: The `passwd` command may still allow you to change a password for a disabled user when alternative authentication methods, such as PAM Kerberos, are used since LDAP does not control these subsystems.

2.5.6.2 Using the deny_local option to prevent access to the local system by unwanted users

LDAP-UX version 4.2 and later provides a simple and effective way to disable system access for local user accounts that are also defined in the LDAP directory server. Without this level of security protection, an LDAP-UX user with the same user name or account number (UID) as a user defined in the local system's `/etc/passwd` file, could illegitimately gain access to the local system. For example, if the root user is defined in the local system's `/etc/passwd` file, an LDAP-UX directory server administrator could create a user named “root” and then log in to the local system based on the password associated with user “root” on the directory server.

To disable system access for local user accounts that are also defined in the LDAP directory server, configure the `deny_local` option in the PAM configuration file `/etc/pam.conf`, entering a line for each service, in the following format:

***service module_type* required libpam_ldap.so.1 deny_local**

where:

service	Specifies the service used for accessing the system
module_type	Specifies the service module type: authentication (auth), account management (account), session management (session), or password management (password). Typically, the `deny_local` option is specified for both authentication and account management, and for all PAM-enabled services.
required	Specifies the control flag as required (mandatory).
libpam_ldap.so.1	Specifies the pathname to the PAM_LDAP library object that implements the service functionality. If the pathname is not absolute, it is assumed to be relative to `/usr/lib/security/$ISA/`.

deny_local Specifies the deny_local option

The following example shows the portion of the /etc/pam.conf file that configures the authentication and account services. As a result, for any attempt to use these services to log in or establish a session on the HP-UX client system, if PAM_LDAP detects an equivalent account name or UID in the /etc/passwd file, it returns PAM_IGNORE (PAM_LDAP does not authenticate the user). If an equivalent account name or UID is not found in the /etc/passwd file, PAM_LDAP returns the appropriate authentication status (which could be, for example, notification that the credential is invalid, the password needs to be updated, or that the authentication succeeded; the status reported depends on the circumstances when the user tries to authenticate).

```
#
# PAM configuration
#
# This pam.conf file is intended as an example only.
#
# Please note that this configuration file has only been modified for the
# default services. Other services can be added or modified as
# needed or desired. If a service is not listed, it will use the
# OTHER classification
#
# the format for a entry is
# <service>  <module_type>  <control>  <module path>  <options>
#
#Notes:
#
# If the path to a library is not absolute, it is assumed to be relative
# to the directory /usr/lib/security/$ISA/
#
# The "$ISA" (i.e Instruction Set Architecture) token is replaced by the
# PAM engine (libpam) with "hpux64" for IA 64-bit modules, or with "hpux32"
# for IA 32-bit modules, or with "pa20_64" for PA 64-bit modules, or with
# NULL for PA 32-bit modules.
#
# For PA applications, library name ending with "so.1" is a symbolic link
# that points to the corresponding PA (32 or 64-bit) backend library.
#
# see pam.conf(4) for more details
#
# Authentication management
#
login    auth required          libpam_hpsec.so.1
login    auth sufficient        libpam_unix.so.1
login    auth required          libpam_ldap.so.1 try_first_pass deny_local
su       auth required          libpam_hpsec.so.1 bypass_setaud
su       auth sufficient        libpam_unix.so.1
su       auth required          libpam_ldap.so.1 try_first_pass deny_local
dtlogin  auth required          libpam_hpsec.so.1
dtlogin  auth sufficient        libpam_unix.so.1
dtlogin  auth required          libpam_ldap.so.1 try_first_pass deny_local
dtaction auth required          libpam_hpsec.so.1
dtaction auth sufficient        libpam_unix.so.1
dtaction auth required          libpam_ldap.so.1 try_first_pass deny_local
ftp      auth required          libpam_hpsec.so.1
ftp      auth sufficient        libpam_unix.so.1
ftp      auth required          libpam_ldap.so.1 try_first_pass deny_local
rcomds   auth required          libpam_hpsec.so.1
rcomds   auth sufficient        libpam_unix.so.1
rcomds   auth required          libpam_ldap.so.1 try_first_pass deny_local
sshd     auth required          libpam_hpsec.so.1
sshd     auth sufficient        libpam_unix.so.1
sshd     auth required          libpam_ldap.so.1 try_first_pass deny_local
```

```
OTHER    auth required          libpam_hpsec.so.1
OTHER    auth sufficient        libpam_unix.so.1
OTHER    auth required          libpam_ldap.so.1 try_first_pass deny_local
#
# Account management
#
login    account required       libpam_hpsec.so.1
login    account sufficient     libpam_unix.so.1
login    account required       libpam_ldap.so.1 deny_local
su       account required       libpam_hpsec.so.1
su       account sufficient     libpam_unix.so.1
su       account required       libpam_ldap.so.1 deny_local
dtlogin  account required       libpam_hpsec.so.1
dtlogin  account sufficient     libpam_unix.so.1
dtlogin  account required       libpam_ldap.so.1 deny_local
dtaction account required       libpam_hpsec.so.1
dtaction account sufficient     libpam_unix.so.1
dtaction account required       libpam_ldap.so.1 deny_local
ftp      account required       libpam_hpsec.so.1
ftp      account sufficient     libpam_unix.so.1
ftp      account required       libpam_ldap.so.1 deny_local
rcomds   account required       libpam_hpsec.so.1
rcomds   account sufficient     libpam_unix.so.1
rcomds   account required       libpam_ldap.so.1 deny_local
sshd     account required       libpam_hpsec.so.1
sshd     account sufficient     libpam_unix.so.1
sshd     account required       libpam_ldap.so.1 deny_local
OTHER    account required       libpam_hpsec.so.1
OTHER    account sufficient     libpam_unix.so.1
OTHER    account required       libpam_ldap.so.1 deny_local
#
    .
    .
    .
```

2.5.6.3 Configuring PAM_LDAP authentication to ignore specific users

When PAM_LDAP is configured to be the first service module in the `/etc/pam.conf` file (a typical configuration in the Trusted Mode Environment), then if you lose access to your directory server, you might have difficulty accessing the system again unless you are included in a set of so-called “recovery users” configured in the `/etc/pam_user.conf` file. LDAP-UX 5.0 (and later) supports the `ignore` option for PAM_LDAP, which you can configure in `pam_user.conf` for specific users (such as root). This feature enables the specified users to be ignored for authentication by PAM_LDAP (PAM returns PAM_IGNORE). LDAP-UX supports this feature in both Standard Mode and Trusted Mode.

The `/etc/pam_user.conf` file is an optional user configuration file for PAM. It is used only when a user-based configuration is needed. It mainly specifies options used by service modules for specific users. The options defined in `/etc/pam.conf` specify the default for users who are not configured in `/etc/pam_user.conf` or for users without a module type configured for them. The `/etc/pam.conf` file is required for PAM to work properly.

To configure the `ignore` option, perform the following steps:

1. For each user that you want bypassed by PAM authentication, enter a line in the `/etc/pam_user.conf` file, using the following format:

   user module_type **libpam_ldap.so.1 ignore**

   where:

   user Specifies the user to be ignored by PAM_LDAP authentication

module_type	Specifies the service module type: authentication (auth), account management (account), session management (session), or password management (password).
libpam_ldap.so.1	Specifies the pathname to the PAM_LDAP library object that implements the service functionality. If the pathname is not absolute, it is assumed to be relative to `/usr/lib/security/$ISA/`.
ignore	Specifies the `ignore` option.

The following is an example of a `pam_user.conf` file, showing the `ignore` option specified for user root under authentication management, account management, session management, and password management. As a result, when user root attempts to log in to the directory server, the PAM_LDAP module does not authenticate the user root; it just returns PAM_IGNORE.

```
#################################################################
# /etc/pam_user.conf                                            #
# Sample configuration for using the ignore option for PAM_LDAP#
# for user root.                                                #
# The format for a entry is                                     #
# <user> <module type> <module path> <options>                  #
#                                                               #
# See pam_user.conf(4) for more details.                        #
#                                                               #
#                                                               #
# NOTE: If the path to a library is not absolute, it is assumed#
# to be relative to the directory /usr/lib/security/$ISA.       #
# The "$ISA (i.e Instruction Set Architecture) token is         #
# replaced by the PAM engine (libpam) with "hpux64" for IA      #
# 64-bit modules, or with "hpux32" for IA 32-bit modules, or    #
# with "pa20_64" for PA 64-bit modules, or with NULL for PA     #
# 32-bit modules.                                               #
# For PA applications, library name ending with "so.1" is a     #
# symbolic link that points to the corresponding PA (32 or 64   #
# bit) backend library.                                         #
#################################################################

root    auth         libpam_ldap.so.1    ignore
root    account      libpam_ldap.so.1    ignore
root    session      libpam_ldap.so.1    ignore
root    password     libpam_ldap.so.1    ignore
```

For more details, see the *pam_user.conf*(4) manpage. For more information about HP-UX user authentication and PAM, see the *HP-UX System Administrator's Guide: Security Management*, available at the following location:

www.hp.com/go/hpux-core-docs (click **HP-UX 11i v3**)

2. Configure the PAM_UPDBE library (`libpam_updbe`) in the `/etc/pam.conf` file.

NOTE: You must configure this library in order for the configuration in `/etc/pam_user.conf` to take effect.

PAM_UPDBE is the user policy definition service module for PAM. It reads options defined in the user configuration file, `/etc/pam_user.conf`, and uses `pam_set_data` to store the information in the PAM handle for use by subsequent service modules. In `/etc/pam.conf`, configure the PAM_UPDBE library for each service module defined in `/etc/pam_user.conf`, using the following format for each line entered:

user module_type **required libpam_updbe.so.1**

where:

user	Specifies the user to be ignored by PAM_LDAP authentication

module_type	Specifies the service module type: authentication (auth), account management (account), session management (session), or password management (password).
required	Specifies the control flag as required (mandatory).
libpam_updbe.so.1	Specifies the pathname to the PAM_UPDBE shared library object that implements the service functionality. If the pathname is not absolute, it is assumed to be relative to `/usr/lib/security/$ISA/`.

For more details, see the *pam_updbe*(5) and *pam_user.conf*(4) manpages.

If PAM_HPSEC has been configured in `/etc/pam.conf` for the service you are going to define for PAM_UPDBE, configure the PAM_UPDBE library in the line immediately following the line that configures the service's PAM_HPSEC library. If PAM_HPSEC is not configured for the given service, configure the PAM_UPDBE library as the first service module in `/etc/pam.conf`. In either case, set the control flag for the PAM_UPDBE library as `required`.

PAM_UPDBE provides interfaces for all four PAM service module types (authentication management, account management, session management, and password management). Each service module reads the options defined for its type.

The following example is a portion of a `/etc/pam.conf` file that defines the PAM_UPDBE library for the user `login` process. Because the PAM_HPSEC library is configured for each service module, the PAM_UPDBE library configuration line is added immediately following the corresponding service's PAM_HPSEC library configuration line.

```
#  Authentication management
#
login   auth  required        libpam_hpsec.so.1
login   auth  required        libpam_updbe.so.1
login   auth  sufficient      libpam_ldap.so.1
login   auth  required        libpam_unix.so.1  try_first_pass
#  Account management
#
login   account  required      libpam_hpsec.so.1
login   account  required      libpam_updbe.so.1
login   account  sufficient    libpam_ldap.so.1
login   account  required      libpam_unix.so.1
#  Session management
#
login   session  required      libpam_hpsec.so.1
login   session  required      libpam_updbe.so.1
login   session  required      libpam_ldap.so.1
login   session  required      libpam_unix.so.1
#  Password management
#
login   password  required     libpam_hpsec.so.1
login   password  required     libpam_updbe.so.1
login   password  sufficient   libpam_ldap.so.1
login   password  required     libpam_unix.so.1  try_first_pass
```

For more information, see the the *pam.conf*(4) and *pam_updbe*(5) manpages.

2.5.7 Configuring subsequent client systems

Once you have configured your directory and one client system, you can configure subsequent client systems using the following steps. If you used `autosetup` to create your LDAP-UX domain, you should continue to use `autosetup` to configure subsequent clients, since it provisions the HP-UX host information in the directory server. To do this, you can run `autosetup` in silent mode, as described in Section 2.3.8.2 (page 56).

1. Use `swinstall` to install LDAP-UX Client Services on the client system. This does not require rebooting the client system.

   Alternatively, use the guided installaton (`autosetup`), for a simpler, automated process. If you use `autosetup`, you can skip to the last step to verify the installation and configuration on the client.

2. If you used the guided installation (`autosetup`) for a simpler, automated procedure, you can skip to the last step to verify the installation and configuration on the client. Otherwise, copy the following files from a configured client to the client being configured:
   - `/etc/opt/ldapux/ldapux_client.conf`
   - `/etc/opt/ldapux/ldapux_profile.ldif`
   - `/etc/opt/ldapux/pcred` only if you have configured a proxy user, not if you are using only anonymous access
   - `/etc/pam.conf`
   - `/etc/nsswitch.conf`
   - `/etc/opt/ldapux/acred` if the `/etc/opt/ldapux/acred` file exists
   - `cert8.bd` and `key3.db` files, if SSL is enabled

   Set all file access mode permission to be the same as those of the first client being configured.

3. Enable the LDAP-UX configuration profile as follows:

   ```
   cd /opt/ldapux/config
   ./create_profile_cache
   ```

   Alternatively you could interactively run the `setup` program to download the profile from the directory and respond "no" when asked if you want to change the current configuration:

   ```
   cd /opt/ldapux/config
   ./setup
   ```

4. If you are using a proxy user, configure and verify the proxy user by calling `ldap_proxy_config` as follows:

   ```
   cd /opt/ldapux/config
   ./ldap_proxy_config  -v
   ```

5. Verify the LDAP-UX Client Services installation and configuration on the client, as described in Section 2.5.2 (page 92).

2.5.8 Downloading the profile periodically

The `setup` program allows you to define a time interval after which the current profile is being automatically refreshed. The start time for this periodic refresh is defined by the time the `setup` program was run and the value defined for ProfileTTL. Therefore, it does not allow you to define a specific time of day when the profile should be downloaded (refreshed). For more detailed information, see the *ldapclientd*(1) manpage.

If you would like to manually control when you want to download the profile, you can use the following steps:

1. When creating your profile entry using `setup`, set the `ProfileTTL` value to 0.
2. Using the command `get_profile_entry -s nss`, write a shell script that downloads the profile. Below is an example that downloads the profile from the directory. Modify this example for your environment. It also compares the new and old profiles and emails a status message:

```
#!/bin/ksh
cp /etc/opt/ldapux/ldapux_profile.ldif /etc/opt/ldapux/ldapux_profile.sav
/opt/ldapux/config/get_profile_entry -s nss 2>&1>/tmp/profile.upd$$
diff /etc/opt/ldapux/ldapux_profile.ldif \
/etc/opt/ldapux/ldapux_profile.sav >> /tmp/profile.upd$$
if [ -s /tmp/profile.upd$$ ]; then
   cat /tmp/profile.upd$$ | mailx -s "Profile cache
refreshed." root@sys01
else
   echo "No changes." | mailx -s "Profile cache refreshed."
root@sys01
fi
rm -f /etc/opt/ldapux/ldapux_profile.sav
rm -f /tmp/profile.upd$$
```

3. Use the `crontab` command to create a `crontab` file (or edit your existing `crontab` file) and specify how frequently you want the profile to be downloaded. For example, assuming the script above is in the file `/ldapux/download_ldap_profile`, the following `crontab` specification specifies that `/ldapux/download_ldap_profile` be executed nightly at midnight:

   ```
   0 0 * * * /ldapux/download_ldap_profile
   ```

   For more information about the `crontab` command, see the *crontab*(1) manpage.

4. Log in as root and schedule the job with the `crontab` command. For example, assuming the `crontab` entry above is in the file `crontab.profile`, the following schedules the profile downloading:

   `crontab crontab.profile`

2.5.9 Using the r-command for PAM_LDAP

LDAP-UX supports use of r-commands (commands for remote exeuction, such as `rlogin`, `rcp`, and so forth) with LDAP account users whose password is hidden, or not in clear text or crypt syntax.

To enable the use of r-commands, follow these steps:

1. Comment out the following line in the `/etc/opt/ldapux/ldapux_client.conf` file:

   #password_as = "x"

2. Modify the account management session in the `/etc/pam.conf` file for PAM_LDAP and add the "`rcommand`" option as follows:

```
# Account management
#
login     account required    libpam_hpsec.so.1
login     account sufficient  libpam_unix.so.1
login     account required    libpam_ldap.so.1 rcommand
```

```
su         account required    libpam_hpsec.so.1
su         account sufficient  libpam_unix.so.1
su         account required    libpam_ldap.so.1
dtlogin    account required    libpam_hpsec.so.1
dtlogin    account sufficient  libpam_unix.so.1
dtlogin    account required    libpam_ldap.so.1
dtaction   account required    libpam_hpsec.so.1
dtaction   account sufficient  libpam_unix.so.1
dtaction   account required    libpam_ldap.so.1
ftp        account required    libpam_hpsec.so.1
ftp        account sufficient  libpam_unix.so.1
ftp        account required    libpam_ldap.so.1
rcomds     account required    libpam_hpsec.so.1
rcomds     account sufficient  libpam_unix.so.1
rcomds     account required    libpam_ldap.so.1 rcommand
sshd       account required    libpam_hpsec.so.1
sshd       account sufficient  libpam_unix.so.1
sshd       account required    libpam_ldap.so.1
OTHER      account sufficient  libpam_unix.so.1
OTHER      account required    libpam_ldap.so.1
```

CAUTION: Setting the user password to be returned as any string for the hidden password, and turning on the "rcommand" option for PAM_LDAP account management could allow users with active accounts on a remote host to rlogin to the local host on to a disabled account.

If you have security concerns, see Section 5.3.10 (page 153) section in chapter 5 and “Sample /etc/pam.conf file for security policy enforcement” (page 357) for information on how to configure the PAM_AUTHZ library and the `rcommand` option under the account management section in the `/etc/pam.conf` file.

3 LDAP Printer configurator support

This chapter contains information describing how LDAP-UX supports the printer configurator, how to set up the printer schema, and how to configure the printer configurator to control its behaviors.

3.1 Overview

Management of network printing is complex, and printers themselves are more complicated. Instead of having printer configuration and information scattered over client systems and printer servers, they can be stored and managed from a single repository. LDAP is suited to build a backend printer configuration database. LDAP-UX enables the centralized management of printers, and the printer entries can easily be distributed to clients to reduce concerns about synchronization of configuration information. LDAP-UX comes with a printer configurator to consolidate printer configuration and control of printer devices into the LDAP Directory Server for a central location of printer management.

3.1.1 Definitions

3.1.1.1 Printer services

HP-UX provides LP spooler system with the LP subsystem to manage printers and print services requests. The LP subsystem is a collection of 18 programs that operate on the resources (files and subdirectories) in LP spool directory to perform their functions, such as `lpadmin`, `rlpdaemon` programs, and `lp` command.

3.1.1.2 Printing protocol

The LP spooler system has built-in support for sending jobs to other hosts that running `rlpdaemon`. `rlpdaemon` is a line printer daemon (LPD) for handling remote spool requests. This feaure enables the user to install a printer on one host and make it accessible from other hosts. It also works with printers/printservers that have network interfaces that support the LDP protocol. The LPD network printing protocol is the widely used network printing protocol in the UNIX world.

3.1.1.3 LP printer types

The LP spooler supports the following three types of printers:

- A network printer which is a printer connected to a network interface or printserver.
- A remote printer is a printer configured on a system other than the one you are logged into when you submit a print request.
- A local printer which is a printer that is directly connected to your system.

NOTE: The LDAP printer configurator only supports the HP LP spooler system, remote printers, network printers and printerservers that support Line Printer Daemon (LPD) protocol. It does not support local printers.

3.2 How the LDAP printer configurator works

The Printer Configurator is a service daemon that provides the following functions:

- Periodically searches the existing printer entries stored in LDAP Directory Server
- Compares the search result with the master printer record file on each scheduled ldapsearch
- Adds the print configuration to client system for each new printer
- Deletes the printer from the client system for each removed printer

- Updates master printer record file

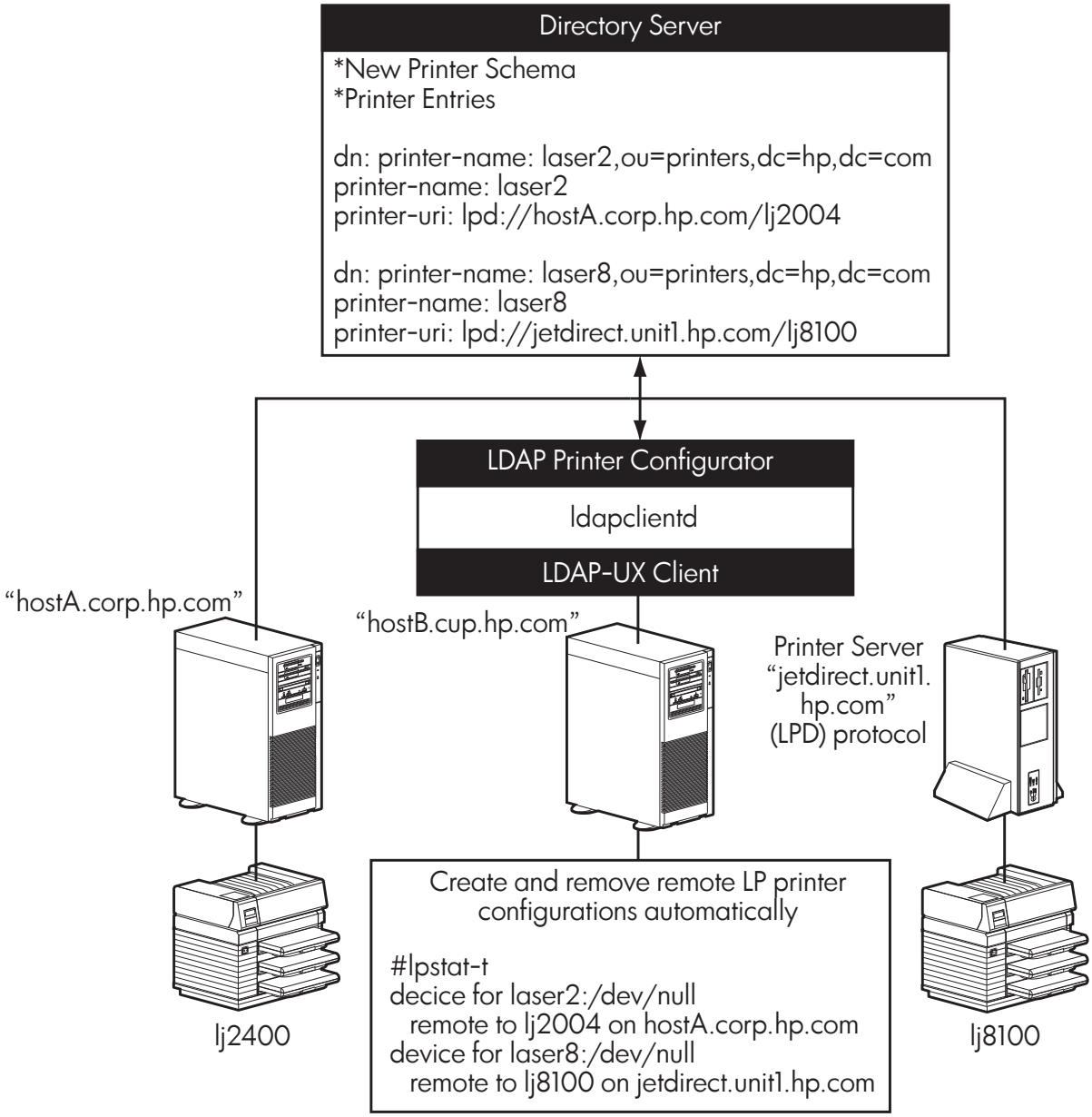
When `ldapclientd` is initialized, it will enable the printer configurator services at the same time. Once the printer configurator is up, it periodically searches for any existing printer entries in the LDAP Direcotry Server based on predefined search filters. If there are any printer entries in the LDAP Directory Server, the printer configurator will extract the LP printer configuration from each printer entry.

Then, the printer configurator compares the printer configuration with the current LP printer configuration in the client system. The result of comparison will generate a list of new or removed printers. For a new printer, the printer configurator adds this printer to the LP printer spool of the client which is running the printer configurator. For a removed printer, the printer configurator deletes this printer from the LP printer spool of the client.

With the printer configurator, if a printer administrator attempts to remove or add a printer, all the administrator has to do is to add or delete the printer entry in the LDAP Directory Server. The printer configuration will be updated automatically without manually setting the printers on each client system. Figure 3-1 (page 117) shows the basic printer configurator set up within an LDAP-UX client, with a directory server that includes the printer entries for the various hosts shown.

NOTE: The system administrator manually adds or removes printers to the HP-UX system. The LDAP Printer Configurator will only add or remove printers that it has discovered in the LDAP directory according to the search filter defined for the printer.

Figure 3-1 Printer configurator architecture



3.3 Printer configuration parameters

The LDAP-UX Client Services provides four printer configuration parameters, `start`, `search_interval`, `max_printers` and `lpadmin_option`available for you to customize and control the behaviors of the printer configurator. These parameters are defined in the `ldapclientd.conf` file. For detailed information on these new parameters, see "Administering LDAP-UX Client Services" (page 129).

3.4 Printer schema

The new printer schema, RFC 3712, is used to create the printer objects that are relevant to the printer configurator services. The draft printer schema can be obtained from the IETF website:

http://www.ietf.org

For the detailed structure information of the new printer schema, see Appendix C. You must import the new printer schema into the LDAP Directory Server to create new printer objects.

NOTE: The LDAP printer configurator supports any Directory Servers that support the LDAP printer schema based on RFC 3712.

3.4.1 An example

The following shows a typical printer object entry:

```
dn: printer-name=printer1,ou=printers,dc=cup,dc=hp,dc=com
objectclass: top
objectclass: printerabstract
objectclass: printerservice
objectclass: printerlpd
printer-name: lj81003
printer-uri: lpd://hostA.hp.com/lj81003
printer-location: 47L
printer-make-model: hp laser jet 81003
printer-service-person: John Louie
```

With the new printer schema, you are able to create printer objects for the LP printer configuration. The minimum information for a printer object entry is the local printer name, remote host name, and the remote printer name. The remote host name is the system or device that the remote printer is connected to. The remote host name must be the fully qualified name.

The `printer-name` attribute provides information of local printer name, the `printer-uri` attribute identifies the remote host name and the remote printer name information. URI stands for uniform resources identifier. The syntax of URI is based on RFC 2396. The following shows an example of the `printer-uri` attribute:

```
printer-uri: lpd://hostA.hp.com/lj2004
```

3.5 Managing the LP printer configuration

The LDAP-UX Client Services provide the printer configurator integration; the product daemon automatically updates the remote LP printer configuration of a client system based on the available printer objects in the Directory Server. The printer configurator provides the printer configuration management; it verifies if the printer configuration has any conflict with the LP printer configurations in the client system before it actually adds or deletes a printer.

Following are five examples to show how the LDAP printer configurator provides central management of printer services based on the printer objects stored in the Directory Server:

Example 1:

An administrator sets up a new printer located in the Engineering Lab and wants this printer to be shared. This printer is physically connected to a system `hostA` and is set up as a local printer `lj2004`. The administrator creates a new printer entry in the directory server as follow:

```
dn: printer-name=laser2,ou=printers,dc=hp,dc=com
printer-name: laser2
printer-uri: lpd://hostA.hp.com/lj2004
```

A new printer configuration for `laser2` is created automatically in every client system if the LDAP printer configurator is running. The print queue for `laser2` is enabled and ready to accept print jobs. Users can sent their print jobs to `laser2` by typing `lp -dlaser2` filename.

Example 2:

IT department would like to store additional service information in the printer object. The administrator modifies the printer object by adding more printer attributes. The modified content of the printer object is shown as below:

```
dn: printer-name=laser2,ou=printers,dc=hp,dc=com
printer-name: laser2
printer-uri: lpd://hostA.cup.hp.com/lj2004
printer-location: Engineering Lab
printer-model: Hewlett Packard laserjet Model 2004N
printer-service-person: David Lott
```

Since the local printer name, remote host name, remote printer name, and the printing protocol information are still the same, the LDAP Printer Configurator will not change the current remote LP printer configuration for `laser2`.

Example 3:

The system `hostA.hp.com` is retired. The Laserjet 2004 printer is now connected to system `hostC` and set up as a local LP printer `lj2004`. The administrator should update the printer object by changing the value in `printer-uri` attribute. The following shows the updated information of print objects:

```
dn: printer-name=laser2,ou=printers,dc=hp,dc=com
printer-name: laser2
printer-model: Hewlett Packard laserjet Model 2004N
printer-service-person: David Lott
```

The current remote LP `laser2` printer configuration is removed from the client system, and the new `laser2` printer configuration with new remote host name information is added to the client system. In fact, if either remote host name or remote printer name of `printer-uri` attribute is modified, the printer configurator will remove the current remote LP printer configuration and create the new printer configuration with the updated resource information.

Example 4:

The remote LP printer, `laser2`, no longer supports LPD printing protocol. IPP printing protocol is implemented instead. The administrator updated the printer object by changing the printing protocol to IPP. The following shows the updated printer objects in the directory server:

```
dn: printer-name=laser2,ou=printers,dc=hp,dc=com
printer-name: laser2
printer-uri: ipp://hostC.hp.com/lj2004
printer-location: Engineering Lab
printer-model: Hewlett Packard laserjet Model 2004N
printer-service-person: David Lott
```

IPP printing protocol is not supported by the LP spool printing system. The only action that the LDAP printer configurator will take is to remove the current `laser2` printer configuration on the client system.

Example 5:

The administrator created a new printer object in the directory server as below:

```
dn: printer-name=laser8,ou=printers,dc=hp,dc=com
printer-name: laser8
printer-uri: lpd://hostD.hp.com/lj81003
```

In this example, the printer configurator adds a new remote LP `laser8` printer configuration to the client system.

However, if the user attempts to remove the `laser8` printer configuration manually, the printer configuration will no longer be managed by the printer configurator. The user has to recreate the printer configuration manually in case the laser8 printer is needed. The printer configurator does not try to create the printer configuration even though the printer object of `laser8` still exists in the directory server.

If the user manually adds a remote LP printer configuration to the client system, the new printer configuration will not be managed by the printer configurator. The user has to remove the printer configuration manually if the remote LP printer is no longer needed.

3.6 Limitations of the printer configurator

- The new LDAP printer schema based on RFC 3712 is imported into the LDAP Directory Server to create the printer objects.
- LDAP-UX Client Services only suports the HP-UX LP spooler system, network printers, and printerservers that support Line Printer Daemon (LPD) protocol. The printer configurator does not support local printers.
- In a global management envoriment, it is hard to determine a default printer for the individual client system. The LDAP printer configurator treats every printer entry as the regular printer. The administrator or user requires to manually select a printer as a default printer for the client system.

4 Dynamic group support

This chapter contains information about how LDAP-UX Client Services supports dynamic groups, how to set up dynamic groups, and how to enable or disable dynamic group caches.

4.1 Overview

A system administrator can associate some users with a group, and apply security policies (e.g. access control, password policies) to the group. As a result, all users belonging to the group inherit the specific policies, such as being able to access a file. In LDAP directories, there are two types of groups: static groups and dynamic groups. A static group defines all users statically. Each user must be added to the group individually and explicitly. Dynamic groups associate users with a group based on conditions. The condition can be specified by an LDAP URL or a search filter. When a user's data matches with the conditions, she/he belongs to the dynamic group. Dynamic groups offer the advantage of flexibility, and allow administrators to easily implement a role-based authorization policy based upon a company's organizational structure. Users can be added to or removed from a group dynamically based on his/her most current status (such a value of one or more attributes in the user's entry).

Since traditional POSIX-style groups are used largely to control file system access rights, dynamic groups in LDAP-UX offers a new and flexible method for defining file system access policies. For example, with file system access control lists (ACLs) it is possible to add group access permission for users that are a member of a particular group (say the "top secret" group). With dynamic groups, instead of needing to insert each individual member in the group, LDAP-UX discovers all users in the directory that have the "top secret" attribute associated with their entries. And when a user's attribute is no longer defined as "top secret", his/her group membership in the "top secret" is automatically revoked (no need to make manual changes to the group).

LDAP-UX Client Services supports dynamic groups and allows you to configure dynamic groups using the same syntaxes as the following directory servers and identity management:

- HP-UX Directory Server or Redhat Directory Server
- Windows Server 2003 R2/2008 Active Directory Server

4.2 Specifying an LDAP URL for a dynamic group

HP-UX Directory Server and Redhat Directory Server define the `memberURL` attribute and the `groupOfURLs` objectclass to represent the dynamic group. All POSIX users who can be found using the LDAP URL belong to the group.

4.2.1 Creating an HP-UX POSIX dynamic group

LDAP-UX Client Services only supports HP-UX POSIX dynamic groups. Use the following procedures to create an HP-UX POSIX dynamic groups:

1. Use the Directory Server Console to create a dynamic group, as described in Section 4.2.1.1.
2. Add the `posixgroup` objectclass and `gidNumber` attribute information to the dynamic group entry created in the preceding step, as described in Section 4.2.1.2.

4.2.1.1 Step 1: Creating a dynamic group

You can use the Directory Server Console to create a dynamic group. For detailed information on how to use the Directory Server Console to create a dynamic group, see the *HP-UX Directory Server administrator guide* available at the following website:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX Directory Server**.

The following shows an example of a dynamic group entry created using the Directory Server Console:

```
dn: cn=dyngroup,ou=groups,dc=example,dc=hp,dc=com
cn=dyngroup
objectClass: top
objectClass: groupofuniquenames
objectClass: groupofnames
```

objectClass: groupofurls

memberURL: ldap:///dc=example,dc=hp,dc=com??sub?(l=California)

The `memberURL` attribute in the above example specifies a sub-tree search starting at any level under dc=example, dc=hp, dc=com to find all entries matching (l=California). Any entries which have objectclass "`account`" and an attribute "`l`" with the value of "`California`" will be returned. With LDAP-UX, an additional criteria will be added that the user entry must be a POSIX account.

4.2.1.2 Step 2: Adding POSIX attributes to a dynamic group

To create an HP-UX POSIX dynamic group, you must use the Directory Server Console, or the `ldapmodify` tool to add the following objectclass and attribute information to the dynamic group entry created in Step 1: Creating a Dynamic Group:

- `posixgroup` objectclass
- `gidNumber` attribute
- `cn` attribute if it does not exist in the group entry.

4.2.1.2.1 Adding attributes to a dynamic group using ldapmodify

Procedures

As an example, to create an HP-UX POSIX dynamic group, use the `ldapmodify` tool to add `posixgroup` and `gidNumber` information to the dynamic group entry created from the Directory Server Console as follows:

1. Create an LDIF update file.

   For example, the following LDIF update file, `new.ldif`, adds a `posixgroup` objectclass and the `gidNumber` attribute to the "`dn: cn=dyngroup,ou=groups,dc=example,dc=hp,dc=com`" entry:

   ```
   dn: cn=dyngroup,ou=groups,dc=example,dc=hp,dc=com
   changetype: modify
   add: objectClass
   objectClass: posixgroup
   -
   add: gidNumber
   gidNumber: 500
   ```

2. Use the `ldapmodify` tool to modify the existing entry with the LDIF file created in step 1.

   For example, the following command modifies the dynamic group entry in the LDAP directory server, `ldaphost1`, using the LDIF update file, `new.ldif`:

   ```
   ldapmodify –D "cn=Directory Manager" –w <passwd> –h ldaphost1 –p
   389 –f new.ldif
   ```

Examples

The following example is an HP-UX POSIX dynamic group entry with `objectClass: posixgroup` and `gidNumber: 500` information added:

```
dn: cn=dyngourp,ou=groups,dc=example,dc=hp,dc=com
```

```
objectClass: groupofuniquenames
objectClass: groupofnames
objectClass: groupofurls
objectClass: posixgroup
objectClass: top
cn: dyngroup
memberURL: ldap:///dc=example,dc=hp,dc=com??sub?(l=California)
gidNumber: 500
```

4.2.2 Changing an HP-UX POSIX static group to a dynamic group

To change an HP-UX POSIX static group to an HP-UX POSIX dynamic group, use the Directory Server Console to add the following objectclass and attribute information to the HP-UX POSIX static group:

- `groupofurls` objectclass
- `memberURL` attribute

For detailed information on how to use the Directory Server Console to modify a group, see the *HP-UX Directory Server administrator guide* available at the following website:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX Directory Server**.

The following shows an example of an HP-UX POSIX static group entry:

```
dn: cn=all,ou=groups,dc=example,dc=hp,dc=com
objectClass: groupofuniquenames
objectClass: groupofnames
objectClass: posixgroup
objectClass: top
cn: all
gidNumber: 1000
memberuid: user1
```

After you add information for `groupofurls` and `memberURL` to the above HP-UX POSIX static group entry, the HP-UX POSIX dynamic group entry is as follows:

```
dn: cn=all,ou=groups,dc=example,dc=hp,dc=com
objectClass: groupofuniquenames
objectClass: groupofnames
```

objectClass: groupofurls

```
objectClass: posixgroup
objectClass: top
cn: all
```

memberURL: ldap:///dc=example,dc=hp,dc=com??sub?(l=California)

```
gidNumber: 1000
memberuid: user1
```

Now, the group "`all`" contains both static group member (i.e. user1) and dynamic members (i.e. all user entries which can be retrieved from the tree of `dc=example,dc=hp,dc=com` and have an attribute with `l=California`).

4.3 Multiple group attribute mappings

By default, LDAP-UX uses the `memberUid` attribute to retrieve group members. With the support of X.500 group member syntax, you can map the default group attribute, `memberUid`, to `member` or/and `uniquemember`, which you specify group members using user DNs. With dynamic group support, LDAP-UX allows you to map `memberUid` to `memberURL` (if you use HP-UX Directory Server or Redhat Directory Server to create dynamic groups) or `nxSearchFilter` (if you use HP OpenView Select Access or HP-UX Select Access for IdMI to create dynamic groups).

You can run the `setup` program and map `memberUid` to multiple attributes as needed. For example, the following output of `/opt/ldapux/config/display_profile_cache` shows that `memberUid` is mapped to both static group attributes, `memberUid`, `member` and `uniquemember`, and dynamic group attribute `memberURL`:

```
Group Service Configuration:

      Attribute:              is mapped to:
```

```
-----------              -------------
name:                        cn
gid:                         gidnumber
members:                     memberuid memberURL
                             member uniquemember
```

LDAP-UX retrieves group members and processes groups that a specific user belongs to by looking into all configured attributes. If needed, you can create a group which include both static and dynamic members. When returning group members, LDAP-UX will return both static and dynamic members that belong to a specific group.

When processing dynamic group attributes, LDAP-UX combines the search filter of the passwd service from the profile with the search filter specified in `membeURL` (e.g. the last component in `memberURL`) or `nxSearchFilter` to retrieve group members. This is to make sure that group members returned are POSIX accounts and meet the configuration set for LDAP-UX.

4.3.1 Examples

The following is an example of the output of `/opt/ldapux/config/display_profile_cache`:

```
PASSWD Service Configuration

       Attribute:            is mapped to:
       ----------            -------------
         name:                   uid
         uid number:             uidnumber
         .....

         Search Descriptor
         search[0]:              dc=example,dc=hp,dc=com?sub?
                                 (objectclass=posixaccount)
```

The sample group entry is:

```
dn: cn=mygroup,ou=Groups,dc=example,dc=hp,dc=com
objectClass: groupofnames
objectClass: groupofuniquenames
objectClass: posixgroup
objectClass: groupofurls
objectClass: top
cn: mygroup
gidNumber: 100
memberUid: user1
member: uid=user2,ou=people,dc=example,dc=hp,dc=com
uniqueMember: uid=user3,ou=people,dc=example,dc=hp,dc=com
memberURL: ldap:///dc=example,dc=hp,dc=com??sub?(uid=p*)
```

When processing `memberURL` to retrieve dynamic members, LDAP-UX combines `(objectclass=posixaccount)` from passwd configuration with `(uid=p*)` as the search filter to search the tree of "`dc=example,dc=hp,dc=com`".

With the above attribute mappings, LDAP-UX will return `user1`, `user2`, `user3` and all users starting with "p" as group members.

4.3.2 Group attribute mappings

To enable the dynamic group feature support, you must run the `setup` program to remap the default group attribute, `memberuid`, to the dynamic group attribute, `memberURL`. If `memberURL` is not mapped to `memberUid`, LDAP-UX will not process dynamic groups.

The attribute mappings are done in step 10 of the Custom Configuration. For detailed information on how to remap the group attributes, see Section 2.4.5.2 (page 73).

Table 4-1 shows attribute mappings between the default group attribute and alternate group attributes:

Table 4-1 Attribute mappings

Default Group Attribute	Dynamic Group Attribute	Static X.500 Group Attribute
memberuid	memberURL	member

If you want to perform group attribute mappings by using the Custom Configuration, ensure that you do not accept the remaining default configuration parameters in step 4 of the Custom Configuration.

4.4 Number of group members returned

With dynamic membership support, as with regular (static) group membership support, the number of group members for a specific group returned by `getgrnam()/getgrgid()/getgrent()` on an HP-UX system is limited by internal buffer sizes. On HP-UX 11i v2 and v3 systems, the buffer size is 7296 bytes for 32–bit applications and 10496 bytes for 64-bit applications. This limitation is mainly impacted by the size of each member name. For detailed description, see the *Preparing your Directory for LDAP-UX Integration* white paper under the "Account and Group Management" collection available at the following website:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX LDAP-UX Integration Software**.

During the login process, information for getting group members is not requested. The login time will not be affected by processing group members.

4.5 Number of groups returned for a specific user

When "`ldap`" is configured in the `/etc/nsswitch.conf` file as a data repository for the `group` service (see *nsswitch.conf*(4)), if an LDAP user logs into an HP-UX system, LDAP-UX is involved to return all groups that the user belongs to. The login application (e.g. login) initializes the user's group access based on the group information returned by LDAP-UX.

Information for getting groups that a specific user belongs to is requested by LDAP-UX during login via `initgroups()`. LDAP-UX returns at most 20 groups for a system limit on HP-UX 11i v2 systems. On HP-UX 11i v3 systems, you can increase the number of groups a user may be a member of (known as NGROUPS). For more information, see the *Group membership expansion: guidelines for deployment* white paper at:

http://www.hp.com/go/hpux-core-docs (click **HP-UX 11i v3**)

If the user belongs to more than 20 groups, only the first 20 groups are returned. The support of dynamic groups does not change the system limitation.

Depending on how you configure groups, if those 20 groups happens to be the last entries of thousands of dynamic groups, the login time could be long and performance could be impacted.

Based on the configuration of `memberUid` attribute mappings, LDAP-UX may return static and/or dynamic groups. The first `memberUid` mapped attribute determines if LDAP-UX returns static or dynamic groups first. If the first `memberUid` mapped attribute is a static group attribute (such as `memberUid`, `member` or `uniquemember`), LDAP-UX returns static groups first. If there are less than 20 static groups, LDAP-UX then returns dynamic groups for the rest groups. However, if the first `memberUid` mapped attribute is a dynamic group attribute (such as `memberURL` or `nxSearchFilter`), LDAP-UX returns dynamic groups first. If there are less than 20 dynamic groups, LDAP-UX then returns static groups for the rest groups.

With this design, a group containing both static and dynamic group attributes will be always processed, but will be limited to the first 20 groups.

For example, if a user belongs to 8 static groups and 20 dynamic groups, and you map `memberUid` to `memberUid memberURL`, LDAP-UX will return 8 static groups and 12 dynamic groups. If you map `memberUid` to `memberURL memberUid`, LDAP-UX will return 20 dynamic groups without any static groups.

4.6 Performance impact for dynamic groups

The dynamic group is specified by either an LDAP URL or a search filter. Depending on how you configure dynamic groups, potentially, there could be a lot of LDAP searches involved. In that case, the performance of those applications calling `getgrnam()`, `getgrgid()` or `getgrent()(3C)` (e.g. the command "id", "groups", etc) will be affected.

To reduce the performance impact, the LDAP-UX client daemon (`ldapclientd`), caches dynamic group information, including dynamic members that belongs to a specific group, and dynamic groups that a specific user belongs to. The caching will reduce the response time the `ldapclientd` daemon to return information. However, before the cache is established (i.e. the very first request) or when the cache expires, you may experience longer response time. See Section 4.7 (page 128) for detailed information on dynamic group caching.

4.6.1 Enabling/disabling enable_dynamic_getgroupsbymember

Processing dynamic groups that a specific user belongs to can potentially impact the user login time. To control the operation for processing dynamic groups a specific user belongs to, LDAP-UX Client Services supports the following configuration parameter, `disable_dynamic_getgroupsbymember`, in the `/etc/opt/ldapux/ldapux_client.conf` file:

enable_dynamic_getgroupsbymember

This integer variable controls whether to enable or disable the operation for processing dynamic groups that a specific user belongs to. The valid values of this option are 1 and 0.

By default, LDAP-UX returns dynamic groups that a user belongs to if the group attribute, `memberUid`, is mapped to `memberURL` or/and `nxSearchFilter`. If a user belongs to many dynamic groups, he/she may experience an unexpected delay when logging into an HP-UX client system.You can reduce the delay by disabling LDAP-UX of returning dynamic groups that a specific user belongs to unless he/she specifically uses the `newgrp` command. As a result, the user will not have access granted to those dynamic groups, and the "`id`" command will not show those groups. To disable it, set `enable_dynamic_getgroupsbymember` to 0. This parameter configuration does not affect the operation of processing dynamic members for a specific group. The default value is 1 to enable it.

NOTE: If the `enable_dynamic_getgroupsbymember` variable is set to 0, LDAP-UX will still return dynamic members for a specific group. If you don't want dynamic members returned, you must not include the `memberURL` and `nxSearchFilter` attributes in the `memberUid` group attribute mapping, which completely disables the dynamic group functionality with LDAP-UX.

4.7 Configuring dynamic group caches

To improve performance of dynamic groups, the LDAP client daemon, `ldapclientd`, caches dynamic group members to reduce the LDAP-UX client response time while retrieving dynamic group information. This cache is maintained in an independent memory space not shared with the cache for other service data.

To configure dynamic group caches, set the parameters defined in the `[dynamic_group]` section of the `/etc/opt/ldapux/ldapclientd.conf` file. See Section 5.1.3 (page 131) for details.

5 Administering LDAP-UX Client Services

This chapter describes how to keep your clients running smoothly and expand your computing environment.

5.1 Using the LDAP-UX client daemon

This section describes the following:

- Overview of ldapclientd daemon operation.
- Configurable parameters and syntax in the `ldapclientd` configuration file, `ldapclientd.conf`.
- Command line syntax and options for the `ldapclientd` command.

5.1.1 Overview

The LDAP-UX client daemon enables LDAP-UX clients to work with LDAP directory servers. It caches entries, supports multiple domains in the Windows Server 2003 R2/2008 Active Directory Server (ADS), supports X.500 group membership, automatically downloads the configuration profiles, reuses connections to the LDAP Directory Server, and manages the remote LP printer configuration.

The client daemon enables LDAP-UX to use multiple domains for directory servers like Active Directory Server (ADS). The daemon also allows PAM Kerberos to authenticate posix users stored in multiple domains.

Automatic Profile Downloading updates the LDAP client configuration profile by downloading a newer copy from the directory server as the profileTTL (Time To Live) expires.

By default, the LDAP printer configurator is enabled, the client daemon, `ldapclientd`, automatically searches printer objects configured in the LDAP server and executes `lpshut`, `lpadmin` and `lpsched` commands to add, modify, and remove printers accordingly for the local system.

By default, `ldapclientd` starts at system boot time. The ldapclientd command can also be used to launch the client daemon manually, or control it when the daemon is already running. For information about the `ldapclientd` command and its parameters, see Section 5.1.2 (page 129) and the *ldapclientd* manpage(s).

IMPORTANT: Starting with LDAP-UX Client Services B.03.20 or later, the client daemon, `/opt/ldapux/bin/ldapclientd`, must be running for LDAP-UX functions to work. With LDAP-UX Client Services B.03.10 or earlier, running the client daemon, `ldapclientd`, is optional.

5.1.2 ldapclientd

5.1.2.1 Starting the client

Use the following syntax to start the client daemon. Note the use of upper and lower-case characters:

```
/opt/ldapux/bin/ldapclientd <[-d <level>] [-o<stdout|syslog|file[=size]>]
[-z]
```

5.1.2.2 Controlling the client

Use the following syntax to control the client daemon:

```
/opt/ldapux/bin/ldapclientd <[-d <level>]
[-o<stdout|syslog|file[=size]>]>
```

```
/opt/ldapux/bin/ldapclientd <[-D <cache>]|-E <cache>|-S [cache]>
/opt/ldapux/bin/ldapclientd <-f| -k| -L| -h| -r>
```

5.1.2.3 Client daemon performance

Performance (client response time) is improved by the use of two techniques:

1. Reuse of connections to the LDAP Directory Server: This feature improves performance by reducing the overhead associated with opening and closing bindings to the directory server and significantly reduces network traffic and server load.
2. Enabling the client cache: Enabling the cache will allow the client to cache the reply information retrieved for the following maps:

   passwd
   group
   dynamic group
   netgroup
   X.500 group membership
   automount

Except for the dynamic group map, all of the above maps share a common memory space. The Dynamic Group map cache is created as an independent memory space. The length of time the reply data is held in the cache is determined by a Time To Live timer. This timer can be set for all maps or can be set independently for each of the maps listed above. The cache can also be flushed by specifying an option on the ldapclientd command. The cache space becomes available for new information after the Time to Live expires or the cache is flushed.

There are two categories of information that are held in the cache. The reply data for those requests that were successful, and replies when the information was not found. For example, when a specific user is trying to logon, the userID may or may not exist in the directory.

The Time to Live for replies that were found in the directory is set by a parameter `poscache_ttl` in the `ldapclient.conf` file and for replies where the information was not found by `negcache_ttl`.

For more information on the client daemon performance, see Section 5.17 (page 185).

5.1.2.4 Command options

For option information, see the *ldapclientd* manpage(s).

5.1.2.5 Diagnostics

By default, errors are logged into syslog if the system log is enabled in the LDAP-UX client start-up configuration file `/etc/opt/ldapux/ldapux_client.conf`. Errors occuring before `ldapclientd` forks into a daemon process leaves an error message directly on the screen.

The following diagnostic messages may be issued:

Message: Already running.

Meaning: An attempt was made to start an LDAP client daemon when one was already running.

Message: Cache daemon is not running (or running but not ready).

Meaning: This message can mean several things:

1. Attempted to use the control option features of `ldapclientd` when no `ldapclientd` daemon process was running, to control.
2. Attempted to start, or control, `ldapclientd` without superuser's privilege.
3. The `ldapclientd` daemon process is too busy with other requests to respond at this time. Try again later.

Message: Problem reading configuration file.

Meaning: The `/etc/opt/ldapux/ldapclientd.conf` file is missing or has a syntax error. If the problem is with its syntax, the error message will be accompanied by a line showing exactly where it could not recognize the syntax, or where it found a setting which is out of range.

5.1.2.6 Warnings

Whenever the system is rebooted, `ldapclientd` launches if `[StartOnBoot]` has the parameter `enabled=yes` in the file `/etc/opt/ldapux/ldapclientd.conf` (the `ldapclientd` configuation file). Downloading profiles takes time, depending on the server's response time and the number of profiles listed in the LDAP-UX start-up file `/etc/opt/ldapux/ldapux_client.conf`.

5.1.3 ldapclientd.conf

The file `ldapclientd.conf` is the configuration file for `/opt/ldapux/bin/ldapclientd`, the LDAP client daemon. For more information about the client daemon, see Section 5.1 (page 129).

5.1.3.1 Missing settings

`ldapclientd` uses the default values for any settings which are not specified in the configuration file.

5.1.3.2 Configuration file syntax

```
# comment
[section]
setting=value
setting=value
. . .
[section]
setting=value
setting=value
. . .
```

Where:

`comment` ldapclientd ignores any line beginning with a # delimiter.

`section` Each section is configured by setting=value information underneath. The section name must be enclosed by brackets ("[ ]") as delimiters. Valid section names are:

NOTE: The LDAP-UX using Windows 2003 R2/2008 Active Directory Server does not support `netgroup`, `automount`, and `publickey` service data.

- [StartOnBoot]
- [general]
- [passwd]
- [group]
- [dynamic_group]
- [netgroup]
- [uiddn]
- [domain_pwd]
- [domain_grp]
- [automount]

-[automountMap]

- [printers]

`setting` This will be different for each section.

`value` Depending on the setting, this can be `<yes|no|number>`.

5.1.3.2.1 Section details

Within a section, the following syntax applies:

[StartOnBoot] Determines if `ldapclientd` starts automatically when the system boots.

enable=<yes|no>

By default, this is enabled after LDAP-UX has been configured by the LDAP-UX `setup` program `/opt/ldapux/config/setup`.

[general] Any cache setting defined here will be used as the default setting for all caches (passwd, group, netgroup, uiddn, domain_pwd, domain_grp, automount, automountmap, and dynamic_group). The cache_size setting defined here will be used for all caches except dynamic_group.

max_conn=<2-500>

The maximum number of connections `ldapclientd` can establish to the directory server (or multiple servers when in a multi-domain Windows environment.

The default value is 100.

connection_ttl=<1-2147483647>

The number of seconds before an inactive connection to the directory server is brought down and cleaned up.

The default value is 300.

num_threads=<1-100>

The number of client request handling threads in `ldapclientd`.

The default value is 10.

socket_cleanup_time=<10-2147483647>

The interval, in seconds, before the next attempt to clean up the socket files created by any LDAP-UX client applications that were terminated abnormally.

The default value is 300.

cache_cleanup_time=<1-300>

The interval, in seconds, between the times when `ldapclientd` identifies and cleans up stale cache entries.

The default value is 10.

update_ldapux_conf_time=<10-2147483647>

This determines how often, in seconds, `ldapclientd` re-reads the `/etc/opt/ldapux/ldapux_client.conf` client configuration file to download new domain profiles.

The default value is 600 (10 minutes).

cache_size=<102400-1073741823>

The maximum number of bytes that should be cached by `ldapclientd` for all services except dynamic_group. This value is the maximum, upper limit, of memory that can be used by `ldapclientd` for all services

except dynamic_group. If this limit is reached, new entries are not cached until enough expired entries are freed to allow it.

The default value is 10000000.

state_dump_time=<0-2147483647>

As state, functions like a virtual between the client and LDAP server, is created for `setXXent()` request, and stays for the subsequent `getXXent()` requests. If no `get` requests are received in the specified time interval (in seconds), the state will be removed. The default value is 300 (in seconds).

max_enumeration_states=<0-95>[%]

The maximum number of states that `ldapclientd` allows. It means the number of enumeration `ldapclientd` will handle simultaneously. This number must be less than max_conn and it is configured as a percentage of max_conn. The minimum value is 0% and maximum value is 95%. The default value is 80%. A value of 0% disables enumeration.

poscache_ttl=<1-2147483647>

The time, in seconds, before a cache entry expires from the positive cache. There is no `[general]` default value for this setting. Each cache section has its own default values (listed below). Specifying a value under `[general]` will override `poscache_ttl` defaults in other sections (where there is no specific `poscache_ttl` definitions for that section).

negcache_ttl=<1-2147483647>

The time, in seconds, before a cache entry expires from the negative cache. There is no `[general]` default value for this setting. Each cache section has its own default value.

proxy_is_restricted=yes|no

If the proxy user is configured in the LDAP-UX profile and defined in `/etc/opt/ldapux/pcred`, this flag attests that the proxy user does not hold privileged LDAP credentials, meaning the proxy user is restricted in its rights to access "private" information in the directory server. As of release B.05.00, `ldapclientd` provides a local interface to allow specialized directory-enabled applications to access arbitrary attributes in HP-UX related directory entries. By default, and if set to `no`, `ldapclientd` will not allow access to attributes beyond that of the RFC2307 schema as well as any attribute defined using the `allowed_attribute` token. If `proxy_is_restricted` is set to `yes`, then you are attesting that the directory server is restricting access to private or other confidential information from access by the proxy user. This allows specialized applications to access any attribute visible to the proxy user. The default value for this setting is `no`, meaning `ldapclientd` assumes the proxy user has rights beyond that of a non-privileged user.

allowed_attribute=*service:attribute*

Some applications, like `/opt/ssh/bin/ssh`, use `ldapclientd` to access information in the directory server, such as the `sshPublicKey` for users and hosts. By setting this parameter, applications can access any defined attribute even if the `proxy_is_restricted` value is set to `no` (the default). There is no internal default set for this parameter. If `allowed_attribute` is not specified, no attributes beyond that defined in RFC2307 (and as mapped in the configuration profile) will be

accessible through ldapclientd's API. However, the default delivered ldapclientd.conf file will set this parameter to allow access to the sshPublicKey attribute for the passwd and hosts service. This parameter can be specified more than once. allowed_attribute example:

```
allowed_attribute=hosts:sshPublicKey
```

[passwd] Cache settings for the passwd cache (which caches name, UID, and shadow information).

enable=<yes|no>

ldapclientd only caches entries for this section, when it is enabled. If the cache is not enabled, ldapclientd will query the directory server for any entry request from this section. Since this impacts LDAP-UX client performance and response time, by default, caching is enabled.

poscache_ttl=<0-2147483647>

The time, in seconds, before a cache entry expires from the positive cache. Since personal data can change frequently, this value is typically smaller than some others.

The default value is 120 (2 minutes)

negcache_ttl=<1-2147483647>

The time, in seconds, before a cache entry expires from the negative cache.

The default value is 240 (4 minutes).

[group] Cache settings for the group cache (which caches name, gid and membership information).

enable=<yes|no>

ldapclientd only caches entries for this section, when it is enabled. By default, caching is enabled.

poscache_ttl=<0-2147483647>

The time, in seconds, before a cache entry expires from the positive cache. Since people are added and removed from groups occasionally, this value is not typically large. If dynamic_group caching is enabled, this value must be less than poscache_ttl of [dynamic_group].

The default value is 240 (4 minutes)

negcache_ttl=<1-2147483647>

The time, in seconds, before a cache entry expires from the negative cache. If dynamic_group caching is enabled, this value must be less than negcache_ttl of [dynamic_group]

The default value is 240 (4 minutes).

[dynamic_group] This section describes the settings for the Dynamic Group cache. This cache manages dynamic group information including name, group ID and membership information. This cache is maintained in a independent memory space not shared with the cache for other maps.

enable=<yes|no>

ldapclientd only caches entries for this section, when it is enabled. Since this impacts LDAP-UX client performance and response time, caching is enabled by default.

poscache_ttl=<0-2147483647>

The time, in seconds, before a cache entry expires from the positive cache. If group caching is enabled, this value must be greater than `poscache_ttl` of `[group]`.

The default value is 43200 (12 hours).

negcache_ttl=<1-2147483647>

The time, in seconds, before a cache entry expires from the negative cache. If group caching is enabled, this value must be greater than `negcache_ttl` of `[group]`

The default value is 43200 (12 hours).

cache_size=<102400–1073741823>

This integer variable specifies the maximum number of bytes that should be cached by `ldapclientd`. This value is the maximum, upper limit, of memory that can be used by `ldapclientd`. If this limit is reached, new entries are not cached until enough expired entries are freed to allow it. The default value is 100000000 (10M).

NOTE: The cache_size option defined in the [general] section is used to configure for all other caches (passwdm netgroup, group, automount, domain_pwd, domain_grp, uiddn).

[netgroup]

Cache settings for the netgroup cache.

enable=<yes|no>

`ldapclientd` only caches entries for this section, when it is enabled. By default, caching is enabled.

poscache_ttl=<0-2147483647>

The time, in seconds, before a cache entry expires from the positive cache. Since people are added and removed from groups occasionally, this value is not typically large.

The default value is 240 (4 minutes)

negcache_ttl=<1-2147483647>

The time, in seconds, before a cache entry expires from the negative cache.

The default value is 240 (4 minutes).

[uiddn]

This cache maps a user's UID to their DN from the directory.

enable=<yes|no>

`ldapclientd` only caches entries for this section, when it is enabled. By default, caching is enabled.

poscache_ttl=<0-2147483647>

The time, in seconds, before a cache entry expires from the positive cache. Typically, once added into a directory, the user's DN rarely changes.

The default value is 86400 (24 hours).

negcache_ttl=<1-2147483647>

The time, in seconds, before a cache entry expires from the negative cache.

The default value is 84400 (24 hours).

[domain_pwd] This cache maps user names and UIDs to the domain holding its entry.

enable=<yes|no>

`ldapclientd` only caches entries for this section, when it is enabled. By default, caching is enabled.

poscache_ttl=<0-2147483647>

The time, in seconds, before a cache entry expires from the positive cache. Since new domains are rarely added to or removed from the forest, the cache is typically valid for a long time.

The default value is 86400 (24 hours)

negcache_ttl=<1-2147483647>

The time, in seconds, before a cache entry expires from the negative cache.

The default value is 86400 (24 hours).

[domain_grp] This cache maps group names and GUIDs to the domain holding its entry.

enable=<yes|no>

`ldapclientd` only caches entries for this section, when it is enabled. By default, caching is enabled.

poscache_ttl=<0-2147483647>

The time, in seconds, before a cache entry expires from the positive cache. Since new domains are rarely added to or removed from the forest, the cache is typically valid for a long time.

The default value is 86400 (24 hours).

negcache_ttl=<1-2147483647>

The time, in seconds, before a cache entry expires from the negative cache.

The default value is 86400 (24 hours).

[automount] Cache settings for the automount entry cache (which caches automount entries in automount maps).

A positive cache means that the automount entry data has been recently retrieved from the LDAP directory server and is stored in the positive cache locally.

A negative cache is used to store the automount entry data about non-existent information. For example, if a user requests information about an automount entry that does not exist, the LDAP directory server will not return an entry, all the negative result will be stored in the negative cache.

enable=<yes|no>

`ldapclientd` only caches entries for this section, when it is enabled. By default, caching is enabled.

poscache_ttl=<0-2147483647>

The time, in seconds, before a cache entry expires from the positive cache. The default value is 1800 (30 minutes).

negcache_ttl=<1-2147483647>

The time, in seconds, before a cache entry expires from the negative cache.

The default value is 1800 (30 minutes).

[automountMap] Cache settings for the automount map cache.

enable=<yes|no>

`ldapclientd` only caches entries for this section, when it is enabled. By default, caching is enabled.

poscache_ttl=<0-2147483647>

The time, in seconds, before a cache entry expires from the positive cache. The default value is 1800 (30 minutes).

negcache_ttl=<1-2147483647>

The time, in seconds, before a cache entry expires from the negative cache.

The default value is 7200 (2 hours).

[printers] Any printer setting defined here will be used by the LDAP printer configurator.

start=<yes|no>

Determines if the printer configurator service will start when `ldapclientd` is initialized. If it is enabled, the printer configurator will start when `ldapclientd` is initialized. By default, the `start` parameter is enabled.

search_interval=<1800-1209600>

Defines the interval, in seconds, before the printer configurator performs a printer search in the directory server. The default value is 86400 (in seconds). The minimum value is 1800 (30 minutes) and the maximum value is 1209600 (2 weeks).

max_printers= 50 (default value)

Defines the maximum printer objects that printer configurator services will handle. For example, a number of 100 printer entries is returned to the printer configurator after a scheduled printer search. If the `max_printers` value is set to 50, only the first 50 printer entries received by the printer configurator will be processed. For this configuration parameter, the value must be greater than 0 and the maximum value is unlimited. The default value is 50.

lpadmin_option

Defines the `lpadmin` options. Do not include the `-p`, `-orm` and `-orp` options in the option fields. The LDAP printer configurator provides the required information of printer name (`-p`), remote machine name (`-orm`) and remote printer name (`-orp`) during the run time. Do not include any other parameters, such as stderr or stdout redirection options. If the option fields of the `lpadmin_option` parameter are empty or the `lpadmin_option` parameter does not exist, the default `lpadmin` options are used. By default, `lpadmin_option=-mrmodel -v/dev/null -ocmrcmodel -osmrsmodel`.

For detailed information about valid options and the syntax, see the *lpadmin* manpages.

5.1.3.3 Configuration file

The LDAP client configuration file is automatically loaded when the product is installed. For additional information, see the *ldapclientd*(4) and *ldapcltd*(4)manpages.

If you update LDAP-UX Client Services from an older version, such as B.03.00 or B.03.10, the new configuration file will be `/opt/ldapux/newconfig/etc/opt/ldapux/ldapclientd.conf`.

5.2 Integrating with Trusted Mode

This section describes features and limitations, PAM configuration changes and configuration parameter for integrating LDAP-UX with Trusted Mode.

5.2.1 Overview

LDAP-UX Client Services B.03.30 or later supports coexistence with Trusted Mode. This means that local-based accounts can benefit from the Trusted Mode security policies, while LDAP-based accounts benefit from the security policies offered by the LDAP server. This release of LDAP-UX also enables LDAP-based and local-based accounts to be audited on the Trusted Mode.

The coexistence of LDAP-UX and Trusted Mode supports certain security features, but also has limitations and usage requirements that you need to be aware of. For detailed information, see "Features and limitations" (page 138).

5.2.2 Features and limitations

This subsection describes features and limitations of integrating LDAP-UX with Trusted Mode.

5.2.2.1 Auditing

Integrating LDAP-UX with Trusted Mode enables accounts stored in the LDAP directory to log in to a local host and to be audited on the Trusted Mode. The following describes the auditing features and limitations. To use these security features, you must enable the audit subsystem on the Trusted Mode local host:

- Auditing of both LDAP-based and local-based (`/etc/passwd`) accounts is possible. By default, auditing is disabled for all LDAP-based accounts. However, you can use the `audusr` (option `-a` or `-d`) command to alter the auditing flag for individual LDAP-based account.
- For LDAP-based accounts that are not yet known to the system, you can configure an initial setting for the auditing flag. You can configure this flag such that when an account becomes known to the system for the first time, auditing for that account is immediately enabled or disabled. This flag is defined as the `initial_ts_auditing` parameter in the `/etc/opt/ldapux/ldapux_client.conf` file.
- You must manage Trusted Mode attributes for all accounts on each host. Trusted Mode attributes for LDAP-based accounts are not stored in the LDAP directory server. For example, enabling auditing for an account on host A does not enable auditing on host B.
- Audit IDs for LDAP-based accounts are unique on each system. Audit IDs are not synchronized across hosts running in the Trusted Mode.
- When an LDAP-based account name is changed, a new audit ID is generated on each host that the account is newly used on. The `initial_ts_auditing` flag is reset to the default value defined in the `/etc/opt/ldapux/ldapux_client.conf` file.
- When an account is deleted from LDAP, the audit information for that account is not removed from the local system. If that account is re-used, the audit information from the previous account is re-used. You can choose to manually remove entries from the Trusted Mode database by removing the appropriate file under the `/tcb/files/auth/...` directory, where "..." defines the directory name based on the first character of the account name.
- You can use the `audisp` command to display information about LDAP-based accounts. However, if an LDAP-based account has never logged in to the system (via telnet, rlogin, and so on), the `audisp -u <username>` command displays the message like "`audisp: all specified users names are invalid.`"

5.2.2.2 Password and account policies

The primary goal of integrating Trusted Mode policies and those policies enforced by an LDAP server is coexistence. This means that Trusted Mode policies are not enforced on LDAP-based accounts, and LDAP server policies are not enforced on local-based accounts. The password and account policies and limitations are described as followings:

- Accounts stored and authenticated through the LDAP directory adhere to the security policies of the directory server being used. These policies are specific to the brand and version of the directory server product deloyed. Examples of these policies include password expiration, password syntax checking, and account expiration. No policies of the HP-UX Trusted Mode product apply to accounts stored in the LDAP server.
- When you integrate LDAP-UX on an HP-UX system with the HP-UX Directory Server or Redhat Directory Server, if an LDAP-based user attempts to log in to the system, but provides the incorrect password multiple times in a row (the default is three times in a row), Trusted Mode attempts to lock the account. However, the Trusted Mode attributes do not impact LDAP-based accounts. So, if the user eventually provides the correct password, he or she can log in.

5.2.2.3 PAM configuration file

- If you integrate LDAP-UX Client Services with the HP-UX Directory Server or Redhat Directory Server, you must define the PAM_LDAP library before the `pam_unix` library in the `/etc/pam.conf` file for all services. You must set the control flag for both PAM_LDAP and PAM_UNIT libraries to `required` under session management. For the proper configuration, see "Sample /etc/pam.ldap.trusted file configured by setup" (page 353) .
- If you integrate LDAP-UX Client Services with the Windows Server 2003 R2/2008 Active Directory Server, you must define the `pam_krb5` library before the `pam_unix` library in the `/etc/pam.conf` file for all services. In addition, the control flag for both `pam_krb5` and `pam_unix`libraries must be set to `required` for `Session management`. For the proper configuration, see the *LDAP-UX Client Services B.05.00 with Microsoft Windows Active Directory Server Administrator's Guide*.

5.2.2.4 Others

- The `authck -d` command removes the `/tcb/files/auth/...` files created for LDAP-based accounts. When the LDAP-based account logs into the system again, a new `/tcb/files/auth/...` file with new audit ID is recreated. Therfore, it is not recommended to run the `authck -d` command when you configure LDAP-UX with Trusted Mode.
- You cannot use the Trusted Mode management subsystem in SAM to manage LDAP-based accounts.
- The LDAP repository and `/etc/passwd` repository must not contain accounts with the same login name or account number.
- Except for the audit flag, you cannot modify other Trusted Mode properties/policies for LDAP-based accounts. For example, attempting to lock an LDAP-based account by modifying the Trusted Mode field for that user does not prevent that account from logging in to the host. Instead, you must disable the account on the LDAP server itself. No runtime warning will be given that the local locking of the account has no effect. It is important that all system administrators are properly trained, so that administrative locks on accounts have the desired effect.

5.2.3 Configuration parameter

LDAP-UX Client Services provides one configuration parameter, `initial_ts_auditing`, available for you to configure the initial auditing setting for the LDAP-based account. This parameter is defined in the `/etc/opt/ldapux/ldapux_client.conf` file.

5.3 PAM_AUTHZ login authorization

The Pluggable Authentication Module (PAM) is an industry standard authentication framework that is supplied as an integrated part of the HP-UX system. PAM gives system administrators the flexibility of choosing any authentication service available on the system to perform authentication. The PAM framework also allows new authentication service modules to be plugged in and made available without modifying the PAM enabled applications. The library `/usr/lib/security/libpam_authz.so.1` (and architecture-dependent library paths) provides the access control functionality described in this section. You can add it to your existing `/etc/pam.conf` as shown in "Policy file".

This section assumes you have some knowledge of how to configure PAM libraries in the `/etc/pam.conf` file. For more information about configuring PAM libraries, see the *HP-UX System Administrator's Guide: Security Management*, available at the following location:

www.hp.com/go/hpux-core-docs (click **HP-UX 11i v3**)

The PAM framework, together with the PAM_AUTHZ service module (which is defined in the PAM_AUTHZ library known as `libpam_authz`) supplied with LDAP-UX Client Services, provide support for `Account Management` services. These services allow the administrator to control who can log in to the system based on netgroup information found in the `/etc/passwd` and `/etc/netgroup` files. PAM and PAM_AUTHZ can also be configured to utilize LDAP-UX Client Services to retrieve the information from a LDAP directory server to perform access of authorization.

NOTE: Beginning with version 5.0 of the product, LDAP-UX Client Services supports integrated compat mode to control which users are visible on a host; user accounts are referenced by netgroups specified in the `/etc/passwd` file. For more information, see "Enabling integrated Compat Mode to control name services and user logins" (page 104)

Starting with LDAP-UX Client Services B.04.00, PAM_AUTHZ has been enhanced to provide administrators a simple security configuration file to set up a local access policy to better meet their need in the organization. PAM_AUTHZ uses the access policy to determine which users are allowed to log in to the system. A policy specifies which groups, LDAP groups, users or other access control objects (such as objects defined by LDAP search filters) are allowed to log in to the system. This flexibility enables you to allow or deny access to a host or application based on a user's membership in a group, or role within a organization. For example, PAM and PAM_AUTHZ can define an access rule that utilizes a LDAP directory server to state that if '`userA`' works for manager '`Sam`' then the criteria is met. When the rule is evaluated, a request would be sent to the LDAP directory and if the attributes were found, the user could be granted or denied access.

NOTE: For information about other means for controlling access to the system, see Section 2.5.6 (page 106).

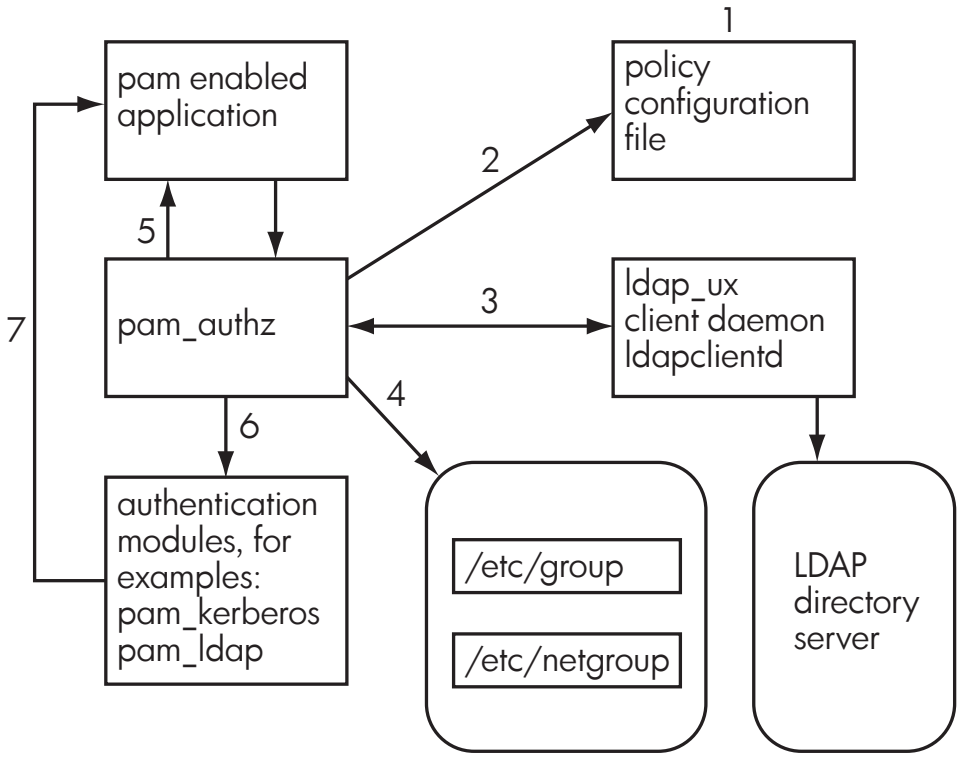
5.3.1 Policy and access rules

Access rules are the basic elements of access control. Administrators create access rules that restrict or permit a user's access permission. A policy is the collection of these different sets of access rules in a given order. This consolidated list of rules defines the overall access strategy of a local client machine. PAM_AUTHZ enables administrators to create an access policy by defining different types of access rules and to save the policy in a file.

5.3.2 How login authorization works

The system administrator can define the access rules and store them in an access policy file. PAM_AUTHZ uses these access rules defined in the policy file to control the login authorization.

Figure 5-1 PAM_AUTHZ environment



The following describes the policy validation processed by PAM_AUTHZ for the user login authorization shown in "PAM_AUTHZ environment" (page 141):

PAM_AUTHZ environment

1. The administrator defines access rules and saves them in a local access policy configuration file.

2. PAM_AUTHZ service module receives an authorization request from PAM framework. It processes all the access rules stored in the access policy configuration file.

3. If a rule indicates that the required information is stored in a LDAP server, PAM_AUTHZ constructs a request message and sends to the LDAP client daemon, `ldapclientd`. The LDAP client daemon performs the actual LDAP query and returns the result to PAM_AUTHZ. Then the access rule is evaluated and the final access right is returned.

4. If a rule indicates that the required information is in the UNIX files. PAM_AUTHZ retrieves user's information from `/etc/passwd`, `/etc/group` or `/etc/netgroup` file through `getpwname()` or `getgrname()` system calls. Then the rule is evaluated and the final access right is returned.

5. PAM_AUTHZ returns the corresponding pam result to PAM framework. The decision is returned to the application which called the PAM API.

6. If the user has the permission to log in, then the decision is returned to the next PAM service module that is configured in the `pam.conf` file, such as PAM_LDAP or PAM_KERBEROS. If the access rule passed but is assigned the `required` action type, then PAM_AUTHZ continues and evaluates the next access policy rule. If the access rule failed and is assigned the `required` action, or if processing reaches the end of the rules (after they all failed), then login is denied.

7. The PAM service module returns the authentication result to the application which called the PAM API.

5.3.3 PAM_AUTHZ supports security policy enforcement

PAM_AUTHZ supports enforcement of account and password policies, stored in an LDAP directory server. This feature works with secure shell (ssh), r-commands with rhost enabled where authentication is not performed by PAM (Pluggable Authentication Module) subsystem, but is performed by the command itself.

For more information on how to configure access rules in the access policy configuration file, set global policy access permissions, and configure the `pam.conf` file for security policy enforcement when using SSH key-pairs or r-commands, see Section 5.3.10 (page 153).

5.3.3.1 Authentication using LDAP

The PAM framework is pluggable, the backend support for PAM's `Authentication`, `Account Management`, `Session Management` and `Password Management` services can be directed to an LDAP directory server. The LDAP-UX Client Services are plugged into the PAM framework by specifying the PAM_LDAP library, `libpam_ldap`, in the `/etc/pam.conf` configuration file. When the PAM_LDAP functions are invoked, the UNIX identity is translated into the distinguished name of an entry in the directory server that represents that user. To perform authentication, PAM_LDAP attempts to bind to the directory server as that identity. If the LDAP bind operation succeeds, then PAM_LDAP will return success to the PAM authentication subsystem.

When PAM_LDAP performs the LDAP bind operation, the LDAP server performs authentication of the user as well as determines if the LDAP account and password policy has passed. If the account is locked, the LDAP bind will fail. If the user's password has expired, the LDAP bind operation will return an error. An LDAP bind operation performs both `authentication` and `account management` operations.

5.3.3.2 Authentication with secure shell (ssh) and r-commands

For LDAP-UX B.04.00 or earlier versions, a user defined in an LDAP directory who tries to log on to a UNIX system using ssh key-pairs or the rhost enabled r-command will always be able to log in even if this user's account has been locked or password has expired. These applications and commands do not need to call the PAM (Pluggable Authentication Module) authentication functions, but perform their own authentication instead. When this occurs, the LDAP bind operation is never performed. Thus, the LDAP directory server is never given the opportunity to perform security policy enforcement.

LDAP-UX Client Services B.04.10 or later provides PAM_AUTHZ features to support enforcement of account and password policies, stored in an LDAP directory server, for applications/commands (such as ssh or r-command) where authentication is not performed by the PAM subsystem, but is performed by the command itself.

5.3.4 Policy file

The system administrator can define a local access policy that can be stored in an access policy file. The default access policy file is `/etc/opt/ldapux/pam_authz.policy`, but it can be stored in an alternate location by setting the `policy` option in `pam.conf`. The PAM_AUTHZ service module uses this local policy file to process the access rules and to control the login authorization. Any service that loads the `libpam_authz.1` library will also load this file. The access policy file location is set per-service in `pam.conf`, so access rules can be customized for each service . For example:

```
login   auth    required libpam_authz.so.1 policy=/etc/opt/ldapux/login.policy
ftp     auth    required libpam_authz.so.1 policy=/etc/opt/ldapux/ftp.policy
```

LDAP-UX Client Services provides a sample configuration file, `/etc/opt/ldapux/pam_authz.policy.template`. This sample file shows you how to configure the policy file to work with PAM_AUTHZ. You can copy this sample file and edit it using the correct syntax to specify the access rules you wish to authorize or exclude from authorization. For detailed information on how to construct an access rule in the policy file, see Section 5.3.7 (page 146).

NOTE: By default, the `allow:unix_local_user` access rule in the `/etc/opt/ldapux/pam_authz.policy.template` file is enabled.

5.3.5 Policy validator

PAM_AUTHZ works as a policy validator. Once it receives a PAM request, it starts to process the access rules defined in `pam_authz.policy`. It validates and determines the user's login authorization based on the user's login name and the information it retrieves from various name services. The result is then returned to the PAM framework.

PAM_AUTHZ processes access rules in the order they are defined in the access policy file. It stops processing the access rules when any one of the access rules is evaluated to be true (match). That rule is called the "authoritative" rule. If any access rule is evaluated to be false (no match), the rule is skipped. If any access rule is evaluated to be true (match) but has the action `required` assigned to it, then access rule processing continues with the next rule. An access rule that has the action `required` assigned to it that evaluates to false (no match) will cause processing to end and the user is restricted from login. If all access rules in the policy file have been evaluated but the user's access right cannot be determined, the user is restricted from login.

NOTE:

- If the user's login name is root or UID is 0, PAM_AUTHZ does not process the access rules defined in the access policy file. The root user is always granted login access.
- The default `<action>` of PAM_AUTHZ is `deny` if no authoritative rule is found.

The following describes situations where PAM_AUTHZ skips an access rule and does not process it:

- An access rule contains the wrong syntax.
- PAM_AUTHZ processes the `ldap_filter` and `ldap_group` types of access rules by querying the LDAP directory server through `ldapclientd` daemon. If LDAP-UX Client Services is not running, PAM_AUTHZ skips all the `ldap_filter` and `ldap_group` types of rules.

5.3.5.1 An example of access rule evaluation

The following shows an example of an access policy file:

```
allow:unix_user:user1,user2,user3,user4
required:ldap_filter:(status=active)
allow:unix_group:group1,group2
deny:unix_group:group11,group12
allow:netgroup:netgroup1,netgroup2
allow::ldap_group:ldapgroup1,ldapgroup2
allow:ldap_filter:(&(manager=Joeh) (department=marketing)(hostname=$[HOSTNAME]))
```

PAM_AUTHZ processes access rules in the order they are defined in the access policy file. It stops evaluating the access rules when any one of the access rule is matched, unless that rule has the action `required` assigned. In the preceding example, if the `user2` user attempts to log in, it matches one of the user names in the first access rule, PAM_AUTHZ stops evaluating the rest of the access rules and allows the `user2` user to log in. For another example, `user5` attempts to log in and this user is only a member of `ldapgroup2`. PAM_AUTHZ validates user5's login access and when the fifth access rule is evaluated to be true, `user5` is granted the login access.

Now assume that the `user6` user has the attribute `status` set to `active`, reports to `Joeh`, the user's job is related to `marketing` and has a `hostname` attribute with the returned value `HostSrv` in his/her user entry in the LDAP directory. PAM_AUTHZ starts to validate user6's login access by evaluating all the access rules defined in the access policy file. The second rule is evaluated to be true, but since the action assigned to this rule is `required`, processing continues with the next rule. The sixth access rule is evaluated to be true, and the `user6` is allowed to log in to the host, `HostSrv`.

5.3.6 Dynamic variable support

Dynamic variable support is a method by which an access rule can be defined where part or all of the policy criteria will be determined at the time the rule is evaluated. For example, the name of the computer from which the user attempts to logon can be substituted into the access rule to be evaluated. See Section 5.3.9 (page 151) for more information on how to define an access rule using dynamic variable support.

5.3.7 Constructing an access rule in the access policy file

In the access policy file, an access rule consists of three fields as follows:

<action>:<type>:<object>

All fields are mandatory except for the `<object>` field when `passwd_compat`, `unix_local_user`, or `Other` is specified in the `<type>` field. If any field is missing or contains the incorrect syntax, the access rule is considered to be invalid and is ignored by PAM_AUTHZ.

These fields have the following limitations:

- No leading or trailing empty space is allowed in a field
- Fields are separated by a separator, :
- No leading or trailing empty space is allowed in a separator
- An access rule is terminated by a carriage return

5.3.7.1 Fields in an access rule

Table 5-1 shows a summary on all possible values and syntax of an access rule:

Table 5-1 Field syntax in an access rule

<action>	<type>	<object>
`deny, allow,required, <pam_code>`	`unix_user`	A list of user name. It can be the multi-valued field. Each value is a character string that is separated by a separator "," (ASCII 2C HEX). Example: user1, user2, user3
`deny, allow, required, <pam_code>`	`unix_local_user`	No value is required.
`deny, allow, required, <pam_code>`	`unix_group`	A list of group name. It can be the multi-valued field. Each value is a character string that is separated by a separator `","` (ASCII 2C HEX). Example: group1, group2, group3
`required, <pam_code>`	`passwd_compat`	No value is required.
`deny, allow, required, <pam_code>`	`netgroup`	A list of netgroup name. It can be the multi-valued field. Each value is a character string that is separated by a separator `","(ASCII 2C HEX).` Example: netgroup1, netgroup2, netgroup3
`deny, allow, required, <pam_code>`	`ldap_group`	It is the Distinguished name of an LDAP group with `groupofnames` objectclass or `groupofuniquenames` objectclass. It is a single-valued field. No separator is required. The syntax of DN is defined in RFC2253. Example: cn=ldapgroup1,cn=groups,dc=mydomain,dc=com
`deny, allow, required, <pam_code>`	`ldap_filter`	It is a single search descriptor that specifies one or more (attribute=value) or (attribute=$[variable_name]) pairs. $[variable_name] is a dynamic variable. It is a single value field. Only one search filter is allowed. No separator is required. The syntax of DN is defined in RFC2254. Example: (&(manager=Joeh)(department=sales)(hostcontrol=$[HOSTNAME]))

Table 5-1 Field syntax in an access rule *(continued)*

<action>	<type>	<object>
deny, allow, required, <pam_code>	other	No value is required.
status	<library_name> The valid value for this field can be rhds or ads.	<function_name> Specifies the function name in <library_name> that is called to evaluate certain policy settings of the login user. Example: status:rhds:check_ads_polcy See the "Account and Password Security Policy Enforcement " section for details.

The following describes three fields defined in an access rule in details:

<action> This field defines a user's final access permission if an access rule is evaluated to be true. Valid entries can be `allow`, `deny`, `required`, and PAM return codes. `Allow`, `deny`, and `required` are character strings and the value itself is not case sensitive. In additional to the general return codes, `allow` and `deny`, LDAP-UX Client Services B.04.10 or later PAM_AUTHZ supports the meaningful PAM return codes to the application which called the PAM API. PAM_AUTHZ does not evaluate an access rule if no option is defined or if the `action` field contains an invalid string.

`<action>` field can be one of following values:

`allow`

This option indicates that a user is granted the login authorization.

`deny`

This option indicates that a user is denied the login authorization.

`required`

This option indicates that a user is denied the login authorization if the rule evaluates to false, or that processing should continue to the next rule if the rule evaluates to true.

`<pam_code>`

One of the following meaningful PAM return codes can be specified in the `<action>` field, the PAM return codes are character strings:

- `PAM_SUCCESS`
- `PAM_PERM_DENIED`
- `PAM_MAXTRIES`
- `PAM_AUTH_ERR`
- `PAM_NEW_AUTHTOK_REQD`
- `PAM_AUTHTOKEN_REQD`
- `PAM_CRED_INSUFFICIENT`
- `PAM_AUTHINFO_UNAVAIL`
- `PAM_USER_UNKNOWN`
- `PAM_ACCT_EXPIRED`
- `PAM_AUTHTOK_EXPIRED`

For example, if the PAM_AUTHZ policy rule indicates that an account has been locked out or a password has expired, PAM_AUTHZ can return an appropriate PAM error code instead of a general deny error code.

<type> The value in this field represents the type of access rule. It defines what kinds of user information that PAM_AUTHZ needs to look for. The value also helps to determine the correct syntax in the following <object> field.

The following describes the valid values for this field:

unix_user, unix_local_user, unix_group, netgroup, ldap_group

Rules that have one of these specified as the <type> field are defining a static list access rule. For this rule, the <object> field is specified as a predefined list of identifiers. The identifiers are matched directly with data in the login request. This <type> field specifies where PAM_AUTHZ will look to determine if the login field is present in the appropriate data store, such as /etc/passwd, /etc/group, etc. If the login field is found, the rule is evaluated to be true. The final access right is determined by the <action> field. See the "Static List Access Rule" section for details.

passwd_compat

Control the access permission using NIS-style escapes in /etc/passwd. This is identical to the default behavior of PAM_AUTHZ when there is no access policy file present. The passwd_compat type supports only status or required in the action field, and anything specified in the <object> field is ignored.

other

PAM_AUTHZ ignores any access rules defined in the <object> field. The access rule is evaluated to be true immediately. For example,

```
allow:other
```

In the above example, all users are granted the login access to the machine. The primary usage of this type of rule is to toggle PAM_AUTHZ default <action>.

ldap_filter

In a role based access management, permission to access a resource can be controlled based on the user's role such as sales force, technical support or subscriber status and are typically defined by common business attributes of users based on company policies. The same concept is applied to the ldap_filter access rule. A search filter is defined in <object> field. A search filter consists of one or more (attribute=value) pairs. If the user entry is successfully retrieved from a directory server by using the search filter, the access rule is considered to be true. Examples of ldap_filter type of access rule are as follows:

```
allow:ldap_filter:(&(manager=paulw)(business
category=marketing))
```

In the above example, if a user reports to paulw and the user's job is related to marketing, then the user is granted the login access. The rule structure is very flexible about how to define access for certain groups of users.

```
PAM_ACCT_EXPIRED:ldap_filter:(nsAccountLock=TRUE)
```

In the above example, if a user account has been locked out and this access rule is evaluated to be true, the PAM_ACCT_EXPIRED code is returned by PAM_AUTHZ.

In LDAP-UX Client Services B.04.10 or later, PAM_AUTHZ supports dynamic variable in the ldap_filter type of the access rule. A search filter can consist of one or more (attribute=$[function_name]) pairs and is defined in the <object> field. The [function_name] is called and the return value is substituted into the search filter. Then the search filter is processed the same as the example above. For detailed information about dynamic variable support, see Section 5.3.9 (page 151).

status

When `status` is specified as the `<action>` field, this defines a rule that is evaluated to perform account and password policy enforcement. This access rule defines a library, in the `<library_name>` field to be loaded, and a function in the `<function_name>` field that specifies a function to be invoked to perform policy evaluation for a particular directory server. See Section 5.3.10.1 (page 153) for detailed information on the supported values and usage of this access rule.

<object> The values in this field define the policy criteria that PAM_AUTHZ uses to validate with the login name. The values in this field are dependent on the option that is stated in the `<type>` field.

5.3.8 Static list access rule

When the value in the <type> field is one of unix_user, unix_group, netgroup, ldap_group, the rule is evaluated using a list of predefined values in the <object> field. Based on the value in the <type> field, PAM_AUTHZ will call the appropriate service to determine if the item requested is present. If the requested information is found then the rule is evaluated to be true.

The following describes these values for this field in details:

unix_user

This option indicates that an administrator wants to control the login access by examining a user's login name with a list of predefined users. If the login name matches one of the user names in the list, the authorization statement is evaluated to be true. The final access right is determined by evaluating the <action> field. An example of a unix_user type of access rule is as follows:

```
allow:unix_user:myuser1,myuser2,myuser3
```

If a myuser3 user attempts to log in, the above access rule is evaluated to be true and the user is granted login access.

unix_local_user

This option indicates that an administrator wants to control the login access by examining a local user's login name with a list of user's accounts in the /etc/passwd file. If the login name matches one of the user accounts defined in /etc/passwd, the authorization statement is evaluated to be true. Otherwise, the rule is skipped. An example of a unix_local_user type of access rule is as follows:

```
allow:unix_local_user
```

As an example, if a user account, myuser5, is defined in /etc/password, the above access rule is evaluated to be true and this user myuser5 is granted permission to log in to the local host.

unix_group

This option specifies that an administrator wants to control the login access right using the user's group membership. You can specify a list of group name in the <object> field. PAM_AUTH retrieves the group information of each listed group by querying the name services specified in nsswitch.conf. That means the group entries may come from any sources (files, nis, LDAP, etc). If the login user belongs to any groups in the list, the access rule is evaluated to be true. Otherwise, the rule is skipped. An example of a unix_group access rule is shown as follows:

```
deny:unix_group:myunixgroup10,myunixgroup11,myunixgroup12
```

A user tries to log in and he is a member of myunixgroup12. The rule is evaluated to be true and the <action> is applied. The user is restricted from access to the machine even with a valid password.

netgroup

This option specifies that the access permission is determined by the user's netgroup membership. You must specify a list of netgroup name in the <object> field. If the user is a member of one of the netgroups specified in the netgroup list, then the access rule is evaluated to be true. PAM_AUTH obtains the netgroup information by querying the name services specified in nsswitch.conf. For example:

```
allow:netgroup:netgroup1,netgroup2,netgroup3
```

A user tries to log in and he belongs to netgroup1. The above access rule is evaluated to be true. The user is granted login access.

NOTE: Beginning with version 5.0 of the product, LDAP-UX Client Services supports integrated compat mode to control which users are visible on a host, where the user accounts are referenced by netgroups specified in the `/etc/passwd` file. For more information, see "Enabling integrated Compat Mode to control name services and user logins" (page 104)

ldap_group

This option specifies that an access rule is based on the non-POSIXGroup membership. PAM_AUTHZ supports LDAP group with `groupOfNames` or `groupOfUniqueNames`objectclass. A list of `ldap_group` names is specified in the `<object>` field. The group membership information is stored in the LDAP directory server. An example of a `ldap_group` type of access rule is as follows:

```
deny:ldap_group:engineering_ldapgroup,support_ldapgroup,epartner_ldapgroup
```

PAM_AUTHZ retrieves group membership of each listed group from the directory server through LDAP-UX client services. Then, it examines if the user's distinguished name (DN) matches any value in the `member` or `uniquemember` attribute.

5.3.9 Dynamic variable access rule

PAM_AUTHZ supports dynamic variables in the `ldap_filter` type of the access rule. A dynamic variable is defined in `<object>` (LDAP search filter) field, it can consist of one or more (attribute=$[variable_name]) pairs. The syntax of an access rule with the dynamic variable is:

<action>:ldap_filter:(attribute=$[variable_name])

For example, if an administrator has an attribute named `hostControl` defined in the directory, and wants to use this attribute to define which host a user can log on to. He may add the following access rule in the access policy file:

```
allow:ldap_filter:(hostControl= hostA)
```

Where `hostA` is the value for the local host that the user must be granted access. If a user, `John`, has a `hostControl` attribute in his user entry in the LDAP directory and the value is `hostA`, then the access rule is evaluated to be true and this user is allowed to log in to the host, `hostA`.

In the above example, a dynamic variable `HOSTNAME` can be used. The previous access rule can be re-defined as follows:

```
allow: ldap_filter: (hostControl=$[HOSTNAME])
```

where `$[HOSTNAME]` represents a dynamic variable function which will be called to retrieve the local host name information. PAM_AUTHZ will then substitute its return value to the search filter.

5.3.9.1 Supported functions for dynamic variables

In LDAP-UX Client Services, PAM_AUTHZ provides the following default dynamic variable functions in the `libpolicy_commonauthz` library. These functions can be used as dynamic variables specified in the `ldap_filter` type of access rules::

HOSTNAME Returns the host name of the local system from which the user attempts to log on. For example, hostA.

HOSTNAMEWD Returns the fully qualified host name of the local system from which the user attempts to log on. For example, hostA.hp.com.

HOSTIP Returns the IP address of the local system from which the user attempts to log on. For example, 12.10.2.105.

TERMINAL Returns the terminal type of the computer from which the user attempts to log on. For example, /dev/pts/0.

Some applications (such as ssh or remsh) do not pass the terminal dynamic variable value to PAM_AUTHZ.

TIMEOFTHEDAY Returns the current time of the computer system from which the user attempts to log on. For example, 20061015125535Z represents October 15, 2006 at 12:55 and 35 seconds GMT. `TIMEOFTHEDAY` follows the "UTC Time" syntax as described by RFC4517.

SERVICE Returns the name of the PAM service from which the user attempts to access. For example, common PAM service names include ftp, login, telnet.

RHOSTIP Returns the IP address of the remote host system from which the user starts the PAM enabled application, such as telnet.

RHOSTNAME Returns the name of the remote host system from which the user starts the PAM enabled application, such as telnet.

RHOSTNAMEWD Returns the name of the fully qualified remote host system from which the user starts the PAM enabled application, such as telnet.

5.3.9.2 Examples

The following shows a sample access rule in the access policy file:

```
allow:ldap_filter:(WorkstationIP=$[HOSTIP])
```

The above policy rule performs a security policy validation for users stored in the LDAP directory server. If a user, `Mary`, has a `WorkstationIP` attribute in her user entry in the LDAP directory and the value is `1.2.3.200`. If Mary attempts to log in to the host with the IP address, `1.2.3.200`, then the access rule is evaluated to be true and this user is granted login access.

5.3.10 Security policy enforcement with secure shell (ssh) or r-commands

PAM_AUTHZ has a limited ability to perform account and password security policy enforcement without requiring LDAP-based authentication. This section provides information on how to configure the security policy enforcement access rule, set up access permissions for global policy attributes and configure PAM configuration file to support enforcement of account and password policies, stored in an LDAP directory server, for applications such as ssh key-pair and r-commands with rhost enabled.

This feature is designed to support applications such as secure shell (ssh) and the r-commands (rlogin, rcp, etc..) with .rhost enabled. With these applications, authentication is not performed by the PAM (Pluggable Authentication Module) subsystem, but is performed by the command itself. In these applications, when authentication is not performed by PAM, the LDAP directory server is not given the opportunity to provide security policy enforcement, which normally occurs during the LDAP authentication process.

To configure and use this feature for ssh key-pair or r-commands, you must perform the following tasks:

- Set security policy enforcement access rule in the access policy file. See Section 5.3.10.1 (page 153) for details.
- Set access permissions for global policy attributes. See Section 5.3.10.2 (page 154) for details.
- Configure the PAM_AUTHZ library and the `rcommand` option in the `/etc/pam.conf` file for the `sshd` and `rcomds` services under the account management section. See Section 5.3.10.3 (page 155) and "Sample /etc/pam.conf file for security policy enforcement" (page 357) for details.

5.3.10.1 Security policy enforcement access rule

Specifying `status` in the `<action>` field of a `pam_authz.policy` access rule triggers use of the account and password security policy enforcement rule. When this rule is evaluated, PAM_AUTHZ will call the `<function_name>` in the library specified by the `<library_name>` field. PAM_AUTHZ returns the value which is one of the PAM return codes described in Section 5.3.10.5 (page 155) below.

This access rule consists of the following three fields:

<action>:<library_name>:<function_name>

Fields in the access rule:

The following describes each field of the above access rule:

action When the `status` option is specified, PAM_AUTHZ returns whatever **<function_name>** in the **<library_name>** returns, which is one of the PAM return codes.

library_name This field specifies the name of the library to be loaded that supports the account and password policies for a particular directory server.

The following describes the valid values for this field:

- `rhds`: If this option is specified, PAM_AUTHZ loads the `/opt/ldapux/lib/libpolicy_rhds` library to process security policy configuration and examine the user's security policy status attributes, stored in the HP-UX Directory Server or Redhat Directory Server.
- `ads`: If this option specified, PAM_AUHZ loads `/opt/ldapux/lib/libpolicy_ads` library to process security policy configuration and examine the user's security policy status attributes, stored in the Windows Server 2003 R2/2008 Active Directory Server.

function_name This field defines the function name in the specified **<library_name>** that PAM_AUTHZ uses to evaluate certain security policy settings with the login user.

The following describes the valid entries for this field:

- `check_rhds_policy`: If this option is specified, PAM_AUTHZ evaluates all the necessary account and password policies settings, stored in the HP-UX Directory Server or Redhat Directory Server, for the login user.
- `check_ads_policy`: If this option is specified, PAM_AUTHZ evaluates all the necessary account and password policies settings, stored in the Windows Server 2003 R2/2008 Active Directory, for the login user.

NOTE: If the `status:rhds:check_ads_policy` access rule is configured in the access policy file, you must perform the following tasks:

- Define the `allow:unix_local_user` access rule in the access policy file to allow the local user to log in.
- Since the `status:rhds:check_ads_policy` access rule is guaranteed to match and return a PAM return code. It is highly recommended to define the `status:rhds:check_ads_policy` access rule at the end of the access policy file. Otherwise, the access rules that are defined after the `status` access rule will not be evaluated.
- PAM_AUTHZ may display account and password policy attributes in the `syslog` file when the debug option is enabled. You can take proper action to protect the `syslog` file. For example, set the `syslog` file permissions, so that the file can only be accessed or viewed by the power user.

WARNING! Enabling the debug option in pam.conf might allow hackers to gain additional information that would enable them to crack password security. For example, they could attempt to log in as a super user (su) and discover that a password has expired (observing the super user's behavior, the hackers could determine when he or she is likely to log in next).

5.3.10.1.1 An example of access rules

The following shows an example of the access rules defined in the access policy file when configuring and using security policy enforcement for ssh key-pair or r-commands:

```
allow:unix_local_user
status:rhds:check_ads_policy
```

5.3.10.2 Setting access permissions for global policy attributes

For PAM_AUTHZ to support security policy enforcement with the HP-UX Directory Server or Red Hat Directory server, PAM_AUTHZ needs access to the security policy configuration attributes. These global policy attributes are all defined under cn=config. Only authorized users can access them. If you use the PAM_AUTHZ enhancement to support the account and password policy enforcement, you must configure LDAP-UX with a proxy user and grant this proxy user read and search rights to search for specific attributes under cn=config. The following example ACI gives a proxy user permission to read and search all global policy attributes:

```
aci: (targetattr= "objectclass ||passwordLockout ||passwordUnlock
 ||passwordMaxFailure ||passwordExp ||passwordMustChange
 ||nsslapd-pwpolicy-local")
 (version 3.0; acl "Proxy global security policy attributes read and
 search rights";
```

```
allow (read,search)
(userdn = "ldap:///uid=proxyuser,ou=Special Users,o=hp.com");)
```

For more information about a list of security policy attributes supported by LDAP-UX, see Section 5.3.10.6 (page 156).

5.3.10.3 Configuring the PAM configuration file

If you want to use PAM_AUTHZ to support enforcement of account and password policies stored in your directory server, you must define the PAM_AUTHZ library and the `rcommand` option in the `/etc/pam.conf` file for the `sshd` and `rcomds` services under the account management section. In addition, the control flag for the PAM_AUTHZ library must be set to `required`. See "Sample /etc/pam.conf file for security policy enforcement" (page 357) for proper configuration.

5.3.10.4 Evaluating the directory server security policy

The following is an example of the access rule in the access policy file:

```
status:rhds:check_rhds_policy
```

If the above access rule is specified in the access policy file, the `check_rhds_policy` routine in the `libpolicy_rhds` library is loaded and executed. PAM_AUTHZ constructs a request message that will be used to find the current security policy configuration as well as examine the specific user's security policy status attributes to determine if the user complies with the security policy. PAM_AUTHZ will search for the following information:

- Global policy attributes under cn=config: `passwordLockout`, `passwordUnlock`, `passwordMaxFailure`, `passwordExp`, `passwordMustChange`, `nsslapdpwpolicy-local`.
- User specific policy attributes: `accountUnlockTime`, `passwordExpirationTime`, `pwdPolicySubEntry`, `passwordRetryCount`, `nsAccountLock`.
- If fine-grained policy is turned on and the sub-tree policy for this user has been configured,, then LDAP-UX searches for password policy attributes at the subtree and user level: `passwordLockout`, `passwordUnlock`, `passwordMaxFailure`, `passwordExp`, `passwordMustChange`.

PAM_AUTHZ performs the following major functionality by evaluating the necessary security policy settings and returns the corresponding PAM return code to the applications/commands which called the PAM API.

- Check whether an account is inactivated or not.
- Check whether an account is locked or not.
- Check whether the password has expired or not.

5.3.10.5 PAM return codes

If the `status:rhds:check_rhds_policy` access rule is specified in the access policy file for HP-UX Directory Server or Redhat Directory Server, PAM_AUTHZ evaluates the necessary security policy settings and returns the possible PAM return codes as follows:

PAM_USER_UNKNOWN	The code returned if the user is not found in the Directory Server or if there is any internal errors (such as an error returned by the server) to find the user's policy attributes.
PAM_ACCT_EXPIRED	The code returned if the user account is inactive.
PAM_ACCT_EXPIRED	The code returned if the user account has been locked out.
PAM_NEW_AUTHTOK_REQD	The code returned if the user's password has expired.
PAM_SUCCESS	The code returned if the user account is active and not locked, and user's password has not expired.

5.3.10.6 Directory server security policies

Global security attributes

In the HP-UX Directory Server or Redhat Directory Server, numerous attributes are used to define the security policies. To support account and password security policy enforcement, PAM_AUTHZ is enhanced to support the global administrative security attributes listed in Table 5-2.

These attributes are used to define the policy rules and are all defined under cn=config. Only authorized users can access them. If you use the PAM_AUTHZ enhancement to support the account and password policy enforcement, you must configure LDAP-UX with a proxy user and grant this proxy user read and search rights to search cn=config.

Table 5-2 Global security attributes

Attribute	Description
`passwordLockout`	This boolean attribute indicates whether users will be locked out of the directory after a given number of failed bind attempts. By default, users will not be locked out of the directory after a series of failed bind attempts.
`passwordUnlock`	This boolean attribute indicates whether users will be locked out of the directory for a specified amount of time or until the password is reset after an account lockout. If the `passwordUnlock` attribute is disabled and the `accountUnlockTime` attribute has a value of 0, then the account will be locked indefinitely.
`passwordMaxFailure`	This integer attribute indicates the maximum number of password failures after which a user will be locked out of the directory. By default, account lockout is disabled.
`passwordExp`	This boolean attribute indicates whether user passwords will expire after a given number of seconds. By default, user passwords do not expire. If this attribute is enabled, you can use the `passwordMaxAge` variable to set the number of seconds after which the password will expire.
`passwordMustChange`	This boolean attribute indicates whether users must change their passwords when they first bind to the Directory Server or when the password has been reset by the Directory Manager.
`nsslapd-pwpolicy-local`	Turns fine-grained (subtree and user level) password policy on and off. If this attribute has a value off, all entries (except for cn=Directory Manager) in the directory will be subjected to the global password policy, the server will ignore any defined subtree and user level password policy. If this attribute has a value on, the server will check for password policies at the subtree and user level and enforce those policies.

Security policy status attributes

PAM_AUTHZ supports a list of attributes that hold general security policy status information for a particular user in the directory server. These attributes are listed in Table 5-3.

Table 5-3 Security policy status attributes

Attribute	Description
`nsAccountLock`	This boolean attribute indicates whether an account is locked or not. If this attributes does not exist, the account is considered unlocked.
`passwordRetryCount`	This integer attribute specifies the number of consecutive failed attempts at entering the correct user password.

Table 5-3 Security policy status attributes *(continued)*

`passwordExpirationTime`	This string attribute defines a date and time when a password is considered expired. The data and time are specified using the "Generalize Time" syntax as referenced in RFC 2252 and specified by the ISO x.208 standard. It uses the format YYYYMMDDHHMMSSTZ, where YYYY= 4 difit year, MM= 2 digit month, DD=2 digit day, HH=2 digit hour, MM=2 digit minute, SS=2 digit second and TZ=tme zone. In LDAP directory servers, they use the GMT time zone which is represented with the letter Z for Zone time. For example, 20060215165535Z represents February 15, 2006 at 16:55 and 35 seconds GMT.
`accountUnlockTime`	This string attribute defines a date and time when an account will be unlocked. The value is represented in the Generalized Time syntax described in the "`passwordExpirationTime`" attribute. If the attribute does not exist, the account is considered unlocked (assuming `nsAccountLock` does not also exist).
`pwdpolicysubentry`	This variable defines the location of the new password policy. The location is expressed in the DN format.

5.4 Adding a directory replica

Your LDAP directory contains configuration profiles downloaded by each client system and name service data accessed by each client system. As your environment grows, you may need to add a directory replica to your environment. LDAP-UX can take advantage of replica directory servers and the alternates if one of them fails. Follow these steps to inform LDAP-UX about multiple directory servers:

1. Create and configure your LDAP directory replica. For the HP-UX Directory Server, see the *HP-UX Directory Server deployment guide*.
2. Edit an existing profile and modify the defaultServerList or preferredServerList attribute to specify a replica directory server. See Section 5.12 (page 183).

   See Appendix B (page 349) for a description of the defaultServerList or preferredServer attribute.
3. On all clients that are to use the replica server, edit the start-up file, `/etc/opt/ldapux/ldapux_client.conf`, to refer to the replica host. Modify the LDAP_HOSTPORT line to specify the replica server.
4. After modifying an existing profile, each client that regularly downloads its profile automatically will get the changes as scheduled. SeeSection 2.5.8 (page 113).

NOTE: Client systems using an LDAP directory replica may not be able to modify the directory replica. In this case, the `passwd` command will not work on those systems. They can use the `ldappasswd` command described in Section 7.4.2 (page 294).

5.5 Managing users and groups

LDAP-UX Integration supports the new set of non-interactive LDAP command-line tools that allow you to list, add, modify or delete user accounts and groups in an LDAP directory server. These new tools provide capabilities to perform those operations without needing to discover the LDAP server information. Each tool uses the LDAP-UX profile's configuration to discover server information, such as the host name and port number of the LDAP directory server and proper search filters for finding users and groups. Each tool provides command options that enable you to alter these configuration parameters. Using these new tools does not require you to have extensive knowledge of the LDAP schema, protocol and LDAP-UX configuration of each directory server product. These tools performs installation specific data model interpretation, such as converting UID-name based group membership (POSIX-style) to X.500 DN based membership (LDAP-style).

The LDAP User and Group (UG) management tools support the following features:

- Create, modify, delete, or list users and groups in an LDAP directory server.
- Modify user or group password.
- Support attribute mapping for definition of POSIX attributes used when creating or modifying entries.
- Support specification of group membership using X.500-style DN based member attributes.
- Provide customized and default templates for defining new user and group entries, which allows arbitrary data models to be used.
- Support SSL or TLS encryption of data connections to the LDAP directory server if requested.
- Provide the ability to connect to an alternate directory server other than that specified by the LDAP-UX configuration profile.
- Discover programmatically if LDAP-UX is installed, configured and operating properly for a specified service.

The HP System Management Homepage (SMH) Users and Groups interface uses these LDAP UG command line tools to implement the web-based user interface functionality that manages POSIX users and groups in an LDAP directory server. This enables HP-UX system administrators to manage users and groups in an LDAP directory server using SMH UG-LDAP web-based interface on an HP-UX 11i v3 system. The HP System Management Homepage (SMH) product supports the LDAP user and group web-based management feature via HP-UX 11i v3 September, 2007 release.

5.5.1 LDAP user and group command-line tools

The LDAP-UX Integration product supports the following LDAP command-line tools for management of user and group information in an LDAP directory server. These LDAP user and group tools exist in the `/opt/ldapux/bin` directory. For detailed information about tool usage, syntax, options, arguments, environment variables and return codes supported by these tools, see Section 7.3 (page 219), or see the *ldapuglist*(1M), *ldapugadd*(1M), *ldapcfinfo*(1M), *ldapugmod*(1M), and *ldapugdel*(1M) manpages.

Use of the `ldapugadd`, `ldapugmod` and `ldapugdel` tools requires specification of LDAP administrator credentials with sufficient privilege to perform the requested operations in an LDAP directory server. Specification of these credentials can be done through the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables or an interactive prompt (`-P`) option. If the LDAP administrator credential has not been specified using the two previous methods, and if configured, the LDAP-UX administrator credential is used if the user running the tool has sufficient privilege to read the `/etc/opt/ldapux/acred` file.

- **`ldapuglist`**

  You can use the `ldapuglist` tool to display and enumerate POSIX-like account and group entries that reside in an LDAP directory server. The `ldapuglist` provides features that allow users and applications to discover and evaluate account and group information stored

in an LDAP directory server, without requiring extensive knowledge of in-use data models or the methods used to retrieve and evaluate that information in the LDAP directory server.

The `ldapuglist` tool uses the LDAP-UX profile configuration, requiring minimal command line options to discover where to search for user or group information, such as the LDAP directory server host and proper search filters for finding users and groups. The `ldapuglist` tool also uses attribute mapping defined in the profile to translate information to POSIX syntax. By default, `ldapuglist` only displays POSIX-related attributes using RFC 2307 attribute names unless you specifically request an attribute list with the `<attr>` option on the command line. This tool provides command options that enable you to alter these configuration parameters.

- **`ldapugadd`**

  You can use the `ldapugadd` tool to add new POSIX accounts and groups to an LDAP directory server. Because the deployed data model may require user or group attributes beyond that of the standard POISIX attributes, the `ldapugadd` tool uses user and group template files to discover the required data model for the types of entries being created. These templates may define arbitrary data models beyond just the required POSIX attributes. Applications can use `ldpacfinfo` to discover the attributes required by the templates that are not part of the standard POSIX data model. To use `ldapugadd`, you must provide LDAP administrator credentials who have sufficient privilege to perform the user or group add operation in the LDAP directory server.

- **`ldapugmod`**

  The `ldapugmod` tool allows HP-UX administrators to modify existing POSIX accounts or groups in an LDAP directory server. When using extended options, you can use `ldapugmod` to modify arbitrary attributes for user or group entries or you can extend existing user or group entries with the POSIX data model. To use `ldapugmod`, you must provide LDAP administrator credentials that have sufficient privilege to perform the user or group modify operations in the LDAP directory server.

- **`ldapugdel`**

  Use the `ldapugdel` tool to remove POSIX related user or group entries from an LDAP directory server. The `ldapugdel` tool can also remove the POSIX related attributes and object classes from user or group entries, without removing the entire entry itself.

- **`ldapcfinfo`**

  The `ldapcfinfo` tool provides several capabilities used to report LDAP-UX configuration and status. When used specifically with the LDAP UG tools, `ldapcfinfo` can be used to discover LDAP-UX configuration details about required attributes when creating new users or groups to an LDAP directory server.

  The `ldapcfinfo` tool can provide the following information by examining LDAP UG template files, LDAP UG configuration file or the LDAP-UX configuration profile:

  — Determine if the LDAP-UX is properly configured and active.
  — Discover the current LDAP UG configuration defaults, such as home directory and login shell.
  — Discover the distinguished name (DN) of the LDAP-UX configuration profile and the LDAP directory server name which stores that profile.
  — Discover search filter, search base or search scope for a particular name service.
  — Discover the attribute mapping information for a specified name service.
  — Discover the list of available template files for a specific name service when you want to add a new user or group entry to an LDAP directory server.
  — Discover LDAP-UX configuration information about required attributes when creating a new user or group entry.

— Discover the recommended list of attributes that an interactive management tool can consider making available for modification for the specified entry.

The following subsequent sections provide examples on how to use `ldapuglist`, `ldapugadd`, `ldapugmod`, `ldapugdel` and `ldapcfinfo` to display, enumerate, add, modify or delete user accounts and groups in an LDAP directory server.

5.5.2 Listing users

You can use `ldapuglist` to list and enumerate POSIX-like account entries in an LDAP directory server. Below are examples of how to use `ldapuglist` to list user entries.

While use of `LDAP_BINDDN` is not typically required to use `ldapuglist`, the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables can be used to specify the distinguished name (DN) and password of a user with sufficient directory server privilege to display protected attributes. Alternately, you can input LDAP administrator bind identity and credential interactively with a prompt (`-P`) option.

Setting the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables is optional when using `ldapuglist`.

The following commands set the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables:

```
export LDAP_BINDDN = "cn=Jane Admin,ou=admins,dc=example,dc=com"
export LDAP_BINDCRED = "Jane's password"
```

The following commands display an account entry for the user, `mlee`:

```
cd /opt/ldapux/bin
./ldapuglist -t passwd -n mlee
```

The output of the above command is as follows:

```
dn: cn=Mike Lee,ou=people,dc=example,dc=com
cn: Mike Lee
uid: mlee
uidNumber: 900
gidNumber: 2010
loginShell: /usr/bin/sh
homeDirectory: /home/mlee
gecos: mlee,Building-5,555-555-5555
```

The following command displays account entries available in the LDAP directory server:

```
./ldapuglist -t passwd
```

The output of the above command is as follows:

```
dn: cn=Mike Lee,ou=people,dc=example,dc=com
cn: Mike Lee
uid: mlee
uidNumber: 900
gidNumber: 2000
loginShell: /usr/bin/sh
homeDirectory: /home/mlee
gecos: mlee,Building-5,555-555-5555

dn: cn=Michael Sheu,ou=people,dc=example,dc=com
cn: Michale Sheu
uid: msheu
uidNumber: 880
gidNumber: 2010
loginShell: /usr/bin/sh
homeDirectory: /home/msheu
gecos: msheu,Building-8,555-555-5000

dn: cn=Pat Fong,ou=people,dc=example,dc=com
cn: Pat Fong
uid: pfong
```

```
uidNumber:750
gidNumber: 2000
loginShell: /usr/bin/sh
homeDirectory: /home/pfong
gecos: pfong,Building-10,555-552-5000
...
...
```

The following command displays an account entry which contains `uid=tscott`:

```
./ldapuglist -t passwd -m -f "(uid=tscott)"
```

The output is as follows. In this example, the `uidNUmber` attribute has been mapped to `employeeNumber` and the `gecos` attribute has been mapped to `cn`, `l` and `telephoneNumber`. With the `-m` option, the `ldapuglist` tool displays the mapped attribute names as well.

```
dn: cn=Tom Scott,ou=people,dc=example,dc=com
cn[cn]: Tom Scott
uid[uid]: tscott
uidNumber[employeeNumber]: 900
gidNumber[gidNumber]: 2010
loginShell[loginShell]: /usr/bin/sh
homeDirectory[homeDirectory]: /home/tscott
gecos[cn]: Tom Scott
gecos[l]: Building-12
gecos[telephoneNumber]: 555-555-6666
```

5.5.3 Listing groups

You can use `ldapuglist` to list and enumerate POSIX-like group entries in an LDAP directory server. Below are examples of how to use `ldapuglist` to display group entries.

Run the following command to list all the posixGroup entries that Mike Phillips belongs to:

```
cd /opt/ldapux/bin
./ldapuglist -t group -f "(memberUid=mphillips)"
```

The output is as follows:

```
dn: cn=group1,ou=groups,dc=example,dc=com
cn: group1
gidNumber: 550
memberUid: mphillips
memberUid: mlou
memberUid: apierce
memberUid: bjones

dn: cn=group2,ou=groups,dc=example,dc=com
cn: group2
gidNumber: 580
memberUid: vtam
memberUid: ajones
memberUid: mphillips
```

Run the following command to list a regular posixGroup entry which contains `cn=groupA`:

```
./ldapuglist -t group -f "(cn=groupA)"
```

The output is as follows:

```
dn: cn=groupA,ou=groups,dc=example,dc=com
cn: groupA
gidNumber: 620
memberUid: user1
memberUid: user3
memberUid: user5
```

Run the following command to list a regular posixGroup entry for the group name, `groupB`:

```
./ldapuglist -t group -n groupB
```

The output is as follows:

```
dn: cn=groupB,ou=groups,dc=example,dc=com
cn: groupB
gidNumber: 620
memberUid: user1
memberUid: user3
memberUid: user5
```

Command arguments

The following describes the `ldapuglist` options/arguments used in the above examples:

`-t <type>` Specifies the type of entry the `ldapuglist` tool needs to discover and process. `<type>` can be `passwd` or `group`. The `passwd` type indicates posixAccount-type entries. The `group` type indicates posixGroup-type entries.

`-n <name>` Specifies a single account or group name. Use of `-n` is the same as `-f "(uid=<name>)"` for accounts and `-f "(cn=<name>)"` for groups.

`-f <filter>` Specifies an LDAP-style search filter, `<filter>`, used to select specific entries from the LDAP directory server. When you use the `-f` option, the filter specified by `<filter>` applies to Posix-style users or groups (depending on whether you specify the `-t passwod` or `-t group` option).

`-m` Displays the names of the mapped attributes when returning results.

5.5.4 Adding a user or a group

When adding user or group entries to the LDAP directory server, the `ldapugadd` tool uses template files to discover the required data models for a new user and group entry. Template files define what object classes and attributes are required to create new user and group entries. LDAP-UX provides the flexibility that allows you to define unique data models for user and group entries. LDAP-UX supports two default template files (for `passwd` and `group` services) for a standard LDAP directory server, along with two default template files for Windows Active Directory Server. These template files can be found under `/etc/opt/ldapux/ug_templates` directory. For detailed information on how to define template files and how to name and create template files, see Section 7.3.5.6 (page 242).

The `ldapugadd` tool uses a local configuration file, `/etc/opt/ldapux/ldapug.conf`, to manage the default values of the `uidNumber_range`, `gidNumber_range`, `user_gidNumber`, `default_homeDirectory` and `default_loginShell` parameters when creating user or group entries for an LDAP directory server. See Section 7.3.5.5 (page 241) for details.

5.5.4.1 Adding users

You can add users to your system as follows:

1. Add the user's posixAccount entry to your LDAP directory.

   You can use your directory's administration tools, the `ldapugadd` command, or the `ldapentry` tool to add a new user entry to your directory. If you are adding a large number of users, you could create a `passwd` file with those users and use the migration tools to add them to your directory. For information about these tools, see *NIS/LDAP Gateway Administrator's Guide* at the following location:

   http://www.hp.com/go/hpux-security-docs

   Click **HP-UX LDAP-UX Integration Software**.

   To add the new user with the HP-UX Directory Server Console, select the Directory tab. Select the directory location in the left panel where your user information is. Select the **Object→New→Other** menu item. Select the posixAccount object class in the dialog box and select OK. Fill in the values for the user and select **OK**.

2. Add the user to the appropriate posixGroup entry.

   You can use your directory's administration tools, or the `ldapmodify` program to add the user to the appropriate group in the directory. Add the user name to the `memberuid` attribute.

   To add the new user with the the HPDS/RHDS Directory Server Console, select the Directory tab. Select the directory location in the left panel where your group information is. Double click on the group where you want to add the user, or select the group and select the **Object→Open** menu item. In the dialog box, select the `memberuid` attribute. Then, select the **Edit→Add** menu item. Fill in the user's UID (login) name in the new field and select **OK**.

3. To verify that the information was added and is accessible to the client, use `nsquery` or `pwget`:

   ```
   nsquery passwd user
   pwget -n user
   ```

5.5.4.2 Examples of adding a user

You can use `ldapugadd` to add new POSIX accounts or groups to an LDAP directory server.

Use `LDAP_BINDDN` to specify the distinguished name (DN) of a user with sufficient directory server privilege to add users or groups in the directory server. Use `LDAP_BINDCRED` to specify a password for the LDAP user specified by `LDAP_BINDDN`. Alternately, you can input LDAP administrator bind identity and credential interactively with a prompt (`-P`) option.

The `LDAP_UGCRED` environment variable specifies the new password of a user or group being created. You must specify the `-PW` option when using `LDAP_UGCRED`. The use of passwords for new groups is not recommended. Alternately, you can use the `-PP` command option to prompt for the password of the user or group being created.

Below are examples of using `ldapugadd` to add user entries.

Run the following command to set the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables

```
export LDAP_BINDDN = "cn=Jane Admin,ou=admins,dc=example,dc=com"
export LDAP_BINDCRED = "Jane's password"
```

Run the following command to specify the `LDAP_UGCRED` environment variable:

```
export LDAP_UGCRED = "user_password"
```

Run the following commands to discover what non-POSIX attributes defined in the default template file are required to create the new user entry:

```
cd /opt/ldapux/bin
./ldapcfinfo -t passwd -R
```

The output of the commands is as follows

```
Surname
```

The following commands add an account entry for the user, `mtam`, with the user's primary login group id, 200. `ldapugadd` creates the password for new user, `mtam`, using the user password specified in the `LDAP_UGCRED` environment variable. After creating the user entry, `ldapugadd` attempts to add this user as a member of the group number 200.

Run the following command to create the new account entry for the user, `mtam`:

```
./ldapugadd -t passwd -PW -f "Mike Tam" -g 200 mtam surname="Tam"
```

Run the following command to display the new user entry, `mtam`:

```
./ldapuglist -t passwd -n mtam
```

Below is the user entry:

```
dn: cn=Mike Tam,ou=people,dc=example,dc=com
cn: Mike Tam
uid: mtam
uidNumber: 2200
gidNumber: 200
homeDirectory: /home/mtam
loginShell: /usr/bin/ksh
```

The following command adds an account entry for the user, `jsmart`, with the user's primary login group id, 200 and the `sn` attribute value. `ldapugadd` creates the password for new user, `jsmart`, using the user password specified in the `LDAP_UGCRED` environment variable. After creating the user entry, `ldapugadd` attempts to add this user as a member of the group number 200. The `ldapugadd` tool dynamically assigns the `uidNumber` value from the pre-configured range.

```
./ldapugadd -t passwd -PW -f "John Smart" -g 200 jsmart surname="Smart"
```

Run the following command to display the new user entry, `jsmart`:

```
./ldapuglist -t passwd -n jsmart sn
```

Below is the new user entry:

```
dn: cn=John Smart,ou=people,dc=example,dc=com
cn: John Smart
uid: jsmart
uidNumber: 2350
gidNumber: 200
homeDirectory: /home/jsmart
loginShell: /usr/bin/ksh
sn: Smart
```

The following command adds an account entry for the user, `tsheu`, with the user's primary login group id, 350, and `gecos` field information. In this example, the `gecos` attribute has been mapped to `cn`, `l` and `telephone` in the LDAP-UX configuration profile. `ldapugadd` creates the password for new user, `tsheu`, using the password specified in the `LDAP_UGCRED` environment variable. After creating the user entry, `ldapugadd` attempts to add this user as a member of the group number 350.

```
./ldapugadd -t passwd -PW -g 350 -I "Tom Sheu,Building-1A,555-555-5555" tsheu surname="Sheu"
```

Use the following command to display the new user entry, `tsheu`, with mapped attribute information:

```
./ldapuglist -t passwd -m -n tsheu
```

Below is the user entry:

```
dn: cn=Tom Sheu,ou=people,dc=example,dc=com
cn[cn]: Tom Sheu
uid[uid]: tsheu
uidNumber[uidnumber]: 2200
gidNumber[gidnumber]: 350
homeDirectory[homeDirectory]: /home/tsheu
```

```
loginShell[loginshell]: /usr/bin/sh
gecos[cn]: Tom Sheu
gecos[l]: Building-1A
gecos[telephone]: 555-555-5555
```

Command arguments applicable to -t passwd

The following are the options and arguments used in the above examples of the `ldapugadd -t passwd` commands:

`-t <type>`
Specifies the type of entry the `ldapugadd` tool operates. `<type>` can be `passwd` or `group`. The `passwd` type represents LDAP user entries which contain POSIX account-related information. The `group` type represents LDAP group entries which contain POSIX group-related information.

`-f <full_name>`
This optional argument only applies to the `passwd` service. This option specifies the user's full name.

`-g <gid/gid_nubmer>`
Specifies the user's primary login group name or id number. After creating the user entry, `ldapugadd` attempts to add the user as a member of the specified group.

`-I <gecos>`
Specifies the `GECOS` fields for the user. Typically the `GECOS` argument contains the following four fields which represent (in order):

- The user's full name
- The user's work location
- The user's work telephone number
- The user's home telephone number (often omitted)

Each field in the `<gecos>` argument must be separated by a comma.

`-PW`
Sets the user or group password attribute. If you specify `-PW`, you must specify either the `LDAP-UGCRED` environment variable or the `-PP` option.

`<uid_name>`
Required argument. Specifies the POSIX style login name for the new user entry. This argument must follow all command-line options and must precede the `<attr>=<value>` parameters (if provided).

`<attr>=<value>`
This option specifies arbitrary LDAP attributes and values. `<attr>=<value>` parameters are optional and must be specified as the last parameters on the command line.

5.5.4.3 Examples of adding a group

Use the following command to add a new group entry for the group name, `groupA`. In this example, `ldapugadd` creates the new group, `groupA`, and defines initial group membership by adding the user account `tsheu` as a member.

```
./ldapugadd -t group -M tsheu groupA
```

Use the following command to display the new group entry, `groupA`:

```
./ldapuglist -t group -f "(cn=groupA)"
```

The output of the group entry is as follows:

```
dn: cn=groupA,ou=Group,dc=example,dc=com
cn: groupA
gidNumber: 550
memberUid: tsheu
```

Command arguments applicable to -t group

The following are the command arguments and options used in the above examples of the `ldapugadd -t group` commands:

`-M <member>`	Defines initial group membership by adding the specified user accounts as members.
`-g <gid_nubmer>`	Specifies the group id number for the new group.
`<group_name>`	Required argument. Specifies the POSIX style group name for the new group entry.

5.5.4.4 Modifying defaults in /etc/opt/ldapux/ldapug.conf

You can use the `ldapugadd -D` command to change default values of the `uidNumber_range`, `gidNumber_range`, `user_gidNumber`, `default_homeDirectory` and `default_loginShell` parameters in the `/etc/opt/ldapux/ldapug.conf` file.

The following commands set new default minimum and maximum ranges of UID numbers in the local configuration file, `/etc/opt/ldapux/ldapug.conf`. The `ldapugadd` tool randomly selects a new ID from this range if you do not specify an account number.

```
cd /opt/ldapux/bin
./ldapugadd -D -t passwd -u 1000:5000
```

The following command sets new default minimum and maximum ranges of GID numbers in the local configuration file, `/etc/opt/ldapux/ldapug.conf`. The `ldapugadd` tool randomly selects a new ID from this range if you do not specify a group number.

```
./ldapugadd -D -t group -g 200:2500
```

The following command sets new default group ID number in the local configuration file, `/etc/opt/ldapux/ldapug.conf`. The `ldapugadd` tool uses this new value when creating new user entries in an LDAP directory server.

```
./ldapugadd -D -t passwd -g 5000
```

The following command sets new default login shell in the local configuration file, `/etc/opt/ldapux/ldapug.conf`. The `ldapugadd` tool uses this new login shell when creating new user entries in an LDAP directory server.

```
./ldapugadd -D -t passwd -s /net/bin/sh
```

The following command sets new default parent home directory in the local configuration file, `/etc/opt/ldapux/ldapug.conf`. The `ldapugadd` tool uses this new home directory when creating new user entries in an LDAP directory server.

```
./ldapugadd -D -t passwd -d /net/home
```

Command arguments applicable to -D

The following describes arguments used in the above examples of the `ldapugadd -D` commands:

`-D`	Uses this option to change local host defaults in the `/etc/opt/ldapux/ldapug.conf` file which are used by `ldapugadd` when creating new user or group entries in an LDAP directory server.
`-u <min_uid>:<max_uid>`	Sets new default minimum and maximum ranges that `ldapugadd` uses when provisioning an UID number for new user entries.
`-g <default_gid>`	Specifies the default group ID number used when creating new user entries.
`-g <min_gid>:<max_gid>`	Sets new default minimum and maximum ranges that `ldapugadd` uses when provisioning a GID number for new group entries.

-s <default_shell>	Specifies the default login shell that ldapugadd uses when creating a new user entry.
-s <default_home>	Specifies the default parent home directory that ldapugadd uses when creating a new user home directory.

5.5.5 Modifying a user

You can use `ldapugmod` tool to modify exiting POSIX accounts or groups in an LDAP directory server. This section provides examples of using `ldapugmod` to modify a user's information.

Use `LDAP_BINDDN` to specify the distinguished name (DN) of a user with sufficient directory server privilege to modify users or groups in the directory server. Use `LDAP_BINDCRED` to specify a password for the LDAP user specified by `LDAP_BINDDN`. Alternately, you can input LDAP administrator bind identity and credential interactively with a prompt (`-P`) option.

The `LDAP_UGCRED` environment variable specifies the new password of a user or group being modified. You must specify the `-PW` option when using `LDAP_UGCRED`. Alternately, you can use the `-PP` command option to prompt for the password of the user or group being modified.

The following commands set the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables:

```
export LDAP_BINDDN = "cn=Jane Admin,ou=Admins,dc=example,dc=com"
export LDAP_BINDCRED = "Jane's password"
```

The following commands are used to change the password of the user, `mtam`, using the new user password defined in `LDAP_UGCRED`:

```
cd /opt/ldapux/bin
export LDAP_UGCRED = "new password"
./ldapugmod -t passwd -PW mtam
```

The following command replaces the `uidNumber` attribute with the new value for the user entry, `mswartz`:

```
./ldapugmod -t passwd -u 300 mswartz
```

The following command replaces the `sn` attribute with the new value for the user entry, `mLou`:

```
./ldapugmod -t passwd mLou "sn=Lou"
```

The following command removes the `sn` attribute and value for the user entry, `alee`:

```
./ldapugmod -t passwd -R "sn=Ann Lee" alee
```

The following command replaces the `gecos` fields with the new values for the user entry, `alouie`:

```
./ldapugmod -t passwd -I "Ann Louie,Building-6,222-2222" alouie
```

The following command adds the `description` attribute and value to the user entry, `mscott`:

```
./ldapugmod -t passwd -A "description=test user entry" mscott
```

Command arguments

The following describes arguments/options used in the above examples for the `ldapugmod -t passwd` commands:

-PW	Sets the user or group password attribute. If you specify `-PW`, you must specify either the `LDAP-UGCRED` environment variable or the `-PP` option.
-A <attrval>	Specifies an attribute and value to be added to a user or group entry. When working with multi-valued attributes, you can use the `-A` option to add a new value for a multi-valued attribute, without removing already existing values for that attributes.
-R <attrval>	Specifies an attribute and value to be removed from a user or group entry. When working with multi-valued attributes, you can use the `-R` option to remove a specified value for a multi-valued attributes.
-u <uidNumber>	Replaces the user's numeric id number.

-I <gecos> Replaces the GECOS fields for the user. Typically the GECOS argument contains the following four fields which represent (in order):

- The user's full name
- The user's work location
- The user's work telephone number
- The user's home telephone number (often omitted)

Each field in the <gecos> argument must be separated by a comma.

<attr>=<value> Allows modification of arbitrary LDAP attributes and values.

5.5.6 Modifying a group

You can use ldapugmod tool to modify exiting groups in an LDAP directory server. This section provides examples of using ldapugmod to modify group entry information.

The following command replaces the gidNumber value for the group entry, GroupA:

```
./ldapugmod -t group -g 2500 groupA
```

In the following example, a group entry contains multiple values of the description attribute. It is as follows:

```
dn: cn=GroupB,ou=Group,dc=example,dc=com
cn: GroupB
gidNumber: 350
MemberUid: tlee
Description: Test Group
Description: A Group Entry
```

Run the following command to replace all instances of the description attribute with new value "Group B Entry" for the GroupB entry:

```
./ldapugmod -t group GroupB "description=Group B Entry"
```

The result of the GroupB entry is as follows:

```
dn: cn=GroupB,ou=Group,dc=example,dc=com
cn: GroupB
gidNumber: 350
MemberUid: tlee
Description: Group B Entry
```

In the following example, a group entry in an LDAP directory server is as follows:

```
dn: cn=GroupC,ou=Group,dc=example,dc=com
cn: GroupC
gidNumber: 500
MemberUid: alouie
Description: A IT Group
Description: A Group Entry
```

Run the following command to add an instance of the description attribute and value to the group entry, GroupC, without removing already existing values for that attributes:

```
./ldapugmod -t group -A "description=Group C Entry" groupC
```

The result of the GroupC entry is as follows:

```
dn: cn=GroupC,ou=Group,dc=example,dc=com
cn: GroupC
gidNumber: 500
MemberUid: alouie
Description: A IT Group
Description: A Group Entry
Description: Group C Entry
```

The following command adds the three members, atam, mlou, mscott, to the group entry, groupA:

```
./ldapugmod -t group -a atam,mlou,mscott GroupA
```

The following command removes one member, `atam` from the group entry, `groupA`:

```
./ldapugmod -t group -r atam GroupA
```

Command arguments

The following describes arguments/options used in the above examples for the `ldapugmod -t group` commands:

`-A <attrval>`	Specifies an attribute and value to be added to an entry. When working with multi-valued attributes, you can use the `-A` option to add a new value for a multi-valued attribute, without removing already existing values for that attributes.
`-g <gidNumber>`	Replaces the group's numeric id number.
`-a <member>[,...]`	Adds one or more members to the specified group. When specifying a list of members, you must use a comma with no white space to separate each member.
`-r <member>[,...]`	Removes one or more members from the specified group. When you specify a list of members, you must use a comma with no white space to separate each member.

5.5.7 Deleting a user or a group

You can use `ldapugdel` to remove POSIX user and group entries from an LDAP directory server. With the `-O` option, `ldapugdel` enables you to remove only POSIX related attributes and object classes from a user or group entry without removing the entire entry.

The `userPassword`, `uid`, `cn` and `description` attributes are commonly used by most other user and group schemas. With the `-O` option, the `ldapugdel` tool does not attempt to remove these attributes. The `uidNumber`, `gidNUmber`, `loginShell`, `homeDirectory`, `gecos` and `memberUid` are more unique to the POSIX schema, and are removed when the `-O` option is specified. The `ldapugdel -t passwd -O` command removes the `posixAccount` object class and following attributes:

- `uidNumber`
- `gidNumber`
- `homeDirecotry`
- `loginShell`
- `gecos`

Use the `ldapugdel -t group -O` command, `ldapugdel` removes the `posixGroup` object class and following attributes:

- `gidNumber`
- `memberUId`
- `userPassword`

5.5.7.1 Examples

This section provides examples of using `ldapugdel`.

Use `LDAP_BINDDN` to specify the distinguished name (DN) of a user with sufficient directory server privilege to delete users or groups in the LDAP directory server. Use `LDAP_BINDCRED` to specify a password for the LDAP user specified by `LDAP_BINDDN`. Alternately, you can input LDAP administrator bind identity and credential interactively with a prompt (`-P`) option.

Run the following commands to specify the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables:

```
export LDAP_BINDDN = "cn=Jane Admin,ou=admins,dc=exmple,dc=com"
export LDAP_BINDCRED = "Jane's password"
```

Run the following commands to delete the entire user account entry, skeith:

```
cd /opt/ldapux/bin
./ldapugdel -t passwd skeith
```

Run the following command to delete only the posixAccount object class and associated attributes, uidnumber, gidNumber, homeDirectory, loginShell and gecos, without delete the entire user entry, msmith:

```
./ldapugdel -t passwd -O msmith
```

Run the following command to delete the entire group entry with the distinguished name, "cn=groupA,ou=groups,dc=example,dc=com":

```
./ldapugdel -t group -D "cn=groupA,ou=groups,dc=example,dc=com"
```

Run the following command to delete only the posixGroup object class and associated attributes, gidNumber, memberUid and userPassword, without delete the entire group entry, groupB:

```
./ldapugdel -t group -O groupB
```

Command arguments

The following describes the ldapugdel options and arguments used in the above examples:

-t <type> Specifies the type of entry the ldapugdel tool needs to delete. <type> can be passwd or group. The passwd type represents LDAP user entries which contain POSIX account-related information. The group type represents LDAP group entries which contains POSIX group-related information.

-O Allows the ldapugdel tool to delete only the posixAccount or posixGroup object class and associated attributes, without deleting the entire user or group entry.

-D The ldapugdel tool searches for the named user or group using the search rules defined by the service search descriptor in LDAP-UX configuration profile. You can use the -D option to specify the distinguished name (DN) of the entry being deleted. You can specify only one of -D, <uid_name> or <group_name> parameter on the command line.

5.5.8 Examining the LDAP-UX configuration

The ldapcfinfo tool provides several capabilities used to report LDAP-UX configuration and status. When used specifically with the LDAP user and group tools, ldapcfinfo can be used to discover LDAP-UX configuration details about required attributes when adding new users or groups to an LDAP directory server.

5.5.8.1 Checking if LDAP-UX is configured

Use the ldapcfinfo -t <type> command to check whether the LDAP-UX is properly configured for a specified service. The valid <type> value can be passwd, group, netgroup, services, rpc, hosts, networks, automount, NIS-based publickey, protocols and pam.

The following commands check whether LDAP-UX is properly configured for the passwd service:

```
cd /opt/ldapux/bin
./ldapcfinfo -t passwd
```

Assume that LDAP-UX is properly configured, below is the output of the above command:

```
INFO:  CFI_CONFIG_SUCCESS:
       "passwd" service appears properly configured for LDAP-UX operation
```

The following command checks to see if LDAP-UX is properly configured for the automount service:

```
./ldapcfinfo -t automount
```

Assume that the automount service is not configured for LDAP-UX support, below is the output of the above command:

```
WARNING:  CFI_CONFIG_FAILURE:
          "automount" service not configured for LDAP-UX support
```

5.5.8.2 Listing available templates

Use the ldapcfinfo -t <type> -L command to display a list of available templates. The valid <type> value can be passwd or group.

Run the following command to display a list of available template files that ldapugadd uses to create a new user entry for the passwd name service:

```
./ldapcfinfo -t passwd -L
```

Assume that the /etc/opt/ldapux/ug_templates/ug_passwd_std.tmpl, /etc/opt/ldapux/ug_templates/ug_passwd_default.tmpl /etc/opt/ldapux/ug_templates/ug_passwd_ads.tmpl files are currently available on the system, the output of the above command is as follows:

```
/etc/opt/ldapux/ug_templates/ug_passwd_ads.tmpl
/etc/opt/ldapux/ug_templates/ug_passwd_std.tmpl
/etc/opt/ldapux/ug_templates/ug_passwd_default.tmpl
```

Run the following command to display a list of available template files that ldapugadd uses to a group entry for the group name service:

```
./ldapcfinfo -t group -L
```

Assume that the /etc/opt/ldapux/ug_templates/ug_group_std.tmpl, /etc/opt/ldapux/ug_templates/ug_group_default.tmpl /etc/opt/ldapux/ug_templates/ug_group_ads.tmpl files are currently available on the system, the output of the above command is as follows:

```
/etc/opt/ldapux/ug_templates/ug_group_ads.tmpl
/etc/opt/ldapux/ug_templates/ug_group_std.tmpl
/etc/opt/ldapux/ug_templates/ug_group_default.tmpl
```

5.5.8.3 Discovering required attributes

Use the ldapcfinfo -t <type> -R command to discover what attributes defined in a template file are required to create a new user or group entry. Because the RFC 2307 POSIX attributes are a static known list and are required, ldapcfinfo displays only non-POSIX attributes. The valid <type> value can be passwd or group.

The following command displays the non-POSIX attributes defined in the default template file, /etc/opt/ldapux/ug_tempates/ug_passwd_std.tmpl, required by the ldapugadd command for the passwd name service:

```
./ldapcfinfo -t passwd -R
```

The output of the command is as follows:

```
Surname
```

5.5.8.4 Displaying configuration defaults

Use the ldapcfinfo -t <type> -D command to display the LDAP default values in the /etc/opt/ldapux/ldapug.conf file used for the ldapugadd command. The valid <type> value can be passwd or group. If you specify the -t password -D option, ldapcfinfo displays UID range, default primary GID number, default home directory and default login shell information. The -t group -D option displays the GID range.

Run the following command to display the LDAP default values in the /etc/opt/ldapux/ldapug.conf file:

```
./ldapcfinfo -t passwd -D
```

Below is the output of the above command for the passwd name service:

```
uidNumber_range=100:20000
default_gidNumber=20
default_homeDirectory=/home
default_loginShell=/usr/bin/sh
```

Run the following command to display the LDAP default configuration values in the `/etc/opt/ldapux/ldapug.conf` file for the group name service:

```
./ldapcfinfo -t group -D
```

Below is the output of the above command:

```
gidNumber_range=100:2000
```

5.5.8.5 Displaying the LDAP-UX profile's DN

Run the following command to display the location of the LDAP-UX configuration profile:

```
./ldapcfinfo -P
```

The output of the command is as follows:

```
dn: cn=ldapux-profile,ou=org,dc=example,dc=com
host: 55.5.55.15:389
```

If SSL is required to download the profile, the output appears as follows:

```
dn: cn=ldapux-profile,ou=org,dc=example,dc=com
hostssl: 55.5.55.15:636
```

5.5.8.6 Displaying default search base

Use the `ldapcfinfo -t <type> -b` command to display the primary (first) configured search base in the LDAP-UX profile configuration for a specific service. The valid `<type>` value can be `passwd` or `group`.

The following command displays the LDAP-UX default search base for the `passwd` name service. In this example, “ou=People,” has been configured as the search base for the `passwd` name service:

```
./ldapcfinfo -t passwd -b
```

The output of the above command is as follows:

```
ou=People,ou=org,dc=example,dc=com
```

The following command displays the LDAP-UX default search base for the `group` name service. In this example, “ou=Groups,” has been configured as the search base for the group name service:

```
./ldapcfinfo -t group -b
```

The output of the above command is as follows:

```
ou=Groups,ou=org,dc=example,dc=com
```

5.5.8.7 Displaying recommended attributes

Use the `ldapcfinfo -t <type> - a <DN>` command to display a recommended list of attributes that an interactive management tool considers making available for modification for the specified entry.

The following command displays the recommended list of attributes for the user account entry with the distinguished name (DN), `"cn=sfong,ou=people,ou=org,dc=example,dc=com"`:

```
./ldapcfinfo -t passwd -a "cn=sfong,ou=people,ou=org,dc=example,dc=com"
```

Below is the output of the command:

```
cn
uid
uidnumber
gidnumber
loginshell
homedirectory
```

```
gecos
description
```

5.5.8.8 Displaying attribute mapping for a specific name service

Use the `ldapcfinfo -t <type> -m` command to display attribute mapping information defined in the LDAP-UX configuration profile. The valid `<type>` value can be `passwd` or `group`.

The following command displays the attribute mapping for the `gecos` attribute which has been mapped to `cn`, `l` and `telephone` attributes:

```
./ldapcfinfo -t passwd -m gecos
```

The output of the above command is as follows:

```
gecos=cn l telephoneNumber
```

The following command displays the attribute mapping for the `gecos` and `uidNumber` attributes. In this example, `gecos` has been mapped to `cn`, `l` and `telephone` attributes, and `uidNumber` has been mapped to the `employeeNumber` attribute:

```
./ldapcfinfo -t passwd -m gecos,uidNumber
```

The output of the above command is:

```
gecos=cn l telephoneNumber
uidNumber=employeeNumber
```

5.6 Managing hosts in an LDAP-UX domain

LDAP-UX B.05.00 introduces utilities that simplify management of hosts, adding to the toolset provided for managing users and groups. Two utilities have been added, `/opt/ldapux/bin/ldaphostmgr` and `/opt/ldapux/bin/ldaphostlist`. These utilities let you discover, create, modify, and remove host objects in the directory server. Similar to the user and group management tools described in Section 5.5 (page 159), these host-management tools integrate with the LDAP-UX configuration, allowing administrators and automated scripts to modify host information without needing to know configuration information such as the directory server host name, directory server tree location, authentication methods, attribute mapping, search filters, and so forth.

As part of the guided installation (see Section 2.3 (page 23)), LDAP-UX uses the `ldaphostmgr` tool to provision information about the current host into the directory server, including the host's ssh public key data. (For more information about using LDAP-UX to manage ssh host keys and to pre-establish trust between hosts, see Chapter 6 (page 193).)

This section describes how to use the LDAP host management tools, `ldaphostmgr` and `ldaphostlist`, by following example usage scenarios. Additional usage scenarios are described in "Managing ssh host keys with LDAP-UX" (page 193).

NOTE: The examples in this section are targeted toward entries stored in an HP-UX Directory server. Windows ADS users should translate the examples to the respective usage in ADS. For example, instead of using an administrator DN of `uid=domadmin,ou=people,dc=mydomain,dc=eample,dc=com`, you might see `cn=administrator,cn=users,dc=mydomain,dc=eample,dc=com` in a Windows domain.

5.6.1 Adding a host

Use the `ldaphostmgr` tool to add, modify, and delete hosts to, in, and from the directory server. `ldaphostmgr` relies on the LDAP-UX configuration profile to determine the proper location to store new hosts. (For information about displaying the configuration profile, see Section 5.10 (page 182); for information about configuration profile object classes and attributes, see "LDAP-UX Client Services object classes" (page 349).) The location where hosts are stored is defined in the profile's `serviceSearchDescriptor` for the `hosts` service. If you used the guided installation (`autosetup`), this location is automatically defined to be `ou=hosts,`*`suffix`* or `cn=computers,`*`suffix`* (for a Windows domain), where *`suffix`* is the base of your directory

tree or base of the Windows domain. If you have an existing configuration profile that was not set up using guided installation, the location where your hosts will be stored might be defined to a different location, or might not be defined at all (using defaults). You can use the `ldapcfinfo` tool to determine where LDAP-UX believes host information should be located. For example:

```
# /opt/ldapux/bin/ldapcfinfo -t hosts -b
ou=Hosts,dc=mydomain,dc=example,dc=com
```

Before adding any hosts to the directory server, verify that the base DN discovered in the previous example is defined to the proper location in the directory server tree. If it is not, you can reconfigure the LDAP-UX profile and modify the `host serviceSearchDescriptor` attribute using the steps outlined in Section 5.12 (page 183).

Use the `-a` option of the `ldaphostmgr` command to add new hosts to the directory, as shown in the following example. (In the examples that follow, assume the PATH environment variable contains `/opt/ldapux/bin`.)

```
# ldaphostlist
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: brewer
ipHostNumber: 16.92.96.116

# id
uid=1173(domadmin) gid=1136(DomainAdmins) groups=1411(HostAdmins)
# ldaphostmgr -a baker
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
# ldaphostlist
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: brewer
ipHostNumber: 16.92.96.116

dn: cn=baker,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: baker
ipHostNumber: 16.89.146.146
```

In the previous example, one host (`brewer`) already existed in the directory server. Another (`baker`) was added using the `-a` option. By default, the IP address for the host is discovered and added. In addition, the owner is assigned by default. You can display the owner, or any attribute, using `ldaphostlist`, as follows:

```
# ldaphostlist -n baker \*
dn: cn=baker,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: baker
ipHostNumber: 16.89.146.146
objectClass: top
objectClass: device
objectClass: iphost
owner: uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com
```

In this case, the owner was assigned to `domadmin`, which is the user that created the entry in the preceding example. You can assign ownership to a different user or group using the `-O` option:

```
# ldaphostmgr -a -O user:bobj chef
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
# ldaphostlist -n chef owner
dn: cn=chef,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: chef
ipHostNumber: 0.0.0.0
owner: uid=bobj,ou=People,dc=mydomain,dc=example,dc=com
```

If you used the guided installation to create your directory server, then by default, owners of hosts have rights to manage information about the hosts. For additional information, see Section 2.3.2 (page 27).

5.6.2 Modifying a host

Use the `-m` option of `ldaphostmgr` to modify existing host entries. If neither `-a`, `-m`, nor `-g` is specified, `-m` is assumed. In the `-a` and `-m` modes, `ldaphostmgr` can be used to add, change, or remove arbitrary attributes. You can manage some attributes using `ldaphostmgr` command-line options; for example, use `-k` to manage the host's ssh public key, and `-i` to manage the host's IP address. You can add arbitrary attributes using the `-A` or `-R` options, or by adding an attribute and value list to the end of the command line. The following example shows how to assign a "role" to a host:

```
# ldaphostmgr -m -r WEBSERVER -A objectclass=labeledURIObject \
              -A "labeledUri=http://baker.mydomain.example.com" baker
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
# ldaphostlist -n baker \*
dn: cn=baker,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: baker
ipHostNumber: 16.89.146.146
objectClass: top
objectClass: device
objectClass: iphost
objectClass: domainEntity
objectClass: labeledURIObject
owner: uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com
entityRole: WEBSERVER
labeledUri: http://baker.mydomain.example.com
```

Adding and removing attributes can be affected when these attributes are multivalued (meaning one attribute type can contain multiple instances, with different values). Managing multivalued attributes is handled differently for arbitrary attributes as opposed to attributes managed by command-line options (like `-r` in the previous example). For example, using `-r` replaces all existing values of the `entityRole` attribute. Refer to the *ldaphostmgr*(1M) manpage for additional information for each usage scheme. IP addresses are stored in the `ipHostNumber` attribute, and managed with the `-i` option. Additional details on how to manage IP addresses are described in Section 5.6.4 (page 177).

5.6.3 Deleting a host

Use the `-d` option of `ldaphostmgr` to remove a host from the directory server. This removes the entire entry from the directory server. To only remove specific attributes from an entry, see the `-R` option in the *ldaphostmgr*(1M) manpage. The following example shows how to remove a host entry:

```
# ldaphostlist entityRole
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: brewer
ipHostNumber: 16.92.96.116

dn: cn=baker,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: baker
ipHostNumber: 16.89.146.146
entityRole: WEBSERVER

dn: cn=chef,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: chef
ipHostNumber: 0.0.0.0

# ldaphostmgr -d chef
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
# ldaphostlist
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: brewer
```

```
ipHostNumber: 16.92.96.116

dn: cn=baker,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: baker
ipHostNumber: 16.89.146.146
```

CAUTION: If you used guided installation to configure LDAP-UX on a host, removing that host entry also removes the proxy user defined for that host. Removing the host's proxy user entry disables the ability of the OS to use LDAP as an OS management repository. When you use guided installation to create a directory server, that directory server will require authenticated access to itself before returning any data. The host's proxy user entry is used to define a way for the host's OS to authenticate to the directory server. Removing the proxy user (host entry) terminates `ldapclientd`'s ability to bind to the directory server. For example, if we remove the proxy user entry for the current host, the following error occurs:

```
# ldaphostmgr -d "$(hostname)"
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=eample,dc=com]:
Password:
# ldaphostlist
ERROR:    BIND_ERR:
          Failed to bind to the directory server.
```

You can restore the host's proxy entry and restore the proxy credential file as follows. You will need to define a new password in order to re-recreate the proxy credentials:

```
# ldaphostmgr -a -P -f -k all -S "$(hostname)"
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=eample,dc=com]:
Password:
Host password:
Re-enter host password:
added DN: cn=brewer,ou=Hosts,dc=mydomain,dc=eample,dc=com
# su
Password:
# /opt/ldapux/config/ldap_proxy_config -i << EOD
> cn=brewer,ou=Hosts,dc=mydomain,dc=eample,dc=com
> [Host Password From Above]
> EOD
```

5.6.4 Managing IP addresses

Use the `-i` option to add or remove IP addresses to or from host entries. Without flags, the `-i` option adds an additional IP address to a host entry. If you have a host with multiple IP interfaces, you can use `-i` to add any additional IP addresses that have not yet been registered. For example:

```
# ldaphostlist -n brewer
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: brewer
ipHostNumber: 16.92.96.113

# ldaphostmgr -i 192.168.10.10 brewer
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
# ldaphostlist -n brewer
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: brewer
```

```
ipHostNumber: 16.92.96.113
ipHostNumber: 192.168.10.10
```

To remove an IP address for a host, use the `-i` option with the `!` flag in front of the IP address to be removed. For example, to remove the address added in the previous example:

```
# ldaphostmgr -i !192.168.10.10 brewer
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
# ldaphostlist -n brewer
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: brewer
ipHostNumber: 16.92.96.113
```

To remove all IP addresses for a host, use the `-i` option with the `!` flag by itself. However, after removing all IP addresses, the special address `0.0.0.0` is added to assure that the `ipHost` object class can remain as part of the host entry. The `ipHost` object class is the only standard object class that is used to identify hosts and distinguish them from other types of devices. Example:

```
# ldaphostmgr -i "!" brewer
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
# ldaphostlist -n brewer
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: brewer
ipHostNumber: 0.0.0.0
```

To remove all `ipHostNumber` attributes, use the `-R` option. This removes both the `ipHostNumber` attribute and the `ipHost` object class. However, when this occurs, `ldaphostlist` is no longer able to display the host, since the `ipHost` object class is the critical object class used to distinguish hosts from other types of devices managed in the directory server. You can use the `-F` option to override the default `ldaphostlist` search filter to find hosts or other types of devices that do not use the `ipHost` object class. Example:

```
# ldaphostmgr -R iphostNumber -R objectclass=iphost brewer
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
# ldaphostlist -n brewer  1
# ldaphostlist -F "(&(objectclass=device)(cn=brewer))"
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: brewer
```

1 No host was found

5.6.5 Managing hosts in groups

There are several ways to group hosts. You can accomplish simple grouping by adding hosts as members of the traditional X.500 group structures. Use the `-G` option to do this. The `-G` option supports the `!` flag to remove a host from a group, similar to the `-i` option. Note that in a POSIX environment, the grouping of hosts is not a native construct. Users may be members of groups, but hosts may not. In an LDAP-based directory server, any object in the directory server may be a member of any X.500 style group. So, grouping of hosts using the `-G` option can add hosts only as members of the X.500 style groups, identified with the `groupOfNames` or `groupOfUniqueNames` object classes. While you can list members of these types of groups using the `/opt/ldapux/bin/ldapsearch` utility, you can also extend the display capabilities of the `ldapuglist` tool to list groups that are standard X.500 groups and contain hosts as members. The following example shows how to add a host to a group (`dbhosts`) that already has one member (`baker`):

```
# ldapuglist -t group -P -F "(cn=dbhosts)" uniqueMember
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=eample,dc=com]:
Password:
```

```
dn: cn=dbhosts,ou=groups,dc=mydomain,dc=eample,dc=com
cn: dbhosts
uniqueMember: cn=baker,ou=Hosts,dc=mydomain,dc=eample,dc=com

# ldaphostmgr -G dbhosts chef
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=eample,dc=com]:
Password:
# ldapuglist -t group -P -F "(cn=dbhosts)" uniqueMember
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=eample,dc=com]:
Password:
dn: cn=dbhosts,ou=groups,dc=mydomain,dc=eample,dc=com
cn: dbhosts
uniqueMember: cn=baker,ou=Hosts,dc=mydomain,dc=eample,dc=com
uniqueMember: cn=chef,ou=Hosts,dc=mydomain,dc=eample,dc=com
```

To remove a host from a group, use the `!` flag in front of the host name:

```
# ldaphostmgr -G !dbhosts baker
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=eample,dc=com]:
Password:
# ldapuglist -t group -P -F "(cn=dbhosts)" uniqueMember
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=eample,dc=com]:
Password:
dn: cn=dbhosts,ou=groups,dc=mydomain,dc=eample,dc=com
cn: dbhosts
uniqueMember: cn=chef,ou=Hosts,dc=mydomain,dc=eample,dc=com
```

To list host entries that are members of a particular group, use the `-g` option of the `ldaphostlist` command. For example, to capture all the ssh host keys for a particular group of hosts, you could use the following command:

```
# ldaphostlist -g webhosts -k
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=eample,dc=com
cn: brewer
ipHostNumber: 0.0.0.0
sshPublicKey: ssh-rsa AAAAB3NzaC16AeE...

dn: cn=raptor,ou=Hosts,dc=mydomain,dc=eample,dc=com
cn: raptor
ipHostNumber: 16.92.96.215
sshPublicKey: ssh-rsa AAAAB3NzaC1yc2EAA...
```

5.6.6 Classifying hosts

Because `ldaphostmgr` lets you attach arbitrary attributes to host entries, you can use these attributes to classify systems and then use that information as a way to group hosts. Aside from grouping hosts using an enumerated list of members in X.500 groups, LDAP directory servers offer an efficient way to group systems based on their attributes. This is typically known as *dynamic grouping*. In the previous example, we created a group of hosts known as `dbhosts` (assuming these hosts might hold some form of data base). We could have just as easily defined a role for these hosts, marking them as DBSERVERs as follows:

```
# ldaphostmgr -r DBSERVER brewer
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=eample,dc=com]:
Password:
# ldaphostmgr -r DBSERVER raptor
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=eample,dc=com]:
Password:
```

Use the `-f` option of `ldaphostlist`, to quickly discover the list of `DBSERVER`s.

```
# ldaphostlist -f "(entityRole=DBSERVER)" \*
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=eample,dc=com
cn: brewer
ipHostNumber: 0.0.0.0
objectClass: top
objectClass: device
```

```
objectClass: ipHost
objectClass: ldapPublicKey
objectClass: domainEntity
owner: uid=domadmin,ou=People,dc=mydomain,dc=eample,dc=com
sshPublicKey: ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAvrJ...
entityRole: DBSERVER

dn: cn=raptor,ou=Hosts,dc=mydomain,dc=eample,dc=com
cn: raptor
ipHostNumber: 16.92.96.215
objectClass: top
objectClass: device
objectClass: ldapPublicKey
objectClass: iphost
objectClass: domainEntity
owner: uid=domadmin,ou=People,dc=mydomain,dc=eample,dc=com
sshPublicKey: ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAxe1...
entityRole: DBSERVER
```

5.6.7 Managing process access rights (proxy_is_restricted)

If you have configured LDAP-UX to use anonymous access to the directory server, you can skip this section.

Under specific conditions described below, the `ldaphostlist` utility will not allow the user to display arbitrary attributes associated with host entries managed in the directory server. If you try to display an attribute and cannot view it as expected, you can use the `-v` option to verify whether this attribute was restricted, as shown in the following example. Suppose a user wanted to display the owner of a host and gets a warning message like the one in the example:

```
# ldaphostlist -n brewer owner
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=eample,dc=com
cn: brewer
ipHostNumber: 0.0.0.0

# ldaphostlist -v -n brewer owner
WARNING:  LST_ATTR_RESTRICTED:
          Attribute "owner" is ignored.  Access rights to the attribute can not
          be determined because proxy access has been defined but
          proxy_is_restricted has not been set. Contact your system
          administrator.
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=eample,dc=com
cn: brewer
ipHostNumber: 0.0.0.0
```

This message can occur if LDAP-UX is configured to use a proxy user to access the directory server data. This is very common in an ADS environment, since by default, the ADS directory server does not allow anonymous access to data.

If you have installed and configured a previous version of LDAP-UX or did not use the guided installation (`autosetup`) to configure LDAP-UX, you would have defined your own proxy user. Because the `ldaphostlist` uses this same proxy user to access directory server data, `ldaphostlist` needs to know if the proxy user has access to data that a nonprivileged user should not be allowed to view. For example, if the proxy user was defined as `cn=administrator,cn=user,dc=mydomain,dc=example,dc=com` (for a Windows domain) or `cn=Directory Manager` (for an HP-UX Directory Server), the proxy user has rights to access any data in the directory server. While it would be bad practice to create a proxy user with privileged access rights, normally the proxy user is only used by `ldapclientd`, which limits what information it requests from the directory server. However, because the user can instruct `ldaphostlist` to view any attribute, `ldaphostlist` does not allow users to specify any attribute to be viewed, since these tools do not know if the proxy user has more privileges than should be granted to the user running the utility.

When a host is configured using the guided installation, an entry representing the host is created; this entry is also used as the proxy user for the OS. Because his host entry is created without

adding any special privileges, the guided installation sets a special flag (`proxy_is_restricted`) inside the `/etc/opt/ldapux/ldapclientd.conf` file to indicate that the proxy user has been created without any additional special privileges. This flag is also used by `ldaphostlist`, to determine if it is safe to request arbitrary attributes from the directory server. `ldaphostlist` assumes that the directory server has defined proper access control limits such that confidential or private information cannot be viewed by the proxy user. The `[general]` section of the client daemon configuration file (`ldapclientd.conf`) controls this behavior:

```
...
# If proxy_is_restricted is set to 1, then you are attesting that the
# directory server is restricting access to private or other confidential
# information from access by the proxy user.
proxy_is_restricted=1

# Allows the ldapclientd interface to return attributes that are associated
# with RFC2307-based services (such as users and groups), but that those
# attributes are not specifically part of the RFC2307 schema.  Any attribute
# specified below should be considered public information.
allowed_attribute=hosts:sshPublicKey
allowed_attribute=passwd:sshPublicKey
```

Setting `proxy_is_restricted` to `1` means that `ldaphostlist` will not restrict the user from displaying any attribute (the directory server may still deny access if access control instructions exist to limit what is visible to the proxy user.)

Only set `proxy_is_restricted` to 1 if you can verify that your proxy user defined in `/etc/opt/ldapux/pcred` does not have rights to access data in the directory server beyond that of any nonprivileged user. To identify what account is defined as the proxy user, use the `ldap_proxy_config` utility as follows, and then examine the directory server's access control settings to verify this account's privileges:

```
# /opt/ldapux/config/ldap_proxy_config -p
PROXY DN: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
```

5.7 Displaying the proxy user's DN

You can display the proxy user's distinguished name by running `/opt/ldapux/config/ldap_proxy_config -p`.

The following command displays the current proxy user:

```
ldap_proxy_config -p
PROXY DN: uid=proxy,ou=people,o=hp.com
```

5.8 Verifying the proxy user

The proxy user information is stored in the file `/etc/opt/ldapux/pcred`.

You can check if the proxy user can authenticate to the directory by running `/opt/ldapux/config/ldap_proxy_config -v` as follows:

```
cd /opt/ldapux/config
./ldap_proxy_config -v
File Credentials verified - valid
```

5.9 Creating a new proxy user

If you need to create a new proxy user and change your client systems to use the new proxy user, use the following steps:

1. Add the new proxy user to your directory with appropriate access controls. See the steps "Create a proxy user" and "Set access permissions for the proxy user" in Section 2.4.4 (page 65) for details.
2. Configure each client to use the new proxy user by running /opt/ldapux/config/ldap_proxy_config. See Section 7.2.6 (page 216) for details. See below for examples.
3. Run /opt/ldapux/config/ldap_proxy_config -p to display the proxy user you just configured and confirm that it is correct.
4. Run /opt/ldapux/config/ldap_proxy_config -v to verify the proxy user is working.

NOTE: While the proxy user information stored in the `pcred` file is protected for root-only access and not stored in plain text, it is not encrypted. Access to the `pcred` file must be restricted to prevent discovery of the proxy user's password. The same is true for the `acred` file.

5.9.1 Example

For example, the following command configures the local client to use a proxy user DN of `uid=proxy,ou=people,o=hp.com` with a password of `abcd1234`:

```
cd /opt/ldapux/config
./ldap_proxy_config -i
uid=proxy,ou=people,o=hp.com
abcd1234
```

The following command displays the current proxy user:

```
./ldap_proxy_config -p
PROXY DN: uid=proxy,ou=people,o=hp.com
```

The following command checks to see if the proxy user can bind to the directory:

```
./ldap_proxy_config -v
File Credentials verified - valid
```

5.10 Displaying the current profile

You can display the profile in use by any client by running `/opt/ldapux/config/display_profile_cache` on that client. The current profile is in the binary file `/etc/opt/ldapux/ldapux_profile.bin`.

```
cd /opt/ldapux/config
./display_profile_cache
```

You can also find out from where in the directory the client downloaded the profile by displaying the file /etc/opt/ldapux/ldapux_client.conf and looking for the line beginning with PROFILE_ENTRY_DN, for example:

```
grep ^PROFILE_ENTRY_DN /etc/opt/ldapux/ldapux_client.conf
PROFILE_ENTRY_DN="cn=profile1,ou=hpuxprofiles,o=hp.com"
```

5.11 Creating a new configuration profile

To create a new profile, run /opt/ldapux/config/setup. When `setup` asks you for the distinguished name (DN) of the profile, give a DN that does not exist and `setup` will prompt you for the parameters to build a new profile. The `setup` program also configures the local client to use the new profile.

Alternatively, you could use your directory administration tools to make a copy of an existing profile and modify it.

You can also use the interactive tool create_profile_entry to create a new profile as follows:

```
cd /opt/ldapux/config
./create_profile_entry
```

Once you create a new profile, configure client systems to use it as described in Section 5.13 (page 183).

5.12 Modifying a configuration profile

You can modify an existing profile directly using your directory administration tools, such as the HPDS Directory Server Console. For a description of the DUAConfigProfile object class, its attributes, and what values each attribute can have, see "LDAP-UX Client Services object classes" (page 349).

The `ldapentry` tool can also be used to modify the existing profile. This can be done with the following command:

```
DNPROFILE="/opt/ldapux/bin/ldapcfinfo -P | grep "^dn:" | cut -d" " -f
2-)"
$ /opt/ldapux/bin/ldapentry -m "$DN_of_profile"
$ cd /opt/ldapux/config
$ ./get_profile_entry -s nss
```

After modifying a profile, each client that regularly downloads its profile automatically will get the changes as scheduled. See Section 2.5.8 (page 113) for details.

5.13 Specifying a different profile for client use

Each client uses the profile specified in its start-up file /etc/opt/ldapux/ldapux_client.conf. To make a client use a different profile in the directory, edit this file and change the DN specified in the PROFILE_ENTRY_DN line. Then download the profile as described in Section 2.5.8 (page 113).

5.14 Changing from anonymous access to proxy access

If you have anonymous access and you want to change to using a proxy user, do the following:

1. Create the proxy user in the directory. With HP-UX Directory Server, you can use the Directory Server Console.
2. Change the `credentialLevel` attribute in your profile to be "proxy".

   If you want proxy access with anonymous access as a backup if proxy access fails, change credentialLevel to be "proxy anonymous".

3. Download the profile to the client. If you have an automated process to download the profile, you can wait until it executes. Or you can download the profile manually by running the following command:

   ```
   cd /opt/ldapux/config
   ./get_profile_entry -s nss
   ```

You can verify that the proxy user is configured with `display_profile_cache` and `ldap_proxy_config`. The `display_profile_cache` command displays the current configuration profile, including the credential level, which is either "proxy," "anonymous," or "proxy anonymous." the `ldap_proxy_config` command displays and verifies the proxy user the client is configured to use. See Section 7.2.4 (page 215), Section 7.2.6 (page 216), and Section 7.2.5 (page 215) for more information.

5.15 Changing from proxy access to anonymous access

If you are using proxy access and you want to change to using anonymous access, do the following:

1. Change the `credentialLevel` attribute in your profile to be "anonymous", using directory administration tools such as the HPDS Directory Server Console.
2. Download the profile to the client. If you have an automated process to download the profile, you can wait until it executes. Or you can download the profile manually as described in Section 2.5.8 (page 113).
3. Remove the proxy information:

   ```
   cd /opt/ldapux/config
   ./ldap_proxy_config -e
   ```

4. Optionally, remove the proxy user from the directory if you no longer need it. With HP-UX Directory Server, you can use the Directory Server Console.

5.16 Performance considerations

This section lists some performance considerations for LDAP-UX Client Services.For additional performance information, see the white paper *LDAP-UX Integration Performance and Tuning Guidelines* at:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX LDAP-UX Integration Software**.

5.16.1 Minimizing enumeration requests

Enumeration requests are directory queries that request all of a database, for example all users or all groups. Enumeration requests of large databases could reduce network and server performance. For this reason, you may want to restrict the use of commands and applications that enumerate.

The following commands generate enumeration requests:

- `finger` (see the *finger*(1) manpage)
- `grget`with no options (see the *grget*(1) manpage)
- `pwget` with no options (see the *pwget*(1) manpage)
- `groups` (see the *groups*(1) manpage)
- `listusers` (see the *listusers*(1) manpage)
- `logins` (see the *logins*(1M) manpage)
- All netgroup calls

In addition, applications written with routines of families such as the `getpwent`, `getgrent`, `gethostent`, and `getnetent` family of calls can enumerate a map, depending on how they are written.

5.17 Client daemon performance

Compared to previous networked name service systems, LDAP directory servers support a number of new features. And the general purpose nature of LDAP allows it to support a variety of applications, beyond those just used by a networked OS. Although directory servers have excellent performance and scalability, the addition of these features, such as security, means that directory applications will benefit from a design that considers performance requirements. In order to maximize of the number of HP-UX clients that can be supported by an LDAP directory server, and also improve client response, the ldapclientd daemon supports both data caching and persistent network connections. Their use, benefits and side-effects are described below.

5.17.1 ldapclientd caching

Caching LDAP data locally allows for much greater response time for name service operations. Caching means that data that has been recently retrieved from the directory server will be retrieved from a local store, instead of the directory server. Caching greatly reduces both directory server load and network usage. For example, when a user logs into the system, the OS typically needs to enquire about his/her account several times in the login process. This occurs as the OS identifies the user, gathers account information and authenticates the user. And further requests often occur as the account starts up new applications once a session is established. With caching, generally only one or two LDAP operations are required.

Caching is also critical to support certain types of applications that make frequent demands on the name service system, either because they are malfunctioning or need this specific type of information frequently.

ldapclientd also supports what is known as a negative cache. This type of cache is used to store meta-data about non-existent information. For example, if an application requests information about an account that does not exist, the directory server will not return an entry, and that negative result will be stored in a cache. Intuitively this type of cache would seem to be

un-necessary. However, applications exist that may perform these operations frequently, either on purpose or because they are malfunctioning. For example, if a file is created with a group ID that does not exist, every time a user displays information about this file, using the ls command, a request to the directory server will be generated.

The ldapclientd daemon currently supports caching of passwd, group, netgroup and automount map information. ldapclientd also maintains a cache which maps user's accounts to LDAP DNs. This mapping allows LDAP-UX to support groupOfNames and groupOfUniqueNames for defining membership of an HP-UX group.

Although there are many benefits to caching, administrators must be aware of the side-effects of their use. Table 5-4 shows some examples to consider:

Table 5-4 Benefits and side-effects for caching

Map Name	Benefits	Example Side-Effect
`passwd`	Reduces greatly the number of requests sent to a directory server during a login or other operation such as displaying files owned by that user.	Removing this information from the directory may not be visible to the operating system until after the cache has expired. In certain cases, this may allow a user to log in to an HP-UX host, even after his account has been removed from the LDAP directory server. (In general this is not a problem when PAM_LDAP is used for authentication, since authentication requests are not cached.)
`group`	Frequent file system access may request information about groups that own particular files. Caching greatly reduces this impact.	Removing a member of a group may not be visible to the file system, until after the cache expires. During this window, a user may be able to access files or other resources based on his/her group membership, which had been revoked.

Table 5-4 Benefits and side-effects for caching *(continued)*

Map Name	Benefits	Example Side-Effect
`netgroup`	netgroups can be heavily used for determining network file system access rights or user login rights. Caching this information greatly reduces this impact	Similar to groups, since netgroups are used to control access to resources, modification of these rights may not appear until after cache information has expired. Users may be allowed or denied login even their rights should allow / deny access, **NOTE:** Beginning with version 5.0 of the product, LDAP-UX Client Services supports integrated compat mode to control which users are visible on a host, where the user accounts are referenced by netgroups specified in the `/etc/passwd` file. As a means to greatly mitigate the performance impacts of compat-mode field masking, LDAP-UX has integrated compat mode support directly into `ldapclientd`, allowing caching of compat-mode user entries. For more information, see "Enabling integrated Compat Mode to control name services and user logins" (page 104)
`automount`	Frequent file system access to a directory may request automount information about a network file system. A positive AutoFS cache greatly reduces LDAP-UX Client response time while retrieving the automount data. Whenever a user attempts to access a directory that does not exist on the physical file system, the AutoFS system is called to determine if that directory is available via the network through AutoFS. A negative AutoFS cache is critical to assure that malfunctioning applications do not place redundant bogus requests on the directory server.	For the positive AutoFS cache, an alteration of the automount maps will sometimes not appear immediately. During this expiration window, a network file system may be granted access, when in fact the automount map should have unmounted from a network file system. For the negative AutoFS cache, an alteration of the automount maps will sometimes not appear immediately. During this expiration window, a user attempting to access a network file system may be denied access, when in fact the automount map should have set up a network file system mount.

NOTE: The `ldapclientd -f` command will flush all caches. For more information, see the *ldapclientd*(1M) manpage.

It is possible to alter the caching lifetime values for each service listed above, in the `/etc/opt/ldapux/ldapclientd.conf` file. See below for additional information. It is also possible to enable or disable a cache using the -E or -D (respectively) options. These options may be useful in determining the effectiveness of caching or helpful in debugging.

5.17.2 ldapclientd persistent connections

Since the HP-UX can generate many requests to an LDAP server, the overhead of establishing a single connection for every request can create excessive network traffic and slow response time for name service requests. Depending on network latency, the connection establishment and tear-down can cause relatively severe delays for client response. However, a persistent connection to the directory server will eliminate this delay.

In the ldapclientd daemon, a pool of active connections is maintained to serve requests from the Name Service Subsystem (NSS). If the NSS needs to perform a request to the directory server, one of the free connections in this pool will be used. If there are no free connections in the pool, a new connection will be established, and added to the pool. If system activity is low, then connections that have been idle for a specified period of time (configurable in the ldapclientd.conf file) then those connections will be dropped, to free up directory server resources. Aside from ldapclientd connection time-out configuration, it is also possible to define a maximum number of connections that ldapclientd may establish. Setting a high number of connections means assures that ldapclientd will not become a bottleneck in performing name service operations to the directory server. However, a high number of connections from a large number of HP-UX clients to the same directory server may exhaust all available connection resources on that directory server. Setting a low number of maximum connections will reduce that resource requirement on the directory server, but may create a performance bottleneck in the ldapclientd.

5.18 Troubleshooting

This section describes troubleshooting techniques as well as problems you may encounter.

5.18.1 Enabling and disabling LDAP-UX logging

When something is behaving incorrectly, enabling logging is one way to examine the events that occur to determine where the problem is. Enable LDAP-UX Client Services logging on a particular client as follows:

1. Edit the local start-up file /etc/opt/ldapux/ldapux_client.conf and uncomment the lines starting with #log_facility and #log_level by removing the initial # symbol. You can set log_level to LOG_INFO to log only unusual events. This is a good place to start. If LOG_INFO is not adequate to identify the problem, set log_level to LOG_DEBUG to log trace information. LOG_DEBUG will provide more information but will significantly reduce performance and generate large log files on active systems.
2. Edit the file /etc/syslog.conf and add a new line at the bottom:

   ```
   local0.debug <tab> /var/adm/syslog/local0.log
   ```

   where **<tab>** is the Tab key on your keyboard.
3. Restart the `syslog` daemon with the following command (for more information about this command, see the *syslogd*(1M) manpage):

   ```
   kill -HUP 'cat /var/run/syslog.pid'
   ```
4. Once logging is enabled, run the HP-UX commands or applications that exhibit the problem.
5. Disable logging by commenting out the log_facility and log_level lines in the start-up file /etc/opt/ldapux/ldapux_client.conf. Comment them out by inserting a "#" symbol in the first column.
6. Examine the log file at /var/adm/syslog/local0.log to see what actions were performed and if any are unexpected. Look for functions with "ldap_." These are standard LDAP function calls.

TIP: Enable LDAP logging only long enough to collect the data you need because logging can significantly reduce performance and generate large log files.

You may want to move the existing log file and start with an empty file: mv /var/adm/syslog/local0.log /var/adm/syslog/local0.log.save

5.18.2 Enabling and disabling PAM logging

When something is behaving incorrectly, enabling logging is one way to examine the events that occur to determine where the problem is. Enable PAM logging on a particular client as follows. For more information about PAM, see the *pam*(3) and *pam.conf*(4) manpages. In addition, see the document *Managing Systems and Workgroups: A Guide for HP-UX System Administrators* at the following location:

www.hp.com/go/hpux-core-docs (click **HP-UX 11i v2**)

1. Add the debug option to each line in `/etc/pam.conf` that contains libpam_ldap, for example:

   ```
   login account sufficient /usr/lib/security/libpam_unix.so.1
   login account required   /usr/lib/security/libpam_ldap.so.1 debug
   su    account sufficient /usr/lib/security/libpam_unix.so.1
   su     account required  /usr/lib/security/libpam_ldap.so.1 debug
   ...
   ```

WARNING! Enabling the debug option in pam.conf might allow hackers to gain additional information that would enable them to crack password security. For example, they could attempt to log in as a super user (su) and discover that a password has expired (observing the super user's behavior, the hackers could determine when he or she is likely to log in next).

2. Edit the file /etc/syslog.conf and add a new line at the bottom like the following:

   ```
   *.debug    <tab> /var/adm/syslog/debug.log
   ```

3. Restart the `syslog` daemon with the following command (for more information about this command, see the *syslogd*(1M) manpage):

   ```
   kill -HUP 'cat /var/run/syslog.pid'
   ```

4. Once logging is enabled, run the HP-UX commands or applications that exhibit the problem.
5. Restore the file /etc/syslog.conf to its previous state; otherwise, you may unintentionally enable logging in other applications.
6. Restart the `syslog` daemon with the following command (for more information about this command, see the *syslogd*(1M) manpage):

   ```
   kill -HUP 'cat /var/run/syslog.pid'
   ```

7. Remove the debug options from /etc/pam.conf.
8. Examine the log file at `/var/adm/syslog/debug.log` to see what actions were performed and if any are unexpected. Look for lines containing "PAM_LDAP."

TIP: Enable PAM logging only long enough to collect the data you need because logging can significantly reduce performance and generate large log files.

You may want to move the existing log file and start with an empty file: mv /var/adm/syslog/debug.log /var/adm/syslog/debug.log.save. Then restore the file when finished.

5.18.3 Directory server log files

You can view log files to see if any unusual events have occurred with your directory. The HP-UX Directory Server logs information to files under

`/var/opt/dirsrv/slapd-<`*`serverID`*`>/log`

where slapd-<*serverID*> is the name of your directory server.

The error logs contain start-up, shut-down, and unusual events. The access logs contain all requests. For more information, see the *HP-UX Directory Server administrator guide* for details.

5.18.4 User cannot log on to client system

If a user cannot log in to a client system, perform the following checks.

- To verify that NSS is working, you can use the `pwget -n` command (for more information, see the *pwget*(1) manpage) or the `nsquery`[2] command, as in the following examples:

  ```
  pwget -n username
  nsquery passwd username
  ```

  If the output shows LDAP is not being searched, check `/etc/nsswitch.conf` to make sure LDAP is specified. If *username* is not found, make sure that the user is in the directory and, if using a proxy user, make sure the proxy user is properly configured.

  If `nsquery` displays the user's information, make sure `/etc/pam.conf` is configured correctly for LDAP. If `/etc/pam.conf` is configured correctly, check the directory's policy management status. It could be the directory's policy management is preventing the bind

2. `nsquery` is a contributed tool included with the ONC/NFS product. For more information, see the *nsquery*(1) manpage.

because, for example the user's password has expired or the login retry limit has been exceeded. To check this try an `ldapsearch` command and bind as the user, for example:

```
cd /opt/ldapux/bin
./ldapsearch -h servername -b "basdDN" uid=username (get user's DN)
./ldapsearch -h servername -b "baseDN" -D "userDN" -w passwd \ uid=username
```

where *userDN* is the DN of the user who cannot log in and *username* is the login of the user. If you cannot bind as the user, check if any directory policies are preventing access.

See below for an example of determining the user's bind DN.

- Display the current configuration profile and check all the values to make sure they are as you expect:

  ```
  cd /opt/ldapux/config
  ./display_profile_cache
  ```

  In particular, check the values for the directory server host and port, the default search base DN, and the credential level. Also, if you have remapped any standard attributes to alternate attributes, or defined any custom search descriptors, make sure these are correct and exist in your database. If any of these are incorrect, correct them as described in Section 5.12 (page 183).

- If you are using a proxy user, make sure the configuration is correct as described in Section 5.8 (page 182).
- Make sure the client system can authenticate to the directory and find a user in the directory by searching for one of your user's information in the directory. Use the ldapsearch command and information from the current profile.

  If you are using a proxy user (determined by the credentialLevel attribute in the configuration profile), try searching for one of your user's information in the directory as the proxy user with a command like the following:

  ```
  cd /opt/ldapux/bin
  ./ldapsearch -h servername -b "baseDN" -D "proxyuser" -w \ passwd uid=username
  ```

  using the name of your directory server (from `display_profile_cache`), search base DN (from `display_profile_cache`), proxy user (from `ldap_proxy_config -p`), proxy user password, and a user name from the directory.

  For example:

  ```
  cd /opt/ldapux/bin
  ./ldapsearch -h sys001.hp.com -b "ou=people, o=hp.com" \
  -D "uid=proxyuser,ou=special users,o=hp.com" -w passwd \ uid=steves
  ```

  You should get output like the following:

  ```
  dn: uid=steves,ou=people o=hp.com
  uid: steves
  cn: Steve Sy
  objectclass: top
  objectclass: account
  objectclass: posixAccount
  loginshell: /bin/ksh
  uidnumber: 2875
  gidnumber: 191
  homedirectory: /home/steves
  gecos: Steve Sy, building 5, x50
  ```

  If you don't, your proxy user may not be configured properly. Make sure you have access permissions set correctly for the proxy user. See the steps "Create a proxy user" and "Set access permissions for the proxy user" in Section 2.4.4 (page 65) for details on configuring the proxy user.

  You can also try binding to the directory as the directory administrator and reading the user's information.

If you are using anonymous access, (determined by the value of the credentialLevel attribute in the configuration profile), try searching for one of your user's information in the directory with a command like the following:

```
./ldapsearch -h servername -b "o=hp.com" uid=username
```

using the name of your directory server (from `display_profile_cache`), search base DN (from `display_profile_cache`), and a user name from the directory.

You should get output similar to the previous example. If you don't, anonymous access may not be configured properly. Make sure you have access permissions set correctly for anonymous access. See the steps "Configure anonymous access" and "Set access permissions for anonymous access" in Section 2.4.4 (page 65) for details on configuring anonymous access.

- Enable PAM logging as described in Section 5.18.2 (page 189) then try logging in again. Check the PAM logs for any unexpected events.
- Enable LDAP-UX logging as described in Section 5.18.1 (page 189), then try logging in again. Check the log file for any unexpected events.
- If you are using HP-UX Directory Server, use the Directory Server Console to authenticate to the directory as the directory administrator. Check the ACIs for the proxy user. Make sure the proxy user or anonymous can view the following attributes listed. If not, change the ACI to allow this. Make sure all users can read their own information. If they cannot, change the ACI to allow this.

  Make sure all users have the following attributes and can read them:

  - `cn`
  - `loginshell`
  - `uid`
  - `uidnumber`
  - `gidnumber`
  - `memberuid`
  - `homedirectory`
  - `gecos`

6 Managing ssh host keys with LDAP-UX

LDAP-UX B.05.00 introduces management of host attributes in the directory server. One of the features integrated with host management is using an LDAP directory server as a trusted repository for a host's ssh public key.

ssh is a great protocol for both protecting data in transit (using encryption), and for validating trust between two parties. However, establishing that trust relationship is a weak aspect of the default ssh toolset. In order for two parties to securely communicate and identify each other, each must know a shared secret, known only to each other, or they must know some other piece of public information that can be used to prove the identity of the remote party. With ssh, both methods are often used, such as using public keys to identify remote hosts.

However, as with all secure methods of communication, how are these secrets or public keys initially shared? There's always a bootstrapping problem to pre-establish trust between parties. The base ssh toolset leaves this exercise to the end users. In some organizations, administrators can attempt to pre-distribute public keys of hosts within their organizations. But this often leads to a scalability problem as the number of hosts in an organization increases. And as more services are moving to virtualized hosts, this can become a significant cost to manage.

With LDAP-UX B.05.00, ssh key management can be centralized in a trusted directory server, eliminating the need for end users to make decisions about the trustworthiness of a remote host and greatly mitigating the scalability issue, compared with distributing keys manually.

6.1 Overview

The following sections provide an overview of managing ssh host keys with LDAP-UX.

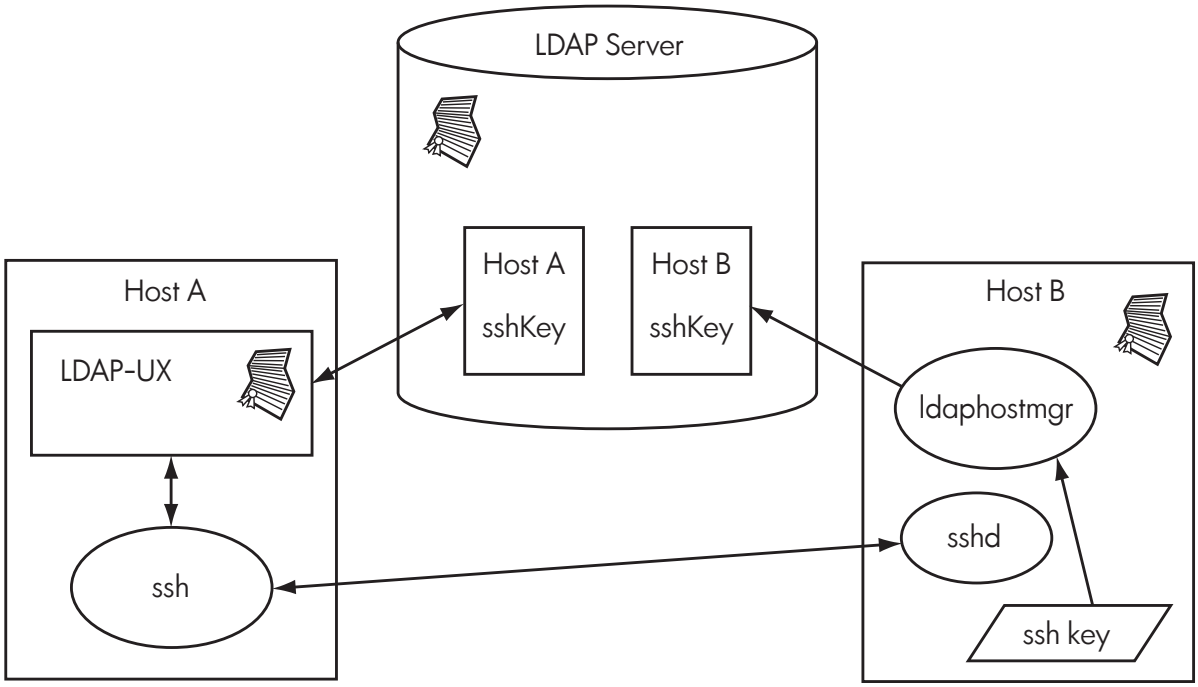
6.1.1 How it works

As previously mentioned, in a basic ssh deployment, each user must to determine if a remote host should be trusted. When establishing a session with a remote host for the first time, the user is presented with a prompt. This prompt displays a "fingerprint" for the remote host's public key and asks if the user still wants to connect, and if the key should be trusted and placed in the user's personal `known_hosts` file. Given the average user's motivation to continue working and limited ability to determine if the remote host's fingerprint is correct, users frequently just reply *yes* to the prompt, uncertain if the remote host is the true host, or if there's a risk of a man-in-the-middle attack.

Starting with LDAP-UX B.05.00 and HP Secure Shell A.05.50 or higher, this burden on the end user is removed. By managing host and public key information in the directory server, ssh itself can verify the correctness of the remote public key, and therefore determine if a trusted connection can be established. And given that private information often travels across this connection, that trust is critical.

When LDAP is used as a repository for managing ssh host keys, the infrastructure shown in Figure 6-1 (page 194) is established:

Figure 6-1 ssh host key management infrastructure



The LDAP directory server includes an SSL certificate. The LDAP-UX library of Host A has a copy of that certificate. When ssh attempts to validate the public key of the remote host Host B, it connects through a library in LDAP-UX. LDAP-UX is configured to securely communicate with the LDAP directory server and to discover keys for the requested hosts. LDAP-UX utilities such as `ldaphostmgr` and `ldaphostlist` can be used to manage those keys in the directory server, from any host configured with LDAP-UX (such as Host B, in the figure). Those utilities can also manage information about any remote host, including the ability to replace or update its keys.

6.1.2 Secure framework

For ssh to determine if the remote host is trusted, ssh must know about the remote host's private key so it can compare that key with the key presented when ssh connects with the remote host. The toolset normally stores these keys in either a host-local `known_hosts` file (`/opt/ssh/etc/ssh_known_hosts`) or in the user's personal `known_hosts` file. To avoid allowing users to make decisions whether a remote host should be trusted, some administrators try to pre-distribute these keys periodically to the host-local ssh `known_hosts` file. However, this process encounters scalability problems as the number of hosts grows.

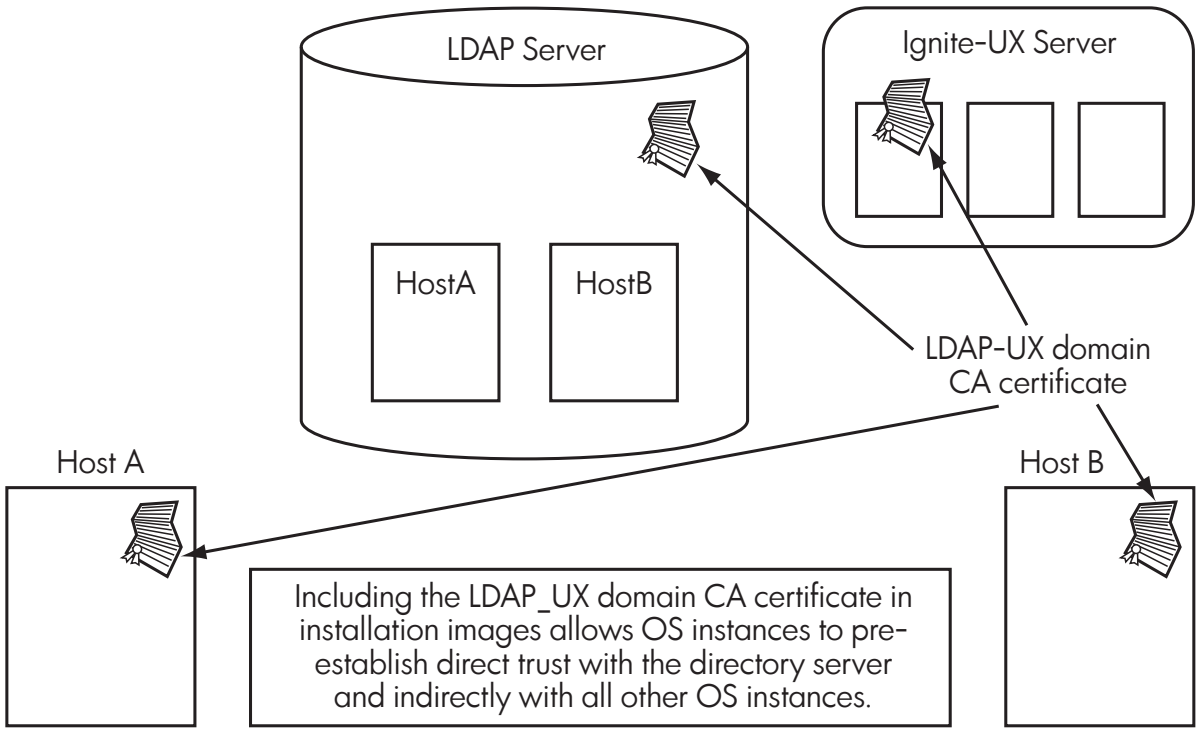
To eliminate this distribution process, the LDAP directory server can be used to store and manage host public keys in a central repository. And LDAP-UX offers tools to manage this information, either centrally or on each host being managed.

Because the LDAP repository contains the public keys of the hosts, the LDAP directory server itself must be trusted to assure that the user can trust the remote host's identity. And the information stored in that directory server must also be trusted. Fortunately, LDAP directory servers meet this requirement well. LDAP directory servers have authentication and access control frameworks that can be used to protect data managed in the directory server and help assure its validity. And LDAP directory servers also support the SSL/TLS protocol, which can not only be used to protect communication with the directory server but, more importantly, to assure the integrity of the data transmitted from the directory and validate the identity of the directory server itself. While a CA (certificate authority) certificate, or a certificate of the directory server itself, is still required to be distributed to each host, distribution of a single CA certificate is a much more manageable task. Instead of every user on every host having to validate trust with every other host connected to, each host needs to trust only one thing: the directory server.

With the LDAP-UX guided installation, and the HP-UX Directory Server, setting up this trust framework is nearly automatic (for more information about this trust framework, see Section 2.3.2.3 (page 33)). When using the guided installation, LDAP-UX generates a server certificate software depot file. This depot file can be installed on each host being managed, and once installed, will establish trust with that central directory server.

As a depot file, this certificate can be pre-distributed as part of an OS installation image, combining the installation and trust setup processes into a single step. In Figure 6-2 (page 195), an HP-UX Ignite server is shown with an HP-UX image and CA certificate. This certificate is distributed automatically to all hosts (this figure shows hosts named Host A and Host B) to establish trust with the LDAP directory server shown. This directory server stores and manages the host public keys for Host A and Host B.

Figure 6-2 ssh host key management trust framework



LDAP-UX uses the `sshPublicKey` attribute as part of the `ldapPublicKey` objectclass to manage ssh public keys in the directory server. The `ldapPublicKey` objectclass is an auxiliary objectclass, which can be attached to host entries in the LDAP directory server. Because hosts accessible through the ssh protocol have an IP address, the `ipHost` structural objectclass is used to instantiate this host information in the directory server.

The following example shows an example of a host entry, displayed in LDIF format:

```
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
objectClass: top
objectClass: device
objectClass: ldapPublicKey
objectClass: iphost
objectClass: domainEntity
sshPublicKey: ssh-rsa AAAAB3Nza...
sshPublicKey: ssh-dss AAAAB3Nza...
sshPublicKey: 1024 35 140898...
owner: uid=domadmin,ou=people,dc=mydomain,dc=example,dc=com
ipHostNumber: 16.92.96.116
cn: hptem079
```

6.1.3 Permissions

The LDAP-UX host management tool (`ldaphostmgr`), which is used to manage ssh public keys in the directory server, manipulates the aforementioned object classes and attributes. This tool relies on the directory server to provide proper access control. To assure that only authorized modifications to the host and public key information is performed, only a restricted set of privileged users should be allowed to modify host information, including the `sshPublickKey` attribute. If you have used the guided installation and, as part of that setup process, created a new HP-UX Directory Server Instance, these access controls are automatically created (for more information about the access controls established by the guided installation, see Section 2.3.2.3 (page 33)). Several sets of users are considered privileged enough to manipulate host information in the directory server, including the `DomainAdmins` group, the `HostAdmins` group, or the owners (owners are any users or members of a group that are listed in the `owner` attribute in the host's entry.) These users, or any user that has rights to manipulate the `sshPublicKey` attribute for the host in the directory server will be granted permission on the HP-UX host to change the ssh key pairs of the host. Normally the permission to modify the host's public and private ssh keys is restricted to the root user. However, the `ldaphostmgr` will elevate its privilege to allow non-root users to modify a host's public key if that user has permission to modify the `sshPublicKey` attribute for the current host.

If a user runs the `ldaphostmgr` tool and attempts to change a host's ssh key, `ldaphostmgr` will verify if the user has the right to modify the `sshPublicKey` for that host. If the directory server rejects this modification, `ldaphostmgr` will not elevate its privilege and not modify the host's ssh key.

6.1.4 Distributed management (manage from any host)

Remote management is an important feature of the `ldaphostmgr` tool. Specifically, if LDAP-UX version B.05.00 or later is installed on a remote host that is part of the same LDAP-UX domain (subscribes to the same LDAP-UX configuration profile) as the current host, it is possible to remotely manage ssh keys on that host. As long as the current user has permissions to log in to the remote host and to manipulate the `sshPublicKey` attribute, the `ldaphostmgr` tool can change the key of any host in the LDAP-UX domain from any other host. This remote management is handled within `ldaphostmgr` itself. The user need not remotely log in to the host to manage it.

However, this means that any user with permission to manage the `sshPublicKey` attribute, must also be a user with POSIX attributes attached (the posixAccount object class), such that the HP-UX OS will allow remote login for this user. See Section 6.2.6 (page 199) for additional details on setting up an ssh key manager account.

6.2 Setting up the key management domain

The first step in setting up an ssh key management domain is to establish the host and key data repository. This repository must be an LDAP directory server and must meet the security requirements previously defined, and explained in additional detail in the subsections that follow. If you have not already targeted a directory server to act as this repository, you should consider using the LDAP-UX guided installation (`autosetup`), which will automatically create a new directory server instance, if desired. This directory server instance will create a default security and management framework. For more information about the guided installation, see Section 2.3 (page 23).

The remaining subsections describe this process, summarized as follows:

- Identify a directory server and a location in that directory server where host and key data will be stored.
- Assign and set up an SSL certificate for the directory server, so that trust can be established between clients and the directory server.

- Define authentication and access control, such that a limited set of privileged users will have the ability to manage host and ssh key data in the directory server.
- Install a CA or server certificate in the `/etc/opt/ldapux/cert8.db` file. This can be done using `/opt/ldapux/bin/certutil`, or by installing the auto-generated LDAP-UX domain CA depot (created with the guided installation).
- Configure LDAP-UX on all host clients. ssh communicates through LDAP-UX to securely connect with the directory server and discover the location of the host and key information. Be sure to configure SSL, which is the default with a guided installation.
- Configure the ssh client to use LDAP for key discovery. This simply requires enabling the feature by setting flags in the `/opt/ssh/etc/ssh_config` and `/opt/ssh/etc/sshd_config` files. HP Secure Shell A.05.50 or higher is required.
- Optionally, define a central ssh configuration, using the LDAP-UX central configuration service. This service will allow you to centrally manage ssh configuration and will distribute that configuration to all LDAP-UX enabled clients that are part of the same LDAP-UX domain.

6.2.1 Host repository

To centrally manage host keys, a central repository must be defined to store that information. For clients to discover host keys, they must know the location of the repository as well as trust that repository and the data managed within it. The following subsections describe how ssh clients can identify and trust that repository.

6.2.2 Data Location

To centrally manage ssh keys for hosts, information about these hosts must be stored in the directory server. Choose a location in the directory information tree (DIT) where these hosts will reside. That location should be within the scope of the base DN specified when you configured LDAP-UX (see "Installing and configuring LDAP-UX Client Services" (page 21)). If you have used a guided installation to create your directory server instance, that location will be `ou=hosts,`*`suffix`*, where *`suffix`* is the root of your directory server data. If you have already configured LDAP-UX, then you can use the `ldapcfinfo` tool to determine where LDAP-UX believes host information should be located.

Example:

```
# /opt/ldapux/bin/ldapcfinfo -t hosts -b
ou=Hosts,dc=mydomain,dc=example,dc=com
```

If the base DN discovered above is not the desired location within the directory tree, you can reconfigure the LDAP-UX profile using the steps outlined in Section 5.12 (page 183).

6.2.3 Trust

Trust between an ssh client and a remote host depends on the ability of the client to validate the identity of the remote host, and vice versa. As previously mentioned, this requires the ssh client to have access to the public key of the remote host. It needs that key to validate the key presented by the remote host when the initial connection is made. Traditionally, this key is stored in the `known_hosts` file on the local client's host. This file is under the control of the user himself, in the `~/.ssh/known_hosts` file, or a system administrator might have put a list of hosts in the `/opt/ssh/etc/ssh_known_hosts` file. This file is assumed to be secured. Proper files system permissions are set to prevent unauthorized modification.

When the ssh public key is managed in the directory server, the integrity of the `sshPublicKey` attribute must be assured to achieve that same level of trust. Just as the user can trust his own `known_hosts` file or the ssh `known_hosts` file set up by his administrator, the user must also be able to trust the integrity of the `sshPublicKey` attribute when managed in the directory server. Trusting the ssh key repository requires that the identity of the directory server can be validated, the data in the directory server cannot be modified by unauthorized users, and the

data transmitted between the client and the directory server is protected. The following three sections describe how to establish this trust.

6.2.4 Validating directory server identity

Just as a web browser uses SSL and SSL CA certificates to identify the validity of a remote web server when verifying that a user is sending credit card information to a legitimate organization instead of an impostor, the LDAP directory server can use the same SSL protocol and certificates to validate the identity of the directory server. To establish this trust, a directory server must have a valid signed server certificate, and the client must have a copy of the public portion of that server certificate, or a CA (Certificate Authority) certificate of the CA that signed the server's certificate. When using the guided installation script to create a new HP-UX Directory Server instance, LDAP-UX automatically creates a CA certificate and server certificate for that directory server instance. The CA certificate is deposited into an SD depot file that can be pre-installed on any HP-UX client. For more information about this depot file see Section 2.3.2.3.3 (page 35). . If you have this depot file, you can install this package on your host with the following command:

```
# /usr/sbin/swinstall -s hostname:/depot/name LDAPUX-DOMAIN-CA
```

If you have your own CA certificate (not created using the guided installation), you can install that CA certificate in the `/etc/opt/ldapux/cert8.db` file as in the following example:

```
# more /tmp/mycacert.txt
-----BEGIN CERTIFICATE-----
MIIBlzCCAQCgAwIBAgICBKIwDQYJKoZIhvcNAQEFBQAwETEPMA0GA1UEAxMGQ0Fj
ZXJ0MB4XDTEwMDEwODIxNDA0OVoXDTIwMDEwODIxNDA0OVowETEPMA0GA1UEAxMG
Q0FjZXJ0MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC8yROkmsMiCIhgki2V
Sk3iJr2SXATnvp/Bmn5p9i2PVjlk63rvJFvyEDYM40TxZmArptMXq4WsfAy4fOMB
KY+yJKK53qc+fQ8k5YERL93RWugBs7SqVhNOtWRPZWcBUNHM7tCywtI1RzbZn8sp
hAOofPPiGVvmhUYrk05Y6UY07wIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAGhaRyvK
Qx5EoIVzE6iVdlEU0wopr33qVpHyEYoqDdziy0cLhqGbVKEzCaM83sdW7S9Cxe87
pr+31xbgJgXbO7kNP3cZytNcd1A7Grc0EmVb8yck+uWx7Oj2CYkac0DboWAn/uhU
hxbIaqFdC+gninQ9ECNEnUFt0hJ6SSNhYWUW
-----END CERTIFICATE-----

# /opt/ldapux/contrib/bin/certutil -A -d /etc/opt/ldapux -a -n "my CA Certificate" -a -t "CT,," < /tmp/mycacert.txt
```

Attempting to use ssh key management without using SSL provides little value, because if the directory server can be impersonated, then the validity of the `sshPublicKey` attribute cannot be trusted, and thus the identity of any remote ssh hosts cannot be validated.

Configuring the directory server with a server certificate also allows it to use the Secure Sockets Layer protocol (SSL). This protocol allows information in transit to be protected from eavesdropping, but even more importantly, from tampering by a man-in-the-middle. Support for SSL meets two of the previous requirements to assure integrity of the sshPublicKey. And when LDAP-UX is configured using the guided installation, SSL is automatically configured. (For more information about the guided installation, see Section 2.3 (page 23).)

6.2.5 Authentication and access control

To assure its integrity, the `sshPublicKey` attribute must be protected from unauthorized modification. LDAP directory servers have the inherent ability to authenticate users before allowing access and to limit operations performed on the LDAP data with access-control policies.

As mentioned previously, any user with permission to modify the `sshPublicKey` attribute for a particular host can also change the ssh key pair of that host using `ldaphostmgr`. This means that permission to modify the `sshPublicKey` attribute must be restricted to trusted administrators. The trust relationship between users and hosts is based on the ability to protect the integrity of the `sshPublicKey` attribute in the directory server.

To allow for management of the sshPublicKey, you need to grant rights to a group of administrators. This process is different for each directory server deployment because access control features of directory servers are different and have not yet been standardized. For the HP-UX Directory Server (the Red Hat Directory Server and Sun Java Directory Server are similar), the `ACI` attribute must be used to define this policy. The following example shows how anyone listed as an owner of a host, a Domain Administrator, or host administrator is allowed to modify

the sshPublicKey attribute. This ACI is automatically created if you create a new directory server instance using the guided installation.

```
dn: dc=mydomain,dc=example,dc=com
aci: (targetattr = "*")(version 3.0;acl "[DOMAINADMIN:ALL:ALL]: Allow changes
 by Domain Administrators";allow (all) (groupdn = "ldap:///cn=DomainAdmins
 ,ou=Groups,dc=mydomain,dc=example,dc=com");)

dn: ou=Hosts,dc=mydomain,dc=example,dc=com
aci: (targetattr = "sshPublicKey || ipHostNumber") (version 3.0;acl "[OWNER:WR
 ITE:HOSTOWNERATTRS]: Allow owner modification of host information";allow (re
 ad,compare,search,write,delete,add) userattr = "owner#USERDN";)
aci: (targetattr = "objectclass || cn || dn || owner || host || ipHostNumber |
 | ipNetmaskNumber || ipNetworkNumber || ipProtocolNumber || ipServicePort ||
  ipServiceProtocol || sshPublicKey || oncRpcNumber || userPassword || userCe
 rtificate" )(version 3.0;acl "[HOSTADMIN:READ-WRITE:HOSTATTRS]: Allow change
 s to Unixattributes by Host Administrators";allow (all) (groupdn = "ldap:///
 cn=HostAdmins,ou=Groups,dc=mydomain,dc=example,dc=com");)
```

6.2.6 Administrative users

Any user with the right to modify the sshPublicKey attribute for a host is considered an ssh key administrator. As seen from the rights in the previous example, anyone that is a member of the DomainAdmins or HostAdmins groups or is listed as the owner (the owner attribute has the DN of the user), is considered an ssh key administrator. As mentioned previously, to protect the integrity of the sshPublicKey attribute, this list of users should be restricted to trusted administrators.

In addition to creating a trusted list of administrators, ldaphostmgr allows for management of keys not only on the local host, but also on any remote host that is a member of the same LDAP-UX domain (uses the same LDAP-UX configuration profile). However, for remote administration to function, the administrators' accounts must also be assigned POSIX account attributes (this is not required if remote administration is not desired.)

You can create an administrator that has the rights to manage ssh public keys using the ldapugadd and ldapugmod utilities, as in Example 6-1the following example:

Example 6-1 Creating an administrator that has the rights to manage ssh public keys

1. Create the new account using ldapugadd:

```
# /opt/ldapux/bin/ldapugadd -P -f "Alice Bobson" abobson Surname=Bobson
# /opt/ldapux/bin/ldapuglist -n abobson
dn: uid=abobson,ou=people,dc=mydomain,dc=example,dc=com
cn: Alice Bobson
uid: abobson
uidNumber: 3840
gidNumber: 20
loginShell: /usr/bin/sh
homeDirectory: /home/abobson
gecos: Alice Bobson
```

2. Add the user to one of the privileged groups (HostAdmins in this case):

```
# /opt/ldapux/bin/ldapugmod -P -t group -a abobson HostAdmins
# /opt/ldapux/bin/ldapuglist -t group -n HostAdmins
dn: cn=HostAdmins,ou=Groups,dc=mydomain,dc=example,dc=com
cn: HostAdmins
memberUid: domadmin
memberUid: abobson
```

If you already have users that are considered administrators, but do not have posixAccount information attached to their directory server entries, you can use the ldapugmod command to extend their accounts with POSIX attributes. The following example shows how to extend posixAccount attributes to an existing user:

Example 6-2 Extending administrator accounts with posixAttributes

1. Identify the account to extend:

```
# /opt/ldapux/bin/ldapuglist -F "(cn=bob alison)" \*
dn: cn=Bob Alison,ou=people,dc=mydomain,dc=example,dc=com
cn: Bob Alison
gecos: Bob Alison,+1-303-555-5432
```

2. Add posixAccount attributes using the -O option of `ldapugmod`:

```
# /opt/ldapux/bin/ldapugmod -P -O -n balison -u 1234 -g users -d /home/balison \
  -s /usr/bin/sh -D "cn=Bob Alison,ou=people,dc=mydomain,dc=example,dc=com"
# /opt/ldapux/bin/ldapuglist -n balison \*
dn: cn=Bob Alison,ou=people,dc=mydomain,dc=example,dc=com
cn: Bob Alison
uid: balison
uidNumber: 1234
gidNumber: 20
loginShell: /usr/bin/sh
homeDirectory: /home/balison
gecos: Bob Alison,+1-303-555-5432
```

If Bob Alison is not already a member of a privileged group, then you can add him as a member of the Host Administrators group, using a similar command as in the previous example:

```
/opt/ldapux/bin/ldapugmod -t group -P -a balison HostAdmins
```

NOTE: In the previous examples, the `HostAdmins` group is a posixGroup. By default, the `ldapugmod` tool only works with posixGroups. However, you can still use `ldapugmod` to modify non-posixGroups if your LDAP-UX profile specifies LDAP-style attribute mapping for LDAP-style groups, and you use the `-D` option to specify the full DN of the group you want to manage.

If you use groupOfUniqueNames for your LDAP-style groups, then your attribute mapping for group membership as defined in the LDAP-UX configuration profile should be:

```
attributemap: group:memberUid=uniqueMember member memberUid
```

If you use groupOfNames for your LDAP-style groups, then your attribute mapping for group membership as defined in the LDAP-UX configuration profile should be:

```
attributemap: group:memberUid=member uniqueMember memberUid
```

To modify a non-posixGroup, you need to use the `-D` option when specifying the group to modify. For example, assume in the following that `cn=Host Administrators` is a groupOfNames, but not a posixGroup. It is possible to add balison as a member using the above attributeMap and the following command:

```
/opt/ldapux/bin/ldapugmod -t group -P -a balison \
   -D "cn=Host Administrators,ou=Groups,dc=mydomain,dc=example,dc=com"
```

6.3 Managing keys in the directory server

If you have not yet set up a directory server to manage your host information, you can use the LDAP-UX guided installation to create a new directory server and configure LDAP-UX to manage hosts in that directory server. The guided installation sets up an environment that meets the host repository requirements described in the previous section.

After you establish a repository and security framework for your host information, as described in the previous section, you can begin to manage those hosts. The remainder of this section describes how to properly configure HP-UX hosts to use the central repository for ssh keys and how to manage the hosts and their keys.

6.3.1 Configuring ssh and sshd to use LDAP-managed keys

On each HP-UX client that is to use LDAP-based ssh public keys, you must install version A.05.50 or higher of the HP Secure Shell product and LDAP-UX version B.05.00 or later. HP Secure Shell A.05.50 or higher is enabled to use the LDAP directory server for public key validation and is dependent on APIs provided in LDAP-UX B.05.00.

You must configure the ssh toolset to use LDAP. To do this, configure the following two new parameters in the `ssh_config` file:

- `UseLdapHostKey`

  Directs the ssh client tools (ssh, scp, sftp) to use the LDAP repository to discover a remote host's public key, if that key is not already found in the known_hosts file.

- `UpdateKeyFromLdap`

  Directs the ssh client tools to update the `known_hosts` file if the key for the specified host does not exist or is incorrect. The key from the LDAP directory server is assumed to be correct, based on the previously described trust agreements between the ssh client and the directory server. If the local user has a key that does not match the one found in the directory server file, the ssh client replaces it in the user's personal `known_hosts` file. Using the `UpdateKeyFromLdap` option allows the user's `known_hosts` file to act as a local cache for the information in the directory server.

NOTE: If you want the ability to revoke or remove keys for hosts (in case those keys are compromised), do not enable the `UpdateKeyFromLdap` option. See Section 6.3.8 (page 206) for additional information.

In the `sshd_config` file, only the UseLdapHostKey option is available. This option has the same effect as in the `ssh_config` file. It is used when administrators want to configure host-based authentication, using the HostBasedAuthentication option. In this case, sshd uses the LDAP directory server to validate the identity of a remote host on an incoming connection. (See Section 6.2.5 (page 198)).

With LDAP-UX B.05.00 or later, it is possible to centrally manage ssh and sshd configuration parameters using the LDAP-UX central configuration service; for more information, see Section 6.5 (page 207).

After completing this step, you have completed the setup process and can now begin to manage keys for hosts using the steps described in the following subsections.

6.3.2 Adding keys for HP-UX hosts

Use the `-k` option of the `ldaphostmgr` command to add or manage public keys for hosts. There are several ways to add or change ssh public keys in the directory server using this option. This section and the sections that follow describe these various methods.

If you use the guided installation when configuring LDAP-UX on a host, during the configuration process the current host and its RSA public key are automatically added to the directory server. You can display the entry for the current host using the following commands:

```
chef(): ldaphostlist -k -n "$(hostname)"
dn: cn=chef,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: chef
cn: chef.mydomain.example.com
ipHostNumber: 16.92.96.225
sshPublicKey: ssh-rsa AAAAB...== BEGIN-KM creationtime=20100413173637Z END-KM
```

Notice in the above command sequence that keys managed by `ldaphostmgr` have an extended field within the comment structure of the public key data. This extended field can be used to determine key age and keep track of expiration information if desired. See Section 6.4.2 (page 207) for additional information.

If you did not configure LDAP-UX on the current host using the guided installation, you might not have an entry in the directory server that represents the current host. In that case, you can add the host using the `-a` option of the `ldaphostmgr` command as follows:

```
brewer(): id
uid=8507(domadmin) gid=220(ldap) groups=88(DomainAdmins)
brewer(): ldaphostmgr -a -f -k rsa "$(hostname)"
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
brewer(): ldaphostlist -k -n "$(hostname)"
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: brewer
cn: brewer.mydomain.example.com
ipHostNumber: 16.92.96.225
sshPublicKey: ssh-rsa AAAA...== BEGIN-KM creationtime=20100423234903Z END-KM
```

In this example, the `-a` option is used to indicate that the host should be added as a new entry to the directory server. The `-f` option indicates that the fully qualified domain name should be added. And the `-k` option indicates the RSA (protocol version 2) key should be added. Other key types can be used. The `-k` option also accepts `rsa1`, `dsa`, and the `all` key-type, which means add/modify all three key types.

NOTE: Whenever you add a new host to the directory server that will contain sshPublicKeys, you must use the `-f` option to add the fully qualified domain name (FQDN) for the host, if the FQDN has not already been set. The ssh toolset uses network naming services (typically DNS) to determine the host name of IP addresses for hosts. In so doing, it resolves to a fully qualified domain name, which ssh needs to validate in the directory server. Notice that in the previous example, you can see the `cn` attribute listed twice, once with the short name and once with the FQDN.

The `ldaphostmgr` and `ldaphostlist` tools provide a smoother user interface for entering user credentials when used by accounts that have posixAccounts managed in the directory server. For the purposes of this demonstration, the `domadmin` user is used, which is created by default when a new directory server instance is created using the guided installation.

When `ldaphostmgr` is used to add a new host, it determines the location to add the host using the LDAP-UX configuration profile. By default, when using a guided installation, this location is `ou=Hosts,`*`defaultBaseDN`*. You can use the `ldapcfinfo` command to determine the location that `ldaphostmgr` will use:

```
# /opt/ldapux/bin/ldapcfinfo -t hosts -b
ou=Hosts,dc=mydomain,dc=example,dc=com
```

See Section 6.2.2 (page 197) for additional information. If you wish to place the host in a different location of the directory server tree, you can use the -B option.

While `ldaphostmgr` can be used to add the current ssh public keys of the local host, it is also possible to add keys of other remote HP-UX hosts managed by LDAP-UX that are in the current LDAP-UX domain. Just specify the name of the remote host; however, if `ldaphostmgr` has no way to identify the remote host, it displays an ssh-like warning message to indicate this:

```
chef (): ldaphostmgr -a -f -k rsa baker
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
WARNING: The identity of the host "baker" could not be verified.
SSH key fingerprint: b4:2f:45:c2:b0:17:a2:7b:a0:a7:88:61:a9:36:f2:4c.
The SSH key for the remote host is unknown.  This host's key is currently not
managed in the directory server and should be positively identified before
adding this key to the directory server. Once added, this key will be
trusted by all other LDAP-enabled ssh clients.  Using ldaphostmgr on the
remote host, instead of adding this key remotely, will avoid generating
this warning message. Do you wish to trust this key (y/n)?: n
ERROR:    HST_UNTRUSTED_REMOTE_HOST:
          The identity of the host "baker" could not be verified. SSH key
```

```
fingerprint: b4:2f:45:c2:b0:17:a2:7b:a0:a7:88:61:a9:36:f2:4c. The SSH
key for the remote host is unknown and is not trusted.
```

If you remotely log in to the host, and can positively identify the host, you can add the host using `ldaphostmgr` as originally demonstrated. Or, if you have the ssh public key of the remote host in a local `known_hosts` file, the above message will not be displayed. If you can positively identify the fingerprint of the remote host, you can answer yes (y) to the WARNING message. Key fingerprints for the local host can be displayed using the `ssh-keygen` command:

```
baker (): ssh-keygen -l -f /opt/ssh/etc/ssh_host_rsa_key.pub
2048 b4:2f:45:c2:b0:17:a2:7b:a0:a7:88:61:a9:36:f2:4c /opt/ssh/etc/ssh_host_rsa_key.pub (RSA)
```

6.3.3 Adding keys for non-HP-UX hosts or devices

Not all hosts and devices that support the ssh protocol in your network will be HP-UX systems. You can use LDAP-UX and the LDAP directory server to manage keys for those hosts, but to assure key integrity, an out-of-band process is required to verify the public key of those devices. There are two methods to do this. In the first method, the public key for a remote device can be provided directly to `ldaphostmgr` in a file:

```
chef (): ldaphostmgr -a -k /tmp/router1_ssh_host_rsa_key.pub router1.mydomain.example.com
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
chef (): ldaphostlist -k -n router1.mydomain.example.com
dn: cn=router1,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: router1.mydomain.example.com
ipHostNumber: 192.0.32.1
sshPublicKey: ssh-rsa AAAAB...== BEGIN-KM creationtime=20100427000132Z END-KM
```

The file provided to `ldaphostmgr` must contain the ssh public key for the remote host/device, and be in the ssh standard public key file format, as found in the `/opt/etc/ssh/ssh_host_rsa_key.pub` file. This format contains the following three fields separated by spaces: *`key-type`*, *`base64-key`*, and *`comments`*.

The second method is to let `ldaphostmgr` automatically discover the public key for the remote host/device. In this case, the `-k` option is used with the `^` flag:

```
chef (): ldaphostmgr -a -k ^rsa router2.mydomain.example.com
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
WARNING: The identity of the host "router2.mydomain.example.com" could not be verified.
SSH key fingerprint: 74:ed:80:36:f9:3f:30:29:11:43:31:ea:27:3f:3b:13.
The SSH key for the remote host is unknown.  This host's key is currently not
managed in the directory server and should be positively identified before
adding this key to the directory server. Once added, this key will be
trusted by all other LDAP-enabled ssh clients.  Using ldaphostmgr on the
remote host, instead of adding this key remotely, will avoid generating
this warning message. Do you wish to trust this key (y/n)?: y
chef (): ldaphostlist -k -n router2.mydomain.example.com
dn: cn=router2.mydomain.example.com,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: router2.mydomain.example.com
ipHostNumber: 192.0.32.2
sshPublicKey: ssh-rsa AAAAB...UlQ== BEGIN-KM creationtime=20100427000942Z END-KM
```

However, with this method, and as indicated by the WARNING prompt, `ldaphostmgr` has no means to verify the validity of the remote host's or device's public key. An out-of-band method must be used to verify the key fingerprint before accepting the key for the specified device, unless other means are available to assure the trust between the local host and the remote host or device.

6.3.4 Adding keys in a batch

You might already be managing and distributing an ssh `known_hosts` file, such as the one found at `/opt/ssh/etc/ssh_known_hosts`. This file contains four fields: *`host-name`*, *`key-type`*, *`base64-key`*, and *`comments`*. However, the *`host-name`* field may be . If your `ssh_known_hosts` file does not have host names, then use the following shell script to add all the keys from the `ssh_known_hosts` file to the directory server automatically.

NOTE: Because this script runs in batch mode, you need to specify the LDAP host administrator's credentials in the LDAP_BINDDN and LDAP_BINDCRED environment variables before running the script (or, alternatively, use the `-E` option to specify those values in a file.)

```
KNOWN_HOSTS_FILE="ssh_known_hosts"
### grep out comments and blank lines
grep -v -e "^[[:space:]]*$" -e "^[[:space:]]*#" \
  "$KNOWN_HOSTS_FILE" > /tmp/myknownhosts$$
exec 4< /tmp/myknownhosts$$
while read pubkey <&4
  do
  hostname="$(echo "$pubkey" | cut -d" " -f 1)"
  keydata="$(echo "$pubkey" | cut -d" " -f 2-)"
  if ( /opt/ldapux/bin/ldaphostlist -n "$hostname" | grep -qi "^dn: " )
    then
    hostop="-m"
  else
    hostop="-a"
  fi
  echo "$keydata" > /tmp/keyfile$$
  /opt/ldapux/bin/ldaphostmgr $hostop -X -f -k /tmp/keyfile$$ "$hostname"
done
rm -f /tmp/keyfile$$
rm -f /tmp/myknownhosts$$
```

6.3.5 Changing keys for HP-UX hosts

If you believe the private key for a host has been compromised, you can change the keys of that host with `ldaphostmgr`. From that host, run the `ldaphostmgr` command with the `-k` option. If the user has privilege to modify the `sshPublicKey` attribute, `ldaphostmgr` will elevate that privilege to allow a non-root user to modify the host's public and private key files `/opt/ssh/etc/ssh_host_rsa_key` and `/opt/ssh/etc/ssh_host_rsa_key.pub`). `ldaphostmgr` will also update the directory server with the new public keys for this host:

```
baker (): ldaphostmgr -k all baker
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
The public key(s) already exists in LDAP server, do you want
to replace it [y/n]? y
```

In this example, the `all` key-type was specified to change all the active key types for the host. This will change all three key types (`RSA`, `RSA1`, and `DSA`) on the host and update those key types on the directory server. If you only want to change one key type or manage just one key type in the directory server, specify just that type (`rsa1`, `rsa`, or `dsa`) instead of `all`.

If the root user has already updated the keys for the remote host, you can use the same process as described above.

6.3.6 Changing key size

To change the key size used on a host, you must first use `ssh-keygen` to change the key, and then use `ldaphostmgr` to upload that key in the directory server. The following example shows how to change the bit size of the RSA key. In the example, we are logged in as root on the host `chef`:

```
# /opt/ssh/bin/ssh-keygen -b 4096 -t rsa -f /opt/ssh/etc/ssh_host_rsa_key
Generating public/private rsa key pair.
Please be patient....   Key generation may take a few minutes
/opt/ssh/etc/ssh_host_rsa_key already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /opt/ssh/etc/ssh_host_rsa_key.
```

```
Your public key has been saved in /opt/ssh/etc/ssh_host_rsa_key.pub.
The key fingerprint is:
ab:92:ec:71:8e:24:b9:5e:b9:1e:26:60:50:84:b9:bb root@chef
The key's randomart image is:
+--[ RSA 4096]----+
| +o              |
|o.               |
|..               |
|o                |
|.o      S        |
|o. . .   .       |
| .+.B.. .        |
|E  B+B .         |
| .oo=.o          |
+-----------------+
# ldaphostmgr -k /opt/ssh/etc/ssh_host_rsa_key.pub chef
bind-dn: uid=domadmin,ou=people,dc=mydomain,dc=example,dc=com
Password:
The public key(s) already exists in LDAP server, do you want
to replace it [y/n]? y
```

Notice that since root is required to run `ssh-keygen` and change the rsa key-pair for the host, you might not be prompted with your default LDAP login (as shown with `domadmin` in the other, previous examples), since root typically does not have an identity managed in the LDAP directory server.

6.3.7 Changing keys for non-HP-UX hosts

Since `ldaphostmgr` cannot directly modify the key files for non-HP-UX hosts (since it is not installed on those hosts), you must use a process similar to the one described in Section 6.3.3 (page 203), except that you must first delete the existing key before adding the new one. If you do not do this, the following error occurs:

```
baker (): ldaphostmgr -k ^rsa router1.mydomain.example.com
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
ERROR:    HST_UNTRUSTED_REMOTE_HOST:
          DANGER: The identity of the host router1.mydomain.example.com appears to be
          invalid. The key discovered for the remote host does not match that
          already managed in the directory server. This can occur if an
          attacker has set up a host to impersonate the true host. Or the key
          for the remote host may have been legitimately changed. Or both
          events may have occurred. ldaphostmgr will not directly replace this
          key in the directory server. Using ldaphostmgr on the remote host,
          instead of adding this key remotely, will avoid generating this
          warning message. Or use ldaphostmgr to first delete the key in the
          directory server for this host before attempting to replace it.
          However, do not replace this key in the directory server without
          using additional validation to verify the key for the remote host is
          valid. Once this key is replaced in the directory server, it will be
          trusted by all other LDAP-enabled ssh clients.
          Host fingerprint: 24:de:77:0e:c2:7a:af:0c:9d:15:ca:a8:8f:bb:65:d7
          LDAP fingerprint: 2e:fd:98:46:31:c7:fa:d9:a8:fd:61:02:bc:6b:2c:bb
```

You can delete and then add the key using the following process:

```
baker (): ldaphostmgr -k !rsa router1.mydomain.example.com
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
baker (): ldaphostmgr -k ^rsa router1.mydomain.example.com
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
WARNING: The identity of the host "router1.mydomain.example.com" could not be verified.
SSH key fingerprint: 24:de:77:0e:c2:7a:af:0c:9d:15:ca:a8:8f:bb:65:d7.
The SSH key for the remote host is unknown.  This host's key is currently not
managed in the directory server and should be positively identified before
adding this key to the directory server. Once added, this key will be
trusted by all other LDAP-enabled ssh clients.  Using ldaphostmgr on the
remote host, instead of adding this key remotely, will avoid generating
this warning message. Do you wish to trust this key (y/n)?: y
```

In this example, you must verify the fingerprint for the key before adding it to the directory server.

A alternative way to change a remote key is to securely obtain the public key file for the remote host and upload it using the `file` option as shown in the first example of Section 6.3.2 (page 201), but without specifying the `-a` option.

6.3.8 Revoking or removing keys

If a key has been compromised, and you want to revoke it and reissue a new key, use the previously described process for changing keys. If, on the other hand, you no longer want to manage keys for a host, you can simply remove the `sshPublicKey` attribute from the host's entry using the `-k` option with the `!` flag, as in the following example:

```
baker (): ldaphostmgr -k !all router1.mydomain.example.com
bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
Password:
baker (): ldaphostlist -n router1.mydomain.example.com sshPublicKey
dn: cn=router1.mydomain.example.com,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: router1.mydomain.example.com
ipHostNumber: 192.0.32.1
```

The `ldaphostlist` command shows that the sshPublicKey has been removed from the `router1` entry.

If you only wish to remove a specific type, you can replace `all` with the key type (rsa, rsa1, or dsa).

NOTE: If you are using the `UpdateKeyFromLdap` option in the `ssh_config` file, use of the `!` flag does not remove cached instances of those keys. If a client has a cached version of a compromised key, it is possible for that client to connect to an impostor host that is using the compromised host key. If you want to remove keys or revoke keys for hosts, you *must not* enable the `UpdateKeysFromLdap` option because when it is enabled, the ssh client tools will update cached versions of changed keys, but only when a connection is made to the true host.

6.4 Managing key age

LDAP-UX B.05.00 provides the ability to track ssh key age and set advisory expiration dates for ssh host keys. By default, `ldaphostmgr` adds key age information to the comment fields within the ssh public key data when new keys are added or changed in the directory server. `ldaphostmgr` can also use this same field to set advisory key expiration dates when new keys are created or existing keys are changed.

Key age expiration information appears within the comment fields and between the BEGIN-KM and END-KM tokens. For example:

```
brewer(): ldaphostlist -k -n "$(hostname)"
dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
cn: brewer
cn: brewer.mydomain.example.com
ipHostNumber: 16.92.96.225
sshPublicKey: ssh-rsa AAAA...== BEGIN-KM creationtime=20100423234903Z END-KM
```

`ldaphostmgr` and `ldaphostlist` can be used to keep track of key age and expiration information, which is described in the following sections.

NOTE: Key expiration data is merely advisory. It is provided to allow the `ldaphostlist` tool to display hosts with keys that are considered expired. HP Secure Shell tools do not reject or take other actions when a key's state is considered expired.

6.4.1 Setting advisory key expiration dates

To set key expiration information, use the `-e` option on `ldaphostmgr`, and specify the number of days (from the current date) when the key is considered expired. The following example shows how to set a key that should be considered expired in 2 years. If the key already exists in the directory server, you are prompted to replace it with a new key, if you so choose.

```
chef (): ldaphostmgr -k rsa -e 730 chef
bind-dn [uid=domadmin,ou=People,dc=cup,dc=hp,dc=com]:
Password:
The public key(s) already exists in LDAP server, do you want
to replace it [y/n]? y
```

To display the key expiration date, use `ldaphostlist` with the `-k` option:

```
chef (): ldaphostlist -k -n chef
dn: cn=chef,ou=Hosts,dc=cup,dc=hp,dc=com
cn: chef
cn: chef.cup.hp.com
ipHostNumber: 16.92.96.225
sshPublicKey: ssh-rsa AAAAB... BEGIN-KM ... expirationtime=20120426204647Z END-KM
```

6.4.2 Key Auditing

To display hosts with expired keys or keys that are older than a specified age, use the `-k` option of `ldaphostlist`. To display keys that are older than a specific age, use the `-k` option followed by the number of days preceded by a dash. For example, to show keys that were created over 1 year ago, use the following command:

```
baker (): ldaphostlist -k -365
dn: cn=chef,ou=Hosts,dc=cup,dc=hp,dc=com
cn: chef
cn: chef.cup.hp.com
ipHostNumber: 16.92.96.225
sshPublicKey: ssh-rsa AAAAB3... BEGIN-KM creationtime=20090426204647Z ... END-KM
```

If you are setting expiration information in keys, you can also use the `-k` option of `ldaphostlist` to display hosts with keys that have expired or will expire within a specified number of days. In this case, specify the `-k` age option without the preceding dash. For example, to display keys that have already expired or will expire within the next 20 days, use the following:

```
baker (): ldaphostlist -k 20
dn: cn=chef,ou=Hosts,dc=cup,dc=hp,dc=com
cn: chef
cn: chef.cup.hp.com
ipHostNumber: 16.92.96.225
sshPublicKey: ssh-rsa AAAAB3... BEGIN-KM ... expirationtime=20100515195500Z END-KM
```

NOTE: The above examples assume the commands were run on May 27th, Midnight UTC, 2010, which is represented by 20100427000000Z.

6.5 Centrally managing ssh configuration

In order to enable ssh key management on hosts, the `ssh_config` file, and optionally the `sshd_config` file, must be configured with the `UseLdapHostKey` parameter, and optionally the `UdateKeyFromLdap` parameter. To mitigate the management costs of changing these configuration files on all hosts, you can configure LDAP-UX to centrally manage the parameters of these files using the LDAP-UX central configuration service, provided by `ldapconfd`. Support for `ldapconfd` is limited to managing HP Secure Shell configuration, as documented in this section.

To do this, you must create a global configuration policy. Do this by first specifying the location of a global configuration policy in the LDAP-UX configuration profile. Then create a configuration policy entry using the `configurableService` objectclass and the `serviceConfigParam` attributes. The above schema for the Central Configuration service is defined in the `/etc/opt/ldapux/schema/ldapux5.0.xml` file delivered with LDAP-UX B.05.00. You can install that schema on your directory server using the `ldapschema` tool, described in Section 7.5.3 (page 301). That schema is automatically installed if you use the guided installation.

Use the `ldapentry` tool to modify the LDAP-UX Configuration profile. For example:

```
chef (): /opt/ldapux/bin/ldapentry -m "$profiledn"
Press <return> to accept default Directory login: "uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com"
Directory login:
Default accepted. "uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com"
password:
```

You are then placed in an editor window, where you can add a central configuration policy. Bolded text in the following example indicates what items were added:

```
version: 1
dn: cn=mydomain-ldapuxProfile,ou=Services,ou=Configuration,dc=mydomain,dc=example,dc=com
objectClass: top
objectClass: DUAConfigprofile
objectClass: configurableService
cn: mydomain-ldapuxProfile
preferredServerList: 127.0.0.1:389
defaultSearchBase: dc=mydomain,dc=example,dc=com
bindTimeLimit: 5
authenticationMethod: tls:simple
credentialLevel: proxy
attributeMap: passwd:userpassword=*NULL*
attributeMap: shadow:userpassword=*NULL*
attributeMap: group:memberUid=member uniqueMember memberUid
serviceSearchDescriptor: passwd:ou=People,
serviceSearchDescriptor: shadow:ou=People,
serviceSearchDescriptor: group:ou=Groups,
serviceSearchDescriptor: pam:ou=People,
serviceSearchDescriptor: rpc:ou=Services,
serviceSearchDescriptor: protocols:ou=Services,
serviceSearchDescriptor: networks:ou=Services,
serviceSearchDescriptor: hosts:ou=Hosts,
serviceSearchDescriptor: services:ou=Services,
serviceSearchDescriptor: printers:ou=Services,
serviceSearchDescriptor: automount:ou=Services,
serviceSearchDescriptor: netgroup:ou=Groups,
cfgGlobalPolicyDN: cn=mydomain-ldapuxProfile,dc=mydomain,dc=example,dc=com
serviceConfigParam: ssh/client/ssh_config:useldaphostkey yes
serviceConfigParam: ssh/client/ssh_config:updatekeyfromldap no
serviceConfigParam: ssh/server/sshd_config:useldaphostkey yes
```

In this example, the `cfgGlobalPolicyDN` attribute was added; it points to an entry that contains the `serviceConfigParam` attributes. In this case, the `cfgGlobalPolicyDN` points back to the profile entry itself, and the `serviceConfingParam` attributes were added directly to the same configuration profile entry.

The format of the `serviceConfigParam` value is in two parts. The first part is a hierarchical description of the service being configured. The second part is the specific parameter being managed. The format of the service description is:

`baseService/serviceSubsystem/...:`

And the format for the parameter section is specific to the configuration file being managed. For the `ssh_config file`, the following service description is used:

`ssh/client/ssh_config:`

For the `sshd_config` file, the following service description is used:

`ssh/server/sshd_config:`

In the previous example, `useldaphostkey` is being centrally managed, and will be added to any host that is part of the same LDAP-UX domain. The following shows an example of how the `ssh_config` file is changed:

.
.
.

```
# buffer size for hpn to non-hpn connections
#  HPNBufferSize 2048

#   Cipher 3des
#   Ciphers aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,arcfour,aes192-cbc,aes256-cbc
#   EscapeChar ~
#   Tunnel no
#   TunnelDevice any:any
#   PermitLocalCommand no

# Turn on/off Visual Fingerprinnt Display mode
#  VisualHostKey no

checkhostip yes

### CCD NOTE:
### The following keyword-argument pairs are configured in LDAP server.
### If you want to add local configurations to this file, add above the
### "CCD NOTE" line. Anything added manually below this line will be
### gone at next LDAP update.

# Keyword-argument pairs defined in LDAP server global entry:
updatekeyfromldap no
useldaphostkey yes
```

The central configuration service (`ldapconfd`) can be used to centrally manage other ssh and sshd parameters. For example, once ssh host keys are managed in a directory server, users connected to hosts managed with LDAP-UX will always have access to the public key for remote hosts. In that case, users should not be prompted about whether they would like to accept keys that have not been verified. So you could consider enabling the strict-host-key-checking feature of ssh (meaning users would not be prompted if an unknown key is discovered). As an example, the following could be added to the global configuration policy DN:

```
serviceConfigParam: ssh/client/ssh_config:strictHostKeyChecking yes
```

Values configured in the global policy will override those defined in the local configuration. For example, if the local `ssh_config` file defines "`strictHostKeyChecking ask`", but the central configuration is defined as above, then the "`strictHostKeyChecking ask`" is commented out by `ldapconfd`, and a "`strickHostKeyChecking yes`" is added to the CCD section of the `ssh_config` file.

6.5.1 Overriding central configuration

There are two ways to allow overriding the global configuration on a specific host:

- Disable `ldapconfd` on that specific host. To completely disable `ldapconfd`, modify the `/etc/opt/ldapux/ldapconfd.conf` file by setting the `enable_ldapconfd` parameter to zero:

  ```
  enable_ldapconfd 0
  ```

- Set a host-specific policy. For example, if the global policy for `strictHostKeyChecking` is set to `yes`, and you want to set it to `ask` for a specific host, you can add a `serviceConfigParam` to the host entry, using either the `ldapentry` or `ldaphostmgr` tool. For example, use the following command to enable the `ask` policy on the "`brewer`" system (assuming Central Configration policy has not been previously set for this host):

  ```
  baker (): ldaphostmgr -A objectclass=networkService \
   -A "serviceConfigParam=ssh/client/ssh_config:strictHostKeyChecking yes" brewer
  bind-dn [uid=domadmin,ou=People,dc=mydomain,dc=example,dc=com]:
  Password:
  baker (): ldaphostlist -n brewer serviceConfigParam
  dn: cn=brewer,ou=Hosts,dc=mydomain,dc=example,dc=com
  cn: brewer
  cn: brewer.mydomain.cup.hp.dom
  ipHostNumber: 192.0.32.11
  serviceConfigParam: ssh/client/ssh_config:strictHostKeyChecking yes
  ```

  With `ldapentry`, just specify the name of the host to edit, as follows:

  ```
  baker (): ldapentry -m hosts brewer
  ```

Then once in the editor established by `ldapentry`, simply add the `networkService` object class and the `serviceConfigParam` as shown in the preceding example.

6.6 Distributing Keys to Non-HP-UX hosts

The integrated ability to automatically use LDAP as an ssh key repository is available in HP Secure Shell A.05.50 or higher. If you plan on using LDAP central ssh key management in a heterogeneous environment, your ssh applications on other platforms might not be able to discover those keys in the directory server. While the `sshPublicKey` attribute is used by other ssh implementations, it is not available on all platforms. To allow a heterogeneous data center to participate in central ssh key management, you might need to distribute keys to non-HP-UX hosts. The following is a sample script that, with platform dependent modifications, can be used to periodically retrieve an update public key list to store in the host's `ssh_known_hosts` file. It could be run as a periodic "cron" job (see the *crontab*(1M) manpage).

A perl script is required to help parse the LDAP host entries. This perl script uses the `perl-ldap` perl module, which is common on most UNIX and Linux platforms:

```
#!/usr/bin/perl
use Net::LDAP::LDIF;
use Net::LDAP::Entry;
use strict;

my $infilename = shift || die "Input LDIF file name required";

my $ldif = Net::LDAP::LDIF->new( $infilename, "r", onerror => 'undef' );

while( not $ldif->eof() ) {
  my $entry = $ldif->read_entry ( );
  if ( $ldif->error() ) {
    print "Error msg: ", $ldif->error(), "\n";
    print "Error lines:\n", $ldif->error_lines(), "\n";
  } else {
    my @names = $entry->get_value("cn");
    my @keys  = $entry->get_value("sshPublicKey");
    foreach my $name (@names) {
      foreach my $key (@keys) {
        print "$name $key\n"
      }
    }
  }
}
$ldif->done();
```

The input to this script is an LDIF file, which must be obtained through the `ldapsearch` command, also available on most platforms. Note that the connection to the directory server should be made with SSL, to make sure the client has some assurance that it is not communicating with an impostor directory server. The following example is for the `ldapsearch` command available with LDAP-UX. Your `ldapsearch` command might require slightly different parameters:

```
ldapsearch -Z -P CACertPath -b "ou=hosts,dc=mydomain,dc=example,dc=com" \
   -h hostname "(&(objectclass=iphost)(sshpublickey=*))" \
   cn sshpublickey > allhostkeys.ldif
```

To create the `known_hosts` file, send the output of `ldapsearch` into the above script. If you named the above perl script `makeKnownHosts.pl`, you would then use:

```
makeKnownHosts.pl allhostkeys.ldif > ssh_known_hosts
```

7 Command and tool reference

This chapter describes the commands and tools associated with the LDAP-UX Client Services.

7.1 The LDAP-UX Client Services components

The LDAP-UX Client Services product, comprising the components listed in Table 7-1, can be found under /opt/ldapux and /etc/opt/ldapux, except where noted. LDAP-UX Client Services libraries are listed in Table 7-2 (page 213) and Table 7-3 (page 213).

Table 7-1 LDAP-UX Client Services components

Component	Description
/etc/opt/ldapux/ldapux_client.conf	The LDAP-UX start-up file, specifies where the directory is, where in the directory the profile data is, and logging.
/etc/pam.ldap	A sample PAM configuration file. The actual PAM configuration file is /etc/pam.conf.
/etc/nsswitch.ldap	A sample Name Service Switch configuration file. The actual NSS configuration file is /etc/nsswitch.conf.
/etc/opt/ldapux/ldapux_profile.bin	The configuration profile translated from ldapux_profile.ldif, in binary format, used by the client. See also `display_profile_cache` below.
`/etc/opt/ldapux/domain_profiles/ldapux_profile.bin.gc`	Global Catalog Server (GCS) profile file (LDAP-UX with Windows AD Server). Specifies which server (and port) serves as the GCS.
/etc/opt/ldapux/ldapux_profile.ldif	The configuration profile downloaded from the LDAP directory, in LDIF format.
`/etc/opt/ldapux/domain_profiles`	Remote domains configuration profile file (LDAP-UX with Windows AD Server).
/opt/ldapux/config/autosetup	Program to configure LDAP-UX Client services. Will also automatically configure an HP-UX directory server instance and create an LDAP-UX domain if one has not already been set up.
/opt/ldapux/config/setup	Program to configure LDAP-UX Client Services.
/opt/ldapux/config/get_profile_entry	Program to download a configuration profile from a directory.
/opt/ldapux/config/display_profile_cache	Program to display the current configuration profile.
/opt/ldapux/config/create_profile_entry	Program to create a new configuration profile.
/opt/ldapux/config/create_profile_schema /opt/ldapux/config/create_profile_cache	Programs called by the setup program.
/opt/ldapux/config/ldap_proxy_config	Program to configure and verify the proxy user.
/opt/ldapux/bin/ldapcfinfo	Tool to report LDAP-UX configuration and status. This tool can be used to discover LDAP-UX configuration details about required attributes when creating new users or groups in an LDAP directory server.

Table 7-1 LDAP-UX Client Services components *(continued)*

Component	Description
/opt/ldapux/bin/ldapuglist /opt/ldapux/bin/ldapugadd /opt/ldapux/bin/ldapugmod /opt/ldapux/bin/ldapugdel /opt/ldapux/bin/ldaphostmgr /opt/ldapux/bin/ldaphostlist	Tools to display, add, modify and delete user and group entries in an LDAP directory server. See Section 7.3 (page 219) for details.
/etc/opt/ldapux/ug_templates/ug_passwd_std.tmpl /etc/opt/ldapux/ug_templates/ug_group_std.tmpl	The default template files are used by `ldapugadd` to discover the required data models for a new user or group entry for a standard LDAP directory server.
/etc/opt/ldapux/ug_templates/ug_passwd_ads.tmpl /etc/opt/ldapux/ug_templates/ug_group_ads.tmpl	The default template files are used by `ldapugadd` to discover the required data models for a new user or group entry for a Windows Active Directory Server.
/etc/opt/ldapux/ldapug.conf	The `ldapugadd` tool uses this configuration file to manage the default values of the `uidNumber_range`, `gidNubmer_range`, default_gidNumber, `default_homeDirectory` and `default_loginShell` attributes when creating a user or a group to an LDAP directory server.
/opt/ldapux/bin/ldapdelete /opt/ldapux/bin/ldapmodify /opt/ldapux/bin/ldapsearch /opt/ldapux/bin/ldapentry /opt/ldapux/bin/ldap_del_entry /opt/ldapux/bin/ldap_new_entry /opt/ldapux/bin/ldap_mod_entry	Tools to delete, modify, and search for entries in a directory. For more information, see Section 7.4 (page 292) and the *HP-UX Directory Server administrator guide*.
/opt/ldapux/bin/ldifdiff	Tool to generate LDIF change records from two input files.
/etc/opt/ldapux/ldapclientd.conf	The ldapclientd daemon configuration file.
/opt/ldapux/bin/ldapclientd	The ldapclientd daemon binary.
/opt/ldapux/bin/ldappasswd	Tool to modify user password in a directory.
/opt/ldapux/bin/ldapschema	Tool to query and extend directory server schema. See the “Schema Extension Utility” section for details.
/opt/ldapux/migrate /opt/ldapux/migrate/ads	A set of scripts for migrating user, group, and other information into a directory. See Section 7.6 (page 326) for more information.
/opt/ldapux/share	Manpages.
/opt/ldapux/contrib/bin/perl	perl, version 5, used by migration scripts.
/opt/ldapux/ypldapd	Files for the NIS/LDAP Gateway product. See *Installing and Administering NIS/LDAP Gateway*.
/opt/ldapux/contrib/bin/beq	Search tool that bypasses the name service switch and queries the backend directly based on the specified library.
/opt/ldapux/contrib/bin/certutil	Command-line tool that creates and modifies the `cert8.db` and `key3.db` database files.

NOTE: For LDAP C SDK libraries information, see "Mozilla LDAP C SDK" (page 337) for details.

Table 7-2 shows LDAP-UX Client Services libraries on HP-UX 11i v2 and v3 PA-RISC machines:

Table 7-2 LDAP-UX Client Services libraries on the HP-UX 11i v2 or v3 PA-RISC machine

Files	Description
/usr/lib/libldap_send.1 (32-bit) /usr/lib/libldap_util.1 (32-bit) /usr/lib/libnss_ldap.1 (32-bit) /usr/lib/libldapci.1 (32-bit) /usr/lib/libldap.1 (32-bit) /usr/lib/security/libpam_ldap.1(32-bit) /usr/lib/security/libpam_authz.1 (32-bit) /usr/lib/pa20_64/libldap.1 (64-bit) /usr/lib/pa20_64/libldap_send.1 (64-bit) /usr/lib/pa20_64/libnss_ldap.1 (64-bit) /usr/lib/security/pa20_64/libpam_ldap.1 (64-bit) /usr/lib/security/pa20_64/libpam_authz.1 (64-bit)	LDAP -UX Client Services libraries.

Table 7-3 shows LDAP-UX Client Services libraries on an HP-UX 11i v2 or v3 Integrity server machine:

Table 7-3 LDAP-UX Client Services libraries on an HP-UX 11i v2 or v3 Integrity server machine

Files	Description
/usr/lib/hpux32/libldap_send.so.1 (32-bit) /usr/lib/hpux32/libldap_util.so.1 (32-bit) /usr/lib/hpux32/libnss_ldap.so.1 (32--bit) /usr/lib/hpux32/libldapci.so.1 (32-bit) /usr/lib/hpux32/libldap.so.1 (32-bit) /usr/lib/security/hpux32/libpam_ldap.so.1 (32-bit) /usr/lib/security/hpux32/libpam_authz.so.1 (32-bit) /usr/lib/hpux64/libldap.so.1 (64-bit) /usr/lib/hpux64/libldap_send.so.1 (64-bit) /usr/lib/hpux64/libnss_ldap.so.1 (64-bit) /usr/lib/security/hpux64/libpam_ldap.so.1 (64-bit) /usr/lib/security/hpux64/libpam_authz.so.1 (64-bit)	LDAP -UX Client Services libraries.

7.2 Client management tools

This section describes the following programs for managing client systems.

display_profile_cache	Displays the currently active profile.
create_profile_entry	Creates a new profile in the directory.
get_profile_entry	Downloads a profile from the directory to LDIF, and creates the profile cache.
ldap_proxy_config	Configures a proxy user.
ldapcfinfo	Displays LDAP-UX configuration and status by examining LDAP UG template files, LDAP UG configuration file or the LDAP-UX configuration profile. See Section 7.3.10 (page 286) or the *ldapcfinfo* manpage for detailed information about tool usage, syntax, options and arguments.

The following tools are called by the setup program and are not typically used separately.

create_profile_schema	Extends the schema in the directory for profiles.
create_profile_cache	Creates a new active profile from an LDIF profile. This is also called by get_profile_entry.

7.2.1 create_profile_entry tool

This tool, found in /opt/ldapux/config, creates a new profile entry in an LDAP directory from information you provide interactively. The directory schema must have the DUAConfigProfile extensions.

7.2.1.1 Syntax

`create_profile_entry`

7.2.2 create_profile_cache tool

This tool, found in /opt/ldapux/config, creates a binary profile file from an LDIF profile file, thus activating the profile for the client. (You can download a profile to LDIF from the directory with get_profile_entry.) Typically you run the `setup` program instead of running this program directly. See also Section 2.5.8 (page 113).

7.2.2.1 Syntax

`create_profile_cache [-i infile] [-o outfile]`

where ***`infile`*** is the LDIF file containing a profile, by default /etc/opt/ldapux/ldapux_profile.ldif and ***`outfile`*** is the name of the binary output file, by default /etc/opt/ldapux/ldapux_profile.bin. The LDIF file must contain an entry for the object class DUAConfigProfile.

7.2.2.2 Examples

The following command creates the binary profile file /etc/opt/ldapux/ldapux_profile.bin from the existing LDIF file /etc/opt/ldapux/ldapux_profile.ldif:

`create_profile_cache`

The following command creates the binary profile file my_profile.bin from the existing LDIF file profile1.ldif:

`create_profile_cache -i profile1.ldif -o my_profile.bin`

Note that you must copy the file **`my_profile.bin`** to **`/etc/opt/ldapux/ldapux_profile.bin`** to activate the profile.

7.2.3 create_profile_schema tool

This tool, found in /opt/ldapux/config, extends the schema of an HP-UX Directory Server with the DUAConfigProfile object class using the information you provide interactively. Typically you run the `setup` program instead of running this program directly.

7.2.3.1 Syntax

```
create_profile_schema
```

7.2.4 display_profile_cache tool

This tool, found in /opt/ldapux/config, displays information from a binary profile (cache) file. By default, it displays the currently active profile in /etc/opt/ldapux/ldapux_profile.bin.

7.2.4.1 Syntax

```
display_profile_cache [-i infile] [-o outfile]
```

where ***infile*** is a binary profile file, /etc/opt/ldapux/ldapux_profile.bin by default, and ***outfile*** is the output file, stdout by default.

NOTE: The binary profile contains mappings for all backend commands (even unused ones) all of which are displayed by `display_profile_cache`. The actual client configuration can be reviewed in the configuration profile LDIF file: `/etc/opt/ldapux/ldapux_profile.ldif`.

7.2.4.2 Examples

The following command displays the profile in the binary profile file /etc/opt/ldapux/ldapux_profile.bin to stdout:

```
display_profile_cache
```

The following command displays the profile in the binary profile file my_profile.bin and writes the output to the file profile:

```
display_profile_cache -i my_profile.bin -o profile
```

7.2.5 get_profile_entry tool

This tool, found in /opt/ldapux/config, downloads a profile from an LDAP directory into an LDIF file and calls create_profile_cache to create a binary profile file, thereby activating it on the client. This tool looks in the local client configuration file /etc/opt/ldapux/ldapux_client.conf for the profile DN.

7.2.5.1 Syntax

```
get_profile_entry -s service [-o outfile]
```

where ***service*** is the name of a supported service, typically NSS, and ***outfile*** is the name of a file to contain the LDIF output, by default /etc/opt/ldapux_profile.ldif.

7.2.5.2 Examples

The following command downloads the profile for the Name Service Switch (NSS) specified in the client configuration file /etc/opt/ldapux/ldapux_client.conf and places the LDIF in the file /etc/opt/ldapux/ldapux_profile.ldif:

```
get_profile_entry -s NSS
```

The following command downloads the profile for the Name Service Switch (NSS) specified in the client configuration file /etc/opt/ldapux/ldapux_client.conf and places the LDIF in the file profile1.ldif:

```
get_profile_entry -s NSS -o profile1.ldif
```

7.2.6 ldap_proxy_config tool

This tool, found in `/opt/ldapux/config`, configures a proxy user or an Admin Proxy user for the client accessing the directory. It stores the proxy user information in the user proxy credential file`/etc/opt/ldapux/pcred`. The Admin Proxy user information is stored in the administrator proxy credential file `/etc/opt/ldapux/acred`. If you are using only anonymous access, you do not need to use this tool. You must run this tool logged in as root. While the data stored in the `pcred` and `acred` files are protected for root-only access and not stored in plain text, the data is not encrypted.

The `/etc/opt/ldapux/pcred` file is used to contain credentials that represent all users of the HP-UX OS to the directory server. For example, when a user wishes to run the `ls -l` command to see who owns a file or directory, the OS must contact the directory server to translate the owner ID number into a name. If the directory server does not allow anonymous access, a proxy user must be created to be used to authenticate to the directory server and represent any user requesting such information.

The `/etc/opt/ldapux/acred` file is used to represent any administrative user (typically root), which should have additional permissions in the directory server beyond that of the non-privileged user. The `acred` file will store the credentials of a user with permissions to modify specific attributes (as needed) based on commands that are performed on the OS. Specifically, the `acred` credential allows a root user to change any user's `nisPublickey` and `nisPrivate` key attributes. Because the `chkey` and `newkey` commands do not prompt for directory user credentials, the `acred` file is required to allow the administrator to reset such attributes. The `acred` file is also used by the `ldapugadd`, `ldapugmod`, `ldapugdel` and `ldaphostmgr` commands. However, those utilities have the ability to prompt for credentials or to obtain them with other methods. So the `acred` file is not required. Because a privileged credential is stored in the `acred` file, creation of the `acred` file is recommended only for managing NIS keys in the directory server, and only if key reset is required. In addition, access to the `acred` file must be restricted.

7.2.6.1 Syntax

`ldap_proxy_config [`*`options`*`]`

where ***`options`*** can be any of the following:

`-A` Action applies to the Admin Proxy user. This option must be specified with other option to apply the operation for the Admin Proxy user.

`-e` erases the currently configured proxy user from the file `/etc/opt/ldapux/pcred`. Has no effect on the proxy user information in the directory itself.

`-i` uses the `-i` option to configure the proxy user interactively from stdin. Use `-A -i`options to configure an Admin Proxy user.

If you use `ldap_proxy_config -i` to configure the proxy user using the simple authentication, type the command with -i then press Return. Next type the proxy user DN then press Return. Finally type the proxy user's credential or password and press Return.

If you configure the proxy user using the SASL DIGEST-MD5 with DN authentication (i.e. use the DN to generate the DIGEST-MD5 hash), type the command with -i then press Return. Next type the proxy user DN then press Return. Next type the proxy user's credential or password and press Return. Finally press Return.

If you configure the proxy user using the SASL/DIGEST-MD5 with UID authentication (i.e. use the UID attribute to generate the DIGEST-MD5 hash), type the command with -i then press Return. Next type the proxy user DN then press Return. Next type the proxy user's credential or password and press Return. Finally type the proxy user's UID and press Return.

When you use the `ldap_proxy_config -A -i` command to configure an Admin Proxy user interactively from stdin, the configuration procedures are similar to the procedures used by the `ldap_proxy_config -i` command for a proxy user.

When configuring an Admin Proxy user, if you only enter the Admin Proxy user's DN without password, the root's password will be used instead.

`-f file` configures the proxy user from the specified file (***`file`***). The ***`file`*** specification must contain two lines: the first line must be the proxy user DN, and the second line must be the proxy user credential or password.

CAUTION: After using this option you should delete or protect the file as it could be a security risk.

`-d DN` sets the proxy user distinguished name to be ***`DN`***. To use this option, the */etc/opt/ldapux/pcred* file must exist.

`-c passwd` sets the proxy user credential or password to be ***`passwd`***. To use this option, the `/etc/opt/ldapux/pcred` file must exist.

`-p` prints the distinguished name of the current proxy user.

`-v` verifies the current proxy user and credential by connecting to the server.

`-h` displays help on this command.

With no options, `ldap_proxy_config` configures the proxy user as specified in the file `/etc/opt/ldapux/pcred`.

For the proxy user, if you switch the authentication method between simple and DIGEST-MD5, you need to use the `ldap_proxy_config -e` command to delete `/etc/opt/ldapux/pcred`, then use the `ldap_proxy_config -i` command to reconfig the proxy user.

For the Admin Proxy user, if you switch the authentication method between simple and DIGEST-MD5, you need to use the `ldap_proxy_config -A -e` command to delete `/etc/opt/ldapux/acred`, then use the `ldap_proxy_config -A -i` to reconfigure the Admin Proxy user.

7.2.6.2 Examples

The following example configures the proxy user as **`uid=proxyuser1,ou=special users,o=hp.com`** with the password **`prox1pw`** and creates or updates the file `/etc/opt/ldapux/pcred` with this information, the proxy user uses the simple authentication:

```
ldap_proxy_config -i
uid=proxyuser1,ou=special users,o=hp.com
prox1pw
```

The following example configures the proxy user as `uid=proxyusr2,ou=special users,o=hp.com` with password **`prox2pw`**, and creates or updates the file `/etc/opt/ldapux/pcred` with this information. The proxy user uses the SASL DIGEST-MD5 authentication and uses the DN to generate the DIGEST-MD5 hash.

```
ldap_proxy_config -i
uid=proxyusr2,ou=special users,o=hp.com
prox2pw
CR>
```

The following example configures the proxy user as `uid=proxyusr3,ou=special users,o=hp.com`, UID `proxyusr3` and password `prox3pw`, and creates or updates the file `/etc/opt/ldapux/pcred` with this information. The proxy user uses the SASL/DIGEST-MD5 authentication and uses the UID to generate the DIGEST-MD5 hash.

```
ldap_proxy_config -i
uid=proxyusr3,ou=special users,o=hp.com
```

```
prox3pw
proxyusr3
```

The following example configures the Admin Proxy user as `uid=adminproxy,ou=special users,o=hp.com` with the password `adminproxpw`, and creates or updates the file `/etc/opt/ldapux/acred` with this information. The Admin Proxy user uses the simple authentication.

```
ldap_proxy_config -A -i
uid=adminproxy,ou=special users,o=hp.com
adminproxpw
```

The following example configures the Admin Proxy user as `uid=adminproxy2,ou=special users,o=hp.com` with password **admin2pw**, and creates or updates the file `/etc/opt/ldapux/acred` with this information. The Admin Proxy user uses the SASL/DIGEST-MD5 authentication and uses the DN to generate the DIGEST-MD5 hash.

```
ldap_proxy_config -A -i
uid=adminproxy2,ou=special users,o=hp.com
admin2pw
CR>
```

The following example configures the Admin Proxy as `uid=adminproxy3,ou=special users,o=hp.com`, UID `adminproxy3`, and password `admin3pw`, and creates or updates the file `/etc/opt/ldapux/acred` with this information. The Admin Proxy user uses the SASL/DIGEST-MD5 authentication and uses the UID to generate the DIGEST-MD5 hash.

```
ldap_proxy_config -A -i
uid=adminproxy3,ou=special users,o=hp.com
admin3pw
adminproxy3
```

The following example displays the current proxy user:

```
ldap_proxy_config -p
PROXY_DN: uid=proxyuser,ou=special users,o=hp.com
```

The following example checks the configured proxy user information and checks whether or not the client can bind to the directory as the proxy user:

```
ldap_proxy_config -v
File Credentials verified - valid
```

The following example configures the proxy user as `uid=proxyuser,ou=special users,o=hp.com` with the password `prox12pw`, and creates or updates the file `/etc/opt/ldapux/pcred` with this information:

```
ldap_proxy_config -d "uid=proxyuser,ou=special users,o=hp.com" -c prox12pw
```

The following example configures the proxy user with the contents of the file `proxyfile` and creates or updates the file `/etc/opt/ldapux/pcred` with this information:

```
ldap_proxy_config -f proxyfile
```

The file `proxyfile` must contain two lines: the proxy user DN on the first line and password on the second line.

7.3 LDAP user and group management tools

The LDAP-UX Integration product supports the following new LDAP command-line tools which enable you to manage user accounts and groups in an LDAP directory server. These new tools exist in the `/opt/ldapux/bin` directory and perform their operations based on the LDAP-UX profile's configuration. Each tool provides command options that enable you to alter these configuration parameters. For detailed information about tool usage, syntax, options, arguments, environment variables, template files, return codes supported by these tools, see Section 7.3 (page 219), or see the *ldapuglist*(1M), *ldapugadd*(1M), *ldapcfinfo*(1M), *ldapugmod*(1M), and *ldapugdel*(1M) manpages.

`ldapuglist` Use the `ldapuglist` tool to display and enumerate subsets of POSIX-like account and group entries that reside in an LDAP directory server.

`ldapugadd` Use the `ldapugadd` tool to add new POSIX accounts or groups to an LDAP directory server.

`ldapugmod` Use the `ldapugmod` tool to modify existing accounts or groups in an LDAP directory server. You can use `ldapugmod` with extended options to modify arbitrary attributes for user or group entries.

`ldapugdel` Use the `ldapugdel` tool to remove POSIX related user or group entries from an LDAP directory server. Use the `-O` option to remove POSIX related attributes and object classes from a user or a group entry without removing entire entry itself.

`ldapcfinfo` Use the `ldapcfinfo` tool to retrieve LDAP-UX configuration information details about required attributes when creating new users or groups. Use this tool to discover LDAP UG configuration defaults, a list of available template files and attribute mapping information. You can also use `ldapcfinfo` to check whether the LDAP-UX product is properly configured and active.

When performing modification, creation and deletion operations on the LDAP directory server, use these tools to input the LDAP administrator bind identity and credential interactively with a prompt (-P) option or by specifying the `LDAP_BINDDN` environment variable for the administrator identity and `LDAP_BINDCRED` environment variable for the administrator's credential. Values set with a prompt (-P) option override values specified in the environment variables. If the two previously mentioned methods have not been specified, the LDAP tool follows the bind configuration specified in the LDAP-UX configuration profile. If the LDAP-UX profile has specified a proxy bind, the LDAP tool reads the credential from either the `/etc/opt/ldapux/acred` or `/etc/opt/ldapux/pcred` file. The `/etc/opt/ldapux/acred` file is used only by users who have sufficient administrative privilege to read this file.

7.3.1 Environment variables

The `ldapuglist`, `ldapugadd`, `ldapugmod` and `ldapugdel` tools support the following environment variables:

`LDAP_BINDDN` Specifies the distinguished name (DN) or other appropriate identity indicator (such as a Kerberos principle id) of a user with sufficient directory server privilege to view, add, modify or delete users and/or groups in the LDAP directory server. If `LDAP_BINDDN` is specified, `LDAP_BINDCRED` must also be specified.

`LDAP_BINDCRED` Specifies a password or other type of credential used for the LDAP user specified by `LDAP_BINDDN`.

The `ldapugadd` and `ldapugmod` tools support the following environment variable:

`LDAP_UGCRED` This variable specifies the new password of a user or group being created or modified. You must use the `-PW` command option when you use this environment variable, to indicate this variable has been set and is used for the

current command. If attribute mapping for the `userPassword` attribute has not been defined or set to "*NULL*" in the LDAP-UX configuration profile, `ldapugadd` or `ldapugmod` creates new passwords using the `userPassword` attribute. See the `-PW` option of Section 7.3.5 (page 232) or Section 7.3.6 (page 250) for additional information.

NOTE: To support non-interactive use of the `ldapuglist`, `ldapugadd`, `ldapugmod` and `ldapugdel` commands, you can use the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables to specify the LDAP administrator's identity and password. Use `LDAP_UGCRED` to specify the user or group password being created or modified. To prevent exposure of these environment variables, you must unset them after use. The `shells(4)` command history log may contain copies of the executed commands that show setting of these variables. You must protect access to a shell's history file. Specification of the LDAP administrator's credentials on the command line is not allowed, because information about the currently running processes can be exposed externally from the session. Using the `-P` command option eliminates the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables by interactively prompting for the required administrator's credentials. Using the `-PP` command option eliminates `LDAP_UGCRED` by interactively prompting for the required password of the user or group being created or modified.

7.3.2 Return value formats

Upon exit, `ldapuglist`, `ldapugadd`, `ldapugmod`, `ldapugdel` or `ldapcfinfo` returns a 0 (zero) exit status if no errors or warnings are encountered. A non-zero exit status is returned and one or more messages are logged to stderr if these tools encounters an error or warnings. Messages follow the below format:

```
ERROR:          <code>:
                <message>
or
WARNING:        <code>:
                <message>
```

Leading extra white space may be inserted to improve readability and follow 80 column screen formatting. `<code>` is a programmatically parsable error key-string, while `<message>` is human-readable text.

7.3.3 Common return codes

Table 7-4 lists common return codes used by `ldapuglist`, `ldapugadd`, `ldapugmod`, `ldapugdel` and `ldapcfinfo`.

For detailed information on a list of specific return codes for each tool, see "Specific return codes for ldapuglist" (page 229), "Specific return codes for ldapugadd" (page 246), "Specific return codes for ldapugmod" (page 258), "Specific return codes for ldapugdel" (page 264), or "Specific return codes for ldapcfinfo" (page 288).

Table 7-4 Common return codes

Return Code	Message
`LDAP_INIT_FAILED`	Unable to initialize LDAP-UX library backend.
`GET_LDAP_CONFIG_FAILED`	Cannot read the `ldapux_profile.bin` file.
`REPLACE_PORT_FAILED`	Cannot reset the port number.
`INVALID_AUTH_MATHOD`	The specified authentication method is invalid.
`READ_INPUT_FAILED`	Unable to read input from stdin for the specified command option value.

Table 7-4 Common return codes *(continued)*

GETENV_FAILED	The LDAP_BINDDN environment variable is set, but LDAP_BINDCRED is not set.
BIND_PASSWORD_EXPIRED	The bind Password has expired.
BIND_INVALID_CRED	The specified bind credential is invalid.
BIND_ERR	LDAP-UX failed to bind to the LDAP directory server.
GET_PROXY_DECRYPT_FAILED	Failed to decrypt proxy and credential information.
MOD_LIMIT_REACHED	There are too many modifications to perform.
SSL_INIT_FAILED	SSL initialization failed.
LOAD_LIB_FAILED	Failed to load the specific library.
LOAD_FUNCTION_FAILED	Failed to load the specific function.
ACCESS_TEMPLATEFILE_FAILED	Unable to access specified template file.
READ_TEMPLATEFILE_FAILED	Unable to read specified template file.
MISSING_DIRECTIVE	The specified template file is missing the required directive.
INVALID_TEMPLATENAME	The template file name specified in the -T <file_name> option is invalid.
NEED_CONFIG_PROXY	Needs to configure LDAP-UX to use proxy credential level.
NEED_ADMIN_CRED	Requires administrator credentials in order to perform privilege user management operations. Specification of the -P option or the LDAP_BINDDN and LDAP_BINDCRED environment variables are required.
NO_PROXY_FILE	LDAP-UX proxy has been configured, but the /etc/opt/ldapux/pcred file does not exist.
BUFFER_OVERFLOW	Buffer overflowed when processing a specific operation.
CHOWN_FAILED	Failed to change the ownership of the files.
CHOWN_FAILED	Cannot modify the specified home directory.
HOMEDIR_CREATE_FAILED	Failed to create the specified home directory.
INVALID_DIR	The specified directory is not a valid directory.
HOMEDIR_EXISTS	The specified home directory already exists.
ACCOUNT_DOESNOT_EXIST	The specified account does not exist in the directory server.
COMMANDLINE_ERR	The valid type of the -t <type> option should be either passwd or group.
COMMANDLINE_ERR	Need to specify a value for the specified option.
COMMANDLINE_ERR	User name has not been specified.
COMMANDLINE_ERR	Group name has not been specified.
COMMANDLINE_ERR	Invalid <attr>=<value> pair specified in the command line.
PASSWD_INCONSISTENT	Entered inconsistent password.
CANNOT_GET_PASSWD	Cannot read password.
ZERO_LENGTH_PASSWD	Password entered is 0 length.
ENV_VAR_NOT_SET	The specific environment variable is not set.
INVALID_SEARCH_SCOPE	The input search scope must be base, one, or sub.

Table 7-4 Common return codes *(continued)*

GROUP_DOESNOT_EXIST	The specified group does not exist in the LDAP directory server.
LOGIN_SHELL_DOESNOT_EXIST	The specified login shell does not exist.
HOMEDIR_DOESNOT_EXIST	The specified home directory does not exist.
LOGIN_SHELL_NOT_EXECUTE	The specified login shell is not executable.
ADD_GR_MEMBER_FAILED	`MemberUid` is mapped to only dynamic group attributes, the add operation fails.
ENTRY_NOT_FOUND	The LDAP search returns no entries.
EXPLODE_DN_FAILED	Cannot convert the specified distinguished name (DN) to its component parts.
EXPLODE_RDN_FAILED	Cannot convert the specified RDN to its component parts.
MODIFY_FAILED	The modification operation failed.

7.3.4 ldapuglist tool

You can use the `ldapuglist` tool to display and enumerate POSIX-like account and group entries stored in an LDAP directory server, without requiring extensive knowledge of the methods used to retrieve and evaluate that information in the LDAP directory server.

The `ldapuglist` tool uses the LDAP-UX profile configuration, requiring minimal command line options to discover where to search for user or group information, such as the LDAP directory server host and proper search filters for finding users and groups. This tool provides command options that enable you to alter these configuration parameters.

The `ldapuglist` tool supports the followings:

- `ldapuglist` uses the existing LDAP-UX authentication configuration to determine how to bind to the LDAP directory server.
- `ldapuglist` performs attribute value translation to POSIX-like syntaxes. For example, if group membership is defined using X.500-style DN strings, `ldapuglist` converts those string to simple member ids.
- `ldapuglist` supports attribute mappings as specified in the LDAP-UX configuration profile. The mapped attributes and values can be displayed. The output format of `ldapuglist` is similar to an LDIF format (RFC 2849). It is not LDIF. Major differences include:
  - `ldapuglist` does not display object classes.
  - By default, `ldapuglist` only displays POSIX-related attributes, unless you specifically request an attribute list with the `<attr>` option on the command line.
  - Output lines are not broken after 80 columns.

7.3.4.1 Synopsis

`ldapuglist` ***`[options] [-t <type>] [-h <hostname>] [-p <port>] [-n <name>] [-f|F <filter>] [-b <base>] [-s <scope> [-N <maxcount>] [<attr>...]`***

7.3.4.2 Options

The `ldapuglist` tool supports the following command options:

`-m` Displays the names of the mapped attributes when returning results. Without the `-m` option, `ldapuglist` displays results as follows:

`fieldname: value`

Where *`fieldname`* is one of the predefined RFC 2307 attribute names, and *`value`* is the value for that field.

With the `-m` option, the `ldapuglist` tool displays the actual attribute mapping name as follows:

`fieldname[mapped attributename]: value`

In the following example, if the RFC 2307 attribute `gecos` has been mapped to the `cn`, `l` (location) and `telephoneNumber` attributes. Without the `-m` option, the output of the `gecos` field is:

```
gecos: Bill Wan,Building 45,1-555-555-5431
```

When the `-m` option is specified, the output representing the `gecos` field is as follows:

```
gecos[cn]: Bill Wang
gecos[l]: Building 45
gecos[telephoneNumber]: 1-555-555-5431
```

When a field has been mapped to multiple attributes, those attributes will appear in the order as defined in the LDAP-UX configuration profile.

Another example, if the RFC 2307 attribute `uidNumber` has been mapped to the `employeeNumber` attribute. Without the `-m` option, the output of the `uidNumber` field is:

```
uidNumber: 520
```

When the `-m` option is specified, the output representing the `uidNumber` field is as follows:

```
uidNumber[employeeNumber]: 520
```

The `ldapuglist` tool ignores the `-m` option if the `-L` option is specified.

`-L` Displays output following `/etc/passwd` or `/etc/group` format.

The output format for a user entry is as follows:

```
uid:userPassword:uidNumber:gidNumber:gecos:homeDirectory:loginShell
```

The output format for a group entry is as follows:

```
cn:userPassword:memberUid,memberUid,...
```

For example, run the following command to display the user entry that contains `uid=mscott`:

```
ldapuglist -t passwd -L -n mscott
```

The output of the command is as follows:

```
mscott:x:200:250:mscott:/home/mscott:/usr/bin/sh
```

The `ldapuglist` tool ignores the `-m` option if the `-L` option is specified. The `<attr>` parameter list is invalid if the `-L` option is specified.

`-P` Prompts for the bind identity (typically LDAP DN or Kerberos principal) and bind password. Without the `-P` option, `ldapuglist` attempts to get the bind identity and password from the environment variables `LDAP_BINDDN` and `LDAP_BINDCRED`. If you do not specify the `LDAP_BINDDN` or `LDAP_BINDCRED` environment variables, `ldapuglist` gets information from the bind configuration specified in the LDAP-UX configuration profile. If the LDAP-UX configuration profile has specified the "proxy" bind, `ldapuglist` reads the bind credential from either the `/etc/opt/ldapux/acred` or `/etc/opt/ldapux/pcred` file. The `/etc/opt/ldapux/acred` file is only used by users who have sufficient administrative privilege to read that file.

`-Z` Requires an SSL connection to the LDAP directory server, even if the LDAP-UX configuration profile does not specify the use of SSL. Using the `-Z` option requires that either a valid directory server or CA certificate is defined in the `/etc/opt/ldapux/cert8.db` file. An error occurs if the SSL connection cannot be established.

`-ZZ` Attempts a TLS connection to the directory server, even if the LDAP-UX configuration profile does not specify the use of TLS. If a TLS connection cannot be established, a non-TLS and non-SSL connection will be established. HP does not recommend you to use `-ZZ` unless alternative methods are used to protect against network eavesdropping. Use of `-ZZ` requires that you define a valid LDAP directory server or CA certificate in the `/etc/opt/ldapux/cert8.db` file.

`-ZZZ` Requires a TLS connection to the LDAP directory server, even if the LDAP-UX configuration profile does not specify the use of TLS. Using the `-ZZZ` option requires that you define a valid directory server or CA certificate in the `/etc/opt/ldapux/cert8.db` file. An error will occur if the TLS connection can not be established.

7.3.4.3 Arguments

The following describes command arguments:

`-t <type>` Specifies the type of entry the `ldapuglist` tool needs to discover and process. The valid types of this option are `passwd` and `group`. The `passwd`

type indicates posixAccount-type entries. The `group` type indicates posixGroup-type entries. Specification of the `<type>` parameter tells `ldapuglist` how to handle processing of search filters and attribute mappings. If you do not specify the `-t` option, `ldapuglist` assumes the `passwd` type. For example, - t group.

-h <hostname> Specifies the host name and optional port number (`hostname:port`) of the LDAP directory server. This option overrides the server list configured in the LDAP-UX configuration profile. This field supports specification of IPv4 and IPv6 addresses. Note that when you specify a port for an IPv6 address, you must specify the IPv6 address in square-bracketed form. If the optional port is unspecified, the port number is assumed to be 389 or 636 for SSL connections (with the `-Z` option). For example, `-h ldapsrvA`.

-p <port> Specifies the port number of the LDAP directory server to contact. The `ldapuglist` tool ignores this option if you specify the port number in the `<hostname>` as part of the `-h` option.

-n <name> Provides a simplified method for discovering a single account or group. Use of `-n` is the same as `-f "(uid=<name>)"` for accounts and `-f "(cn=<cname>)"` for groups. Do not specify `-f` and `-F` on the command line if you use `-n`. For example, the following command displays an account entry for the user, `mlee`:

```
ldapuglist -t passwd -n mlee
```

The output from the above command is as follows:

```
dn: cn=Mike Lee,ou=people,dc=example,dc=com
cn: Mike Lee
uid: mlee
uidNumber: 900
gidNumber: 2010
loginShell: /usr/bin/sh
homeDirectory: /home/mlee
gecos: mlee,Building-5,555-555-5555
```

-f <filter> Specifies an LDAP-style search filter, `<filter>`, used to select specific entries from the LDAP directory. When you use the `-f` option, the filter specified by `<filter>` applies to Posix-style users or groups (depending on whether you specify the `-t passwd` or `-t group` option).

The filter specified with `-f` is amended with the default ldapux(5) search filter for either the user or group object types. In addition, when you use `-f`, if a known attribute for the particular service has been mapped as defined in the LDAP-UX configuration profile, then the mapped attribute name is substituted in the search filter.

For example, if the `uidNumber` attribute has been mapped to the `employeeNumber` attribute, the following command lists a POSIX account that has `uidNumber=51552`:

```
ldapuglist -t passwd -f "(uidNumber=51552)"
```

For the above example, the mapped attribute name is substituted in the search filter, and the resulting search filter used by LDAP-UX is as follows:

```
(&(objectclass=posixAccount)(employeeNumber=51552))
```

The `-f` option also supports generation of search filters for the multi-mapped attributes, `gecos` and `memberUid`. In the case of `gecos`, each mapped attribute is used in the search filter using the LDAP and operation (&). In the case of `memberUid`, each mapped attribute is used in the search filter using the LDAP or operation (|).

In the following example, the gecos attribute has been mapped to cn, l and telephoneNumber. If the argument to -f is "(gecos=Jane Smith,BLD-5D,555-1212)", then the resulting search filter presented to the LDAP directory server is as follows:

```
(&(objectclass=posixAccount)(&(cn=Jane Smith)
(l=BLD-5D)(telephoneNumber=555-1212)))
```

As another example using memberUid, if memberUid has been mapped to member and memberUid. If the argument to -f is "(memberUid=jsmith)", then the resulting search filter presented to the LDAP directory server is:

```
(&(objectclass=posixGroup)(|(member= cn=Jane
Smith,ou=people,ou=myorg,dc=com) (memberUid=jsmith)))
```

NOTE:

- When you use -f and any of the attributes specified in the search filter have been mapped to "*NULL*", ldapuglist will return an error.
- Attributes that are not part of the LDAP-UX configuration profile mapping are not modified. For the list of attributes that may be mapped, see *RFC 2307: An Approach for Using LDAP as a Network Information Service*.
- Do not specify -n and -f on the same command line. Doing so causes an error.

-F <filter>

Similar to -f, except that the specified <filter> is immutable. The LDAP-UX user or group search filter defined in the configuration profile is not amended to the specified filter, and attribute mapping does not apply to the <filter>.

For example, the following command lists an account entry with "(uid=EricB)":

```
ldapuglist -t passwd -F "(uid=EricB)"
```

NOTE:

- When you use -F, the specified filter must apply to either user or group entries and matches the -t passwd or -t group option. The ldapuglist tool generates unpredictable results if the search filter specified with -F discovers group entries, but the -t passwd option was specified.
- Do not specify -n and -F on the same command line. Doing so causes an error.

-b <base>

This option overrides the search base as defined in the LDAP-UX configuration profile. Specifies the DN of the search base that defines where ldapuglist starts the search in an LDAP directory server. If unspecified, ldapuglist uses the defaultSearchBase as defined in the LDAP-UX configuration profile.

-s <scope>

This option overrides the search scope as defined in the LDAP-UX configuration profile. Specifies how deep in the directory tree to perform the search. The <scope> argument can be one of the following:

- **base**: Search only the entry specified in the -b option.
- **one**: Search only the immediate children of the entry specified in the -b option.

- **sub**: Perform a sub-tree search starting at the point identified in the `-b` option.

`-N <maxcount>` Specifies the maximum number of entries to be returned. If you do not specify this option, the maximum number of entries to be returned is 200 by default. Some LDAP directory servers will limit the number of entries returned for a particular search request, regardless of how many entries are requested. If the `<maxcount>` limit is set too high, it may not be possible to determine if a search has returned complete results, because the directory server might have truncated the number of returned entries before reaching the requested maximum count. Although some LDAP directory servers indicate when a specified search exceeds an enumeration limit. If the `<maxcount>` limit is above the directory server's internal configured limit, it is not always possible to determine if all results have been returned. However, a reasonable assumption is that if maximum number of entries have been returned, additional entries are likely still available to display that match the search criteria than just those displayed. For example, `-N 150`.

`<attr>` Specifies additional LDAP attributes to display aside from the predefined RFC 2307 attributes for users or groups. The `<attr>` argument may not be used if the `-L` option is specified. Attributes specified in the `<attr>` list are assumed to not be part of RFC 2307 and thus are not be mapped. When you specify the `-m` option, the output format for a value specified by an `<attr>` name is always in the following form:

```
attributename[attributename]: value
```

NOTE: The `ldapuglist` tool does not allow you to use the `<attr>` parameter when `ldapuglist` binds to the directory server using the LDAP-UX proxy user. This limitation prevents regular HP-UX users from discovering LDAP data that was previously not displayed by LDAP-UX. Use of the `<attr>` parameter requires that the user has the rights to use the LDAP-UX administrator credential (`/etc/opt/ldapux/acred`) or the user running `ldapuglist` has specified an identity using the `-P` option or the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables.

7.3.4.4 Output Format

Output from `ldapuglist` follows a consistent format, regardless of which attributes you use to define information in an LDAP directory. The output format is as follows:

```
dn: dn1
field1: value1
field2: value2
field3:: base64-encodeded-value3
...
dn: dn2
field1: value1
field2: value2
...
```

Each entry is preceded by a DN, followed by one or more field-value pairs. The DN and each field-value pair are on a separate line, separated by a carriage-return and line-feed character. The field and value are separated by a colon and a space character. Each entry is separated by a blank line. If an un-encodable character is encountered (carriage-return or line-feed for example) in a value string, the whole value is base64 encoded and the field-value separator is changed to two colons and a space character.

When you specify the `-t passwd` option, `ldapuglist` displays the following fields for a user entry:

- `cn`
- `uid`
- `userPassword`
- `uidNumber`
- `gidNumber`
- `homeDirectory`
- `loginShell`
- `gecos`

When you specify the `-t group` option, `ldapuglist` displays the following fields for a group entry:

- `cn`
- `userPassword`
- `gidNumber`
- `memberUid`

When you specify the `-m` option, the output format for both users and groups is changed to the following:

```
dn: dn1
field1[attribute1]: value1
field2[attribute2]: value2
field3[attribute3]:: base64-encodeded-value3
...
```

7.3.4.5 Special considerations for output format

This section describes special considerations for the output format from `ldapuglist` that you may need to be aware of.

7.3.4.5.1 Multi-values attributes

Although some of the attributes used in LDAP directory servers are multi-valued attributes, the `ldapuglist` tool displays only the first value discovered for each RFC 2307 attribute for each entry, because these fields appear only once in a POSIX account or group. For non-RFC 2307 attributes (those specified via the `<attr>` argument list), if the attribute is multi-valued, `ldapuglist` displays multiple values. This rule does not apply to the `memberUid` field because POSIX groups can have multiple members.

Because the `gecos` attribute can be mapped to multiple attributes, the `gecos` field can appear multiple times in an entry if you use the `-m` option, once for each mapped attribute. For example, if the `gecos` attribute is mapped to `cn`, `l` and `telephoneNumber`, `ldapuglist` displays once for each mapped attribute as follows:

```
gecos[cn]: Bill Hu
gecos[l]: Building 6A
gecos[telephoneNUmber]: +1-555-555-4321
```

7.3.4.5.2 Non-POSIX accounts and groups

If you use `ldapuglist` with the `-F` option, `ldapuglist` displays users and groups that are not posixAccounts or posixGroups. Thus, these entries may not contain the required fields that store POSIX account and group information (such as the `uidNumber` attribute). When displaying these entries, the specified fields are missing from the output. As non-POSIX accounts and groups are not required to contain POSIX attributes, use of the `-L` option may result in unexpected output. Data between the ":" characters may be empty, such as "::x:::".

7.3.4.5.3 Encoding of the DN

`ldapuglist` displays DN strings according to the encoding rules defined in RFC4514. The escape character "\" precedes special characters, which may be the character itself or a 2 digit hex representation of the character.

7.3.4.5.4 Passwords

In some cases, `ldapuglist` cannot access the user or group password fields. This can occur in the following cases:

- The `ldapuglist` tool has insufficient privilege to access the password field.
- The passwords are not used to authenticate users (such as when X.500 certificates are used).
- The password is not stored in the LDAP directory server. The password might be stored in a third-party repository such as a Kerberos Key Domain Controller.
- The password is stored in a format that cannot be parsed by HP-UX (such as {SSHA}, the Salted Secure Hash Algorithm).

If the password is not available to `ldapuglist`, `ldapuglist` does not display the `userPassword` field. If you specify the `-L` option, the password field will contain the "x" character.

7.3.4.6 Specific return codes for ldapuglist

The `ldapuglist` tool returns a list of return codes shown in Table 7-5.

Table 7-5 Return codes for ldapuglist

Return Code	Message
LST_SEARCH_FAILED	Search operation failed.
LST_COMMANDLINE_ERR	The <attr> parameter may not be used when the -L option is specified.
LST_COMMANDLINE_ERR	The requested input options cannot be specified at the same time.
LST_COMMANDLINE_ERR	The "maxcount" value must be greater than 0.
LST_SEARCH_BASE_TOO_LONG	The specified search base is too long.
LST_SEARCH_FILTER_TOO_LONG	The specified search filter is too long.
LST_ATTR_MAP_EMPTY	The attribute mapping evaluates to an empty search filter. For example, `ldapuglist -t passwd -f "(gecos=)"` The output of the command displays the "LST_ATTR_MAP_EMPTY" error because the gecos values are not specified in the command line, `ldapuglist` evaluates the gecos attribute to an empty search filter.
LST_ATTR_MAP_NULL	One or more of the attributes specified in the search filter is not mapped or mapped to *NULL*, cannot create search filter. For example, ldapuglist -t passwd -f "(userpassword=userp)" The output of the above command displays the "LST_ATTR_MAP_NULL" error because the userpassword attribute is mapped to *NULL* in the LDAP-UX configuration profile.
LST_ATTR_NOT_ALLOWD	The attribute is not allowed when bind to the directory server with the LDAP-UX proxy user.

7.3.4.7 Limitations

The `ldapuglist` tool has the following limitations:

- The `ldapuglist` tool does not support enumeration of members of a dynamic group, such as those defined by the dynamic group attributes, `memberURL` or `msDS-AzLDAPQuery`.
- The `ldapuglist` tool does not perform conversion of the locale character set to and from the UTF-8 character set.

7.3.4.8 Examples

This section provides examples of using `ldapuglist`:

While use of `LDAP_BINDDN` is not typically required to use `ldapuglist`, the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables can be used to specify the distinguished name (DN) and password of a user with sufficient directory server privilege to display protected attributes.

Setting the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables is optional when using `ldapuglist`.

The following commands specify the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables:

```
export LDAP_BINDDN = "cn=Jane Admin,ou=admins,dc=example,dc=com"
export LDAP_BINDDN = "Jane_password"
```

Run the following command to go to the `/opt/ldapux/bin` directory where `ldapuglist` resides:

```
cd /opt/ldapux/bin
```

Run the following command to display the account entry for the user name, `ascott`:

```
./ldapuglist -t passwd -L -n ascott
```

The output of the above command displays the `/etc/passwd` format as follows:

```
ascott:x:125:250:ascott:/home/ascott:/usr/bin/sh
```

Run the following command to list an account entry that contains `uid=mlee`:

```
./ldapuglist -t passwd -f "(uid=mlee)"
```

The output is as follows:

```
dn: cn=Michael Lee,ou=people,dc=example,dc=com
cn: Michael Lee
uid: mlee
uidNumber: 2201
gidNumber: 318
homeDirectory: /home/mlee
loginShell: /usr/bin/ksh
gecos: mlee,San Francisco,555-555-5555
```

Run the following command to list an account entry that contains `uid=jscott`:

```
./ldapuglist -t passwd -m -f "(uid=jscott)"
```

The output is as follows. Assume that the `gecos` attribute has been mapped to `cn`, `l`, and `telephoneNumber`. With the `-m` option, the `ldapuglist` tool displays the mapped attribute names as well.

```
dn: cn=John Scott,ou=people,dc=example,dc=com
cn[cn]: John Scott
uid[uid]: jscott
uidNumber[uidNumber]: 2225
gidNumber[gidNumber]: 252
homeDirectory[homeDirectory]: /home/mlee
loginShell[loginShell]: /usr/bin/ksh
gecos[cn]: jscott
gecos[l]: San Jose
```

```
gecos[telephoneNumber]: 555-555-9999
```

Run the following command to list an account entry having the `mfreise` account name that does not contain POSIX attributes:

```
./ldapuglist -t passwd -m -F "(uid=mfreise)"
```

The output is as follows:

```
dn: cn=Michael Freise,ou=people,dc=example,dc=com
cn[cn]: Michael Freise
uid[uid]: mlee
gecos[cn]: Michael Freise
gecos[l]: San Jose
gecos[telephoneNumber]: 555-555-5555
```

Use the following command to list all posixGroup entries that Mike Lou belongs to:

```
./ldapuglist -t group -f "(memberUid=mlou)"
```

The output is as follows:

```
dn: cn=group1,ou=groups,dc=example,dc=com
cn: group1
gidNumber: 550
memberUid: mlou
memberUid: apierce
memberUid: bjones
```

```
dn: cn=group2,ou=groups,dc=example,dc=com
cn: group2
gidNumber: 550
memberUid: vtam
memberUid: ajones
memberUid: mlou
```

Run the following command to list a regular posixGroup entry for the group name, `groupA`:

```
./ldapuglist -t group -f "(cn=groupA)"
```

The output is as follows:

```
dn: cn=groupA,ou=groups,dc=example,dc=com
cn: groupA
gidNumber: 620
memberUid: user1
memberUid: user3
memberUid: user5
```

Run the following command to list a group entry that does not require posixGroup attributes. This command uses `((cn=groupA)(objectclass=groupOfUniqueNames))` as the search filter:

```
./ldapuglist -t group -F "(&(cn=groupA)(objectclass=groupOfUniqueNames))"
```

The output is as follows:

```
dn: cn=groupA,ou=groups,dc=example,dc=com
cn: groupA
```

Run the following commands to unset the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables.

```
unset LDAP_BINDDN
unset LDAP_BINDCRED
```

7.3.5 ldapugadd tool

You can use the `ldapugadd` tool to add new POSIX accounts and groups to an LDAP directory server (as noted by the first and second syntaxes in Section 7.3.5.2 (page 233)). You can use `ldapugadd` to modify the `/etc/opt/ldapux/ldapug.conf` file to set defaults for creation of new users or groups (as noted by the third syntax, Section 7.3.5.2 (page 233)).

The `ldapugadd` tool uses user and group template files that allow `ldapugadd` to conform to the information model used for the types of entries being created. To use `ldapugadd`, you must provide LDAP administrator credentials that have sufficient privilege to perform the user or group add operation in the LDAP directory server.

This tool provides command-line options that enable you to add the following information to the user or group entry:

For POSIX accounts

- User's full name
- User ID (account name)
- User ID number
- User password
- Primary group membership
- Home directory
- Login shell
- Gecos
- Comments

For POSIX groups

- Group ID (group name)
- Group ID number
- Group members

LDAP-UX supports a local LDAP UG configuration file, `/etc/opt/ldapux/ldapug.conf`. The `ldapugadd` tool uses the `ldapug.conf` file to manage the default values for the configuration parameters, `uidNumber_range`, `gidNumber_range`, `user_gidNumber`, `default_homeDirectory` and `default_loginShell`. The `ldapugadd` tool uses these values when creating new user and group entries in an LDAP directory server if a command-line option is not provided for that specific value. You can use the `ldapugadd -D` command to change the value defined in the `ldapug.conf` file. See Section 7.3.5.5 (page 241) for more information.

Template files are required by the `ldapugadd` tool. These template files define what data is required to create new user and group entries and allow `ldapugadd` to discover required attributes. Because each organization may have different required data models for user and group entries (LDAP directory servers allow for a variety of attributes to be stored in user and group entries), these templates may define arbitrary data models beyond just the required POSIX attributes. Before creating new entries, applications can use the `ldapcfinfo` tool to discover the attributes required by the templates that are not part of the standard POSIX data model. For more information, see Section 7.3.5.6 (page 242) .

7.3.5.1 Syntax translation

LDAP-UX supports syntax translation for the `memberUid` and `gecos` attributes. This translation allows storage of this information in a format more interoperable with other directory-enabled applications. The LDAP user and group tools allow creation and modification of these attributes in the LDAP-native syntaxes, even when specified using POSIX syntaxes.

For example, if the LDAP-UX configuration profile indicates the `gecos` attribute has been mapped to `cn`, `l` and `telephoneNumber` attributes, then when you specify the GECOS values separated

by a comma for each mapped attribute in the ldapugadd command, the comma-separated list is parsed and each comma-separated component is placed in the cn, l and telephoneNumber attributes. If the memberUid attribute has been mapped to the member attribute (where the member ID syntax is defined using a distinguished name [DN]), then ldapugadd translates the memberUid account name to a DN before placing the member attribute. If the memberUid attribute has been mapped to more than one attribute type, ldapugadd uses the first attribute defined by the mapping.

7.3.5.2 Synopsis

```
ldapugadd [-t passwd] [options] <uid_name>
[-h <hostname>] [-p <port>] [-b <base>] [-u <uid_number>]
[-g <group/gid>] [-f <full_name>] [-x <domain>] [-G  <group/gid>[,...]]

[-s <login_shell>] [-d <home_directory>] [-I <gecos>][-c <comment>]
[-m [-k <skel_dir> [-T <template_file> [[<attr>=<value>][...]]
```

```
ldapugadd -t group [options] [-h <hostname>] [-p <port>]
[-b <base>] [-g <gidNumber>], [-x <domain>] [-M <member>[,...]]
[-c <comment>] [-T <template_file>] <group_name>
[[<attr>=<value>][...]]
```

```
ldapugadd -D [-g <default_gid>] [-d <default_home>] [-s <defult_shell>]
[-u <min_uid>:<max_uid>] [-g <min_gid>:<max_gid>]
```

7.3.5.3 Options

The ldapugadd tool supports the following command options:

-P Prompts for the administrator bind identity (typically LDAP DN or Kerberos principal) and bind password. If you do not specify the -P option, ldapugadd discovers the bind identity and password from the environment variables LDAP_BINDDN and LDAP_BINDCRED. Values set with a prompt (-P) option override values specified in the environment variable. If the LDAP_BINDDN and LDAP_BINDCRED environment variables have not been specified, ldapugadd uses the bind configuration specified in the LDAP-UX configuration profile. If the LDAP-UX configuration profile has specified the "proxy" bind, ldapugadd reads the bind credential from either the /etc/opt/ldapux/acred or /etc/opt/ldapux/pcred file. The /etc/opt/ldapux/acred file is only used by users that have sufficient administrative privilege to read this file.

-PP Prompts for the password of the user or group being created. If attribute mapping for the userPassword attribute in the LDAP-UX configuration profile has not been defined or set to *NULL*, ldapugadd will create new password in the userPassword attribute. To ensure accuracy, the user is prompted twice for the password. The ldapugadd tool relies on the LDAP directory server for setting of password policy, such as user-must-change-password-at-first-login.

-PW Sets the user or group password attribute. If attribute mapping for the userPassword attribute in the LDAP-UX configuration profile has not been defined or set to *NULL*, ldapugadd will create new password in the userPassword attribute. If you specify -PW, you must specify either the LDAP-UGCRED environment variable or the -PP option.

-Z Requires an SSL connection to the LDAP directory server, even if the LDAP-UX configuration profile does not specify the use of SSL. Using the -Z option requires that you define either a valid LDAP directory server or CA certificate in the /etc/opt/ldapux/cert8.db file. An error occurs if the SSL connection cannot be established.

-ZZ Attempts a TLS connection to the directory server, even if the LDAP-UX configuration profile does not specify the use of TLS. If a TLS connection cannot be established, a non-TLS and non-SSL connection will be established. HP recommends that you do not use -ZZ unless alternative methods are used to protect from network eavesdropping. Use of -ZZ requires that you define either a valid LDAP directory server or CA certificate in the /etc/opt/ldapux/cert8.db file.

-ZZZ Requires a TLS connection to the LDAP directory server, even if the LDAP-UX configuration profile does not specify the use of TLS. Using the `-ZZZ` option requires that you define either a valid directory server or CA certificate in the `/etc/opt/ldapux/cert8.db` file. An error will occur if the TLS connection can not be established.

-F Forces creation of new user or group entries even if the following error conditions occur:

- The user name or group name already exists in the directory server.
- The user ID or group ID number already exists in the directory server.
- The shell specified with the `-s` option does not exist on the local system or is not an executable.
- You attempt to add a member to a group when that member is not defined in the LDAP directory server.

Some directory servers perform their own attribute uniqueness checks. In this case, even if you specify the `-F` option, `ldapugadd` is unable to add the new entry.

-S Displays the distinguished name (DN) of the newly created entry.

7.3.5.4 Arguments

The following describes command arguments:

-h <hostname> Specifies the host name and optional port number (`hostname:port`) of the LDAP directory server. This option overrides the server list specified by the LDAP-UX configuration profile. The `<hostname>` field supports specification of IPv4 and IPv6 addresses. If you specify a port for an IPv6 address, you must specify the IPv6 address in square-bracketed form. If the optional port is unspecified, the port number defaults to 389 or 636 for SSL connections (`-Z`).

-p <port> Specifies the port number of the LDAP directory server to contact. The `ldapugadd` tool ignores this option if the port number is specified in the `<hostname>` parameter as part of the `-h` option.

-b <base> This option overrides the value of the `${basedn}` substitution construct used in the respective template file. Instead of discovering the `${basedn}` value from the LDAP-UX configuration profile, the tool uses the value defined in the `<base>` argument. See Section 7.3.5.6 (page 242) for additional information. The `<base>` value is an LDAP distinguished name.

-t <type> Specifies the service type of entry the `ldapadd` tool operates. The valid service types of this argument are `passwd` and `group`. The `passwd` type represents LDAP user entries that contain POSIX account-related information. The `group` type represents LDAP group entries that contain POSIX group-related information. If you do not specify this argument, `ldapugadd` defaults to `passwd`.

The command line arguments that are applicable depend on the service specified.

7.3.5.4.1 Arguments applicable to -D

Use the `ldapugadd -D` command to change local host default values for the UG tool configuration parameters, `uidNumber_range`, `gidNumber_range`, `user_gidnumber`, `default_homeDirectroy` and `default_loginShell`, in `/etc/opt/ldapux/ldapug.conf` file.

The following is a list of valid arguments:

-D Uses this option to permanently alter local host defaults in the `/etc/opt/ldapux/ldapug.conf` file. The `ldapugadd` tool uses these defaults when creating new user or group

entries in an LDAP directory server. Configuration changes using the `-D` options change the default values in the LDAP UG tool configuration file, `/etc/opt/ldapux/ldapug.conf`.

`-u <min_uid>:<max_uid>` Sets new default minimum and maximum ranges that `ldapugadd` uses when provisioning an UID number for newly created user entries. The UID range is inclusive of the specified end values.

`-g <default_gid>` Specifies the default group ID number used when creating new user entries. To avoid `ldapugadd` from displaying warning messages, you must specify this group ID which represents a POSIX-style group stored in the LDAP directory. If this group ID is not defined in the LDAP directory, `ldapugadd` displays a warning message every time it adds a new user using this default group ID, because `ldapugadd` cannot add the user as a member of that group.

`-g <min_gid>:<max_gid>` Sets new default minimum and maximum ranges that `ldapugadd` uses when provisioning a GID number for newly created group entries. The GID range is inclusive of the specified end values. Use the colon character to indicate that a range has been specified.

`-s <default_shell>` Specifies the default login shell to use when creating new user entries.

`-d <default_home>` Specifies the default parent home directory to use when creating new user home directories.

7.3.5.4.2 Arguments applicable to -t passwd

The following is a list of valid arguments for `-t passwd`:

`<uid_name>` Required. Specifies the POSIX style login name for the new user entry. This user name must conform to HP-UX login name requirements. For more information about login name requirements, see *passwd*(4) manpage. The `<uid_name>` argument is a required parameter. This argument must follow all command-line options and must precede the `<attr>=<value>` parameters (if provided).

`-f <full_name>` Optional. This option is required only for the `passwd` service and is used to specify the user's full name. If you do not specify this argument, the user's full name defaults to the account name.

`-u <uid_number>` Optional. Specifies the user's numeric ID number. If the specified `uidNumber` value already exists in the directory server, `ldapugadd` does not add the new entry and returns an error status, unless you specify the `-F` option.

If this argument is not specified, `ldapugadd` randomly selects a new user ID number from the uidNumber range specified by the `ldapugadd -D -u` command. If you do not specify the uidNumber range with the `ldapugadd -D -u` command, `ldapugadd` randomly selects a value from default UID range specified in the `/etc/opt/lapux/ldapug.conf` file. If `ldapugadd` randomly selects a uidNumber that is already in use on the directory server, `ldapugadd` then randomly selects another uidNumber and tries again until it finds an unused uidNumber or exhausts retry attempts. Retry attempts are limited

to 90% of the range of available uidNumbers (specified with `-D -u <min_uid>:<max_uid>`).

-g <group/gid>

Optional. Specifies the user's primary login group name or ID number. After creating the user entry, `ldapugadd` attempts to add the user as a member of the specified group using the `ldapugmod -t group` command.

To support numeric group names, `ldapugadd` always attempts to resolve the specified argument as a group name (even if it is a numeric string). If the specified argument is not found as a group name, `ldapugadd` checks to see if the argument is a numeric string and if so, uses that as the group ID number. If that numeric group cannot be found in any active name service repository, `ldapugadd` issues an ERROR message. If the specific argument is not numeric and can not be found in an active name service repository, `ldapugadd` exits with an ERROR and does not create the new entry.

If you do not specify this argument, the user becomes a member of the default login group as specified by the `ldapugadd -D -g <default_gid>` command.

-G <group/gid>[,...]

Optional. Specifies the user's alternate group memberships. `<group/gid>` is the POSIX group name or the group ID number. The specified <group> name must exist in the directory server (not in the `/etc/group` file). If the specified group name is invalid or does not exist in the directory server, `ldapugadd` issues a warning message for each invalid group. To support numeric group names, `ldapugadd` always attempts to resolve the specified argument as a group name (even if it is a numeric string). If the specified argument is not found as a group name, `ldapugadd` checks to see if the argument is a numeric string and if so, use that as the group ID number. Only if the user entry is successfully created , `ldapuguadd` will call the `ldapugmod -t group` for each <group> specified to add the user to listed groups. If you specify more than one group, you must separate each group by a comma. No white space is allowed between or within group names. If `ldapugadd` fails to add the user as a member of a particular group, `ldapugadd` issues a warning message and continues to add the user to the remaining groups specified.

If you do not specify this argument, `ldapugadd` does not add the user to alternate groups.

-s <login_shell>

Optional. Specifies the full path name to the executable that is used to handle login sessions for this user.

If this argument is not specified, the default, as configured by the `ldapugadd -D -s <default_shell>` command, is used.

-d <home_directory>

Optional. Specifies the full path name (including the user name) of the user's home directory.

If you do not specify this argument, the combination of the default base directory as configured by `ldapugadd -D -d <home_directory>` and the user's account name is used. If

you want to create the home directory on this system, you must specify the `-m` option.

`-I <gecos>`

Optional. Specifies `GECOS` fields for the user. Typically the `GECOS` argument contains the following four fields which represent (in order):

- The user's full name
- The user's work location
- The user's work telephone number
- The user's home telephone number (often omitted)

You must separate each field in the `<gecos>` argument by a comma. If the data within the `<gecos>` argument contains any white space or other characters that may be parsed by the shell, you must protect the entire string by enclosing quotes. White space cannot be used between the each field and the separating commas.

LDAP-UX supports attribute mapping of the `gecos` attribute to multiple attributes. If attribute mapping has been specified in the LDAP-UX configuration profile, each field is mapped to its representative attribute, in the order specified.

If you do not specify the -I option, ldapugadd does not add the <gecos> attribute to the user entry.

WARNING! If you specify the -I option and you have defined attribute mapping for the gecos attribute, be careful not to specify the same attributes in the command line that are also used in the gecos map. In the following example, if the gecos attribute has been mapped to cn, l, and telephoneNumber. Because -f below represents the cn attribute when creating new user account entry, the following command can produce unpredictable results because cn is specified by both -f and the gecos mapping.

```
ldapugadd -f "Jim Bailey" -I "Jim
Bailey,Boston,555-1234" jbailey \
"sn=Bailey" "telePhoneNumber=555-1234"
```

In this example, because of the gecos attribute mapping, the cn and telephoneNumber attributes are specified twice. The ldapugadd tool results an error when the same attribute and value are added to the directory server.

Use the ldapcfinfo tool to determine gecos attribute mapping configuration.

NOTE: Because the gecos attribute may be mapped to one or more attributes, the number of values specified with -I (between the commas) should, but is not required to, match the number of mapped attributes. If there are more mapped attributes than specified values in -I, then trailing mapped attributes are not added to the directory server. If more values than mapped attributes exist, extra values are combined in the last mapped attribute.

-c <comment>

Optional. Specifies a comment that will be stored in the description attribute as defined by RFC 2307. LDAP-UX does not support attribute mappings for the description attribute. If you do not specify this option, the description attribute is not added to the user entry. Because the field often contains white spaces, you must protect it from shell parsing by enclosing it in quote characters. For example:

```
-c "example description"
```

-T <template_file>

Optional. Specifies the LDIF template file to be used to create new user entries. The <template_file> parameter may be a full or relative path name or a short name. A short name is defined as the distinguishing portion of the template file name. For example, for the passwd service, if the short name "operator" is specified, the resulting template file is /etc/opt/ldapux/ug_templates/ug_passwd_operator.tmpl. All LDAP-UX default template files are stored in the /etc/opt/ldapux/

	ug_templates directory. A full or relative path name must begin with a slash (/) or a period (.) character. If you do not specify this argument, ldapugadd uses the default template file /etc/opt/ldapux/ug_templates/ug_passwd_default.tmpl.
-x <domain>	Optional. Specifies the user's domain name. Use this option to specify the ${domain} value that can be used in the template file. If you do not specify this value, the domain name is created by using the first dc component of the new user's distinguished name. If the distinguished name does not contain any dc components, and the ${domain} variable is specified in the template file, ldapugadd generates an error.
-m	Optional. Creates a new home directory for the defined user. User and group ownership of the newly created directory is assigned to the user and his/her primary login group. If the -k option is specified, the files and sub-directories found in <skel_dir> are copied to the user's home directory, and user and group ownership permissions are altered as specified above. If the -k option is not specified, skeleton files are copied from /etc/skel. The -m option requires the user has sufficient privilege to create the new home directory, copy skeleton files and change ownership of those files and directories. The ldapugadd tool creates a user's home directory only after successfully adding the user entry to the directory server and adding the user to the primary and secondary groups. If ldapugadd is unable to properly create the user's home directory, per the above process, the newly created changes in the directory server are not removed. See the "Security Considerations" section below for more information.
-k <skel_dir>	Optional. The ldapugadd tool ignores the -k option unless you specify the -m option. The <skel_dir> argument specifies a directory which contains skeleton files and directories that need to be copied into newly created user home directories. Also see -m.
<attr>=<value>[...]	Optional. Enables specification of arbitrary LDAP attributes and values. Because of potential object class requirements, additional information beyond the basic POSIX account and group data you might need to specify in order to create new entries in the LDAP directory server. For example, if the person object class is used as a structural class for posixAccounts, then the sn (surname) attribute must be specified in order to properly create a new entry. This attribute needs to be defined in the template file, and attribute/value pair needs to be specified at the end of the ldapugadd command line. The<attr>=<value> parameter is used to specify attributes required by the template file. However, if an attribute is specified which is not defined in the defined template file, that attribute/value pair is considered as an optional attribute/value which will be added to the entry exactly as specified. <attr>=<value> parameters are optional, but you must specify them as the last parameters on the command line.

7.3.5.4.3 Arguments applicable to -t group

The following is a list of valid arguments for `-t group`:

`<group_name>`

Required argument. Specifies the POSIX textual style group name for the new group entry. `<group_name>` is a required argument. It must follow all command line options and must precede the `<atr>=<value>` parameters if provided. This group name must conform to HP-UX group name requirements. For more information about group name requirements, see the *group*(4) manpage.

`-g <gidNumber>`

Optional. Specifies the group ID number. If the specified gidNumber already exists in the directory server, `ldapugadd` does not add the new entry and return an error status, unless the `-F` option is specified.

If you do not specify this argument, `ldapugadd` provisions a new group ID number by randomly selecting a value from the gidNumber range specified by the `ldapugadd -D -g <min_gid>:<max_gid>` command. If `ldapugadd` randomly selects a gidNumber that is already in use on the LDAP directory server, `ldapugadd` randomly selects another gidNumber and tries again until it finds an unused gidNumber or exhausts retry attempts. Retry attempts are limited to 90% of the range of available gidNumbers (specified with `-D -g <min_gid>:<max_gid>`).

`-x <domain>`

Optional. Specifies the group's domain name. Use this option to specify the `${domain}` value that can be used in the template file. If you do not specify this value, the domain name is created by using the first `dc` component of the new group's distinguished name. If the distinguished name does not contain any `dc` components, and the `${domain}` variable is specified in the template file, `ldapugadd` generates an error.

`-M <member>`

Optional. Defines initial group membership by adding the specified user accounts as members. If you specify more than one member, you must separate each account name by a comma. No white space is allowed between or within account names. Use of `-M` requires that the specified user's account is already defined in the LDAP directory server, unless the `-F` option is specified. When you use the `-F` option, the user's group membership is defined using the `memberUid` attribute, regardless of the attribute mapping configuration defined by the LDAP-UX configuration profile. Use of the `-F` option is not recommended, and will not succeed if the directory server does not support the `memberUid` attribute.

The `ldapugadd` tool follows the same membership syntax as defined by the LDAP_UX configuration profile attribute mapping. Specifically, if the LDAP-UX has mapped the RFC 2307 group membership attribute, `memberUid`, to a DN-based membership attribute such as `member` or `uniqueMember`, then `ldapugadd` defines membership using the DN of the specified user. If the `memberUid` attribute has been mapped to more than one attribute type, `ldapugadd` uses the first attribute defined by the mapping.

NOTE: If the `ldapugadd` tool can only add members that follow a static membership syntax (such as `memberUid`, `member` and `uniqueMember`) to a group. The `ldapugadd` tool will fail if the only mapping defined by the LDAP-UX configuration profile uses a dynamic group membership syntax (such as `memberURL`).

`-c <comment>`

Optional. Specifies a comment that is stored in the `description` attribute as defined by RFC 2307. LDAP-UX does not support attribute mappings for the `description` attribute. If you do not specify this option, the `description` attribute is not added to the group entry.

`-T <template_file>`

Optional. Specifies the LDIF template file that is used to create new group entries. If you do not specify the `-T` option, `ldapugadd` uses the default template file either `/etc/opt/ldapux/ug_templates/ug_passwd_default.tmpl` or `/etc/opt/ldapux/ug_templates/ug_group_default.tmpl` depending on the service type you specify (`-t passwd` or `-t group`).

The `<template_file>` parameter can be either a full or relative path name or a short name. See Section 7.3.5.6 (page 242) for details.

`<attr>=<value>`

Optional. Enables specification of arbitrary LDAP attributes and values. Because of potential object class requirements, additional information beyond the basic POSIX account and group data may need to be specified in order to create new entries in the LDAP directory server. For example, if the `person` object class is used as a structural class for posixAccounts, then the `sn` (surname) attribute must be specified in order to properly create a new entry. This attribute needs to be defined in the template file, and attribute/value pair needs to be specified on the `ldapugadd` command line. The `<attr>=<value>` parameter is used to specify attributes required by the template file. However, if you specify an attribute that is not defined in the defined template file, that attribute/value pair is considered as an optional attribute/value and will be added to the entry exactly as specified.

`<attr>=<value>` parameters are optional, but you must specify them as the last parameters on the command line.

7.3.5.5 LDAP UG tool configuration file

LDAP-UX supports a local configuration file, `/etc/opt/ldapux/ldapug.conf`. The `ldapugadd` tool uses the `ldapug.conf` file to manage the following default values when creating new user and group entries in an LDAP directory server:

- A default group ID for new users.
- The valid UID number range for new users.
- The valid GID number range for new groups.
- The base path for a new user's home directory. By default, LDAP-UX appends the user's account name to the base path to create the full path name.
- The default login shell for new users.

LDAP-UX provides the default `ldapug.conf` file as follows:

```
#
# This file is used by the ldapugadd tool for management
# of default values for creating new user and group entries.
```

```
# This file can not be modified directly, but instead through
# the ldapugadd -D command.
#
uidNumber_range=100:20000
gidNumber_range=100:2000
default_gidNumber=20
default_homeDirectory=/home
default_loginShell=/usr/bin/sh
```

NOTE: You can not modify the `ldapug.conf` file directly. To change the local host default values defined in the `/etc/opt/ldapux/ldapug.conf`, you must use the `ldapugadd -D` command with applicable command options to alter them. See Section 7.3.5.4.1 (page 234) for details.

7.3.5.6 Template files

Template files define user and group entries that allow `ldapugadd` to discover the required data models for new user and group entries. Template files define what object classes and attributes are required to create new user and group entries and allow `ldapugadd` to discover required attributes and data elements before creating the entries. LDAP-UX provides customers the flexibility that allows each directory deployment to define unique data models for users and groups when adding new entries to an LDAP directory server.

7.3.5.6.1 Template file naming

The `ldapugadd` tool supports multiple template files per name service. LDAP-UX only supports the `passwd` and `group` services. All template files are stored in the `/etc/opt/ldapux/ug_templates` directory. Define the template file name using the following format:

`ug_serviceName_Name.tmpl`

Where

`serviceName` Is the name of the supported service, either `passwd` or `group`.

`Name` Is the arbitrary name of the specific template file. The name, `default`, is reserved as the default template name and is used when a specific template name is not specified.

For example, `ug_passwd_default.tmpl` is the default template file for the `passwd` name service and `ug_group_default.tmpl` is the default template file for the `group` name service. `ug_passwd_vpn_user.tmpl` may be used when creating new users of "VPN" type. Template files stored outside of the `ug_templates` directory do not need to follow any specific format described above.

When specifying the name of a template file as part of the `-T` option on the command line, either the exact file name or a short name may be used. The file name can be either a full or a relative path name, but it must begin with a slash (/) or a period (.) character. That file name can exist anywhere in the file system.

When specifying a short name, the file must exist under the `/etc/opt/ldapux/ug_templates` directory and must follow the format specified above. A short name is defined as the distinguishing portion of the template file name. For example, if you define the short name "operator" for the `passwd` service, the template file can be `/etc/opt/ldapux/ug_templates/ug_passwd_operator.tmpl`. All LDAP-UX default template files are stored in the `/etc/opt/ldapux/ug_templates` directory. A full or relative path name must begin with a slash (/) or a period (.) character.

If you do not specify the `-T` option, `ldapugadd` uses the default template file either `/etc/opt/ldapux/ug_templates/ug_passwd_default.tmpl` or `/etc/opt/ldapux/ug_templates/ug_group_default.tmpl`, depending on the service type you specify (`-t passwd` or `-t group`).

7.3.5.6.2 Default template files

The LDAP-UX Integration product provides two default template files for a standard directory server for a `passwd` and `group` service entry.

Default template files for a standard directory server

Below is a default template file for the `passwd` name service:

NOTE: The template file used by the guided installation (`autosetup`) differs from this one: its template file excludes `ou=people` from the first line because that subtree is directly registered in the configuration profile.

```
dn: uid=${uid},ou=people,${basedn}
objectclass: inetOrgPerson
objectclass: posixAccount
sn: ${surname}
${posixProfile}
```

Below is a default template for the group name service:

NOTE: The template file used by the guided installation (`autosetup`) differs from this one: its template file excludes `ou=groups` from the first line because that subtree is directly registered in the configuration profile.

```
dn: cn=${cn},ou=groups,${basedn}
objectclass: groupOfNames
objectclass: posixGroup
${posixProfile}
```

Default template files for a Windows ADS

Below is a default template for the `passwd` name service:

NOTE: The template files used by the guided installation (`autosetup`) differ from these two: its template files exclude `cn=users` from the first line because that subtree is directly registered in the configuration profile.

```
dn: cn=${cn},cn=users,${basedn}
objectclass: user
${posixProfile}
sAMAccountName: ${uid}
msSFU30NisDomain: ${domain}
#By default, ldapugadd creates disabled accounts.
#Change below to 544 to enable accounts by default.
userAccountControl: 546
```

Below is a default template for the `group` name service:

```
dn: cn=${cn},cn=users,${basedn}
objectclass: group
${posixProfile}
sAMAccountName: ${cn}
msSFU30NisDomain: ${domain}
```

LDAP-UX provides two default templates file (for user and group entries) for a standard LDAP directory server, along with two default template files for Windows Active Directory Server under the `/etc/opt/ldapux/ug_templates` directory. By default, LDAP-UX creates the symbolic links for two default template files, `/etc/opt/ldapux/ug_templates/ug_passwd_default.tmpl` that points to `/etc/opt/ldapux/ug_templates/ug_passwd_std.tmpl` and `/etc/opt/ldapux/ug_templates/ug_group_default.tmpl`

that points to /etc/opt/ldapux/ug_templates/ug_group_std.tmpl for a standard LDAP directory server.

For detailed information on how to use the correct format to define template files, see Section 7.3.5.6.3 (page 244).

7.3.5.6.3 Defining template files

Pre-defined substitution constructs

Each template file must follow the LDIF data format and also permit substitution of values from the ldapugadd command. Each template file can be built using custom RFC 2307–type attributes and values. Customized attribute values are defined using the ${<name>} construct. The LDAP-UX supports several pre-defined substitution constructs, ${<name>}, where <name> represents:

posixProfile Represents all RFC 2307-type attributes and values for the particular name service (either passwd or group). If LDAP-UX configuration has defined attribute mapping for particular attributes, the mapped attributes are substituted in its place. When you use the **posixProfile** construct for posixAccount-type entries, LDAP-UX will add the following attributes and values to the new user entry:

- cn
- uid
- userPassword
- uidNumber
- gidNumber
- gecos
- homeDirectory
- loginShell

When you use the **posixProfile** construct with posixGroup-type entries, LDAP-UX will add the following attributes and values to the new group entry:

- cn
- userPassword
- gidNumber
- memberUid

NOTE: Because use of **posixProfile** supports attribute mapping, if the above attributes have been mapped as configured in the LDAP-UX configuration profile, the mapped attributes and values are added to the entry instead of the RFC 2307–defined attributes. For example, if the posixAccount attribute gecos has been mapped to cn, l and telephoneNumber then LDAP-UX adds cn, l and telephoneNumber information to the entry instead of gecos. If the posixGroup attribute memberUid has been mapped to uniqueMember, then LDAP-UX adds uniqueMember information to the entry instead of memberUid.

basedn Represents the substitution of the distinguished name of the default search base (defaultSearchBase) as defined in the LDAP-UX configuration profile.

uid Represents the user's account name when used in a passwd template file.

uidNumber Represents the user's account ID number when you define it in a passwd template file for the new user entry.

cn Represents the users's full name when you define it in a `passwd` template file. Represents the group name when you define it in a `group` template file.

gidNumber Represents the group ID number when you specify it in a group template file for the new group entry.

In addition, comments are allowed. Comments are on a separate line and the first character is the # (hash) character.

Guidelines for template files

Use the following guidelines when creating template files:

- Use the first line of the template file to define the distinguished name (DN) of the new entry. Because each DN is unique, the first component of the DN (the relative distinguished name or RDN) must be able to construct a unique value for each new entry. Define the RDN using a `${<name>}` construct. Typically, you can use the `cn` or `uid` attribute in the RDN for new user entries and the `cn` attribute for new group entries.
- Define each template file for only one entry in the LDAP directory server.
- Each template file can be built using custom attributes and values. Customized attribute values are defined using the ${<name>} construct. However, for each non-RFC 2307 attribute used, you must specify each of those attributes on the command line with an "`<attr>=<value>`" pair argument when using `ldapugadd` to create a new entry.

  For example, the following command adds the non-RFC 2307 addtribute and value pair, sn=Michael, with the UID name `Mhu` to a new user entry based on the default template file, `ug_passwd_default.tmpl`:

  ```
  ldapugadd -t passwd -f "Michael Hu" Mhu -c "an example user entry" "sn=Michael"
  ```
- Each template file can contain comment lines. Each comment line must begin with the "#" character.
- Do not specify the `userPassword` attribute in the template file. Use the `-PP` option or the `LDAP_UGCRED` environment variable to specify an initial password of the user or group being created.
- You cannot specify the `memberUid` attribute in the template file, because the number of eventual members of a group can not be statically defined when the group is newly created. The `ldapugadd` tool ignores the `memberUid` attribute if specified in the template file.

7.3.5.6.4 Multi-valued attributes in template files

LDAP-UX supports multi-valued attributes defined in a template file. This means that the same attribute name and/or value can be specified more than once in the template file.

For example, in the following template file, `secondaryTeams` is a multi-valued attribute that can be specified twice for each new posixAccount entry created. In this case, `ldapugadd` will fill each attribute value in order specified in the template file based on the order that those attributes are specified on the command line. If not enough attribute values are specified on the command line to fill the attribute values used in the template file, `ldapugadd` returns an error.

```
dn: uid=${uid},ou=people,${basedn}
objectclass: person
objectclass: myOrg
objectclass: posixAccount
sn: ${sn}
primaryTeam: ${primaryTeam}
secondaryTeams: ${secondaryTeams}
secondaryTeams: ${secondaryTeams}
${posixProfile}
```

7.3.5.7 Security considerations

The following are security considerations when using `ldapugadd`:

- Use of `ldapugadd` requires permissions of an LDAP administrator when it performs its operations on the directory server. The rights for creation of new LDAP directory entries under the requested subtree, along with creation of the required attributes in that entry must be granted to the LDAP administrator identity when executing `ldapugadd`.
- As with any POSIX-type identity, the HP-UX operating system uses the specified user and group ID number to determine rights and capabilities in the OS as well as in the file system. For example, the root user ID 0, typically has unlimited OS administration and file access rights. Before creating a new entry, you must be aware of the selected user and group ID number and any policy that may be associated with that ID.
- If you use `ldapugadd` to randomly assign a user or group ID number, it only checks for ID collisions found in the LDAP directory server, and not other policy repositories. When you set user and group ID number ranges by using the `-D -u` or `-D -g` option, you must set a range that is not used by other user or group ID repositories, and ensure that collisions will not occur with existing users or groups that exist in other repositories.
- Modification of this identity repository will likely have impacts as defined by the organization's security policy. Users of `ldapugadd` are expected to have full knowledge of the impact to the organization's security policy when adding new identity information to that identity repository.

7.3.5.8 Specific return codes for ldapugadd

The `ldapugadd` tool returns a list of return codes shown in Table 7-6.

Table 7-6 Return codes for ldapugadd

Return Code	Message
ADD_USER_TO_GRP_FAILED	Failed to add a user to the group.
ADD_SKELDIR_DOESNOT_EXIST	Specified Skeleton directory does not exist.
ADD_SETENV_FAILED	The `ldapugadd` tool failed the internal `putenv` function call with the specified bind environment variable, it returns this error.
ADD_INFO_MISSING	Information is missing. For examples, UID number is missing, group number is missing.
ADD_GETNUM_FAILED	Failed to get a valid gid number or UID number when creating a new user or group entry.
ADD_SYNTAX_ERR	A syntax error exists in the specified template file.
ADD_ATTR_REQUIRED	Attribute is required. For examples, attribute "`sn`" is required, attribute "`telephonenumber`" is required.
ADD_NUM_RANGE_ERR	Specified option has invalid range value. For example, option `-u` has invalid range value.
ADD_WRONG_G_OPT	Option `-g <default_gid>` or `-g <min_gid>:<max_gid>` has been specified more than once.
ADD_NOT_PERMIT	You do not have the permission to alter `/etc/opt/ldapux/ldapug.conf`.

Table 7-6 Return codes for ldapugadd *(continued)*

ADD_INVALID_KEYWORD	The specified keyword value is invalid, `ldapugadd` ignored the keyword. For example, if `/usr/bin/jsh` does not exist in the system, the `ldapugadd -D -s /usr/bin/jsh` command displays the following warnings:
	`WARNING: LOGIN_SHELL_DOESNOT_EXIST: Login shell /usr/bin/jsh' does not exist.`
	`WARNING: ADD_INVALID_KEY Invalid keyword (default_loginShell), ignored.`

ADD_RENAME_FAILED	Failed to rename the internal temporary file to `/etc/opt/ldapux/ldapug.conf`.
ADD_UPDATE_OK	A specific operation has been updated successfully. For example, “uidnumber_range” defined in `ldapug.conf` has been updated successfully.
ADD_K_IGNORED	Option `-m` is not specified, therefore, `-k` ignored when adding a new account.
ADD_TWO_DN_ERR	DN has been specified more than once.
ADD_GID_GNAME_ERR	Options `-g` and `-e` cannot be specified at the same time.
ADD_NOT_IN_LDAP	The specified group does not exist in the LDAP directory. Could not add a user to the specified group.
ADD_FAIL_TO_UPDATE	Failed to update the default value in `/etc/opt/ldapuux_ldapug.conf`.
ADD_FAILED	The LDAP add operation failed.

7.3.5.9 Limitations

The following are limitations of `ldapugadd`:

- Because LDAP directory servers require data to be stored according to the UTF-8 (RFC3629) character encoding method, all characters passed into `ldapugadd` are assumed to UTF-8, and part of the ISO-10646 character set. `ldapugadd` does not perform conversion of the locale character set to and from the UTF-8 character set.
- Because `ldapugadd` calls functions to discover if the group exists before adding a user to a group, it is possible to encounter timing issues with cached information. For example, if an administrator uses the `grget` command to see if a group exists, this group information is cached by both `ldapclientd (1M)` and `pwgrd(1M)`. If the group does not exist when calling `grget`, and the administrator shortly creates this group with `ldapugadd`, the information that the group still does not exist will still be cached. Then, when adding a new user and specifying that this user is a member of the just created group, `ldapugadd` generates an error to indicate that the user cannot be added to the group. To resolve this, you must flush the `pwgrd` and `ldapclientd` caches.

7.3.5.10 Examples

This section provides examples of using the `ldapugadd` tool:

The following commands specify the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables:

```
export LDAP_BINDDN = "cn=Jane Admin,ou=admins,dc=example,dc=com"
export LDAP_BINDCRED = "Jane's Password"
```

The following command specifies the `LDAP_UGCRED` environment variable:

```
export LDAP_UGCRED = "user_password"
```

Run the following commands to discover what non-POSIX attributes defined in the default template file are required to create the new user entry:

```
cd /opt/ldapux/bin
./ldapcfinfo -t passwd -R
```

The output of the commands is as follows:

```
Surname
```

The following commands add an account entry for the user, `alam`, with the user's primary login group id, 300, and the surname, `Lam`. The `ldapugadd` tool creates the password for new user, `alam`, using the user password specified in the `LDAP_UGCRED` environment variable. After creating the user entry, `ldapugadd` attempts to add this user as a member of the group number 300. The `uidNumber` value is assigned dynamically from the pre-configured range.

```
cd /opt/ldapux/bin
./ldapugadd -t passwd -PW -f "Adrian Lam" -g 300 alam surname="Lam"
```

Run the following command to display the new user entry, `alam`:

```
./ldapuglist -t passwd -n alam sn
```

Below is the user entry:

```
dn: cn=Adrian Lam,ou=people,dc=example,dc=com
cn: Adrian Lam
uid: alam
uidNumber: 2200
gidNumber: 300
homeDirectory: /home/alam
loginShell: /usr/bin/ksh
sn: Lam
```

The following command adds an account entry for the user, `mscott`, with the user's primary login group id, 200, and `gecos` field information. In this example, the `gecos` attribute has been mapped to `cn`, `l` and `telephoneNumber` in the LDAP-UX configuration profile. `ldapugadd` creates the password for new user, `mscott`, using the password specified in the `LDAP_UGCRED` environment variable. After creating the user entry, `ldapugadd` attempts to add this user as a member of the group number 200.

```
./ldapugadd -t passwd -PW -g 200 \
-I "Mike Scott,Building-3A,555-555-5555" mscott surname="Scott"
```

Use the following command to display the new user entry, `mscott`, with mapped attribute information:

```
./ldapuglist -t passwd -m -n mscott
```

Below is the user entry:

```
dn: cn=Mike Scott,ou=people,dc=example,dc=com
cn[cn]: Mike Scott
uid[uid]: mscott
uidNumber[uidnumber]: 2200
gidNumber[gidnumber]: 200
homeDirectory[homedirectory]: /home/mscott
loginShell[loginshell]: /usr/bin/sh
gecos[cn]: Mike Scott
gecos[l]: Building-3A
gecos[telephoneNumber]: 555-555-5555
```

The following command adds an account entry for the user, `mwang`, with the user's primary login group id, 350. In this example, `ldapugadd` creates the user home directory `/home/wang` and assigns user and group ownership of the newly created directory to the user `mwang` and his primary login group after successfully adding the user entry to the directory server and adding the user to the primary login group. `ldapugadd` uses the password specified in the `LDAP_UGCRED` environment variable to create the password for the new user, `mwang`.

```
./ldapugadd -t passwd -PW -f "Mike Wang" -g 350 \
-m -d "/home/wang" mwang surname="Wang"
```

Use the following command to display the new user entry, `mwang`:

```
./ldapuglist -t passwd -n mwang sn
```

The output of the user entry is as follows:

```
dn: cn=Mike Wang,ou=people,dc=example,dc=com
cn: Mike Wang
uid: mwang
uidNumber: 2255
gidNumber: 350
homeDirectory: /home/wang
loginShell: /usr/bin/sh
sn: Wang
```

The following command adds a new group entry for the group name, `groupA`. In this example, `ldapugadd` creates the new group, `groupA`, and defines the initial group membership by adding the user account, `mwang`, as a member.

```
./ldapugadd -t group -M mwang groupA
```

Use the following command to display the new group entry, `groupA`:

```
./ldapuglist -t group -f "(cn=groupA)"
```

The output of the group entry is as follows:

```
dn: cn=groupA,ou=Group,dc=example,dc=com
cn: groupA
gidNumber: 550
memberUid: mwang
```

The following command sets new default minimum and maximum ranges of UID numbers in the local configuration file, `/etc/opt/ldapux/ldapug.conf`. When creating a new user account, the `ldapugadd` tool randomly selects a new ID from this range if an account number has not been specified.

```
./ldapugadd -D -t passwd -u 200:5000
```

The following command sets new default minimum and maximum ranges of GID numbers in the local configuration file, `/etc/opt/ldapux/ldapug.conf`. When creating a new group, the `ldapugadd` tool randomly selects a new ID from this range if a group number has not been specified.

```
./ldapugadd -D -t group -g 300:3000
```

The following command sets the new default group ID number in the local configuration file, `/etc/opt/ldapux/ldapug.conf`. The `ldapugadd` tool uses this value when creating a new user entry in an LDAP directory server.

```
./ldapugadd -D -t passwd -g 500
```

The following command sets the new default login shell in the local configuration file, `/etc/opt/ldapux/ldapug.conf`. The `ldapugadd` tool uses this login shell when creating a new user entry in an LDAP directory server.

```
./ldapugadd -D -t passwd -s /usr/net/bin/sh
```

Run the following commands to unset the `LDAP_BINDDN`, `LDAP_BINDCRED` and `LDAP_UGCRED` environment variables:

```
unset LDAP_BIND
unset LDAP_BINDCRED
unset LDAP_UGCRED
```

7.3.6 ldapugmod tool

The ldapugmod tool enables HP-UX administrators to modify existing POSIX accounts or groups in an LDAP directory server. When using extended options, you can use ldapugmod to modify arbitrary attributes for user or group entries or you can extend existing user or group entries with the POSIX data model. To use ldapugmod, you must provide LDAP administrator credentials that have sufficient privilege to perform the user or group modification operations in the LDAP directory server.

7.3.6.1 Synopsis

```
ldapugmod [-t passwd] [options]  [-h <hostname>] [-p <port>]
[-f <full_name>] [-n <new_name>] [-u <uid_number>] [-g <group/gid>]
[-s <login_shell] [-d <home_directory>[-m]] [-c <comment>] [-I <gecos>]
[[-A <attrval>][...]]
[[-R <attrval>][...]] [-D <DN>|<uid_name>]
[[<attr>=<value>][...]]
ldapugmod -t group [options]  [-h <hostname>] [-p <port>]
[-n new_name>] [-g <gid_number>] [-a <member>[,...]] [-r <member>[,...]]
[-c <comment>] [[-A <attrval>][...]][[-R <attrval>][...]]
[-D <DN>|<group_name>] [[<attr>=<value>][...]]
```

7.3.6.2 Options

The ldapugmod tool supports the following command options:

-P Prompts for the administrator's bind identity (typically LDAP DN or Kerberos principal) and bind password. If you do not specify the -P option, ldapugmod discovers the bind identity and password from the environment variables LDAP_BINDDN and LDAP_BINDCRED. If the LDAP_BINDDN and LDAP_BINDCRED environment variables have not been specified, ldapugmod uses the bind configuration specified in the LDAP-UX configuration profile. If the LDAP-UX configuration profile has specified the "proxy" bind, ldapugmod reads the bind credential from either the /etc/opt/ldapux/acred or the /etc/opt/ldapux/pcred file. The /etc/opt/ldapux/acred file is only used by users who have sufficient administrative privilege to read this file.

-PP Prompts for the password of the user or group being modified. If you do not specify the -PP option, the ldapugmod tool retrieves the password for the modified user or group from the LDAP_UGCRED environment variable if the -PW option is specified. Use of -PP implies the use of -PW.

-PW Changes the user or group password attribute. If attribute mapping for the userPassword attribute in the LDAP-UX configuration profile has not been defined or set to *NULL*, ldapugmod will create new passwords in the userPassword attribute. If you specify the -PW option, you must also specify either the LDAP_UGCRED environment variable or the -PP option.

-O With ldapugmod, you can add posixAccount and posixGroup attributes to a user or group entry that does not already contain the posixAccount or posixGroup object class. This ability requires use of the -D option. With the -O option, ldapugmod attempts to add the posixAccount or posixGroup objectclass and respective attributes (depending on if the -t passwd or -t group option is specified) to the entry being modified. When used with Microsoft Windows Active Directory service, if the user or group entry is build using the abstract "User" or "Group" class, ldapugmod assumes that the abstract class already includes the required Microsoft SFU attributes, and thus does not add the posixAccount or posixGroup object class to the entry.

-Z Requires an SSL connection to the LDAP directory server, even if the LDAP-UX configuration profile does not specify the use of SSL. If you use the -Z option, you must define either a valid directory server or CA certificate in the /etc/opt/ldapux/cert8.db file. An error occurs if the SSL connection can not be established.

-ZZ Attempts a TLS connection to the directory server, even if the LDAP-UX configuration does not require the use of TLS. If a TLS connection cannot be established, a non-TLS and non-SSL connection will be established. Do not use `-ZZ` unless alternative methods are used to protect against network eavesdropping. Use of `-ZZ` requires that you define a valid server or a CA certificate in the `/etc/opt/ldapux/cert8.db` file.

-ZZZ Requires a TLS connection to the LDAP directory server, even if the LDAP-UX configuration profile does not specify the use of TLS. Using the `-ZZZ` option requires that you define a valid directory server or a CA certificate in the `/etc/opt/ldapux/cert8.db` file. An error occurs if the TLS connection cannot be established.

-N Allows you to rename the relative distinguished name (RDN) of an LDAP directory server. In some cases, when an attribute is modified, it might be the same attribute that is used in the RDN portion of the entry's distinguished name. Changing the attribute and value that is used in the RDN requires changing the RDN.

For example, an entry in the directory server is named "`cn=Robert Smith,ou=IT,dc=example,dc=com`". If the `cn` attribute is changed to "`cn=Bob Smith`", then the entry DN also needs to change to "`cn=Bob Smith,ou=IT,dc=example,dc=com`"

Modification of an RDN is generally discouraged because the DN is often used as a unique way to identify the entry in the directory server. Often DN is used to define membership in a group. To prevent accidental changes to the DN, you must specify the `-N` option to allow changes to the RDN. When the DN of an entry changes, the group membership information for this entry might be inconsistent. Most directory servers have the inherent ability to update all entries that refer to the updated DN of a changed entry. Therefore, `ldapugmod` does not attempt to perform modifications to other entries in the directory server that refer to this entry by its DN.

NOTE: The `ldapugmod` tool does not allow you to rename multi-valued RDNs. For example, an RDN of "`cn=test1+cn=test2`" is not supported

-F Forces `ldapugmod` to modify the user or group entry in an LDAP directory server even if particular error conditions occur. Those error conditions that can be overwritten are as follows:

- The changed user name or group name already exists in the directory server.
- The changed user ID or group ID number already exists in the directory server.
- An attempt is made to modify the group of a user with a group ID that cannot be found in any name service repository. In this case, the group ID number must be specified.
- An attempt is made to force `ldapugmod` to add a member to a group when that member is not defined in the LDAP directory server. In this case, membership is always defined using the `memberUid` attribute, regardless of attribute mapping defined for group membership.

-S Displays the Distinguish Name (DN) of the deleted or updated entry when the operation successfully completes.

7.3.6.3 Arguments

The following describes command arguments:

-t <type> Specifies whether the command-line arguments are applicable to modify the user or group entry. The valid types of this argument are `passwd` and `group`. If you do not specify this argument, `ldapugmod` defaults to `passwd`. The `passwd` type represents LDAP user entries that contain POSIX

account-related information. The group type represents LDAP group entries that contain POSIX group-related information.

-h <hostname> Specifies the host name and optional port number (hostname:port) of the LDAP directory server. This option overrides the server list specified by the LDAP-UX configuration profile. This field supports specification of IPv4 and IPv6 addresses. If you specify a port for an IPv6 address, you must specify the IPv6 address in square-bracketed form. If you do not specify the optional port, the port number defaults to 389 or 636 for SSL connections (-Z).

-p <port> Specifies the port number of the LDAP directory server to contact. The ldapugadd tool ignores this option if you specify the port number in the <hostname> parameter as part of the -h option.

-D <DN> The ldapugmod tool searches for the named user or group using the search rules defined by the service search descriptor in the LDAP-UX configuration profile. You can use the -D option to specify the exact distinguished name (DN) of the entry being modified. If you specify the -D option, you do not need to specify the <uid_name> or <group_name> parameter.

-A <attrval> Specifies an attribute and value to be added to an entry. The format of <attrval> is "attribute=value", where attribute is the name of the attribute to add, and value is the specific instance of that attribute. When working with multi-valued attributes, you can use the -A option to add a new value for a multi-valued attribute, without removing already existing values for that attribute. The use of the -A parameter interacts with the optional <attr>=<value> parameters. You can specify the -A option more than once per command line. The value portion of the <attrval> may be an empty string.

For example, if an entry in an LDAP directory is as follows:

```
dn: uid=mLee,ou=people,dc=example,dc=com
cn: Mark Lee
cn: Michael Lee
uid: mLee
uidNumber: 2200
gidNumber: 212
homeDirectory: /home/mLee
loginShell: /usr/bin/ksh
gecos: Mark Lee,San Jose,+1 555-555-5555
```

Perform the following ldapugmod command for the user entry, mLee:

```
ldapugmod -t passwd -A "cn=Mackey Lee" mLee
```

The above command adds an instance of the cn attribute, cn=Mackey Lee to the entry. The following is the result of the mLee entry:

```
dn: uid=mLee,ou=people,ou=IT,dc=example,dc=com
cn: Mark Lee
cn: Michael Lee
cn: Mackey Lee
uid: mLee
uidNumber: 2200
gidNumber: 212
homeDirectory: /home/mLee
loginShell: /usr/bin/ksh gecos: Mark Lee,San Jose, +1
555-555-5072
```

-R <attrval> Specifies an attribute or specific values of an attribute to be removed from the entry. The format of <attrval> is attribute=value, where attribute is the name of the attribute to remove, and value is a specific instance of that attribute if the attribute is multi-valued. The use of the -R option

interacts with the optional <attr>=<value> parameters. See the <attr>=<value> option below for details. You can specify the -R option more than once per command line.

-n <new_name> Specifies the new name of the user or group. This option replaces the uid attribute for user entries or the cn attribute for group entries with the new name, or the mapped attribute if attribute mapping has been specified for that attribute. The <new_name> argument specifies the new name of the user or group. Using -n is the same as replacing the corresponding attribute. For example, the following two commands perform the same operation, replacing the old UID with new UID for a user entry (assuming no attribute mapping) :

```
ldapugmod -t passwd -n newuid olduid
```

Is the same as:

```
ldapugmod -t passwd olduid "uid=newuid"
```

7.3.6.3.1 Options applicable to -t passwd

The following is a list of valid options for -t passwd:

<uid_name> Required. Specifies the POSIX style login name of the user entry to modify. You must specify the <uid_name> parameter unless you specify the -D option. This user name must conform to HP-UX login name requirements. For more information about login name requirements, see the *passwd*(4) manpage.

-f <full_name> Replaces the user's full name. If is an empty string (a pair of double quotes: ""), ldapugmod removes the cn (or mapped) attribute. See the "WARNING" section below for impacts when using this option.

-u <uidNumber> Replaces the user's numeric ID number. If uidNumber is an empty string (a pair of double quotes: ""), ldapugmod removes the uidNumber or mapped attribute. If the specified uidNumber value already exists in the directory server, ldapugmod does not modify the entry and returns an error exit status, unless you specify the -F option.

-g <group/gid> Replaces the user's primary login group ID number. If <group/gid> is an empty string (a pair of double quotes: ""), ldapugmod will remove the gidNumber or mapped attribute. In order to support numeric group names, ldapugmod treats the -g argument as a group name. If ldapugmod cannot find a matched numeric group name in the directory server, it checks to see if the value is numeric and then checks to see if the specified group ID number exists. If it does not exist, ldapugmod exits with an error, unless you specify the -F option.

NOTE: The dapugmod tool does not modify the user's group membership when chaining the primary group ID. Adding the user as a member of the new group and possibly removing the member from the previous group must be done with separate ldapudmod operations.

-s <login_shell>

Replaces the full path name to the executable that is used to handle login sessions for this user.

If the <login_shell> argument is an empty string (a pair of double quotes: ""), ldapugmod removes the loginShell or mapped attribute.

The ldapudmod tool issues a WARNING if the specified login shell does not exist on the local system. See the "WARNING" section below for impacts when using this option.

-d <home_directory>

Replaces the full path name (including the user name) of the user's home directory. If the <home_directory> argument is an empty string (a pair of double quotes: ""), ldapugmod removes the homeDirectory or mapped attribute.

-m

Move the user's home directory to the location specified with the -d option. -m requires that you also specify the -d option. If the specified <home_directory> already exists, the user's current home directory does not exist or the user running ldapugmod does not have sufficient permissions to move the directory, ldapugmod returns an error.

-I <gecos>

Replaces gecos fields for the user. If <gecos> is an empty string, ldapugmod removes the gecos or mapped attribute(s).

Typically the gecos argument contains four fields which represent in the following order:

- The user's full name
- The user's work location
- The user's work telephone number
- The user's home telephone number (often omitted)

Each field in the <gecos> argument must be separated by a comma. Although the fields specified within the <gecos> argument can contain white space (such as "Bill Smith,Building 6,555-1234"). White space cannot be used between each field and the separating commas.

LDAP-UX supports attribute mapping of the gecos field to multiple attributes. If attribute mapping has been specified in the LDAP-UX configuration profile, each field is mapped to its representative attribute, in the order specified.

WARNING! If you specify the `-I` option and you have defined attribute mapping for the `gecos` attribute, be careful not to specify the same attributes in the command line that are also used in the `gecos` map. In the following example, the `gecos` attribute has been mapped to `cn`, `l`, and `telephoneNumber` attributes. The following command can produce unpredictable results:

```
ldapugmod -I "lisa Hu,Austine,222-1234" lhu "cn=lisa
 Hu" "sn=Hu"\
"telePhoneNumber=222-1234"
```

In the above example, because of the `gecos` attribute mapping, the `cn` and `telephoneNumber` are specified twice, it results an error when the same attribute and value are added to the directory server. Use the `ldapcfinfo` tool to check `gecos` attribute mapping configuration.

If the `<gecos>` argument is an empty string, `ldapugmod` removes the `gecos` or mapped attributes. HP does not recommend that you use the `-I` option, because the `gecos` attribute is often mapped to required attributes. See the "WARNING" section below for impacts when using this option.

`-c <comment>` Replaces a comment that will be stored in the `description` attribute as defined by RFC 2307. LDAP-UX does not support attribute mappings for the `description` attribute.

`<attr>=<value>` Enables modification of arbitrary LDAP attributes and values. The `<value>` parameter may be an empty string. However this usage does not remove attributes and their values from the directory server. Instead use the `-R` option to remove arbitrary attributes. See the "WARNING" section below for impacts when using this option

7.3.6.3.2 Options applicable to -t group

The following is a list of valid options for `-t group`:

`<group_name>` Required. Specifies the POSIX style textual group name for the group entry to modify. You must specify the group name if you do not specify the `-D` option. This group name must conform to HP-UX group name requirements. For more information about group name requirements, see the *group*(4)manpage.

`-g <gidNumber>` Replaces the group's numeric ID number. If the specified `gidNumber` value already exists in the directory server, `ldapugmod` does not modify the group entry and return an error status, unless you specify the `-F` option.

`-a <member>[,...]` Adds one or more members to the specified group.

The `ldapugmod` tool follows the same membership syntax defined by the LDAP-UX configuration profile attribute mapping. Specifically, if LDAP-UX has mapped the RFC 2307 group membership attribute, `memberUid`, to a DN-based membership attribute such as `member` or `uniqueMember`, then `ldapugmod` defines membership using the DN of the specified user. When specifying a list of members, you must use a comma with no white space to separate each member. If the `memberUid` attribute has been mapped to more than one attribute

type, ldapugmod uses the first attribute defined by the mapping. If the specified <member> does not exist in the LDAP directory, you must use -F to define the member, and only use the memberUid attribute syntax.

NOTE: The ldapugmod tool can add members only to a group that follow a static membership syntax (such as memberUid, member and uniqueMember). If the only membership mapping defined in the LDAP-UX configuration profile uses a dynamic group membership syntax (such as memberURL), ldapugmod fails to add a member to a group.

-r <member>[,...]

Removes one or more members from the specified group.

The ldapugmod tool searches for membership in the group using the memberUid, member, uniqueMember, and msSFU30posixMember attributes and removes all values that represent the specified user (either DN or UID name). The ldapugmod tool consults the LDAP-UX configuration profile for attribute mappings to determine which attributes need to be modified to remove the user membership. When specifying a list of members, you must use a comma with no white space to separate each member.

-c <comment>

Replaces a comment that is stored in the description attribute as defined by RFC 2307. LDAP-UX does not support attribute mappings for the description attribute. If <comment> is an empty string, ldapugmod removes the description or mapped attribute.

<attr>=<value>>

Enables modification of arbitrary LDAP attributes and values. The <value> parameter may be an empty string. However this usage does not remove attributes and their values from the directory server. Instead, use the -R option to remove arbitrary attributes. See the "WARNING" section below for impacts when using this option

7.3.6.4 Warnings

Under common usage, ldapugmod uses the LDAP replace operation when changing values of an attribute in an entry. This feature can impact attributes that have multiple values, by removing all occurrences of an attribute value and replacing it with the one specified on the ldapugmod command line. For example, if the -n argument is used to specify a new name for a posixGroup, all occurances of the cn attribute are replaced by the value specified for the -n argument. This mode of operation applies to all command argument specified values, including -u, -g, -s, -d, -I and -c.

When you use the <attr>=<value> parameter to modify an existing attribute, the ldapugmod command also uses the LDAP replace operation. The replace operation removes all occurrences of the specified attribute for an entry and replaces it with the value specified. If there are multiple values for a single attribute in an entry, the use of a single <attr>=<value> parameter will replace all values with the single value specified on the command line. You can specify more than one occurrence of the same attribute on the command line, if that attribute is multi-valued. In that case, both values are created in the entry.

Use of -A or -R changes this behavior (for both the above-listed command arguments and the <attr>=<value> parameter). Any attribute specified as an argument to the -A or -R causes ldapugmod to perform an LDAP add operation instead of an LDAP replace operation.

NOTE: The `ldapugmod` tool does not allow you to use the same attribute and value pair more than once, either as part of `<attr>=<value>`, `-R` or `-A`, or with other command line options. The `ldapugmod` tool exits with error status before sending any conflict modification request to the LDAP directory server.

Example 1

In this example, an entry in an LDAP directory is as follows:

```
dn: uid=mLee,ou=people,dc=example,dc=com
cn: Mark Lee
cn: Michael Lee
uid: mlee
uidNumber: 2200
gidNumber: 212
homeDirectory: /home/mlee
loginShell: /usr/bin/ksh
gecos: Mark Lee,New York,555-666-6000
description: test user entry
description: multi-valued attribute entry
```

Perform the following `ldapugmod` command for the user entry, `mlee`:

```
cd /opt/ldapux/bin
./ldapugmod -t passwd mlee "cn=Mackey Lee"
```

The above commands replace all instances of `cn` with the single value, `Mackey Lee`. The resulting `mlee` entry is as follows:

```
dn: uid=mLou,ou=people,dc=example,dc=com
cn: Mackey Lee
uid: mlee
uidNumber: 2200
gidNumber: 212
homeDirectory: /home/mlee
loginShell: /usr/bin/ksh
gecos: Mark Lee,New York,555-666-6000
description: test user entry
description: multi-valued attribute entry
```

Perform the following `ldapugmod` command for the user entry, `mlee`:

```
./ldapugmod -t passwd -c "Mackey user entry" mlee
```

This command replaces all instances of `description` with the single comment, `Mackey user entry`. The result of the `mlee` entry is as follows:

```
dn: uid=mLou,ou=people,dc=example,dc=com
cn: Mackey Lee
uid: mlee
uidNumber: 2200
gidNumber: 212
homeDirectory: /home/mlee
loginShell: /usr/bin/ksh
gecos: Mark Lee,New York,555-666-6000
description: Mackey user entry
```

Example 2

In this example, the entry in an LDAP directory is as follows:

```
dn: uid=slou,ou=people,dc=example,dc=com
cn: Smith Lou
cn: Smitta Lou
uid: slou
uidNumber: 2500
gidNumber: 120
homeDirectory: /home/slou
```

```
loginShell: /usr/bin/ksh
gecos: Smith Lou,San Jose,+1 555-510-5000
```

Perform the following `ldapugmod` command for the user entry, `slou`:

```
./ldapugmod -t passwd -R "cn=Smitta Lou" slou "cn=Smitty Lou"
```

The above command removes the instance of `Smitta Lou` and replaces it with the value, `Smitty Lou`. The resulting `slou` entry is as follows:

```
dn: uid=slou,ou=people,dc=example,dc=com
cn: Smith Lou
cn: Smitty Lou
uid: slou
uidNumber: 2500
gidNumber: 120
homeDirectory: /home/slou
loginShell: /usr/bin/ksh
gecos: Smith Lou,San Jose,+1 555-510-5000
```

Example 3

In this example, the entry in an LDAP directory is as follows:

```
dn: uid=jscott,ou=people,dc=example,dc=com
cn: John Scott
cn: Joe Scott
uid: jscott
uidNumber: 2500
gidNumber: 120
homeDirectory: /home/jscott
loginShell: /usr/bin/ksh
gecos: John Scott,San Jose,+1 555-555-5555
```

Perform the following `ldapugmod` command for the user entry, `jscott`:

```
./ldapugmod -t passwd -A "cn=Joesh Scott" jscott
```

This command adds an instance of the `cn` attribute, `cn=Joesh Scott` to the entry. The result of the user entry is as follows:

```
dn: uid=jscott,ou=people,dc=example,dc=com
cn: John Scott
cn: Joe Scott
cn: Joesh Scott
uid: jscott
uidNumber: 2500
gidNumber: 120
homeDirectory: /home/jscott
loginShell: /usr/bin/ksh
gecos: John Scott,San Jose,+1 555-555-5555
```

7.3.6.5 Specific return codes for ldapugmod

The `ldapugmod` tool returns a list of return codes shown in Table 7-7.

Table 7-7 Return codes for ldapugmod

Return Code	Message
MOD_CANNOT_GET_USER_HOMEDIR	Cannot discover user's home directory information.

Table 7-7 Return codes for ldapugmod *(continued)*

MOD_COMMANDLINE_ERR	Member(s) need to be specified for the specified option. For exmaple,
	`ldapugmod -t group -r ""`
	The output of the command is as follows:
	`ERROR: MOD_COMMANDLINE_ERR: member(s) need to be specified for -r option.`
	`ldapugmod -t group -a ""`
	The output of the command is as follows:
	`ERROR: MOD_COMMANDLINE_ERR: member(s) need to be specified for -a option.`

MOD_MEMBER_SKIPPED	Cannot remove user account from the specified group, will be skipped.
MOD_DUP_REQUEST	Duplicate modification requests are found in the command options. For example, `ldapugmod -A "cn=Mike Lee" -A "cn=Mike Lee" mlee` After running the above command, `ldapugmod` exits with the `MOD_DUP_REQUEST` error status because duplicate modification requests are specified.
MOD_CONFLICT_REQUEST	Conflict modification requests are found in the command options.
MOD_RENAME_RDN_FAILED	Rename entry's RDN failed.
MOD_NEW_RDN_NEEDED	The specified command deletes the existing value in the RDN, but no new value for the RDN has been provided.
MOD_MEMBER_EXIST	The account entry being added is already a member of the specified group.
MOD_HOMEDIR_DOESNOT_EXIST	The user's home directory does not exist.
MOD_MISSING_INFORMATION	Cannot move user's home directory, missing information.

7.3.6.6 Security considerations

Be aware of the following security considerations when you use `ldapugmod`:

- The `ldapugmod` tool requires an LDAP administrator permissions when it performs operations on the directory server. The rights to modify existing LDAP directory entries under the requested subtree, and to create, modify and remove the required attributes in that entry must be granted to the administrator identity that you specify when executing `ldapugmod`.
- With any POSIX-type identity, the user and group ID numbers are used by the HP-UX operating system to determine rights and capabilities in the OS as well as in the file system. For example, a root user ID 0 has unlimited OS administration and file access rights. Before modifying an entry, you must be aware of the selected user and group ID number and any policy that may be associated with that ID.
- Modification (renaming) of a POSIX account does not automatically modify that account's membership in groups, unless the LDAP directory server intrinsically provides that capability. Some LDAP directory servers have a feature known as "referential integrity", which performs modification or removal of DN-type attributes if the specified DN is either changed or removed

- As it may occur in any identity repository, modification of this repository will likely have impacts as defined by the organization security policy. When using `ldapugmod`, you are expected to have full knowledge of the organization security policy and the impact of modifying identity information in that identity repository.

7.3.6.7 Limitations

Because LDAP directories require data be stored according to the UTF-8 (RFC3629) character encoding method, all characters displayed by `ldapugmod` are UTF-8, and assumed to be part of the ISO-10646 character set. The `ldapugmod` tool does not perform conversion of the locale character set to or from the UTF-8 character set.

7.3.6.8 Examples

The following commands set the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables:

```
export LDAP_BINDDN = "cn=Jane Admin,ou=admins,dc=example,dc=com"
export LDAP_BINDCRED = "Jane_Password"
```

Run the following command to go to the `/opt/ldapux/bin` directory where `ldapugmod` resides:

```
cd /opt/ldapux/bin
```

The following commands are used to change the password of the user, `mlee`, using the new user password defined in `LDAP_UGCRED`:

```
export LDAP_UGCRED = "mlee's new Password"
./ldapugmod -t passwd -PW mlee
```

The following command replaces the `uidNumber` value for the user entry, `mMackey`:

```
./ldapugmod -t passwd -u 300 mMackey
```

The following command replaces the `sn` value for the user entry, `mLou`:

```
./ldapugmod -t passwd mLou "sn=Lou"
```

The following command replaces the `gecos` fields for the user entry, `mLou`:

```
./ldapugmod -t passwd -I "Mike Lou,Building-6,222-2222" mLou
```

The following command adds the `description` attribute and value to the user entry, `atam`:

```
./ldapugmod -t passwd -A "description=test user entry" atam
```

The following command extends the existing user entry, `userid=212,ou=users,dc=example,dc=com`, with the POSIX attributes and values for `homeDirectory`, `uid`, and `gidNumber`. The `ldapugmod` tool adds the PosixAccount object class to the entry.

```
./ldapugmod -t passwd -D "userid=212,ou=users,dc=example,dc=com"
-O -A "homeDirectory=/home/testusr" -A "gidNumber=200" -A "uid=testusr"
```

The following command adds the three members, `atam`, `mlou`, `mscott`, to the group entry, `groupA`:

```
./ldapugmod -t group -a atam,mlou,mscott GroupA
```

The following command removes one member, `atam` from the group entry, `groupB`:

```
./ldapugmod -t group -r atam GroupB
```

The following command replaces all instances of the `description` attribute with value "`Group C Entry`" for the group entry, `GroupC`:

```
./ldapugmod -t group GroupC "description=Group C Entry"
```

7.3.7 ldapugdel tool

Use the `ldapugdel` tool to remove POSIX-related user or group entries from an LDAP directory server. If you use `ldapugdel` with the `-O` option, `ldapugdel` removes the POSIX related attributes and object classes from user or group entries, without removing the entire entry itself.

7.3.7.1 Removing attributes only

You can use `ldapugdel` to remove POSIX user and group entire entries from an LDAP directory server. With the `-O` option, `ldapugdel` enables you to remove only POSIX related attributes and object classes from user or group entries, without removing entire entries.

Because mapped attributes are attributes that are often shared with other LDAP-enabled applications, `ldapugdel` does not support attribute mapping. For example, if the `uidNumber` attribute has been mapped to the `employeeNumber` attribute, `ldapugdel` attempts to remove `uidNumber` and not `employeeNumber`.

The `usePassword`, `uid`, `cn`, and `description` attributes are commonly used by most other user and group schemas. With the `-O` option, the `ldapugdel` tool does not attempt to remove these attributes. The `uidNumber`, `gidNUmber`, `loginShell`, `homeDirectory`, `gecos`, and `memberUid` attributes are more unique to the POSIX schema, and are removed when the `-O` option is specified. Use the `ldapugdel -t passwd -O` command, `ldapugdel` removes the `posixAccount` object class and attributes, `uidNumber`, `gidNumber`, `homeDirecotry`, `loginShell`, and `gecos`. Use the `ldapugdel -t group -O` command, `ldapugdel` removes the `posixGroup` object class and the `gidNumber`, `memberUId`, and `userPassword` attributes.

The `ldapugdel` tool also supports a `<protAttr>` list with the `-O` option that enables you to tell `ldapugdel` not to remove specific attributes defined in the `<protoAttr>` list.

Using the `-O -x -t` option forces `ldapugdel` to remove the additional attributes, `cn`, `uid`, or `description`. Removal of these attributes only occurs if allowed by the remaining object classes for that entry.

For detailed information, see the description of the `-O`, `-x` and `-y` arguments that follow.

7.3.7.2 Synopsis

`ldapugdel` ***`[options] [-t <type>] [-h <hostname>] [-p <port>] [-O [<protAttr>[,...]]] <-D <DN>>|<uid_name>|<group_name>`***

7.3.7.3 Options

The `ldapugdel` tool supports the following command options:

`-P` Prompts for the administrator's bind identity (typically LDAP DN or Kerberos principal) and bind password. If you do not specify the `-P` option, `ldapugdel` discovers the bind identity and password from the environment variables `LDAP_BINDDN` and `LDAP_BINDCRED`. If you do not specify the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables, `ldapugdel` uses the bind configuration specified in the LDAP-UX configuration profile. If the LDAP-UX configuration profile specifies the “proxy” bind, `ldapugdel` reads the bind credential from either the `/etc/opt/ldapux/acred` or the `/etc/opt/ldapux/pcred` file. The `/etc/opt/ldapux/acred` file is only used by users who have sufficient administrative privilege to read this file.

`-x` Uses this option only with the `-O` option. Use `-O -x -t passwd` to force the `ldapugdel` tool to remove the common `uid`, `cn`, and `description` attributes from a user entry. Use `-O -x -t group` to force the `ldapugdel` tool to remove the `cn` and `description` attributes from a group entry. Because use of `-x` removes common attributes typically used by other LDAP-enabled applications, HP rarely recommends you to use the `-x` option when removing posixAccount or posixGroup related attributes. If removal of the `uid`, `cn`, or `description` causes an object class violation, `ldapugdel` generates a

warning message. With the `-x` option, LDAP-UX tries to remove as many attributes as allowed by the directory server.

`-y` Uses this option only with the `-O` and `-t passwd` options. This option forces `ldapugdel` to remove the `userPassword` attribute from the user entry. HP does not recommend you to use the `-y` option when removing posixAccount related attributes.

`-Z` Requires an SSL connection to the LDAP directory server, even if the LDAP-UX configuration does not require the use of SSL. Using the `-Z` option requires that either a valid directory server or a CA certificate is defined in the `/etc/opt/ldapux/cert8.db` file. An error occurs if the SSL connection cannot be established.

`-ZZ` Attempts a TLS connection to the directory server, even if the LDAP-UX configuration does not require the use of TLS. If a TLS connection cannot be established, a non-TLS and non-SSL connection will be established. Do not use `-ZZ` unless alternative methods are used to protect against network eavesdropping. Use of `-ZZ` requires that either a valid directory server or a CA certificate is defined in the `/etc/opt/ldapux/cert8.db` file.

`-ZZZ` Requires a TLS connection to the LDAP directory server, even if the LDAP-UX configuration does not require the use of TLS. Using the `-ZZZ` option requires that either a valid directory server or a CA certificate is defined in the `/etc/opt/ldapux/cert8.db` file. An error occurs if the TLS connection cannot be established.

`-S` Displays the Distinguish Name (DN) of the deleted or updated entry when the operation successfully completes.

7.3.7.4 Arguments

The following describes command arguments:

`-h <hostname>` Specifies the host name and optional port number (`hostname:port`) of the LDAP directory server. This option overrides the server list defined by LDAP-UX configuration profile. This field supports specification of IPv4 and IPv6 addresses. If you specify a port for an IPv6 address, you must specify the IPv6 address in a square-bracketed form. If you do not specify the optional port, the port number defaults to 389 or 636 for SSL connection (`-Z`). For example, `-h ldapsrvA:389`.

`-p <port>` Specifies the port number of the LDAP directory server to contact. The `ldapugdel` tool ignores this option if you specify the port number in the `<hostname>` field as part of the `-h` option.

`-t <type>` Specifies the type of entry the `ldapdel` tool needs to delete. The valid types of this argument are `passwd` and `group`. If you do not specify this argument, `ldapugdel` defaults to `passwd`. The `passwd` type represents LDAP user entries containing POSIX account-related information. The `group` type represents LDAP group entries containing POSIX group-related information. For example, `-t passwd`.

`-D <DN>` The `ldapugdel` tool searches for the named user or group using the search rules defined by the service search descriptor in the LDAP-UX configuration profile. You can use the `-D` option to specify the exact distinguished name (DN) of the entry being deleted. You can specify only one of `-D`, `<uid_name>` or `<group_name>` parameter on the command line.

<uid_name>

Specifies the name of the user entry that you want to delete. `ldapugdel` uses the configured LDAP search filter to discover the entry to be removed, such as `(&(objectclass=posixAccount)(uid=name))`. If more than one entry matches this search filter, only the first discovered entry is removed. You can specify only one of `-D`, `<uid_name>` or `<group_name>` parameter on the command line.

<group_name>

Specifies the name of the group entry that you want to delete. The `ldapugdel` tool uses the configured LDAP search filter to discover the entry to be removed, such as `(&(objectclass=posixGroup)(cn=name))`. If more than one entry matches this search filter, `ldapugdel` removes only the first discovered entry. You can specify only one of `-D`, `<uid_name>` or `<group_name>` parameter on the command line.

-O [<protAttr>[,...]]

Enables the `ldapugdel` tool to delete only the posixAccount or posixGroup object class and associated attributes, without deleting the entire user or group entry. With the `-t passwd` option, the `ldapugdel` tool removes the posixAccount object class and the following attributes:

- `uidNumber`
- `gidNumber`
- `homeDirectory`
- `loginShell`
- `gecos`

With the `-t group` option, the `ldapugdel` tool removes the posixGroup object class and the following attributes:

- `gidNumber`
- `memberUid`
- `userPassword`

The `<protAttr>` list consists of one or more of the previous attribute names separated by commas with no white space. If you specify the `<portAttr>` list, `ldapugdel` will not remove the specified attributes.

NOTE: Keep the following considerations in mind when using the `-O` option:

- The `ldapugdel` tool does not support attribute mappings. For example, if the `uidNumber` attribute has been mapped to the `employeeNumber` attribute, `ldapugdel` will attempt to remove `uidNumber` attribute and not `employeeNumber`.
- Because the `uid`, `cn` and `description` attributes are commonly used by other user or group object classes, `ldapugdel` does not attempt to remove `uid` and `description` attributes for a user entry or `cn` and `description` attributes for a group entry, unless failure to remove those attributes can cause an object class violation (because the remaining object classes for that entry do not define them as their attributes). Use of the `-O -x` option forces `ldapugdel` to remove those attributes if allowed by the remaining object classes for that entry.
- Because the `userPassword` attribute is often used by other user-related object classes, `ldapugdel` does not attempt to remove the `userPassword` attribute when removing user entries. Use of the `-O -y -t passwd` options forces `ldapugdel` to remove this attribute if allowed by the remaining object classes in that entry.
- The `ldapugdel` tool attempts to remove the `posixAccount` and `posixGroup` object classes only if they are present. In some cases, when a user or group entry is built using an abstract class, the `posixAccount` and `posixGroup` object classes might not be present in the entry.
- If `ldapugdel` determines that the entry being deleted is stored on a Windows ADS directory server, `ldapugdel` does not remove the `homeDirectory` attribute. The Window user entry contains the `User` object class and the `homeDirecotroy` attribute is part of the `User` object class.
- The Microsoft Services for UNIX (SFU) schema does not use RFC 2307 standard attributes. Also `ldapugdel` does not support attribute mapping as defined by the LDAP-UX configuration profile when the tool is used to access a Windows ADS with msSFU 2.0 or msSFU3.0/3.5 schema installed. When the `-O` option is specified and `ldapugdel` determines that it is connected to a Windows ADS with these schema installed, `ldapugdel` does not remove the mapped POSIX object class and attributes (msSFU30xxx or msSFU20xxx) for the specified user or group entry.
- The `-O` option functions properly with Windows Server 2003 R2/2008 ADS, because it uses standard RFC 2307 attributes, with the exception of the `homeDirectory` attribute.

7.3.7.5 Specific return codes for ldapugdel

The `ldapugdel` tool returns a list of return codes shown in Table 7-8 (page 265).

Table 7-8 Return codes for ldapugdel

Return Codes	Message
`DEL_COMMANDLINE_ERR`	Invalid POSIX attributes.
`DEL_MULTIPLE_ENTRY_FOUND`	Multiple entries found that match the same name. Please use a DN to specify a specific entry.
`DEL_DELETE_FAILED`	The LDAP deletion operation failed.
`DEL_SEARCH_FAILED`	The LDAP search for subSchemaSubEntry, attributeTypes or objectClasses failed.
`DEL_PARSE_ERROR`	Unable to analyze LDAP directory server's schema. This operation is required in order to determine which attributes may be legally removed.

7.3.7.6 Security considerations

Be aware of the following security considerations when you use `ldapugdel`:

- Use of `ldapugdel` requires permissions of an LDAP administrator when it performs its operations on the directory server. The rights to delete or modify existing LDAP directory entries under the requested subtree and to remove the required attributes in that entry must be granted to the administrator identity that you specify when you execute `ldapugdel`.
- Removal of a POSIX account does not automatically remove that account's membership in groups, unless the LDAP directory server provides that capability. Some LDAP directory servers have a feature called "referential integrity", which performs modification or removal of DN-type attributes if the specified DN is either changed or removed.
- As it may occur in any identity repository, modifying the repository can likely have impacts as defined by the organization security policy. When using `ldapugdel`, you are expected to have full knowledge of the organization security policy and the impact of deleting identity information from that identity repository.
- Do not use `ldapugdel` as part of a modification process on a user or group entry (deleting and re-adding the entry as a way to modify that entry.) User and group entries in an LDAP directory often contain information about the user or group that is outside the POSIX information model. Deleting an entry will delete all information about the user or group. When the entry is re-added, recovery of the non-POSIX information may not be possible.

7.3.7.7 Limitations

Because LDAP directory servers require data to be stored according to the UTF-8 (RFC3629) character encoding method, all characters provided to `ldapugdel` are assumed to be UTF-8 and part of the ISO-10646 character set. The `ldapugdel` tool does not perform conversion of the locale character set to or from the UTF-8 character set.

7.3.7.8 Examples

The section provides examples of using `ldapugdel`:

Run the following commands to specify the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables:

```
export LDAP_BINDDN = "cn=Jane Admin,ou=admins,dc=exampe,dc=com"
export LDAP_BINDCRED = "Jane's password"
```

Run the following command to go to the `/opt/ldapux/bin` directory where `ldapugdel` resides:

```
cd /opt/ldapux/bin
```

Run the following command to delete the entire user account entry, `astein`, on the LDAP directory server, `ldapsrvA`. The `-h` option overrides the server list defined by the LDAP-UX configuration profile.

```
./ldapugdel -t passwd -h ldapsrvA:389 astein
```

Run the following command to delete the entire user account entry, `msmart`:

```
./ldapugdel -t passwd msmart
```

Run the following command to delete the entire group entry with the distinguished name, `"cn=group1,ou=groups,dc=example,dc=com"`:

```
./ldapugdel -t group -D "cn=group1,ou=groups,dc=example,dc=com"
```

Run the following command to delete only the posixAccount object class and associated attributes, `uidnumber`, `gidNumber`, `homeDirectory`, `loginShell` and `gecos`, without delete the entire user entry, `msmith`:

```
./ldapugdel -t passwd -O msmith
```

Run the following command to delete only the posixAccount object class and associated attributes, `uidnumber`, `gidNumber` and `gecos`, without delete the entire user entry, `mlee`:

```
./ldapugdel -t passwd -O "homeDirectory,loginShell" mlee
```

Run the following command to delete only the posixGroup object class and associated attributes, `gidNumber`, `memberUid` and `userPassword`, without delete the entire group entry, `groupA`:

```
./ldapugdel -t group -O groupA
```

The following command forces `ldapugdel` to remove the common `uid`, `cn` and `description` attributes from the user entry, `jswartz`:

```
./ldapugdel -t passwd -O -x jswartz
```

Run the following commands to unset the `LDAP_BINDDN` and `LDAP_BINDCRED` environment variables:

```
unset LDAP_BINDDN
unset LDAP_BINDCRED
```

7.3.8 ldaphostmgr tool

Use the `ldaphostmgr` tool to add, modify, or delete information about hosts (OS instances) that are part of the organization. The `ldaphostmgr` tool:

- Uses the existing ldapux(5) configuration, requiring only a minimal number of command-line options to discover where to search for host information, such as what directory server(s) to contact and proper search filters for finding hosts.
- Uses the existing ldapux(5) authentication configuration to determine how to bind to the LDAP directory server.
- Supports attribute-mapping for attributes defined by the `ipHost` objectclass. Additional attributes used in a host entry (such as `owner`, `entityRole`, and so on) are not mapped.
- Can be used to centrally manage ssh public keys for hosts.

.

7.3.8.1 Synopsis

```
ldaphostmgr [-a | -m | -d]
[-F] [-I] [-X] [-Z] [-ZZ] [-ZZZ] [-P] [-C] [-f] [-S] [-V]
[-h servername] [-p port] [-B relbase] [-x domain]
[-O owner[...]] [[-G group] [...]]
-k [[!|?|^]keytype] [-e days_to_expire]]
[-i [ipAddr]] [...]] [[-r role][...]]
[[-A attrval] [...]] [[-R attrval] [...]]
[-c comment] [-E envfile] {(-D DN) | host_name}
[[attr=value][...]]
```

7.3.8.2 Options and Arguments

The `ldaphostmgr` tool supports the following options and arguments:

`-a`

Adds a new host to the directory server. The host is added to the base specified by the host service search descriptor in the LDAP-UX configuration profile entry (unless the `-D` option is used to specify the fully qualified DN). When an entry is created, the device and ipHost object classes are used. Optionally, additional object classes can be used to describe the host entry. See Section 7.3.8.3 (page 274) for more information.

On ADS, the `Computer` object class is used.

The `-a` (add), `-d` (delete), and `-m` (modify) options are mutually exclusive. The `-m` option is the default if none of these three options is specified.

`-m`

Modifies an existing host entry. The `-a` (add), `-d` (delete), and `-m` (modify) options are mutually exclusive. The `-m` option is the default if none of these three options is specified.

`-d`

Deletes a host entry. This removes the specified host entry from the directory server, and removes the host as a member from any group that contains this host as a member. The `-a` (add), `-d` (delete), and `-m` (modify) options are mutually exclusive. The `-m` option is the default if none of these three options is specified.

`-F`

Forces creation of a new host entry even if the following error conditions occur:

- Setting the owner of a host to an owner that does not exist. If the `-O` option specified a DN, then that DN is used as the owner. If a user or group is specified, then the owner is set to the DN of the user identity used by `ldaphostmgr` when performing the command. In this case, if the current user identity is already marked as an owner of the host, a MODIFY_FAILED error is returned.
- Creating or changing the key for the specified remote host, even if the identity of the remote host could not be verified. This usage is not recommended since the key loaded into the directory server cannot be trusted.

The `-F` option does not override any enforcement that occurs on the directory server itself, such as adding an attribute without also adding a corresponding required object class or modifying an attribute for which the user does not have sufficient directory privilege.

`-I`

Adds/modifies additional information about the host:

```
entityVersion=$(/usr/bin/uname -sr)
entityModel=$(/usr/bin/model)
```

On ADS, instead of `entityVersion`, the `operatingSystem` and `operatingSystemVersion` attributes are used. `entityModel` is not defined in an ADS environment.

Note that if an `-I` option is specified and the host being managed is remote, a remote login to that host is required and performed by `ldaphostmgr` to discover that information. This means that when the LDAP credentials are specified (through the prompt or LDAP_BINDDN), those credentials must also represent a POSIX account, such that a remote login to that host can be performed

using that identity. Specifying `-I` on a remote host will fail if LDAP-UX (version > B.05.00) is not installed on that host.

-X Does not prompt for information, including the host's password or other interactive confirmation prompts. If required information cannot be discovered, the command exits with an error. The `-F` option can be used to force an override for most confirmation prompts.

-Z Requires an SSL connection to the directory server, even if the ldapux(5) configuration does not require the use of SSL. Use of `-Z` requires that either a valid server or CA certificate be defined in the `/etc/opt/ldapux/cert8.db` file. An error occurs if the SSL connection could not be established. See Section 7.3.8.4 (page 274) for additional details.

-ZZ Attempts a TLS connection to the directory server, even if the ldapux(5) configuration does not require the use of TLS. If a TLS connection cannot be established, a non-TLS and non-SSL connection is established. Using `-ZZ` is not recommended unless alternative methods are used to protect against network eavesdropping. Use of `-ZZ` requires that either a valid server or CA certificate be defined in the `/etc/opt/ldapux/cert8.db` file. See Section 7.3.8.4 (page 274) for additional details.

-ZZZ Requires a TLS connection to the directory server, even if the ldapux(5) configuration does not require the use of TLS. Use of `-ZZZ` requires that either a valid server or CA certificate be defined in the `/etc/opt/ldapux/cert8.db` file. An error occurs if the TLS connection could not be established. See Section 7.3.8.4 (page 274) for additional details.

-P Specifies that the host should be assigned a password. This is typically used when the host acts as a proxy user for an LDAP-UX connection to the directory server. In this case, the LDAP administrator should grant the host the privilege to read LDAP RFC 2307 schema attributes in the directory server. This option prompts for the host password, unless the password has been specified in the LDAP_HOSTCRED environment variable. If the `-X` option is specified, the host password must be specified in the LDAP_HOSTCRED environment variable, or an error is returned.

-C If the directory server authentication credentials have not been specified in the LDAP_BINDDN and LDAP_BINDCRED environment variables, then the `-C` option tells `ldaphostmgr` to use the credentials specified in the `/etc/opt/ldapux/acred` file. If that file does not exist, or the user running `ldaphostmgr` does not have sufficient privilege to read that file, then `ldaphostmgr` prompts for directory server authentication credentials, unless the `-X` option was specified. Without the `-C` option, the `acred` file is not used.

-f If the *`host_name`* specified is a short name (without the fully qualified DNS domain), the `-f` option adds/modifies the fully qualified host name to the host entry. Example: `cn=host.domain.org`. Both the short and full name are added to the `cn` (or mapped) attribute. The `-f` option applies to both the `-a` and `-m` operations. If *`host_name`* is already fully qualified (contains a domain), then the `-f` option has no effect. Only a

host_name is added to the entry. `ldaphostmgr` uses the `/etc/resolv.conf` file to determine the domain.

If the `-D` option is specified, the value of the RDN (relative distinguished name) is used to determine the *host_name*.

-S

Displays the DN of the created, modified, or deleted host entry, at the end of the output.

-v

Displays additional information used to analyze and troubleshoot usage issues.

-h *servername*

Specifies the host name and optional port number (*servername*:*port*) of the directory server where this entry should be added. This option overrides the server list configured by ldapux(5). The *servername* field also supports specification of IPv4 and IPv6 addresses. If you specify a port for an IPv6 address, the IPv6 address must be specified in square-bracketed form. If the optional port is unspecified, the port number is assumed to be 389 or 636 for SSL connections (`-Z`). See Section 7.3.8.4 (page 274) for additional details.

-G *group*

Specifies a group to which this host should be added or removed as a member. The group entry must already exist and the object class must be either a `groupOfNames` or `groupOfUniqueNames`. Specify the group as:

`[!]`*short_name*|*DN*

Where *short_name* is the name of the group as found in the `cn` attribute of the group. If the *short_name* is used, the search base specified in the LDAP-UX configuration profile for the `group` service is used to determine where to find the groups. However, the search filter from the profile is not used, instead forcing the groups found to be of type `groupOfNames` or `groupOfUniqueNames`. If more than one group is found with the same name, an error is returned.

If the `!` option is specified, the host is removed as a member from the specified group. If the `!` is specified by itself, the host is removed from all groups of which it is a member.

The `-G` option can be specified more than once.

-O *owner*

Specifies the owner of the host. If the HP-UX Directory Server was installed using the LDAP-UX guided installation, access control instructions are created such that the owner of the host is granted administrative rights to manage data about the host, as well as change the ssh keys for the host. The *owner* can be specified as either an individual or a group:

```
[!]DN
[!]user: user_name
[!]group:group_name
```

Where *user_name* is a UNIX account name and *group_name* is a UNIX group name, that is maintained in the LDAP directory server. If the optional `!` (ASCII 33) character is specified, the resulting user or group is removed as an owner of the host. If `!` is specified by itself, all values of the `owner` attribute are removed from this entry. Removing all `owner` attributes from an entry is not recommended because the `owner` attribute may be used to grant access control rights for the defined administrators.

To replace an owner of the host, you can specify the `-O` option twice to remove the existing user and add a new one. For example:

```
ldaphostmgr -O !user:olduser -O user:newuser hostname
```

If the user is adding a new host entry (`-a` option) and if the `-O` option is not specified, the `owner` attribute is assigned the DN of the current user (as authenticated by `ldaphostmgr`). Refer to Security Considerations for additional information.

On ADS, the owner information is stored in the `managedBy` attribute. Because the `managedBy` attribute is single-valued on ADS, only one owner may be assigned to the host.

If DN is specified, `ldaphostmgr` checks to see if the DN exists in LDAP server. If it does not exist, `ldaphostmgr` prompts to see if the DN should be added anyway (unless the `-X` option is specified, in which case an error is returned). If the `-F` option is specified, `ldaphostmgr` sets the `owner` attribute to the specified DN, even if that DN does not exist in the directory server.

`-c` *`comment`*

Specifies the comment/description to be associated with the host entry. The comment text is added as a value in the `description` attribute. If the `description` attribute exists, then all values are replaced with the specified comment. If the `!` option is specified, the `description` attribute is removed entirely.

`-k [!|?|^]`*`keytype`*

Adds, changes, removes, or validates ssh key(s) for the host. The *`keytype`* is either a key-string as defined in the `-t` option of the *ssh-keygen* manpage (currently defined as `rsa1`, `rsa`, and `dsa`), the key-string `all`, or a file path name that references a file that contains keys for the host. The key-file format is the same as a host-key file (such as found in `/etc/opt/ssh/ssh*.pub`), except that more than one key can be specified, on separate lines. If a key-file is specified, the key(s) found in the key-file are simply added/modified in the host entry, without validation of the actual keys used on the host. The `!`, `?`, and `^` controls do not apply when using a key-file.

When adding or modifying keys (neither the `!` nor `?` controls are specified) and keytype is one of the specified keystrings (not a key-file path), then for the specified key type (or all key types), the following action is performed:

- If the key of that type exists on the host, but does not yet exist in the directory server entry for this host, then that key is added to the directory server entry for the host.
- If the key of that type does not exist on the host, a new key on the host is created, and that key is added to the directory server entry for this host. If the host entry already contains a key of the same type, that key is replaced in the entry with the newly created key.
- If the key of that type exists on both the host and in the host's directory server entry, then `ldaphostmgr` changes the current key of that type on the host and then replaces that key in the host's directory server entry. `ldaphostmgr` will prompt for confirmation before changing an existing key on the host, unless the `-X` option is specified (in which case, the key is not changed unless `-F` is also specified.)

If you specify the `!` option, the specified key(s) is(are) removed from the host entry in the directory server. The actual keys on the host are not removed.

If you specify the `?` option, the key(s) on the host are validated against those found in the representative directory entry for the specified host. This option is usually used on the local host, so that the owner can verify that host key integrity as represented by the directory server. Note that often the `?` character can be interpreted by the shell (man shells(4)), and therefore should be escaped or enclosed in quotation marks.

When adding or modifying keys for a remote host, `ldaphostmgr` attempts to connect to that remote host using ssh itself. However, ssh itself may not be able to trust the identity of the remote host if a local copy of the remote host's key is not available in a local known_hosts file or in the LDAP directory server. If the identity of the remote host cannot be positively identified, `ldaphostmgr` issues a WARNING and prompts for confirmation that the remote key should be trusted. If the user chooses to trust the unidentified host, `ssh-keyscan` is used to discover the remote public keys and add/replace them in the directory server entry. Because untrusted discovery is subject to man-in-the-middle or spoofing attacks, this method for key discovery is not recommended unless the key fingerprint can be validated.

Specifying the `^` option disables remote key management, and indicates to `ldaphostmgr` that the remote host cannot be directly managed by the solution. Instead, the result from a direct `ssh-keyscan` should be used to discover the remote host's public keys. For example, an appliance that supports ssh, but does not have HP-UX on it, cannot respond properly to remote management commands. Again, `ldaphostmgr` issues a WARNING and prompts for confirmation that the remote key should be trusted. Because untrusted discovery is subject to unauthorized attacks, this method for key discovery is not recommended unless the key fingerprint can be validated.

NOTE: If the `^` flag is specified and the target is the local host, `ldaphostmgr` simply takes the current public key(s) and uploads them to the directory server. Since the keys on the local host are considered trusted, a WARNING prompt is not displayed.

If the `-X` option is specified, `ldaphostmgr` does not prompt, and fails without adding the keys to the directory entry, unless the `-F` option is also specified. Use of `^`, `-X`, and `-F`, or answering "yes" to the "Untrusted Discovery:" prompt is not recommended as the primary method for discovery of host keys unless an external and validated transport method can be used to validate the integrity of the updated keys. For example, if the user can create a trusted session to the host (such as connecting to the physical console), the `ldaphostmgr -k ?` command can be used to validate that the keys of the host found in the directory server match that specified in the `/etc/opt/ssh/*.pub` files.

Note that if a `-k` option is specified and the host being managed is remote, a remote login to that host is required and performed

by `ldaphostmgr` to modify the remote keys. This means that when the LDAP credentials are specified (through the prompt or LDAP_BINDDN), they must also represent a POSIX account, such that a remote login to that host can be performed by `ldaphostmgr` using that identity.

The `-k` option is not supported with ADS.

`-e` *`days-to-expire`*

To keep track of when keys were originally generated, `ldaphostmgr` adds a unique *`management-string`* to the comment field of the public key. The *`management-string`* begins with BEGIN-KM and ends with END-KM. This field is an extensible attribute/value array, which contains at least the `creationtime` attribute, which identifies when the key was created. In addition, when the `-e` option is specified, the `expirationtime` attribute can also be added. Discovery of hosts with expired keys can be performed with the `-k` option of the `ldaphostlist` command. Combined use of `ldaphostlist` and `ldaphostmgr` can be used to keep expired keys up-to-date. See the -k option for additional information

`-i` *`ipaddr`*

Adds the specified IP Address to the host entry, in the `ipHostNumber` attribute (or `mapped` attribute). The *`ipaddr`* can be either an IPv4 or IPv6 style address. IPv6 style addresses are normalized to match the format recommended by the RFC2307-bis IETF draft. If `!` is specified at the beginning of the *`ipaddr`*, the specified IP address is removed instead. If `!` is specified, but no IP address is specified, then all values specified in the `ipHostNumber` attribute are removed and replaced with the value `0.0.0.0`. Because the `ipHost` object class is critical for distinguishing host entries in an LDAP directory server, by default `ldaphostmgr` adds the `ipHost` objectclass and the `ipHostNumber` attribute, using the discovered IP Address for the host.

NOTE: If `!` is specified to remove a specific IP address, and you remove the last IP address associated with the host, `ldaphostmgr` also removes the `ipHost` objectclass. This could prevent the host from appearing in LDAP-UX (depending on the hosts service descriptor search filter in the LDAP-UX profile.) If you want to maintain the object classification of the `ipHost`, use `!` by itself, to replace it with a `0.0.0.0`.

`-r` *`role`*

Specifies an organizational role for this host. Role is a free-format key-string that will be assigned to the `entityRole` attribute. The value specified in role replaces all values for the `entityRole` attribute. The `-r` option can be specified more than once if more than one role applies to the host. Organizations should consider standardizing role key-strings, such that they can be used in LDAP search filters to discover and manage classes of systems.

If `!` is specified at the beginning of the role, the specified role is removed instead. If `!` is specified, but no role is specified, then all values specified in the `entityRole` attribute are removed. Note: On ADS, this attribute does not exist by default and would require modifying the ADS schema to add this attribute type. Refer to the

ldapschema(1M) manpage and the `/etc/opt/ldapux/schema/ldapux50.xml` file provided.

-x *domain* Short, conventional, name of the domain. This option pecifie the value for the `entityDomain` attribute. Only one domain can be specified. If `!` alone is specified, or is specified at the beginning of the domain, the domain is removed. On ADS, this value is not used because the location of the host is implied by its location in the directory tree. Instead, refer to the `-D` or `-C` options to control the domain of a host managed in ADS.

-A *attrval* Specifies an attribute and value to be added to an entry. The format of *`attrval`* is *`attribute=value`*, where *`attribute`* is the name of the attribute to add, and *`value`* is the specific instance of that attribute. The `-A` option is used when working with multi-valued attributes, to add a new value for a multi-valued attribute without removing already existing values for that attribute. Note that use of the –A option interacts with the optional *`attr=value`* parameters. The `-A` option can be specified more than once per command line. The *`value`* portion of the *`attrval`* can be an empty string.

If you add a new attribute type to an entry, you also need to add the associated object class if that object class is not already part of the host entry.

-R *attrval* Specifies an attribute or specific values of an attribute to be removed from the entry. The format of *`attrval`* is *`attribute`*`[=`*`value`*`]`, where *`attribute`* is the name of the attribute to remove, and *`value`* is the specific instance of that attribute if the attribute is multi-valued. Note that use of the `-R` option interacts with the optional *`attr=value`* parameters. The `-R` option can be specified more than once per command line.

-B *relbase* Specifies where the host entry should be found/created. However, instead if specifying a full base DN, *`relbase`* is relative to the default search base configured in LDAP-UX (refer to the `-b` option of the *ldapcfinfo*(1M) command.) For example, if the default search base is `dc=example,dc=org` and `-B ou=mycomputers` is specified, then the default parent for the host will be `ou=mycomputers,dc=example,dc=org`. If the `-a` option is specified, then the new host entry is created under the resulting DN. If the `-m` option is specified, then only the resulting DN is used for the search base when discovering hosts. And if the `-d` option is specified, then `ldaphostmgr` searches for hosts to delete only under the resulting DN. The resulting parent DN is assumed to already exist in the directory server. It will not be created, even when the `-a` or `-F` options are specified.

-E *envfile* Read environment variables from a file using the following syntax for the file format:

`ENV_NAME``=`*`value`*

where *`value`* can be quoted. The quote (ASCII 34) character and the escape character (backslash ASCII 92) must always be escaped if they are part of the value itself. For example:

```
LDAP_HOSTCRED="Rfxw-\"92"
```

In this example, the password value will be: `Rfxw-"92`

-D *DN*, or *host_name*

Specifies the host DN or POSIX host name for which to apply the operation. Specifying either `-D` *DN*, or *host_name* is required, even if the intent is to manage data for the local host. Specify the host's true full or short name when using *host_name*. Do not specify `localhost` when attempting to modify the local host.

If `host_name` is specified, it is positional-dependent on the `ldaphostmgr` command line and should be placed after all the command options.

If *host_name* is specified, `ldaphostmgr` constructs the DN of the entry using the host search base as the parent DN. If the search base for the host's service as defined in the profile is the same as the default search base, then by default `ldaphostmgr` adds a host container to the default search base. For example, if the default search base is `dc=myorg,dc=org`, then `ldaphostmgr` builds the DN by adding both the `ou=hosts` container (or `cn=computers` for ADS) and the host name to the DN, resulting in `cn=hostname,ou=hosts,dc=myorg,dc=org`. If `-D` *DN* is specified, then the host name is extracted from the value defined in the RDN component of the specified *DN*.

attr=value

Allows modification of arbitrary LDAP attributes and values. *value* can be an empty string. However, this usage does not remove attributes and their values from the directory server. Instead, use the `-R` option to remove arbitrary attributes:

See Section 7.3.8.6 (page 275) for the impact of using this option.

7.3.8.3 Object Classes

By default, `ldaphostmgr` uses the device and ipHost object class when creating new entries (or the computer object class for ADS). Using certain options will cause additional attributes and their corresponding object classes to be added to host entries that are being created or modified. These include the following object classes:

- ldapPublicKey—used when the `-k` option is specified.
- domainEntity—used when the `-r` or `-P` option is specified.

The ldapPublicKey and domainEntity object classes are not added to entries stored in ADS.

7.3.8.4 Binding to the Directory Server

The `ldaphostmgr` is designed to take advantage of the existing ldapux(5) configuration for determining to which directory server to bind and how to perform the bind operation. `ldaphostmgr` consults the ldapux(5) configuration profile for the following information:

- The list of LDAP directory server hosts
- The authentication method (simple passwords, SASL/DIGEST-MD5, and so on)

If neither the LDAP_BINDDN or LDAP_BINDCRED environment variable is specified, `ldaphostmgr` also consults the ldapux(5) configuration for additional information:

- The type of credential (user, proxy, or anonymous) to use
- The credential used for binding as a proxy user (either `/etc/opt/ldapux/acred` for administrative users, or `/etc/opt/ldapux/pcred` for nonprivileged users)

As with ldapux(5), `ldaphostmgr` attempts to contact the first available directory server as defined in the ldapux(5) host list. As soon as a connection is established, further directory servers on the host list are not contacted. Once connected, `ldaphostmgr` first determines if the environment variables LDAP_BINDDN or LDAP_BINDCRED were specified. If both are specified, then `ldaphostmgr` attempts to bind to the directory server using the specified credentials and

configured LDAP-UX authentication method. If the neither of the above mentioned environment variables were specified, then `ldaphostmgr` determines if the configured credential type is "proxy" and, if so, attempts to bind to the directory server using the configured LDAP-UX proxy credential. If configured, the `acred` proxy credential is used for administrative users (determined if the user running `ldaphostmgr` has enough privilege to read the `/etc/opt/ldapux/acred` file). An additional requirement when managing a remote host, is that the specified credential must also have POSIX account attributes specified in his/her directory server entry. This means that if the `acred` credentials are used, they too must represent a POSIX account.

NOTE: To prevent discovery of the LDAP administrator's credentials, the LDAP user DN and password cannot be specified as command-line options to the `ldaphostmgr` utility.

7.3.8.5 Security Considerations

- Use of `ldaphostmgr` requires permissions of an LDAP administrator when it performs its operations on the directory server. The rights to create new LDAP directory entries under the requested subtree, along with creation of the required attributes in that entry must be granted to the LDAP administrator identity that is specified when executing `ldaphostmgr`.
- When creating, changing, or validating the host keys of a remote host, `ldaphostmgr` attempts to create a session on the remote host using the identity of the user running the `ldaphostmgr` commend. This means the specified LDAP identity must have an associated posixAccount object class. The session to the remote host is established using ssh itself. If the ssh public key for the remote host is not defined in the directory server or in a local `known_hosts` file, the user is prompted before creating a connection to the remote host (since in this condition, it is possible the remote host is an impostor). Such connections should not be allowed unless the key fingerprint can be validated.
- If the current user has sufficient privilege to modify the `sshPublicKey` attribute in a representative host entry in the directory server, `ldaphostmgr` allows the current user to modify the public and private key pairs for the host (local or remote). `ldaphostmgr` runs as a `setuid` program and temporarily elevates its privilege in this situation.
- As would occur in any identity repository, modification of this repository will likely have impacts as defined by the organization's security policy. Users of `ldaphostmgr` are expected to have full knowledge of the impact to the organization's security policy when adding, removing, or modifying host information to that repository.
- To support non-interactive use of the `ldaphostmgr` command, specification of the LDAP user's credentials is required through use of the LDAP_BINDDN and LDAP_BINDCRED environment variables. To prevent exposure of these environment variables, they should be unset after use. Note that the shells(4) command history log may contain copies of the executed commands that show the setting of these variables. Access to a shell's history file must be protected. As an alternative, the environment variables used by `ldaphostmgr` can be specified in a file, using the `-E` option. Specification of the LDAP administrator's credentials on the command line is not allowed, since information about the currently running processes can be exposed externally from the session. Allowing interactive prompting for these credentials (not specifying `-X`) eliminates the need to set the LDAP_BINDDN and LDAP_BINDCRED environment variables.

7.3.8.6 Usage Notes

Under common usage, `ldaphostmgr` uses the LDAP replace operation when changing values of an attribute in an entry. This feature can impact attributes that have multiple values, by removing all occurrences of an attribute value and replacing it with the one specified on the `ldaphostmgr` command line. For example, if the `-c` argument is used to specify a new description for a host, all occurances of the `description` attribute are replaced by the value specified for the `-c` argument. This mode of operation applies to the `-I` command argument as well.

When the *attr=value* parameter is used to modify an existing attribute, the `ldaphostmgr` command also uses the LDAP replace operation. The replace operation will remove all occurrences of the specified attribute for an entry and replace it with the value specified. If there are multiple values for a single attribute in an entry, the use of a single *attr=value* parameter will replace all values with the single value specified on the command line. Note that it is possible to specify more than one occurrence of the same attribute on the command line, if that attribute is multi-valued, in which case, both values will be created in the entry.

Use of `-A` or `-R` changes this behavior (for both the above-listed command arguments and the *attr=value* parameters). Any attribute specified as an argument to the `-A` or `-R` option will cause `ldaphostmgr` to perform an LDAP add operation instead of an LDAP replace operation. Example: Suppose an entry in an LDAP directory appears as follows:

```
dn: cn=chef,ou=Hosts,dc=cup,dc=hp,dc=com
cn: chef
ipHostNumber: 0.0.0.0
objectClass: top
objectClass: device
objectClass: iphost
objectClass: domainEntity
owner: uid=domadmin,ou=People,dc=cup,dc=hp,dc=com
entityRole: WebServer
entityRole: DBServer
```

Performing the following `ldaphostmgr` command:

```
ldaphostmgr chef "entityRole=NFSServer"
```

Replaces all instances of cn:

```
dn: cn=chef,ou=Hosts,dc=cup,dc=hp,dc=com
cn: chef
ipHostNumber: 1.2.3.4
objectClass: top
objectClass: device
objectClass: iphost
objectClass: domainEntity
owner: uid=domadmin,ou=People,dc=cup,dc=hp,dc=com
entityRole: NFSServer
```

As a general rule, be cautious before using `ldaphostmgr` to change multi-valued attributes.

Also note that `ldaphostmgr` does not allow use of the same attribute and value pair more than once, either specified as part of *attr=value*, `-R` or `-A`, or from other command-line options (for example `-i` for *ipAddress* where *ipAddress* is mapped to some other attribute). `ldaphostmgr` will exit with error status before sending any conflicting modification request to the directory server.

7.3.8.7 Errors and Warnings

Upon exit, `ldaphostmgr` returns a 0 (zero) exit status if no errors or warnings were encountered. If `ldaphostmgr` encounters an error or warning; a nonzero exit status is returned, and one or more messages are logged to `stderr`. Messages have the following format:

```
ERROR: code:
          message
or
WARNING: code:
          message
```

Leading extra white space might be inserted to improve readability and follow 80-column screen formatting. *code* is a programmatically parsable error key-string, while *message* is human-readable. Refer to the *LDAP-UX Client Services Administrator's Guide* for a list of possible error codes generated by the LDAP user and group management tools.

7.3.8.8 External Influences

7.3.8.8.1 Environment Variables

The `ladpahostmgr` tool supports the following environment variables:

`LDAP_HOSTCRED` When used in combination with the `-PW` option, LDAP_HOSTCRED specifies the proxy password of the newly created host. Also, if the ldapux(5) attributed mapping for the `userPassword` attribute has not been defined or set to "*NULL*", `ldaphostmgr` creates new passwords in the `userPassword` attribute.

`LDAP_BINDDN` Specifies the DN of a user with sufficient directory server privilege to create new users and/or groups in the LDAP directory server. While this variable is optional, if LDAP_BINDDN is specified, LDAP_BINDCRED must also be specified. Furthermore, if `ldaphostmgr` is used to manage information about a remote host, and the `-k` or `-I` option is specified, the specified LDAP_BINDDN must also represent a POSIX account, such that `ldaphostmgr` can remotely connect to that host to discover/modify that information on the remote host. When doing so, the POSIX ID of the specified user is used to remotely log in to the host.

`LDAP_BINDCRED` Specifies a password or other type of credential used for the user specified by the LDAP_BINDDN. While this variable is optional, if LDAP_BINDCRED is specified, LDAP_BINDDN must also be specified.

7.3.8.8.2 LDAP-UX Profile

`ldaphostmgr` makes use of the LDAP-UX configuration profile to determine the information model used in the directory server to store POSIX attributes. Refer to the *LDAP-UX Client Services Administrator's Guide* for additional information about the configuration profile.

7.3.8.9 Limitations

- Since LDAP directories require data be stored according to the UTF-8 (RFC3629) character encoding method, all characters passed into `ldaphostmgr` are assumed to be UTF-8, and part of the ISO-10646 character set. `ldaphostmgr` does not perform conversion of the locale character set to/from the UTF-8 character set.

7.3.8.10 Examples

Examples of how to use `ldaphostmgr` can be found in the *LDAP-UX Client Services Administrators Guide*.

7.3.8.11 See Also

ldaphostlist(1m), ldapugadd(1m), ldapugmod(1m), ldapugdel(1m), ldapcfinfo(1m) and ldapux(5)

7.3.9 ldaphostlist tool

Use the `ldaphostlist` tool to display and enumerate host entries that reside in an LDAP-based directory server. Although `ldaphostlist` provides output similar to the `ldapsearch` command, it satisfies a few specific feature requirements. These features allow applications to discover and evaluate hosts stored in an LDAP directory server without requiring intimate knowledge of the methods used to retrieve and evaluate that information in the LDAP directory server. In addition, `ldaphostlist` can be used to discover expiration information about ssh host keys, if that information is managed in the directory server. Except for the optional trailing *`attr`* list, the tool's parameters are not positional-dependent. Unless the trailing *`attr`* list is provided, `ldaphostlist` only displays the `cn` (host name) and `ipHostNumber` (IP Address) attributes.

The `ldaphostlist` tool provides the following functions:

- Uses the existing ldapux(5) configuration, requiring only a minimal number of command-line options to discover where to search for host information, such as what directory server(s) to contact and proper search filters for finding accounts and groups. Provides command options to let you change these configuration parameters.
- Uses the existing ldapux(5) authentication configuration to determine how to bind to the LDAP directory server.
- Supports attribute mapping as configured by ldapux(5). Fields returned from `ldaphostlist` use a consistent format, similar to that defined by RFC2307, even when different attributes are actually used to store the information in the directory server. Note that although this format is similar to LDIF, it is not LDIF. Major differences include:
  - Object classes are not displayed (unless specifically requested in the *attr* list)
  - By default `ldaphostlist` displays only POSIX-related attributes for a host, unless an attribute list or option is specifically requested on the command line. This means only `ipHostNumber` and `cn` are displayed by default.
  - Output lines are not broken after 80 columns.

7.3.9.1 Synopsis

```
ldaphostlist [-m] [-L] [-P] [-Z] [-ZZ] [-ZZZ] [-v]
[-h servername] [-p port] [-b base]
 [-s scope] [-n hostname | [-g groupname][-f|F filter]
-N [maxcount] [-k [keyAge]] [attr] [...]]
```

7.3.9.2 Options and Arguments

The `ldaphostlist` tool supports the following options and arguments:

-m

Exposes the names of the mapped attributes when returning results. Normally `ldaphostlist` returns results as:

fieldname`: `*value*

where *fieldname* is one of the pre-defined RFC2307 attribute names, and *value* is the resulting value for that field, after attribute mapping has been applied.

With `-m`, the actual attribute name is exposed as follows:

fieldname`[`*attributename*`]: `*value*

For example, if the RFC2307 attribute `cn` has been mapped to the `hostName` attribute. Without the `-m` option, the output of the `cn` field would appear as:

`cn: `*value-of-hostName*

When `-m` is used, and assuming the same conditions as above, the output representing the *gecos* field would appear as:

`cn[hostName]: `*value-of-hostName*

The `-m` option does not apply if the `-L` option is specified.

-L

Displays the host output in the following (`/etc/host`) format:

ipAddress hostname `[…]`

If a host entry contains more than one name, those names are repeated on the same line, separated by spaces.

If a host entry contains more than one *ipAddress*, a separate line for each IP address is displayed, using the same list of host names.

When the `-L` option is specified, the `-m` option is ignored, and the *attr* parameter list is invalid.

-P Prompts for the user's bind DN and password. Without `-P`, `ldaphostlist` attempts to bind to the directory server using the environment variables LDAP_BINDDN and LDAP_BINDCRED. Or if those were not specified, the bind will be anonymous or as the LDAP-UX proxy user, if configured.

-Z Requires an SSL connection to the directory server, even if the ldapux(5) configuration does not require the use of SSL. Use of `-Z` requires that either a valid server or CA certificate be defined in the `/etc/opt/ldapux/cert8.db` file. An error occurs if the SSL connection could not be established. See Section 7.3.9.5 (page 283) for additional details.

-ZZ Attempts a TLS connection to the directory server, even if the ldapux(5) configuration does not require the use of TLS. If a TLS connection cannot be established, a non-TLS and non-SSL connection is established. Using `-ZZ` is not recommended (use `-Z` or `-ZZZ` instead) unless alternative methods are used to protect against network eavesdropping. Use of `-ZZ` requires that either a valid server or CA certificate be defined in the `/etc/opt/ldapux/cert8.db` file. See Section 7.3.9.5 (page 283) for additional details.

-ZZZ Requires a TLS connection to the directory server, even if the ldapux(5) configuration does not require the use of TLS. Use of `-ZZZ` requires that either a valid server or CA certificate be defined in the `/etc/opt/ldapux/cert8.db` file. An error occurs if the TLS connection could not be established. See Section 7.3.9.5 (page 283) for additional details.

-v Displays additional information used to analyze and troubleshoot usage issues. If attributes from a requested *`attr`* list are not displayed as expected, `-v` may provide additional information.

-h *servername* Specifies the host name and optional port number (*`hostname`*:*`port`*) of the directory server where the hosts are managed. This option overrides the server list configured by ldapux(5). The *`hostname`* field also supports specification of IPv4 and IPv6 addresses. If you specify a port for an IPv6 address, the IPv6 address must be specified in square-bracketed form. If the optional port is unspecified, the port number is assumed to be 389 or 636 for SSL connections (`-Z`). Refer to "Binding to the Directory Server" below for additional details.

-p *port* Specifies the port number of the directory server to contact. This option is ignored if the port number is specified in the *`servername`* as part of the `-h` option. Refer to the Binding to the Directory Server section for additional details.

If the `!` option is specified, the host is removed as a member from the specified group. If the `!` is specified by itself, the host is removed from all groups of which it is a member.

-n *hostname* Provides a simplified method for discovering a single host. Use of `-n` is the same as `-f "(cn=`*`name`*`)"`. If `-n` is used, the `-g`, `-F` and `-f` options cannot be specified on the command line.

-g *groupname* Limits the hosts returned to those that are also members of the specified group. The LDAP group is discovered by searching for any entries under the default base (as configured in the LDAP-UX

profile or specified using `-b`) that are of the `groupOfNames`, or `groupOfUniqueNames` object class and have the specified *`groupname`*. `ldaphostlist` enumerates the members of the specified group, searching for members that are hosts, and then displays those entries. The `-f` or `-F` option can be used to further narrow the list of returned host entries.

Note that the `-n` and `-m` options are mutually exclusive

`-b` *`base`*

Overrides the search base as defined in the ldapux(5) configuration. *`base`* is a distinguished name (DN) that describes the lowest (with the tree branches facing up) location in the directory tree at which to start the search. If *`base`* is not specified, `ldaphostlist` uses the search base from the hosts `serviceSearchDescriptor` or `defaultSearchBase`, as defined in the LDAP-UX configuration profile, per section 4.6 of RFC 4876. If a partial DN is put into the`serviceSearchDescriptor`, it is combined with the `defaultSearchBase`. For example, if we have the following::

```
defaultsearchbase: dc=chn,dc=hp,dc=com
servicesearchdescriptor:
hosts:ou=hosts,?sub?(objectclass=iphost)
```

Then the search base for the hosts service will be:

```
ou=hosts,dc=chn,dc=hp,dc=com
```

`-s` *`scope`*

Overrides the search scope as defined in the ldapux(5) configuration. *`scope`* specifies how deep in the directory tree the search should search. *`scope`* can be one of the following keywords:

- `base`

  Performs a search only on the base specified with the `-b` option.

- `one`

  Searches all entries that are immediate child entries of the base.

- `sub`

  Searches all all entries below and including the base.

`-f` *`filter`*

Specifies an LDAP-style search filter, used to select specific host entries from the LDAP directory. When `-f` is used, the specified *`filter`* is assumed to apply to POSIX-style host entries. This means the filter specified with `-f` is amended with the default ldapux(5) search filter for the host object type. In addition, when `-f` is used, if a known attribute for the host service (see the lists defined in Section 7.3.9.3 (page 282)), has been mapped as defined by the ldapux(5) configuration profile, then the mapped attribute name is substituted in the search filter.

Consider an example with the following command:

```
ldaphostlist -f "(cn=myhost)"
```

And assume the LDAP-UX product has been configured as follows:

- ◎The configuration profile defines the search filter for the host service as `"(objectclass=ipHost)"`.
- ◎The `cn` attribute for the host service has been mapped to the `hostName` attribute.

Then the actual search filter used by `ldaphostlist` would be:

```
(&(objectclass=ipHost)(hostName=myhost))
```

Notes:

- ◎When `-f` is used and any of the attributes specified in the search filter have been mapped to "*NULL*", `ldaphostlist` returns an error.
- Attributes that are not part of the LDAP-UX configuration profile mapping for the host service are not modified. Refer to RFC2307: *An Approach for Using LDAP as a Network Information Service* for the list of attributes that may be mapped.
- ◎Specifying `-n` and `-f` on the same command line results in an error.

`-F` *`filter`*

Similar to `-f`, except that *`filter`* is assumed to be immutable, and the ldapux(5) host filter from the configuration profile is not amended to the specified filter, nor will attribute mapping apply to the filter.

Notes:

- When `-F` is used, the specified filter should still apply to host entries. `ldaphostlist` will produce undefined results if the search filter specified with `-F` discovers user accounts instead of host entries.
- Specifying `-n` and `-F` on the same command line results in an error.

`-N` *`maxcount`*

Specifies the maximum number of entries to be returned. If this option is not specified, the maximum number of entries to be returned is 200 by default (unless `-g` is specified). Some directory servers limit the number of entries returned for a particular search request, regardless of how many entries are requested. If the *`maxcount`* limit is set too high, it might not be possible to determine if a search has returned complete results, since the directory server might have truncated the number of returned entries before reaching the requested maximum count. Although some directory servers will indicate if a specified search exceeds an enumeration limit, if *`maxcount`* is above the directory server's internal configured limit, it is not always possible to determine if all results have been returned. However, a reasonable assumption is that if *`maxcount`* entries have been returned, additional entries are likely still available that match the search criteria than just those displayed. The `-N` option is ignored if the `-g` option is specified.

`-key [ - ] [`*`keyage`*`]`

Displays the sshPublicKey for each host discovered. If keyage is preceded by "`-`", `ldaphostlist` displays only those host entries that have keys that were generated more than *`keyage`* days ago. If *`keyage`* is not preceeded by "`-`", `ldaphostlist` displays only those entries that have keys that are considered expired or that will expire within *`keyage`* days. Host entries might not have key age or expiration information defined in the directory server, and therefore this *`keyage`* option will apply to only those host entries that do. Please see the `ldaphostmgr` command and the `-k` and `-e` options for additional information about key ages and

expiration. Use of `-k` is only recommended if the user performing the search request is not subject to directory server search-size limits, since `ldaphostmgr` must retrieve each entry to determine its *keyage* meets the specified criteria.

If `-k` is specified, but none of the `-n`, `-g`, `-f`, nor `-F` options are specified, then only hosts that have `sshPublicKey` attributes are displayed.

keyage is optional. If it is not specified, all hosts that have sshPublicKeys will be displayed, unless limited by the `-n`, `-g`, `-f` or `-F` options.

attr

Specifies additional LDAP attributes to display besides the pre-defined RFC2307 attributes for hosts. Do not use *attr* with the `-L` option. Attributes specified in the *attr* list are assumed to not be part of RFC2307, and are therefore not mapped. When the `-m` option is specified, a value specified by *attr* is always in the following the output format:

attributename[*attributename*]: *value*

When binding to the directory server using the LDAP-UX proxy user, `ldaphostlist` does not allow use of the *attr* argument, unless the system administrator has attested that the proxy user does not have permissions beyond that of a nonprivileged user. This limitation prevents regular HP-UX users from discovering LDAP data not previously displayed by LDAP-UX. Use of the *attr* argument requires that either the user has the rights to use the LDAP-UX Administrator Credential (`/etc/opt/ldapux/acred`), or that the user running `ldaphostlist` has specified an identity using the `-P` option or LDAP_BINDDN and LDAP_BINDCRED environment variables. See Section 7.3.9.7 (page 284) for additional information.

7.3.9.3 Output Format

Output from `ldaphostlist` follows a consistent format, regardless of which attributes are used to define information in an LDAP directory. The output format is:

```
dn: dn1
field1: value1
field2: value2
field3:: base64-encodeded-value3
…

dn: dn2
field1: value1
field2: value2
…
```

Each entry is preceded by a DN, followed by one or more field-value pairs. The DN and each field-value pair is e on a separate line, separated by a carriage-return and line-feed character. The field and value are separated by a colon and space character. Each entry is separated by a blank line. In the event an unencodable character is encountered (carriage-return or line-feed for example) in a value string, the whole value is base64-encoded and the field-value separator changes to two colons and a space character. See “Unencodable Characters” in Section 7.3.9.3 (page 282).

By default the following fields are returned:

```
cn
ipAddress
```

Note that when the -m option is specified, the output format changes to the following:

```
dn: dn1
field1[attribute1]: value1
field2[attribute2]: value2
field3[attribute3]:: base64-encodeded-value3
…
```

7.3.9.4 Special Considerations for Output Format

UTF8

Since LDAP directories require data to be stored according to the UTF-8 (RFC3629) character encoding method, all characters displayed by ldaphostlist are UTF-8, and assumed to be part of the ISO-10646 character set. ldaphostlist does not perform conversion of the locale character set to/from the UTF-8 character set.

Unencodable Characters (Base64 encoding)

In the ldaphostlist output format, each displayed field is delimited by a new line (carriage-return and line-feed). To assure that ldaphostlist displays only printable and LDIF encodable characters, all characters less than 32 (ASCII space), except for 9 (ASCII horizontal tab), and the character 127 (ASCII delete) will cause the value to be converted into a base-64 encoded string. Characters above 127 are assumed be from the UTF-8 character set, and printable. If the output lines are long, the data is not broken into multiple lines.

Encoding of the DN

ldaphostlist displays DN strings according to the encoding rules defined in RFC4514. The backslash escape character (\ ) precedes special characters, which can be the character itself or a 2-digit hex representation of the character.

7.3.9.5 Binding to the Directory Server

ldaphostlist is designed to take advantage of the existing ldapux(5) configuration for determining to which directory server to bind, and how to perform the bind operation. ldaphostlist consults the ldapux(5) configuration profile for the following information:

- The list of LDAP directory server hosts
- The authentication method (simple passwords, SASL/DIGEST-MD5, and so on)

If neither of the environment variables LDAP_BINDDN and LDAP_BINDCRED were specified, ldaphostlist also consults the ldapux(5) configuration for the following additional information:

- The type of credential (user, proxy or anonymous) to use
- The credential used for binding as a proxy user (either /etc/opt/ldapux/acred for administrative users or /etc/opt/ldapux/pcred for nonprivileged users)

ldaphostlist displays an error message if LDAP_BINDDN is specified and LDAP_BINDCRED is not, unless the -P option was specified.

As with ldapux(5), ldaphostlist attempts to contact the first available directory server as defined in the ldapux(5) host list. As soon as a connection is established, further directory servers on the host list are not contacted. Once connected, ldaphostlist first determines if the environment variables LDAP_BINDDN and LDAP_BINDCRED were specified (if the -P option was not specified). If so, then ldaphostlist attempts to bind to the directory server using the specified credentials and configured LDAP-UX authentication method. If these environment variables were not specified, then ldaphostlist determines if the configured credential type is “proxy” and, if so, attempts to bind to the directory server using the configured LDAP-UX proxy credential. If configured, the acred proxy credential is used for administrative users (determined if the user running ldaphostlist has enough privilege to read the /etc/opt/ldapux/acred file). Otherwise, the credential configured in /etc/opt/ldapux/pcred is used. If the proxy credential is not configured and the -P option has not been specified, ldaphostlist connects anonymously.

NOTE: To prevent discovery of the LDAP administrator's credentials, the LDAP user DN and password cannot be specified as command-line options to the `ldaphostlist` utility.

7.3.9.6 Errors and Warnings

Upon exit, `ldaphostlist` returns a 0 (zero) exit status if no errors or warnings were encountered. If `ldaphostlist` encounters an error or warning; a nonzero exit status is returned, and one or more messages are logged to `stderr`. Messages have the following format:

```
ERROR: code:
          message
or
WARNING: code:
          message
```

Leading extra white space might be inserted to improve readability and follow 80-column screen formatting. *`code`* is a programmatically parsable error key-string, while *`message`* is human-readable. Refer to the *LDAP-UX Client Services Administrator's Guide* for a list of possible error codes generated by the LDAP user and group management tools.

7.3.9.7 External influences

7.3.9.7.1 Environment Variables

The `ladpahostlist` tool supports the following environment variables:

`LDAP_BINDDN` Specifies the DN of a user with sufficient directory server privilege to discover and enumerate hosts in the LDAP directory server. While this variable is optional, if LDAP_BINDDN is specified, LDAP_BINDCRED must also be specified.

`LDAP_BINDCRED` Specifies a password or other type of credential used for the user specified by the LDAP_BINDDN. While this variable is optional, if LDAP_BINDCRED is specified, LDAP_BINDDN must also be specified.

7.3.9.7.2 LDAP-UX Configuration

If `ldaphostlist` binds to the directory server using the proxy user's credential (this can happen if LDAP-UX is configured to use the proxy user, and credentials were not provided to `ldaphostlist`, as described in Binding to the Directory Server), the attributes displayed by `ldaphostlist` might be limited. This can occur because `ldaphostlist` must assume that the LDAP-UX proxy user has more rights to view data in the directory server than a nonprivileged user. (For example, assume an administrator configured the `cn=Directory Manager` as a proxy user). In this scenario, `ldaphostlist` will only display the `cn`, `ipHostNumber`, and `sshPublicKey` attributes, even when the *`attr`* list is requested. If LDAP-UX is configured to use the proxy user, you can indicate to `ldaphostlist` that the proxy user does not have special privileges. To do so, modify the `proxy_is_restricted` parameter in the `/etc/opt/ldapux/ldapclientd.conf` file. Setting `proxy_is_restricted` to `1` allows `ldaphostlist` to display any attribute requested in the *`attr`* list, if the proxy user is allowed to view that attribute.

7.3.9.8 Security Considerations

To support non-interactive use of the `ldaphostlist` command, specification of the LDAP user's credentials may be required. In non-interactive mode, these credentials are specified in the LDAP_BINDDN and LDAP_BINDCRED environment variables. To prevent exposure of these environment variables, they should be unset after use. Note that the shells(4) command history log may contain copies of the executed commands that show setting of these variables. Access to a shell's history file must be protected. Specification of the LDAP user's credentials on the command line is not allowed since information about the currently running processes can be exposed externally from the session. Specifying the `-P` option allows for interactive prompting

of the user's credentials, and therefore eliminates the need to specify the LDAP_BINDDN and LDAP_BINDCRED environment variables.

`ldaphostlist` only displays attributes for hosts for which the user has sufficient privilege to view. By default, (if neither the `-P` option nor the environment variables have been specified), `ldaphostlist` binds to the directory server anonymously, or uses the proxy user's credentials if configured. When `ldaphostlist` uses the proxy user's credentials to bind, the information displayed might be limited. See Section 7.3.9.7 (page 284) for additional information.

7.3.9.9 LDAP-UX Profile

`ldaphostlist` makes use of the LDAP-UX configuration profile to determine the information model used in the directory server to store POSIX attributes. Refer to the *LDAP-UX Client Services Administrator's Guide* for additional information about the configuration profile.

7.3.9.10 Limitations

`ldaphostlist` does not perform conversion of the locale character set to/from the UTF-8 character set.

7.3.9.11 Examples

Examples of how to use ldaphostmgr can be found in the *LDAP-UX Client Services Administrators Guide*.

7.3.9.12 See Also

ldaphostmgr(1m), ldapugadd(1m), ldapugmod(1m), ldapugdel(1m), ldapcfinfo(1m) and ldapux(5)

7.3.10 ldapcfinfo tool

Use the `ldapcfinfo` tool to discover LDAP-UX configuration information about the LDAP-UX product. The `ldapcfinfo` tool can also be used to discover the list of required attributes when creating new users or groups to an LDAP directory server. Non-interactive LDAP applications can use this tool to find LDAP-UX configuration details when adding new users or groups. The `ldapcfinfo` tool can also report if LDAP-UX is properly configured and active for the specified service.

7.3.10.1 Synopsis

```
ldapcfinfo [-t <type>] [-T <template_file>] [-a <DN>] [-m <atobName>]
[-A|-P|-D|-L |-R|-b|-s |-f |-h]
```

7.3.10.2 Options

NOTE: Because each of the `-a`, `-D`, `-A`, `-P`, `-R`, `-L`, `-b`, `-f`, `-h` and `-m` options described below generates arbitrary output formats, you may only use one of these options per invocation of the `ldapcfinfo` command. Use of multiple of these options in a single command line may prevent you from distinguishing outputs applied to a specific option, and will result in an error. The `-T` option is ignored unless the `-R` option is specified.

The `ldapcfinfo` tool supports the following command options:

`-t <type>`
Specifies the type of the service name for which to retrieve configuration information. The valid service names are `passwd`, `group`, `netgroup`, `services`, `rpc`, `hosts`, `networks`, `automount`, NIS-based `publickey`, `protocols` and `pam`. If you do not specify this argument, `ldapcfinfo` defaults to the `passwd` name service (if applicable to the argument specified). If the `-t` option is the only argument specified on the command line, `ldapcfinfo` reports if LDAP-UX is properly configured and active for the specified service.

`-A`
Reports if the user running the `ldapcfinfo` command has the ability to access the LDAP administrator's credential, if configured. `ldapcfinfo` returns zero exit status if the user has rights to access the LDAP administrator's credential. The `ldapcfinfo` tool returns a non-zero exit status if the user does not have permission. For detailed information about the LDAP-UX administrator credential, see Section 2.4.7 (page 84) :

`-P`
Displays the distinguished name (DN) of the LDAP-UX configuration profile and LDAP directory server that stores that profile. The output format is as follows:

```
dn: distinguishedName
host: hostname/ip:port
```

If SSL or TLS is configured to download the profile, `host:` is replaced with `hostssl:`.

`-R`
Displays the required attributes as defined in the default template file or the template file specified with the `-T` option. If you do not specify the `-T` option, you must specify the `-t passwd` or `-t group` option to indicate which default template file to be examined. Each attribute required by the requested template file displays on separate lines, one per line. Because the RFC 2307 POSIX attributes

are a static known list and are required, only non-POSIX attributes are displayed.

-T <template_file> Specifies the LDIF template file to be used to create new user or group entries. The <template_file> parameter can be either a full or relative path name or a short name. A short name is defined as the distinguishing portion of the template file name. For example, for the passwd service, if the short name "operator" is specified, the resulting template file is /etc/opt/ldapux/ug_templates/ug_passwd_operator.tmpl.

All LDAP-UX default template files are stored in the /etc/opt/ldapux/ug_templates directory. A full or relative path name must begin with a slash (/) or a period (.) character.

If you do not specify this parameter, the default template file with the -t passwd option is /etc/opt/ldapux/ug_templates/ug_passwd_default.tmpl. With -t group, the default template file is /etc/opt/ldapux/ug_templates/ug_group_default.tmpl.

For detailed information about template file, see Section 7.3.5.6 (page 242).

-L Displays the list of available template files for the service specified with the -t option. The ldapcfinfo tool displays the full path name of the template files, each on a separate line.

-D Displays the LDAP default configuration values in the /etc/opt/ldapux/ldapug.conf file used for the ldapugadd command. When you specify the -t passwd option, ldapcfinfo displays the UID range, default primary GID number, default home and default shell values. If you specify the -t group option, ldapcfinfo displays the GID range.

-b Displays the primary (first) configured search base for a particular service as defined with the -t option. If you do not specify the -t option, ldapcfinfo displays the LDAP-UX default search base for the passwd service. Output format for the -b option will follow the format defined in RFC 4514, *Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names*.

-s Displays the primary (first) configured search scope for a particular service as defined with the -t option. If you do not specify the -t option, ldapcfinfo displays the LDAP-UX default search scope for the passwd service. Output value for the -s option can be base, one or sub, which represents the search scopes as defined in RFC 4516, *Lightweight Directory Access Protocol (LDAP): Uniform Resource Locator*.

-f Displays the primary (first) configured search filter for the particular service defined with the -t option. If you do not specify the -t option, the passwd service is the default. Output format is an LDAP filter following the format defined by RFC 4515, *Lightweight Directory Access Protocol (LDAP): String Representation of Search Filters*.

-m <atobName>[,..] Displays attribute or objectclass mapping for the requested attribute or object class name. The <atobName> parameter is either one of the RFC 2307 attributes or the objectclass defined for the specific service requested. If the requested attribute is mapped to more than one target attribute, ldapcfinfo displays each mapped attribute

on the same line, separated by a white space. Attribute and objectclass names are case-insensitive. The [atobName] can be specified multiple times in a comma separated list. No white space should is allowed in the list.

-a <DN> Displays the recommended list of attributes that an interactive management tool considers making available for modification for the specified entry. In order for this operation to function properly, you must specify the `-t` option with the`-a` option.

-h Displays help text.

7.3.10.3 Specific return codes for ldapcfinfo

The `ldapcfinfo` tool returns a list of the return codes as shown in Table 7-9.

Table 7-9 Return codes for ldapcfinfo

Return Code	Message
CFI_COMMANDLINE_ERR	Unknown option.
CFI_COMMANDLINE_ERR	Missing argument for the specified command option.
CFI_COMMANDLINE_ERR	Only one of the `-D`, `-A`, `-b`, `-f`, `-h`, `-m`, `s`, `a`, `-P`, `-L` or `-R` options may be specified per invocation.
CFI_COMMANDLINE_ERR	Specified attribute name (-m) missing or invalid.
CFI_COMMANDLINE_ERR	Too many attributes specified with the `-m` option.
CFI_COMMANDLINE_ERR	Unable to validate the specified template file.
CFI_COMMANDLINE_ERR	The specified argument is unknown.
CFI_INVALID_SERVICENAME	Invalid service name specified.
CFI_TEMPLATEDIR_MISSING	Template directory is missing.
CFI_TEMPLATEFILE_MISSING	A default template file for the specified service was not found.
CFI_TEMPLATE_SYNTAX1	Too long of a line found in the template file.
CFI_INTERNAL_ERROR	Internal error: Unknown service ID for the specified service found.
CFI_INTERNAL_ERROR	Internal error: Unknown bsf type passed into the specified service.
CFI_OPEN_FILE_FAILED	Unable to open file.
CFI_INVALID_CONFIGFILE	Unable to find the specified service section in the configuration file.
CFI_INVALID_CONFIGFILE	Either `LDAP_HOSTPORT`, `LDAP_HOSTPORT_SSL` or `PROFILE_DN` missing from the configuration file.
CFI_PARSE_CONFIG_FAILED	Unable to parse following configuration line in NSS subsection of the configuration file.
CFI_SEARCH_SERVICEID_FAILED	Unable to find service id for the specified service.
CFI_UNKNOWN_ATTR	Unknown attribute for the specified service.
CFI_VERIFY_PROXYCRED_FAILED	Unable to verify LDAP-UX proxy credential.
CFI_BIND_FAILED	Unable to bind to directory server specified in the LDAP-UX configuration profile. Please check configured host and port numbers as well as proxy credential information if configured.
CFI_SEARCH_BASE_NOT_EXIST	LDAP Error 32: Configured LDAP-UX search base does not exist.
CFI_VERIFY_DS_ACCESS_FAILED	Unable to verify directory server access.

Table 7-9 Return codes for ldapcfinfo *(continued)*

CFI_NOACRED	LDAP-UX administrator credential file does not exist.
CFI_NOACRED_PERM	Insufficient permissions to read the LDAP-UX administrator credential file.
CFI_ACRED_INVALID	LDAP-UX administrator credential file contains invalid credentials.
CFI_ACRED_GOOD	LDAP-UX administrator credential file valid.
CFI_NO_CF_CONFIG	The `/etc/opt/ldapux/ldapug.conf` file is missing.
CFI_READCONFIG	Unable to read the `/etc/opt/ldapux/ldapug.conf` file.
CFI_INVALID_SV_FOR_REC	Invalid service name for the `-a` option. Only `passwd` and `group` services are accepted.
CFI_INVALID_SV_FOR_DEF	Invalid service name for the `-D` option. Only `passwd` and `group` services are accepted.
CFI_UGCONF_INVALID	Invalid configuration file. Missing required configuration parameters.
CFI_CONFIG_SUCCESS	The specified service appears properly configured for LDAP-UX operation.
CFI_CONFIG_FAILURE	The specified service not configured for LDAP-UX support.

7.3.10.4 Examples

This section provides examples of using the `ldapcfinfo` tool:

The following command checks to see if the LDAP-UX is properly configured for the `passwd` service:

```
cd /opt/ldapux/bin
./ldapcfinfo -t passwd
```

If the LDAP-UX is properly configured, below is the output of the above command:

```
INFO: CFI_CONFIG_SUCCESS
      "passwd" service appears properly configured for LDAP-UX operation
```

The following command displays the attribute mapping for the `gecos` attribute which has been mapped to the `cn`, `l`, and `telephone` attributes:

```
./ldapcfinfo -t passwd -m gecos
```

The output of the command is as follows:

```
gecos=cn l telephoneNumber
```

The following command displays the attribute mapping for the `gecos` and `uidNumber` attributes. In this example,`gecos` has been mapped to `cn`, `l` and `telephone` attributes, and `uidNumber` has been mapped to the `employeeNumber` attribute:

```
./ldapcfinfo -t passwd -m gecos,uidNumber
```

The output of the command is as follows:

```
gecos=cn l telephoneNumber
uidNumber=employeeNumber
```

The following command displays the LDAP-UX default search base for the `passwd` name service. In this example, “ou=People,” has been configured as the search base for the `passwd` name service.

```
./ldapcfinfo -t passwd -b
```

The output of the command is as follows:

```
ou=People,dc=example,dc=com
```

The following command displays the LDAP-UX default search base for the `group` name service. In this example, "ou=Groups," has been configured as the search base for the `group` name service.

`./ldapcfinfo -t group -b`

The output of the command is as follows

```
ou=Groups,ou=org,dc=example,dc=com
```

The following command displays the location of the LDAP-UX configuration profile:

`./ldapcfinfo -P`

The output of the command is as follows:

```
dn: cn=ldapux-profile,ou=org,dc=example,dc=com
host: 55.2.22.15:389
```

If SSL is required to download the profile, the output appears as follows:

```
dn: cn=ldapux-profile,ou=org,dc=example,dc=com
hostssl: 55.2.22.15:636
```

The following command displays the non-POSIX attributes defined in the default template file, `/etc/opt/ldapux/ug_templates/ug_passwd_std.tmpl`, required by the `ldapugadd` command for the `passwd` name service:

`./ldapcfinfo -t passwd -R`

The output of the command is as follows:

```
surname
```

The following command displays the list of available template files for the `passwd` name service:

`./ldapcfinfo -t passwd -L`

Assume that `/etc/opt/ldapux/ug_templates/ug_passwd_std.tmpl`, `/etc/opt/ldapux/ug_templates/ug_passwd_default.tmpl` `/etc/opt/ldapux/ug_templates/ug_passwd_ads.tmpl` are currently available on the system, the output of the above command is as follows:

```
/etc/opt/ldapux/ug_templates/ug_passwd_ads.tmpl
/etc/opt/ldapux/ug_templates/ug_passwd_std.tmpl
/etc/opt/ldapux/ug_templates/ug_passwd_default.tmpl
```

The following command displays the LDAP default configuration values in the `/etc/opt/ldapux/ldapug.conf` file used for the `ldapugadd` tool:

`./ldapcfinfo -t passwd -D`

Below is the output of the above command for the `passwd` name service:

```
uidNumber_range=100:20000
default_gidNumber=20
default_homeDirectory=/home
default_loginShell=/usr/bin/sh
```

The following command displays the LDAP default configuration values in the `/etc/opt/ldapux/ldapug.conf` file for the `group` name service:

`./ldapcfinfo -t group -D`

Below is the output of the command:

```
gidNumber_range=100:2000
```

The following command displays the recommended list of attributes that an interactive management tool considers making available for modification for the user account entry with the distinguished name, "`cn=slouie,ou=people,ou=org,dc=example,dc=com`":

`./ldapcfinfo -t passwd -a "cn=slouie,ou=people,ou=org,dc=example,dc=com"`

Below is the output of the command:

```
cn
uid
uidnumber
gidnumber
```

```
loginshell
homedirectory
gecos
description
```

7.4 LDAP directory tools

This section briefly describes the `ldapentry`, `ldappasswd`, `ldapsearch`, `ldapmodify` and `ldapdelete`.

For detailed information about `ldapsearch`, `ldapmodify`, and `ldapdelete`, see the *HP-UX Directory Server administrator guide* available at the following website:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX Directory Server**.

7.4.1 ldapentry

`ldapentry` is a script tool that simplifies the task of adding, modifying and deleting entries in a Directory Server. It supports the following name services: passwd, group, hosts, rpc, services, networks, and protocols.

`ldapentry` accepts run-time options either on the command line, or via environment variables, which can be defined locally, in the configuration profile or are read in from the configuration profile. The add and modify functions open an entry into an editor with a pre-defined template to aid the user in providing the necessary directory attributes. The template file is customizable and can be found in `/etc/opt/ldapux/ldapentry.templates`.

The `ldapentry` command also accepts options through environment variables, configuration files, and the LDAP configuration profiles.

Configuration variable

Configuration variables can be defined in the following locations (from most specific to most general):

1. as shell environment variables
2. in a user 'rc' configuration file (`~/.ux_ldap_admin_rc`)
3. in a global configuration file (`/etc/opt/ldapux/ldapclient.conf`)
4. in the configuration profile (`/etc/opt/ldapux/ldapux_profile.ldif`)

The order of evaluation is that any settings on more specific locations will overwrite any settings on more general locations.

Environment variables

The following environment variables can be defined:

LDAP_BINDDN	The DN of the LDAP user allowed to add, delete, or modify the entry.
LDAP_BINDCRED	The password for the above specified LDAP user. It is recommended to not store the password in any configuration file, the user will be prompted for it when running ldapentry.
LDAP_HOST	Host name of LDAP directory server.
LDAP_BASEDN	The DN of the search base which tells `ldapentry` where to start the search for the entry. In case of adding an entry, LDAP_BASEDN determines the insert base.
LDAP_SCOPE	The scope of LDAP search (sub, one, base). Will default to sub if LDAP_BASEDN is defined, but LDAP_SCOPE is not. You must define LDAP_BASEDN, if you define LDAP_SCOPE.
INSERT_BASE	This DN tells ldapentry where to insert new entries. This value will default to LDAP_BASEDN or a default discovered by the configuration profile. INSERT_BASE is only used when adding entries.
EDITOR	The editor to use when an entry is added or modified.

7.4.1.1 Syntax

`ldapentry -<a|m|d> [options] <service value | dn>`

where

-a Adds a new entry to the directory.

-m Modifies an existing entry in the directory.

-d Deletes an existing entry in the directory.

`options`

-f Forces command execution with warning override.

-v Displays verbose information.

-b Specifies the DN of the search/insert base which defines where `ldapentry` starts the search/insert for the entry.

This option is optional if the `LDAP_BASED` variable is set. If specified, this option overwrites the `LDAP_BASEDN` variable setting.

-h Specifies the host name of the LDAP directory. If not specified, `ldapentry` uses the local host.

-p Specifies the TCP port number that the LDAP directory uses. The default is 389.

-D Specifies the distinguished name (DN) of an administrator who has the authority to add, modify, or delete entries in the LDAP directory.

This option is optional if the `LDAP_BINDDN` environment variables has been set. If specified, this option overwrites the `LDAP_BINDDN` variable setting.

`service`

The name of the service that will determine the type of entry to edit. Can be either passwd, group, hosts, rpc, services, or networks.

`value`

The name of the entry recognized by the directory to be added, modified, or deleted.

`dn`

The full distinguished name of the entry to add, modify or delete.

For more detailed information, see the *ldapentry*(1) manpage.

7.4.1.2 Examples

The following configuration variables are defined in the user's configuration file as ~/.ux_ldap_admin_rc:

```
LDAP_BINDDN="cn=Jane Admin,ou=admins,dc=hp,dc=com" LDAP_HOST="myhost"
```

The Command

`ldapentry -a passwd UserA`

will try to bind to the directory on server myhost as Jane Admin, prompt for the credentials, and retrieve the service search descriptor from the profile LDIF file based on the service name passwd. It will then open the template file with the editor defined by the environment variable EDITOR and collect the input to pass it to ldapmodify to add the new entry.

The Command

`ldapentry -m "uid=UserA, ou=People, o=hp.com”`

will try to bind to the directory on server myhost as Jane Admin, prompt for the credentials, and use the entered DN to retrieve the entry from the directory. It will then populate a template with the retrieved information, and collect the changes to pass to ldapmodify for execution.

NOTE: Although the ldapentry tool will allow the users to modify any information on the EDITOR window, the directory server has the final decision on accepting the modification. If the user makes an invalid LDIF syntax, violates the directory's schema or does not have the priviledge to perform the modificaiton, the ldapentry tool will report the error after the EDITOR window is closed when it tries to update the directory server with the information. The user will be given the option to re-enter the EDITOR and correct the error.

7.4.2 ldappasswd

This section describes the `ldappasswd` command and its parameters. The `ldappasswd` command, installed in `/opt/ldapux/bin`, is needed on clients that use an LDAP directory replica because the replica cannot be modified by the `passwd(1)` command, or any other command.

7.4.2.1 Syntax

`ldappasswd [`*`options`*`]`

where ***`options`*** can be any of the following:

`-b` ***`basedn`***	specifies ***`basedn`*** as the base distinguished name of where to start searching.
`-h` ***`host`***	specifies ***`host`*** as the LDAP server name or IP address.
`-c`	generates an encrypted password on the client. Use this parameter for directories that do not automatically encrypt passwords. The default is to send the new password in plain text to the directory. The HP-UX Directory Server and Redhat Directory Server support automatic encryption of passwords.
`-v`	prints the software version and exits.
`-p` ***`port`***	specifies ***`port`*** as the LDAP server TCP port number.
`-D` ***`binddn`***	specifies ***`binddn`*** as the bind distinguished name.
`-w` ***`passwd`***	specifies ***`passwd`*** as the bind password (for simple authentication).
`-l` ***`login`***	specifies ***`login`*** as the UID of the account to change; defaults to the current user.

7.4.2.2 Examples

The following is a command the directory administrator can use to change the password in the directory for the user **`steves`**:

```
ldappasswd -h sys001.hp.com -p 389 -b "ou=people,o=hp.com"
-D "cn=Jane Admin,ou=admins,dc=hp,dc=com" -w passwd -l steves
```

7.4.3 ldapsearch

You use the `ldapsearch` command-line utility to locate and retrieve LDAP directory entries. This utility opens a connection to the specified server using the specified distinguished name and password, and locates entries based on the specified search filter. Search results are returned in LDIF format. For detailed information, see the *HP-UX Directory Server configuration, command, and file reference* available at the following website:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX Directory Server**.

7.4.3.1 Syntax

`ldapsearch -b basedn [optional_options][filter] [optional_list_of_attributes]`

where

filterfilter	Specifies an LDAP search filter. Do not specify a search filter if you supply search filters in a file using the -f option.
optional_options	Specifies a series of command-line options. These must be specified before the search filter, if used.
optional_list_of_attributes	are spaces-separaed attributes that reduct the scope of the attributes returned in the search results. This list of attributes must appear after the search filter. For details, see the *HP-UX Directory Server administrator guide*.

7.4.3.2 ldapsearch options

This section lists the most commonly used `ldapsearch` command-line options. For more information, see the *HP-UX Directory Server configuration, command, and file reference*.

-b Specifies the starting point for the search. The value specified here must be a distinguished name that currently exits in the database.

-D Specifies the distinguished name (DN) with which to authenticate to the server. If specified, this value must be a DN recognized by the Directory Server, and it must also have the authority to search for the entries.

-h Specifies the host name or IP address of the Directory Server. If you do not specify a host, `ldapsearch` uses the local host.

-l Specifies the maximum number of seconds to wait for a search request to complete.

-P Specifies the TCP port number that the Directory Server uses. The default is 389.

-s Specifies the scope of the search. The scope can be one of the following:

- base: Search only the entry specified in the –`b` option or defined by the `LDAP_BASEDN` environment variable.
- one: Search only the immediate children of the entry specified in the `-b` option.
- sub: Search the entry specified in the `-b` option and all of its descendants. Perform a subtree search starting at the point identified in the `-b` option. This is the default.

-w Specifies the password associated with the distinguished name that is specified in the -D option.

-x Specifies that the search results are sorted on the server rather than on the client. In general, it is faster to sort on the server rather than on the client.

-f Specifies the file containing the search filter(s) to be used in the search. Omit this opiton if you want to supply a search filter directly to the command-line.

7.4.4 ldapmodify

You use the `ldapmodify` command-line utility to add or modify entries in an existing LDAP directory. `ldapmodify` opens a connection to the specified server using the distinguished name and password you supply, and adds or modifies the entries based on the LDIF update statements contained in a specified file. Because `ldapmodify` uses LDIF update statements, `ldapmodify` can do everything `ldapdelete` can do. For detailed information, see the *HP-UX Directory Server administrator guide* available at the following website:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX Directory Server**.

7.4.4.1 Syntax

`ldapmodify [`*`optional_options`*`]`

where

`optional_options` Specifies a series of command-line options.

7.4.4.2 ldapmodify options

The section lists the most commonly used.`ldapmodify` options. For more information, see the *HP-UX Directory Server configuration, command, and file reference*.

-a Allows you to add LDIF entries to the directory without requiring the `changetype:add` LDIF update statement. This provides a simplified method of adding entries to the directory.

-B Specifies the suffix under which the new entries will be added.

-D Specifies the distinguished name (DN) with which to authenticate to the server. If specified, this value must be a DN recognized by the Directory Server, and it must also have the authority to search for the entries.

-f This option specifies the file containing the LDIF update statements used to define the directory modification. If you do not supply this option, the update statements are read from `stdin`.

-h Specifies the host name or IP address of the Directory Server. If not specified, `ldapmodify` uses the local host.

-p Specifies the TCP port number that the Directory Server uses. The default is 389.

-q Causes each add to be performed silently as opposed to being echoed to the screen individually.

-w Specifies the password associated with the distinguished name that is specified in the `-D` option.

7.4.5 ldapdelete

You use the `ldapdelete` command-line utility to delete entries from an existing LDAP directory. `ldapdelete` opens a connection to the specified server using the distinguished name and password you provide, and deletes the entry or entries. For details, see the *HP-UX Directory Server administrator guide* available at the following website:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX Directory Server**.

7.4.5.1 Syntax

`ldapdelete [`*`optional_options`*`]`

where

`optional_options` Specifies a series of command-line options.

7.4.5.2 ldapdelete options

The section lists `ldapdelete` options most commonly used. For detailed information, see the *HP-UX Directory Server configuration, command, and file reference*.

-D Specifies the distinguished name (DN) with which to authenticate to the server. If specified, this value must be a DN recognized by the Directory Server, and it must also have the authority to delete the entries.

-h Specifies the name of the host on which the Directory Server is running. If you do not specify a host, `ldapdelete` uses the local host.

-P Specifies the TCP port number that the Directory Server uses. The default is 389.

-dn Specifies the `DN` of the entry to be deleted.

-w Specifies the password associated with the distinguished name that is specified in the `-D` option.

7.5 Schema extension utility

7.5.1 Overview

A directory schema is a collection of attribute type definitions, object class definitions and other information supported by a directory server. Schema controls the type of data that can be stored in a directory server. Although there are some recommended schemas that came originally from the X.500 standards, mostly for representing individuals and organizations, there is no universal schema standard in place for every possible application. Also, there is no standard method for installing the schema definition on a directory server. To support a particular schema definition, LDAP developers are required to manually create schema definition files in the specific format tailored for each version of a supported directory server. They also have to create a custom install program for each variety of directory servers.

To address these issues, LDAP-UX Client Services supports the schema extension utility. This tool queries the current status of the LDAP schema on an LDAP directory server and extends the LDAP server schema with new schema definitions. This tool allows creation of a schema definition in a general format, that can be installed on a number of different directory servers types (such as HP-UX Directory Server, Red Hat Directory Server, Windows Active Directory Server, and so forth). A user with valid directory server administration privileges can use this tool to query and extend schema definitions stored in an XML schema definition file into the LDAP directory server.

7.5.1.1 Benefits of the schema extension tool

The schema extension tool provides the following benefits:

- Assists application developers to easily install their application schemas to the LDAP directory server.
- Supports automated schema integration into the directory server environment.
- Extends the LDAP directory server schema with new schema definitions dynamically using the schema extension tool, or stores schema extension instructions in the specified file (usually in LDIF format) so the schema can be extended into the directory server manually.
- Reduces user effort in schema extension.
- Simplifies schema management.

7.5.2 How the schema extension utility works

The schema extension utility, `/opt/ldapux/bin/ldapschema`, automatically maps a custom schema definition in a general purpose format to the schema definition format required by the specific LDAP directory server. The HP-UX Directory Server, Redhat Directory Server, and Windows Active Directory Server (ADS) are fully supported by the `ldapschema` tool.

The schema extension utility extends the LDAP directory server with new object classes and attribute types specified in a schema definition file. This utility extends only object classes and attribute types that are not yet defined in a Directory Server schema. No new matching rules or syntaxes can be installed on a Directory Server using this tool. If any attribute types specified in the new schema definition use matching rules or syntaxes that are not defined in the LDAP directory server, the schema extension tool maps these attribute types using alternate matching rules and syntaxes the directory server supports. If no alternate matching rule or syntax is found on an LDAP directory server, the default substitute matching rule or syntax will be used instead. See Section 7.5.7 (page 315) for details.

The schema definitions are stored in an XML format file. This allows you to specify a general schema definition that can be extended on different types and versions of directory servers. See Section 7.5.4 (page 305), Section 7.5.4.2 (page 307) and Section 7.5.4.4 (page 309) for details.

For this release of LDAP-UX Client Services, the `setup` tool has not been integrated with `ldapschema`. You will continue to use the `setup` tool to extend the directory server schema

with printer, public key and automount schemas. For Windows Active Directory Server, you will continue to run the `setup` tool to extend the directory server with the automount schema.

7.5.2.1 Operations performed by the schema extension utility

The schema extension utility, `ldapschema`, supports the following two modes of operation:

1. Query Schema Status

   Based on the set of attribute types and object classes defined in the input schema definition file, this tool queries their status on the directory server schema without applying any changes to the LDAP directory server. `ldapschema` checks if new attribute types and object classes specified in the input schema file are already defined on the directory server. This tool also determines if definitions installed on the LDAP directory server match definitions specified in the schema file being queried.

2. Extend a Directory Server with Schema Definitions

   This utility supports the extend mode of operation. It can add attribute types and object classes defined in the input schema file that are not yet installed on the LDAP server to that server's schema. Only new valid attribute types and object classes can be added to the LDAP server schema. To execute the `ldapschema` utility in the extend mode, most LDAP directory servers require specifying the distinguished name and password of an administrator who has permissions to modify the schema on that server.

7.5.2.2 DTD and XML files used by ldapschema

The `ldspschema` tool uses the following XML files to perform its operations:

- LDAP Schema Definition Files

  This tool queries and extends the LDAP directory server schema with the input schema definitions stored in an XML schema definition file. Several predefined files (such as rfc3712.xml and rfc2256.xml, etc...) are stored in the `/etc/opt/ldapux/schema` directory. But the schema definition file can be stored in any directory with any file name. The file name is passed to the tool as one of the required arguments. See Section 7.5.4 (page 305) for details.

- Documentation Type Definition (DTD) Template

  LDAP-UX provides the predefined Document Type Definition template, `/etc/opt/ldapux/schema/schema.dtd`. Each schema definition file must adhere to DTD template specified in `/etc/opt/ldapux/schema/schema.dtd` file. Every XML file used by the `ldapscheam` utility must include `/etc/opt/ldapux/schema/schema.dtd` as its DTD. This DTD file is used by `ldapschema` to validate new attribute types and object classes before they can be added to the LDAP directory server. See Section 7.5.4 (page 305) for details.

WARNING! Do not modify the `schema.dtd` file, or create your own DTD template file. Modifying this file will cause `ldapschema` to fail.

- Supported Matching Rules and Syntaxes File

  The `ldapschema` utility performs LDAP directory server schema search to obtain the complete list of schema syntaxes and matching rules that the directory server supports. HP-UX Directory Server and Redhat Directory Server provide a list of supported matching rules and syntaxes as part of the schema search.

  However, some directory servers (such as Windows Active Directory Server) do not provide a list of supported syntaxes and/or matching rules as part of the directory server schema search. To support Windows ADS, LDAP-UX provides the predefined LDAP directory server definition file, `/etc/opt/ldapux/schema/schema-ads.xml`, which contains a list of schema syntaxes that Windows Active Directory Server supports.

If you choose to use the `ldapschema` tool with the directory server other than HP-UX Directory Server or Redhat Directory Server or Windows Active Directory Server, and the LDAP directory server doesn't provide a list of supported matching rules and syntaxes as part of the directory server schema search. Then, you need to define your own supported matching rules and syntaxes file. For detailed information on how to create an XML file containing supported matching rules and syntaxes for your directory server, see Section 7.5.6 (page 313).

- Mapping Rules For Unsupported Matching Rules and Syntaxes File

  If matching rules and/or LDAP syntaxes used in attribute type definitions in the schema definition file are not supported on the LDAP directory server, the `ldapschema` tool maps them using alternate matching rules and syntaxes the LDAP server supports. LDAP-UX provides the `/etc/opt/ldapux/schema/map-rules.xml` file which defines a list of default substitution matching rules and syntaxes, and alternate matching rules and syntaxes. See Section 7.5.7 (page 315) for details.

7.5.3 ldapschema (schema extension) tool

The `ldapschema` utility allows schema developers to define LDAP schemas using a universal XML syntax, greatly simplifying the ability to support different directory server variations. It can be used to query the current status of the LDAP schema on the LDAP directory server, as well as extend the LDAP directory server schema with new attribute types and object classes. The `ldapschema` utility was designed to support directory servers from several vendors and is currently supported with the HP-UX Directory Server, Redhat Directory Server, and Microsoft Windows Active Directory Server.

7.5.3.1 Syntax for ldapschema

```
ldapschema -q <schema> -T <ds_type> -V <ds_version> [options]
ldapschema -e <schema> -T <ds_type> -V <ds_version> [options]
```

7.5.3.1.1 Required command options

The following describes required options:

`-q <schema>` Queries the schema status on the LDAP directory without applying any changes to the LDAP directory server. The schema definitions can be obtained from the file name specified in the `<schema>` argument. `ldapschema` checks if any attribute types and/or object classes of the LDAP schema are already installed on the LDAP server. Also, determines if definitions installed on the LDAP server match definitions specified in the schema file being queried. See the "Schema Definition File" section for details.

`-e <schema>` Extends the LDAP directory server schema with attribute types and object classes defined in the specified schema. Schema definition is obtained from the schema file. See the "Schema Definition File" section for details. On most LDAP directory servers this option requires specifying the `-D binddn` option and either the `-j filename` or the `-w -` option to specify the credentials of an administrator who has permissions to modify the schema on the directory server.

`-T ds_type` Specifies type of LDAP directory server.

The following types of LDAP directory servers are fully supported by `ldapschema`:

Table 7-10 Supported directory servers

Type of Directory Server	ds_type
HP-UX Directory Server	hpds
Windows Active Directory Server	ads
Redhat Directory Server	rhds

The `ldapschema` utility may work with other types of LDAPv3 directory servers, although its behavior has not been verified.

Table 7-11 lists names of LDAPv3 directory servers which are reserved for future support:

Table 7-11 Reserved LDAPv3 directory servers

Type of Directory Server	ds_type
openLDAP Directory Server	openldap
Oracle Information Directory	oracle

Table 7-11 Reserved LDAPv3 directory servers *(continued)*

Novell e-Directory Server	eDirectory
IBM Tivoli Directory Server	ibm
MAC OS X Directory Server	mac
Sun One Directory Server	sun
Computer Associates Directory Server	ca
iPlanet Directory Server	iPlanet

`-V ds_version` The version of the LDAP directory server. The `strcasecmp()` function compares the version specified by this –V option and the version defined in the XML files the `ldapschema` utility processes. The version specified by the –V option and the version defined in the XML files must be consistent. For example, the schema definition file contains the following object class definition:

```
<objectClassDefinition>
       <oid>1.2.345.6.789</oid>
       <name>sampleObject</name>
       <must>sampleAttributeA</must>
       <must only="rhds"
       versionGreaterOrEqual="6.2">sampleAttributeB</must>
<objectClassDefinistion>
```

If the `ldapschema` utility is called with `<ds_version>` set to “6.2.1”, the `sampleObject` definition has two mandatory attributes, `sampleAttributeA` and `sampleAttributeB`. The strcasecmp(“6.2.1”, “6.2”) returns a positive integer, so `sampleAttributeB` is included in the definition of the object class `sampleObject`.

On the other hand, if the `ldapschema` utility is called with <ds_version> set to “6.02.1”, the sampleObject definition has only one mandatory attribute, `sampleAttributeA`. The strcasecmp(“6.02.1”, “6.2”) returns a negative integer, so `sampleAttributeB` is not included in the definition of the object class `sampleObject`.

The `ldapschema` utility ignores `<ds_version>` if the LDAP directory server version-specific attributes “`versionGreaterOrEqual`” and “`versionLessThan`” are not used in the XML files being processed (i.e., the schema definition files, the LDAP directory server definition file and the mapping rules file). If the XML files include any definitions with “`versionGreaterOrEqual`” attribute set, `strcasecmp()` must return zero or a positive integer to include directory-specific information in the LDAP schema definition. If the XML files include any definitions with “`versionLessThan`” attribute set, `strcasecmp()` must return a negative integer to include directory-specific information in the LDAP schema definition. Also, “`versionGreaterOrEqual`” and “`versionLessThan`” can be used simultaneously to define a range of version of the LDAP directory server. See Section 7.5.5 (page 311) for details.

7.5.3.1.2 Additional options (optional)

The following describes a list of options that are optional:

`-h hostname` Specifies the LDAP directory server host name or IP address. (Default: localhost)

-p <port>	Specifies the LDAP directory server TCP port number. (Default: 389 for regular connections, 636 for SSL connections.)
-D <binddn>	Specifies distinguished name (DN) of an administrator who has permissions to read and modify LDAP directory server schema.
-j <filename>	Specifies an administrator's password in the file (for simple authentication).
-w-	Inputs an administrator's password from the prompt (for simple authentication).
-Z	Establishes an SSL-encrypted connection.
-ZZ	Specifies StartTLS request.
-ZZZ	Enforces startTLS request (requires successful server response).
-P path	Specifies path to SSL certificate database. (Default: `/etc/opt/ldapux`)
-3	Verifies the host name in SSL certificates.
-s-	Disables syntax substitution in attribute types. Normally, if an attribute type uses an LDAP syntax not supported on the LDAP directory server, it is mapped to use a higher level (more inclusive) syntax supported by that server. If this option is specified, any attribute types that use unsupported LDAP syntax will not be added to the LDAP directory server schema. See Section 7.5.7 (page 315) for more details.
-m-	Disables matching rule substitution in attribute types. Normally, if an attribute type uses a matching rule not supported on the LDAP directory server, it is mapped to use a higher level (less specific) matching rule supported by that server. If this option is specified, any attribute types that use unsupported matching rules will not be added to the LDAP directory server schema. See Section 7.5.7 (page 315) for more details.
-f <filename>	Stores schema extension instructions in the specified file (usually in LDIF format). Do not apply any changes to the LDAP directory server schema. This option requires specifying the `-e` option.
-F	Forces installation of schema even if it contains any invalid attribute type or object class definitions, or some components specified in the schema file are already present in the LDAP directory server.

7.5.3.2 Security

For security reasons, the LDAP administrator's password may not be specified on the command line. It can be specified at the prompt (-w - option), in a file (-j <filename> option), or using the `LDAP_BINDCRED` environmental variable described in the "Environment Variables" section below.

7.5.3.3 Environment variables

The `ldapschema` utility supports the following environment variables:

`LDAP_BINDDN` The distinguished name (DN) of an administrator who has permissions to read and modify LDAP directory server schema.

`LDAP_BINCRED` The password for the privileged LDAP directory user.

`LDAP_HOST` The host name of the LDAP directory server. The `LDAP_HOST` variable uses the "hostname:port" format. If the port is not specified, default port number is 389 for regular connections, or 636 for SSL connections.

Options specified on the command line override the values in environment variables. For example, if the `-j /home/secret.txt` option is specified on the command line, and the `LDAP_BINDCRED` environmental variable is set, the password of the LDAP directory server administrator is obtained from the `/home/secret.txt` file.

7.5.3.4 Examples

This section describes examples using the `ldapschema` tool.

7.5.3.4.1 An example for querying the schema status

The following command queries the status of RFC 2307 schema on Red Hat directory server, `ldaphost`, with version 7.1:

```
ldapschema -q /etc/opt/ldapux/schema/rfc2307.xml -h ldaphost -T rhds
 -V 7.1
```

The LDAP directory server version number bears no effect unless also specified in the XML files being processed. Version specification must follow the same format as version specification used in the `/etc/opt/ldapux/schema/rfc2307.xml` and `/etc/opt/ldapux/map-rules.xml` files.

7.5.3.4.2 An example for extending the new schema into the directory server

The following procedures are used to extend HP-UX Directory Server, `ldaphost`, with custom `Sample` schema:

1. Create the schema definition file, `/etc/opt/ldapux/schema/sample.xml`, which contains attribute type and object class definitions for the `Sample` schema.
2. This step is recommended. Query the current status of the `Sample` schema on the server by running the following command:

   ```
   ldapschema -q /etc/opt/ldapux/schema/sample.xml -h ldaphost
   -T rhds -V 7.1 -D "<binddn>" -j /tmp/secret.txt
   ```

   The administrator's password can be specified at the prompt (-w - option) or in a file (-j <password_file> option).
3. Based on the results produced by Step 2, correct any invalid definitions.
4. Extend the HP-UX Directory Server schema with new `Sample` schema elements by executing the following command:

   ```
   ldapschema -e /etc/opt/ldapux/schema/sample.xml -h ldaphost
   -T rhds -V 7.1 -D "<binddn>" -j /tmp/secret.txt
   ```

Note that LDAP directory server version number bears no effect unless also specified in the XML files being processed. Version specification must follow the same format as version specification used in the `/etc/opt/ldapux/schema/sample.xml` and `/etc/opt/ldapux/schema/map-rules.xml` files.

7.5.4 Schema definition file

The ldapschema utility queries and extends LDAP directory server based on the XML schema definition file. When using the ldapschema tool, the schema argument used with the -q or -e option must correspond to the XML file containing the appropriate schema definition.

Several predefined files (such as rfc3712.xml, rfc2256.xml, etc...) are stored in the /etc/opt/ldapux/schema directory. But the schema definition file can be stored in any directory with any file name.

Each schema definition file must adhere to Document Type Definition (DTD) template specified in /etc/opt/ldapux/schema/schema.dtd file. Every XML file used by the ldapschema utility must include /etc/opt/ldapux/schema/schema.dtd as its DTD. See Line 2 in Section 7.5.4.1 (page 306).

WARNING! Every XML file used with ldapschema utility must include /etc/opt/ldapux/schema/schema.dtd file as its DTD template. Do not modify this file, or create your own DTD template file. The /etc/opt/ldapux/schema/schema.dtd file is created to validate attribute type and object class definitions before they can be added to the LDAP directory server schema. Altering the format of any schema elements in this file will cause ldapschema to fail.

The schema definition file, enclosed by <schemaDefinition> tags, specifies schema name, schema description and schema source, followed by any number of attribute type and object class definitions. The schema name, schema description and schema source XML tags are optional.

The following describes the schemaName, schemaDescription, and schemaSource tags:

<schemaName>	Optional, specifies the name of schema definition file.
<schemaDescription>	Optional, contains a brief one line schema description.
<schemaSource>	An optional field used to specify the X-ORIGIN field of extended attribute types and object classes, if used.

In the schema definition file, after general schema information is specified, attribute type definitions, if any, must be specified followed by any object class definitions.

7.5.4.1 Sample RFC3712.xml file

A sample `rfc3712.xml` file below defines two attribute types, `printer-name` and `printer-aliases`, followed by one object class, `printerLPR`, as specified in RFC3712:

```
Line 1:  <?xml version="1.0" encoding="UTF-8"?>
Line 2:  <!DOCTYPE schemaDefinition SYSTEM "/etc/opt/ldapux/schema/schema.dtd">
Line 3:
LINE 4:  <schemaDefinition>
Line 5:
Line 6:  <schemaName>rfc3712</schemaName>
Line 7:  <schemaDescription>Printer Services Schema</schemaDescription>
Line 8:  <schemaSource>RFC3712</schemaSource>
Line 9:
Line 10: <attributeTypeDefinition>
LINe 11:       <oid>1.3.18.0.2.4.1135</oid>
Line 12:       <name>printer-name</name>
Line 13:       <desc>A site-specific administrative name of this printer</desc>
Line 14:       <equality>caseIgnoreMatch</equality>
Line 15:       <substr>caseIgnoreSubstringsMatch</substr>
Line 16:       <syntax>1.3.6.1.4.1.1466.115.121.1.15</syntax>
Line 17:       <length>127<length>
LIne 18:       <singleValued/>
Line 19: </attributeTypeDefinition>
Line 20:
Line 21: <attributeTypeDefinition>
LINe 22:       <oid>1.3.18.0.2.4.1108</oid>
Line 23:       <name>printer-aliases</name>
Line 24:       <desc>Names in addition to the printer-name</desc>
Line 25:       <equality>caseIgnoreMatch</equality>
Line 26:       <substr>caseIgnoreSubstringsMatch</substr>
Line 27:       <syntax>1.3.6.1.4.1.1466.115.121.1.15</syntax>
Line 28:       <length>127<length>
Line 29: </attributeTypeDefinition>
Line 30:
inee 31: <objectClassDefinition>
LINe 32:       <oid>1.3.18.0.2.6.253</oid>
Line 33:       <name>printerLPR</name>
Line 34:       <desc>LPR information</desc>
Line 35:       <type>AUXILIARY</type>
Line 36:       <must>printer-name</must>
Line 37:       <may>printer-aliases</may>
Line 38: </objectClassDefinition>
Line 39:
Line 40: </schemaDefinition>
```

NOTE: Line 1–2 are required in every schema definition file. Attribute type and object class definitions closely follow the format specified in RFC 2252. Values specified for all XML tags except the `<dsSpecific>` tag must not be quoted. Only the description field (enclosed by <desc>...</desc> tags) can contain spaces.

7.5.4.2 Defining attribute types

Each attribute type definition, enclosed by `<attributeTypeDefinition>` tags, can contain the following case-sensitive tags, in the order specified:

<oid> Required. Exactly one numeric id must be specified. The <oid> value must adhere to RFC 2252 format specification.

<name> Required. At least one attribute type name must be specified. Do not use quotes around the name values. The `<name>` value must adhere to RFC 2252 format specification.

<displayName> Optional. At most one display name can be specified. This tag specifies a display name of the attribute type used by LDAP clients and administrative tools. Currently, <displayName> applies only to Active Directory Server (ADS) to specify lDAPDisplayName and adminDisplayName if different from the <name> value.

<desc> Optional. At most one description can be specified. Do not use quotes around the description value.

<obsolete> Optional, use only if applicable. Obsolete attribute types cannot be used in definitions of any other attribute types or object classes. At most one obsolete flag can be specified.

<subTypeOf> Optional, use if an attribute type has a super-type. At most one super-type can be specified. The specified super-type must already exist on the LDAP directory server, or its definition must be specified in the same schema definition file.

<equality> Optional. At most one equality rule can be specified.

<ordering> Optional. At most one ordering rule can be specified.

<substr> Optional. At most one substring matching rule can be specified.

<syntax> Required if an attribute type has no super-type. At most one LDAP syntax value can be specified.

<length> Optional indication of the maximum length of a value of this attribute. RFC 2252 specifies this value in curly braces following the attribute type's syntax. For instance, "1.3.6.4.1.1466.0{64}" can be expressed using the following tags:

```
<syntax>1.3.6.4.1.1466.0</syntax>
<length>64</length>
```

At most one syntax length value can be specified. `<length>` must contain a positive integer value.

<singleValued> Optional, use if the `SINGLE-VALUE` flag is set. At most one `singleValued` flag can be specified.

<collective> Optional, use if the `COLLECTIVE-VALUE` flag is set. At most one `collective` flag can be specified.

<noUserModification> Optional, use if `NO-USER-MODIFICATION` flag is set. At most one `noUserModification` flag can be specified.

<usage> Optional, must contain one of the following possible values:

- `userApplications`
- `directoryOperation`
- `distributedOperation`
- `dSAOperation`

At most one `usage` value can be specified..

<indexed>	Optional, use if an attribute type requires indexing. At most one `indexed` flag can be specified.
<dsSpecific>	Optional, use to specify any directory-specific information about the attribute type. See Section 7.5.5 (page 311) for details.

7.5.4.3 Attribute type definition requirements

To add the new schema to the LDAP directory server, each attribute type definition must meet the following requirements:

- The attribute type has a `<oid>` tag with one numeric id value which adheres to RFC 2252 format specification.
- The attribute type has at least one `<name>` tag with the attribute type name. Each name must adhere to RFC 2252 format specification.
- No other attribute types in the schema definition file or on the LDAP directory server have the same OID value.
- No other attribute types in the schema definition file or on the LDAP directory server have the same name values.
- The specified super-type used by the attribute type must already exit on the LDAP directory server or its definition must be specified in the same schema definition file.
- The attribute type specifies either an LDAP syntax value or a super-type. Some directory servers, for example ADS, do not support attribute type inheritance. For such directory servers, the LDAP syntax for the sub-type attribute is obtained from the super-type definition and the super-type/sub-type relationship is ignored
- The matching rules and syntaxes used by this attribute type are supported by the LDAP directory server. See the “Mapping Unsupported Matching Rules and LDAP Syntaxes” section for details.
- The inheritance hierarchy has no cycles (no circular dependencies exist in the super-class/sub-class relationships).
- If the attribute type has a super-type, they both have the same value defined in the `<usage>` tag.

7.5.4.4 Defining object classes

Each object class definition, enclosed by the <objectClassDefinition> tags, can contain the following case-sensitive tags, in the order specified:

<oid>	Required. Exactly one numeric id must be specified. The <oid> value must adhere to RFC 2252 format specification.
<name>	Required. At least one object class name must be specified. Do not use quotes around the name values. The <name> value must adhere to RFC 2252 format specification.
<displayName>	Optional. At most one display name can be specified. This tag specifies a display name of the object class used by LDAP clients and administrative tools. Currently, <displayName> applies only to Active Directory Server (ADS) to specify lDAPDisplayName and adminDisplayName if different from the <name> value.
<desc>	Optional. At most one object class description can be specified. Do not use quotes around the description value.
<obsolete>	Optional, use only if applicable. Obsolete object class cannot be used in definitions of any other object classes. At most one obsolete flag can be specified.
<subClassOf>	Optional, use if an object class has super-classes. The specified super-class must already exist on the LDAP directory server, or its definition must be specified in the same schema definition file. If the LDAP directory server allows only one super-class, then only the first <subClassOf> value will be used.
<type>	Optional, must contain one of the following possible values: STRUCTURAL, AUXILIARY, ABSTRACT. At most one type value can be specified.
<must>	Optional, use if an object class has mandatory attributes. The specified attributes must already exist on the LDAP directory, or its definition must be specified in the same schema definition file.
<may>	Optional, use if an object class has optional attributes. The specified attributes must already exist on the LDAP directory server, or its definition must be specified in the same schema definition file.
<rdn>	Optional, defines the recommended attribute to use for the relative distinguished name for new entries created with this object class. Currently, <rdn> applies only to Active Directory Server (ADS). At most one RDN can be specified.
<extendAuxiliaryClass>	Optional, applies only to AUXILIARY object classes. This tag is used to extend an object class already defined in the LDAP server schema with this new AUXILIARY object class. Currently, <extendAuxiliaryClass> applies only to Active Directory Server (ADS) to include the new AUXILIARY class as an "auxiliaryClass" in the definition of another object class already defined in the LDAP server schema.
<dsSpecific>	Optional, use to specify any directory-specific information about the object type. See Section 7.5.5 (page 311) for details.

7.5.4.5 Object class definition requirements

To add the new schema to the LDAP directory server, each object class definition must meet the following requirements:

- The object class definition contains a `<oid>` tag with one numeric id value which adheres to RFC 2252 format specification.
- The object class definition has at least one `<name>` tag with the object class name. Each name must adhere to RFC 2252 format specification.
- No other object classes in the schema definition file or on the LDAP directory server have the same numeric id value.
- No other object classes in the schema definition file or on the LDAP directory server have the same name value.
- The super-class(es) used by the object class must be defined.
- The attribute(s) used by the object classes must be defined.
- The inheritance hierarchy has no cycles (no circular dependencies exist in the super-class and sub-class relationships).
- An `ABSTRACT` object class can specify only `ABSTRACT` object class(es) as its super-class(es).
- An `AUXILIARY` object class can specify `ABSTRACT` or `AUXILIARY` object class(es) as its super-class(es).
- A `STRUCTURAL` object class can specify `ABSTRACT` or `STRUCTURAL` object class(es) as its super-class(es).

7.5.4.6 Predefined schema definition files

The following LDAP schema definition files are delivered with the LDAP-UX product:

- `/etc/opt/ldapux/schema/rfc2256.xml`
- `/etc/opt/ldapux/schema/rfc2307.xml`
- `/etc/opt/ldapux/schema/rfc2307-bis.xml`
- `/etc/opt/ldapux/schema/rfc2926.xml`
- `/etc/opt/ldapux/schema/rfc3712.xml`

These files are provided as examples to demonstrate how to define new LDAP schema definition files to use with the ldapschema utility. Since these files define attribute types and object classes that come pre-installed on most LDAP directory servers they are not intended for extending the LDAP directory server schema. Instead, these files are provided for reference when creating the new schema definition files to query and extend the LDAP directory server schema with the new attribute type and object class definitions.

7.5.5 Defining directory-specific information

Attribute type and object class definitions can be extended with directory-specific information using the <dsSpecific> tag. This is useful to maintain a single schema definition file for different types and versions of LDAP directory servers.

7.5.5.1 Example of defining directory-specific information in the attribute type definition

This section takes an example to illustrate how directory-specific information can be specified in a single attribute type definition to support HP-UX Directory Server, Redhat Directory Server, and Windows Active Directory Server definitions simultaneously.

The following is an example of the attribute type definition with directory-specific information using the <dsSpecific> tag:

```
Line 1:  <attributeTypeDefinition>
Line 2:        <oid>1.23.456.7.89101112.1.314.1.51.6<oid>
Line 3:        <name>sampleAttribute</name>
Line 4:        <displayName vendor="ads">
LINE 5:          versionGreaterOrEqual="2003">my-sample-attribute</displayName>
LINE 6:        <equlaity>caseIgnoreMatch</equality>
Line 7:        <syntax>1.3.6.1.4.1.1466.115.121.1.15</syntax>
Line 8:        <dsSpecific vendor="rhds" versionGreaterorEqual="6.2"
Line 9:                                  versionLessThan="7.1"
Line 10:            <field attr="X-ORIGIN">'Custom Schema'</field>
Line 11:       </dsSpecific>
Line 12:       <dsSpecific vendor="ads" versionLessThan="2003">
Line 13:            <field attr="systemOnly">TRUE</field>
Line 14:            <field attr="rangeLower">256</field>
Line 15:       </dsSpecific>
Line 16:       <dsSpecific vendor="ads" versionGreaterOrEqual="2003">
Line 17:            <field attr="rangeLower">512</field>
Line 18:       </dsSpecific>
Line 19:  </attributeTypeDefinition>
```

For the above example, on Red Hat Directory Server 6.2 through 7.0, the X-ORIGIN flag for sampleAttribute will be set to 'Custom Schema' as specified in the dsSpecific field. On Red Hat Directory Server 6.1 and earlier, or 7.1 and later, the X-ORIGIN flag for sampleAttribute will be set to the value specified in the <schemaSource>

On Active Directory Server 2000, the sampleAttribute is added using the same display name as specified by the <name> value, with the rangeLower attribute set to 256, and the systemOnly attribute set to TRUE.

On Active Directory Server 2003, the sampleAttribute is added using "my-sample-attribute" display name, with the rangeLower attribute set to 512, and the systemOnly attribute set to FALSE, which is the default value.

Table 7-12 Directory specific information

Attribute	RHDS 6.2–7.0	RHDS 7.1	ADS 2000	ADS 2003
Name	sampleAttribute	sampleAttribute	sampleAttribute	sampleAttribute
Display Name	N/A	N/A	sampleAttribute	my-sample-attribute
X-ORIGIN	'Custom Schema'	As Specified in <schemaSource>	N/A	N/A
systemOnly	N/A	N/A	TRUE	FALSE (default)
rangeLower	N/A	N/A	256	512

Also, the 1.3.6.1.4.1.1466.115.121.1.15 syntax is not supported on the Windows ADS, it is mapped to the corresponding Directory String syntax supported on Windows ADS, which is attributeSyntax = 2.5.5.12, oMSyntax=64. See Section 7.5.7 (page 315) for details.

7.5.5.2 Example of defining directory-specific information in the object class definition

Directory-specific information can be specified in the object class definitions as well as in optional and mandatory attributes.

The following is an example of the object class definition with directory-specific information using the <dsSpecific> tag and XML attributes, not and only:

```
Line 1:  <objectClassDefinition>
Line 2:        <oid>1.23.456.7.89101112.1.314.1.51.7<oid>
Line 3:        <name>sampleObject</name>
Line 4:        <must only="ads">serverRole</must>
Line 5:        <must not="ads">userPassword</must>
Line 6:        <may>sampleAttribute</may>
Line 7:        <dsSpecific vendor="ads">
Line 8:            <field attr="systemOnly">TRUE</field>
Line 9         </dsSpecific>
Line 10:  </objectClassDefinition>
```

For the above example, on Windows Active Directory Server, this object class has a mandatory attribute type, serverRole, and an optional attribute type, sampleAttribute. On all other types of directory servers, this object class has a mandatory attribute type, userPassword and an optional attribute, sampleAttribute. On Windows Active Directory Server, this object class has the systemOnly attribute set to TRUE.

NOTE: Directory-specific attributes and values specified in <dsSpecific> fields are not validated. You need to ensure that the values specified in these fields are legitimate and adhere to the LDAP directory server rules. The field value must be specified exactly as it is to appear in the attribute type or object class definition, using single and double quotes as applicable.

Attributes and values specified in the <dsSpecific> fields override the default attribute type and object class configurations. For example, on Windows Active Directory Server the default value of the isDefunct attribute is set to False. If the following <dsSpecific> attribute is defined, the specific setting will override the default setting and will result in the element being defunct.

```
<dsSpecific vendor="ads">
    <field attr="isDefunct">TRUE</field>
</dsSpecific>
```

7.5.6 LDAP directory server definition file

To properly install new attribute types in an LDAP directory server schema, the ldapschema utility needs to determine whether the LDAP server supports the matching rules and LDAP syntaxes used by the new attribute type definitions. The `ldapschema` utility performs an LDAP search for supported matching rules and syntaxes on the LDAP server. However, some types of directory servers do not provide this information as part of the search.

You can perform the following commands to determine if your directory server returns information about supported matching rules and LDAP syntaxes

1. To determine <schema DN>, run the following command:

   ```
   /opt/ldapux/bin/ldapsearch –b "" –s base "(objectclass=*)" subsechemasubentry
   ```

2. To obtain a list of supported matching rules and LDAP syntaxes, run the following command using schema DN information obtained from step 1:

   ```
   /opt/ldapux/bin/ldapsearch –b "<schema DN>" –s base "(objectclass=*)"  \
   matchingRules ldapSyntaxes
   ```

If the latter LDAP search in step 2 does not return a complete list of supported matching rules and LDAP syntaxes, the directory server definitions must be specified in the `/etc/opt/ldapux/schema/schema-<ds_type>.xml` file. The `<ds_type>` value must correspond to the same value specified with the `-T` option on the `ldapschema` command line. The case defined in `<ds_type>` must match identically to the case specified in the `-T` argument.

The LDAP directory server definition, enclosed by `<dsSchemaDefintion>` tags, optionally specifies schema description, followed by any number of supported matching rules and LDAP syntaxes definitions. For example, LDAP-UX provides the `/etc/opt/ldapux/schema/schema-ads.xml` file which can be used to obtain a list of syntaxes and matching rules that Windows ADS supports. Run `ldapschema` with the `–T ads` option, the corresponding directory server definition is obtained from the `/etc/opt/ldapux/schema/schema-ads.xml` file.

After general schema information is specified, supported matching rules, if any, must be specified followed by any supported LDAP syntaxes definitions.

7.5.6.1 Example of the directory server definition file

The example below defines two syntaxes with <oid> values of 2.5.5.1 and 2.5.5.2 supported on Windows ADS:

```
Line 1:  <?xml version="1.0" encoding="UTF-8"?>
Line 2: <!DOCTYPE dsSchemaDefinition SYSTEM "/etc/opt/ldapux/schema/schema.dtd">
Line 3
LINE 4:  <dsSchemaDefinition>
LINE 5:
Line 6:  <schemaDescription>ADS Syntaxes</schemaDescription>
Line 7:
Line 8:  <syntaxDefinition vendor="ads">
LINe 9:        <oid>2.5.5.1</oid>
Line 10:       <dessc>Distinguished Name</desc>
Line 11:       <oMSyntax>127</oMSyntax>
Line 12: </syntaxDefintion>
Line 13:
Line 14: <syntaxDefinition vendor="ads">
LINe 15:       <oid>2.5.5.2</oid>
Line 16:       <desc>Object Identifier</desc>
Line 17:       <oMSyntax>6</oMSyntax>
Line 18: </syntaxDefintion>
LINE 19:
Line 20: </dsSchemaDefintion>
```

Lines 1-2 are required in every LDAP directory server definition file. LDAP syntax and matching rule definitions closely follow the format specified in RFC 2252. Values specified for all XML tags must not be quoted. Only the description field (enclosed by <desc>...<desc> tages) can contain spaces.

NOTE: Only LDAP syntaxes and matching rules fully supported by the LDAP directory server can be specified in this file. The `vendor`, `versionGreaterOrEqual` and `versionLessThan` attributes can be used to specify directory-specific information.

See the `/etc/opt/ldapux/schema/schema-ads.xml` file for an example of LDAP directory server definition files.

7.5.6.2 Defining matching rules

Each `<syntaxDefinition>` tag can contain the following case-sensitive tags, in the order specified:

`<oid>` Required. Exactly one numeric id must be specified.

`<name>` Required. At least one matching rule name must be specified. Do not use quotes around the name values.

`<desc>` Optional. At most one description can be specified.

`<obsolete>` Optional, use it only if it is applicable. Obsolete matching rules cannot be used in definitions of any other attribute types. At most one obsolete flag can be specified.

`<syntax>` Required. The syntax used by the matching rule definition must also be supported on the LDAP directory server. At most one LDAP syntax value can be specified per matching rule definition.

7.5.6.3 Defining LDAP syntaxes

Each `<syntaxDefinition>` tag can contain the following case-sensitive tags, in the order specified:

`<oid>` Required. Exactly one numeric id must be specified.

`<desc>` Optional. At most one description can be specified.

`<oMSyntax>` Required on Windows ADS only, ignored on other types of LDAP directory servers

7.5.7 Mapping unsupported matching rules and LDAP syntaxes

If matching rules and/or LDAP syntaxes used in attribute type definitions in the schema definition file are not supported on the LDAP directory server, the `ldapschema` tool maps them to alternate matching rules and syntaxes the LDAP server supports. LDAP-UX provides the `/etc/opt/ldapux/schema/map-rules.xml` file which defines a list of default substitution matching rules and syntaxes, and alternate matching rules and syntaxes.

The matching rules are specified in `<equality>`, `<ordering>` or `<substr>` in the attribute type definition. The LDAP syntax is specified in the `<syntax>` tag of the attribute type definition.

The purpose of the mapping rules file is to allow an LDAP schema to be installed on an LDAP directory server even if some of matching rules and LDAP syntaxes used in the definition of that schema are not supported by the directory server. The `/etc/opt/ldapux/schema/map-rules.xml` file uses the following mapping rules guideline:

- Map more restrictive syntaxes to less restrictive syntaxes.
- Map more specific matching rules to less specific matching rules.

For example, the Integer syntax contains a subset of characters of the IA5 string syntax. Therefore, it is acceptable to map the Integer syntax to the IA5 string syntax, since the IA5 string syntax is a super-set of the integer syntax.

7.5.7.1 Examples of alternate matching rules and syntaxes in /etc/opt/ldapux/map-rules.xml

The following shows examples of alternate matching rules and syntaxes defined in the `/etc/opt/ldapux/map-rules.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mappingPolicies SYSTEM "/etc/opt/ldapux/schema/schema.dtd">

<mappingPolicies>
<defaultMatchingRulesReplacements>
      <defaultMatchingRule>
            <matchingRule>caseIgnoreMatch</matchingRule>
      </defaultMatchingRule>
</defaultMatchingRulesReplacements>

<defaultSyntaxesReplacements>
      <defaultSyntax only="ads">
            <syntax>2.5.5.12</syntax>
            <desc>Active Directory String syntax.</desc>
            <oMSyntax>64</oMSyntax>
      </defaultSyntax>

      <defaultSyntax not="ads">
            <syntax>1.3.6.1.4.1.1466.115.121.1.15</syntax>
            <desc>Directory String syntax.</desc>
      </defaultSyntax>
</defaultSyntaxesReplacements>

<matchingRulesReplacements>
      <matchingRules>
            <matchingRule>IntegerMatch</matchingRule>
            <subRule>
                <matchingRule>numericStringMatch</matchingRule>
            </subRule>
      </matchingRules>
</matchingRulesReplacements>

<syntaxesReplacements>
   <syntaxes>
      <syntax>1.3.6.1.4.1.1466.115.121.1.26</syntax>
      <desc> IA5 String Syntax.</desc>
      <equivSyntax>
```

```
            <syntax>2.5.5.5</syntax>
            <desc>Active Directory IA5 String LDAP Syntax.</desc>
            <oMSyntax>22</oMSyntax>
      </equivSyntax>
      <subSyntax>
            <syntax>1.3.6.1.4.1.1466.115.121.1.15</syntax>
            <desc>Directory String syntax.</desc>
      </subSyntax>
   </syntaxes>
</syntaxesReplacements>
</mappingPolicies>
```

How ldapschema maps unsupported matching rules and LDAP syntaxes

If any mapping rules or the syntax used by an attribute type are not supported on the LDAP server, the `ldapschema` utility checks if the appropriate substitution rule is specified in the `/etc/opt/ldapux/map-rules.xml` file. If it is specified, `ldapschema` locates the first available matching rule or syntax supported on the LDAP server, and uses it in the attribute type definition instead. If the substitution rule is not specified, or none of the substitution matching rules or syntaxes are supported on the LDAP directory server, `ldapschema` checks if the default substitution can be used.

The "`vendor`", "`versionGreaterOrEqual`" and "`versionLessThan`" XML attributes can be used to specify directory-specific information stored in `<defaultMatchingRule>` and `<defaultSyntax>` tags. If the default substitution is not supported on the LDAP server, the attribute type cannot be added to the LDAP directory server schema.

Examples

For example, an attribute type with `IA5String` syntax (`1.3.6.1.4.1.1466.115.121.1.26`) is installed on Windows ADS, where this IA5 String syntax is not supported. `ldapschema` will try using the first specified equivalent or substitution syntax supported by the target LDAP directory server. The specified equivalent syntax of `2.5.5.5` syntax with the `oMSyntax` value of 22 is supported on windows ADS and will be used in place of the original syntax value.

As another example, assume an attribute type with a Boolean equality rule is being installed on the LDAP server where this matching rule is not supported. Since no substitution policy is specified for this matching rule in the example above, the default substitution matching rule, `caseIgnoreMatch`, would be used instead, if the LDAP server supports it. If the LDAP server does not support `caseIgnoreMatch`, that attribute type cannot be installed on the LDAP server, unless its definition is modified to use another supported equality matching rule.

If the `-s-` option is specified in the `ldapsechema` tool, syntax substitution in attribute types is disabled. Any attribute types with unsupported LDAP syntaxes will not be added to the LDAP directory server schema. The `-m-` option with the `ldapschema` tool disables matching rule substitution. Any attribute types with unsupported matching rules will not be added to the LDAP directory server schema.

7.5.8 Return values from ldapschema

The `ldapschema` tool returns the following values:

`0` The operation is successful.

`-1` The operation fails.

In addition, `ldapschema` prints to STDOUT the overall status of the schema being queried or extended. Based on the schema status, any combination of the following messages is displayed. Detailed explanations of each message are specified in the square brackets following the message body text.

7.5.8.1 Schema status messages

`SCHEMA_NEW` The `<schema>` file contains attribute types and object classes that are not defined in the LDAP directory server schema.

[The **`SCHEMA_NEW`** message indicates all attribute types and object classes defined in the <schema> file are new to the LDAP directory server. The **`SCHEMA_NEW`** message indicates none of the specified definitions are currently installed in the LDAP server schema.]

`SCHEMA_FOUND` Subset of attribute types and/or object classes defined in the `<schema>` file are already part of the LDAP server schema.

[The `SCHEMA_FOUND` message indicates one or more attribute type or object class definitions specified in the <schema> file are already installed in the LDAP server schema. Such elements will be excluded from being extended on the LDAP server. Only attribute types and object classes with new and unique numeric oids and names can be added to the LDAP server schema. Check the messages containing ATTRIB_FOUND and OBJECT_FOUND described below for details.

The ldapschema utility may install any remaining new elements that are not already defined in the LDAP server schema if both of the following two conditions are met:

- The LDAP schema defined in the <schema> file is compatible with the LDAP server schema. The two schemas are compatible if the definitions of any elements found in the LDAP server schema match their definitions specified in the <schema> file.

  If the `SCHEMA_MISMATCH` message is displayed, the two schemas are not compatible. This means one or more elements installed on the LDAP server have definitions different from those specified in the <schema> file. Installation of any remaining new elements is not recommended. See definition of the `SCHEMA_MISMATCH` message below.

  If the `SCHEMA_MISMATCH` message is not displayed, the two schemas are compatible. The schema specified in the <schema> file partially exists on the LDAP server schema, and can be extended with any remaining new valid attribute type and object class definitions.

- The LDAP schema defined in the <schema> file is valid.

  If the `SCHEMA_INVALID` message is displayed, one or more definitions specified in the <schema> file are invalid and cannot be added to the LDAP server schema. Such definitions need to be corrected before the new schema elements can be extended on the LDAP server.

If the SCHEMA_INVALID message is not displayed, the schema definition in the <schema> file is valid. It partially exists on the LDAP server schema, and can be extended with any remaining new valid attribute type and object class definitions.]

SCHEMA_EXISTS No changes to the LDAP server schema are needed. All attribute types and object classes defined in the <schema> file are already part of the LDAP directory server schema.

[The SCHEMA_EXISTS message indicates the schema specified in the <schema> file is already installed on the LDAP directory server. All attribute types and object classes defined in the <schema> file are already part of the schema on the LDAP directory server. Only attribute types and object classes with new and unique numeric oids and names can be added to the LDAP server schema. Check the messages containing ATTRIB_FOUND and OBJECT_FOUND described below for details. Since the definitions specified in the <schema> file are already installed in the LDAP server schema, the ldapschema utility will make no changes to the LDAP directory server schema.]

SCHEMA_OK All attribute types and object classes specified in the <schema> file are valid.

[The SCHEMA_OK message indicates the definitions of attribute types and object classes specified in the <schema> file have valid XML format and conform to the DTD template and the LDAP directory server schema policies. This message also indicates no mismatching/incompatible definitions specified in the <schema> file are installed on the LDAP server.]

SCHEMA_INVALID The <schema> file contains one or more invalid definition of attribute types and/or object classes. Review the messages above and correct any errors in the schema definition file.

[The SCHEMA_INVALID message indicates some of the attribute types and/or object classes specified in the <schema> file have invalid definitions. This condition occurs if the definition does not conform to the LDAP directory server schema policies or the DTD template. Review the “Defining Attribute Types” and “Defining Object Classes” sections for details. Also, check the messages containing ATTRIB_INVALID, ATTRIB_UNRESOLVED, OBJECT_INVALID and OBJECT_UNRESOLVED described below for details.

Any invalid elements and any elements that depend on them will be excluded from being extended on the LDAP server. For example, if an attribute type 'sampleAttributeA' has an invalid <usage> value, and an object class 'sampleObjectO' includes 'sampleAttributeA' as a mandatory or an optional attribute, neither 'sampleAttributeA' nor 'sampleObjectO' can be added to the LDAP server schema until the <usage> value is corrected. Running the ldapschema utility in verbose mode (the -v option) can provide additional information about invalid attribute type and object class definitions. HP recommends correcting any invalid definitions before extending the LDAP directory server schema with any remaining new valid definitions.]

SCHEMA_MISMATCH The <schema> file contains one or more attribute types or object classes already installed in the LDAP server schema with incompatible (i.e., mismatching) definitions. Review the messages above and verify definitions of any mismatching schema elements. Any remaining schema

elements defined in the <schema> file cannot be added to the LDAP server schema unless the force flag ("-F" option) is specified.

[The SCHEMA_MISMATCH message indicates one or more attribute types or object classes defined in the <schema> file are already installed on the LDAP directory server, however, their definitions do not match. This means that some attribute type or object class definitions specified in the <schema> file do not match the LDAP server schema definitions of the elements with the same numeric oids or names.

Check the messages containing ATTRIB_MISMATCH and OBJECT_MISMATCH described below for the exact instances of attribute types and object classes, respectively, causing the schema mismatch.

The mismatch is caused by any differences in element definitions, such as equality matching rule, single-valued setting, attribute syntax, object class type, attribute types an object class includes, etc. For example, if an attribute type 'sampleAttributeA' installed on the LDAP directory server specifies IA5 String syntax, but the definition of 'sampleAttributeA' in the <schema> file specifies Unicode String syntax, the two attribute types are mismatching. HP does not recommend installing schemas containing mismatching definitions. If the <schema> file defines any new valid attribute types or object classes that are not present in the LDAP directory server schema and you would like to install them anyway, use the force flag (the -F option) to add them to the LDAP server schema.]

SCHEMA_REJECTED The <schema> file contains no valid attribute type or object class definitions that can be added to the LDAP directory server schema. It defines elements already installed in the LDAP directory server schema, or contains invalid definitions that hence cannot be installed. Review the messages above and correct any errors in the schema definition file.

[The SCHEMA_REJECTED message indicates no attribute type or object class definitions specified in the <schema> file meet the requirement of being both new and valid, and, therefore, cannot be added to the LDAP server schema. Any invalid definitions need to be corrected before they can be added to the LDAP directory server schema.

Check the messages containing ATTRIB_INVALID, ATTRIB_UNRESOLVED, ATTRIB_MISMATCH, OBJECT_INVALID, OBJECT_UNRESOLVED, OBJECT_MISMATCH, SCHEMA_INVALID and SCHEMA_MISMATCH for details on which attribute type and object class definitions prevent the schema from being installed.

If the <schema> file contains any mismatching or invalid definitions, HP does not recommend installing the schema on the LDAP server.]

7.5.8.2 Attribute type status messages

ATTRIB_INVALID Attribute type definition is missing a numeric oid. Edit the schema definition file to specify one <oid> tag and its value for every <attributeTypeDefiniton> definition.

[This message indicates the <oid> tag and its value need to be specified in the <attributeTypeDefiniton> definition in the <schema> file.]

ATTRIB_INVALID Attribute type definition is missing a name. Edit the schema definition file to specify at least one <name> tag and its value for every `<attributeTypeDefiniton>` definition.

[This message indicates the <name> tag and its value need to be specified in the `<attributeTypeDefiniton>` definition in the <schema> file.]

ATTRIB_INVALID Attribute type " <attribute name>" has an invalid numericoid. Edit the schema definition file to specify an RFC 2252 compliant <oid> value for this attribute type. Valid numericoid must consist of digits (0-9) that can be separated by a period (.). Leading zeroes are not allowed. See RFC 2252 for details.

This message indicates the <oid> tag and its value need to be corrected in the `<attributeTypeDefiniton>` definition in the <schema> file. The <oid> value must be compliant with RFC 2252. See RFC 2252 for details.

ATTRIB_INVALID Attribute type "`<attribute name>`" has an invalid name. Edit the schema definition file to specify an RFC 2252 compliant <name > value for this attribute type. Valid name characters include letters (A-z), digits (0-9), semicolons (;) and dashes (-). Valid name must begin with an alphabet letter (A-z). See RFC 2252 for details.

[This message indicates the <name > tag and its value need to be corrected in the `<attributeTypeDefiniton>` definition in the <schema> file. The attribute type name value must be compliant with RFC 2252. See RFC 2252 for details.]

ATTRIB_INVALID Attribute type "`<attribute name>`" must have the same usage (<usage> tag) value as its super-type. Edit the schema definition file to correct the usage value for this attribute or its super-type.

If the attribute type specifies a supertype, both this attribute type and its supertype must have the same <usage> tag value. This message indicates the <usage> tag value of the specified attribute type and the <usage> tag value of its supertype do not match. Edit the <schema> file to correct the discrepancy.

ATTRIB_INVALID Attribute type "`<attribute name>`" is missing a syntax value. Edit the schema definition file to specify a syntax (<syntax> tag) value, or a valid super-type (<subTypeOf>) value.

Most LDAP directory servers require attribute type definitions to specify either the syntax value or a super-type value. This message indicates that the specified attribute type definition in the <schema> file does not specify either of these values. Edit the <schema> file to specify either the <syntax> tag and its value, or a <subTypeOf> tag and its value in the specified attribute type definition.

ATTRIB_INVALID Attribute type "`<attribute name>`" cannot be labeled as obsolete (<obsolete> tag) if any other attribute types or object classes depend on it. Edit the schema definition file to remove the <obsolete> tag from this attribute type definition in order for it to be added to the LDAP server schema.

Obsolete attribute types cannot be added to the LDAP directory server schema if any other attribute types or object classes depend on them. This messages indicates the given attribute type cannot specify the <obsolete> tag in its definition if it is used as a super-type in any other

attribute types, or if it is used as a mandatory or optional attribute in any object classes. Edit the <schema> file to correct this discrepancy.

ATTRIB_UNRESOLVED Super-type used in "`<attribute name>`" attribute type definition is not defined in any LDAP schema.

[This message indicates the super-type specified with the <subTypeOf> tag in the given attribute type definition is undefined. Edit the <schema> file to correct the name of the super- type in the attribute type definition. The super-type used in the attribute type definition must be defined either in the LDAP directory server schema or in the <schema> file before this attribute type can be installed.]

ATTRIB_UNRESOLVED Matching Rule "<matching rule name> " used in the `<attribute name>` attribute type definition cannot be mapped because "-m -" option is specified. This matching rule is not supported on the LDAP server.

[This message indicates the matching rule specified with the <equality>, <ordering> or <substr> tag in the given attribute type definition is not supported on the LDAP directory server. Option -m - disables matching rule substitution in attribute types. Edit the <schema> file to specify an alternate matching rule supported on the LDAP server, or execute the `ldapschema` utility without the -m - option to substitute this matching rule with an alternative matching rule supported on the LDAP server.]

ATTRIB_UNRESOLVED Matching Rule "<rule name> " used in the `<attribute name>` attribute type definition cannot be mapped. This matching rule is not supported on the LDAP server.

ATTRIB_UNRESOLVED LDAP syntax "<syntax oid>" used in "`<attribute name>`" attribute type definition cannot be mapped because "-s -" option is specified. This LDAP syntax is not supported on the LDAP server.

[This message indicates the LDAP syntax specified with the <syntax> tag in the given attribute type definition is not supported on the LDAP directory server. Option -s - disables syntax substitution in attribute types. Edit the <schema> file to specify an alternate syntax supported on the LDAP server, or execute the `ldapschema` utility without the -s - option to substitute this syntax with an alternative syntax supported on the LDAP server.]

ATTRIB_UNRESOLVED LDAP syntax "<syntax oid>" used in "`<attribute name>`" attribute type definition cannot be mapped. This LDAP syntax is not supported on the LDAP server.

[This message indicates the LDAP syntax specified with the <syntax> tag in the given attribute type definition is not supported on the LDAP directory server. The default substitution syntax specified in the `/etc/opt/ldapux/schema/map-rules.xml` file is not supported on the LDAP directory server either. Edit the <schema> file to specify an alternate syntax supported on the LDAP server, or edit the `/etc/opt/ldapux/schema/map-rules.xml` file to specify a default substitution syntax supported on the LDAP server.]

ATTRIB_FOUND Attribute type "`<attribute name>`" is already installed in the LDAP server schema.

[This message indicates the LDAP directory server schema already includes a definition of an attribute type definition with the same

numeric oid or name. If the `ldapschema` utility is executed in the extend mode, the given attribute type will not be added to the LDAP directory server schema. This message is displayed in verbose mode only.]

ATTRIB_MISMATCH Definition of attribute type "`<attribute name>`" is incompatible with the definition already installed in the LDAP server schema.

ATTRIB_REJECTED attribute type "`<attribute name>`" will not be added to the LDAP server schema because it is already part of the LDAP schema.

[This message indicates the LDAP directory server schema already includes a definition of an attribute type definition with the same numeric oid or name.]

ATTRIB_REJECTED attribute type "`<attribute name>`" will not be added to the LDAP server schema because its definition is invalid.

[This message indicates definition of the specified attribute type is invalid. If the `ldapschema` utility is executed in the extend mode, the given attribute type will not be added to the LDAP directory server schema. Check the messages containing ATTRIB_INVALID for details.]

7.5.8.3 Object class status messages

OBJECT_INVALID Object class definition is missing a numeric oid. Edit the schema definition file to specify one <oid> tag and its value for every `<objectClassDefiniton>` definition.

[This message indicates the <oid> tag and its value need to be specified in the `<objectClassDefinition>` definition in the <schema> file.]

OBJECT_INVALID Object Class definition is missing a name. Edit the schema definition file to specify at least one <name> tag and its value for every `<ObjectClassDefiniton>` definition.

[This message indicates the <name> tag and its value need to be specified in the `<objectClassDefinition>` definition in the <schema> file.]

OBJECT_INVALID Object class "`<object name>`" has an invalid object type value. Edit the schema definition file to modify the value specified with the <type> tag, which can be one of the following:

- `STRUCTURAL`
- `AUXILIARY`
- `ABSTRACT`

[This message indicates the <type> tag value needs to be corrected in the`<objectClassDefinition>` definition in the <schema> file. Possible object class type values are `STRUCTURAL`, `AUXILIARY` or `ABSTRACT`. Any other type values are rejected. If the <type> tag is not specified in the `<objectClassDefinition>` definition, the default object class type value is STRUCTURAL. See RFC 2252 for details.]

OBJECT_UNRESOLVED Super-class used in "`<object name>`" object class definition is not defined in any LDAP schema.

[This message indicates the super-class specified with the <subClassOf> tag in the given object class definition is undefined.

Edit the <schema> file to correct the name of the super-class in the object class definition. The super-class used in the object class definition must be defined either in the LDAP directory server schema or in the <schema> file before this object class can be installed.]

OBJECT_UNRESOLVED Mandatory attribute used in the `<object name>` object class definition is not defined in any LDAP server schema.

[This message indicates the mandatory attribute type specified with the <must> tag in the given object class definition is undefined. Edit the <schema> file to correct the name of the mandatory attribute in the object class definition. The mandatory attribute used in the object class definition must be defined either in the LDAP directory server schema or in the <schema> file before this object class can be installed.]

OBJECT_UNRESOLVED Optional attribute used in "`<object name>`" object class definition is not defined in any LDAP server schema.

[This message indicates the mandatory attribute type specified with the <may> tag in the given object class definition is undefined. Edit the <schema> file to correct the name of the optional attribute in the object class definition. The optional attribute used in the object class definition must be defined either in the LDAP directory server schema or in the <schema> file before this object class can be installed.]

OBJECT_FOUND Object class "`<object name>`" is already installed in the LDAP server schema.

[This message indicates the LDAP directory server schema already includes a definition of an object class definition with the same numeric oid or name. If the `ldapschema` utility is executed in the extend mode, the given object class will not be added to the LDAP directory server schema. This message is displayed in verbose mode only.]

OBJECT _MISMATCH Definition of object class "`<object name>`" is incompatible with the definition already installed in the LDAP server schema.

OBJECT_REJECTED Object class "`<object name>`" will not be added to the LDAP server schema because it is already part of the LDAP schema.

[This message indicates the LDAP directory server schema already includes a definition of an object class definition with the same numeric oid or name.]

OBJECT_REJECTED Object class "`<object name>`" will not be added to the LDAP server schema because its definition is invalid.

[This message indicates definition of the specified object class is invalid. If the ldapschema utility is executed in the extend mode, the given object class will not be added to the LDAP directory server schema. Check the messages containing `OBJECT_INVALID` for details.]

7.5.8.4 Matching rules status messages

RULE_INVALID Matching rule is missing a numeric oid. Edit the schema definition file to specify one <oid> tag and its value for every `<matchingRuleDefiniton>` definition.

[This message indicates the <oid> tag and its value need to be specified in the `<matchingRuleDefinition>` definition in the `/etc/opt/ldapux/`

schema/schema-ds_type.xml file, where ds_type corresponds to the same value specified with the -T option on the command line when executing the ldapschema utility.]

RULE_INVALID Matching rule is missing a name. Edit the schema definition file to specify at least one <name> tag and its value for every <matchingRuleDefiniton> definition.

[This message indicates the <name> tag and its value need to be specified in the <matchingRuleDefinition> definition in the /etc/opt/ldapux/schema/schema-ds_type.xml file, where ds_type corresponds to the same value specified with the -T option on the command line when executing the ldapschema utility.]

RULE_INVALID Matching rule is missing an LDAP syntax. Edit the schema definition file to specify one <syntax> tag and its value for every <matchingRuleDefiniton> definition.

[This message indicates the <syntax> tag and its value need to be specified in the <matchingRuleDefinition> definition in the /etc/opt/ldapux/schema/schema-ds_type.xml file, where ds_type corresponds to the same value specified with the -T option on the command line when executing the ldapschema utility.]

RULE_INVALID Matching rule "<rule name>" used in the "<attribute name>" attribute type definition is not supported on the LDAP server. Matching rule "<substitute rule name>" will be used instead.

[This message indicates the specified matching rule <matching rule name> is not supported on the LDAP directory server. However, it was successfully mapped with a higher level (less specific) matching rule supported by that server, <substitute matching rule name>, as specified in the /etc/opt/ldapux/schema/map-rules.xml file. The attribute types which uses this matching rule with the <substr>, <ordering>, <equality> tags will use be queried or extended on the LDAP directory server using <substitute matching rule name>].

7.5.8.5 LDAP syntax status messages

SYNTAX_INVALID LDAP syntax is missing a numeric oid. Edit the schema definition file to specify one <oid> tag and its value for every <syntaxDefiniton> definition.

[This message indicates the <oid> tag and its value need to be specified in the <syntaxDefinition> definition in the /etc/opt/ldapux/schema/schema-ds_type.xml file, where ds_type corresponds to the same value specified with the -T option on the command line when executing the ldapschema utility.]

SYNTAX_INVALID LDAP syntax is missing an oMSyntax value. Edit the schema definition file to specify one <oMSyntax> tag and its value for every <syntaxDefiniton> definition.

[This message indicates the <oMSyntax> tag and its value need to be specified in the <syntaxDefinition> definition in the /etc/opt/ldapux/schema/schema-ds_type.xml file, where ds_type corresponds to the same value specified with the -T option on the command line when executing the ldapschema utility. The <oMSyntax> tag is required for LDAP syntax definitions supported by the Active Directory Server.]

SYNTAX_UNRESOLVED LDAP syntax "<syntax oid>" used in the "<attribute name>" attribute type definition is not supported on the LDAP server. LDAP syntax "<substitute syntax oid>" will be used instead

[This message indicates the specified syntax <syntax oid> is not supported on the LDAP directory server. However, it was successfully mapped with a higher level (more inclusive) syntax supported by that server, <substitute syntax oid>, as specified in the /etc/opt/ldapux/schema/map-rules.xml file. The attribute types which uses this syntax with the <syntax> tag will use be queried or extended on the LDAP directory server using the <substitute syntax oid>.]

Extending schema containing invalid or incompatible attribute types or object classes is not recommended. To install elements defined in a schema file containing invalid or incompatible definitions requires specifying the force option (-F).

7.6 Name service migration scripts

This section describes the shell and perl scripts that can migrate your name service data either from source files or NIS maps to your LDAP directory. These scripts are found in `/opt/ldapux/migrate`. The two shell scripts `migrate_all_online.sh` and `migrate_all_nis_online.sh` migrate all your source files or NIS maps, while the perl scripts `migrate_passwd.pl`, `migrate_group.pl`, `migrate_hosts.pl`, and so forth, migrate individual maps. The shell scripts call the perl scripts.

The migration scripts require perl, version 5 or later, which is installed with the NIS/LDAP Gateway in /opt/ldapux/contrib/bin/perl.

7.6.1 Naming context

The naming context specifies where in your directory your name service data will be, under the base DN. For example, if your base DN is "ou=unix,o=hp.com," the passwd map would be at "ou=People,ou=unix,o=hp.com". Table 7-13 shows the default naming context for the supported services. The default will work in most cases.

Table 7-13 Default naming context

Map Name	Location in the Directory Tree
passwd	ou=People
group	ou=Groups
netgroup	ou=Netgroup
hosts	ou=Devices
networks	ou=Networks
protocols	ou=Protocols
rpc	ou=Rcp
services	ou=Services

If you change the default naming context, modify the file migrate_common.ph and change it to reflect your naming context.

7.6.2 Migrating all your files

The two shell scripts `migrate_all_online.sh` and `migrate_all_nis_online.sh` migrate all your name service data either to LDIF or into your directory. The `migrate_all_online.sh` shell script gets information from the appropriate source files, such as /etc/passwd, /etc/group, /etc/hosts, and so forth. The `migrate_all_nis_online.sh` script gets information from your NIS maps using the `ypcat` command (for more information about this command, see the *ypcat*(1) manpage). The scripts take no parameters but prompt you for needed information. They also prompt you for whether to leave the output as LDIF or to add the entries to your directory. These scripts call the perl scripts described in Section 7.6.3 (page 327). You will need to modify these scripts to ensure that any calls to perl scripts not listed in Table 7-14 (page 327) are commented out, you need to comment out the following scripts in the file:

- `$PERL /opt/ldapux/migrate/migrate_fstab.pl`
- `$PERL /opt/ldapux/migrate/migrate_netgroup_byuser.pl`
- `$PERL /opt/ldapux/migrate/migrate_netgroup_byhost.pl`

NOTE: The scripts use `ldapmodify` to add entries to your directory. If you are starting with an empty directory, it may be faster for you to use `ldif2db` or `ns-slapd ldif2db` with the LDIF file. For details on `ldif2db` and `ns-slapd`, see the *HP-UX Directory Server configuration, command, and file reference*.

7.6.3 Migrating individual files

The migration scripts shown below can be used to migrate the service data, groups, hosts, netgroup, services, protocols, rpc, passwd individually from each of your source files in /etc to LDIF. These scripts are called by the shell scripts described in Section 7.6.2 (page 326). These scripts get their information from the input source file and output LDIF.

7.6.3.1 Migration scripts

The migration scripts are described in Table 7-14.

Table 7-14 Migration scripts

Script Name	Description
migrate_base.pl	creates base DN information.
migrate_group.pl	migrates groups in /etc/group.
migrate_hosts.pl [1]	migrates hosts in /etc/hosts.
migrate_netgroup.pl[2]	migrates netgroups in /etc/netgroup.
migrate_passwd.pl	migrates users in /etc/passwd.
migrate_protocols.pl	migrates protocols in /etc/protocols.
migrate_rpc.pl	migrates RPCs in /etc/rpc.
migrate_services.pl[3]	migrates services in /etc/services.

1 systems have been configured with the same host name, then the migration script migrate_host.pl will create multiple entries in its resulting LDIF file with the same distinguished name for the host name for each of the IP addresses. Since distinguished names need to be unique in an LDAP directory, users need to first manually merge the IP addresses with one designated host record and delete the duplicated records in their LDIF file. A resulting merge might look as follows:

```
. . . .
dn: cn=machineA, ou=devices, ou=unix, o=hp.com
objectClass: top
objectClass: ipHost
objectClass: device
ipHostNumber: 15.13.130.72
ipHostNumber: 15.13.104.4
ipHostNumber: 15.13.95.92
cn: mymachine
cn: hpma01.cup.hp.com
. . . .
```

2 Netgroup
- The NIS optimization maps 'byuser' and 'byhost' are not utilized.
-Each triple is stored as a single string.
-Each triple must be enclosed by parentheses, e.g "(machine, user, domain)" is a valid triple while "machine, user, domain" is not.

3 When migrating services data into the LDAP directory, users should keep in mind that only multiple protocols can be associated with one service name, but *not* multiple service ports.

7.6.3.2 Environment variables

When using the perl scripts to migrate individual files, you need to set the following environment variable:

LDAP_BASEDN The base distinguished name where you want to put data in the LDAP directory.

For example, the following command sets the base DN to "o=hp.com":

```
export LDAP_BASEDN="o=hp.com"
```

7.6.3.3 General syntax for perl migration scripts

All the perl migration scripts use the following general syntax:

scriptname inputfile [outputfile]

where

scriptname is the name of the particular script you are using. The scripts are listed below.

inputfile is the name of the appropriate name service source file corresponding to the script you are using.

outputfile is optional and is the name of the file where the LDIF is written. stdout is the default output.

7.6.4 Examples

The following command converts all name service files in /etc to LDIF:

```
$ migrate_all_online.sh
```

The following commands convert /etc/passwd into LDIF and output it to stdout:

```
$ export LDAP_BASEDN="dc=hp,dc=com"
$ migrate_passwd.pl /etc/passwd

dn: uid=jbloggs,ou=People,dc=hp,dc=com
uid: jbloggs
cn: Joe Bloggs
objectclass: top
objectclass: posixAccount
objectclass: account
userPassword: {crypt}daCXgaxahRNkg
loginShell: /bin/ksh
uidNumber: 20
gidNumber: 20
homeDirectory: /home/jbloggs
gecos: Joe Bloggs,42U-C3,555-1212
```

The following commands convert /etc/group into LDIF and place the result in /tmp/group.ldif:

```
$ export LDAP_BASEDN="o=hp.com"
$ migrate_group.pl /etc/group /tmp/group.ldif

dn: cn=mira.hp.com,ou=Groups,o=hp.com
objectclass: posixGroup
objectclass: top
cn: mira
 cn: mira.hp.com
userPassword: {crypt}*
gidNumber: 325
```

The following command migrates /etc/hosts into LDIP and place the result in /tmp/host.ldif:

```
export LDAP_BASEDN="o=hp.com"
migrate_hosts.pl /etc/hosts /tmp/host.ldif
dn: cn=hostA.hp.com,ou=Hosts,o=hp.com
objectclass: ipHost
objectclass: device
```

```
objectclass: top
ipHostNumber: 10.1.2.5
cn: HostA
cn: HostA.hp.com
```

7.7 Unsupported contributed tools and scripts

This section describes contributed tools and scripts which are not officially supported by HP at the present time.

7.7.1 beq (search) tool

The new `beq` tool expands the search capability beyond that currently offered by `nsquery`, which is limited to hosts, passwd, and group. This search utility bypasses the name service switch and queries the backend directly based on the specified library. The search will include the following services: pwd, grp, shd, srv, prt, rpc, hst, net, ngp, and grm.

The syntax for this tool, along with example output, is shown below.

7.7.1.1 Syntax

```
beq -k [n|d] -s <service> (-l <library>) (-h | -H <#>) <id1> (id1> (<id2> (...))
```

where

k [n|d] Required. The search key may be either n for name string or d for digit (a numeral search).

-s <service> Required. Indicates what backends are to be searched for information.

-l <library> Query the backend directly. Bypass the APIs and skip the name service switch.

-h Provides Help on this command.

-H <#> Specifies Help level (0-5). Larger numbers provide more information. If you specify -h or -H, no other parameters are needed.

Service	Description
pwd	Password
grp	Group
shd	Shadow Password
srv	Service
prt	Protocol
rpc	RPC
hst	Host
net	Network
ngp	Netgroup
grm	Group Membership

7.7.1.2 Examples

1. An example `beq` command using `iuser1` (user name) as the search key, `pwd` (password) as the service, and `ldap` as the library in 32-bit mode on an HP-UX 11i v2 or v3 PA-RISC machine is shown below:

```
./beq -k n -s pwd -l /usr/lib/libnss_ldap.1 iuser1

nss_status ............. NSS_SUCCESS
pw_name..........(iuser1)
pw_passwd........(*)
pw_uid...........(101)
pw_gid...........(21)
pw_age...........()
pw_comment.......()
pw_gecos.........(gecos data in files)
pw_dir...........(/home/iuser1)
```

```
pw_shell.........(/usr/bin/sh)
pw_audid.........(0)
pw_audflg.........(0)
```

Use the following beq command if you are running 64-bit applications on an HP-UX 11i v2 or v3 Integrity server machine:

```
./beq -k n -s pwd -l /usr/lib/hpux64/libnss_ldap.so.1 iuser1
```

Use the following beq command if you are running 32-bit applications on an HP-UX 11i v2 or v3 Integrity server machine:

```
./beq -k n -s pwd -l /usr/lib/hpux32/libnss_ldap.so.1 iuser1
```

2. An example beq command using user name adm as the search key, pwd (password) as the service, and files as the library on a 32-bit HP-UX 11i v2 or v3 PA-RISC machine is shown below:

```
./beq -k n -s pwd -l /usr/lib/libnss_files.1 adm
nss_status .............. NSS_SUCCESS
pw_name..........(adm)
pw_passwd.........(*)
pw_uid............(4)
pw_gid............(4)
pw_age............()
pw_comment........()
pw_gecos..........()
pw_dir............(/var/adm)
pw_shell..........(sbin/sh)
pw_audid.........(0)
pw_audflg.........(0)
```

Use the following beq command if you are running 64-bit applications on an HP-UX 11i v2 or v3 Integrity server machine:

```
./beq -k n -s pwd -l /usr/lib/hpux64/libnss_files.so.1 adm
```

Use the following beq command if you are running 32-bit applications on an HP-UX 11i v2 or v3 Integrity server machine:

```
./beq -k n -s pwd -l /usr/lib/hpux32/libnss_files.so.1 adm
```

3. An example beq command using UID number 102 as the search key, pwd (password) as the service, and ldap as the library in 32-bit mode on an HP-UX 11i v2 or v3 PA-RISC machine is shown below:

```
./beq -k d -s pwd -l /usr/lib/libnss_ldap.1 102
nss_status .............. NSS_SUCCESS
pw_name..........(user2)
pw_passwd.........(*)
pw_uid............(102)
pw_gid............(21)
pw_age............()
pw_comment........()
pw_gecos..........(gecos data in files)
pw_dir............(/home/user2)
pw_shell..........(/usr/bin/sh)
pw_audid.........(0)
pw_audflg.........(0)
```

Use the following beq command if you are running 64-bit applications on an HP-UX 11i v2 or v3 Integrity server machine:

```
./beq -k d -s pwd -l /usr/lib/hpux64/libnss_ldap.so.1 102
```

Use the following beq command if you are running 32-bit applications on an HP-UX 11i v2 or v3 Integrity server machine:

```
./beq -k d -s pwd -l /usr/lib/hpux32/libnss_ldap.so.1 102
```

4. An example `beq` command using group name `igrp1` as the search key, `grp` (group) as the service, and `ldap` as the library in 32-bit mode on an HP-UX 11i v2 or v3 PA-RISC machine is shown below:

```
./beq -k n -s grp -l /usr/lib/libnss_ldap.1 igrp1
```

```
nss_status .............. NSS_SUCCESS
gr_name..........(igrp1)
gr_passwd.........(*)
gr_gid............(21)
pw_age............()
gr_mem
(iuser1)
(iuser2)
(iuser3)
```

Use the following `beq` command if you are running 64-bit applications on an HP-UX 11i v2 or v3 Integrity server machine:

```
./beq -k n -s grp -l /usr/lib/hpux64/libnss_ldap.so.1 igrp1
```

Use the following `beq` command if you are running 32-bit applications on an HP-UX 11i v2 or v3 Integrity server machine:

```
./beq -k n -s grp -l /usr/lib/hpux32/libnss_ldap.so.1 igrp1
```

5. An example `beq` command using a gid number as the search key, `grp` (group) as the service, and `ldap` as the library in 32-bit mode on an HP-UX 11i v2 or v3 PA-RISC machine is shown below:

```
./beq -k d -s grp -l /usr/lib/libnss_ldap.l 22
```

```
nss_status .............. NSS_SUCCESS
gr_name..........(igrp2)
gr_passwd.........(*)
gr_gid............(22)
pw_age............()
gr_mem
(iuser1)
```

Use the following `beq` command if you are running 64-bit applications on an HP-UX 11i v2 or v3 Integrity server machine:

```
./beq -k d -s grp -l /usr/lib/hpux64/libnss_ldap.so.l 22
```

Use the following `beq` command if you are running 32-bit applications on an HP-UX 11i v2 or v3 Integrity server machine:

```
./beq -k d -s grp -l /usr/lib/hpux32/libnss_ldap.so.l 22
```

7.7.2 certutil (certificate database) tool

You can use the `certutil` command-line utility to create and modify the `cert8.db` and `key3.db` database files. This tool can also list, generate, modify, or delete certificates within the `cert8.db` file. You can also use this tool to create, change the password, generate new public and private key pairs, display the contents of the key database, or delete key pairs within the `key3.db` file. For detailed command options and their arguments, see *Using the Certificate Database Tool* available at the following website:

http://www.mozilla.org/projects/security/pki/nss/tools/certutil.html

7.7.3 uid2dn (display user's DN) tool

This tool, found in `/opt/ldapux/contrib/bin`, displays user's Distinguish Name (DN) information for a given UID.

7.7.3.1 Syntax

`uid2dn [`*`UID`*`]`

where *`uid`* is a user's UID information.

7.7.3.2 Examples

The following command displays the user's DN information for a given user's UID `john`:

`./uid2dn john`

The output shows below after you run the above command:

```
CN=john lee,CN=Users,DC=usa,DC=example,DC=hp,DC=com
```

7.7.4 get_attr_map.pl (get attributemap from profile) tool

This tool, found in `/opt/ldapux/contrib/bin`, gets the attributemap information for a given name service from the profile file `/etc/opt/ldapux/ldapux_profile.ldif`.

NOTE: The `get_attr_map.pl` tool is being deprecated in LDAP-UX Integration B.04.15. This tool may not be supported in a future release. Consider using the `ldapcfinfo -t <type> -m <AttriName>` command to perform the same task. See Section 7.3.10 (page 286) for details.

7.7.4.1 Syntax

`get_attr_map.pl [<`*`service`*`>.<`*`attribute`*`>]`

where ***`services`*** is the name of the supported service, ***`attribute`*** is the name of an attribute.

7.7.4.2 Examples

The following command gets the `homedirectory` attribute information for the `passwd` service:

`./get_attr_map.pl passwd homedirectory`

The following command gets the `uidnumber` attribute information for the `passwd` service:

`./get_attr_map.pl passwd uidnumber`

8 User tasks

This chapter describes tasks pertaining to the management of users.

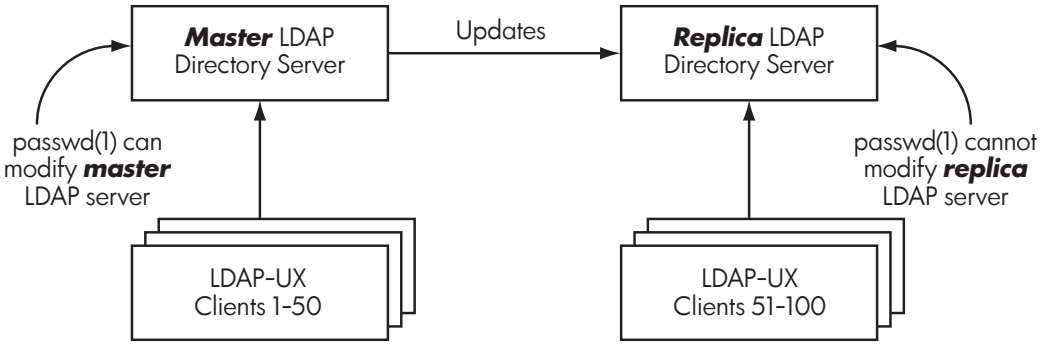
8.1 Modifying passwords

With LDAP-UX Client Services, users change their password with the `passwd` command. Depending on how you have PAM configured and depending on where the user's information is, in the directory or in /etc/passwd, users may get prompted for their password twice as PAM looks in the configured locations for the user's information.

Since LDAP directory replicas may not be modifiable, the `passwd` command may not work on clients configured to use a directory replica. In this case you could use the `ldappasswd` command (for more information about the `ldappasswd` command, see the *ldappasswd*(8) manpage). You might wrap an `ldappasswd` command in a `passwd` wrapper, similar to the `yppasswd` command. The wrapper would ask the user for the old password, call `ldapsearch` to find the current user's DN, then call `ldappasswd` and specify the master LDAP directory server. See Figure 8-3 (page 336) for an example you can modify and use.

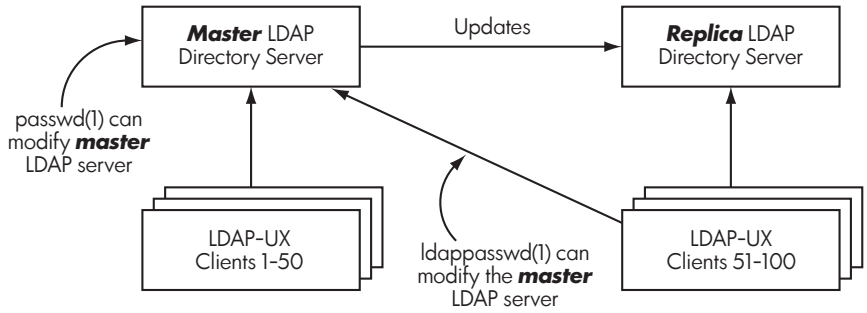
For example, referring to Figure 8-1 (page 335), say clients 1-50 use the master directory server on sys001 and clients 51-100 use the replica directory server on sys002. The `passwd` command on clients 1-50 can modify passwords in the master directory on sys001. However, the `passwd` command on clients 51-100 will fail because the replica server on sys002 cannot be modified.

Figure 8-1 Cannot change passwords on replica servers



One way to allow clients 51-100 to change their passwords is to create a new `passwd` command wrapper on these clients that calls `ldappasswd`, which modifies the master directory, as shown in Figure 8-2 (page 335). When the replica server is updated depends on how you have configured the replication. All other LDAP requests continue to go to the replica server through PAM and NSS. See below. For a sample `passwd` wrapper command, see Figure 8-3 (page 336) .

Figure 8-2 Changing passwords on master server with ldappasswd



See Section 7.4.2 (page 294) for details of this command.

Figure 8-3 Sample passwd command wrapper

```
#!/usr/bin/ksh
#
# You can put a default master LDAP server host name
# here.  Otherwise the local host is the default.
#
#LDAP_MASTER="masterHostName"

if [[ "$1" != "" ]]
  then
  LDAP_MASTER="$1"
fi

if [[ "$LDAP_MASTER" = "" ]]
  then
  eval "$(sed -e "1,/Service: NSS/d" /etc/opt/ldapux/ldapux_client.conf | \
    grep "^LDAP_HOSTPORT")"
  LDAP_MASTER="$(echo $LDAP_HOSTPORT | cut -d" " -f 1)"
fi

LDAP_BASEDN="$(grep -i "^defaultsearchbase:" \
  /etc/opt/ldapux/ldapux_profile.ldif | cut -d" " -f 2-99)"

/opt/ldapux/bin/ldappasswd -b "$LDAP_BASEDN" -h $LDAP_MASTER
```

Alternatively, your users can use a simple LDAP gateway through a web browser connected to the directory to change their password. The advantage to this method is that your users can also change their other personal information as described below.

8.2 Modifying personal information

On HP-UX, users change their personal information (sometimes called "gecos" information) such as full name, phone number, and location with the `chfn` command which changes /etc/passwd. HP-UX users change their login shell with the *chsh*(1) command, which also changes /etc/passwd. Because of authentication and access permission requirements, these commands do not directly support LDAP-managed data..

If directory server access control permissions allow, users can instead use the `ldapugmod` command to change some of their own attributes. You may need to grant users permissions to modify their own attributes. Directory server vendors may use unique methods for granting access control rights. For HP-UX Directory Server, you can review the default self-write rights granted to users in Section 2.3.2.3.2 (page 34). However, before you grant additional rights, be aware of the security impact. For example, if you allow a user to modify his own `entityRole` attribute, and that attribute is used to define access rights, then you may be granting unintentional access rights. In addition, if you want users to be able to change their own login shell, you could grant self-write permissions to the `loginShell` attribute. However, when you grant rights to modify the `loginShell` attribute, users would be able to change it to any value, meaning they can modify any program. The `chsh` command limits what valid shells may be used on a host. But users would be able to bypass this restriction if they are granted self-write rights to the `loginShell` attribute.

Also, if you have the HP-UX Directory Server, in addition to being able to use the `ldapugmod` tool, you can use the Directory Server Console or the `ldapmodify` command to change personal information.

9 Mozilla LDAP C SDK

This chapter describes the Mozilla LDAP SDK for C and the SDK file components.

9.1 Overview

The LDAP-UX Client Services provides Mozilla LDAP C SDK 6.0.5 support. The LDAP C SDK is a Software Development Kit that contains a set of LDAP Application Programming Interfaces (API) to allow you to build LDAP-enabled clients. Mozilla LDAP C SDK 6.0.5 supports IPv6 addressing. The functionality implemented in the SDK closely follows the interface outlined in RFC 2251. Using the functionality provided with the SDK, you can enable your clients to connect to LDAP v3-compliant servers and perform the LDAP functions.

The API functions provided by the LDAP C SDK allow you to perform the following major LDAP operations:

- Search for retrieving a list of entries
- Add new entries to the directory
- Update existing entries
- Delete entries
- Rename entries

NOTE: For the detailed information on how to use the LDAP API functions contained in the Mozilla SDK for C, and how to enable your client applications to connect to the LDAP servers, see the *Mozilla LDAP C SDK Programmer's Guide* at the following website:

http://www.mozilla.org/directory/csdk-docs/

9.2 The Mozilla LDAP C SDK file components

Table 9-1 (page 337) shows the Mozilla LDAP C SDK 6.0.5 file components on an HP-UX PA-RISC machine:

Table 9-1 Mozilla LDAP C SDK file components on the PA-RISC machine

Files	Description
/usr/lib/libldap.sl (32-bit) /usr/lib/pa20_64/libldap.sl (64-bit)	Main LDAP C SDK API libraries
/opt/ldapux/lib/libfreebl_32fpu_3.sl (32-bit) /opt/ldapux/lib/libfreebl_32int_3.sl (32-bit) /opt/ldapux/lib/libnspr4.sl (32-bit) /opt/ldapux/lib/libnss3.sl (32-bit) /opt/ldapux/lib/libplc4.sl (32-bit) /opt/ldapux/lib/libplds4.sl (32-bit) /opt/ldapux/lib/libsoftokn3.sl (32-bit) /opt/ldapux/lib/libssl3.sl (32-bit) /opt/ldapux/lib/pa20_64/libnspr4.sl (64-bit) /opt/ldapux/lib/pa20_64/libnss3.sl (64-bit) /opt/ldapux/lib/pa20_64/libplc4.sl (64-bit) /opt/ldapux/lib/pa20_64/libplds4.sl (64-bit) /opt/ldapux/lib/pa20_64/libsoftokn3. sl (64-bit) /opt/ldapux/lib/pa20_64/libssl3.sl (64-bit)	LDAP C SDK dependency libraries

Table 9-1 Mozilla LDAP C SDK file components on the PA-RISC machine *(continued)*

Files	Description
/usr/include/*	Include files from LDAP C SDK
/opt/ldapux/contrib/bin/certutil	Unsupported command tool that creates and modifies the certificate database files, `cert8.db` and `key3.db`
/opt/ldapux/contrib/ldapsdk/examples	Unsupported LDAP C SDK examples
/opt/ldapux/contrib/ldapsdk/source.tar.gz	Mozilla LDAP C SDK source (for license compliance)
/opt/ldapux/bin/ldapdelete /opt/ldapux/bin/ldapmodify /opt/ldapux/bin/ldapsearch /opt/ldapux/bin/ldapcmp /opt/ldapux/bin/ldapcompare	Tools to delete, modify, and search for entries in a directory; for details, see the *HP-UX Directory Server administrator guide*

Table 9-2 (page 339) shows the Mozilla LDAP C SDK 6.0.5 file components on an HP-UX Integrity server machine:

Table 9-2 Mozilla LDAP C SDK file components on an Integrity server machine

Files	Description
/usr/lib/hpux32/libldap.so (32-bit) /usr/lib/hpux64/libldap.so (64-bit)	Main LDAP C SDK API libraries
/opt/ldapux/lib/hpux32/libfreebl3.so (32–bit) /opt/ldapux/lib/hpux32/libnspr4.so (32-bit) /opt/ldapux/lib/hpux32/libnss3.so (32-bit) /opt/ldapux/lib/hpux32/libplc4.so (32-bit) /opt/ldapux/lib/hpux32/libsoftokn3.so (32-bit) /opt/ldapux/lib/hpux32/libssl3.so (32-bit) /opt/ldapux/lib/hpux32/libplds4.so (32-bit) /opt/ldapux/lib/hpux64/libfreebl3.s0 (64-bit) /opt/ldapux/lib/hpux64/libnspr4.so (64-bit) /opt/ldapux/lib/hpux64/libnss3.so (64-bit) /opt/ldapux/lib/hpux64/libplc4.so (64-bit) /opt/ldapux/lib/hpux64/libplds4.so (64-bit) /opt/ldapux/lib/hpux64/libsoftokn3.so (64-bit) /opt/ldapux/lib/hpux64/libssl3.so (64-bit) /opt/ldapux/lib/freebl_pure32_3.sl (32-bit) /opt/ldapux/lib/libfreebl_32fpu_3.sl (32-bit) /opt/ldapux/lib/libfreebl_32int_3.sl (32-bit) /opt/ldapux/lib/libnspr4.sl (32-bit) /opt/ldapux/lib/libnss3.sl (32-bit) /opt/ldapux/lib/libplc4.sl (32-bit) /opt/ldapux/lib/libplds4.sl (32-bit) /opt/ldapux/lib/libsoftokn3.sl (32-bit) /opt/ldapux/lib/libssl3.sl (32-bit) /opt/ldapux/lib/pa20_64/libnspr4.sl (64-bit) /opt/ldapux/lib/pa20_64/libnss3.sl (64-bit) /opt/ldapux/lib/pa20_64/libplc4.sl (64-bit) /opt/ldapux/lib/pa20_64/libplds4.sl (64-bit) /opt/ldapux/lib/pa20_64/libsoftokn3. sl (64-bit) /opt/ldapux/lib/pa20_64/libssl3.sl (64-bit)	LDAP C SDK dependency libraries
/usr/include/*	Include files from LDAP C SDK
/opt/ldapux/contrib/bin/certutil	Unsupported command tool that creates and modifies the certificate database files, `cert8.db` and `key3.db`
/opt/ldapux/contrib/ldapsdk/examples	Unsupported Mozilla LDAP C SDK examples
/opt/ldapux/contrib/ldapsdk/source.tar.gz	Mozilla LDAP C SDK source (for license compliance)
/opt/ldapux/bin/ldapdelete /opt/ldapux/bin/ldapmodify /opt/ldapux/bin/ldapsearch /opt/ldapux/bin/ldapcmp /opt/ldapux/bin/ldapcompare	Tools to delete, modify, and search for entries in a directory; for details, see the *HP-UX Directory Server administrator guide*

Table 9-3 (page 340) shows header files that support the LDAP libraries existing under /usr/include, except where noted:

Table 9-3 Mozilla LDAP C SDK API header files

Header Files	Description
/usr/include/ldap.h	Main LDAP functions, structures and defines.
/usr/include/ldap-extension.h	Support for LDAP v3 extended operations, controls and other server specific features. This file must be included in source code that uses LDAP v3 extended operations or controls.
/usr/include/ldap_ssl.h	Support for creation of SSL connections. This file must be included in source code that requires SSL connections.
/usr/include/srchpref.h	Support for LDAP search preferences configuration files (ldapsearchprefs.conf). A common method used by applications that use the OpenLDAP API to define organizational search preferences.
/usr/include/disptmpl.h	Support for LDAP display templates. Allows applications to convert LDAP entries into displayable text strings and HTML.
/usr/include/lber.h	Support for creating messages that follow the Basic Encoding Rules syntax. These APIs are used when building extended LDAP operations or controls. This file is a support file for ldap.h and does not need to included in source code.
/usr/include/ldap-standard.h	Contains basic LDAP defines. This file is a support file for ldap.h and does not need to be included in source code.
/usr/include/ldap-platform.h	Contains platform specific information for compiling on a variety of platforms. This file is a support file for ldap.h and does not need to be included in source code.
/opt/ldapux/include/ldap-to-be-deprecated.h	LDAP APIs that will not be available in the future. Do not use this header file for newly created LDAP-enabled applications.
/opt/ldapux/include/ldap-deprecated.h	LDAP APIs that have been deprecated. Do not use.

NOTE: If you attempt to use the LDAP C SDK in your code , you only need to put in "`#include <ldap.h>`" in the code and compile with the `-lldap` parameter to link with the LDAP C SDK library.

9.3 Legacy versions of the LDAP SDK

Version 6.0.5 of the Mozilla LDAP SDK includes changes to improve compliance with the LDAP C API specification defined by the IETF document `draft-ietf-ldapext-ldap-c-api-05.txt`. These changes modify lower-level BER structures. While the majority of these changes are maintained within the SDK itself, or opaque to the applications, those applications that use or modify binary data stored in the directory server or that make direct use of non-integrated LDAP extensions or controls, will likely be impacted. The impacted applications will be incompatible with version 6.0.5 unless they are re-compiled.

To ease the transition to Mozilla LDAP SDK 6.0.5, the previous version of LDAP SDK, 5.17.1 is available in the `/opt/dirsrv/legacy/5` directory and can be loaded by applications by using LD_LIBRARY_PATH or LD_PRELOAD as appropriate.

NOTE: No header files are provided for the legacy LDAP SDK because new applications should be built using the new LDAP SDK 6.0.5. Support for the legacy LDAP SDK will end with a future version of LDAP-UX.

The legacy version of the LDAP C SDK does not support IPv6 addressing. If your application needs to support IPv6, be sure to use LDAP C SDK 6.0.5.

10 Support and other resources

10.1 Contacting HP

HP encourages your comments concerning this document. We are truly committed to providing documentation that meets your needs.

To make comments and suggestions about product documentation, send a message to:

http://www.hp.com/bizsupport/feedback/ww/webfeedback.html

Please include document title, manufacturing part number, and any comment, error found, or suggestion for improvement you have concerning this document. Also, please include what we did right so we can incorporate it into other documents.

NOTE: HP cannot provide product support through this email address. To obtain product support, contact your HP Support Representative, your HP Services Representative, or your authorized HP reseller. For more information about support services, see the support website:

http://www.hp.com/go/support

For other ways to contact HP, see the Contact HP website:

http://welcome.hp.com/country/us/en/contact_us.html

10.2 New and changed information in this edition

This edition documents the new features introduced with LDAP-UX Client Services version B.05.00, including the following (features pertaining to Windows are documented more fully in the *LDAP-UX Client Services B.05.00 with Microsoft Windows Active Directory Server Administrator's Guide*):

- **Automated setup (simplified guided installation mode)**

  This release provides automated setup, which allows HP-UX to be quickly configured to integrate into an LDAP directory server for centralized identity and OS management. Guided installation mode allows for one-step integration into a Windows domain or LDAP-UX domain. Guided installation mode can also provision a new HP-UX Directory instance with a pre-created management domain.

- **SSH Host Key Management**

  LDAP-UX can be used to centrally manage public keys for HP Secure Shell (ssh) hosts. By provisioning host public keys into the directory server, trust between hosts and users can be pre-established, eliminating the man-in-the-middle threats. Additionally, LDAP-UX allows for central management of ssh configuration parameters.

  NOTE: This feature is not supported when using LDAP-UX Client Services with Windows ADS.

- **Offline Credential Caching**

  LDAP-UX can use locally cached user, group, and authentication credentials when contact with the directory server is lost, providing high availability for the OS and its applications. For patch requirements, the *LDAP-UX Integration B.05.00 Release Notes*.

NOTE: This feature is not supported when using LDAP-UX Client Services with Windows ADS.

- **IPv6 support**

  LDAP-UX OS integration and management tools can now connect to directory servers through IPv6 addressing.

- **compat mode performance enhancement**

  For organizations that rely on the legacy netgroup `/etc/passwd` filtering, the compat mode performance enhancement significantly improves performance when numerous and large netgroups are used in the `/etc/passwd` file for controlling passwd fields.

- **Local-only profile support**

  The centrally managed LDAP-UX configuration profile uses a schema defined by RFC 4876. For environments where modification of the directory server schema is not allowed and new schema cannot be installed, the local-only profile allows LDAP-UX to manage configuration on the local hosts instead of the directory server. You need to use the `-l` option with the customized `setup` program to obtain this feature.

- **User Group Management Tools Enhancements**

  The user and group management tools are enhanced to provide the following:
  - The DN of the current user as a default when prompting for a DN before binding to the directory server.
  - The ability to change or reset a user's ADS password if SSL has been configured. This includes the ability of an administrator to reset a user's password.

- **pam_authz Enhancements**

  The following `pam_authz` is enhancements have been made:
  - `pam_authz` now allows granular access control policies to be applied to individual PAM services (such as ftp, telnet, ssh, imapd, and so forth). Different policies can be applied to each service.
  - `pam_authz` now supports a new action for rules. In addition to `allow` or `deny`, the `required` rule means that rule must pass and remaining rules must also be processed.
  - Previously, `pam_authz` supported two modes, the `netgroup` mode, where netgroups were specified in the `/etc/passwd` file, or the `pam_authz.policy` mode, where rules were defined in the `pam_authz.policy` file. Those two modes were mutually exclusive. A new condition rule in the `pam_authz.policy` file now allows both modes.

- **LDAP Host management tools**

  LDAP-UX Integration B.05.00 supports two new LDAP command-line tools, `ldaphostmgr` and `ldaphostlist`, that allow you to manage information about hosts in the directory server, including ssh public keys. Using HP Secure Shell version 5.5 or higher, LDAP-UX ssh key management can pre-establish trust between hosts.

  - **`ldaphostmgr`**

    Use the `ldaphostmgr` tool to add, modify, or delete information about hosts (OS instances) that are part of the organization. The `ldaphostmgr` tool uses the existing `ldapux(5)` configuration, requiring only a minimal number of command-line options to discover where to search for host information, such as what directory server(s) to contact and proper search filters for finding hosts. It also uses the existing `ldapux(5)` authentication configuration to determine how to bind to the LDAP directory server. `ldaphostmgr` can be used to centrally manage ssh public keys for hosts, and supports attribute-mapping for attributes defined by the ipHost objectclass. Additional attributes used in a host entry (such as owner, entityRole, and so on) are not mapped.

— **`ldaphostlist`**

Use the `ldaphostlist` tool to display and enumerate host entries that reside in an LDAP-based directory server. Although `ldaphostlist` provides output similar to the `ldapsearch` command, it satisfies a few specific feature requirements that allow applications to discover and evaluate hosts stored in an LDAP directory server without requiring intimate knowledge of the methods used to retrieve and evaluate that information in the LDAP directory server. In addition, `ldaphostlist` can be used to discover expiration information about ssh host keys if that information is managed in the directory server.

For detailed information about tool usage, syntax, options, environment variables and return codes supported by these tools, refer to the *LDAP-UX Client Services B.05.00 Administrator's Guide* or manpages *ldaphostmgr*(1M) and *ldaphostlist*(1M).

- **The ignore option for PAM_LDAP support**

  If PAM_LDAP is configured to be the first service module in the `/etc/pam.conf` file (a typical configuration in the Trusted Mode Environment), then when you lose access to your directory server, you will have trouble accessing the system unless a set of so-called "recovery users" is configured in the `/etc/pam_user.conf` file. This release supports the `ignore` option for PAM_LDAP, which enables PAM_LDAP to be completely disregarded for specific local users.

  To enable this feature, you must set the `ignore` option for PAM_LDAP in the `pam_user.conf` file for per-user configuration. When you use this option for PAM_LDAP, PAM returns PAM_IGNORE. For detailed information on how to configure and use this feature, refer to the *LDAP-UX Client Services B.05.00 Administrator's Guide*.

- **`proxy_is_restricted` and `allowed_attribute` flags added to configuration file**

  The `proxy_is_restricted` and `allowed_attribute` flags are added to the [general] section of the configuration file, `ldapclientd.conf`:

  — `proxy_is_restricted=yes|no`

  If the proxy user is configured in the LDAP-UX profile and defined in /etc/opt/ldapux/pcred, this flag attests that the proxy user does not hold privileged LDAP credentials, meaning the proxy user is restricted in its rights to access "private" information in the directory server.

  — `allowed_attribute=`*`service:attribute`*

  Some applications, like `/opt/ssh/bin/ssh`, use `ldapclientd` to access information in the directory server, such as the sshPublicKey for users and hosts. By setting `allowed_attribute`, applications can access any defined attribute even if the `proxy_is_restricted` value is set to `no`(the default).

  These configuration parameters are required to help the `ldaphostlist` and `ldapuglist` tools determine if it is OK for them to display arbitrary attributes. If you used autosetup to configure LDAP-UX, these values are automatically set. If you have an existing installation or use the custom install `setup` program, and are also using a proxy user, you should update these values.

10.3 Related information

You can download the latest version of this document from the following website:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX LDAP-UX Integration Software**.

The following and related documents are also available from the same site:

- *LDAP-UX Client Services B.05.00 with Microsoft Windows Active Directory Server Administrator's Guide*
- *LDAP-UX Integration B.05.00 Release Notes*

For more information about LDAP-UX Integration and related products and solutions, visit the following HP website:

http://h71028.www7.hp.com/enterprise/us/en/os/hpux11i-security-components.html

10.4 Typographic conventions

This document uses the following typographical conventions:

Book Title	Title of a book or other document.
http://www.hp.com	A website address that is a hyperlink to the site.
Emphasis	Text that is emphasized.
Bold	Text that is strongly emphasized. The defined use of an important word or phrase.
`Command`	Command name or qualified command phrase.
`user input`	Commands and other text that you type.
`computer output`	Text displayed by the computer. Name of a daemon, parameter, or parameter option.
`variable`	The name of an environment variable, for example `PATH` or `errno`.
`value`	A value that you may replace in a command or function, or information in a display that represents several possible values.
[ ]	The contents are optional in formats and command descriptions.
{ }	The contents are required in formats and command descriptions.
|	Separates items in a list of choices. In the following example, you must specify either item-a or item-b: {item-a | item-b}
\	The continuous line symbol.
find(1)	HP-UX manpage. In this example, "find" is the manpage name and "1" is the manpage section.
Enter	The name of a keyboard key. Note that **Return** and **Enter** both refer to the same key. A sequence such as **Ctrl+A** indicates that you must hold down the key labeled **Ctrl** while pressing the **A** key.

A Configuration worksheet

Use the worksheet shown in Table A-1 to help you configure LDAP-UX Client Services. See “Installing and configuring LDAP-UX Client Services” (page 21) for details.

Table A-1 LDAP-UX Client Services configuration worksheet

LDAP-UX Client Services Configuration Worksheet	
Directory administrator DN:	
Directory server host:	
Directory server port:	
Configuration profile DN:	
Base DN of name service data:	
Credential type:	
Proxy user DN:	
Source of user, group data:	
Migration method:	

See Table A-2 for explanations and examples. For installation and configuration details, see “Installing and configuring LDAP-UX Client Services” (page 21).

Table A-2 LDAP-UX Client Services configuration worksheet explanation

LDAP-UX Client Services Configuration Worksheet	
Directory administrator DN:	The distinguished name of a directory administrator allowed to modify the directory. Example: cn=directory manager
Directory server host:	The host name or IP address where your directory server is running. Example: sys001.hp.com (12.34.56.78)
Directory server port:	The TCP port number your directory server is using. Example: 389
Configuration profile DN:	The distinguished name where your configuration profile is. Example: cn=profile1, o=hp.com
Base DN of name service data:	The distinguished name where your name service data is. Example: ou=People, o=hp.com
Credential type:	The method clients use to access the directory. Can be "anonymous," "proxy," or "proxy anonymous." Example: anonymous Default: anonymous
Proxy user DN:	The distinguished name of the proxy user, if needed. Example: cn=proxyuser,ou=special users, o=hp.com
Source of user, group data:	Where you get your user and group data from to migrate into the directory. Example: /etc/passwd and /etc/group on sys001
Migration method:	How you will migrate your user and group data into the directory, for example, using the migration scripts. Example: migrate_all_online.sh edited to remove all but migrate_passwd.pl, migrate_group.pl, and migrate_base.pl

B LDAP-UX Client Services object classes

This Appendix describes the object classes LDAP-UX Client Services uses for configuration profiles.

In release B.02.00, LDAP-UX Client Services used two object classes for configuration profiles:

1. posixDUAProfile
2. posixNamingProfile

With release B.03.00, the posixDUAProfile and posixNamingProfile object classes have been replaced by a single STRUCTURAL objectclass DUAConfigProfile.

In addition, four new attributes are added. These changes are to reflect the definition shown in the most current IETF draft "A Configuration Schema for LDAP Based Directory User Agents" (in the document file titled, draft-joslin-config-schema-07.txt). This allows LDAP-UX to integrate with configuration profiles that are supported by other vendors.

The object class `DUAConfigProfile` is defined as follows:

```
objectclass DUAConfigProfile
     superior top
     requires
          cn
     allows
          authenticationMethod,
          attributeMap,
          bindTimeLimit,
          credentialLevel,
          defaultSearchBase,
          defaultSearchScope,
          defaultServerList,
          followReferrals,
          objectclassMap,
          preferredServerList,
          profileTTL,
          searchTimeLimit,
          serviceAuthenticationMethod,
          serviceCredentialLevel,
          servicesearchDescriptor
```

B.1 Profile attributes

The attributes of DUAConfigProfile is defined as follows:

`cn` is the common name of the profile entry.

`attributeMap` is a mapping from RFC 2307 attributes to alternate attributes. Use this if your entries do not conform to RFC 2307. Each entry consists of: *`Service`*`:`*`Attribute`*`=`*`Altattribute`* where *`Service`* is one of the supported services: passwd, group, shadow, pam, networks, hosts, protocols, services, rpc, or netgroup. *`Attribute`* is an attribute of the service as defined by RFC 2307. *`Altattribute`* is the attribute that should be used instead of the standard attribute.

For example, pam:userPassword=ntUserPassword maps the userPassword attribute to ntUserPassword for the pam service. passwd:uidnumber=employeeNumber maps the uidnumber attribute to employeeNumber for the passwd service.

NOTE: The userPassword attribute is mapped to *NULL* to prevent passwords from being returned for increased security and to prevent PAM_UNIX from authenticating users in the LDAP directory. Mapping to *NULL* or any other nonexistent attribute means do not return anything.

`authenticationMethod` is how the client binds to the directory. The value can be "simple" indicating bind using a user name and password. If this attribute has no value, "simple" is the default.

`bindTimeLimit` is how long, in seconds, the client should wait to bind before aborting. 0 (zero) means no time limit. If this attribute has no value, the default is no time limit.

`credentialLevel` is the identity clients use when binding to the directory. The value must be one of the following: "proxy", "anonymous", or "proxy anonymous". "proxy" means use the configured proxy user. "anonymous" means use anonymous access. "proxy anonymous" means use the configured proxy user and if that fails, bind anonymously. If this attribute has no value, "anonymous" is the default.

`defaultSearchBase` is the base DN where clients can find name service information, for example `ou=hpusers,o=hp.com`. This attribute must have a value.

`defaultServerList` is a list of one or more host IP addresses and optional port numbers where LDAP directory servers are running. Each host is searched in the order given. The LDAP-UX client searches the servers until it finds one that responds, defaultServerList is used only if the preferedServerList attribute has no value, or if none of the specified servers in preferredServerList responds the client request. If neither defaultServerList nor preferredServerList specifies a host, the LDAP-UX client does not try to connect to any LDAP directory server. See preferredServerList below.

For example, `15.10.120.150:300` is the host at IP address 15.10.120.150 using port number 300. When specifying multiple hosts, each host:port entry must be separated by a space.

`followReferrals` specifies whether or not referrals should be followed. If the entry is 0 (zero) or FALSE, referrals will not be followed. If the attribute has no value, any other numeric value, or TRUE referrals will be followed.

`preferredServerList` is a list of one or more host IP addresses and optional port numbers where LDAP directory servers are running. Each host is searched in the order given. If this attribute has no value, or if none of the specified servers satisfies the client's request, the defaultServerList is used. See defaultServerList for more information.

For example, `15.13.128.145:250` is the host at IP address 15.13.128.145 using port number 250. When specifying multiple hosts, each host:port entry must be separated by a space.

`profileTTL` is the recommended time interval before refreshing the cached configuration profile.

`searchTimeLimit`

is how long, in seconds, a client should wait for directory searches before aborting. 0 (zero) means no time limit. If this attribute has no value, the default is no time limit.

`serviceSearchDescriptor`

is one to three custom search descriptors for each service. The format is *`Service`*`:`*`BaseDN`*`?`*`Scope`*`?(`*`Filter`*`)` where *`Service`* is one of the supported services passwd, group, shadow, or pam. *`BaseDN`* is the base DN at which to start searches. *`Scope`* is the search scope and can be one of the following: one, base, sub. *`Filter`* is an LDAP search filter, typically the object class. Each service can have up to three custom search descriptors.

For example, the following defines a search descriptor for the passwd service specifying a baseDN of `ou=people,ou=unix,o=hp.com`, a search scope of sub, and a search filter of the posixAccount object class.

```
passwd:ou=people,ou=unix,o=hp.com?sub?(objectclass=posixAccount)
```

C Sample /etc/pam.ldap.trusted file configured by setup

This appendix provides the sample PAM configuration file, `/etc/pam.ldap.trusted` generated by `setup` and used as the `/etc/pam.conf` file to support the coexistence of LDAP-UX and Trusted Mode. This `/etc/pam.ldap.trusted` file must be used as the `/etc/pam.conf` file if your directory server is the HP-UX Directory Server or Redhat Directory Server and your LDAP client is in the Trusted Mode. If your system is in a Standard Mode, you still need to use the`/etc/pam.ldap` file as the `/etc/pam.conf` file.

The following is a sample PAM configuration file, `/etc/pam.ldap.trusted`, used for the HP-UX 11i v2 system:

```
#
# PAM configuration
#
# This pam.conf file is intended as an example only.
#
#

#####################################################################
# This configuration file has only been modified for default   #
# services. Other services can be added or modified as needed  #
# or desired. If a service is not listed, it will use the      #
# OTHER classification.                                        #
#                                                              #
# the format for a entry is                                    #
# <service> <module_type> <control> <module path> <options>    #
#                                                              #
# see pam.conf(4) for more details                             #
#                                                              #
# NOTE: This pam.conf file is recommended only if you convert  #
# your system to a Trusted System. If your system is in the    #
# Standard Mode, use the pam.ldap file as an example.          #
#                                                              #
# NOTE: If the path to a library is not absolute, it is assumed#
# to be relative to the directory /usr/lib/security/$ISA.      #
# The "$ISA (i.e Instruction Set Architecture) token is        #
# replaced by the PAM engine (libpam) with "hpux64" for IA     #
# 64-bit modules, or with "hpux32" for IA 32-bit modules, or   #
# with "pa20_64" for PA 64-bit modules, or with NULL for PA    #
# 32-bit modules.                                              #
# For PA applications, library name ending with "so.1" is a    #
# symbolic link that points to the corresponding PA (32 or 64  #
# bit) backend library.                                        #
#####################################################################
#
# Authentication management
#
login      auth required       libpam_hpsec.so.1
login      auth sufficient     libpam_ldap.so.1
login      auth required       libpam_unix.so.1 try_first_pass
su         auth required       libpam_hpsec.so.1
su         auth sufficient     libpam_ldap.so.1
su         auth required       libpam_unix.so.1 try_first_pass
dtlogin    auth required       libpam_hpsec.so.1
dtlogin    auth sufficient     libpam_ldap.so.1
dtlogin    auth required       libpam_unix.so.1 try_first_pass
dtaction   auth required       libpam_hpsec.so.1
dtaction   auth sufficient     libpam_ldap.so.1
dtaction   auth required       libpam_unix.so.1 try_first_pass
ftp        auth required       libpam_hpsec.so.1
ftp        auth sufficient     libpam_ldap.so.1
ftp        auth required       libpam_unix.so.1 try_first_pass
```

```
rcomds     auth required       libpam_hpsec.so.1
rcomds     auth sufficient     libpam_ldap.so.1
rcomds     auth required       libpam_unix.so.1 try_first_pass
sshd       auth required       libpam_hpsec.so.1
sshd       auth sufficient     libpam_ldap.so.1
sshd       auth required       libpam_unix.so.1 try_first_pass
OTHER      auth sufficient     libpam_ldap.so.1
OTHER      auth required       libpam_unix.so.1 try_first_pass
# Account management
#
login      account required    libpam_hpsec.so.1
login      account sufficient  libpam_ldap.so.1
login      account required    libpam_unix.so.1
su         account required    libpam_hpsec.so.1
su         account sufficient  libpam_ldap.so.1
su         account required    libpam_unix.so.1
dtlogin    account required    libpam_hpsec.so.1
dtlogin    account sufficient  libpam_ldap.so.1
dtlogin    account required    libpam_unix.so.1
dtaction   account required    libpam_hpsec.so.1
dtaction   account sufficient  libpam_ldap.so.1
dtaction   account required    libpam_unix.so.1
ftp        account required    libpam_hpsec.so.1
ftp        account sufficient  libpam_ldap.so.1
ftp        account required    libpam_unix.so.1
rcomds     account required    libpam_hpsec.so.1
rcomds     account sufficient  libpam_ldap.so.1
rcomds     account required    libpam_unix.so.1
sshd       account required    libpam_hpsec.so.1
sshd       account sufficient  libpam_ldap.so.1
sshd       account required    libpam_unix.so.1
ftp        account required    libpam_unix.so.1
OTHER      account sufficient  libpam_ldap.so.1
OTHER      account required    libpam_unix.so.1
# Session management
#
login      session required    libpam_hpsec.so.1
login      session requried    libpam_ldap.so.1
login      session required    libpam_unix.so.1
dtlogin    session required    libpam_hpsec.so.1
dtlogin    session required    libpam_ldap.so.1
dtlogin    session required    libpam_unix.so.1
dtaction   session required    libpam_hpsec.so.1
dtaction   session required    libpam_ldap.so.1
dtaction   session required    libpam_unix.so.1
ftp        session  required   libpam_hpsec.so.1 bypass_limit_login
                                      bypass_umask bypass_nologin
ftp        session requried    libpam_ldap.so.1
ftp        session required    libpam_unix.so.1
rcomds     session required    libpam_hpsec.so.1 bypass_limit_login
rcomds     session required    libpam_ldap.so.1
rcomds     session required    libpam_unix.so.1
sshd       session required    libpam_hpsec.so.1
sshd       session required    libpam_ldap.so.1
sshd       session required    libpam_unix.so.1
OTHER      session required    libpam_ldap.so.1
OTHER      session required    libpam_unix.so.1
# Password management
#
login      password required    libpam_hpsec.so.1
login      password sufficient  libpam_ldap.so.1
login      password required    libpam_unix.so.1 try_first_pass
passwd     password required    libpam_hpsec.so.1
passwd     password sufficient  libpam_ldap.so.1
passwd     password required    libpam_unix.so.1 try_first_pass
```

```
dtlogin    password required    libpam_hpsec.so.1
dtlogin    password sufficient  libpam_ldap.so.1
dtlogin    password required    libpam_unix.so.1 try_first_pass
sshd       password required    libpam_hpsec.so.1
sshd       password sufficient  libpam_ldap.so.1
sshd       password required    libpam_unix.so.1 try_first_pass
OTHER      password sufficient  libpam_ldap.so.1
OTHER      password required    libpam_unix.so.1 try_first_pass
```

D Sample /etc/pam.conf file for security policy enforcement

This appendix provides the sample PAM configuration file, `/etc/pam.conf` file configured to support account and password policy enforcement. In the `/etc/pam.conf` file, the PAM_AUTHZ library must be configured for the `sshd` and `rcommds` services under `account management` role.

The following is a sample PAM configuration file, `/etc/pam.conf`, used on the HP-UX 11i v2 system. You can configure the file as such after it is generated by either `autosetup` or `setup`.

```
#
# PAM configuration
#
# This pam.conf file is intended as an example only.
#
#

#####################################################################
# This configuration file has only been modified for default        #
# services. Other services can be added or modified as needed       #
# or desired. If a service is not listed, it will use the           #
# OTHER classification.                                             #
#                                                                   #
# the format for a entry is                                         #
# <service> <module_type> <control> <module path> <options>         #
#                                                                   #
# see pam.conf (4) for more details                                 #
#                                                                   #
#####################################################################
#
# Authentication management
#
login      auth required        libpam_hpsec.so.1
login      auth sufficient      libpam_unix.so.1
login      auth required        libpam_ldap.so.1 try_first_pass
su         auth required        libpam_hpsec.so.1
su         auth sufficient      libpam_unix.so.1
su         auth required        libpam_ldap.so.1 try_first_pass
dtlogin    auth required        libpam_hpsec.so.1
dtlogin    auth sufficient      libpam_unix.so.1
dtlogin    auth required        libpam_ldap.so.1 try_first_pass
dtaction   auth required        libpam_hpsec.so.1
dtaction   auth sufficient      libpam_unix.so.1
dtaction   auth required        libpam_ldap.so.1 try_first_pass
ftp        auth required        libpam_hpsec.so.1
ftp        auth sufficient      libpam_unix.so.1
ftp        auth required        libpam_ldap.so.1 try_first_pass
rcomds     auth required        libpam_hpsec.so.1
rcomds     auth sufficient      libpam_unix.so.1
rcomds     auth required        libpam_ldap.so.1 try_first_pass
sshd       auth required        libpam_hpsec.so.1
sshd       auth sufficient      libpam_unix.so.1
sshd       auth required        libpam_ldap.so.1 try_first_pass
OTHER      auth sufficient      libpam_unix.so.1
OTHER      auth required        libpam_ldap.so.1 try_first_pass
# Account management
#
login      account required     libpam_hpsec.so.1
login      account required     libpam_authz.so.1
login      account sufficient   libpam_unix.so.1
login      account required     libpam_ldap.so.1
su         account required     libpam_hpsec.so.1
su         account sufficient   libpam_unix.so.1
su         account required     libpam_ldap.so.1
```

```
dtlogin    account required     libpam_hpsec.so.1
dtlogin    account sufficient   libpam_unix.so.1
dtlogin    account required     libpam_ldap.so.1
dtaction   account required     libpam_hpsec.so.1
dtaction   account sufficient   libpam_unix.so.1
dtaction   account required     libpam_ldap.so.1
ftp        account required     libpam_hpsec.so.1
ftp        account sufficient   libpam_ldap.so.1
ftp        account required     libpam_unix.so.1
rcomds     account required     libpam_hpsec.so.1
rcomds     account required     libpam_authz.so.1
rcomds     account sufficient   libpam_unix.so.1
rcomds     account required     libpam_ldap.so.1 rcommand
sshd       account required     libpam_hpsec.so.1
sshd       account required     libpam_authz.so.1
sshd       account sufficient   libpam_unix.so.1
sshd       account required     libpam_ldap.so.1 rcommand
OTHER      account sufficient   libpam_unix.so.1
OTHER      account required     libpam_ldap.so.1
# Session management
#
login      session required     libpam_hpsec.so.1
login      session sufficient   libpam_unix.so.1
login      session required     libpam_ldap.so.1
dtlogin    session required     libpam_hpsec.so.1
dtlogin    session sufficient   libpam_unix.so.1
dtlogin    session required     libpam_ldap.so.1
dtaction   session required     libpam_hpsec.so.1
dtaction   session sufficient   libpam_unix.so.1
dtaction   session required     libpam_ldap.so.1
ftp        session required     libpam_hpsec.so.1 bypass_limit_login
                                bypass_umask bypass_nologin
ftp        session sufficient   libpam_unix.so.1
ftp        session required     libpam_ldap.so.1
rcomds     session required     libpam_hpsec.so.1 bypass_limit_login
rcomds     session sufficient   libpam_unix.so.1
rcomds     session required     libpam_ldap.so.1
sshd       session required     libpam_hpsec.so.1
sshd       session sufficient   libpam_unix.so.1
sshd       session required     libpam_ldap.so.1
OTHER      session sufficient   libpam_unix.so.1
OTHER      session required     libpam_ldap.so.1
# Password management #
login      password required    libpam_hpsec.so.1
login      password sufficient  libpam_unix.so.1
login      password required    libpam_ldap.so..1 try_first_pass
passwd     password required    libpam_hpsec.so.1
passwd     password sufficient  libpam_unix.so.1
passwd     password required    libpam_ldap.so.1 try_first_pass
dtlogin    password required    libpam_hpsec.so.1
dtlogin    password sufficient  libpam_unix.so.1
dtlogin    password required    libpam_ldap.so.1 try_first_pass
sshd       password required    libpam_hpsec.so.1
sshd       password sufficient  libpam_unix.so.1
sshd       password required    libpam_ldap.so.1 try_first_pass
OTHER      password sufficient  libpam_unix.so.1
OTHER      password required    libpam_ldap.so.1 try_first_pass
```

E Samples of LDAP-UX configuration files created or modified by autosetup

The sections in this appendix provide samples of the configuration files modified or created by the autosetup:

- Section E.1: NSS configuration file /etc/nsswitch.conf
- Section E.2: PAM configuration file /etc/pam.conf
- Section E.3: Startup configuration file /etc/opt/ldapux/ldapux_client.conf
- Section E.4: Client daemon configuration file /etc/opt/ldapux/ldapux_client.conf

E.1 NSS configuration file after autosetup configuration

The autosetup script automatically configures the NSS configuration file /etc/nsswitch.conf (in addition to the PAM configuration file /etc/pam.conf file, as documented in Section E.2 (page 359)) to support the LDAP backend. To configure the NSS module, autosetup first determines whether the /etc/nsswitch.conf file exists. If the file does not exist on the system, autosetup creates the /etc/nsswitch.conf file as shown:

```
#
# /etc/nsswitch.conf:
#
# The file is created by autosetup of LDAPUX only if the system does
# not have /etc/nsswitch.conf exist during the autosetup is executing.
#
passwd:        files ldap
group:         files ldap
hosts:         dns [NOTFOUND=return] files ldap
ipnodes:       dns [NOTFOUND=return] files
networks:      files
protocols:     files
rpc:           files
publickey:     files
netgroup:      files
automount:     files
aliases:       files
services:      files
```

E.2 PAM configuration file after autosetup configuration

The autosetup script configures LDAP support by adding, for all services of each service module type (auth, account, session, and password) defined in the /etc/pam.conf file, the PAM_LDAP library object /usr/lib/security/libpam_ldap.so.1 after the line that defines the PAM_UNIX module libpam_unix.so.1 on an HP-UX 11i v2 or v3 system. The following shows the /etc/pam.conf file after it has been modified by autosetup.

```
#
# PAM configuration
#
# Notes:
#
# If the path to a library is not absolute, it is assumed to be
# relative to one of the following directories:
#     /usr/lib/security         (PA 32-bit)
#     /usr/lib/security/pa20_64 (PA 64-bit)
#     /usr/lib/security/hpux32  (IA 32-bit)
#     /usr/lib/security/hpux64  (IA 64-bit)
#
# The IA file name convention is normally used; for example:
#     libpam_unix.so.1
#
# For PA libpam_unix.so.1 is a symbolic link to the PA library:
#     ln -s libpam_unix.1 libpam_unix.so.1
#
# Also note that the use of pam_hpsec(5) is mandatory for some of
# the services. See pam_hpsec(5).
#
# Authentication management
#
```

```
login        auth       required       libpam_hpsec.so.1
login        auth       sufficient     libpam_unix.so.1
login        auth       required       libpam_ldap.so.1 try_first_pass
su           auth       required       libpam_hpsec.so.1 bypass_setaud
su           auth       sufficient     libpam_unix.so.1
su           auth       required       libpam_ldap.so.1 try_first_pass
dtlogin      auth       required       libpam_hpsec.so.1
dtlogin      auth       sufficient     libpam_unix.so.1
dtlogin      auth       required       libpam_ldap.so.1 try_first_pass
dtaction     auth       required       libpam_hpsec.so.1
dtaction     auth       sufficient     libpam_unix.so.1
dtaction     auth       required       libpam_ldap.so.1 try_first_pass
ftp          auth       required       libpam_hpsec.so.1
ftp          auth       sufficient     libpam_unix.so.1
ftp          auth       required       libpam_ldap.so.1 try_first_pass
rcomds       auth       required       libpam_hpsec.so.1
rcomds       auth       sufficient     libpam_unix.so.1
rcomds       auth       required       libpam_ldap.so.1 try_first_pass
sshd         auth       required       libpam_hpsec.so.1
sshd         auth       sufficient     libpam_unix.so.1
sshd         auth       required       libpam_ldap.so.1 try_first_pass
OTHER        auth       required       libpam_hpsec.so.1
OTHER        auth       sufficient     libpam_unix.so.1
OTHER        auth       required       libpam_ldap.so.1 try_first_pass
#
# Account management
#
login        account    required       libpam_hpsec.so.1
login        account    sufficient     libpam_unix.so.1
login        account    required       libpam_ldap.so.1
su           account    required       libpam_hpsec.so.1
su           account    sufficient     libpam_unix.so.1
su           account    required       libpam_ldap.so.1
dtlogin      account    required       libpam_hpsec.so.1
dtlogin      account    sufficient     libpam_unix.so.1
dtlogin      account    required       libpam_ldap.so.1
dtaction     account    required       libpam_hpsec.so.1
dtaction     account    sufficient     libpam_unix.so.1
dtaction     account    required       libpam_ldap.so.1
ftp          account    required       libpam_hpsec.so.1
ftp          account    sufficient     libpam_unix.so.1
ftp          account    required       libpam_ldap.so.1
rcomds       account    required       libpam_hpsec.so.1
rcomds       account    sufficient     libpam_unix.so.1
rcomds       account    required       libpam_ldap.so.1
sshd         account    required       libpam_hpsec.so.1
sshd         account    sufficient     libpam_unix.so.1
sshd         account    required       libpam_ldap.so.1
OTHER        account    required       libpam_hpsec.so.1
OTHER        account    sufficient     libpam_unix.so.1
OTHER        account    required       libpam_ldap.so.1
#
# Session management
#
login        session    required       libpam_hpsec.so.1
login        session    sufficient     libpam_unix.so.1
login        session    required       libpam_ldap.so.1
dtlogin      session    required       libpam_hpsec.so.1
dtlogin      session    sufficient     libpam_unix.so.1
dtlogin      session    required       libpam_ldap.so.1
ftp          session    required       libpam_hpsec.so.1 bypass_limit_login bypass_umask bypass_nologin
ftp          session    sufficient     libpam_unix.so.1
ftp          session    required       libpam_ldap.so.1
rcomds       session    required       libpam_hpsec.so.1 bypass_limit_login
rcomds       session    sufficient     libpam_unix.so.1
rcomds       session    required       libpam_ldap.so.1
sshd         session    required       libpam_hpsec.so.1
sshd         session    sufficient     libpam_unix.so.1
sshd         session    required       libpam_ldap.so.1
OTHER        session    required       libpam_hpsec.so.1
OTHER        session    sufficient     libpam_unix.so.1
OTHER        session    required       libpam_ldap.so.1
#
# Password management
#
login        password   required       libpam_hpsec.so.1
login        password   sufficient     libpam_unix.so.1
login        password   required       libpam_ldap.so.1 try_first_pass
passwd       password   required       libpam_hpsec.so.1
passwd       password   sufficient     libpam_unix.so.1
passwd       password   required       libpam_ldap.so.1 try_first_pass
dtlogin      password   required       libpam_hpsec.so.1
dtlogin      password   sufficient     libpam_unix.so.1
dtlogin      password   required       libpam_ldap.so.1 try_first_pass
sshd         password   required       libpam_hpsec.so.1
sshd         password   sufficient     libpam_unix.so.1
sshd         password   required       libpam_ldap.so.1 try_first_pass
OTHER        password   required       libpam_hpsec.so.1
OTHER        password   sufficient     libpam_unix.so.1
OTHER        password   required       libpam_ldap.so.1 try_first_pass
```

E.3 ldapux_client.conf file after autosetup configuration

The `autosetup` script creates the start-up file `/etc/opt/ldapux/ldapux_client.conf` on the LDAP-UX client system, enabled for TLS support (`enable_startTLS` is set to `1`). The following shows the `ldapux_client.conf` that is configured by `autosetup`.

```
 LDAP-UX Client Services Configuration File
# file name: /etc/opt/ldapux/ldapux_client.conf
#
# This file contains two sections of information.
# The first, the [NSS] section, contains the general configuration
# for the LDAP-UX Client Services product. You can edit the
# configuration file to turn the configuration flags on and off.
# The second, the [profile] section, is generated either from the
# create_profile entry or the setup program.
# If you are an experienced administrator, you may edit this file.
# If the information in this file is not accurate, however, you will
# not be able to retrieve the Configuration Profile entry.
#
# Non-LDAP-UX Integration applications can take advantage of this file
# and the profile management tools. You should add the general
# configuration under the section for your product as was done in the
# [NSS] section, and your application will process the configuration
# under that section.
#
# Your application can call the profile management tools to retrieve
# the profile from the Directory Server and run a specific program to
# your application afterwards.


[NSS]
# This section processes all general configuration flags for LDAP-UX
# Integration.
# To enable logging:
#
#    *uncomment the log_facility and log_level
#    *modify the values if appropriate.
#
# Logging uses the syslog facility. You may have to modify the syslog
# Configuration and signal the syslog daemon to accept the log_facility
# and log_level configured here. See man syslogd(1M) for information on
# using syslog.
#
# LOG_INFO will log only unusual events. LOG_DEBUG logs trace information,
# and will reduce performance and generate large log files on active systems.
#
# options to log_facility: LOG_USER, LOG_MAIL, LOG_DAEMON, LOG_AUTH,
#                          LOG_SYSLOG, LOG_LOCAL0, LOG_LOCAL1,
#                          LOG_LOCAL2, LOG_LOCAL3, LOG_LOCAL4,
#                          LOG_LOCAL5, LOG_LOCAL6, LOG_LOCAL7
#
# options to log_level:    LOG_DEBUG, LOG_INFO

#log_facility=LOG_LOCAL0
#log_level=LOG_INFO
#
#
# You can disable specific users so that they are unable to log in
# through the LDAP server by uncommenting the "disable_uid_range"
# flag and adding the UID numbers you want to disable. For example:
#
#   disable_uid_range=0-100,120,300-400
#
# Note:  The list of UID numbers must be on one line and the maximum
# number of ranges is 20. The system will ignore the typos and white spaces.
#
#disable_uid_range=0

# You can set the user password to be returned as any string (consisting
# of characters from the encrypted password and the "*" character) instead
# of "*" when the password is hidden.  By returning something other than "*"
# for the hidden password, along with a specific pam_ldap configuration,
# r-commands such as rlogin will work with ldap users on the equivalent
# remote host.  Since the password field of each /etc/passwd entry
# contains an "x" when supporting shadow password, the example provided
# below sets the return password to "x".
#
# The default setting is to return "*" for hidden password.
#
# Warning:
```

```
#    Setting the user password to be returned as any string for the hidden
#    password could allow users with active accounts on a remote host to
#    rlogin to the local host on to a disabled account.
#
#password_as="x"

# You can use the following configuration to specify initial Trusted Mode
# auditing for LDAP users. "0" will tell LDAP-UX to set initial auditing
# to be "off" for all LDAP users logging into this HP-UX client system, "1"
# will set initial auditing to be "on". You can change auditing by using
# "audusr -a/-d" (see "audusr" manual page).
#
# Note: Setting "initial_ts_auditing=1" will not enable auditing unless
# you have already started the auditing system, which can be done using
# SAM or "audsys -n" (see "audsys" manual page).
#
#initial_ts_auditing=0

# You can use the following configuration to specify which keytab file to
# use.  If you don't specify a keytab file here, then the default keytab
# file will be used. The default is /etc/krb5.keytab or the one specified
# in /etc/krb5.conf file.
#
# Note: The following line is just an example. If your keytab file for
# LDAP-UX is not /etc/opt/ldapux/ldapux.keytab, you need to replace it
# with the one you want.
#
#kerberos_keytab_file=/etc/opt/ldapux/ldapux.keytab

# To use case insensitive matching for the netgroup service, for the
# innetgr() API, uncomment the line below.
#netgroup_case_ignore=1

# startTLS triggers a TLS negotiation with the communications layer
# of the LDAP Directory Server, allowing channel-level encryption
# for data security purposes.
#
# LDAP-UX performs a startTLS operation to establish TLS connection
# through an unencrypted port such as 389. Please reference to LDAP-UX
# administrator's guide and RFC2830 for more detailed information.
#
# By default startTLS is disabled.  The support of startTLS
# extended operation is enabled when enable_startTLS is set to 1.
# To disable the feature, please set the value of enable_startTLS option
# to 0 or comment out the option.

# PLEASE READ
# Setting enable_startTLS to 1 does not alone configure TLS session
# encryption.  It merely specifies that TLS should be used instead of
# SSL when encryption/validation is required.  Just as with SSL,
# in order to fully enable TLS, the /etc/opt/ldapux/cert8.db must
# contain a CA or LDAP server certificate and TLS/SSL must be enabled in
# the LDAP-UX configuration profile (created by the /opt/ldapux/config/setup
# tool).
#
# Note: In future LDAP-UX releases, TLS will be enabled by default instead of
# SSL for new installations.
#
enable_startTLS=1

# You can use the following configuration to adjust the level of validation
# done of the SSL certificates of LDAP servers.  There are three options
# available for peer_cert_policy:
# WEAK   performs no validation of SSL certificates.
# CERT   is the default and verifies that the issuers of peer SSL certificates
#        are trusted.
# CNCERT performs both the CERT check and also verifies that the common name
#        or subjectAltName values embedded in the certificate matches the
#        address used to connect to the LDAP server, as described in RFC 4513.
#        Please note that LDAP-UX normally stores the IP address of LDAP
#        servers in the configuration profile, and certificates normally
#        embed the host name or fully qualified host name.  Therefore the
#        preferredserverlist setting in the profile may need to be adjusted
#        to address the LDAP server using its host name if this option is
#        used.  Host names may not be used in the profile if the system is
#        configured to use LDAP-UX for host name resolution.  Please see the
#        documentation for details on manually adjusting the profile.
#
#peer_cert_policy=CERT

# LDAP-UX returns group information requested by initgroups(3C), which
```

```
# initializes the user's group access list. The following configuration
# controls if LDAP-UX should return dynamic groups that a user belongs to.
#
# If "enable_dynamic_getgroupsbymember" is set to 1, which is the default,
# LDAP-UX returns both static and dynamic groups that a user belongs to.
# As a result, the user has the access right granted to all those groups.
#
# If "enable_dynamic_getgroupsbymember" is set to 0, LDAP-UX returns only
# static groups that a user belongs to. As a result, the user has only the
# access rights granted to static groups, and does not have the access
# rights granted to dynamic groups.
#
# If you experience an unexpected delay when logging into the system, HP
# recommends that you uncomment the following line and set
# "enable_dynamic_getgroupsbymember" to 0.
#
#enable_dynamic_getgroupsbymember=1

# Prior to B.04.20, LDAP-UX appended the string, "#'*'B" when constructing
# search filters using the attribute uniquemember. Starting from B.04.20,
# this behavior has been turned off.  You may re-enable this feature
# by setting enable_bitstring to 1.  Please refer to "A Summary of
# the X.500(96) User schema for use wwith LDAPv3", RFC2256 as well
# as "Lightweight Directory Access Protocol (v3): Attribute Syntax
# Definitions", RFC2252, for more details on the Name And Optional UID
# syntax.
#
#enable_bitstring=1

# Setting "enable_compat_mode=1" enables LDAP-UX to process "+"/"-"
# entries in /etc/passwd and /etc/group as they are in compat mode
# while "ldap" is still configured as a regular repository for "passwd"
# and "group" in /etc/nsswitch.conf (e.g. /etc/nsswitch.ldap).
#enable_compat_mode=0

[profile]
#This section contains information clients need to access the configuration
#profile entry from an LDAP Directory Server.
#More than one application can share this file.
#For each application,
#the format begins with the keyword "Service:" followed by the service name,
#followed by one or more configuration information lines,
#followed by a line with "$" as the last line of the service,
#followed by another service with the same format if any. For example:
#
#  Service: <service_name>
# <one or more configuration information lines>
# $
#
#  Service: <service_name>
# <one or more configuration information lines>
# $
#
#The name service that LDAP-UX Client Services supports is "NSS".
#For example:
#
#  Service:NSS
#  # More than one 'host:port' can be included in this field,
#  # delimited by ' '. For example:
#  LDAP_HOSTPORT="abc.efg.hp.com def.anywhere.com"
#  # The configuration profile entry name in the Directory Server. For example:
#  PROFILE_ENTRY_DN="cn=myprofile, ou=myorgunit, o=myorg"
#  #The application program the application is to execute after
#  #the configuration profile entry is retrieved from the application.
#  #For example:
#  PROGRAM="/opt/ldapux/config/create_profile_cache"
#  $
Service: NSS
LDAP_HOSTPORT_SSL="16.92.120.190:389"
PROFILE_ENTRY_DN="cn=DOC-ldapuxProfile,ou=Services,ou=Configuration,dc=doc,dc=acme,dc=com"
PROGRAM="/opt/ldapux/config/create_profile_cache"
$
```

E.4 ldapclientd.conf file after autosetup configuration

Before starting the LDAP-UX client daemon process, `autosetup` edits the client daemon configuration file `/etc/opt/ldapux/ldapclientd.conf` to enable the LDAP-UX client daemon `ldapclientd` to launch automatically whenever the system is rebooted and to enable

the directory server to restrict proxy user rights. The following shows the `ldapclientd.conf` that is configured by `autosetup`.

```
#!/sbin/sh
# @(#) $Revision: 1.12 $
# ldap client daemon configuration.
#
# Please note, the below keys are case sensitive.
#
# Example:
#
# [passwd]
# enable=yes
# poscache_ttl=600
# negcache_ttl=600
#
# Note that "TTLs" (time to live) values are in seconds.
# Note that cache sizes are in bytes.
#
[StartOnBoot]
enable=yes

[general]
# If the proxy user is used and defined in /etc/opt/ldapux/pcred, this
# flag indicates if the proxy user does not hold privileged LDAP
# credentials, meaning the proxy user is restricted in it's rights to
# access "private" information in the directory server.  Because
# ldapclientd provides an interface to access arbitrary information
# (attributes), ldapclientd needs to know if the proxy credential has
# more rights that it should.
#
# By default, and if set to zero, ldapclientd assumes the proxy user
# has privledged credentials, and thus will not allow access to attributes
# beyond that of the RFC2307 schema.  However, you can ammend the list of
# allowed attributes using the allowed_attribute paramter defined below.
#
# If proxy_is_restricted is set to 1, then you are attesting that the
# directory server is restricting access to private or other confidential
# information from access by the proxy user.
proxy_is_restricted=1

# Allows the ldapclientd interface to return attributes that are associated
# with RFC2307-based services (such as users and groups), but that those
# attributes are not specifically part of the RFC2307 schema.  Any attribute
# specified below should be considered public information.
allowed_attribute=hosts:sshPublicKey
allowed_attribute=passwd:sshPublicKey

# Maximum number of connections ldapclientd can establish to
# the directory server (or multiple servers when in a multi-domain
# environment).
#
max_conn=100

#
# Time between an inactive connection to the directory server is
# brought down and cleaned up.
#
connection_ttl=300

#
# Number of threads in ldapclientd.
#
num_threads=10

#
# Time to clean up socket files created by client applications that
# were terminated abnormally.
#
socket_cleanup_time=300

#
# Interval between how often ldapclientd identifies and cleans up
# stale cache entries.
#
cache_cleanup_time=10

#
# How often ldapclientd should re-read the ldapux-clientd.conf file.
#
update_ldapux_conf_time=600
```

```
#
# Maximum number of bytes that should be cached by ldapclientd.
# This value is the maximum upper limit of memory that can be
# used by ldapclientd.  If this limit is reached, new entries are
# not cached, until enough expired entries are freed.
#
cache_size=10000000

#
# A state, a virtual connection between the client and LDAP server,
# is created for the setXXent() request, and stays for the subsequent
# getXXent() requests. If no getXXent() requests are received in the
# specified time interval (seconds), the state will be removed.
state_dump_time=300

#
# Maximum number of states ldapclientd allows. "States" are the number
# of enumerations ldapclientd will handle simultaneously. This number
# must be less than max_conn and it is configured as % of max_conn.
#
max_enumeration_states=80%

#
# How often ldapclientd should re-build the compat information to
# reflect changes of "+/-" entries in /etc/passwd and /etc/group, as
# well as changes of netgroup configuration.
# The default value is 86400 seconds (1 day), the allowed range is
# from 600 seconds (10 minutes) to 2592000 seconds (30 days).
#
flush_compat_info_time=86400

#
[passwd]
enable=yes

[group]
enable=yes

[netgroup]
enable=yes

[uiddn]
enable=yes

[domain_pwd]
enable=yes

[domain_grp]
enable=yes

[automount]
enable=yes

[automountmap]
enable=yes

[dynamic_group]
# "dynamic_group" has its own default cache_size, poscache_ttl and negcache_ttl.
cache_size=10000000
enable=yes
poscache_ttl=43200
negcache_ttl=43200

[longterm_cache]
#    Should long term cache enabled ?
# enable=no
#    How long before data is considered stale and not usabled. 1,209600 = 2 weeks.
# longterm_expired_interval=1209600
#    How frequently should save long term data to permanemt storage. 900 = 15 minutes.
# longterm_cache_backup_interval=900
#    How much memory to allocate fpr the long term cache, which stores user and group
#    information. This cache is only used by the working set of users and groups. The
#    working set means any user or group being used or displayed on the system. If you
#     have enumerous large groups with enumerous members, this value should be at least
#     twice as large as the combined size of all those groups.
# longterm_cache_size=50000000
#     Should long term caching support enumeration of users and groups. If getpwent()
#     and getgrent() are not required, this can be disabled.
# longterm_enum_enable=no
#     How frequently should the HP-UX client go to the directory server to refresh the
```

```
#     enumeration cache. 86400 = one per day.
# longterm_enum_search_interval=86400
#

#enable=no
#longterm_expired_interval=1209600
#longterm_cache_backup_interval=900
#longterm_cache_size=50000000
#longterm_enum_enable=no
#longterm_enum_search_interval=86400

[printers]
#  Define the status of the printer configurator when ldapclientd starts.
#  Option "yes" means the printer configurator service will be activated
#  when ldapclientd starts. "no" means the printer configurator will be
#  disabled when ldapclientd starts. Default is "yes".
start=yes

# Define the maximum printer objects that the printer configurator service
# will handle. The value must be greater than 0.
# Default value is 50.
max_printers=50

# Define the interval, in seconds, before the printer configurator service
# searches for printer objects. The minimum value is 1800 (30 minutes) and the
# maximum value is 1209600 (14 day). Default value is 86440 seconds.
search_interval=86400

# User defined lpadmin options. If the lpadmin_option field is empty or the
# lpadmin_option is commended out, the default lpadmin options are used.
#
# "-mrmodel -v/dev/null -ocmrcmodel -osmrsmodel"
#
# Please DO NOT INCLUDE the -p -orm -orp options in the option field.
# The required information of printer name (-p), remote machine name (-orm) and
# remote printer name (-orp) will be provided by printer configurator during
# the run time.
#
# To enable the user define lpadmin options,
# remove the followng # sign and customize the lpadmin options.
# lpadmin_option=-mrmodel -v/dev/null -ocmrcmodel -osmrsmodel
```

Glossary

Access Control Instruction	A specification controlling access to entries in a directory.
Access Control List	One or more ACIs.
ACI	*See* See Access Control Instruction.
Configuration profile	An entry in an LDAP directory containing information common to many clients, that allows clients to access user, group and other information in the directory. Clients download the profile from the directory. *See also* See also Client Configuration File..
DIGEST-MD5	Message Digest version 5. It is a one-way hash function and always generates 20 bytes of output from text data.
domain	*See* LDAP-UX domain.
IETF	Internet Engineering Task Force; the organization that defines the LDAP specification. See the IETF website at http://www.ietf.org.
LDAP	*See* See Lightweight Directory Access Protocol.
LDAP Data Interchange Format (LDIF)	The format used to represent directory server entries in text form.
LDAP-UX domain	A collection of users, groups and hosts that are managed in the LDAP directory server and are defined by the LDAP-UX configuration profile. All hosts configured to point to the same LDAP-UX configuration profile are considered part of that domain. Not to be confused with Windows domains, the directory server administration domain, or a DNS domain.
LDIF	*See* See LDAP Data Interchange Format.
Lightweight Directory Access Protocol (LDAP)	A standard, extensible set of conventions specifying communication between clients and servers across TCP/IP network connections. *See also* See also SLAPD..
Name Service Switch (NSS)	A framework that allows a host to get name information from various sources such as local files in /etc, NIS, NIS+, or an LDAP directory without modifying applications. For more information, see the *switch*(4) manpage.
Network Information Service (NIS)	A distributed database system providing centralized management of common configuration files, such as /etc/passwd and /etc/hosts.
NIS	*See* See Network Information Service.
NSS	*See* See Name Service Switch.
PAM	*See* See Pluggable Authentication Mechanism.
PAM Authorization Service Module	*See* The PAM Authorization Service Module allows the administrator to control which user subgroups of a large repository can login to the system pam_authz5 manpage.
Pluggable Authentication Module (PAM)	A framework that allows different authentication service modules to be made available without modifying applications. For more information, see the See *pam_ldap*(5), *pam*(3), and *pam.conf*(4) manpages.
Profile	*See* See Configuration profile.
RFC	Request for Comments; a document and process of standardization from the IETF.
RFC 2307	The IETF specification for using LDAP as a Network Information Service. See the RFC at the following location: http://www.ietf.org/rfc/rfc2307.txt.
SLAPD	The University of Michigan's stand-alone implementation of LDAP, without the need for an X.500 directory.

Start-up file A text file containing information the client needs to access an LDAP directory and download a configuration profile.
See also See also Configuration profile.configurationstart-up file ldapux_client.confstart-up file ldapux_client.confldapux_client.conf start-up fileclient start-up file ldapux_client.conf.

ypldapd The NIS/LDAP Gateway daemon, part of the NIS/LDAP Gateway subproduct. ypldapd replaces the NIS ypserv daemon by accepting NIS client requests and getting the requested information from an LDAP directory rather than from NIS maps.

See *NIS/LDAP Gateway Administrator's Guide* at:

http://www.hp.com/go/hpux-security-docs

Click **HP-UX LDAP-UX Integration Software**.

Index

Symbols

/etc/group, 59, 65
/etc/nsswitch.conf, 63, 69
/etc/nsswitch.ldap, 63, 211
/etc/opt/ldapux/acred, 216
/etc/opt/ldapux/pcred, 216
/etc/pam.conf, 69
/etc/pam.ldap, 211
/etc/passwd, 59, 65

A

access control instruction (ACI), 65, 66, 85, 367
  within LDAP-UX domain, 28
access control instruction (ACL)
  in LDAP-UX domain, 29
access control list (ACL)
  in LDAP-UX domain, 29
access control rights
  within LDAP-UX domain, 34
access log, 190
access policy file, 143
acred file, 216
add directory replica, 158
Adding Users, 164
administration groups
  access control rights
    within LDAP-UX domain, 34
administrators
  in the LDAP-UX environment, 38
anonymous access, 18, 61, 183, 184, 192
attribute, 65, 66, 183, 184, 192, 349
  remap, 59
attributeMap, 349
authentication, 16, 66, 191
authenticationMethod, 350
AutoFS support
  configuring, 95
autosetup, 23, 27, 211
  (*see also* guided installation)
  command examples, 43
  command options, 39
  command syntax and options, 38
  environment variables, 41
  reconfiguring LDAP-UX with, 54
  silent mode, 40
  what it does, 25

B

base DN, 22, 71
beq search tool, 92, 212
bind to directory, 18, 61, 74
bindTimeLimit, 350
boot, 64

C

CA certificate
  created by guided installation, 35
  depot file, 35
certificate database files, 80
  (*see also* security database files)
  created by autosetup, 26
  created by certutil, 80
Certificate Database Tool (*see* certutil)
certificates
  created by guided installation, 35
certutil
  creating security database files, 80
certutil tool, 212
change client's access method, 183, 184
change client's profile, 183
change passwords, 294, 335
change personal information, 336
chfn, 336
chsh, 336
ciphers
  SSL/TLS, 82
client administration tools
  ldappasswd, 294
client management tools, 214
client start-up file ldapux_client.conf, 18, 183, 211
cn, 65, 66, 192, 349
commands supported, 17
components, 211, 213, 337, 338, 340
configuration
  client, 68
  custom, 73
  customized
    summary, 57
  directory, 65
  guided
    summary, 23
  quick, 69
  start-up file ldapux_client.conf, 18, 183, 211
  subsequent clients, 112
  worksheet, 21, 347
Configuration Administrator, 38, 46
configuration files
  configured by autosetup, 359
configuration profile, 18, 60, 61, 70, 191, 367
  attributes, 349
  changing a client's, 183
  changing access, 183, 184
  creating, 183
  displaying, 182
  guided installation, 30
  location, 211
  modifying, 183
  object classes, 349
  within LDAP-UX domain, 29

create profile, 183
create proxy user, 182
create_profile_cache program, 214
create_profile_entry program, 214
create_profile_schema program, 215
credential caching (*see* offline credential caching)
credentialLevel, 183, 184, 350
custom configuration, 73
customized installation, 56

D

debugging, 189
default template files, 243
defaultSearchBase, 350
defaultServerList, 350
defining template files, 244
Digest-MD5, 62
directory
  access log, 190
  add replica, 158
  bind, 18, 61, 74
  configuration, 65
  error log, 190
  index entries, 66
  LDAP, 16, 21
  log files, 190
  replica, 335
  tools, 212, 337, 338, 339, 340
  white paper, 21, 65, 67, 185
Directory Administrator, 38
directory information tree
  within LDAP-UX domain, 27, 28
Directory Manager, 38, 45
directory server administration domain, 23, 38
display profile, 182
display proxy user DN, 182
display_profile_cache program, 215, 332, 333
DIT (*see* directory information tree)
domain
  directory server administration, 38
  directory server management, 23, 38
  LDAP-UX domain, 27
  various types of, 36
DomainAdmins group
  access control rights
    within LDAP-UX domain, 34
dtlogin, 17
dynamic variable
  access rule, 151
  support, 145

E

enumeration requests, 185
error log, 190

F

finger, 17, 185
followReferrals, 350
ftp, 17

G

gecos, 192
get_profile_entry program, 215
getgrent, 17, 185
gethostent, 185
getnetent, 185
getpwent, 17, 185
gidnumber, 65, 66, 192
grget, 17, 92, 185
group data, 59, 90
  base DN, 71
groups, 17, 185
  administration
    within LDAP-UX domain, 34
  DomainAdmins
    within LDAP-UX domain, 34
  HostAdmins
    within LDAP-UX domain, 34
  UserAdmins
    within LDAP-UX domain, 34
guided installation, 23
  autosetup command syntax, 38
  command examples, 43
  command options, 39
  configuration profile, 30
  creating a new directory server, 44
  environment variables, 41
  Existing Directory Server mode, 50
  Existing LDAP-UX Domain Installation mode, 53
  into existing LDAP-UX domain, 53
  LDAP-UX domain, 27
  New Directory Server mode, 44
  provisioning an existing directory server, 50
  reconfiguring LDAP-UX with, 54
  silent mode, 40
  what it does, 25

H

homedirectory, 65, 192
host keys, 193
HostAdmins group
  access control rights
    within LDAP-UX domain, 34
HP-UX Directory Server, 21

I

id, 17
IETF, 65, 367
import data into directory, 90, 326
improving performance, 185
index directory entries, 66
installation, 64
  customized
    planning, 59
    summary, 56, 57
  customizeded
    benefits, 22
  guided
    benefits, 22

planning, 23
summary, 23

L

LDAP, 367
LDAP directory, 16, 21
LDAP UG Tool configuration file, 241
LDAP-UX
configuration files
configured by autosetup, 359
PAM configuration file
sample for security policy enforcement, 357
PAM configuration file (trusted)
configured by autosetup, 353
LDAP-UX domain
access control instruction (ACI), 29
access control list (ACL), 29
administrator, 38
administrator name, 42, 46
configuration profile, 29
directory information tree, 27, 28
information model, 28, 29
name, 46, 52, 55
overview, 27
schema, 31
security framework, 28, 33
LDAP-UX Domain Administrator, 38
ldap_proxy_config program, 184
ldap_proxy_config tool, 216
ldapcfinfo command, 92
ldapclientd.conf file
configured by autosetup, 363
ldapdelete program, 297, 332
ldapentry, 292
ldaphostlist tool, 277
ldaphostmgr tool, 266
ldapmodify program, 296, 336
ldappasswd program, 294, 335
ldapsearch program, 295
ldapugadd, 159, 232
ldapugdel, 261
ldapuglist, 159, 223
ldapugmod, 159, 250
ldapux_client.conf start-up file, 18, 183, 211
ldapux_client.conf startup file
configured by autosetup, 361
LDIF, 367
LDIF file, 59, 90, 211
listusers, 17, 92, 185
logging
HP-UX Directory Server, 190
LDAP-UX, 189
PAM, 189
login, 17
login authorization, 63
logins, 17, 92, 185
loginshell, 192
logname, 17
ls, 17, 92

M

map posix attributes, 59
memberuid, 65, 66, 192
migrate
data into directory, 90
migrate NIS maps, 326
migration tools, 59, 326
modify profile, 183

N

name service, 16, 63
naming context
default, 326
migration scripts, 326
NativeLdapClient subproduct, 64
newgrp, 17
NIS, 15, 59, 61, 90
migrate maps, 326
NIS/LDAP Gateway, 90
nsquery, 17, 92, 190
NSS, 16, 63, 69, 211, 367
NSS configuration file
configured by autosetup, 359
nsswitch.conf file
configured by autosetup, 359

O

o=hp.com, 22, 61
object class
posixAccount, 65
posixDUAProfile, 70, 349
posixGroup, 65
posixNamingProfile, 70, 349
objectclass, 66
offline credential caching, 102
overview, 15
owners
access control rights
within LDAP-UX domain, 35

P

PAM, 16, 69, 211, 367
PAM configuration file
configured by autosetup, 359
sample for security policy enforcement, 357
PAM configuration file (trusted)
configured by setup, 353
pam.conf
configured by setup, 353
sample for security policy enforcement, 357
pam.conf file, 109
configured by autosetup, 359
pam.ldap.trusted file
configured by setup, 353
PAM_AUTHZ
login authorization, 140
security policy enforcement, 142
PAM_AUTHZ authentication, 57
PAM_AUTHZ library, 140

PAM_UPDBE library (libpam_updbe)
configuring, 110
pam_user.conf file, 109
passwd, 17, 335
password, change, 294, 335
pcred file, 216
securing, 182
performance, 185
perl, 212, 326
pin file
created by guided installation, 36
planning your environment
customized installation, 59
posix schema RFC 2307, 59, 65, 70, 367
posixAccount object class, 65
posixDUAProfile object class, 70, 349
posixGroup object class, 65
posixNamingProfile object class, 70, 349
preferredServerList, 350
product components, 211, 213, 337, 338, 340
Profile TTL, 71
profile, configuration, 18, 60, 61, 70, 191, 367
attributes, 349
changing a client's, 183
changing access, 183, 184
creating, 183
displaying, 182
location, 211
modifying, 183
object classes, 349
PROFILE_ENTRY_DN, 183
profileTTL, 350
proxy user, 18, 61, 66, 74, 183, 184, 191, 192, 216
access permissions, 66
creating, 182
displaying DN, 182
recreating after inadvertent removal
using autosetup, 54
verifying, 182
within LDAP-UX domain, 33
pwget, 17, 92, 185

Q

quick configuration, 69

R

r-commands, 153
reboot, 64
reconfiguring LDAP-UX
with autosetup (guided installation), 54
referral, 74
remap posix attributes, 59
remsh, 17
replica, 335
replica directory, 158
RFC 2307 posix schema, 59, 65, 367
rlogin, 17
root login, 59, 60

S

schema
within LDAP-UX domain, 29, 31
schema, posix, RFC 2307, 59, 65, 70, 367
search time limit, 74
searchTimeLimit, 351
secure shell (ssh), 153
managing host keys with LDAP-UX, 193
Secure Socket Layer (*see* SSL)
security
within LDAP-UX domain, 33
security database files
created by autosetup, 26
creating, 80
self
access control rights
within LDAP-UX domain, 35
server certificate
created by guided installation, 35
service and support, 343
serviceSearchDescriptor, 351
setup, 56, 57, 69, 183, 211
ssh host keys, 193
SSL
configuring, 79
within LDAP-UX domain, 35
start-up file ldapux_client.conf, 18, 183, 211
su, 17
subproduct, NativeLdapClient, 64
supported commands, 17
swinstall, 64
syslog daemon, 189, 190

T

telnet, 17
template file naming, 242
template files, 242
testing clients, 92
time limit on searches, 74
TLS
configuring, 79
within LDAP-UX domain, 35
tools
client management, 214
create_profile_cache, 214
create_profile_entry, 214
create_profile_schema, 215
directory, 212, 337, 338, 339, 340
display_profile_cache, 215, 332, 333
get_profile_entry, 215
ldap_proxy_config, 216
ldapdelete, 297, 332
ldaphostlist, 277
ldaphostmgr, 266
ldapmodify, 296, 336
ldappasswd, 294, 335
ldapsearch, 295
migration, 326
perl, 212

Transport Layer Security (*see* SSL)
troubleshooting, 189
  directory logging, 190
  LDAP-UX logging, 189
  PAM logging, 189
  SSL/TLS ciphers, 82
  syslog, 189, 190
  user cannot log in, 190
TTL, profile, 71, 350
typographic conventions, 346

U

uid, 65, 66, 192
uidnumber, 65, 66, 192
user and group management tools, 219
user cannot log in, 190
user data, 59, 90
  base DN, 71
UserAdmins group
  access control rights
    within LDAP-UX domain, 34
userpassword, 65
users, 64
  access control rights
    within LDAP-UX domain, 35

V

verify configuration, 92
verify proxy user, 182

W

white paper, directory configuration, 21, 65, 67, 185
who, 17
whoami, 17
worksheet, configuration, 21, 347