



Red Hat

Red Hat Directory Server 11

Configuration, Command, and File Reference

Reference guide for configuring Directory Server

Red Hat Directory Server 11 Configuration, Command, and File Reference

Reference guide for configuring Directory Server

Legal Notice

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This is a reference of configuration parameter, the server schema, files, and command-line utilities for Red Hat Directory Server.

Table of Contents

PREFACE	28
LEGAL NOTICE	28
AUTHORS	28
MAKING OPEN SOURCE MORE INCLUSIVE	30
ABOUT THIS REFERENCE	31
1. DIRECTORY SERVER OVERVIEW	31
CHAPTER 1. INTRODUCTION	32
1.1. DIRECTORY SERVER CONFIGURATION	32
1.2. DIRECTORY SERVER INSTANCE FILE REFERENCE	32
1.3. USING DIRECTORY SERVER COMMAND-LINE UTILITIES	32
CHAPTER 2. FILE LOCATIONS OVERVIEW	33
2.1. DIRECTORY SERVER INSTANCE-INDEPENDENT FILES AND DIRECTORIES	33
2.2. DIRECTORY SERVER INSTANCE-SPECIFIC FILES AND DIRECTORIES	33
2.2.1. Configuration Files	34
2.2.1.1. Overview of the Directory Server Configuration	34
2.2.1.1.1. LDIF and Schema Configuration Files	35
2.2.1.1.2. How the Server Configuration Is Organized	37
2.2.1.2. Accessing and Modifying Server Configuration	38
2.2.1.2.1. Access Control for Configuration Entries	38
2.2.1.2.2. Changing Configuration Attributes	39
2.2.2. Database Files	41
2.2.3. LDIF Files	42
2.2.4. Lock Files	43
2.2.5. Log Files	43
2.2.6. PID Files	43
2.2.7. Backup Files	43
2.3. ADMINISTRATION SERVER FILES AND DIRECTORIES	44
CHAPTER 3. CORE SERVER CONFIGURATION REFERENCE	45
3.1. CORE SERVER CONFIGURATION ATTRIBUTES REFERENCE	45
3.1.1. cn=config	45
3.1.1.1. nsslapd-accesslog (Access Log)	45
3.1.1.2. nsslapd-accesslog-level (Access Log Level)	46
3.1.1.3. nsslapd-accesslog-list (List of Access Log Files)	47
3.1.1.4. nsslapd-accesslog-logbuffering (Log Buffering)	47
3.1.1.5. nsslapd-accesslog-logexpirationtime (Access Log Expiration Time)	48
3.1.1.6. nsslapd-accesslog-logexpirationtimeunit (Access Log Expiration Time Unit)	48
3.1.1.7. nsslapd-accesslog-logging-enabled (Access Log Enable Logging)	49
3.1.1.8. nsslapd-accesslog-logmaxdiskspace (Access Log Maximum Disk Space)	50
3.1.1.9. nsslapd-accesslog-logminfreediskspace (Access Log Minimum Free Disk Space)	50
3.1.1.10. nsslapd-accesslog-logrotationsync-enabled (Access Log Rotation Sync Enabled)	51
3.1.1.11. nsslapd-accesslog-logrotationsynchour (Access Log Rotation Sync Hour)	51
3.1.1.12. nsslapd-accesslog-logrotationsyncmin (Access Log Rotation Sync Minute)	51
3.1.1.13. nsslapd-accesslog-logrotationtime (Access Log Rotation Time)	52
3.1.1.14. nsslapd-accesslog-logrotationtimeunit (Access Log Rotation Time Unit)	52
3.1.1.15. nsslapd-accesslog-maxlogsize (Access Log Maximum Log Size)	53
3.1.1.16. nsslapd-accesslog-maxlogsperdir (Access Log Maximum Number of Log Files)	53
3.1.1.17. nsslapd-accesslog-mode (Access Log File Permission)	54

3.1.1.18. nsslapd-allow-anonymous-access	55
3.1.1.19. nsslapd-allow-hashed-passwords	55
3.1.1.20. nsslapd-allow-unauthenticated-binds	56
3.1.1.21. nsslapd-allowed-sasl-mechanisms	56
3.1.1.22. nsslapd-anonlimitsdn	57
3.1.1.23. nsslapd-attribute-name-exceptions	58
3.1.1.24. nsslapd-auditlog (Audit Log)	58
3.1.1.25. nsslapd-auditlog-list	59
3.1.1.26. nsslapd-auditlog-logexpirationtime (Audit Log Expiration Time)	59
3.1.1.27. nsslapd-auditlog-logexpirationtimeunit (Audit Log Expiration Time Unit)	60
3.1.1.28. nsslapd-auditlog-logging-enabled (Audit Log Enable Logging)	60
3.1.1.29. nsslapd-auditlog-logmaxdiskspace (Audit Log Maximum Disk Space)	61
3.1.1.30. nsslapd-auditlog-logminfreediskspace (Audit Log Minimum Free Disk Space)	61
3.1.1.31. nsslapd-auditlog-logrotationsync-enabled (Audit Log Rotation Sync Enabled)	62
3.1.1.32. nsslapd-auditlog-logrotationsynchour (Audit Log Rotation Sync Hour)	62
3.1.1.33. nsslapd-auditlog-logrotationsyncmin (Audit Log Rotation Sync Minute)	63
3.1.1.34. nsslapd-auditlog-logrotationtime (Audit Log Rotation Time)	63
3.1.1.35. nsslapd-auditlog-logrotationtimeunit (Audit Log Rotation Time Unit)	64
3.1.1.36. nsslapd-auditlog-maxlogsize (Audit Log Maximum Log Size)	64
3.1.1.37. nsslapd-auditlog-maxlogsperdir (Audit Log Maximum Number of Log Files)	65
3.1.1.38. nsslapd-auditlog-mode (Audit Log File Permission)	65
3.1.1.39. nsslapd-auditfaillog (Audit Fail Log)	66
3.1.1.40. nsslapd-auditfaillog-list	67
3.1.1.41. nsslapd-auditfaillog-logexpirationtime (Audit Fail Log Expiration Time)	67
3.1.1.42. nsslapd-auditfaillog-logexpirationtimeunit (Audit Fail Log Expiration Time Unit)	67
3.1.1.43. nsslapd-auditfaillog-logging-enabled (Audit Fail Log Enable Logging)	68
3.1.1.44. nsslapd-auditfaillog-logmaxdiskspace (Audit Fail Log Maximum Disk Space)	68
3.1.1.45. nsslapd-auditfaillog-logminfreediskspace (Audit Fail Log Minimum Free Disk Space)	69
3.1.1.46. nsslapd-auditfaillog-logrotationsync-enabled (Audit Fail Log Rotation Sync Enabled)	69
3.1.1.47. nsslapd-auditfaillog-logrotationsynchour (Audit Fail Log Rotation Sync Hour)	69
3.1.1.48. nsslapd-auditfaillog-logrotationsyncmin (Audit Fail Log Rotation Sync Minute)	70
3.1.1.49. nsslapd-auditfaillog-logrotationtime (Audit Fail Log Rotation Time)	70
3.1.1.50. nsslapd-auditfaillog-logrotationtimeunit (Audit Fail Log Rotation Time Unit)	71
3.1.1.51. nsslapd-auditfaillog-maxlogsize (Audit Fail Log Maximum Log Size)	71
3.1.1.52. nsslapd-auditfaillog-maxlogsperdir (Audit Fail Log Maximum Number of Log Files)	72
3.1.1.53. nsslapd-auditfaillog-mode (Audit Fail Log File Permission)	72
3.1.1.54. nsslapd-bakdir (Default Backup Directory)	73
3.1.1.55. nsslapd-certdir (Certificate and Key Database Directory)	73
3.1.1.56. nsslapd-certmap-basedn (Certificate Map Search Base)	74
3.1.1.57. nsslapd-config	74
3.1.1.58. nsslapd-cn-uses-dn-syntax-in-dns	75
3.1.1.59. nsslapd-connection-buffer	75
3.1.1.60. nsslapd-connection-nocanon	76
3.1.1.61. nsslapd-conntablesize	76
3.1.1.62. nsslapd-counters	77
3.1.1.63. nsslapd-csnlogging	77
3.1.1.64. nsslapd-defaultnamingcontext	78
3.1.1.65. nsslapd-disk-monitoring	78
3.1.1.66. nsslapd-disk-monitoring-grace-period	79
3.1.1.67. nsslapd-disk-monitoring-logging-critical	79
3.1.1.68. nsslapd-disk-monitoring-readonly-on-threshold	79
3.1.1.69. nsslapd-disk-monitoring-threshold	80
3.1.1.70. nsslapd-dn-validate-strict	81

3.1.1.71. nsslapd-ds4-compatible-schema	81
3.1.1.72. nsslapd-enable-turbo-mode	81
3.1.1.73. nsslapd-enable-upgrade-hash	82
3.1.1.74. nsslapd-enquote-sup-oc (Enable Superior Object Class Enquoting)	83
3.1.1.75. nsslapd-entryusn-global	83
3.1.1.76. nsslapd-entryusn-import-initval	84
3.1.1.77. nsslapd-errorlog (Error Log)	84
3.1.1.78. nsslapd-errorlog-level (Error Log Level)	85
3.1.1.79. nsslapd-errorlog-list	86
3.1.1.80. nsslapd-errorlog-logexpirationtime (Error Log Expiration Time)	87
3.1.1.81. nsslapd-errorlog-logexpirationtimeunit (Error Log Expiration Time Unit)	87
3.1.1.82. nsslapd-errorlog-logging-enabled (Enable Error Logging)	88
3.1.1.83. nsslapd-errorlog-logmaxdiskspace (Error Log Maximum Disk Space)	88
3.1.1.84. nsslapd-errorlog-logminfreediskspace (Error Log Minimum Free Disk Space)	88
3.1.1.85. nsslapd-errorlog-logrotationsync-enabled (Error Log Rotation Sync Enabled)	89
3.1.1.86. nsslapd-errorlog-logrotationsynchour (Error Log Rotation Sync Hour)	89
3.1.1.87. nsslapd-errorlog-logrotationsyncmin (Error Log Rotation Sync Minute)	90
3.1.1.88. nsslapd-errorlog-logrotationtime (Error Log Rotation Time)	90
3.1.1.89. nsslapd-errorlog-logrotationtimeunit (Error Log Rotation Time Unit)	91
3.1.1.90. nsslapd-errorlog-maxlogsize (Maximum Error Log Size)	91
3.1.1.91. nsslapd-errorlog-maxlogsperdir (Maximum Number of Error Log Files)	92
3.1.1.92. nsslapd-errorlog-mode (Error Log File Permission)	92
3.1.1.93. nsslapd-force-sasl-external	93
3.1.1.94. nsslapd-groupevalnestlevel	94
3.1.1.95. nsslapd-idletimeout (Default Idle Timeout)	94
3.1.1.96. nsslapd-ignore-virtual-attrs	95
3.1.1.97. nsslapd-instancedir (Instance Directory)	95
3.1.1.98. nsslapd-ioblocktimeout (IO Block Time Out)	95
3.1.1.99. nsslapd-lastmod (Track Modification Time)	96
3.1.1.100. nsslapd-ldapautobind (Enable Autobind)	96
3.1.1.101. nsslapd-ldapientrysearchbase (Search Base for LDAP Authentication Entries)	97
3.1.1.102. nsslapd-ldapfilepath (File Location for LDAP Socket)	97
3.1.1.103. nsslapd-ldapgidnumbertype (Attribute Mapping for System GUID Number)	98
3.1.1.104. nsslapd-ldaplisten (Enable LDAP)	98
3.1.1.105. nsslapd-ldapimapprootdn (Autobind Mapping for Root User)	99
3.1.1.106. nsslapd-ldapimapprotoentries (Enable Autobind Mapping for Regular Users)	99
3.1.1.107. nsslapd-ldapuidnumbertype	100
3.1.1.108. nsslapd-ldifdir	100
3.1.1.109. nsslapd-listen-backlog-size	101
3.1.1.110. nsslapd-listenhost (Listen to IP Address)	101
3.1.1.111. nsslapd-localhost (Local Host)	102
3.1.1.112. nsslapd-localuser (Local User)	102
3.1.1.113. nsslapd-lockdir (Server Lock File Directory)	103
3.1.1.114. nsslapd-localssf	103
3.1.1.115. nsslapd-logging-hr-timestamps-enabled (Enable or Disable High-resolution Log Timestamps)	103
3.1.1.116. nsslapd-maxbersize (Maximum Message Size)	104
3.1.1.117. nsslapd-maxdescriptors (Maximum File Descriptors)	104
3.1.1.118. nsslapd-maxsasliosize (Maximum SASL Packet Size)	105
3.1.1.119. nsslapd-maxthreadsperconn (Maximum Threads per Connection)	106
3.1.1.120. nsslapd-minssf	106
3.1.1.121. nsslapd-minssf-exclude-rootdse	107
3.1.1.122. nsslapd-moddn-aci	107
3.1.1.123. nsslapd-malloc-mmap-threshold	108

3.1.1.124. nsslapd-malloc-mxfast	108
3.1.1.125. nsslapd-malloc-trim-threshold	109
3.1.1.126. nsslapd-nagle	109
3.1.1.127. nsslapd-ndn-cache-enabled	110
3.1.1.128. nsslapd-ndn-cache-max-size	110
3.1.1.129. nsslapd-outbound-ldap-io-timeout	111
3.1.1.130. nsslapd-pagedsizelimit (Size Limit for Simple Paged Results Searches)	111
3.1.1.131. nsslapd-plug-in	112
3.1.1.132. nsslapd-plugin-binddn-tracking	112
3.1.1.133. nsslapd-plugin-logging	112
3.1.1.134. nsslapd-port (Port Number)	113
3.1.1.135. nsslapd-privatenamespaces	113
3.1.1.136. nsslapd-pwpolicy-inherit-global (Inherit Global Password Syntax)	114
3.1.1.137. nsslapd-pwpolicy-local (Enable Subtree- and User-Level Password Policy)	114
3.1.1.138. nsslapd-readonly (Read Only)	115
3.1.1.139. nsslapd-referral (Referral)	115
3.1.1.140. nsslapd-referralmode (Referral Mode)	116
3.1.1.141. nsslapd-require-secure-binds	116
3.1.1.142. nsslapd-requiresrestart	117
3.1.1.143. nsslapd-reserveddescriptors (Reserved File Descriptors)	117
3.1.1.144. nsslapd-return-exact-case (Return Exact Case)	118
3.1.1.145. nsslapd-rewrite-rfc1274	119
3.1.1.146. nsslapd-rootdn (Manager DN)	119
3.1.1.147. nsslapd-rootpw (Root Password)	119
3.1.1.148. nsslapd-rootpwstoragescheme (Root Password Storage Scheme)	120
3.1.1.149. nsslapd-rundir	121
3.1.1.150. nsslapd-sasl-mapping-fallback	121
3.1.1.151. nsslapd-sasl-max-buffer-size	122
3.1.1.152. nsslapd-sasxpath	122
3.1.1.153. nsslapd-schema-ignore-trailing-spaces (Ignore Trailing Spaces in Object Class Names)	123
3.1.1.154. nsslapd-schemacheck (Schema Checking)	123
3.1.1.155. nsslapd-schemadir	124
3.1.1.156. nsslapd-schemamod	125
3.1.1.157. nsslapd-schemareplace	125
3.1.1.158. nsslapd-search-return-original-type-switch	125
3.1.1.159. nsslapd-securelistenhost	126
3.1.1.160. nsslapd-securePort (Encrypted Port Number)	126
3.1.1.161. nsslapd-security (Security)	127
3.1.1.162. nsslapd-sizelimit (Size Limit)	127
3.1.1.163. nsslapd-snmp-index	128
3.1.1.164. nsslapd-SSLclientAuth	128
3.1.1.165. nsslapd-ssl-check-hostname (Verify Hostname for Outbound Connections)	129
3.1.1.166. nsslapd-syntaxcheck	129
3.1.1.167. nsslapd-syntaxlogging	130
3.1.1.168. nsslapd-threadnumber (Thread Number)	131
3.1.1.169. nsslapd-timelimit (Time Limit)	131
3.1.1.170. nsslapd-tmpdir	132
3.1.1.171. nsslapd-unhashed-pw-switch	132
3.1.1.172. nsslapd-validate-cert	133
3.1.1.173. nsslapd-verify-filter-schema	133
3.1.1.174. nsslapd-versionstring	134
3.1.1.175. nsslapd-workingdir	135
3.1.1.176. passwordAllowChangeTime	135

3.1.1.177. passwordChange (Password Change)	135
3.1.1.178. passwordCheckSyntax (Check Password Syntax)	136
3.1.1.179. passwordDictCheck	137
3.1.1.180. passwordExp (Password Expiration)	137
3.1.1.181. passwordExpirationTime	137
3.1.1.182. passwordExpWarned	138
3.1.1.183. passwordGraceLimit (Password Expiration)	138
3.1.1.184. passwordHistory (Password History)	139
3.1.1.185. passwordInHistory (Number of Passwords to Remember)	139
3.1.1.186. passwordIsGlobalPolicy (Password Policy and Replication)	140
3.1.1.187. passwordLegacyPolicy	140
3.1.1.188. passwordLockout (Account Lockout)	140
3.1.1.189. passwordLockoutDuration (Lockout Duration)	141
3.1.1.190. passwordMaxAge (Password Maximum Age)	141
3.1.1.191. passwordBadWords	142
3.1.1.192. passwordMaxClassChars	142
3.1.1.193. passwordMaxFailure (Maximum Password Failures)	143
3.1.1.194. passwordMaxRepeats (Password Syntax)	143
3.1.1.195. passwordMaxSeqSets	144
3.1.1.196. passwordMaxSequence	144
3.1.1.197. passwordMin8Bit (Password Syntax)	145
3.1.1.198. passwordMinAge (Password Minimum Age)	145
3.1.1.199. passwordMinAlphas (Password Syntax)	146
3.1.1.200. passwordMinCategories (Password Syntax)	146
3.1.1.201. PasswordMinDigits (Password Syntax)	147
3.1.1.202. passwordMinLength (Password Minimum Length)	147
3.1.1.203. PasswordMinLowers (Password Syntax)	148
3.1.1.204. PasswordMinSpecials (Password Syntax)	148
3.1.1.205. PasswordMinTokenLength (Password Syntax)	148
3.1.1.206. PasswordMinUppers (Password Syntax)	149
3.1.1.207. passwordMustChange (Password Must Change)	149
3.1.1.208. passwordPalindrome	150
3.1.1.209. passwordResetFailureCount (Reset Password Failure Count After)	150
3.1.1.210. passwordUserAttributes	151
3.1.1.211. passwordSendExpiringTime	151
3.1.1.212. passwordStorageScheme (Password Storage Scheme)	152
3.1.1.213. passwordTPRDelayExpireAt	152
3.1.1.214. passwordTPRDelayValidFrom	153
3.1.1.215. passwordTPRMaxUse	153
3.1.1.216. passwordTrackUpdateTime	154
3.1.1.217. passwordUnlock (Unlock Account)	154
3.1.1.218. passwordWarning (Send Warning)	155
3.1.1.219. retryCountResetTime	155
3.1.2. cn=changelog5,cn=config	156
3.1.2.1. cn	156
3.1.2.2. nsslapd-changelogcompactdb-interval	156
3.1.2.3. nsslapd-changelogdir	157
3.1.2.4. nsslapd-changelogmaxage (Max Changelog Age)	157
3.1.2.5. nsslapd-changelogmaxentries (Max Changelog Records)	158
3.1.2.6. nsslapd-changelogmaxconcurrentwrites (Max Concurrent Rewrites)	159
3.1.2.7. nsslapd-changelogtrim-interval (Replication Changelog Trimming Interval)	159
3.1.2.8. nsslapd-encryptionalgorithm (Encryption Algorithm)	160
3.1.2.9. nsSymmetricKey	160

3.1.3. Changelog Attributes	160
3.1.3.1. changes	161
3.1.3.2. changeLog	161
3.1.3.3. changeNumber	161
3.1.3.4. changeTime	161
3.1.3.5. changeType	162
3.1.3.6. deleteOldRdn	162
3.1.3.7. filterInfo	162
3.1.3.8. newRdn	163
3.1.3.9. newSuperior	163
3.1.3.10. targetDn	163
3.1.4. cn=encryption	163
3.1.4.1. allowWeakCipher	164
3.1.4.2. allowWeakDHParam	164
3.1.4.3. nsSSL3Ciphers	165
3.1.4.4. nsSSLActivation	165
3.1.4.5. nsSSLClientAuth	166
3.1.4.6. nsSSLEnabledCiphers	166
3.1.4.7. nsSSLPersonalitySSL	167
3.1.4.8. nsSSLSessionTimeout	167
3.1.4.9. nsSSLSupportedCiphers	168
3.1.4.10. nsSSLToken	168
3.1.4.11. nsTLS1	168
3.1.4.12. nsTLSAllowClientRenegotiation	169
3.1.4.13. sslVersionMin	169
3.1.4.14. sslVersionMax	170
3.1.5. cn=features	170
3.1.5.1. oid	171
3.1.6. cn=mapping tree	171
3.1.7. Suffix Configuration Attributes under cn=suffix_DN	171
3.1.7.1. cn	172
3.1.7.2. nsslapd-backend	172
3.1.7.3. nsslapd-distribution-function	172
3.1.7.4. nsslapd-distribution-plugin	173
3.1.7.5. nsslapd-parent	173
3.1.7.6. nsslapd-referral	174
3.1.7.7. nsslapd-state	174
3.1.8. Replication Attributes under cn=replica,cn=suffixDN,cn=mapping tree,cn=config	175
3.1.8.1. cn	175
3.1.8.2. nsds5DebugReplicaTimeout	175
3.1.8.3. nsDS5Flags	176
3.1.8.4. nsDS5ReplConflict	176
3.1.8.5. nsDS5ReplicaAutoReferral	177
3.1.8.6. nsState	177
3.1.8.7. nsDS5ReplicaAbortCleanRUV	177
3.1.8.8. nsds5ReplicaBackoffMin and nsds5ReplicaBackoffMax	178
3.1.8.9. nsDS5ReplicaBindDN	178
3.1.8.10. nsDS5ReplicaBindDNGroup	178
3.1.8.11. nsDS5ReplicaBindDNGroupCheckInterval	179
3.1.8.12. nsDS5ReplicaChangeCount	179
3.1.8.13. nsDS5ReplicaCleanRUV	180
3.1.8.14. nsDS5Replicald	180
3.1.8.15. nsDS5ReplicaLegacyConsumer	181

3.1.8.16. nsDS5ReplicaName	181
3.1.8.17. nsds5ReplicaProtocolTimeout	182
3.1.8.18. nsDS5ReplicaPurgeDelay	182
3.1.8.19. nsDS5ReplicaReapActive	183
3.1.8.20. nsDS5ReplicaReferral	183
3.1.8.21. nsDS5ReplicaReleaseTimeout	184
3.1.8.22. nsDS5ReplicaRoot	185
3.1.8.23. nsDS5ReplicaTombstonePurgeInterval	185
3.1.8.24. nsDS5ReplicaType	185
3.1.8.25. nsds5Task	186
3.1.9. Replication Attributes under cn=ReplicationAgreementName,cn=replica,cn=suffixName,cn=mapping tree,cn=config	187
3.1.9.1. cn	187
3.1.9.2. description	187
3.1.9.3. nsDS5ReplicaBindDN	188
3.1.9.4. nsDS5ReplicaBindMethod	188
3.1.9.5. nsds5ReplicaBootstrapBindDN	189
3.1.9.6. nsds5ReplicaBootstrapBindMethod	189
3.1.9.7. nsds5ReplicaBootstrapCredentials	190
3.1.9.8. nsds5ReplicaBootstrapTransportInfo	191
3.1.9.9. nsDS5ReplicaBusyWaitTime	191
3.1.9.10. nsDS5ReplicaChangesSentSinceStartup	192
3.1.9.11. nsDS5ReplicaCredentials	192
3.1.9.12. nsds5ReplicaEnabled	193
3.1.9.13. nsds5ReplicaFlowControlPause	193
3.1.9.14. nsds5ReplicaFlowControlWindow	194
3.1.9.15. nsDS5ReplicaHost	194
3.1.9.16. nsDS5ReplicaLastInitEnd	195
3.1.9.17. nsDS5ReplicaLastInitStart	195
3.1.9.18. nsDS5ReplicaLastInitStatus	196
3.1.9.19. nsDS5ReplicaLastUpdateEnd	196
3.1.9.20. nsDS5ReplicaLastUpdateStart	197
3.1.9.21. nsds5replicaLastUpdateStatus	197
3.1.9.22. nsDS5ReplicaPort	198
3.1.9.23. nsDS5ReplicaReapActive	198
3.1.9.24. nsDS5BeginReplicaRefresh	199
3.1.9.25. nsDS5ReplicaRoot	199
3.1.9.26. nsDS5ReplicaSessionPauseTime	199
3.1.9.27. nsds5ReplicaStripAttrs	200
3.1.9.28. nsDS5ReplicatedAttributeList	201
3.1.9.29. nsDS5ReplicatedAttributeListTotal	201
3.1.9.30. nsDS5ReplicaTimeout	202
3.1.9.31. nsDS5ReplicaTransportInfo	202
3.1.9.32. nsDS5ReplicaUpdateInProgress	203
3.1.9.33. nsDS5ReplicaUpdateSchedule	203
3.1.9.34. nsDS5ReplicaWaitForAsyncResults	204
3.1.9.35. nsDS5Ouv	204
3.1.9.36. nsrvReplicaLastModified	204
3.1.9.37. nsds5ReplicaProtocolTimeout	204
3.1.10. Synchronization Attributes under cn=syncAgreementName,cn=WindowsReplica,cn=suffixName,cn=mapping tree,cn=config	205
3.1.10.1. nsds7DirectoryReplicaSubtree	206
3.1.10.2. nsds7DirsyncCookie	206

3.1.10.3. nsds7NewWinGroupSyncEnabled	207
3.1.10.4. nsds7NewWinUserSyncEnabled	207
3.1.10.5. nsds7WindowsDomain	208
3.1.10.6. nsds7WindowsReplicaSubtree	208
3.1.10.7. oneWaySync	208
3.1.10.8. winSyncInterval	209
3.1.10.9. winSyncMoveAction	209
3.1.11. cn=monitor	210
3.1.12. cn=replication	213
3.1.13. cn=sasl	213
3.1.13.1. nsSaslMapBaseDNTemplate	213
3.1.13.2. nsSaslMapFilterTemplate	213
3.1.13.3. nsSaslMapPriority	214
3.1.13.4. nsSaslMapRegexString	214
3.1.14. cn=SNMP	214
3.1.14.1. nssnmpenabled	215
3.1.14.2. nssnmporganization	215
3.1.14.3. nssnmplocation	215
3.1.14.4. nssnmpcontact	216
3.1.14.5. nssnmpdescription	216
3.1.14.6. nssnmpmasterhost	216
3.1.14.7. nssnmpmasterport	217
3.1.15. SNMP Statistic Attributes	217
3.1.16. cn=tasks	220
3.1.16.1. Task Invocation Attributes for Entries under cn=tasks	221
3.1.16.2. cn=import	224
3.1.16.3. cn=export	228
3.1.16.4. cn=backup	232
3.1.16.5. cn=restore	233
3.1.16.6. cn=index	235
nsIndexVLVAttribute	236
3.1.16.7. cn=schema reload task	236
3.1.16.8. cn=memberof task	237
3.1.16.9. cn=fixup linked attributes	239
3.1.16.10. cn=syntax validate	240
3.1.16.11. cn=USN tombstone cleanup task	241
3.1.16.12. cn=cleanallruv	243
3.1.16.13. cn=abort cleanallruv	245
3.1.16.14. cn=automember rebuild membership	247
3.1.16.15. cn=automember export updates	248
3.1.16.16. cn=automember map updates	250
3.1.16.17. cn=des2aes	251
3.1.17. cn=uniqueid generator	251
3.1.18. Root DSE Configuration Parameters	252
3.1.18.1. nsslapd-return-default-opattr	252
3.2. CONFIGURATION OBJECT CLASSES	253
3.2.1. changeLogEntry (Object Class)	253
3.2.2. directoryServerFeature (Object Class)	254
3.2.3. nsBackendInstance (Object Class)	254
3.2.4. nsChangelog4Config (Object Class)	255
3.2.5. nsDS5Replica (Object Class)	255
3.2.6. nsDS5ReplicationAgreement (Object Class)	257
3.2.7. nsDSWindowsReplicationAgreement (Object Class)	259

3.2.8. nsEncryptionConfig	261
3.2.9. nsEncryptionModule	262
3.2.10. nsMappingTree (Object Class)	263
3.2.11. nsSaslMapping (Object Class)	263
3.2.12. nsslapdConfig (Object Class)	264
3.2.13. passwordPolicy (Object Class)	264
3.3. ROOT DSE ATTRIBUTES	267
3.3.1. dataversion	267
3.3.2. defaultNamingContext	267
3.3.3. lastusn	267
3.3.4. namingContexts	268
3.3.5. netscapemdsuffix	269
3.3.6. supportedControl	269
3.3.7. supportedExtension	269
3.3.8. supportedFeatures	269
3.3.9. supportedLDAPVersion	270
3.3.10. supportedSASLMechanisms	270
3.3.11. vendorName	270
3.3.12. vendorVersion	270
3.4. LEGACY ATTRIBUTES	271
3.4.1. Legacy Server Attributes	271
3.4.1.1. LDAPServer (Object Class)	271
3.4.1.2. changeLogMaximumAge	272
3.4.1.3. changeLogMaximumConcurrentWrites	272
3.4.1.4. changeLogMaximumSize	272
3.4.1.5. generation	273
3.4.1.6. nsSynchUniqueAttribute	273
3.4.1.7. nsSynchUserIDFormat	273
CHAPTER 4. PLUG-IN IMPLEMENTED SERVER FUNCTIONALITY REFERENCE	274
4.1. SERVER PLUG-IN FUNCTIONALITY REFERENCE	274
4.1.1. 7-bit Check Plug-in	274
4.1.2. ACL Plug-in	275
4.1.3. ACL Preoperation Plug-in	275
4.1.4. Account Policy Plug-in	276
4.1.5. Account Usability Plug-in	277
4.1.6. AD DN Plug-in	277
4.1.7. Attribute Uniqueness Plug-in	278
4.1.8. Auto Membership Plug-in	279
4.1.9. Binary Syntax Plug-in	280
4.1.10. Bit String Syntax Plug-in	280
4.1.11. Bitwise Plug-in	281
4.1.12. Boolean Syntax Plug-in	281
4.1.13. Case Exact String Syntax Plug-in	282
4.1.14. Case Ignore String Syntax Plug-in	283
4.1.15. Chaining Database Plug-in	283
4.1.16. Class of Service Plug-in	284
4.1.17. Content Synchronization Plug-in	285
4.1.18. Country String Syntax Plug-in	285
4.1.19. Delivery Method Syntax Plug-in	286
4.1.20. deref Plug-in	286
4.1.21. Distinguished Name Syntax Plug-in	287
4.1.22. Distributed Numeric Assignment Plug-in	288

4.1.23. Enhanced Guide Syntax Plug-in	288
4.1.24. Facsimile Telephone Number Syntax Plug-in	289
4.1.25. Fax Syntax Plug-in	289
4.1.26. Generalized Time Syntax Plug-in	290
4.1.27. Guide Syntax Plug-in	291
4.1.28. HTTP Client Plug-in	291
4.1.29. Integer Syntax Plug-in	292
4.1.30. Internationalization Plug-in	293
4.1.31. JPEG Syntax Plug-in	293
4.1.32. ldbm database Plug-in	294
4.1.33. Linked Attributes Plug-in	295
4.1.34. Managed Entries Plug-in	296
4.1.35. MemberOf Plug-in	296
4.1.36. Multi-master Replication Plug-in	297
4.1.37. Name and Optional UID Syntax Plug-in	298
4.1.38. Numeric String Syntax Plug-in	298
4.1.39. Octet String Syntax Plug-in	299
4.1.40. OID Syntax Plug-in	300
4.1.41. PAM Pass Through Auth Plug-in	300
4.1.42. Pass Through Authentication Plug-in	301
4.1.43. Password Storage Schemes	302
Strong Password Storage Schemes	302
Weak Password Storage Schemes	302
4.1.44. Posix Winsync API Plug-in	303
4.1.45. Postal Address String Syntax Plug-in	304
4.1.46. Printable String Syntax Plug-in	304
4.1.47. Referential Integrity Postoperation Plug-in	305
4.1.48. Retro Changelog Plug-in	306
4.1.49. Roles Plug-in	307
4.1.50. RootDN Access Control Plug-in	307
4.1.51. Schema Reload Plug-in	308
4.1.52. Space Insensitive String Syntax Plug-in	308
4.1.53. State Change Plug-in	309
4.1.54. Syntax Validation Task Plug-in	310
4.1.55. Telephone Syntax Plug-in	310
4.1.56. Teletex Terminal Identifier Syntax Plug-in	311
4.1.57. Telex Number Syntax Plug-in	311
4.1.58. URI Syntax Plug-in	312
4.1.59. USN Plug-in	313
4.1.60. Views Plug-in	313
4.2. LIST OF ATTRIBUTES COMMON TO ALL PLUG-INS	314
4.2.1. nsslapdPlugin (Object Class)	314
4.2.2. nsslapd-logAccess	315
4.2.3. nsslapd-logAudit	315
4.2.4. nsslapd-pluginDescription	316
4.2.5. nsslapd-pluginEnabled	316
4.2.6. nsslapd-pluginId	317
4.2.7. nsslapd-pluginInitfunc	317
4.2.8. nsslapd-pluginPath	317
4.2.9. nsslapd-pluginPrecedence	318
4.2.10. nsslapd-pluginType	318
4.2.11. nsslapd-pluginVendor	318
4.2.12. nsslapd-pluginVersion	319

4.3. ATTRIBUTES ALLOWED BY CERTAIN PLUG-INS	319
4.3.1. nsslapd-dynamic-plugins	319
4.3.2. nsslapd-pluginConfigArea	320
4.3.3. nsslapd-pluginLoadNow	320
4.3.4. nsslapd-pluginLoadGlobal	321
4.3.5. nsslapd-plugindepends-on-type	321
4.3.6. nsslapd-plugindepends-on-named	321
4.4. DATABASE PLUG-IN ATTRIBUTES	322
4.4.1. Database Attributes under cn=config,cn=ldbm database,cn=plugins,cn=config	322
4.4.1.1. nsslapd-backend-implement	322
4.4.1.2. nsslapd-backend-opt-level	323
4.4.1.3. nsslapd-directory	324
4.4.1.4. nsslapd-exclude-from-export	324
4.4.1.5. nsslapd-db-transaction-wait	324
4.4.1.6. nsslapd-db-private-import-mem	325
4.4.1.7. nsslapd-db-deadlock-policy	325
4.4.1.8. nsslapd-idl-switch	326
4.4.1.9. nsslapd-idlistscanlimit	326
4.4.1.10. nsslapd-lookthroughlimit	327
4.4.1.11. nsslapd-mode	327
4.4.1.12. nsslapd-pagedidlistscanlimit	328
4.4.1.13. nsslapd-pagedlookthroughlimit	328
4.4.1.14. nsslapd-rangelookthroughlimit	329
4.4.1.15. nsslapd-search-bypass-filter-test	330
4.4.1.16. nsslapd-search-use-vlv-index	330
4.4.1.17. nsslapd-subtree-rename-switch	330
4.4.2. Database Attributes under cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config	331
4.4.2.1. nsslapd-cache-autosize	331
4.4.2.2. nsslapd-cache-autosize-split	332
4.4.2.3. nsslapd-db-checkpoint-interval	332
4.4.2.4. nsslapd-db-circular-logging	333
4.4.2.5. nsslapd-db-compactdb-interval	333
4.4.2.6. nsslapd-db-debug	334
4.4.2.7. nsslapd-db-durable-transactions	334
4.4.2.8. nsslapd-db-home-directory	335
4.4.2.9. nsslapd-db-idl-divisor	336
4.4.2.10. nsslapd-db-locks	336
4.4.2.11. nsslapd-db-locks-monitoring-enable	337
4.4.2.12. nsslapd-db-locks-monitoring-pause	337
4.4.2.13. nsslapd-db-locks-monitoring-threshold	338
4.4.2.14. nsslapd-db-logbuf-size	338
4.4.2.15. nsslapd-db-logdirectory	339
4.4.2.16. nsslapd-db-logfile-size	339
4.4.2.17. nsslapd-db-page-size	340
4.4.2.18. nsslapd-db-spin-count	340
4.4.2.19. nsslapd-db-transaction-batch-max-wait	341
4.4.2.20. nsslapd-db-transaction-batch-min-wait	341
4.4.2.21. nsslapd-db-transaction-batch-val	342
4.4.2.22. nsslapd-db-trickle-percentage	343
4.4.2.23. nsslapd-db-verbose	343
4.4.2.24. nsslapd-import-cache-autosize	344
4.4.2.25. nsslapd-dbcachesize	345
4.4.2.26. nsslapd-dbnocache	346

4.4.2.27. nsslapd-search-bypass-filter-test	346
4.4.3. Database Attributes under cn=monitor,cn=ldbm database,cn=plugins,cn=config	347
4.4.4. Database Attributes under cn=database_name,cn=ldbm database,cn=plugins,cn=config	348
4.4.4.1. nsslapd-cachesize	348
4.4.4.2. nsslapd-cachememsize	349
4.4.4.3. nsslapd-directory	350
4.4.4.4. nsslapd-dncachememsize	350
4.4.4.5. nsslapd-readonly	351
4.4.4.6. nsslapd-require-index	351
4.4.4.7. nsslapd-require-internalop-index	351
4.4.4.8. nsslapd-suffix	352
4.4.4.9. vlvBase	352
4.4.4.10. vlvEnabled	353
4.4.4.11. vlvFilter	353
4.4.4.12. vlvIndex (Object Class)	354
4.4.4.13. vlvScope	354
4.4.4.14. vlvSearch (Object Class)	355
4.4.4.15. vlvSort	356
4.4.4.16. vlvUses	356
4.4.5. Database Attributes under cn=database,cn=monitor,cn=ldbm database,cn=plugins,cn=config	356
4.4.6. Database Attributes under cn=monitor,cn=userRoot,cn=ldbm database,cn=plugins,cn=config	359
4.4.7. Database Attributes under cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config	360
4.4.7.1. cn	360
4.4.7.2. nsIndex	361
4.4.7.3. nsIndexType	361
4.4.7.4. nsMatchingRule	362
4.4.7.5. nsSystemIndex	363
4.4.8. Database Attributes under cn=index,cn=database_name,cn=ldbm database,cn=plugins,cn=config	363
4.4.8.1. nsIndexIDListScanLimit	364
4.4.8.2. nsSubStrBegin	364
4.4.8.3. nsSubStrEnd	365
4.4.8.4. nsSubStrMiddle	365
4.4.9. Database Attributes under cn=attributeName,cn=encrypted attributes,cn=database_name,cn=ldbm database,cn=plugins,cn=config	366
4.4.9.1. nsAttributeEncryption (Object Class)	367
4.4.9.2. nsEncryptionAlgorithm	367
4.5. DATABASE LINK PLUG-IN ATTRIBUTES (CHAINING ATTRIBUTES)	368
4.5.1. Database Link Attributes under cn=config,cn=chaining database,cn=plugins,cn=config	368
4.5.1.1. nsActiveChainingComponents	368
4.5.1.2. nsMaxResponseDelay	369
4.5.1.3. nsMaxTestResponseDelay	369
4.5.1.4. nsTransmittedControls	370
4.5.2. Database Link Attributes under cn=default instance config,cn=chaining database,cn=plugins,cn=config	370
4.5.2.1. nsAbandonedSearchCheckInterval	370
4.5.2.2. nsBindConnectionsLimit	371
4.5.2.3. nsBindRetryLimit	371
4.5.2.4. nsBindTimeout	372
4.5.2.5. nsCheckLocalACI	372
4.5.2.6. nsConcurrentBindLimit	373
4.5.2.7. nsConcurrentOperationsLimit	373
4.5.2.8. nsConnectionLife	373
4.5.2.9. nsOperationConnectionsLimit	374

4.5.2.10. nsProxiedAuthorization	374
4.5.2.11. nsReferralOnScopedSearch	375
4.5.2.12. nsSizeLimit	375
4.5.2.13. nsTimeLimit	375
4.5.3. Database Link Attributes under cn=database_link_name,cn=chaining database,cn=plugins,cn=config	376
4.5.3.1. nsBindMechanism	376
4.5.3.2. nsFarmServerURL	377
4.5.3.3. nsMultiplexorBindDN	377
4.5.3.4. nsMultiplexorCredentials	378
4.5.3.5. nshoplimit	378
4.5.3.6. nsUseStartTLS	379
4.5.4. Database Link Attributes under cn=monitor,cn=database instance name,cn=chaining database,cn=plugins,cn=config	379
4.6. PAM PASS THROUGH AUTH PLUG-IN ATTRIBUTES	380
4.6.1. pamConfig (Object Class)	381
4.6.2. pamExcludeSuffix	382
4.6.3. pamFallback	382
4.6.4. pamFilter	382
4.6.5. pamIDAttr	382
4.6.6. pamIDMapMethod	383
4.6.7. pamIncludeSuffix	383
4.6.8. pamMissingSuffix	383
4.6.9. pamSecure	384
4.6.10. pamService	384
4.7. ACCOUNT POLICY PLUG-IN ATTRIBUTES	384
4.7.1. altstateattrname	386
4.7.2. alwaysRecordLogin	386
4.7.3. alwaysRecordLoginAttr	386
4.7.4. limitattrname	387
4.7.5. specattrname	387
4.7.6. stateattrname	388
4.8. AD DN PLUG-IN ATTRIBUTES	388
4.8.1. cn	388
4.8.2. addn_base	389
4.8.3. addn_filter	389
4.9. AUTO MEMBERSHIP PLUG-IN ATTRIBUTES	389
4.9.1. autoMemberDefaultGroup	390
4.9.2. autoMemberDefinition (Object Class)	391
4.9.3. autoMemberExclusiveRegex	391
4.9.4. autoMemberFilter	391
4.9.5. autoMemberGroupingAttr	392
4.9.6. autoMemberInclusiveRegex	392
4.9.7. autoMemberProcessModifyOps	393
4.9.8. autoMemberRegexRule (Object Class)	393
4.9.9. autoMemberScope	394
4.9.10. autoMemberTargetGroup	394
4.10. DISTRIBUTED NUMERIC ASSIGNMENT PLUG-IN ATTRIBUTES	394
4.10.1. dnaPluginConfig (Object Class)	395
4.10.2. dnaFilter	395
4.10.3. dnaInterval	396
4.10.4. dnaMagicRegen	396
4.10.5. dna.MaxValue	397

4.10.6. dnaNextRange	397
4.10.7. dnaNextValue	398
4.10.8. dnaPrefix	398
4.10.9. dnaRangeRequestTimeout	399
4.10.10. dnaScope	399
4.10.11. dnaSharedCfgDN	400
4.10.12. dnaThreshold	400
4.10.13. dnaType	401
4.10.14. dnaSharedConfig (Object Class)	401
4.10.15. dnaHostname	402
4.10.16. dnaPortNum	402
4.10.17. dnaRemainingValues	403
4.10.18. dnaRemoteBindCred	403
4.10.19. dnaRemoteBindDN	404
4.10.20. dnaRemoteBindMethod	404
4.10.21. dnaRemoteConnProtocol	405
4.10.22. dnaSecurePortNum	405
4.11. LINKED ATTRIBUTES PLUG-IN ATTRIBUTES	406
4.11.1. linkScope	406
4.11.2. linkType	406
4.11.3.managedType	407
4.12. MANAGED ENTRIES PLUG-IN ATTRIBUTES	407
4.12.1. managedBase	407
4.12.2. managedTemplate	408
4.12.3. originFilter	408
4.12.4. originScope	409
4.13. MEMBEROF PLUG-IN ATTRIBUTES	409
4.13.1. cn	409
4.13.2. memberOfAllBackends	410
4.13.3. memberOfAttr	410
4.13.4. memberOfAutoAddOC	411
4.13.5. memberOfEntryScope	411
4.13.6. memberOfEntryScopeExcludeSubtree	412
4.13.7. memberOfGroupAttr	412
4.14. ATTRIBUTE UNIQUENESS PLUG-IN ATTRIBUTES	413
4.14.1. cn	413
4.14.2. uniqueness-attribute-name	413
4.14.3. uniqueness-subtrees	414
4.14.4. uniqueness-across-all-subtrees	414
4.14.5. uniqueness-top-entry-oc	414
4.14.6. uniqueness-subtree-entries-oc	415
4.15. POSIX WINSYNC API PLUG-IN ATTRIBUTES	415
4.15.1. posixWinsyncCreateMemberOfTask	416
4.15.2. posixWinsyncLowerCaseUID	416
4.15.3. posixWinsyncMapMemberUID	416
4.15.4. posixWinsyncMapNestedGrouping	417
4.15.5. posixWinsyncMsSFUSchema	417
4.16. RETRO CHANGELOG PLUG-IN ATTRIBUTES	417
4.16.1. isReplicated	418
4.16.2. nsslapd-attribute	418
4.16.3. nsslapd-changelogdir	419
4.16.4. nsslapd-changelogmaxage (Max Changelog Age)	419
4.16.5. nsslapd-exclude-attrs	420

4.16.6. nsslapd-exclude-suffix	420
4.17. ROOTDN ACCESS CONTROL PLUG-IN ATTRIBUTES	421
4.17.1. rootdn-allow-host	421
4.17.2. rootdn-allow-ip	422
4.17.3. rootdn-close-time	422
4.17.4. rootdn-days-allowed	422
4.17.5. rootdn-deny-ip	423
4.17.6. rootdn-open-time	424
CHAPTER 5. DIRECTORY ENTRY SCHEMA REFERENCE	425
5.1. ABOUT DIRECTORY SERVER SCHEMA	425
5.1.1. Schema Definitions	425
5.1.1.1. Object Classes	425
5.1.1.1.1. Required and Allowed Attributes	426
5.1.1.1.2. Object Class Inheritance	426
5.1.1.2. Attributes	426
5.1.1.2.1. Directory Server Attribute Syntaxes	427
5.1.1.2.2. Single- and Multi-Valued Attributes	430
5.1.2. Default Directory Server Schema Files	430
5.1.3. Object Identifiers (OIDs)	432
5.1.4. Extending the Schema	433
5.1.5. Schema Checking	433
5.1.6. Syntax Validation	434
5.2. ENTRY ATTRIBUTE REFERENCE	434
5.2.1. abstract	434
5.2.2. accessTo	435
5.2.3. accountInactivityLimit	435
5.2.4. acctPolicySubentry	435
5.2.5. administratorContactInfo	436
5.2.6. adminRole	436
5.2.7. adminUrl	436
5.2.8. aliasedObjectName	436
5.2.9. associatedDomain	437
5.2.10. associatedName	437
5.2.11. attributeTypes	437
5.2.12. audio	438
5.2.13. authorCn	438
5.2.14. authorityRevocationList	438
5.2.15. authorSn	439
5.2.16. automountInformation	439
5.2.17. bootFile	439
5.2.18. bootParameter	440
5.2.19. buildingName	440
5.2.20. businessCategory	440
5.2.21. c (countryName)	441
5.2.22. cACertificate	441
5.2.23. carLicense	441
5.2.24. certificateRevocationList	442
5.2.25. cn (commonName)	442
5.2.26. co (friendlyCountryName)	442
5.2.27. cosAttribute	443
5.2.28. cosIndirectSpecifier	443
5.2.29. cosPriority	443

5.2.30. cosSpecifier	444
5.2.31. cosTargetTree	444
5.2.32. cosTemplateDn	444
5.2.33. crossCertificatePair	445
5.2.34. dc (domainComponent)	445
5.2.35. deltaRevocationList	445
5.2.36. departmentNumber	446
5.2.37. description	446
5.2.38. destinationIndicator	446
5.2.39. displayName	446
5.2.40. dITRedirect	447
5.2.41. dmdName	447
5.2.42. dn (distinguishedName)	447
5.2.43. dNSRecord	448
5.2.44. documentAuthor	448
5.2.45. documentIdentifier	448
5.2.46. documentLocation	449
5.2.47. documentPublisher	449
5.2.48. documentStore	449
5.2.49. documentTitle	450
5.2.50. documentVersion	450
5.2.51. drink (favouriteDrink)	450
5.2.52. dSAQuality	450
5.2.53. employeeNumber	451
5.2.54. employeeType	451
5.2.55. enhancedSearchGuide	451
5.2.56. fax (facsimileTelephoneNumber)	452
5.2.57. gecos	452
5.2.58. generationQualifier	453
5.2.59. gidNumber	453
5.2.60. givenName	453
5.2.61. homeDirectory	454
5.2.62. homePhone	454
5.2.63. homePostalAddress	455
5.2.64. host	455
5.2.65. houseIdentifier	455
5.2.66. inetDomainBaseDN	456
5.2.67. inetDomainStatus	456
5.2.68. inetSubscriberAccountId	456
5.2.69. inetSubscriberChallenge	457
5.2.70. inetSubscriberResponse	457
5.2.71. inetUserHttpURL	457
5.2.72. inetUserStatus	457
5.2.73. info	458
5.2.74. initials	458
5.2.75. installationTimeStamp	458
5.2.76. internationalISDNNumber	459
5.2.77. ipHostNumber	459
5.2.78. ipNetmaskNumber	459
5.2.79. ipNetworkNumber	460
5.2.80. ipProtocolNumber	460
5.2.81. ipServicePort	461
5.2.82. ipServiceProtocol	461

5.2.83. janetMailbox	461
5.2.84. jpegPhoto	462
5.2.85. keyWords	462
5.2.86. knowledgeInformation	462
5.2.87. l (localityName)	463
5.2.88. labeledURI	463
5.2.89. loginShell	463
5.2.90. macAddress	464
5.2.91. mail	464
5.2.92. mailAccessDomain	465
5.2.93. mailAlternateAddress	465
5.2.94. mailAutoReplyMode	465
5.2.95. mailAutoReplyText	465
5.2.96. mailDeliveryOption	466
5.2.97. mailEnhancedUniqueMember	466
5.2.98. mailForwardingAddress	466
5.2.99. mailHost	467
5.2.100. mailMessageStore	467
5.2.101. mailPreferenceOption	467
5.2.102. mailProgramDeliveryInfo	468
5.2.103. mailQuota	468
5.2.104. mailRoutingAddress	468
5.2.105. manager	468
5.2.106. member	469
5.2.107. memberCertificateDescription	469
5.2.108. memberNisNetgroup	470
5.2.109. memberOf	470
5.2.110. memberUid	471
5.2.111. memberURL	471
5.2.112. mepManagedBy	471
5.2.113. mepManagedEntry	472
5.2.114. mepMappedAttr	472
5.2.115. mepRDNAAttr	472
5.2.116. mepStaticAttr	473
5.2.117. grpAddHeader	473
5.2.118. grpAllowedBroadcaster	473
5.2.119. grpAllowedDomain	474
5.2.120. grpApprovePassword	474
5.2.121. grpBroadcasterPolicy	474
5.2.122. grpDeliverTo	474
5.2.123. grpErrorsTo	475
5.2.124. grpModerator	475
5.2.125. grpMsgMaxSize	475
5.2.126. grpMsgRejectAction	475
5.2.127. grpMsgRejectText	476
5.2.128. grpNoDuplicateChecks	476
5.2.129. grpRemoveHeader	476
5.2.130. grpRFC822MailMember	477
5.2.131. mobile	477
5.2.132. mozillaCustom1	477
5.2.133. mozillaCustom2	477
5.2.134. mozillaCustom3	478
5.2.135. mozillaCustom4	478

5.2.136. mozillaHomeCountryName	478
5.2.137. mozillaHomeLocalityName	478
5.2.138. mozillaHomePostalCode	479
5.2.139. mozillaHomeState	479
5.2.140. mozillaHomeStreet	479
5.2.141. mozillaHomeStreet2	479
5.2.142. mozillaHomeUrl	480
5.2.143. mozillaNickname (xmozzilanickname)	480
5.2.144. mozillaSecondEmail (xmozzilasecondemail)	480
5.2.145. mozillaUseHtmlMail (xmozzilausehtmlmail)	481
5.2.146. mozillaWorkStreet2	481
5.2.147. mozillaWorkUrl	481
5.2.148. multiLineDescription	481
5.2.149. name	482
5.2.150. netscapeReversiblePassword	482
5.2.151. NisMapEntry	482
5.2.152. nisMapName	483
5.2.153. nisNetgroupTriple	483
5.2.154. nsAccessLog	483
5.2.155. nsAdminAccessAddresses	484
5.2.156. nsAdminAccessHosts	484
5.2.157. nsAdminAccountInfo	484
5.2.158. nsAdminCacheLifetime	484
5.2.159. nsAdminCgiWaitPid	485
5.2.160. nsAdminDomainName	485
5.2.161. nsAdminEnableEnduser	485
5.2.162. nsAdminEndUserHTMLIndex	485
5.2.163. nsAdminGroupName	486
5.2.164. nsAdminOneACLDir	486
5.2.165. nsAdminSIEDN	486
5.2.166. nsAdminUsers	487
5.2.167. nsAIMid	487
5.2.168. nsBaseDN	487
5.2.169. nsBindDN	487
5.2.170. nsBindPassword	488
5.2.171. nsBuildNumber	488
5.2.172. nsBuildSecurity	488
5.2.173. nsCertConfig	488
5.2.174. nsClassname	489
5.2.175. nsConfigRoot	489
5.2.176. nsscpAIMScreenname	489
5.2.177. nsDefaultAcceptLanguage	489
5.2.178. nsDefaultObjectClass	490
5.2.179. nsDeleteclassname	490
5.2.180. nsDirectoryFailoverList	490
5.2.181. nsDirectoryInfoRef	490
5.2.182. nsDirectoryURL	491
5.2.183. nsDisplayName	491
5.2.184. nsErrorLog	491
5.2.185. nsExecRef	491
5.2.186. nsExpirationDate	492
5.2.187. nsGroupRDNCOMPONENT	492
5.2.188. nsHardwarePlatform	492

5.2.189. nsHelpRef	493
5.2.190. nsHostLocation	493
5.2.191. nsICQid	493
5.2.192. nsInstalledLocation	493
5.2.193. nsJarfilename	494
5.2.194. nsLdapSchemaVersion	494
5.2.195. nsLicensedFor	494
5.2.196. nsLicenseEndTime	495
5.2.197. nsLicenseStartTime	495
5.2.198. nsLogSuppress	495
5.2.199. nsmsgDisallowAccess	496
5.2.200. nsmsgNumMsgQuota	496
5.2.201. nsMSNid	496
5.2.202. nsNickName	496
5.2.203. nsNYR	497
5.2.204. nsOsVersion	497
5.2.205. nsPidLog	497
5.2.206. nsPreference	497
5.2.207. nsProductName	498
5.2.208. nsProductVersion	498
5.2.209. nsRevisionNumber	498
5.2.210. nsSecureServerPort	498
5.2.211. nsSerialNumber	499
5.2.212. nsServerAddress	499
5.2.213. nsServerCreationClassname	499
5.2.214. nsServerID	500
5.2.215. nsServerMigrationClassname	500
5.2.216. nsServerPort	500
5.2.217. nsServerSecurity	501
5.2.218. nsSNMPContact	501
5.2.219. nsSNMPDescription	501
5.2.220. nsSNMPEnabled	501
5.2.221. nsSNMPLocation	502
5.2.222. nsSNMPMasterHost	502
5.2.223. nsSNMPMasterPort	502
5.2.224. nsSNMPOrganization	502
5.2.225. nsSuiteSpotUser	503
5.2.226. nsTaskLabel	503
5.2.227. nsUniqueAttribute	503
5.2.228. nsUserIDFormat	504
5.2.229. nsUserRDNComponent	504
5.2.230. nsValueBin	504
5.2.231. nsValueCES	504
5.2.232. nsValueCIS	505
5.2.233. nsValueDefault	505
5.2.234. nsValueDescription	505
5.2.235. nsValueDN	505
5.2.236. nsValueFlags	505
5.2.237. nsValueHelpURL	506
5.2.238. nsValueInt	506
5.2.239. nsValueSyntax	506
5.2.240. nsValueTel	506
5.2.241. nsValueType	507

5.2.242. nsVendor	507
5.2.243. nsViewConfiguration	507
5.2.244. nsViewFilter	507
5.2.245. nsWellKnownJarfiles	508
5.2.246. nswmExtendedUserPrefs	508
5.2.247. nsYIMid	508
5.2.248. ntGroupAttributes	508
5.2.249. ntGroupCreateNewGroup	509
5.2.250. ntGroupDeleteGroup	509
5.2.251. ntGroupDomainId	509
5.2.252. ntGroupId	510
5.2.253. ntGroupType	510
5.2.254. ntUniqueId	511
5.2.255. ntUserAcctExpires	511
5.2.256. ntUserAuthFlags	511
5.2.257. ntUserBadPwCount	511
5.2.258. ntUserCodePage	512
5.2.259. ntUserComment	512
5.2.260. ntUserCountryCode	512
5.2.261. ntUserCreateNewAccount	513
5.2.262. ntUserDeleteAccount	513
5.2.263. ntUserDomainId	513
5.2.264. ntUserFlags	513
5.2.265. ntUserHomeDir	514
5.2.266. ntUserHomeDirDrive	514
5.2.267. ntUserLastLogoff	514
5.2.268. ntUserLastLogon	515
5.2.269. ntUserLogonHours	515
5.2.270. ntUserLogonServer	515
5.2.271. ntUserMaxStorage	516
5.2.272. ntUserNumLogons	516
5.2.273. ntUserParms	516
5.2.274. ntUserPasswordExpired	516
5.2.275. ntUserPrimaryGroupId	517
5.2.276. ntUserPriv	517
5.2.277. ntUserProfile	517
5.2.278. ntUserScriptPath	518
5.2.279. ntUserUniqueId	518
5.2.280. ntUserUnitsPerWeek	518
5.2.281. ntUserUsrComment	518
5.2.282. ntUserWorkstations	519
5.2.283. o (organizationName)	519
5.2.284. objectClass	519
5.2.285. objectClasses	520
5.2.286. obsoletedByDocument	520
5.2.287. obsoletesDocument	520
5.2.288. oncRpcNumber	521
5.2.289. organizationalStatus	521
5.2.290. otherMailbox	521
5.2.291. ou (organizationalUnitName)	522
5.2.292. owner	522
5.2.293. pager	522
5.2.294. parentOrganization	523

5.2.295. personalSignature	523
5.2.296. personalTitle	523
5.2.297. photo	523
5.2.298. physicalDeliveryOfficeName	524
5.2.299. postalAddress	524
5.2.300. postalCode	525
5.2.301. postOfficeBox	525
5.2.302. preferredDeliveryMethod	525
5.2.303. preferredLanguage	526
5.2.304. preferredLocale	526
5.2.305. preferredTimeZone	526
5.2.306. presentationAddress	526
5.2.307. protocolInformation	527
5.2.308. pwdReset	527
5.2.309. ref	527
5.2.310. registeredAddress	528
5.2.311. roleOccupant	528
5.2.312. roomNumber	528
5.2.313. searchGuide	529
5.2.314. secretary	529
5.2.315. seeAlso	529
5.2.316. serialNumber	530
5.2.317. serverHostName	530
5.2.318. serverProductName	530
5.2.319. serverRoot	531
5.2.320. serverVersionNumber	531
5.2.321. shadowExpire	531
5.2.322. shadowFlag	532
5.2.323. shadowInactive	532
5.2.324. shadowLastChange	533
5.2.325. shadowMax	533
5.2.326. shadowMin	534
5.2.327. shadowWarning	534
5.2.328. singleLevelQuality	535
5.2.329. sn (surname)	535
5.2.330. st (stateOrProvinceName)	535
5.2.331. street	536
5.2.332. subject	536
5.2.333. subtreeMaximumQuality	536
5.2.334. subtreeMinimumQuality	537
5.2.335. supportedAlgorithms	537
5.2.336. supportedApplicationContext	537
5.2.337. telephoneNumber	538
5.2.338. teletexTerminalIdentifier	538
5.2.339. telexNumber	538
5.2.340. title	539
5.2.341. ttl (TimeToLive)	539
5.2.342. uid (userID)	539
5.2.343. uidNumber	540
5.2.344. uniqueIdentifier	540
5.2.345. uniqueMember	541
5.2.346. updatedByDocument	541
5.2.347. updatesDocument	541

5.2.348. userCertificate	541
5.2.349. userClass	542
5.2.350. userPassword	542
5.2.351. userPKCS12	542
5.2.352. userSMIMECertificate	543
5.2.353. vacationEndDate	543
5.2.354. vacationStartDate	543
5.2.355. x121Address	544
5.2.356. x500UniqueIdentifier	544
5.3. ENTRY OBJECT CLASS REFERENCE	544
5.3.1. account	545
5.3.2. accountPolicy	545
5.3.3. alias	546
5.3.4. bootableDevice	546
5.3.5. cacheObject	547
5.3.6. cosClassicDefinition	548
5.3.7. cosDefinition	549
5.3.8. cosIndirectDefinition	550
5.3.9. cosPointerDefinition	550
5.3.10. cosSuperDefinition	551
5.3.11. cosTemplate	552
5.3.12. country	552
5.3.13. dcObject	553
5.3.14. device	554
5.3.15. document	554
5.3.16. documentSeries	556
5.3.17. domain	557
5.3.18. domainRelatedObject	559
5.3.19. dSA	560
5.3.20. extensibleObject	560
5.3.21. friendlyCountry	561
5.3.22. groupOfCertificates	562
5.3.23. groupOfMailEnhancedUniqueNames	562
5.3.24. groupOfNames	563
5.3.25. groupOfUniqueNames	564
5.3.26. groupOfURLs	565
5.3.27. ieee802Device	566
5.3.28. inetAdmin	567
5.3.29. inetDomain	568
5.3.30. inetOrgPerson	568
5.3.31. inetSubscriber	571
5.3.32. inetUser	572
5.3.33. ipHost	573
5.3.34. ipNetwork	574
5.3.35. ipProtocol	575
5.3.36. ipService	575
5.3.37. labeledURIObject	576
5.3.38. locality	577
5.3.39. mailGroup	578
5.3.40. mailRecipient	578
5.3.41. mepManagedEntry	580
5.3.42. mepOriginEntry	580
5.3.43. mepTemplateEntry	580

5.3.44. netscapeCertificateServer	581
5.3.45. netscapeDirectoryServer	581
5.3.46. NetscapeLinkedOrganization	582
5.3.47. netscapeMachineData	582
5.3.48. NetscapePreferences	582
5.3.49. netscapeReversiblePasswordObject	583
5.3.50. netscapeServer	583
5.3.51. netscapeWebServer	584
5.3.52. newPilotPerson	585
5.3.53. nisMap	587
5.3.54. nisNetgroup	587
5.3.55. nisObject	588
5.3.56. nsAdminConfig	589
5.3.57. nsAdminConsoleUser	590
5.3.58. nsAdminDomain	590
5.3.59. nsAdminGlobalParameters	591
5.3.60. nsAdminGroup	591
5.3.61. nsAdminObject	592
5.3.62. nsAdminResourceEditorExtension	593
5.3.63. nsAdminServer	593
5.3.64. nsAIMpresence	594
5.3.65. nsApplication	594
5.3.66. nsCertificateServer	596
5.3.67. nsComplexRoleDefinition	596
5.3.68. nsContainer	597
5.3.69. nsCustomView	597
5.3.70. nsDefaultObjectClasses	597
5.3.71. nsDirectoryInfo	598
5.3.72. nsDirectoryServer	599
5.3.73. nsFilteredRoleDefinition	600
5.3.74. nsGlobalParameters	600
5.3.75. nsHost	601
5.3.76. nsICQpresence	602
5.3.77. nsLicenseUser	603
5.3.78. nsManagedRoleDefinition	603
5.3.79. nsMessagingServerUser	604
5.3.80. nsMSNpresence	605
5.3.81. nsNestedRoleDefinition	605
5.3.82. nsResourceRef	606
5.3.83. nsRoleDefinition	607
5.3.84. nsSimpleRoleDefinition	607
5.3.85. nsSNMP	608
5.3.86. nsTask	609
5.3.87. nsTaskGroup	610
5.3.88. nsTopologyCustomView	610
5.3.89. nsTopologyPlugin	611
5.3.90. nsValueItem	611
5.3.91. nsView	612
5.3.92. nsYIMpresence	613
5.3.93. ntGroup	613
5.3.94. ntUser	615
5.3.95. oncRpc	618
5.3.96. organization	618

5.3.97. organizationalPerson	620
5.3.98. organizationalRole	622
5.3.99. organizationalUnit	624
5.3.100. person	625
5.3.101. pilotObject	626
5.3.102. pilotOrganization	627
5.3.103. pkıCA	629
5.3.104. pkıUser	629
5.3.105. posixAccount	630
5.3.106. posixGroup	631
5.3.107. referral	632
5.3.108. residentialPerson	632
5.3.109. RFC822LocalPart	634
5.3.110. room	636
5.3.111. shadowAccount	636
5.3.112. simpleSecurityObject	637
5.3.113. strongAuthenticationUser	638
CHAPTER 6. OPERATIONAL ATTRIBUTES AND OBJECT CLASSES	639
6.1. ACCOUNTUNLOCKTIME	639
6.2. ACI	639
6.3. ALTSERVER	639
6.4. CREATETIMESTAMP	640
6.5. CREATORSNAME	640
6.6. DITCONTENTRULES	640
6.7. DITSTRUCTURERULES	640
6.8. ENTRYUSN	641
6.9. INTERNALCREATORSNAME	641
6.10. INTERNALMODIFIERSNAME	642
6.11. HASSUBORDINATES	642
6.12. LASTLOGINTIME	642
6.13. LASTMODIFIEDBY	643
6.14. LASTMODIFIEDTIME	643
6.15. LDAPSUBENTRY	643
6.16. LDAPSNTAXES	644
6.17. MATCHINGRULES	644
6.18. MATCHINGRULEUSE	644
6.19. MODIFYTIMESTAMP	645
6.20. MODIFIERSNAME	645
6.21. NAMEFORMS	645
6.22. NSACCOUNTLOCK	645
6.23. NSAIMSTATUSGRAPHIC	646
6.24. NSAIMSTATUSTEXT	646
6.25. NSBACKENDSUFFIX	646
6.26. NSCPENTRYDN	647
6.27. NSDS5REPLCONFLICT	647
6.28. NSICQSTATUSGRAPHIC	647
6.29. NSICQSTATUSTEXT	647
6.30. NSIDLETIMEOUT	648
6.31. NSIDLISTSCANLIMIT	648
6.32. NSLOOKTHROUGHLIMIT	648
6.33. NSPAGEDIDLISTSCANLIMIT	649
6.34. NSPAGEDLOOKTHROUGHLIMIT	649

6.35. NSPAGEDSIZELIMIT	649
6.36. NSPARENTUNIQUEID	650
6.37. NSROLE	650
6.38. NSROLEDN	650
6.39. NSROLEFILTER	651
6.40. NSSCHEMACSN	651
6.41. NSSIZELIMIT	652
6.42. NSTIMELIMIT	652
6.43. NSTOMBSTONE (OBJECT CLASS)	652
6.44. NSUNIQUEID	653
6.45. NSYIMSTATUSGRAPHIC	653
6.46. NSYIMSTATUSTEXT	653
6.47. NUMSUBORDINATES	653
6.48. PASSWORDGRACEUSERTIME	654
6.49. PASSWORDRETRYCOUNT	654
6.50. PWDPOLICYSUBENTRY	654
6.51. PWDUPDATETIME	655
6.52. SUBSCHEMASUBENTRY	655
6.53. GLUE (OBJECT CLASS)	655
6.54. PASSWORDOBJECT (OBJECT CLASS)	656
6.55. SUBSCHEMA (OBJECT CLASS)	656
CHAPTER 7. LOG FILE REFERENCE	658
7.1. ACCESS LOG REFERENCE	658
7.1.1. Access Logging Levels	659
7.1.2. Default Access Logging Content	659
7.1.3. Access Log Content for Additional Access Logging Levels	667
7.1.4. Common Connection Codes	669
7.2. ERROR LOG REFERENCE	669
7.2.1. Error Log Logging Levels	670
7.2.2. Error Log Content	671
7.2.3. Error Log Content for Other Log Levels	673
7.3. AUDIT LOG REFERENCE	678
7.4. LDAP RESULT CODES	679
7.5. REPLACING LOG FILES WITH A NAMED PIPE	683
7.5.1. Using the Named Pipe for Logging	684
7.5.2. Starting the Named Pipe with the Server	685
7.5.3. Using Plug-ins with the Named Pipe Log	685
7.5.3.1. Loading Plug-ins with the Named Pipe Log Script	686
7.5.3.2. Writing Plug-ins to Use with the Named Pipe Log Script	686
CHAPTER 8. CONFIGURATION FILE REFERENCE	688
8.1. CERTMAP.CONF	688
CHAPTER 9. COMMAND-LINE UTILITIES	691
9.1. DS-REPLCHECK	691
9.2. LDIF	691
9.3. DBSCAN	691
9.4. DS-LOGPIPE.PY	693
9.5. DN2RDN	695
9.6. PWDHASH	696
APPENDIX A. TESTING SCRIPTS AVAILABLE WITH DIRECTORY SERVER	697
A.1. LDCLT (LOAD STRESS TESTS)	697

A.1.1. Syntax	697
A.1.2. Idclt Options	697
A.1.3. Results from Idclt	703
A.1.4. Exiting Idclt and Idclt Exit Codes	704
A.1.5. Usage Scenarios	705
A.1.5.1. Generating LDIFs	706
A.1.5.2. Adding Entries	708
A.1.5.3. Search Operations	709
A.1.5.4. Modify Operations	710
A.1.5.5. modrdn Operations	710
A.1.5.6. Delete Operations	711
A.1.5.7. Bind Operations	712
A.1.5.8. Running Operations on Random Base DNs	713
A.1.5.9. TLS Authentication	713
A.1.5.10. Abandon Operations	713
A.2. RSEARCH (SEARCH STRESS TESTS)	713
A.2.1. Syntax	714
A.2.2. Options	714
A.2.3. Usage Scenarios	717
A.2.3.1. Allowed Configuration Files	717
A.2.3.2. Results from rsearch	718
A.2.3.3. Search Testing	718
A.2.3.4. Authentication Testing	719
A.2.3.5. Modify Operation Testing	720
A.2.3.6. Compare Operation Testing	721
A.2.3.7. Delete Operation Testing	721
A.2.3.8. Changing Time Limits	722
A.2.3.9. Bind Testing with Any Operation	723
A.2.3.10. Performing Multi-Threaded Testing	724
APPENDIX B. REPLICATION AGREEMENT STATUS	725
GLOSSARY	729
APPENDIX C. REVISION HISTORY	740

PREFACE

Reference guide for configuring Directory Server

LEGAL NOTICE

Copyright 2021 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux is the registered trademark of Linus Torvalds in the United States and other countries.

Java is a registered trademark of Oracle and/or its affiliates.

XFS is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The **OpenStack** Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

AUTHORS

Marc Muehlfeld

Red Hat Customer Content Services

mmuehlfeld@redhat.com

Petr Bokoč

Red Hat Customer Content Services

Tomáš Čapek

Red Hat Customer Content Services

Petr Kovář

Red Hat Customer Content Services

Ella Deon Ballard

Red Hat Customer Content Services

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

ABOUT THIS REFERENCE

Red Hat Directory Server (Directory Server) is a powerful and scalable distributed directory server based on the industry-standard Lightweight Directory Access Protocol (LDAP). Directory Server is the cornerstone for building a centralized and distributed data repository that can be used in an intranet, over an extranet with trading partners, or over the public Internet to reach customers.

This reference covers the server configuration and the command-line utilities. It is designed primarily for directory administrators and experienced directory users who want to use the command-line to access the directory. After configuring the server, use this reference to help maintain it.

The Directory Server can also be managed through the Directory Server Console, a graphical user interface. The *Red Hat Directory Server Administration Guide* describes how to do this and explains individual administration tasks more fully.

1. DIRECTORY SERVER OVERVIEW

The major components of Directory Server include:

- An LDAP server – The LDAP v3-compliant network daemon.
- Directory Server Console – A graphical management console that dramatically reduces the effort of setting up and maintaining your directory service.
- SNMP agent – Can monitor the Directory Server using the Simple Network Management Protocol (SNMP).

CHAPTER 1. INTRODUCTION

Directory Server is based on an open-systems server protocol called the Lightweight Directory Access Protocol (LDAP). The Directory Server is a robust, scalable server designed to manage large scale directories to support an enterprise-wide directory of users and resources, extranets, and e-commerce applications over the Internet. The Directory Server runs as the **ns-slapd** process or service on the machine. The server manages the directory databases and responds to client requests.

Most Directory Server administrative tasks can be performed through the Directory Server Console, the graphical user interface provided with the Directory Server. For information on the use of the Directory Server Console, see the *Red Hat Directory Server Administration Guide*.

This reference deals with the other methods of managing the Directory Server by altering the server configuration attributes using the command line and using command-line utilities and scripts.

1.1. DIRECTORY SERVER CONFIGURATION

The format and method for storing configuration information for Directory Server and a listing for all server attributes are found in two chapters, [Chapter 3, Core Server Configuration Reference](#) and [Chapter 4, Plug-in Implemented Server Functionality Reference](#).

1.2. DIRECTORY SERVER INSTANCE FILE REFERENCE

[Section 2.1, “Directory Server Instance-independent Files and Directories”](#) has an overview of the files and configuration information stored in each instance of Directory Server. This is useful reference to helps administrators understand the changes or absence of changes in the course of directory activity. From a security standpoint, this also helps users detect errors and intrusion by highlighting normal changes and abnormal behavior.

1.3. USING DIRECTORY SERVER COMMAND-LINE UTILITIES

Directory Server comes with a set of configurable command-line utilities that can search and modify entries in the directory and administer the server. [Chapter 9, Command-Line Utilities](#) describes these command-line utilities and contains information on where the utilities are stored and how to access them.

CHAPTER 2. FILE LOCATIONS OVERVIEW

Red Hat Directory Server is compatible with the Filesystem Hierarchy Standards (FHS). For further information on the FHS, see <http://refspecs.linuxfoundation.org/fhs.shtml>.

2.1. DIRECTORY SERVER INSTANCE-INDEPENDENT FILES AND DIRECTORIES

The following are the Directory Server's instance-independent default file and directory locations:

Type	Location
Command-line utilities	/usr/bin/ /usr/sbin/
Systemd unit files	/usr/lib/systemd/system/dirsrv.target /etc/systemd/system/dirsrv.target.wants/

2.2. DIRECTORY SERVER INSTANCE-SPECIFIC FILES AND DIRECTORIES

To separate multiple instances running on the same host, certain files and directories contain the name of the instance. You set the instance name during the Directory Server setup. By default, this is the host name without domain name. For example, if your fully-qualified domain name is **server.example.com**, the default instance name is **server**.

The following are the Directory Server's instance-specific default file and directory locations:

Type	Location
Backup files	/var/lib/dirsrv/slapd- <i>instance_name</i> /bak/
Configuration files	/etc/dirsrv/slapd- <i>instance_name</i> /
Certificate and key databases	/etc/dirsrv/slapd- <i>instance_name</i> /
Database files	/var/lib/dirsrv/slapd- <i>instance_name</i> /db/
LDIF files	/var/lib/dirsrv/slapd- <i>instance_name</i> /ldif/
Lock files	/var/lock/dirsrv/slapd- <i>instance_name</i> /
Log files	/var/log/dirsrv/slapd- <i>instance_name</i> /
PID file	/var/run/dirsrv/ <i>instance_name</i> .pid

Type	Location
Systemd unit files	/etc/systemd/system/dirsrv.target.wants/dirsrv@ <i>instance_name</i> .service

2.2.1. Configuration Files

Each Directory Server instance stores its configuration files in the **/etc/dirsrv/slappd-*instance*** directory.

The configuration information for Red Hat Directory Server is stored as LDAP entries within the directory itself. Therefore, changes to the server configuration must be implemented through the use of the server itself rather than by simply editing configuration files. The principal advantage of this method of configuration storage is that it allows a directory administrator to reconfigure the server using LDAP while it is still running, thus avoiding the need to shut the server down for most configuration changes.

2.2.1.1. Overview of the Directory Server Configuration

When the Directory Server is set up, its default configuration is stored as a series of LDAP entries within the directory, under the subtree **cn=config**. When the server is started, the contents of the **cn=config** subtree are read from a file (**dse.ldif**) in LDIF format. This **dse.ldif** file contains all of the server configuration information. The latest version of this file is called **dse.ldif**, the version prior to the last modification is called **dse.ldif.bak**, and the latest file with which the server successfully started is called **dse.ldif.startOK**.

Many of the features of the Directory Server are designed as discrete modules that plug into the core server. The details of the internal configuration for each plug-in are contained in separate entries under **cn=plugins,cn=config**. For example, the configuration of the Telephone Syntax Plug-in is contained in this entry:

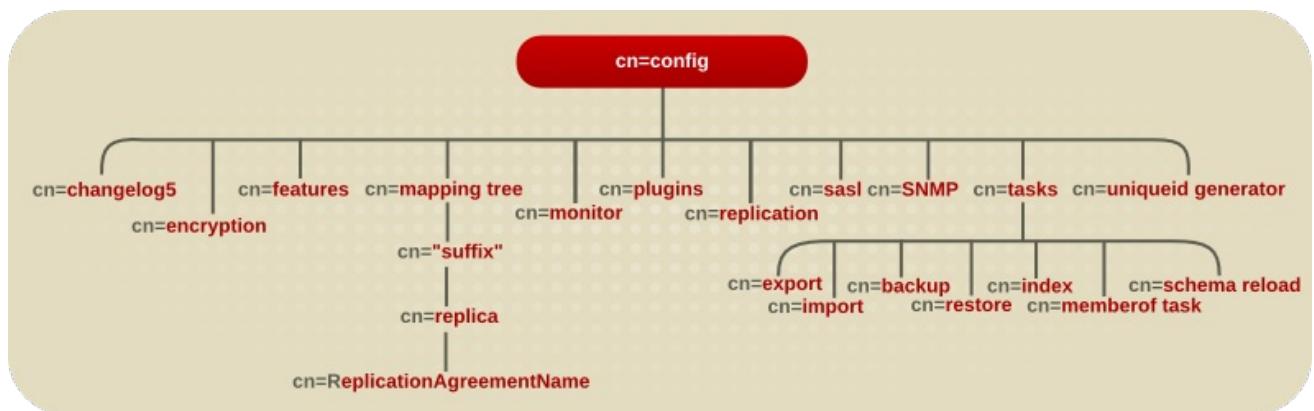
cn=Telephone Syntax,cn=plugins,cn=config

Similarly, database-specific configuration is stored under

cn=ldbm database,cn=plugins,cn=config for local databases and **cn=chaining database,cn=plugins,cn=config** for database links.

The following diagram illustrates how the configuration data fits within the **cn=config** directory information tree.

Figure 2.1. Directory Information Tree Showing Configuration Data



2.2.1.1. LDIF and Schema Configuration Files

The Directory Server configuration data are stored in LDIF files in the **/etc/dirsrv/slappd-instance** directory. Thus, if a server identifier is **phonebook**, then for a Directory Server, the configuration LDIF files are all stored under **/etc/dirsrv/slappd-phonebook**.

This directory also contains other server instance-specific configuration files.

Schema configuration is also stored in LDIF format, and these files are located in the **/etc/dirsrv/schema** directory.

The following table lists all of the configuration files that are supplied with the Directory Server, including those for the schema of other compatible servers. Each file is preceded by a number which indicates the order in which they should be loaded (in ascending numerical and then alphabetical order).

Table 2.1. Directory Server LDIF Configuration Files

Configuration Filename	Purpose
dse.ldif	Contains front-end Directory Specific Entries created by the directory at server startup. These include the Root DSE ("") and the contents of cn=config and cn=monitor (acis only).
00core.ldif	Contains only those schema definitions necessary for starting the server with the bare minimum feature set (no user schema, no schema for any non-core features). The rest of the schema used by users, features, and applications is found in 01common.ldif and the other schema files. Do not modify this file.
01common.ldif	Contains LDAPv3 standard operational schema, such as subschemaSubentry , LDAPv3 standard user and organization schema defined in RFC 2256 (based on X.520/X.521), inetOrgPerson and other widely-used attributes, and the operational attributes used by Directory Server configuration. Modifying this file causes interoperability problems. User-defined attributes should be added through the Directory Server Console.
05rfc2247.ldif	Schema from RFC 2247 and related pilot schema, from "Using Domains in LDAP/X500 Distinguished Names."
05rfc2927.ldif	Schema from RFC 2927, "MIME Directory Profile for LDAP Schema." Contains the ldapSchemas operational attribute required for the attribute to show up in the subschema subentry.

Configuration Filename	Purpose
10presence.ldif	Legacy. Schema for instant messaging presence (online) information; the file lists the default object classes with the allowed attributes that must be added to a user's entry in order for instant-messaging presence information to be available for that user.
10rfc2307.ldif	Schema from RFC 2307, "An Approach for Using LDAP as a Network Information Service." This may be superseded by 10rfc2307bis , the new version of rfc2307 , when that schema becomes available.
20subscriber.ldif	Contains new schema elements and the Nortel subscriber interoperability specification. Also contains the adminRole and memberOf attributes and inetAdmin object class, previously stored in the 50ns-delegated-admin.ldif file.
25java-object.ldif	Schema from RFC 2713, "Schema for Representing Java® Objects in an LDAP Directory."
28pilot.ldif	Contains pilot directory schema from RFC 1274, which is no longer recommended for new deployments. Future RFCs which succeed RFC 1274 may deprecate some or all of 28pilot.ldif attribute types and classes.
30ns-common.ldif	Schema that contains objects classes and attributes common to the Directory Server Console framework.
50ns-admin.ldif	Schema used by Red Hat Administration Server.
50ns-certificate.ldif	Schema for Red Hat Certificate Management System.
50ns-directory.ldif	Contains additional configuration schema used by Directory Server 4.12 and earlier versions of the directory, which is no longer applicable to current releases of Directory Server. This schema is required for replicating between Directory Server 4.12 and current releases.
50ns-mail.ldif	Schema used by Netscape Messaging Server to define mail users and mail groups.
50ns-value.ldif	Schema for servers' value item attributes.
50ns-web.ldif	Schema for Netscape Web Server.

Configuration Filename	Purpose
60pam-plugin.ldif	Reserved for future use.
99user.ldif	User-defined schema maintained by Directory Server replication consumers which contains the attributes and object classes from the suppliers.

2.2.1.1.2. How the Server Configuration Is Organized

The **dse.ldif** file contains all configuration information including directory-specific entries created by the directory at server startup, such as entries related to the database. The file includes the root Directory Server entry (or DSE, named by "") and the contents of **cn=config** and **cn=monitor**.

When the server generates the **dse.ldif** file, it lists the entries in hierarchical order in the order that the entries appear in the directory under **cn=config**, which is usually the same order in which an LDAP search of subtree scope for base **cn=config** returns the entries.

dse.ldif also contains the **cn=monitor** entry, which is mostly read-only, but can have ACIs set on it.



NOTE

The **dse.ldif** file does not contain every attribute in **cn=config**. If the attribute has not been set by the administrator and has a default value, the server will not write it to **dse.ldif**. To see every attribute in **cn=config**, use **ldapsearch**.

Configuration Attributes

Within a configuration entry, each attribute is represented as an attribute name. The value of the attribute corresponds to the attribute's configuration.

The following code sample is an example of part of the **dse.ldif** file for a Directory Server. The example shows, among other things, that schema checking has been enabled; this is represented by the attribute **nsslapd-schemacheck**, which takes the value **on**.

```
dn: cn=config
objectclass: top
objectclass: extensibleObject
objectclass: nsslapdConfig
nsslapd-accesslog-logging-enabled: on
nsslapd-enquote-sup-oc: off
nsslapd-localhost: phonebook.example.com
nsslapd-schemacheck: on
nsslapd-port: 389
nsslapd-localuser: dirsrv
...
...
```

Configuration of Plug-in Functionality

The configuration for each part of Directory Server plug-in functionality has its own separate entry and set of attributes under the subtree **cn=plugins,cn=config**. The following code sample is an example of the configuration entry for an example plug-in, the Telephone Syntax plug-in.

```

dn: cn=Telephone Syntax,cn=plugins,cn=config
objectclass: top
objectclass: nsSlapdPlugin
objectclass: extensibleObject
cn: Telephone Syntax
nsslapd-pluginType: syntax
nsslapd-pluginEnabled: on

```

Some of these attributes are common to all plug-ins, and some may be particular to a specific plug-in. Check which attributes are currently being used by a given plug-in by performing an **ldapsearch** on the **cn=config** subtree.

For a list of plug-ins supported by Directory Server, general plug-in configuration information, the plug-in configuration attribute reference, and a list of plug-ins requiring restart for configuration changes, see [Chapter 4, Plug-in Implemented Server Functionality Reference](#).

Configuration of Databases

The **cn=UserRoot** subtree under the database plug-in entry contain configuration data for the databases containing the default suffix created during setup.

These entries and their children have many attributes used to configure different database settings, like the cache sizes, the paths to the index files and transaction logs, entries and attributes for monitoring and statistics; and database indexes.

Configuration of Indexes

Configuration information for indexing is stored as entries in the Directory Server under the following information-tree nodes:

- **cn=index,cn=UserRoot,cn=ldbm database,cn=plugins,cn=config**
- **cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config**

For more information about indexes in general, see the *Red Hat Directory Server Administration Guide*. For information about the index configuration attributes, see [Section 4.4.1, “Database Attributes under cn=config,cn=ldbm database,cn=plugins,cn=config”](#).

2.2.1.2. Accessing and Modifying Server Configuration

This section discusses access control for configuration entries and describes the various ways in which the server configuration can be viewed and modified. It also covers restrictions to the kinds of modification that can be made and discusses attributes that require the server to be restarted for changes to take effect.

2.2.1.2.1. Access Control for Configuration Entries

When the Directory Server is installed, a default set of access control instructions (ACIs) is implemented for all entries under **cn=config**. The following code sample is an example of these default ACIs.

```

aci: (targetattr = "")(version 3.0; acl "Local Directory Administrators Group"; allow (all)
groupdn = "ldap://ou=Directory Administrators,dc=example,dc=com";)

```

These default ACIs allow all LDAP operations to be carried out on all configuration attributes by the following users:

- Members of the Configuration Administrators group.
- The user acting as the administrator, the **admin** account that was configured at setup. By default, this is the same user account which is logged into the Console.
- Members of local Directory Administrators group.
- The SIE (Server Instance Entry) group, usually assigned using the **Set Access Permissions** process the main console.

For more information on access control, see the *Red Hat Directory Server Administration Guide*.

2.2.1.2.2. Changing Configuration Attributes

Server attributes can be viewed and changed in one of three ways: through the Directory Server Console, by performing **Idapsearch** and **Idapmodify** commands, or by manually editing the **dse.ldif** file.



NOTE

Before editing the **dse.ldif** file, the server *must* be stopped; otherwise, the changes are lost. Editing the **dse.ldif** file is recommended only for changes to attributes which cannot be altered dynamically. See [Configuration Changes Requiring Server Restart](#) for further information.

The following sections describe how to modify entries using LDAP (both by using Directory Server Console and by using the command line), the restrictions that apply to modifying entries, the restrictions that apply to modifying attributes, and the configuration changes requiring restart.

Modifying Configuration Entries Using LDAP

The configuration entries in the directory can be searched and modified using LDAP either using the Directory Server Console or by performing **Idapsearch** and **Idapmodify** operations in the same way as other directory entries. The advantage of using LDAP to modify entries is changes can be made while the server is running.

For further information, see the "Creating Directory Entries" chapter in the *Red Hat Directory Server Administration Guide*. However, certain changes do require the server to be restarted before they are taken into account. See [Configuration Changes Requiring Server Restart](#) for further information.



NOTE

As with any set of configuration files, care should be taken when changing or deleting nodes in the **cn=config** subtree as this risks affecting Directory Server functionality.

The entire configuration, including attributes that always take default values, can be viewed by performing an **Idapsearch** operation on the **cn=config** subtree:

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b "cn=config" -s sub -x "(objectclass=*)"
```

- *bindDN* is the DN chosen for the Directory Manager when the server was installed (**cn=Directory Manager** by default).

- *password* is the password chosen for the Directory Manager.

To disable a plug-in, use **ldapmodify** to edit the **nsslapd-pluginEnabled** attribute:

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x
dn: cn=Telephone Syntax,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: off
```

Restrictions to Modifying Configuration Entries and Attributes

Certain restrictions apply when modifying server entries and attributes:

- The **cn=monitor** entry and its child entries are read-only and cannot be modified, except to manage ACIs.
- If an attribute is added to **cn=config**, the server ignores it.
- If an invalid value is entered for an attribute, the server ignores it.
- Because **ldapdelete** is used for deleting an entire entry, use **ldapmodify** to remove an attribute from an entry.

Configuration Changes Requiring Server Restart

Some configuration attributes cannot be altered while the server is running. In these cases, for the changes to take effect, the server needs to be shut down and restarted. The modifications should be made either through the Directory Server Console or by manually editing the **dse.ldif** file. Some of the attributes that require a server restart for any changes to take effect are listed below. This list is not exhaustive; to see a complete list, run **ldapsearch** and search for the **nsslapd-requiresrestart** attribute. For example:

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b "cn=config" -s sub -x "
(objectclass=*)" | grep nsslapd-requiresrestart
```

nsslapd-cachesize	nsslapd-certdir
nsslapd-db cachesize	nsslapd-dbncache
nsslapd-plugin	nsslapd-changelogdir
nsslapd-changelogmaxage	nsslapd-changelogmaxentries
nsslapd-port	nsslapd-schemadir
nsslapd-saslpath	nsslapd-securereport
nsslapd-tmpdir	nsSSLclientauth
nsSSLSessionTimeout	nsslapd-conntablesize

nsslapd-lockdir	nsslapd-maxdescriptors
nsslapd-reservedescriptors	nsslapd-listenhost
nsslapd-schema-ignore-trailing-spaces	nsslapd-securelistenhost
nsslapd-workingdir	nsslapd-return-exact-case
nsslapd-maxbersize ^[a]	

[a] Although this attribute requires a restart, it is not returned in the search.

Deleting Configuration Attributes

All core configuration attributes are present, even if they are not written in the `/etc/dirsrv/slapd-instance-name/dse.ldif` file, because they all have default values used by the server.

For details about deleting core configuration attributes and a list of attributes that cannot be deleted, see the corresponding section in the [Red Hat Directory Server Administration Guide](#).

2.2.2. Database Files

Each Directory Server instance contains the `/var/lib/dirsrv/slapd-instance/db` directory for storing all of the database files. The following is a sample listing of the `/var/lib/dirsrv/slapd-instance/db` directory contents.

Example 2.1. Database Directory Contents

```
db.001 db.002 __db.003 DBVERSION log.0000000001 userroot/
```

- **db.00x** files – Used internally by the database and should not be moved, deleted, or modified in any way.
- **log.xxxxxxxxxxx** files – Used to store the transaction logs per database.
- **DBVERSION** – Used for storing the version of the database.
- **userRoot** – Stores the user-defined suffix (user-defined databases) created at setup; for example, **dc=example,dc=com**.



NOTE

If a new database is created (for example, **testRoot**) to store the directory tree under a new suffix, the directory named **testRoot** also appears in the `/var/lib/dirsrv/slapd-instance/db` directory.

The following is a sample listing of the **userRoot** directory contents.

Example 2.2. userroot Database Directory Contents

```
ancestorid.db
DBVERSION
entryrdn.db
id2entry.db
nsuniqueid.db
numsubordinates.db
objectclass.db
parentid.db
```

The **userroot** subdirectory contains the following files:

- **ancestorid.db** – Contains a list of IDs to find the ID of the entry's ancestor.
- **entrydn.db** – Contains a list of full DNs to find any ID.
- **id2entry.db** – Contains the actual directory database entries. All other database files can be recreated from this one, if necessary.
- **nsuniqueid.db** – Contains a list of unique IDs to find any ID.
- **numsubordinates.db** – Contains IDs that have child entries.
- **objectclass.db** – Contains a list of IDs which have a particular object class.
- **parentid.db** – Contains a list of IDs to find the ID of the parent.

2.2.3. LDIF Files

Sample LDIF files are stored in the **/var/lib/dirsrv/slappd-instance/ldif** directory for storing LDIF-related files. [Example 2.3, “LDIF Directory Contents”](#) lists the **/ldif** directory contents.

Example 2.3. LDIF Directory Contents

```
European.ldif
Example.ldif
Example-roles.ldif
Example-views.ldif
```

- **European.ldif** – Contains European character samples.
- **Example.ldif** – Is a sample LDIF file.
- **Example-roles.ldif** – Is a sample LDIF file similar to **Example.ldif**, except that it uses roles and class of service instead of groups for setting access control and resource limits for directory administrators.

**NOTE**

The LDIF files exported by **db2ldif** or **db2ldif.pl** scripts in the instance directory are stored in **/var/lib/dirsrv/slappd-instance/ldif**.

2.2.4. Lock Files

Each Directory Server instance contains a **/var/lock/dirsrv/slappd-instance** directory for storing lock-related files. The following is a sample listing of the **locks** directory contents.

Example 2.4. Lock Directory Contents

```
exports/ imports/ server/
```

The lock mechanisms control how many copies of the Directory Server process can be running at one. For example, if there is an import job, then a lock is placed in the **imports/** directory to prevent any other **ns-slappd** (normal), **ldif2db** (another import), or **db2ldif** (export) operations from running. If the server is running as normal, there is a lock in the **server/** directory, which prevents import operations (but not export operations), while if there is an export operation, the lock in the **exports/** directory allows normal server operations but prevents import operations.

The number of available locks can affect overall Directory Server performance. The number of locks is set in the **nsslapd-db-locks** attribute. Tuning that attribute value is described in the *Performance Tuning Guide*.

2.2.5. Log Files

Each Directory Server instance contains a **/var/log/dirsrv/slappd-instance** directory for storing log files. The following is a sample listing of the **/logs** directory contents.

Example 2.5. Log Directory Contents

```
access          access.20200228-171925 errors
access.20200221-162824 access.rotationinfo   errors.20200221-162824
access.20200223-171949 audit           errors.rotationinfo
access.20200227-171818 audit.rotationinfo slapd.stats
```

- The content of the **access**, **audit**, and **error** log files is dependent on the log configuration.
- The **slapd.stats** file is a memory-mapped file which cannot be read by an editor. It contains data collected by the Directory Server SNMP data collection component. This data is read by the SNMP subagent in response to SNMP attribute queries and is communicated to the SNMP master agent responsible for handling Directory Server SNMP requests.

[Chapter 7, Log File Reference](#) contains a solid overview of the access, error, and audit log file formats and the information in them.

2.2.6. PID Files

slapd-serverID.pid and **slapd-serverID.startpid** files are created in the **/var/run/dirsrv** directory when the server is up and running. Both files store the server's process ID.

2.2.7. Backup Files

Each Directory Server instance contains the following directory and file for storing backup-related files:

- **/var/lib/dirsrv/slappd-*instance*/bak** – This contains a directory dated with the *instance*, time and date of the database backup, such as ***instance-2020_05_02_16_56_05/***, which in turn holds the database backup copy.
- **/etc/dirsrv/slappd-*instance*/dse_original.ldif** – This is a backup copy of the **dse.ldif** configuration file from the time of installation.

2.3. ADMINISTRATION SERVER FILES AND DIRECTORIES

The following are the Administration Server's default file and directory locations:

Type	Location
Log files	/var/log/dirsrv/admin-serv/
Configuration files	/etc/dirsrv/admin-serv/
Certificate and key databases	/etc/dirsrv/admin-serv/
Runtime files:	/var/run/dirsrv/admin-serv.*
Systemd unit file	/etc/systemd/system/multi-user.target.wants/dirsrv-admin.service
Command-line Utilities	/usr/bin/ /usr/sbin/

CHAPTER 3. CORE SERVER CONFIGURATION REFERENCE

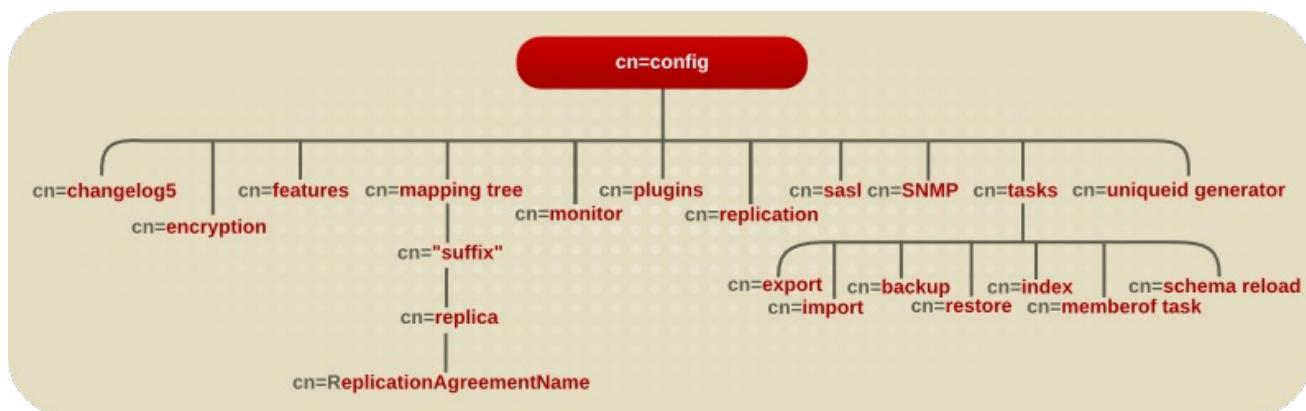
The chapter provides an alphabetical reference for all core (server-related) attributes. Section 2.2.1.1, “[Overview of the Directory Server Configuration](#)” contains a good overview of the Red Hat Directory Server configuration files.

3.1. CORE SERVER CONFIGURATION ATTRIBUTES REFERENCE

This section contains reference information on the configuration attributes that are relevant to the core server functionality. For information on changing server configuration, see [Section 2.2.1.2, “Accessing and Modifying Server Configuration”](#). For a list of server features that are implemented as plug-ins, see [Section 4.1, “Server Plug-in Functionality Reference”](#). For help with implementing custom server functionality, contact Directory Server support.

The configuration information stored in the **dse.ldif** file is organized as an information tree under the general configuration entry **cn=config**, as shown in the following diagram.

Figure 3.1. Directory Information Tree Showing Configuration Data



Most of these configuration tree nodes are covered in the following sections.

The **cn=plugins** node is covered in [Chapter 4, Plug-in Implemented Server Functionality Reference](#). The description of each attribute contains details such as the DN of its directory entry, its default value, the valid range of values, and an example of its use.



NOTE

Some of the entries and attributes described in this chapter may change in future releases of the product.

3.1.1. cn=config

General configuration entries are stored in the **cn=config** entry. The **cn=config** entry is an instance of the **nsslapdConfig** object class, which in turn inherits from **extensibleObject** object class.

3.1.1.1. nsslapd-accesslog (Access Log)

This attribute specifies the path and filename of the log used to record each LDAP access. The following information is recorded by default in the log file:

- IP address (IPv4 or IPv6) of the client machine that accessed the database.
- Operations performed (for example, search, add, and modify).

- Result of the access (for example, the number of entries returned or an error code).

For more information on turning access logging off, see the "Monitoring Server and Database Activity" chapter in the *Red Hat Directory Server Administration Guide*.

For access logging to be enabled, this attribute must have a valid path and parameter, and the **nsslapd-accesslog-logging-enabled** configuration attribute must be switched to **on**. The table lists the four possible combinations of values for these two configuration attributes and their outcome in terms of disabling or enabling of access logging.

Table 3.1. dse.ldif File Attributes

Attribute	Value	Logging enabled or disabled
nsslapd-accesslog-logging-enabled nsslapd-accesslog	on empty string	Disabled
nsslapd-accesslog-logging-enabled nsslapd-accesslog	on <i>filename</i>	Enabled
nsslapd-accesslog-logging-enabled nsslapd-accesslog	off empty string	Disabled
nsslapd-accesslog-logging-enabled nsslapd-accesslog	off <i>filename</i>	Disabled

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid filename.
Default Value	/var/log/dirsrv/slapd- <i>instance</i> /access
Syntax	DirectoryString
Example	nsslapd-accesslog: /var/log/dirsrv/slapd- <i>instance</i> /access

3.1.1.2. nsslapd-accesslog-level (Access Log Level)

This attribute controls what is logged to the access log.

You do not have to restart the server for this setting to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	<ul style="list-style-type: none"> * 0 - No access logging * 4 - Logging for internal access operations * 256 - Logging for connections, operations, and results * 512 - Logging for access to an entry and referrals * These values can be added together to provide the exact type of logging required; for example, 516 (4 + 512) to obtain internal access operation, entry access, and referral logging.
Default Value	256
Syntax	Integer
Example	nsslapd-accesslog-level: 256

3.1.1.3. nsslapd-accesslog-list (List of Access Log Files)

This read-only attribute, which cannot be set, provides a list of access log files used in access log rotation.

Parameter	Description
Entry DN	cn=config
Valid Values	
Default Value	None
Syntax	DirectoryString
Example	nsslapd-accesslog-list: accesslog2,accesslog3

3.1.1.4. nsslapd-accesslog-logbuffering (Log Buffering)

When set to **off**, the server writes all access log entries directly to disk. Buffering allows the server to use access logging even when under a heavy load without impacting performance. However, when debugging, it is sometimes useful to disable buffering in order to see the operations and their results right away instead of having to wait for the log entries to be flushed to the file. Disabling log buffering can severely impact performance in heavily loaded servers.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-accesslog-logbuffering: off

3.1.1.5. nsslapd-accesslog-logexpirationtime (Access Log Expiration Time)

This attribute specifies the maximum age that a log file is allowed to reach before it is deleted. This attribute supplies only the number of units. The units are provided by the **nsslapd-accesslog-logexpirationtimeunit** attribute.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 to the maximum 32 bit integer value (2147483647) A value of -1 or 0 means that the log never expires.
Default Value	-1
Syntax	Integer
Example	nsslapd-accesslog-logexpirationtime: 2

3.1.1.6. nsslapd-accesslog-logexpirationtimeunit (Access Log Expiration Time Unit)

This attribute specifies the units for **nsslapd-accesslog-logexpirationtime** attribute. If the unit is unknown by the server, then the log never expires.

Parameter	Description
Entry DN	cn=config
Valid Values	month week day
Default Value	month
Syntax	DirectoryString

Parameter	Description
Example	nsslapd-accesslog-logexpirationtimeunit: week

3.1.1.7. nsslapd-accesslog-logging-enabled (Access Log Enable Logging)

Disables and enables accesslog logging but only in conjunction with the **nsslapd-accesslog** attribute that specifies the path and parameter of the log used to record each database access.

For access logging to be enabled, this attribute must be switched to **on**, and the **nsslapd-accesslog** configuration attribute must have a valid path and parameter. The table lists the four possible combinations of values for these two configuration attributes and their outcome in terms of disabling or enabling of access logging.

Table 3.2. dse.ldif Attributes

Attribute	Value	Logging Enabled or Disabled
nsslapd-accesslog-logging-enabled	on empty string	Disabled
nsslapd-accesslog		
nsslapd-accesslog-logging-enabled	on <i>filename</i>	Enabled
nsslapd-accesslog		
nsslapd-accesslog-logging-enabled	off empty string	Disabled
nsslapd-accesslog		
nsslapd-accesslog-logging-enabled	off <i>filename</i>	Disabled
nsslapd-accesslog		

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString

Parameter	Description
Example	nsslapd-accesslog-logging-enabled: off

3.1.1.8. nsslapd-accesslog-logmaxdiskspace (Access Log Maximum Disk Space)

This attribute specifies the maximum amount of disk space in megabytes that the access logs are allowed to consume. If this value is exceeded, the oldest access log is deleted.

When setting a maximum disk space, consider the total number of log files that can be created due to log file rotation. Also, remember that there are three different log files (access log, audit log, and error log) maintained by the Directory Server, each of which consumes disk space. Compare these considerations to the total amount of disk space for the access log.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647), where a value of -1 means that the disk space allowed to the access log is unlimited in size.
Default Value	-1
Syntax	Integer
Example	nsslapd-accesslog-logmaxdiskspace: 100000

3.1.1.9. nsslapd-accesslog-logminfreediskspace (Access Log Minimum Free Disk Space)

This attribute sets the minimum allowed free disk space in megabytes. When the amount of free disk space falls below the value specified on this attribute, the oldest access logs are deleted until enough disk space is freed to satisfy this attribute.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647)
Default Value	-1
Syntax	Integer
Example	nsslapd-accesslog-logminfreediskspace: -1

3.1.1.10. nsslapd-accesslog-logrotationsync-enabled (Access Log Rotation Sync Enabled)

This attribute sets whether access log rotation is to be synchronized with a particular time of the day. Synchronizing log rotation this way can generate log files at a specified time during a day, such as midnight to midnight every day. This makes analysis of the log files much easier because they then map directly to the calendar.

For access log rotation to be synchronized with time-of-day, this attribute must be enabled with the **nsslapd-accesslog-logrotationsynchour** and **nsslapd-accesslog-logrotationsyncmin** attribute values set to the hour and minute of the day for rotating log files.

For example, to rotate access log files every day at midnight, enable this attribute by setting its value to **on**, and then set the values of the **nsslapd-accesslog-logrotationsynchour** and **nsslapd-accesslog-logrotationsyncmin** attributes to **0**.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-accesslog-logrotationsync-enabled: on

3.1.1.11. nsslapd-accesslog-logrotationsynchour (Access Log Rotation Sync Hour)

This attribute sets the hour of the day for rotating access logs. This attribute must be used in conjunction with **nsslapd-accesslog-logrotationsync-enabled** and **nsslapd-accesslog-logrotationsyncmin** attributes.

Parameter	Description
Entry DN	cn=config
Valid Range	0 through 23
Default Value	0
Syntax	Integer
Example	nsslapd-accesslog-logrotationsynchour: 23

3.1.1.12. nsslapd-accesslog-logrotationsyncmin (Access Log Rotation Sync Minute)

This attribute sets the minute of the day for rotating access logs. This attribute must be used in conjunction with **nsslapd-accesslog-logrotationsync-enabled** and **nsslapd-accesslog-logrotationsynchour** attributes.

Parameter	Description
Entry DN	cn=config
Valid Range	0 through 59
Default Value	0
Syntax	Integer
Example	nsslapd-accesslog-logrotationsyncmin: 30

3.1.1.13. nsslapd-accesslog-logrotationtime (Access Log Rotation Time)

This attribute sets the time between access log file rotations. This attribute supplies only the number of units. The units (day, week, month, and so forth) are given by the **nsslapd-accesslog-logrotationtimeunit** attribute.

Directory Server rotates the log at the first write operation after the configured interval has expired, regardless of the size of the log.

Although it is not recommended for performance reasons to specify no log rotation since the log grows indefinitely, there are two ways of specifying this. Either set the **nsslapd-accesslog-maxlogsperdir** attribute value to **1** or set the **nsslapd-accesslog-logrotationtime** attribute to **-1**. The server checks the **nsslapd-accesslog-maxlogsperdir** attribute first, and, if this attribute value is larger than **1**, the server then checks the **nsslapd-accesslog-logrotationtime** attribute. See [Section 3.1.1.16, “nsslapd-accesslog-maxlogsperdir \(Access Log Maximum Number of Log Files\)”](#) for more information.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647), where a value of -1 means that the time between access log file rotation is unlimited.
Default Value	1
Syntax	Integer
Example	nsslapd-accesslog-logrotationtime: 100

3.1.1.14. nsslapd-accesslog-logrotationtimeunit (Access Log Rotation Time Unit)

This attribute sets the units for the **nsslapd-accesslog-logrotationtime** attribute.

Parameter	Description
Entry DN	cn=config
Valid Values	month week day hour minute
Default Value	day
Syntax	DirectoryString
Example	nsslapd-accesslog-logrotationtimeunit: week

3.1.1.15. nsslapd-accesslog-maxlogsize (Access Log Maximum Log Size)

This attribute sets the maximum access log size in megabytes. When this value is reached, the access log is rotated. That means the server starts writing log information to a new log file. If the **nsslapd-accesslog-maxlogsperdir** attribute is set to **1**, the server ignores this attribute.

When setting a maximum log size, consider the total number of log files that can be created due to log file rotation. Also, remember that there are three different log files (access log, audit log, and error log) maintained by the Directory Server, each of which consumes disk space. Compare these considerations to the total amount of disk space for the access log.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647), where a value of -1 means the log file is unlimited in size.
Default Value	100
Syntax	Integer
Example	nsslapd-accesslog-maxlogsize: 100

3.1.1.16. nsslapd-accesslog-maxlogsperdir (Access Log Maximum Number of Log Files)

This attribute sets the total number of access logs that can be contained in the directory where the access log is stored. Each time the access log is rotated, a new log file is created. When the number of files contained in the access log directory exceeds the value stored in this attribute, then the oldest version of the log file is deleted. For performance reasons, Red Hat recommends *not* setting this value to **1** because the server does not rotate the log, and it grows indefinitely.

If the value for this attribute is higher than **1**, then check the **nsslapd-accesslog-logrotationtime** attribute to establish whether log rotation is specified. If the **nsslapd-accesslog-logrotationtime** attribute has a value of **-1**, then there is no log rotation. See [Section 3.1.1.13, “nsslapd-accesslog-logrotationtime \(Access Log Rotation Time\)”](#) for more information.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to the maximum 32 bit integer value (2147483647)
Default Value	10
Syntax	Integer
Example	nsslapd-accesslog-maxlogsperdir: 10

3.1.1.17. nsslapd-accesslog-mode (Access Log File Permission)

This attribute sets the access mode or file permission with which access log files are to be created. The valid values are any combination of **000** to **777** (these mirror the numbered or absolute UNIX file permissions). The value must be a 3-digit number, the digits varying from **0** through **7**:

- **0** - None
- **1** - Execute only
- **2** - Write only
- **3** - Write and execute
- **4** - Read only
- **5** - Read and execute
- **6** - Read and write
- **7** - Read, write, and execute

In the 3-digit number, the first digit represents the owner's permissions, the second digit represents the group's permissions, and the third digit represents everyone's permissions. When changing the default value, remember that **000** does not allow access to the logs and that allowing write permissions to everyone can result in the logs being overwritten or deleted by anyone.

The newly configured access mode only affects new logs that are created; the mode is set when the log rotates to a new file.

Parameter	Description
Entry DN	cn=config
Valid Range	000 through 777
Default Value	600
Syntax	Integer

Parameter	Description
Example	nsslapd-accesslog-mode: 600

3.1.1.18. nsslapd-allow-anonymous-access

If a user attempts to connect to the Directory Server without supplying any bind DN or password, this is an *anonymous bind*. Anonymous binds simplify common search and read operations, like checking the directory for a phone number or email address, by not requiring users to authenticate to the directory first.

However, there are risks with anonymous binds. Adequate ACIs must be in place to restrict access to sensitive information and to disallow actions like modifies and deletes. Additionally, anonymous binds can be used for denial of service attacks or for malicious people to gain access to the server.

Anonymous binds can be disabled to increase security (off). By default, anonymous binds are allowed (on) for search and read operations. This allows access to *regular directory entries*, which includes user and group entries as well as configuration entries like the root DSE. A third option, **rootdse**, allows anonymous search and read access to search the root DSE itself, but restricts access to all other directory entries.

Optionally, resource limits can be placed on anonymous binds using the **nsslapd-anonlimitsdn** attribute as described in [Section 3.1.22, “nsslapd-anonlimitsdn”](#).

Changes to this value will not take effect until the server is restarted.

Parameter	Description
Entry DN	cn=config
Valid Values	on off rootdse
Default Value	on
Syntax	DirectoryString
Example	nsslapd-allow-anonymous-access: on

3.1.1.19. nsslapd-allow-hashed-passwords

This parameter disables the pre-hashed password checks. By default, the Directory Server does not allow pre-hashed passwords to be set by anyone other than the Directory Manager. You can delegate this privilege to other users when you add them to the Password Administrators group. However in some scenarios, like when the replication partner already controls the pre-hashed passwords checking, this feature has to be disabled on the Directory Server.

Parameter	Description
Entry DN	cn=config

Parameter	Description
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-allow-hashed-passwords: off

3.1.1.20. nsslapd-allow-hashed-passwords

Unauthenticated binds are connections to Directory Server where a user supplies an empty password. Using the default settings, Directory Server denies access in this scenario for security reasons.



WARNING

Red Hat recommends not enabling unauthenticated binds. This authentication method enables users to bind without supplying a password as any account, including the Directory Manager. After the bind, the user can access all data with the permissions of the account used to bind.

You do not have to restart the server for this setting to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-allow-unauthenticated-binds: off

3.1.1.21. nsslapd-allowed-sasl-mechanisms

Per default, the root DSE lists all mechanisms the SASL library supports. However in some environments only certain ones are preferred. The **nsslapd-allowed-sasl-mechanisms** attribute allows you to enable only some defined SASL mechanisms.

The mechanism names must consist of uppercase letters, numbers, and underscores. Each mechanism can be separated by commas or spaces.

**NOTE**

The **EXTERNAL** mechanism is actually not used by any SASL plug-in. It is internal to the server, and is mainly used for TLS client authentication. Hence, the **EXTERNAL** mechanism cannot be restricted or controlled. It will always appear in the supported mechanisms list, regardless what is set in the **nsslapd-allowed-sasl-mechanisms** attribute.

This setting does not require a server restart to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid SASL mechanism
Default Value	None (all SASL mechanisms allowed)
Syntax	DirectoryString
Example	nsslapd-allowed-sasl-mechanisms: GSSAPI, DIGEST-MD5, OTP

3.1.1.22. nsslapd-anonlimitsdn

Resource limits can be set on authenticated binds. The resource limits can set a cap on how many entries can be searched in a single operation (**nsslapd-sizeLimit**), a time limit (**nsslapd-timelimit**) and time out period (**nsslapd-idletimeout**) for searches, and the total number of entries that can be searched (**nsslapd-lookthroughlimit**). These resource limits prevent denial of service attacks from tying up directory resources and improve overall performance.

Resource limits are set on a user entry. An anonymous bind, obviously, does not have a user entry associated with it. This means that resource limits usually do not apply to anonymous operations.

To set resource limits for anonymous binds, a template entry can be created, with the appropriate resource limits. The **nsslapd-anonlimitsdn** configuration attribute can then be added that points to this entry and applies the resource limits to anonymous binds.

Parameter	Description
Entry DN	cn=config
Valid Values	Any DN
Default Value	None
Syntax	DirectoryString

Parameter	Description
Example	nsslapd-anonlimitsdn: cn=anon template,ou=people,dc=example,dc=com

3.1.1.23. nsslapd-attribute-name-exceptions

This attribute allows non-standard characters in attribute names to be used for backwards compatibility with older servers, such as "_" in schema-defined attributes.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-attribute-name-exceptions: on

3.1.1.24. nsslapd-auditlog (Audit Log)

This attribute sets the path and filename of the log used to record changes made to each database.

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid filename
Default Value	/var/log/dirsrv/slapd- <i>instance</i> /audit
Syntax	DirectoryString
Example	nsslapd-auditlog: /var/log/dirsrv/slapd- <i>instance</i> /audit

For audit logging to be enabled, this attribute must have a valid path and parameter, and the **nsslapd-auditlog-logging-enabled** configuration attribute must be switched to **on**. The table lists the four possible combinations of values for these two configuration attributes and their outcome in terms of disabling or enabling of audit logging.

Table 3.3. Possible Combinations for nsslapd-auditlog

Attributes in dse.ldif	Value	Logging enabled or disabled
nsslapd-auditlog-logging-enabled	on	Disabled
nsslapd-auditlog	empty string	
nsslapd-auditlog-logging-enabled	on	Enabled
nsslapd-auditlog	<i>filename</i>	
nsslapd-auditlog-logging-enabled	off	Disabled
nsslapd-auditlog	empty string	
nsslapd-auditlog-logging-enabled	off	Disabled
nsslapd-auditlog	<i>filename</i>	

3.1.1.25. nsslapd-auditlog-list

Provides a list of audit log files.

Parameter	Description
Entry DN	cn=config
Valid Values	
Default Value	None
Syntax	DirectoryString
Example	nsslapd-auditlog-list: auditlog2,auditlog3

3.1.1.26. nsslapd-auditlog-logexpirationtime (Audit Log Expiration Time)

This attribute sets the maximum age that a log file is allowed to be before it is deleted. This attribute supplies only the number of units. The units (day, week, month, and so forth) are given by the **nsslapd-auditlog-logexpirationtimeunit** attribute.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 to the maximum 32 bit integer value (2147483647) A value of -1 or 0 means that the log never expires.

Parameter	Description
Default Value	-1
Syntax	Integer
Example	nsslapd-auditlog-logexpirationtime:1

3.1.1.27. nsslapd-auditlog-logexpirationtimeunit (Audit Log Expiration Time Unit)

This attribute sets the units for the **nsslapd-auditlog-logexpirationtime** attribute. If the unit is unknown by the server, then the log never expires.

Parameter	Description
Entry DN	cn=config
Valid Values	month week day
Default Value	week
Syntax	DirectoryString
Example	nsslapd-auditlog-logexpirationtimeunit: day

3.1.1.28. nsslapd-auditlog-logging-enabled (Audit Log Enable Logging)

Turns audit logging on and off.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-auditlog-logging-enabled: off

For audit logging to be enabled, this attribute must have a valid path and parameter and the **nsslapd-auditlog-logging-enabled** configuration attribute must be switched to **on**. The table lists the four possible combinations of values for these two configuration attributes and their outcome in terms of disabling or enabling of audit logging.

Table 3.4. Possible combinations for nsslapd-auditlog and nsslapd-auditlog-logging-enabled

Attribute	Value	Logging enabled or disabled
nsslapd-auditlog-logging-enabled	on	Disabled
nsslapd-auditlog	empty string	
nsslapd-auditlog-logging-enabled	on	Enabled
nsslapd-auditlog	<i>filename</i>	
nsslapd-auditlog-logging-enabled	off	Disabled
nsslapd-auditlog	empty string	
nsslapd-auditlog-logging-enabled	off	Disabled
nsslapd-auditlog	<i>filename</i>	

3.1.1.29. nsslapd-auditlog-logmaxdiskspace (Audit Log Maximum Disk Space)

This attribute sets the maximum amount of disk space in megabytes that the audit logs are allowed to consume. If this value is exceeded, the oldest audit log is deleted.

When setting a maximum disk space, consider the total number of log files that can be created due to log file rotation. Also remember that there are three different log files (access log, audit log, and error log) maintained by the Directory Server, each of which consumes disk space. Compare these considerations with the total amount of disk space for the audit log.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647), where a value of -1 means that the disk space allowed to the audit log is unlimited in size.
Default Value	-1
Syntax	Integer
Example	nsslapd-auditlog-logmaxdiskspace: 10000

3.1.1.30. nsslapd-auditlog-logminfreediskspace (Audit Log Minimum Free Disk Space)

This attribute sets the minimum permissible free disk space in megabytes. When the amount of free disk space falls below the value specified by this attribute, the oldest audit logs are deleted until enough disk space is freed to satisfy this attribute.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 (unlimited) 1 to the maximum 32 bit integer value (2147483647)
Default Value	-1
Syntax	Integer
Example	nsslapd-auditlog-logminfreediskspace: -1

3.1.1.31. nsslapd-auditlog-logrotationsync-enabled (Audit Log Rotation Sync Enabled)

This attribute sets whether audit log rotation is to be synchronized with a particular time of the day. Synchronizing log rotation this way can generate log files at a specified time during a day, such as midnight to midnight every day. This makes analysis of the log files much easier because they then map directly to the calendar.

For audit log rotation to be synchronized with time-of-day, this attribute must be enabled with the **nsslapd-auditlog-logrotationsynchour** and **nsslapd-auditlog-logrotationsyncmin** attribute values set to the hour and minute of the day for rotating log files.

For example, to rotate audit log files every day at midnight, enable this attribute by setting its value to **on**, and then set the values of the **nsslapd-auditlog-logrotationsynchour** and **nsslapd-auditlog-logrotationsyncmin** attributes to **0**.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-auditlog-logrotationsync-enabled: on

3.1.1.32. nsslapd-auditlog-logrotationsynchour (Audit Log Rotation Sync Hour)

This attribute sets the hour of the day for rotating audit logs. This attribute must be used in conjunction with **nsslapd-auditlog-logrotationsync-enabled** and **nsslapd-auditlog-logrotationsyncmin** attributes.

Parameter	Description
Entry DN	cn=config
Valid Range	0 through 23
Default Value	None (because nsslapd-auditlog-logrotationsync-enabled is off)
Syntax	Integer
Example	nsslapd-auditlog-logrotationsynchour: 23

3.1.1.33. nsslapd-auditlog-logrotationsyncmin (Audit Log Rotation Sync Minute)

This attribute sets the minute of the day for rotating audit logs. This attribute must be used in conjunction with **nsslapd-auditlog-logrotationsync-enabled** and **nsslapd-auditlog-logrotationsynchour** attributes.

Parameter	Description
Entry DN	cn=config
Valid Range	0 through 59
Default Value	None (because nsslapd-auditlog-logrotationsync-enabled is off)
Syntax	Integer
Example	nsslapd-auditlog-logrotationsyncmin: 30

3.1.1.34. nsslapd-auditlog-logrotationtime (Audit Log Rotation Time)

This attribute sets the time between audit log file rotations. This attribute supplies only the number of units. The units (day, week, month, and so forth) are given by the **nsslapd-auditlog-logrotationtimeunit** attribute. If the **nsslapd-auditlog-maxlogsperdir** attribute is set to **1**, the server ignores this attribute.

Directory Server rotates the log at the first write operation after the configured interval has expired, regardless of the size of the log.

Although it is not recommended for performance reasons to specify no log rotation, as the log grows indefinitely, there are two ways of specifying this. Either set the **nsslapd-auditlog-maxlogsperdir** attribute value to **1** or set the **nsslapd-auditlog-logrotationtime** attribute to **-1**. The server checks the **nsslapd-auditlog-maxlogsperdir** attribute first, and, if this attribute value is larger than **1**, the server then checks the **nsslapd-auditlog-logrotationtime** attribute. See [Section 3.1.1.37, “nsslapd-auditlog-maxlogsperdir \(Audit Log Maximum Number of Log Files\)”](#) for more information.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647), where a value of -1 means that the time between audit log file rotation is unlimited.
Default Value	1
Syntax	Integer
Example	nsslapd-auditlog-logrotationtime: 100

3.1.1.35. nsslapd-auditlog-logrotationtimeunit (Audit Log Rotation Time Unit)

This attribute sets the units for the **nsslapd-auditlog-logrotationtime** attribute.

Parameter	Description
Entry DN	cn=config
Valid Values	month week day hour minute
Default Value	week
Syntax	DirectoryString
Example	nsslapd-auditlog-logrotationtimeunit: day

3.1.1.36. nsslapd-auditlog-maxlogsize (Audit Log Maximum Log Size)

This attribute sets the maximum audit log size in megabytes. When this value is reached, the audit log is rotated. That means the server starts writing log information to a new log file. If **nsslapd-auditlog-maxlogsperdir** to **1**, the server ignores this attribute.

When setting a maximum log size, consider the total number of log files that can be created due to log file rotation. Also, remember that there are three different log files (access log, audit log, and error log) maintained by the Directory Server, each of which consumes disk space. Compare these considerations to the total amount of disk space for the audit log.

Parameter	Description
Entry DN	cn=config

Parameter	Description
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647), where a value of -1 means the log file is unlimited in size.
Default Value	100
Syntax	Integer
Example	nsslapd-auditlog-maxlogsize: 50

3.1.1.37. nsslapd-auditlog-maxlogsize (Audit Log Maximum Number of Log Files)

This attribute sets the total number of audit logs that can be contained in the directory where the audit log is stored. Each time the audit log is rotated, a new log file is created. When the number of files contained in the audit log directory exceeds the value stored on this attribute, then the oldest version of the log file is deleted. The default is **1** log. If this default is accepted, the server will not rotate the log, and it grows indefinitely.

If the value for this attribute is higher than **1**, then check the **nsslapd-auditlog-logrotationtime** attribute to establish whether log rotation is specified. If the **nsslapd-auditlog-logrotationtime** attribute has a value of **-1**, then there is no log rotation. See [Section 3.1.1.34, “nsslapd-auditlog-logrotationtime \(Audit Log Rotation Time\)”](#) for more information.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to the maximum 32 bit integer value (2147483647)
Default Value	1
Syntax	Integer
Example	nsslapd-auditlog-maxlogsize: 10

3.1.1.38. nsslapd-auditlog-mode (Audit Log File Permission)

This attribute sets the access mode or file permissions with which audit log files are to be created. The valid values are any combination of **000** to **777** since they mirror numbered or absolute UNIX file permissions. The value must be a combination of a 3-digit number, the digits varying from **0** through **7**:

- 0 – None
- 1 – Execute only
- 2 – Write only
- 3 – Write and execute

- 4 - Read only
- 5 - Read and execute
- 6 - Read and write
- 7 - Read, write, and execute

In the 3-digit number, the first digit represents the owner's permissions, the second digit represents the group's permissions, and the third digit represents everyone's permissions. When changing the default value, remember that **000** does not allow access to the logs and that allowing write permissions to everyone can result in the logs being overwritten or deleted by anyone.

The newly configured access mode only affects new logs that are created; the mode is set when the log rotates to a new file.

Parameter	Description
Entry DN	cn=config
Valid Range	000 through 777
Default Value	600
Syntax	Integer
Example	nsslapd-auditlog-mode: 600

3.1.1.39. nsslapd-auditfaillog (Audit Fail Log)

This attribute sets the path and filename of the log used to record failed LDAP modifications.

If **nsslapd-auditfaillog-logging-enabled** is enabled, and **nsslapd-auditfaillog** is not set, the audit fail events are logged to the file specified in **nsslapd-auditlog**.

If you set the **nsslapd-auditfaillog** parameter to the same path as **nsslapd-auditlog**, both are logged in the same file.

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid filename
Default Value	/var/log/dirsrv/slapd- <i>instance</i> /audit
Syntax	DirectoryString
Example	nsslapd-auditfaillog: /var/log/dirsrv/slapd- <i>instance</i> /audit

To enable the audit fail log, this attribute must have a valid path and the **nsslapd-auditfaillog-logging-enabled** attribute must be set to **on**

3.1.1.40. nsslapd-auditfaillog-list

Provides a list of audit fail log files.

Parameter	Description
Entry DN	cn=config
Valid Values	
Default Value	None
Syntax	DirectoryString
Example	nsslapd-auditfaillog-list: auditfaillog2,auditfaillog3

3.1.1.41. nsslapd-auditfaillog-logexpirationtime (Audit Fail Log Expiration Time)

This attribute sets the maximum age of a log file before it is removed. It supplies to the number of units. Specify the units, such as day, week, month, and so forth in the **nsslapd-auditfaillog-logexpirationtimeunit** attribute.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 to the maximum 32 bit integer value (2147483647) A value of -1 or 0 means that the log never expires.
Default Value	-1
Syntax	Integer
Example	nsslapd-auditfaillog-logexpirationtime: 1

3.1.1.42. nsslapd-auditfaillog-logexpirationtimeunit (Audit Fail Log Expiration Time Unit)

This attribute sets the units for the **nsslapd-auditfaillog-logexpirationtime** attribute. If the unit is unknown by the server, the log never expires.

Parameter	Description
Entry DN	cn=config

Parameter	Description
Valid Values	month week day
Default Value	week
Syntax	DirectoryString
Example	nsslapd-auditfaillog-logexpirationtimeunit: day

3.1.1.43. nsslapd-auditfaillog-logging-enabled (Audit Fail Log Enable Logging)

Turns on and off logging of failed LDAP modifications.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-auditfaillog-logging-enabled: off

3.1.1.44. nsslapd-auditfaillog-logmaxdiskspace (Audit Fail Log Maximum Disk Space)

This attribute sets the maximum amount of disk space in megabytes the audit fail logs are can consume. If the size exceed the limit, the oldest audit fail log is deleted.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647), where a value of -1 means that the disk space allowed to the audit fail log is unlimited in size.
Default Value	-1
Syntax	Integer
Example	nsslapd-auditfaillog-logmaxdiskspace: 10000

3.1.1.45. nsslapd-auditfaillog-logminfreediskspace (Audit Fail Log Minimum Free Disk Space)

This attribute sets the minimum permissible free disk space in megabytes. When the amount of free disk space is lower than the specified value, the oldest audit fail logs are deleted until enough disk space is freed.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 (unlimited) 1 to the maximum 32 bit integer value (2147483647)
Default Value	-1
Syntax	Integer
Example	nsslapd-auditfaillog-logminfreediskspace: -1

3.1.1.46. nsslapd-auditfaillog-logrotationsync-enabled (Audit Fail Log Rotation Sync Enabled)

This attribute sets whether audit fail log rotation is to be synchronized with a particular time of the day. Synchronizing log rotation this way can generate log files at a specified time during a day, such as midnight to midnight every day. This makes analysis of the log files much easier because they then map directly to the calendar.

For audit fail log rotation to be synchronized with time-of-day, this attribute must be enabled with the **nsslapd-auditfaillog-logrotationsynchour** and **nsslapd-auditfaillog-logrotationsyncmin** attribute values set to the hour and minute of the day for rotating log files.

For example, to rotate audit fail log files every day at midnight, enable this attribute by setting its value to **on**, and then set the values of the **nsslapd-auditfaillog-logrotationsynchour** and **nsslapd-auditfaillog-logrotationsyncmin** attributes to **0**.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-auditfaillog-logrotationsync-enabled: on

3.1.1.47. nsslapd-auditfaillog-logrotationsynchour (Audit Fail Log Rotation Sync Hour)

This attribute sets the hour of the day the audit fail log is rotated. This attribute must be used in conjunction with **nsslapd-auditfaillog-logrotationsync-enabled** and **nsslapd-auditfaillog-logrotationsyncmin** attributes.

Parameter	Description
Entry DN	cn=config
Valid Range	0 through 23
Default Value	None (because nsslapd-auditfaillog-logrotationsync-enabled is off)
Syntax	Integer
Example	nsslapd-auditfaillog-logrotationsynchour: 23

3.1.1.48. nsslapd-auditfaillog-logrotationsyncmin (Audit Fail Log Rotation Sync Minute)

This attribute sets the minute the audit fail log is rotated. This attribute must be used in conjunction with **nsslapd-auditfaillog-logrotationsync-enabled** and **nsslapd-auditfaillog-logrotationsynchour** attributes.

Parameter	Description
Entry DN	cn=config
Valid Range	0 through 59
Default Value	None (because nsslapd-auditfaillog-logrotationsync-enabled is off)
Syntax	Integer
Example	nsslapd-auditfaillog-logrotationsyncmin: 30

3.1.1.49. nsslapd-auditfaillog-logrotationtime (Audit Fail Log Rotation Time)

This attribute sets the time between audit fail log file rotations. This attribute supplies only the number of units. The units (day, week, month, and so forth) are given by the **nsslapd-auditfaillog-logrotationtimeunit** attribute. If the **nsslapd-auditfaillog-maxlogsperdir** attribute is set to **1**, the server ignores this attribute.

Directory Server rotates the log at the first write operation after the configured interval has expired, regardless of the size of the log.

Although it is not recommended for performance reasons to specify no log rotation, as the log grows indefinitely, there are two ways of specifying this. Either set the **nsslapd-auditfaillog-maxlogsperdir** attribute value to **1** or set the **nsslapd-auditfaillog-logrotationtime** attribute to **-1**. The server checks

the **nsslapd-auditfaillog-maxlogsperdir** attribute first, and, if this attribute value is larger than **1**, the server then checks the **nsslapd-auditfaillog-logrotationtime** attribute. See [Section 3.1.1.52, “nsslapd-auditfaillog-maxlogsperdir \(Audit Fail Log Maximum Number of Log Files\)”](#) for more information.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647), where a value of -1 means the time between audit fail log file rotation is unlimited.
Default Value	1
Syntax	Integer
Example	nsslapd-auditfaillog-logrotationtime: 100

3.1.1.50. nsslapd-auditfaillog-logrotationtimeunit (Audit Fail Log Rotation Time Unit)

This attribute sets the units for the **nsslapd-auditfaillog-logrotationtime** attribute.

Parameter	Description
Entry DN	cn=config
Valid Values	month week day hour minute
Default Value	week
Syntax	DirectoryString
Example	nsslapd-auditfaillog-logrotationtimeunit: day

3.1.1.51. nsslapd-auditfaillog-maxlogsize (Audit Fail Log Maximum Log Size)

This attribute sets the maximum audit fail log size in megabytes. When this value is reached, the audit fail log is rotated. That means the server starts writing log information to a new log file. If the **nsslapd-auditfaillog-maxlogsperdir** parameter is set to **1**, the server ignores this attribute.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647), where a value of -1 means the log file is unlimited in size.

Parameter	Description
Default Value	100
Syntax	Integer
Example	nsslapd-auditfaillog-maxlogsize: 50

3.1.1.52. nsslapd-auditfaillog-maxlogsize (Audit Fail Log Maximum Number of Log Files)

This attribute sets the total number of audit fail logs that can be contained in the directory where the audit log is stored. Each time the audit fail log is rotated, a new log file is created. When the number of files contained in the audit log directory exceeds the value stored on this attribute, then the oldest version of the log file is deleted. The default is **1** log. If this default is accepted, the server will not rotate the log, and it grows indefinitely.

If the value for this attribute is higher than **1**, then check the **nsslapd-auditfaillog-logrotationtime** attribute to establish whether log rotation is specified. If the **nsslapd-auditfaillog-logrotationtime** attribute has a value of **-1**, then there is no log rotation. See [Section 3.1.1.49, “nsslapd-auditfaillog-logrotationtime \(Audit Fail Log Rotation Time\)”](#) for more information.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to the maximum 32 bit integer value (2147483647)
Default Value	1
Syntax	Integer
Example	nsslapd-auditfaillog-maxlogsize: 10

3.1.1.53. nsslapd-auditfaillog-mode (Audit Fail Log File Permission)

This attribute sets the access mode or file permissions with which audit fail log files are to be created. The valid values are any combination of **000** to **777** since they mirror numbered or absolute UNIX file permissions. The value must be a combination of a 3-digit number, the digits varying from **0** through **7**:

- 0 - None
- 1 - Execute only
- 2 - Write only
- 3 - Write and execute
- 4 - Read only
- 5 - Read and execute

- 6 - Read and write
- 7 - Read, write, and execute

In the 3-digit number, the first digit represents the owner's permissions, the second digit represents the group's permissions, and the third digit represents everyone's permissions. When changing the default value, remember that **000** does not allow access to the logs and that allowing write permissions to everyone can result in the logs being overwritten or deleted by anyone.

The newly configured access mode only affects new logs that are created; the mode is set when the log rotates to a new file.

Parameter	Description
Entry DN	cn=config
Valid Range	000 through 777
Default Value	600
Syntax	Integer
Example	nsslapd-auditfaillog-mode: 600

3.1.1.54. nsslapd-bakdir (Default Backup Directory)

This parameter sets the path to the default backup directory. The Directory Server user must have write permissions in the configured directory.

This setting does not require a server restart to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	Any local directory path.
Default Value	/var/lib/dirsrv/slapd- <i>instance</i> /bak
Syntax	DirectoryString
Example	nsslapd-bakdir: /var/lib/dirsrv/slapd- <i>instance</i> /bak

3.1.1.55. nsslapd-certdir (Certificate and Key Database Directory)

This parameter defines the full path to the directory that Directory Server uses to store the Network Security Services (NSS) database of the instance. This database contains the private keys and certificates of the instance.

As a fallback, Directory Server extracts the private key and certificates to this directory, if the server cannot extract them to the `/tmp/` directory in a private name space. For details about private name spaces, see the **PrivateTmp** parameter description in the `systemd.exec(5)` man page.

The directory specified in **nsslapd-certdir** must be owned by the user ID of the server, and only this user ID must have read-write permissions in this directory. For security reasons, no other users should have permissions to read or write to this directory.

The service must be restarted for changes to this attribute to take effect.

Parameter	Description
Entry DN	<code>cn=config</code>
Valid Values	An absolute path
Default Value	<code>/etc/dirsrv/slapd-<i>instance_name</i>/</code>
Syntax	<code>DirectoryString</code>
Example	<code>nsslapd-certdir: /etc/dirsrv/slapd-<i>instance_name</i>/</code>

3.1.1.56. nsslapd-certmap-basedn (Certificate Map Search Base)

This attribute can be used when client authentication is performed using TLS certificates in order to avoid limitations of the security subsystem certificate mapping, configured in the `/etc/dirsrv/slapd-instance_name/certmap.conf` file. Depending on the configuration in this file, the certificate mapping may be done using a directory subtree search based at the root DN. If the search is based at the root DN, then the **nsslapd-certmap-basedn** attribute may force the search to be based at some entry other than the root. The valid value for this attribute is the DN of the suffix or subtree to use for certificate mapping.

Parameter	Description
Entry DN	<code>cn=config</code>
Valid Values	Any valid DN
Default Value	
Syntax	<code>DirectoryString</code>
Example	<code>nsslapd-certmap-basedn: ou=People,dc=example,dc=com</code>

3.1.1.57. nsslapd-config

This read-only attribute is the config DN.

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid configuration DN
Default Value	
Syntax	DirectoryString
Example	nsslapd-config: cn=config

3.1.1.58. nsslapd-cn-uses-dn-syntax-in-dns

This parameter allows you to enable a DN inside a CN value.

The Directory Server DN normalizer follows [RFC4514](#) and keeps a white space if the RDN attribute type is not based on the DN syntax. However the Directory Server's configuration entry sometimes uses a **cn** attribute to store a DN value. For example in **dn: cn="dc=A,dc=com"**, **cn=mapping tree,cn=config**, the **cn** should be normalized following the DN syntax.

If this configuration is required, enable the **nsslapd-cn-uses-dn-syntax-in-dns** parameter.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-cn-uses-dn-syntax-in-dns: off

3.1.1.59. nsslapd-connection-buffer

This attribute sets the connection buffering behavior. Possible values:

- **0**: Disable buffering. Only single Protocol Data Units (PDU) are read at a time.
- **1**: Regular fixed size **LDAP_SOCKET_IO_BUFFER_SIZE** of **512** bytes.
- **2**: Adaptable buffer size.

The value **2** provides a better performance if the client sends a large amount of data at once. This is, for example, the case for large add and modify operations, or when many asynchronous requests are received over a single connections like during a replication.

Parameter	Description
Entry DN	cn=config
Valid Values	0 1 2
Default Value	1
Syntax	Integer
Example	nsslapd-connection-buffer: 1

3.1.1.60. nsslapd-connection-nocanon

This option allows you to enable or disable the SASL **NOCANON** flag. Disabling avoids the Directory Server looking up DNS reverse entries for outgoing connections.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-connection-nocanon: on

3.1.1.61. nsslapd-conntablesize

This attribute sets the connection table size, which determines the total number of connections supported by the server.

Increase the value of this attribute if Directory Server is refusing connections because it is out of connection slots. When this occurs, the Directory Server’s error log file records the message **Not listening for new connections — too many fds open**.

It may be necessary to increase the operating system limits for the number of open files and number of open files per process, and it may be necessary to increase the **ulimit** for the number of open files (**ulimit -n**) in the shell that starts Directory Server.

The size of the connection table is cap with **nsslapd-maxdescriptor**. See [Section 3.1.1.17, “nsslapd-maxdescriptors \(Maximum File Descriptors\)”](#) for more information.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=config
Valid Values	Operating-system dependent
Default Value	The maximum number of files that the Directory Server process can open. See the gettablesize() glibc function.
Syntax	Integer
Example	nsslapd-conntablesize: 4093

3.1.1.62. nsslapd-counters

The **nsslapd-counters** attribute enables and disables Directory Server database and server performance counters.

There can be a performance impact by keeping track of the larger counters. Turning off 64-bit integers for counters can have a minimal improvement on performance, although it negatively affects long term statistics tracking.

This parameter is enabled by default. To disable counters, stop the Directory Server, edit the **dse.ldif** file directly, and restart the server.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-counters: on

3.1.1.63. nsslapd-csnlogging

This attribute sets whether change sequence numbers (CSNs), when available, are to be logged in the access log. By default, CSN logging is turned on.

Parameter	Description
Entry DN	cn=config

Parameter	Description
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-csnlogging: on

3.1.1.64. nsslapd-defaultnamingcontext

This attribute gives the naming context, of all configured naming contexts, which clients should use by default as a search base. This value is copied over to the root DSE as the **defaultNamingContext** attribute, which allows clients to query the root DSE to obtain the context and then to initiate a search with the appropriate base.

Parameter	Description
Entry DN	cn=config
Valid Values	Any root suffix DN
Default Value	The default user suffix
Syntax	DN
Example	nsslapd-defaultnamingcontext: dc=example,dc=com

3.1.1.65. nsslapd-disk-monitoring

This attribute enables a thread which runs every ten (10) seconds to check the available disk space on the disk or mount where the Directory Server database is running. If the available disk space drops below a configured threshold, then the server begins reducing logging levels, disabling access or audit logs, and deleting rotated logs. If that does not free enough available space, then the server shuts down gracefully (after a warning and grace period).

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString

Parameter	Description
Example	nsslapd-disk-monitoring: on

3.1.1.66. nsslapd-disk-monitoring-grace-period

Sets a grace period to wait before shutting down the server after it hits half of the disk space limit set in [Section 3.1.1.69, “nsslapd-disk-monitoring-threshold”](#). This gives the administrator time to clean out the disk and prevent a shutdown.

Parameter	Description
Entry DN	cn=config
Valid Values	Any integer (sets value in minutes)
Default Value	60
Syntax	Integer
Example	nsslapd-disk-monitoring-grace-period: 45

3.1.1.67. nsslapd-disk-monitoring-logging-critical

Sets whether to shut down the server if the log directories pass the halfway point set in the disk space limit, [Section 3.1.1.69, “nsslapd-disk-monitoring-threshold”](#).

If this is enabled, then logging is *not* disabled and rotated logs are *not* deleted as means of reducing disk usage by the server. The server simply goes toward a shutdown process.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-disk-monitoring-logging-critical: on

3.1.1.68. nsslapd-disk-monitoring-readonly-on-threshold

If the free disk space reaches half of the value you set in the **nsslapd-disk-monitoring-threshold** parameter, Directory Server shuts down the instance after the grace period set in **nsslapd-disk-**

monitoring-grace-period is reached. However, if the disk runs out of space before the instance is down, data can be corrupted. To prevent this problem, enable the **nsslapd-disk-monitoring-readonly-on-threshold** parameter, the Directory Server sets the instance to read-only mode when the threshold is reached.



IMPORTANT

With this setting, Directory Server does not start if the free disk space is below half of the threshold configured in the **nsslapd-disk-monitoring-threshold**.

The service must be restarted for changes to this attribute to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	<code>nsslapd-disk-monitoring-readonly-on-threshold: off</code>

3.1.1.69. nsslapd-disk-monitoring-threshold

Sets the threshold, in bytes, to use to evaluate whether the server has enough available disk space. Once the space reaches half of this threshold, then the server begins a shut down process.

For example, if the threshold is 2MB (the default), then once the available disk space reaches 1MB, the server will begin to shut down.

By default, the threshold is evaluated backs on the disk space used by the configuration, transaction, and database directories for the Directory Server instance. If the [Section 3.1.1.67, “nsslapd-disk-monitoring-logging-critical”](#) attribute is enabled, then the log directory is included in the evaluation.

Parameter	Description
Entry DN	cn=config
Valid Values	* 0 to the maximum 32-bit integer value (2147483647) on 32-bit systems * 0 to the maximum 64-bit integer value (9223372036854775807) on 64-bit systems
Default Value	2000000 (2MB)
Syntax	DirectoryString

Parameter	Description
Example	nsslapd-disk-monitoring-threshold: 2000000

3.1.1.70. nsslapd-dn-validate-strict

The [Section 3.1.1.66, “nsslapd-syntaxcheck”](#) attribute enables the server to verify that any new or modified attribute value matches the required syntax for that attribute.

However, the syntax rules for DNs have grown increasingly strict. Attempting to enforce DN syntax rules in [RFC 4514](#) could break many servers using older syntax definitions. By default, then **nsslapd-syntaxcheck** validates DNs using [RFC 1779](#) or [RFC 2253](#).

The **nsslapd-dn-validate-strict** attribute explicitly enables strict syntax validation for DNs, according to section 3 in [RFC 4514](#). If this attribute is set to **off** (the default), the server normalizes the value before checking it for syntax violations.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-dn-validate-strict: off

3.1.1.71. nsslapd-ds4-compatible-schema

Makes the schema in **cn=schema** compatible with 4.x versions of Directory Server.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-ds4-compatible-schema: off

3.1.1.72. nsslapd-enable-turbo-mode

The Directory Server turbo mode is a feature that enables a worker thread to be dedicated to a connection and continuously read incoming operations from that connection. This can improve the performance on very active connections, and the feature is enabled by default.

Worker threads are processing the LDAP operation received by the server. The number of worker threads is defined in the **nsslapd-threadnumber** parameter. Every five seconds, each worker thread evaluates if the activity level of its current connection is one of the highest among all established connections. Directory Server measures the activity as the number of operations initiated since the last check, and switches a worker thread in turbo mode if the activity of the current connection is one of the highest.

If you encounter long execution times (**etime** value in log files) for bind operations, such as one second or longer, deactivating the turbo mode can improve the performance. However, in some cases, long bind times are a symptom of networking or hardware issues. In these situations, disabling the turbo mode does not result in improved performance.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-enable-turbo-mode: on

3.1.1.73. nsslapd-enable-upgrade-hash

During a simple bind, Directory Server has access to the plain text password due to the nature of bind operations. If the **nsslapd-enable-upgrade-hash** parameter is enabled and a user authenticates, Directory Server checks if the **userPassword** attribute of the user uses the hashing algorithm set in the **passwordStorageScheme** attribute. If the algorithm is different, the server hashes the plain text password with the algorithm from **passwordStorageScheme** and updates the value of the user's **userPassword** attribute.

For example, if you import a user entry with a password that is hashed using a weak algorithm, the server automatically re-hashes the passwords on the first login of the user using the algorithm set in **passwordStorageScheme**, which is, by default, **PBKDF2_SHA256**.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString

Parameter	Description
Example	nsslapd-enable-upgrade-hash: on

3.1.1.74. nsslapd-enquote-sup-oc (Enable Superior Object Class Enquoting)

This attribute is deprecated and will be removed in a future version of Directory Server.

This attribute controls whether quoting in the **objectclass** attributes contained in the **cn=schema** entry conforms to the quoting specified by Internet draft RFC 2252. By default, the Directory Server conforms to RFC 2252, which indicates that this value should not be quoted. Only very old clients need this value set to **on**, so leave it **off**.

Turning this attribute on or off does not affect Directory Server Console.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-enquote-sup-oc: off

3.1.1.75. nsslapd-entryusn-global

The **nsslapd-entryusn-global** parameter defines if the **USN** plug-in assigns unique update sequence numbers (USN) across all back end databases or to each database individually. For unique USNs across all back end databases, set this parameter to **on**.

For further details, see [Section 6.8, “entryusn”](#).

You do not have to restart the server for this setting to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-entryusn-global: off

3.1.1.76. nsslapd-entryusn-import-initval

Entry update sequence numbers (USNs) are not preserved when entries are exported from one server and imported into another, including when initializing a database for replication. By default, the entry USNs for imported entries are set to zero.

It is possible to configure a different initial value for entry USNs using **nsslapd-entryusn-import-initval**. This sets a starting USN which is used for all imported entries.

There are two possible values for **nsslapd-entryusn-import-initval**:

- An integer, which is the explicit start number used for every imported entry.
- *next*, which means that every imported entry uses whatever the highest entry USN value was on the server before the import operation, incremented by one.

Parameter	Description
Entry DN	cn=config
Valid Values	Any integer next
Default Value	
Syntax	DirectoryString
Example	nsslapd-entryusn-import-initval: next

3.1.1.77. nsslapd-errorlog (Error Log)

This attribute sets the path and filename of the log used to record error messages generated by the Directory Server. These messages can describe error conditions, but more often they contain informative conditions, such as:

- Server startup and shutdown times.
- The port number that the server uses.

This log contains differing amounts of information depending on the current setting of the Log Level attribute. See [Section 3.1.1.78, “nsslapd-errorlog-level \(Error Log Level\)”](#) for more information.

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid filename
Default Value	/var/log/dirsrv/slapd- <i>instance</i> /errors
Syntax	DirectoryString

Parameter	Description
Example	nsslapd-errorlog: /var/log/dirsrv/slapd- <i>instance</i> /errors

For error logging to be enabled, this attribute must have a valid path and filename, and the **nsslapd-errorlog-logging-enabled** configuration attribute must be switched to **on**. The table lists the four possible combinations of values for these two configuration attributes and their outcome in terms of disabling or enabling of error logging.

Table 3.5. Possible Combinations for nsslapd-errorlog Configuration Attributes

Attributes in dse.ldif	Value	Logging enabled or disabled
nsslapd-errorlog-logging-enabled	on	Disabled
nsslapd-errorlog	empty string	
nsslapd-errorlog-logging-enabled	on	Enabled
nsslapd-errorlog	<i>filename</i>	
nsslapd-errorlog-logging-enabled	off	Disabled
nsslapd-errorlog	empty string	
nsslapd-errorlog-logging-enabled	off	Disabled
nsslapd-errorlog	<i>filename</i>	

3.1.1.78. nsslapd-errorlog-level (Error Log Level)

This attribute sets the level of logging for the Directory Server. The log level is additive; that is, specifying a value of **3** includes both levels **1** and **2**.

The default value for **nsslapd-errorlog-level** is **16384**.

You do not have to restart the server for this setting to take effect.

Parameter	Description
Entry DN	cn=config

Parameter	Description
Valid Values	<ul style="list-style-type: none"> * 1 – Trace function calls. Logs a message when the server enters and exits a function. * 2 – Debug packet handling. * 4 – Heavy trace output debugging. * 8 – Connection management. * 16 – Print out packets sent/received. * 32 – Search filter processing. * 64 – Config file processing. * 128 – Access control list processing. * 1024 – Log communications with shell databases. * 2048 – Log entry parsing debugging. * 4096 – Housekeeping thread debugging. * 8192 – Replication debugging. * 16384 – Default level of logging used for critical errors and other messages that are always written to the error log; for example, server startup messages. Messages at this level are always included in the error log, regardless of the log level setting. * 32768 – Database cache debugging. * 65536 – Server plug-in debugging. It writes an entry to the log file when a server plug-in calls slapi-log-error. * 262144 – Access control summary information, much less verbose than level 128. This value is recommended for use when a summary of access control processing is needed. Use 128 for very detailed processing messages. * 524288 – LMDB database debugging.
Default Value	16384
Syntax	Integer
Example	nsslapd-errorlog-level: 8192

3.1.1.79. nsslapd-errorlog-list

This read-only attribute provides a list of error log files.

Parameter	Description
Entry DN	cn=config
Valid Values	
Default Value	None
Syntax	DirectoryString
Example	nsslapd-errorlog-list: errorlog2,errorlog3

3.1.1.80. nsslapd-errorlog-logexpirationtime (Error Log Expiration Time)

This attribute sets the maximum age that a log file is allowed to reach before it is deleted. This attribute supplies only the number of units. The units (day, week, month, and so forth) are given by the **nsslapd-errorlog-logexpirationtimeunit** attribute.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 to the maximum 32 bit integer value (2147483647) A value of -1 or 0 means that the log never expires.
Default Value	-1
Syntax	Integer
Example	nsslapd-errorlog-logexpirationtime:1

3.1.1.81. nsslapd-errorlog-logexpirationtimeunit (Error Log Expiration Time Unit)

This attribute sets the units for the **nsslapd-errorlog-logexpirationtime** attribute. If the unit is unknown by the server, then the log never expires.

Parameter	Description
Entry DN	cn=config
Valid Values	month week day
Default Value	month
Syntax	DirectoryString

Parameter	Description
Example	nsslapd-errorlog-logexpirationtimeunit: week

3.1.1.82. nsslapd-errorlog-logging-enabled (Enable Error Logging)

Turns error logging on and off.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-errorlog-logging-enabled: on

3.1.1.83. nsslapd-errorlog-logmaxdiskspace (Error Log Maximum Disk Space)

This attribute sets the maximum amount of disk space in megabytes that the error logs are allowed to consume. If this value is exceeded, the oldest error log is deleted.

When setting a maximum disk space, consider the total number of log files that can be created due to log file rotation. Also, remember that there are three different log files (access log, audit log, and error log) maintained by the Directory Server, each of which consumes disk space. Compare these considerations to the total amount of disk space for the error log.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647), where a value of -1 means that the disk space allowed to the error log is unlimited in size.
Default Value	-1
Syntax	Integer
Example	nsslapd-errorlog-logmaxdiskspace: 10000

3.1.1.84. nsslapd-errorlog-logminfreediskspace (Error Log Minimum Free Disk Space)

This attribute sets the minimum allowed free disk space in megabytes. When the amount of free disk space falls below the value specified on this attribute, the oldest error log is deleted until enough disk space is freed to satisfy this attribute.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 (unlimited) 1 to the maximum 32 bit integer value (2147483647)
Default Value	-1
Syntax	Integer
Example	nsslapd-errorlog-logminfreediskspace: -1

3.1.1.85. nsslapd-errorlog-logrotationsync-enabled (Error Log Rotation Sync Enabled)

This attribute sets whether error log rotation is to be synchronized with a particular time of the day. Synchronizing log rotation this way can generate log files at a specified time during a day, such as midnight to midnight every day. This makes analysis of the log files much easier because they then map directly to the calendar.

For error log rotation to be synchronized with time-of-day, this attribute must be enabled with the **nsslapd-errorlog-logrotationsynchour** and **nsslapd-errorlog-logrotationsyncmin** attribute values set to the hour and minute of the day for rotating log files.

For example, to rotate error log files every day at midnight, enable this attribute by setting its value to **on**, and then set the values of the **nsslapd-errorlog-logrotationsynchour** and **nsslapd-errorlog-logrotationsyncmin** attributes to **0**.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-errorlog-logrotationsync-enabled: on

3.1.1.86. nsslapd-errorlog-logrotationsynchour (Error Log Rotation Sync Hour)

This attribute sets the hour of the day for rotating error logs. This attribute must be used in conjunction with **nsslapd-errorlog-logrotationsync-enabled** and **nsslapd-errorlog-logrotationsyncmin** attributes.

Parameter	Description
Entry DN	cn=config
Valid Range	0 through 23
Default Value	0
Syntax	Integer
Example	nsslapd-errorlog-logrotationsynchour: 23

3.1.1.87. nsslapd-errorlog-logrotationsyncmin (Error Log Rotation Sync Minute)

This attribute sets the minute of the day for rotating error logs. This attribute must be used in conjunction with **nsslapd-errorlog-logrotationsync-enabled** and **nsslapd-errorlog-logrotationsynchour** attributes.

Parameter	Description
Entry DN	cn=config
Valid Range	0 through 59
Default Value	0
Syntax	Integer
Example	nsslapd-errorlog-logrotationsyncmin: 30

3.1.1.88. nsslapd-errorlog-logrotationtime (Error Log Rotation Time)

This attribute sets the time between error log file rotations. This attribute supplies only the number of units. The units (day, week, month, and so forth) are given by the **nsslapd-errorlog-logrotationtimeunit** (Error Log Rotation Time Unit) attribute.

Directory Server rotates the log at the first write operation after the configured interval has expired, regardless of the size of the log.

Although it is not recommended for performance reasons to specify no log rotation, as the log grows indefinitely, there are two ways of specifying this. Either set the **nsslapd-errorlog-maxlogsperdir** attribute value to **1** or set the **nsslapd-errorlog-logrotationtime** attribute to **-1**. The server checks the **nsslapd-errorlog-maxlogsperdir** attribute first, and, if this attribute value is larger than **1**, the server then checks the **nsslapd-errorlog-logrotationtime** attribute. See [Section 3.1.1.91, “nsslapd-errorlog-maxlogsperdir \(Maximum Number of Error Log Files\)”](#) for more information.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647), where a value of -1 means that the time between error log file rotation is unlimited).
Default Value	1
Syntax	Integer
Example	nsslapd-errorlog-logrotationtime: 100

3.1.1.89. nsslapd-errorlog-logrotationtime (Error Log Rotation Time)

This attribute sets the units for **nsslapd-errorlog-logrotationtime** (Error Log Rotation Time). If the unit is unknown by the server, then the log never expires.

Parameter	Description
Entry DN	cn=config
Valid Values	month week day hour minute
Default Value	week
Syntax	DirectoryString
Example	nsslapd-errorlog-logrotationtimeunit: day

3.1.1.90. nsslapd-errorlog-maxlogsize (Maximum Error Log Size)

This attribute sets the maximum error log size in megabytes. When this value is reached, the error log is rotated, and the server starts writing log information to a new log file. If **nsslapd-errorlog-maxlogsperdir** is set to **1**, the server ignores this attribute.

When setting a maximum log size, consider the total number of log files that can be created due to log file rotation. Also, remember that there are three different log files (access log, audit log, and error log) maintained by the Directory Server, each of which consumes disk space. Compare these considerations to the total amount of disk space for the error log.

Parameter	Description
Entry DN	cn=config

Parameter	Description
Valid Range	-1 1 to the maximum 32 bit integer value (2147483647) where a value of -1 means the log file is unlimited in size.
Default Value	100
Syntax	Integer
Example	nsslapd-errorlog-maxlogsize: 100

3.1.1.91. nsslapd-errorlog-maxlogsperdir (Maximum Number of Error Log Files)

This attribute sets the total number of error logs that can be contained in the directory where the error log is stored. Each time the error log is rotated, a new log file is created. When the number of files contained in the error log directory exceeds the value stored on this attribute, then the oldest version of the log file is deleted. The default is **1** log. If this default is accepted, the server does not rotate the log, and it grows indefinitely.

If the value for this attribute is higher than **1**, then check the **nsslapd-errorlog-logrotationtime** attribute to establish whether log rotation is specified. If the **nsslapd-errorlog-logrotationtime** attribute has a value of **-1**, then there is no log rotation. See [Section 3.1.1.88, “nsslapd-errorlog-logrotationtime \(Error Log Rotation Time\)”](#) for more information.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to the maximum 32 bit integer value (2147483647)
Default Value	1
Syntax	Integer
Example	nsslapd-errorlog-maxlogsperdir: 10

3.1.1.92. nsslapd-errorlog-mode (Error Log File Permission)

This attribute sets the access mode or file permissions with which error log files are to be created. The valid values are any combination of **000** to **777** since they mirror numbered or absolute UNIX file permissions. That is, the value must be a combination of a 3-digit number, the digits varying from **0** through **7**:

- 0 - None
- 1 - Execute only

- 2 - Write only
- 3 - Write and execute
- 4 - Read only
- 5 - Read and execute
- 6 - Read and write
- 7 - Read, write, and execute

In the 3-digit number, the first digit represents the owner's permissions, the second digit represents the group's permissions, and the third digit represents everyone's permissions. When changing the default value, remember that **000** does not allow access to the logs and that allowing write permissions to everyone can result in the logs being overwritten or deleted by anyone.

The newly configured access mode only affects new logs that are created; the mode is set when the log rotates to a new file.

Parameter	Description
Entry DN	cn=config
Valid Range	000 through 777
Default Value	600
Syntax	Integer
Example	nsslapd-errorlog-mode: 600

3.1.1.93. nsslapd-force-sasl-external

When establishing a TLS connection, a client sends its certificate first and then issues a BIND request using the SASL/EXTERNAL mechanism. Using SASL/EXTERNAL tells the Directory Server to use the credentials in the certificate for the TLS handshake. However, some clients do not use SASL/EXTERNAL when they send their BIND request, so the Directory Server processes the bind as a simple authentication request or an anonymous request and the TLS connection fails.

The **nsslapd-force-sasl-external** attribute forces clients in certificate-based authentication to send the BIND request using the SASL/EXTERNAL method.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off

Parameter	Description
Syntax	String
Example	nsslapd-force-sasl-external: on

3.1.1.94. nsslapd-groupevalnestlevel

This attribute is deprecated, and documented here only for historical purposes.

The Access Control Plug-in does not use the value specified by the **nsslapd-groupevalnestlevel** attribute to set the number of levels of nesting that access control performs for group evaluation. Instead, the number of levels of nesting is hardcoded as **5**.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to 5
Default Value	5
Syntax	Integer
Example	nsslapd-groupevalnestlevel: 5

3.1.1.95. nsslapd-idletimeout (Default Idle Timeout)

This attribute sets the amount of time in seconds after which an idle LDAP client connection is closed by the server. A value of **0** means that the server never closes idle connections. This setting applies to all connections and all users. Idle timeout is enforced when the connection table is walked, when **poll()** does not return zero. Therefore, a server with a single connection never enforces the idle timeout.

Use the **nsIdleTimeout** operational attribute, which can be added to user entries, to override the value assigned to this attribute. For details, see the "Setting Resource Limits Based on the Bind DN" section in the *Red Hat Directory Server Administration Guide*.



NOTE

For very large databases, with millions of entries, this attribute must have a high enough value that the online initialization process can complete or replication will fail when the connection to the server times out. Alternatively, the **nsIdleTimeout** attribute can be set to a high value on the entry used as the supplier bind DN.

Parameter	Description
Entry DN	cn=config

Parameter	Description
Valid Range	0 to the maximum 32 bit integer value (2147483647)
Default Value	3600
Syntax	Integer
Example	nsslapd-idletimeout: 3600

3.1.1.96. nsslapd-ignore-virtual-attrs

This parameter allows to disable the virtual attribute lookup in a search entry.

If you do not require virtual attributes, you can disable virtual attribute lookups in search results to increase the speed of searches.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-ignore-virtual-attrs: off

3.1.1.97. nsslapd-instancedir (Instance Directory)

This attribute is deprecated. There are now separate configuration parameters for instance-specific paths, such as **nsslapd-certdir** and **nsslapd-lockdir**. See the documentation for the specific directory path that is set.

3.1.1.98. nsslapd-ioblocktimeout (IO Block Time Out)

This attribute sets the amount of time in milliseconds after which the connection to a stalled LDAP client is closed. An LDAP client is considered to be stalled when it has not made any I/O progress for read or write operations.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to the maximum 32 bit integer value (2147483647) in ticks

Parameter	Description
Default Value	10000
Syntax	Integer
Example	nsslapd-ioblocktimeout: 10000

3.1.1.99. nsslapd-lastmod (Track Modification Time)

This attribute sets whether the Directory Server maintains the **creatorsName**, **createTimestamp**, **modifiersName**, and **modifyTimestamp** operational attributes for newly created or updated entries.



IMPORTANT

Red Hat recommends not disabling tracking these attributes. If disabled, entries do not get a unique ID assigned in the **nsUniqueId** attribute and replication does not work.

You do not have to restart the server for this setting to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-lastmod: on

3.1.1.100. nsslapd-ldapiautobind (Enable Autobind)

The **nsslapd-ldapiautobind** sets whether the server will allow users to autobind to Directory Server using LDAPI. Autobind maps the UID or GUID number of a system user to a Directory Server user, and automatically authenticates the user to Directory Server based on those credentials. The Directory Server connection occurs over UNIX socket.

Along with enabling autobind, configuring autobind requires configuring mapping entries. The **nsslapd-ldapimaprootdn** maps a root user on the system to the Directory Manager. The **nsslapd-ldapimaprotoentries** maps regular users to Directory Server users, based on the parameters defined in the **nsslapd-ldapuidnumbertype**, **nsslapd-ldapgidnumbertype**, and **nsslapd-ldapentrysearchbase** attributes.

Autobind can only be enabled if LDAPI is enabled, meaning the **nsslapd-ldapilisten** is **on** and the **nsslapd-ldapfilepath** attribute is set to an LDAPI socket.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-ldapiautobind: off

3.1.1.101. nsslapd-ldapientrysearchbase (Search Base for LDAPI Authentication Entries)

With autobind, it is possible to map system users to Directory Server user entries, based on the system user's UID and GUID numbers. This requires setting Directory Server parameters for which attribute to use for the UID number (**nsslapd-ldapuidnumbertype**) and GUID number (**nsslapd-ldapgidnumbertype**) and setting the search base to use to search for matching user entries.

The **nsslapd-ldapientrysearchbase** gives the subtree to search for user entries to use for autobind.

Parameter	Description
Entry DN	cn=config
Valid Values	DN
Default Value	The suffix created when the server instance was created, such as dc=example,dc=com
Syntax	DN
Example	nsslapd-ldapientrysearchbase: ou=people,dc=example,dc=com

3.1.1.102. nsslapd-ldapfilepath (File Location for LDAPI Socket)

LDAPI connects a user to an LDAP server over a UNIX socket rather than TCP. In order to configure LDAPI, the server must be configured to communicate over a UNIX socket. The UNIX socket to use is set in the **nsslapd-ldapfilepath** attribute.

Parameter	Description
Entry DN	cn=config
Valid Values	Any directory path

Parameter	Description
Default Value	/var/run/dirsrv/slapd-example.socket
Syntax	Case-exact string
Example	nsslapd-ldapidnumbertype: /var/run/slapd-example.socket

3.1.1.103. nsslapd-ldapidnumbertype (Attribute Mapping for System GUID Number)

Autobind can be used to authenticate system users to the server automatically and connect to the server using a UNIX socket. To map the system user to a Directory Server user for authentication, the system user's UID and GUID numbers should be mapped to be a Directory Server attribute. The **nsslapd-ldapidnumbertype** attribute points to the Directory Server attribute to map system GUIDs to user entries.

Users can only connect to the server with autobind if LDAPI is enabled (**nsslapd-ldapilisten** and **nsslapd-ldapfilepath**), autobind is enabled (**nsslapd-ldapiabind**), and autobind mapping is enabled for regular users (**nsslapd-ldapimaptocentries**).

Parameter	Description
Entry DN	cn=config
Valid Values	Any Directory Server attribute
Default Value	gidNumber
Syntax	DirectoryString
Example	nsslapd-ldapidnumbertype: gidNumber

3.1.1.104. nsslapd-ldapilisten (Enable LDAPI)

The **nsslapd-ldapilisten** enables LDAPI connections to the Directory Server. LDAPI allows users to connect to the Directory Server over a UNIX socket rather than a standard TCP port. Along with enabling LDAPI by setting **nsslapd-ldapilisten** to **on**, there must also be a UNIX socket set for LDAPI in the **nsslapd-ldapfilepath** attribute.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on

Parameter	Description
Syntax	DirectoryString
Example	nsslapd-ldaplisten: on

3.1.1.105. nsslapd-ldapimaprootdn (Autobind Mapping for Root User)

With autobind, a system user is mapped to a Directory Server user and then automatically authenticated to the Directory Server over a UNIX socket.

The root system user (the user with a UID of 0) is mapped to whatever Directory Server entry is specified in the **nsslapd-ldapimaprootdn** attribute.

Parameter	Description
Entry DN	cn=config
Valid Values	Any DN
Default Value	cn=Directory Manager
Syntax	DN
Example	nsslapd-ldapimaprootdn: cn=Directory Manager

3.1.1.106. nsslapd-ldapimaptoentries (Enable Autobind Mapping for Regular Users)

With autobind, a system user is mapped to a Directory Server user and then automatically authenticated to the Directory Server over a UNIX socket. This mapping is automatic for root users, but it must be enabled for regular system users through the **nsslapd-ldapimaptoentries** attribute. Setting this attribute to **on** enables mapping for regular system users to Directory Server entries. If this attribute is not enabled, then only root users can use autobind to authenticate to the Directory Server, and all other users connect anonymously.

The mappings themselves are configured through the **nsslapd-ldapiuidnumbertype** and **nsslapd-ldapgidnumbertype** attributes, which map Directory Server attributes to the user's UID and GUID numbers.

Users can only connect to the server with autobind if LDAPI is enabled (**nsslapd-ldaplisten** and **nsslapd-ldapfilepath**) and autobind is enabled (**nsslapd-ldapiautobind**).

Parameter	Description
Entry DN	cn=config
Valid Values	on off

Parameter	Description
Default Value	off
Syntax	DirectoryString
Example	nsslapd-ldapimaptointries: on

3.1.1.107. nsslapd-Idapiuidnumbertype

Autobind can be used to authenticate system users to the server automatically and connect to the server using a UNIX socket. To map the system user to a Directory Server user for authentication, the system user's UID and GUID numbers must be mapped to be a Directory Server attribute. The **nsslapd-Idapiuidnumbertype** attribute points to the Directory Server attribute to map system UIDs to user entries.

Users can only connect to the server with autobind if LDAPI is enabled (**nsslapd-Idapilisten** and **nsslapd-ldapfilepath**), autobind is enabled (**nsslapd-Idapiautobind**), and autobind mapping is enabled for regular users (**nsslapd-ldapimaptointries**).

Parameter	Description
Entry DN	cn=config
Valid Values	Any Directory Server attribute
Default Value	uidNumber
Syntax	DirectoryString
Example	nsslapd-Idapiuidnumbertype: uidNumber

3.1.1.108. nsslapd-ldifdir

Directory Server exports files in LDAP Data Interchange Format (LDIF) format to the directory set in this parameter when using the **db2ldif** or **db2ldif.pl**. The directory must be owned by the Directory Server user and group. Only this user and group must have read and write access in this directory.

The service must be restarted for changes to this attribute to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	Any directory writable by the Directory Server user

Parameter	Description
Default Value	/var/lib/dirsrv/slapd- <i>instance_name</i> /ldif/
Syntax	DirectoryString
Example	nsslapd-ldifdir: /var/lib/dirsrv/slapd- <i>instance_name</i> /ldif/

3.1.1.109. nsslapd-listen-backlog-size

This attribute sets the maximum of the socket connection backlog. The listen service sets the number of sockets available to receive incoming connections. The backlog setting sets a maximum length for how long the queue for the socket (sockfd) can grow before refusing connections.

Parameter	Description
Entry DN	cn=config
Valid Values	The maximum 64-bit integer value (9223372036854775807)
Default Value	128
Syntax	Integer
Example	nsslapd-listen-backlog-size: 128

3.1.1.110. nsslapd-listenhost (Listen to IP Address)

This attribute allows multiple Directory Server instances to run on a multihomed machine (or makes it possible to limit listening to one interface of a multihomed machine). There can be multiple IP addresses associated with a single host name, and these IP addresses can be a mix of both IPv4 and IPv6. This parameter can be used to restrict the Directory Server instance to a single IP interface.

If a host name is given as the **nsslapd-listenhost** value, then the Directory Server responds to requests for every interface associated with the host name. If a single IP interface (either IPv4 or IPv6) is given as the **nsslapd-listenhost** value, Directory Server only responds to requests sent to that specific interface. Either an IPv4 or IPv6 address can be used.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=config
Valid Values	Any local host name, IPv4 or IPv6 address

Parameter	Description
Default Value	
Syntax	DirectoryString
Example	nsslapd-listenhost: ldap.example.com

3.1.1.111. nsslapd-localhost (Local Host)

This attribute specifies the host machine on which the Directory Server runs. This attribute creates the referral URL that forms part of the MMR protocol. In a high-availability configuration with failover nodes, that referral should point to the virtual name of the cluster, not the local host name.

Parameter	Description
Entry DN	cn=config
Valid Values	Any fully qualified host name.
Default Value	Hostname of installed machine.
Syntax	DirectoryString
Example	nsslapd-localhost: phonebook.example.com

3.1.1.112. nsslapd-localuser (Local User)

This attribute sets the user as whom the Directory Server runs. The group as which the user runs is derived from this attribute by examining the user's primary group. Should the user change, then all of the instance-specific files and directories for this instance need to be changed to be owned by the new user, using a tool such as **chown**.

The value for the **nsslapd-localuser** is set initially when the server instance is configured.

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid user
Default Value	
Syntax	DirectoryString
Example	nsslapd-localuser: dirsrv

3.1.1.113. nsslapd-lockdir (Server Lock File Directory)

This is the full path to the directory the server uses for lock files. The default value is `/var/lock/dirsrv/slapd-instance`. Changes to this value will not take effect until the server is restarted.

Parameter	Description
Entry DN	<code>cn=config</code>
Valid Values	Absolute path to a directory owned by the server user ID with write access to the server ID
Default Value	<code>/var/lock/dirsrv/slapd-<i>instance</i></code>
Syntax	DirectoryString
Example	<code>nsslapd-lockdir: /var/lock/dirsrv/slapd-<i>instance</i></code>

3.1.1.114. nsslapd-localssf

The **nsslapd-localssf** parameter sets the security strength factor (SSF) for LDAPI connections. Directory Server allows LDAPI connections only if the value set in **nsslapd-localssf** is greater or equal than the value set in the **nsslapd-minssf** parameter. Therefore, LDAPI connections meet the minimum SSF set in **nsslapd-minssf**.

You do not have to restart the server for this setting to take effect.

Parameter	Description
Entry DN	<code>cn=config</code>
Valid Values	0 to the maximum 32-bit integer value (2147483647)
Default Value	71
Syntax	Integer
Example	<code>nsslapd-localssf: 71</code>

3.1.1.115. nsslapd-logging-hr-timestamps-enabled (Enable or Disable High-resolution Log Timestamps)

Controls whether logs will use high resolution timestamps with nanosecond precision, or standard resolution timestamps with one second precision. Enabled by default. Set this option to **off** to revert log timestamps back to one second precision.

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-logging-hr-timestamps-enabled: on

3.1.1.116. nsslapd-maxbersize (Maximum Message Size)

Defines the maximum size in bytes allowed for an incoming message. This limits the size of LDAP requests that can be handled by the Directory Server. Limiting the size of requests prevents some kinds of denial of service attacks.

The limit applies to the total size of the LDAP request. For example, if the request is to add an entry and if the entry in the request is larger than the configured value or the default, then the add request is denied. However, the limit is not applied to replication processes. Be cautious before changing this attribute.

This setting does not require a server restart to take effect.

Parameter	Description
Entry DN	cn=config
Valid Range	0 - 2 gigabytes (2,147,483,647 bytes) Zero 0 means that the default value should be used.
Default Value	2097152
Syntax	Integer
Example	nsslapd-maxbersize: 2097152

3.1.1.117. nsslapd-maxdescriptors (Maximum File Descriptors)

This attribute sets the maximum, platform-dependent number of file descriptors that the Directory Server tries to use. A file descriptor is used whenever a client connects to the server and also for some server activities, such as index maintenance. File descriptors are also used by access logs, error logs, audit logs, database files (indexes and transaction logs), and as sockets for outgoing connections to other servers for replication and chaining.

The number of descriptors available for TCP/IP to serve client connections is determined by **nsslapd-conntablesize**, and is equal to the **nsslapd-maxdescriptors** attribute minus the number of file descriptors used by the server as specified in the **nsslapd-reserveddescriptors** attribute for non-client

connections, such as index management and managing replication. The **nsslapd-reserveddescriptors** attribute is the number of file descriptors available for other uses as described above. See [Section 3.1.1.143, “nsslapd-reserveddescriptors \(Reserved File Descriptors\)”](#).

The number given here should not be greater than the total number of file descriptors that the operating system allows the **ns-slapd** process to use. This number differs depending on the operating system.

If this value is set too high, the Directory Server queries the operating system for the maximum allowable value, and then use that value. It also issues a warning in the error log. If this value is set to an invalid value remotely, by using the Directory Server Console or **ldapmodify**, the server rejects the new value, keep the old value, and respond with an error.

Some operating systems let users configure the number of file descriptors available to a process. See the operating system documentation for details on file descriptor limits and configuration. The **dsktune** program (explained in the *Red Hat Directory Server Installation Guide*) can be used to suggest changes to the system kernel or TCP/IP tuning attributes, including increasing the number of file descriptors if necessary. Increased the value on this attribute if the Directory Server is refusing connections because it is out of file descriptors. When this occurs, the following message is written to the Directory Server’s error log file:

Not listening for new connections -- too many fds open

See [Section 3.1.1.61, “nsslapd-conntablesize”](#) for more information about increasing the number of incoming connections.



NOTE

UNIX shells usually have configurable limits on the number of file descriptors. See the operating system documentation for further information about **limit** and **ulimit**, as these limits can often cause problems.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to 65535
Default Value	4096
Syntax	Integer
Example	nsslapd-maxdescriptors: 4096

3.1.1.118. nsslapd-maxsasliosize (Maximum SASL Packet Size)

When a user is authenticated to the Directory Server over SASL GSS-API, the server must allocate a certain amount of memory to the client to perform LDAP operations, according to how much memory the client requests. It is possible for an attacker to send such a large packet size that it crashes the Directory Server or ties it up indefinitely as part of a denial of service attack.

The packet size which the Directory Server will allow for SASL clients can be limited using the **nsslapd-maxsaslosize** attribute. This attribute sets the maximum allowed SASL IO packet size that the server will accept.

When an incoming SASL IO packet is larger than the **nsslapd-maxsaslosize** limit, the server immediately disconnects the client and logs a message to the error log, so that an administrator can adjust the setting if necessary.

This attribute value is specified in bytes.

Parameter	Description
Entry DN	cn=config
Valid Range	* -1 (unlimited) to the maximum 32-bit integer value (2147483647) on 32-bit systems * -1 (unlimited) to the maximum 64-bit integer value (9223372036854775807) on 64-bit systems
Default Value	2097152 (2MB)
Syntax	Integer
Example	nsslapd-maxsaslosize: 2097152

3.1.119. nsslapd-maxthreadsperconn (Maximum Threads per Connection)

Defines the maximum number of threads that a connection should use. For normal operations where a client binds and only performs one or two operations before unbinding, use the default value. For situations where a client binds and simultaneously issues many requests, increase this value to allow each connection enough resources to perform all the operations. This attribute is not available from the server console.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to maximum threadnumber
Default Value	5
Syntax	Integer
Example	nsslapd-maxthreadsperconn: 5

3.1.120. nsslapd-minssf

A *security strength factor* is a relative measurement of how strong a connection is according to its key strength. The SSF determines how secure an TLS or SASL connection is. The **nsslapd-minssf** attribute

sets a minimum SSF requirement for any connection to the server; any connection attempts that are weaker than the minimum SSF are rejected.

TLS and SASL connections can be mixed in a connection to the Directory Server. These connections generally have different SSFs. The higher of the two SSFs is used to compare to the minimum SSF requirement.

Setting the SSF value to 0 means that there is no minimum setting.

Parameter	Description
Entry DN	cn=config
Valid Values	Any positive integer
Default Value	0 (off)
Syntax	DirectoryString
Example	nsslapd-minssf: 128

3.1.1.121. nsslapd-minssf-exclude-rootdse

A *security strength factor* is a relative measurement of how strong a connection is according to its key strength. The SSF determines how secure an TLS or SASL connection is.

The **nsslapd-minssf-exclude-rootdse** attribute sets a minimum SSF requirement for any connection to the server except for queries for the root DSE. This enforces appropriate SSF values for most connections, while still allowing clients to get required information about the server configuration from the root DSE without having to establish a secure connection first.

Parameter	Description
Entry DN	cn=config
Valid Values	Any positive integer
Default Value	0 (off)
Syntax	DirectoryString
Example	nsslapd-minssf-exclude-rootdse: 128

3.1.1.122. nsslapd-moddn-aci

This parameter controls the ACI checks when directory entries are moved from one subtree to another and using source and target restrictions in moddn operations. For backward compatibility, you can disable the ACI checks.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-moddn-aci: on

3.1.1.123. nsslapd-malloc-mmap-threshold

If a Directory Server instance is started as a service using the **systemctl** utility, environment variables are not passed to the server unless you set them in the **/etc/sysconfig/dirsrv** or **/etc/sysconfig/dirsrv-*instance_name*** file. For further details, see the **systemd.exec(3)** man page.

Instead of manually editing the service files to set the **M_MMAP_THRESHOLD** environment variable, the **nsslapd-malloc-mmap-threshold** parameter enables you to set the value in the Directory Server configuration. For further details, see the **M_MMAP_THRESHOLD** parameter description in the **mallopt(3)** man page.

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	cn=config
Valid Range	0 - 33554432
Default Value	See the M_MMAP_THRESHOLD parameter description in the mallopt(3) man page.
Syntax	Integer
Example	nsslapd-malloc-mmap-threshold: 33554432

3.1.1.124. nsslapd-malloc-mxfast

If a Directory Server instance is started as a service using the **systemctl** utility, environment variables are not passed to the server unless you set them in the **/etc/sysconfig/dirsrv** or **/etc/sysconfig/dirsrv-*instance_name*** file. For further details, see the **systemd.exec(3)** man page.

Instead of manually editing the service files to set the **M_MXFAST** environment variable, the **nsslapd-malloc-mxfast** parameter enables you to set the value in the Directory Server configuration. For further details, see the **M_MXFAST** parameter description in the **mallopt(3)** man page.

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	cn=config
Valid Range	0 - 80 * (sizeof(size_t) / 4)
Default Value	See the M_MXFAST parameter description in the mallopt(3) man page.
Syntax	Integer
Example	nsslapd-malloc-mxfast: 1048560

3.1.1.125. nsslapd-malloc-trim-threshold

If a Directory Server instance is started as a service using the **systemctl** utility, environment variables are not passed to the server unless you set them in the **/etc/sysconfig/dirsrv** or **/etc/sysconfig/dirsrv-instance_name** file. For further details, see the **systemd.exec(3)** man page.

Instead of manually editing the service files to set the **M_TRIM_THRESHOLD** environment variable, the **nsslapd-malloc-trim-threshold** parameter enables you to set the value in the Directory Server configuration. For further details, see the **M_TRIM_THRESHOLD** parameter description in the **mallopt(3)** man page.

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to 2^31-1
Default Value	See the M_TRIM_THRESHOLD parameter description in the mallopt(3) man page.
Syntax	Integer
Example	nsslapd-malloc-trim-threshold: 131072

3.1.1.126. nsslapd-nagle

When the value of this attribute is **off**, the **TCP_NODELAY** option is set so that LDAP responses (such as entries or result messages) are sent back to a client immediately. When the attribute is turned on, default TCP behavior applies; specifically, sending data is delayed so that additional data can be grouped into one packet of the underlying network MTU size, typically 1500 bytes for Ethernet.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-nagle: off

3.1.1.127. nsslapd-ndn-cache-enabled

Normalizing distinguished names (DN) is a resource intensive task. If the **nsslapd-ndn-cache-enabled** parameter is enabled, Directory Server caches normalized DNs in memory. Update the **nsslapd-ndn-cache-max-size** parameter to set the maximum size of this cache.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-ndn-cache-enabled: on

3.1.1.128. nsslapd-ndn-cache-max-size

Normalizing distinguished names (DN) is a resource intensive task. If the **nsslapd-ndn-cache-enabled** parameter is enabled, Directory Server caches normalized DNs in memory. The **nsslapd-ndn-cache-max-size** parameter sets the maximum size of this cache.

If a DN requested is not cached already, it is normalized and added. When the cache size limit is exceeded, Directory Server removes the least recently used 10,000 DNs from the cache. However, a minimum of 10,000 DNs is always kept cached.

Parameter	Description
Entry DN	cn=config
Valid Values	0 to the maximum 32-bit integer value (2147483647)

Parameter	Description
Default Value	20971520
Syntax	Integer
Example	nsslapd-ndn-cache-max-size: 20971520

3.1.1.129. nsslapd-outbound-ldap-io-timeout

This attribute limits the I/O wait time for all outbound LDAP connections. The default is **300000** milliseconds (5 minutes). A value of **0** means that the server does not impose a limit on I/O wait time.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to the maximum 32-bit integer value (2147483647)
Default Value	300000
Syntax	DirectoryString
Example	nsslapd-outbound-ldap-io-timeout: 300000

3.1.1.130. nsslapd-pagedsizelimit (Size Limit for Simple Paged Results Searches)

This attribute sets the maximum number of entries to return from a search operation *specifically which uses the simple paged results control*. This overrides the **nsslapd-sizelimit** attribute for paged searches.

If this value is set to zero, then the **nsslapd-sizelimit** attribute is used for paged searches as well as non-paged searches.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 to the maximum 32 bit integer value (2147483647)
Default Value	
Syntax	Integer
Example	nsslapd-pagedsizelimit: 10000

3.1.1.131. nsslapd-plug-in

This read-only attribute lists the DNs of the plug-in entries for the syntax and matching rule plug-ins loaded by the server.

3.1.1.132. nsslapd-plugin-binddn-tracking

Sets the bind DN used for an operation as the modifier of an entry, even if the operation itself was initiated by a server plug-in. The specific plug-in which performed the operation is listed in a separate operational attribute, **internalModifiersname**.

One change can trigger other, automatic changes in the directory tree. When a user is deleted, for example, that user is automatically removed from any groups it belonged to by the Referential Integrity Plug-in. The initial deletion of the user is performed by whatever user account is bound to the server, but the updates to the groups (by default) are shown as being performed by the plug-in, with no information about which user initiated that update. The **nsslapd-plugin-binddn-tracking** attribute allows the server to track which user originated an update operation, as well as the internal plug-in which actually performed it. For example:

```
dn: cn=my_group,ou=groups,dc=example,dc=com
modifiersname: uid=jsmith,ou=people,dc=example,dc=com
internalModifiersname: cn=referential integrity plugin,cn=plugins,cn=config
```

This attribute is disabled by default.

Parameter	Description
Entry DN	cn=config
Valid Range	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-plugin-binddn-tracking: on

3.1.1.133. nsslapd-plugin-logging

By default, even if access logging is set to record internal operations, plug-in internal operations are not logged in the access log file. Instead of enabling the logging in each plug-in's configuration, you can control it globally with this parameter.

When enabled, plug-ins use this global setting and log access and audit events if enabled.

If **nsslapd-plugin-logging** is enabled and **nsslapd-accesslog-level** is set to record internal operations, unindexed searches and other internal operations are logged into the access log file.

In case **nsslapd-plugin-logging** is not set, unindexed searches from plug-ins are still logged in the Directory Server error log.

Parameter	Description
Entry DN	cn=config
Valid Range	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-plugin-logging: off

3.1.1.134. nsslapd-port (Port Number)

This attribute gives the TCP/IP port number used for standard LDAP communications. To run TLS over this port, use the Start TLS extended operation. This selected port must be unique on the host system; make sure no other application is attempting to use the same port number. Specifying a port number of less than **1024** means the Directory Server has to be started as **root**.

The server sets its **uid** to the **nsslapd-localuser** value after startup. When changing the port number for a configuration directory, the corresponding server instance entry in the configuration directory must be updated.

The server has to be restarted for the port number change to be taken into account.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to 65535
Default Value	389
Syntax	Integer
Example	nsslapd-port: 389



NOTE

Set the port number to zero (**0**) to disable the LDAP port if the LDAPS port is enabled.

3.1.1.135. nsslapd-privatenamespaces

This read-only attribute contains the list of the private naming contexts **cn=config**, **cn=schema**, and **cn=monitor**.

Parameter	Description
Entry DN	cn=config
Valid Values	cn=config, cn=schema, and cn=monitor
Default Value	
Syntax	DirectoryString
Example	nsslapd-prvatenamespaces: cn=config

3.1.1.136. nsslapd-pwpolicy-inherit-global (Inherit Global Password Syntax)

When the fine-grained password syntax is not set, new or updated passwords are not checked even though the global password syntax is configured. To inherit the global fine-grained password syntax, set this attribute to **on**.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-pwpolicy-inherit-global: off

3.1.1.137. nsslapd-pwpolicy-local (Enable Subtree- and User-Level Password Policy)

Turns fine-grained (subtree- and user-level) password policy on and off.

If this attribute has a value of **off**, all entries (except for **cn=Directory Manager**) in the directory are subjected to the global password policy; the server ignores any defined subtree/user level password policy.

If this attribute has a value of **on**, the server checks for password policies at the subtree- and user-level and enforce those policies.

Parameter	Description
Entry DN	cn=config
Valid Values	on off

Parameter	Description
Default Value	off
Syntax	DirectoryString
Example	nsslapd-pwpolicy-local: off

3.1.1.138. nsslapd_READONLY (Read Only)

This attribute sets whether the whole server is in read-only mode, meaning that neither data in the databases nor configuration information can be modified. Any attempt to modify a database in read-only mode returns an error indicating that the server is unwilling to perform the operation.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd_READONLY: off

3.1.1.139. nsslapd_referral (Referral)

This multi-valued attribute specifies the LDAP URLs to be returned by the suffix when the server receives a request for an entry not belonging to the local tree; that is, an entry whose suffix does not match the value specified on any of the suffix attributes. For example, assume the server contains only entries:

```
ou=People,dc=example,dc=com
```

but the request is for this entry:

```
ou=Groups,dc=example,dc=com
```

In this case, the referral would be passed back to the client in an attempt to allow the LDAP client to locate a server that contains the requested entry. Although only one referral is allowed per Directory Server instance, this referral can have multiple values.

**NOTE**

To use TLS communications, the referral attribute should be in the form **ldaps://server-location**.

Start TLS does not support referrals.

For more information on managing referrals, see the "Configuring Directory Databases" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid LDAP URL
Default Value	
Syntax	DirectoryString
Example	nsslapd-referral: ldap://ldap.example.com/dc=example,dc=com

3.1.1.140. nsslapd-referralmode (Referral Mode)

When set, this attribute sends back the referral for any request on any suffix.

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid LDAP URL
Default Value	
Syntax	DirectoryString
Example	nsslapd-referralmode: ldap://ldap.example.com

3.1.1.141. nsslapd-require-secure-binds

This parameter requires that a user authenticate to the directory over a protected connection such as TLS, StartTLS, or SASL, rather than a regular connection.

**NOTE**

This only applies to authenticated binds. Anonymous binds and unauthenticated binds can still be completed over a standard channel, even if **nsslapd-require-secure-binds** is turned on.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-require-secure-binds: on

3.1.1.142. nsslapd-requiresrestart

This parameter lists what other core configuration attributes require that the server be restarted after a modification. This means that if any attribute listed in **nsslapd-requiresrestart** is changed, the new setting does not take effect until after the server is restarted. The list of attributes can be returned in an **ldapsearch**:

```
ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -b "cn=config" -s sub -x "(objectclass=*)" | grep nsslapd-requiresrestart
```

This attribute is multi-valued.

Parameter	Description
Entry DN	cn=config
Valid Values	Any core server configuration attribute
Default Value	
Syntax	DirectoryString
Example	nsslapd-requiresrestart: nsslapd-cachesize

3.1.1.143. nsslapd-reserveddescriptors (Reserved File Descriptors)

This attribute specifies the number of file descriptors that Directory Server reserves for managing non-client connections, such as index management and managing replication. The number of file descriptors that the server reserves for this purpose subtracts from the total number of file descriptors available for servicing LDAP client connections (See [Section 3.1.1.117, “nsslapd-maxdescriptors \(Maximum File Descriptors\)”](#)).

Most installations of Directory Server should never need to change this attribute. However, consider increasing the value on this attribute if all of the following are true:

- The server is replicating to a large number of consumer servers (more than 10), or the server is maintaining a large number of index files (more than 30).

- The server is servicing a large number of LDAP connections.
- There are error messages reporting that the server is unable to open file descriptors (the actual error message differs depending on the operation that the server is attempting to perform), but these error messages are *not* related to managing client LDAP connections.

Increasing the value on this attribute may result in more LDAP clients being unable to access the directory. Therefore, the value on this attribute is increased, also increase the value on the **nsslapd-maxdescriptors** attribute. It may not be possible to increase the **nsslapd-maxdescriptors** value if the server is already using the maximum number of file descriptors that the operating system allows a process to use; see the operating system documentation for details. If this is the case, then reduce the load on the server by causing LDAP clients to search alternative directory replicas. See [Section 3.1.1.61, "nsslapd-conntablesize"](#) for information about file descriptor usage for incoming connections.

To assist in computing the number of file descriptors set for this attribute, use the following formula:

$$\text{nsslapd-reserveddescriptor} = 20 + (\text{NlrbmBackends} * 4) + \text{NglobalIndex} + \text{ReplicationDescriptor} + \text{ChainingBackendDescriptors} + \text{PTADescriptors} + \text{SSLDescriptors}$$

- *NlrbmBackends* is the number of ldbm databases.
- *NglobalIndex* is the total number of configured indexes for all databases including system indexes. (By default 8 system indexes and 17 additional indexes per database).
- *ReplicationDescriptor* is eight (8) plus the number of replicas in the server that can act as a supplier or hub (*NSupplierReplica*).
- *ChainingBackendDescriptors* is *NchainingBackend* times the *nsOperationConnectionsLimit* (a chaining or database link configuration attribute; **10** by default).
- *PTADescriptors* is **3** if PTA is configured and **0** if PTA is not configured.
- *SSLDescriptors* is **5** (4 files + 1 listensocket) if TLS is configured and **0** if TLS is not configured.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to 65535
Default Value	64
Syntax	Integer
Example	nsslapd-reserveddescriptors: 64

3.1.1.144. nsslapd-return-exact-case (Return Exact Case)

Returns the exact case of attribute type names as requested by the client. Although LDAPv3-compliant clients must ignore the case of attribute names, some client applications require attribute names to match exactly the case of the attribute as it is listed in the schema when the attribute is returned by the

Directory Server as the result of a search or modify operation. However, most client applications ignore the case of attributes; therefore, by default, this attribute is disabled. Do not modify it unless there are legacy clients that can check the case of attribute names in results returned from the server.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-return-exact-case: off

3.1.1.145. nsslapd-rewrite-rfc1274

This attribute is deprecated and will be removed in a later version.

This attribute is used only for LDAPv2 clients that require attribute types to be returned with their RFC 1274 names. Set the value to **on** for those clients. The default is **off**.

3.1.1.146. nsslapd-rootdn (Manager DN)

This attribute sets the distinguished name (DN) of an entry that is not subject to access control restrictions, administrative limit restrictions for operations on the directory, or resource limits in general. There does not have to be an entry corresponding to this DN, and by default there is not an entry for this DN, thus values like **cn=Directory Manager** are acceptable.

For information on changing the root DN, see the "Creating Directory Entries" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid distinguished name
Default Value	
Syntax	DN
Example	nsslapd-rootdn: cn=Directory Manager

3.1.1.147. nsslapd-rootpw (Root Password)

This attribute sets the password associated with the Manager DN. When the root password is provided, it is encrypted according to the encryption method selected for the **nsslapd-rootpwstorageScheme** attribute. When viewed from the server console, this attribute shows the value *. When viewed from the **dse.ldif** file, this attribute shows the encryption method followed by the encrypted string of the password. The example shows the password as displayed in the **dse.ldif** file, not the actual password.



WARNING

When the root DN is configured at server setup, a root password is required. However, it is possible for the root password to be deleted from **dse.ldif** by directly editing the file. In this situation, the root DN can only obtain the same access to the directory as allowed for anonymous access. Always make sure that a root password is defined in **dse.ldif** when a root DN is configured for the database. The **pwdhash** command-line utility can create a new root password. For more information, see [Section 9.6, “pwdhash”](#).



IMPORTANT

When resetting the Directory Manager’s password from the command line, *do not* use curly braces ({}) in the password. The root password is stored in the format {password-storage-scheme}hashed_password. Any characters in curly braces are interpreted by the server as the password storage scheme for the root password. If that text is not a valid storage scheme or if the password that follows is not properly hashed, then the Directory Manager cannot bind to the server.

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid password, encrypted by any one of the encryption methods which are described in Section 4.1.43, “Password Storage Schemes” .
Default Value	
Syntax	DirectoryString {encryption_method }encrypted_Password
Example	nsslapd-rootpw: {SSHA}9Eko69APCJfF

3.1.1.148. nsslapd-rootpwstorageScheme (Root Password Storage Scheme)

This attribute sets the method used to encrypt the Directory Server’s manager password stored in the **nsslapd-rootpw** attribute. For further details, such as recommended strong password storage schemes, see [Section 4.1.43, “Password Storage Schemes”](#).

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	See Section 4.1.43, "Password Storage Schemes" .
Default Value	PBKDF2_SHA256
Syntax	DirectoryString
Example	nsslapd-rootpwstoragescheme: PBKDF2_SHA256

3.1.1.149. nsslapd-rundir

This parameter sets the absolute path to the directory in which Directory Server stores run-time information, such as the PID file. The directory must be owned by the Directory Server user and group. Only this user and group must have read and write access in this directory.

The service must be restarted for changes to this attribute to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	Any directory writable by the Directory Server user
Default Value	/var/run/dirsrv/
Syntax	DirectoryString
Example	nsslapd-rundir: /var/run/dirsrv/

3.1.1.150. nsslapd-sasl-mapping-fallback

By default, only first matching SASL mapping is checked. If this mapping fails, the bind operation will fail even if there are other matching mappings that might have worked. SASL mapping fallback will keep checking all of the matching mappings.

You do not have to restart the server for this setting to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off

Parameter	Description
Syntax	DirectoryString
Example	nsslapd-sasl-mapping-fallback: off

3.1.1.151. nsslapd-sasl-max-buffer-size

This attribute sets the maximum SASL buffer size.

Parameter	Description
Entry DN	cn=config
Valid Values	0 to the maximum 32 bit integer value (2147483647)
Default Value	67108864 (64 kilobytes)
Syntax	Integer
Example	nsslapd-sasl-max-buffer-size: 67108864

3.1.1.152. nsslapd-saslpath

Sets the absolute path to the directory containing the Cyrus-SASL SASL2 plug-ins. Setting this attribute allows the server to use custom or non-standard SASL plug-in libraries. This is usually set correctly during installation, and Red Hat strongly recommends not changing this attribute. If the attribute is not present or the value is empty, this means the Directory Server is using the system provided SASL plug-in libraries which are the correct version.

If this parameter is set, the server uses the specified path for loading SASL plug-ins. If this parameter is not set, the server uses the **SASL_PATH** environment variable. If neither **nsslapd-saslpath** or **SASL_PATH** are set, the server attempts to load SASL plug-ins from the default location, **/usr/lib/sasl2**.

Changes made to this attribute will not take effect until the server is restarted.

Parameter	Description
Entry DN	cn=config
Valid Values	Path to plug-ins directory.
Default Value	Platform dependent
Syntax	DirectoryString

Parameter	Description
Example	nsslapd-saslpath: /usr/lib/sasl2

3.1.1.153. nsslapd-schema-ignore-trailing-spaces (Ignore Trailing Spaces in Object Class Names)

Ignores trailing spaces in object class names. By default, the attribute is turned off. If the directory contains entries with object class values that end in one or more spaces, turn this attribute on. It is preferable to remove the trailing spaces because the LDAP standards do not allow them.

For performance reasons, server restart is required for changes to take effect.

An error is returned by default when object classes that include trailing spaces are added to an entry. Additionally, during operations such as add, modify, and import (when object classes are expanded and missing superiors are added) trailing spaces are ignored, if appropriate. This means that even when **nsslapd-schema-ignore-trailing-spaces** is **on**, a value such as **top** is not added if **top** is already there. An error message is logged and returned to the client if an object class is not found and it contains trailing spaces.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-schema-ignore-trailing-spaces: on

3.1.1.154. nsslapd-schemacheck (Schema Checking)

This attribute sets whether the database schema is enforced when entries are added or modified. When this attribute has a value of **on**, Directory Server will not check the schema of existing entries until they are modified. The database schema defines the type of information allowed in the database. The default schema can be extended using the object classes and attribute types. For information on how to extend the schema using the Directory Server Console, see the "Extending the Directory Schema" chapter in the *Red Hat Directory Server Administration Guide*.

**WARNING**

Red Hat strongly discourages turning off schema checking. This can lead to severe interoperability problems. This is typically used for very old or non-standard LDAP data that must be imported into the Directory Server. If there are not a lot of entries that have this problem, consider using the **extensibleObject** object class in those entries to disable schema checking on a per entry basis.

**NOTE**

Schema checking works by default when database modifications are made using an LDAP client, such as **ldapmodify** or when importing a database from LDIF using **ldif2db**. If schema checking is turned off, every entry has to be verified manually to see that they conform to the schema. If schema checking is turned on, the server sends an error message listing the entries which do not match the schema. Ensure that the attributes and object classes created in the LDIF statements are both spelled correctly and identified in **dse.ldif**. Either create an LDIF file in the schema directory or add the elements to **99user.ldif**.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-schemacheck: on

3.1.1.155. nsslapd-schemadir

This is the absolute path to the directory containing the Directory Server instance-specific schema files. When the server starts up, it reads the schema files from this directory, and when the schema is modified through LDAP tools, the schema files in this directory are updated. This directory must be owned by the server user ID, and that user must have read and write permissions to the directory.

Changes made to this attribute will not take effect until the server is restarted.

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid path

Parameter	Description
Default Value	/etc/dirsrv/ <i>instance_name</i> /schema
Syntax	DirectoryString
Example	nsslapd-schemadir: /etc/dirsrv/ <i>instance_name</i> /schema

3.1.1.156. nsslapd-schemamod

Online schema modifications require a lock protection that are impacting the performance. If schema modifications are disabled, setting this parameter to **off** can increase the performance.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-schemamod: on

3.1.1.157. nsslapd-schemareplace

Determines whether modify operations that replace attribute values are allowed on the **cn=schema** entry.

Parameter	Description
Entry DN	cn=config
Valid Values	on off replication-only
Default Value	replication-only
Syntax	DirectoryString
Example	nsslapd-schemareplace: replication-only

3.1.1.158. nsslapd-search-return-original-type-switch

If the attribute list passed to a search contains a space followed by other characters, the same string is returned to the client. For example:

```
# ldapsearch -b <basedn> "(filter)" "sn someothertext"
dn: <matched dn>
sn someothertext: <sn>
```

This behavior is disabled by default, but can be enabled using this configuration parameter.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-search-return-type-switch: off

3.1.1.159. nsslapd-securelistenhost

This attribute allows multiple Directory Server instances to run on a multihomed machine (or makes it possible to limit listening to one interface of a multihomed machine). There can be multiple IP addresses associated with a single host name, and these IP addresses can be a mix of both IPv4 and IPv6. This parameter can be used to restrict the Directory Server instance to a single IP interface; this parameter also specifically sets what interface to use for TLS traffic rather than regular LDAP connections.

If a host name is given as the **nsslapd-securelistenhost** value, then the Directory Server responds to requests for every interface associated with the host name. If a single IP interface (either IPv4 or IPv6) is given as the **nsslapd-securelistenhost** value, Directory Server only responds to requests sent to that specific interface. Either an IPv4 or IPv6 address can be used.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=config
Valid Values	Any secure host name, IPv4 or IPv6 address
Default Value	
Syntax	DirectoryString
Example	nsslapd-securelistenhost: ldaps.example.com

3.1.1.160. nsslapd-securePort (Encrypted Port Number)

This attribute sets the TCP/IP port number used for TLS communications. This selected port must be unique on the host system; make sure no other application is attempting to use the same port number. Specifying a port number of less than **1024** requires that Directory Server be started as **root**. The server sets its **uid** to the **nsslapd-localuser** value after startup.

The server only listens to this port if it has been configured with a private key and a certificate, and **nsslapd-security** is set to **on**; otherwise, it does not listen on this port.

The server has to be restarted for the port number change to be taken into account.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to 65535
Default Value	636
Syntax	Integer
Example	nsslapd-securePort: 636

3.1.1.161. nsslapd-security (Security)

This attribute sets whether the Directory Server is to accept TLS communications on its encrypted port. This attribute should be set to **on** for secure connections. To run with security on, the server must be configured with a private key and server certificate in addition to the other TLS configuration.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-security: off

3.1.1.162. nsslapd-sizelimit (Size Limit)

This attribute sets the maximum number of entries to return from a search operation. If this limit is reached, **ns-slapd** returns any entries it has located that match the search request, as well as an exceeded size limit error.

When no limit is set, **ns-slapd** returns every matching entry to the client regardless of the number found. To set a no limit value whereby the Directory Server waits indefinitely for the search to complete, specify a value of **-1** for this attribute in the **dse.ldif** file.

This limit applies to everyone, regardless of their organization.



NOTE

A value of **-1** on this attribute in **dse.ldif** file is the same as leaving the attribute blank in the server console, in that it causes no limit to be used. This cannot have a null value in **dse.ldif** file, as it is not a valid integer. It is possible to set it to **0**, which returns **size limit exceeded** for every search.

The corresponding user-level attribute is **nsSizeLimit**.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 to the maximum 32 bit integer value (2147483647)
Default Value	2000
Syntax	Integer
Example	nsslapd-sizelimit: 2000

3.1.1.163. nsslapd-snmp-index

This parameter controls the SNMP index number of the Directory Server instance.

If you have multiple Directory Server instances on the same host listening all on port 389 but on different network interfaces, this parameter allows you to set different SNMP index numbers for each instance.

Parameter	Description
Entry DN	cn=config
Valid Values	0 to the maximum 32 bit integer value (2147483647)
Default Value	0
Syntax	Integer
Example	nsslapd-snmp-index: 0

3.1.1.164. nsslapd-SSLclientAuth

**NOTE**

The **nsslapd-SSLclientAuth** parameter will be deprecated in a future release and is currently maintained for backward compatibility. Use the new parameter **nsSSLClientAuth**, stored under **cn=encryption,cn=config**, instead. See [Section 3.1.4.5, "nsSSLClientAuth"](#).

3.1.1.165. nsslapd-ssl-check-hostname (Verify Hostname for Outbound Connections)

This attribute sets whether an TLS-enabled Directory Server should verify authenticity of a request by matching the host name against the value assigned to the common name (**cn**) attribute of the subject name (**subjectDN** field) in the certificate being presented. By default, the attribute is set to **on**. If it is on and if the host name does not match the **cn** attribute of the certificate, appropriate error and audit messages are logged.

For example, in a replicated environment, messages similar to the following are logged in the supplier server's log files if it finds that the peer server's host name does not match the name specified in its certificate:

```
[DATE] - SSL alert: ldap_sasl_bind:"",LDAP_SASL_EXTERNAL) 81 (Netscape runtime error -12276 -
  Unable to communicate securely with peer: requested domain name does not
  match the server's certificate.)
```



```
[DATE] NSMMReplicationPlugin - agmt="cn=SSL Replication Agreement to host1"
  (host1.example.com:636):
  Replication bind with SSL client authentication failed:
  LDAP error 81 (Can't contact LDAP server)
```

Red Hat recommends turning this attribute on to protect Directory Server's outbound TLS connections against a man in the middle (MITM) attack.

**NOTE**

DNS and reverse DNS must be set up correctly in order for this to work; otherwise, the server cannot resolve the peer IP address to the host name in the subject DN in the certificate.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-ssl-check-hostname: on

3.1.1.166. nsslapd-syntaxcheck

This attribute validates all modifications to entry attributes to make sure that the new or changed values conform to the required syntax for that attribute type. Any changes which do not conform to the proper syntax are rejected, when this attribute is enabled. All attribute values are validated against the syntax definitions in [RFC 4514](#).

By default, this is turned on.

Syntax validation is only run against new or modified attributes; it does not validate the syntax of existing attribute values. Syntax validation is triggered for LDAP operations such as adds and modifies; it does not happen after operations like replication, since the validity of the attribute syntax should be checked on the originating supplier.

This validates all supported attribute types for Directory Server, with the exception of binary syntaxes (which cannot be verified) and non-standard syntaxes, which do not have a defined required format. The *unvalidated* syntaxes are as follows:

- Fax (binary)
- OctetString (binary)
- JPEG (binary)
- Binary (non-standard)
- Space Insensitive String (non-standard)
- URI (non-standard)

The **nsslapd-syntaxcheck** attribute sets whether to validate and reject attribute modifications. This can be used with the [Section 3.1.1.167, “nsslapd-syntaxlogging”](#) attribute to write warning messages about invalid attribute values to the error logs.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-syntaxcheck: on

3.1.1.167. nsslapd-syntaxlogging

This attribute sets whether to log syntax validation failures to the errors log. By default, this is turned off.

If the [Section 3.1.1.166, “nsslapd-syntaxcheck”](#) attribute is enabled (the default) and the **nsslapd-syntaxlogging** attribute is also enabled, then any invalid attribute change is rejected and written to the errors log. If only **nsslapd-syntaxlogging** is enabled and **nsslapd-syntaxcheck** is disabled, then invalid changes are allowed to proceed, but a warning message is written to the error log.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-syntaxlogging: off

3.1.1.168. nsslapd-threadnumber (Thread Number)

This performance tuning-related value sets the number of threads, Directory Server creates at startup. If the value is set to **-1** (default), Directory Server enables the optimized auto-tuning based on the available hardware. Note that if auto-tuning is enabled, the **nsslapd-threadnumber** shows the auto-generated number of threads while Directory Server is running.



NOTE

Red Hat recommends to use the auto-tuning setting for optimized performance.

For further details, see the corresponding section in the [Red Hat Directory Server Performance Tuning Guide](#).

Parameter	Description
Entry DN	cn=config
Valid Range	-1 to the maximum number of threads supported by the system's thread and processor limits
Default Value	-1
Syntax	Integer
Example	nsslapd-threadnumber: -1

3.1.1.169. nsslapd-timelimit (Time Limit)

This attribute sets the maximum number of seconds allocated for a search request. If this limit is reached, Directory Server returns any entries it has located that match the search request, as well as an exceeded time limit error.

When no limit is set, **ns-slapd** returns every matching entry to the client regardless of the time it takes. To set a no limit value whereby Directory Server waits indefinitely for the search to complete, specify a value of **-1** for this attribute in the **dse.ldif** file. A value of zero (**0**) causes no time to be allowed for searches. The smallest time limit is 1 second.



NOTE

A value of **-1** on this attribute in the **dse.Idif** is the same as leaving the attribute blank in the server console in that it causes no limit to be used. However, a negative integer cannot be set in this field in the server console, and a null value cannot be used in the **dse.Idif** entry, as it is not a valid integer.

The corresponding user-level attribute is **nsTimeLimit**.

Parameter	Description
Entry DN	cn=config
Valid Range	-1 to the maximum 32 bit integer value (2147483647) in seconds
Default Value	3600
Syntax	Integer
Example	nsslapd-timelimit: 3600

3.1.1.170. nsslapd-tmpdir

This is the absolute path of the directory the server uses for temporary files. The directory must be owned by the server user ID and the user must have read and write access. No other user ID should have read or write access to the directory. The default value is **/tmp**.

Changes made to this attribute will not take effect until the server is restarted.

3.1.1.171. nsslapd-unhashed-pw-switch

When you update the **userPassword** attribute, Directory Server encrypts the password and stores it in **userPassword**. However, in certain situations, for example, when synchronizing passwords with Active Directory (AD), Directory Server must pass the unencrypted password to a plug-in. In this case, the server stores the unencrypted password in the temporary **unhashed#user#password** attribute in the so-called **entry extension** and, depending on the scenario, also in the changelog. Note that Directory Server does not store the temporary **unhashed#user#password** attribute on the server's hard disk.

The **nsslapd-unhashed-pw-switch** parameter controls whether and how Directory Server stores the unencrypted password. For example, you must set **nsslapd-unhashed-pw-switch** to **on** to synchronize passwords from Directory Server to Active Directory.

You can set the parameter to one of the following values:

- **off**: Directory Server neither stores the unencrypted password in the entry extension nor in the changelog. Set this value if you do not use password synchronization with AD or any plug-ins that require access to the unencrypted password.
- **on**: Directory Server stores the unencrypted password in the entry extension and in the changelog. Set this value if you configure password synchronization with AD.

- **nolog**: Directory Server stores the unencrypted password only in the entry extension but not in the changelog. Set this value if local Directory Server plug-ins require access to the unencrypted password, but no password synchronization with AD is configured.

Parameter	Description
Entry DN	cn=config
Valid Values	off on nolog
Default Value	off
Syntax	DirectoryString
Example	nsslapd-unhashed-pw-switch: off

3.1.1.172. nsslapd-validate-cert

If the Directory Server is configured to run in TLS and its certificate expires, then the Directory Server cannot be started. The **nsslapd-validate-cert** parameter sets how the Directory Server should respond when it attempts to start with an expired certificate:

- **warn** allows the Directory Server to start successfully with an expired certificate, but it sends a warning message that the certificate has expired. This is the default setting.
- **on** validates the certificate and will prevent the server from restarting if the certificate is expired. This sets a hard failure for expired certificates.
- **off** disables all certificate expiration validation, so the server can start with an expired certificate without logging a warning.

Parameter	Description
Entry DN	cn=config
Valid Values	warn on off
Default Value	warn
Syntax	DirectoryString
Example	nsslapd-validate-cert: warn

3.1.1.173. nsslapd-verify-filter-schema

The **nsslapd-verify-filter-schema** parameter defines how Directory Server verifies search filters with attributes that are not specified in the schema.

You can set **nsslapd-verify-filter-schema** to one of the following options:

- **reject-invalid:** Directory Server rejects the filter with an error if it contains any unknown element.
- **process-safe:** Directory Server replaces unknown components with an empty set, and logs a warning with the **notes=F** flag in the **/var/log/dirsrv/slappd-instance_name/access** log file. Before you switch **nsslapd-verify-filter-schema** from **warn-invalid** or **off** to **process-safe**, monitor the access log and fix queries from applications that cause log entries with **notes=F** flag. Otherwise, the operation result changes and Directory Server might not return all the matching entries.
- **warn-invalid:** Directory Server logs a warning with the **notes=F** flag in the **/var/log/dirsrv/slappd-instance_name/access** log file, and continues scanning the full database.
- **off:** Directory Server does not verify filters.

Note that, for example, if you set **nsslapd-verify-filter-schema** to **warn-invalid** or **off**, a filter, such as **(&(non_existent_attribute=example)(uid=user_name))** evaluates the **uid=user_name** entry and returns it only if it contains **non_existent_attribute=example**. If you set **nsslapd-verify-filter-schema** to **process-safe**, Directory Server does not evaluate that entry and does not return it.



NOTE

Setting **nsslapd-verify-filter-schema** to **reject-invalid** or **process-safe** can prevent high load due to unindexed searches for attributes that are not specified in the schema.

Parameter	Description
Entry DN	cn=config
Valid Values	reject-invalid, process-safe, warn-invalid, off
Default Value	warn-invalid
Syntax	DirectoryString
Example	<code>nsslapd-verify-filter-schema: warn-invalid</code>

3.1.1.174. nsslapd-versionstring

This attribute sets the server version number. The build data is automatically appended when the version string is displayed.

Parameter	Description
Entry DN	cn=config
Valid Values	Any valid server version number.
Default Value	

Parameter	Description
Syntax	DirectoryString
Example	nsslapd-versionstring: Red Hat-Directory/11.3

3.1.1.175. nsslapd-workingdir

This is the absolute path of the directory that the server uses as its current working directory after startup. This is the value that the server would return as the value of the `getcwd()` function, and the value that the system process table shows as its current working directory. This is the directory a core file is generated in. The server user ID must have read and write access to the directory, and no other user ID should have read or write access to it. The default value for this attribute is the same directory containing the error log, which is usually `/var/log/dirsrv/slapd-instance`.

Changes made to this attribute will not take effect until the server is restarted.

3.1.1.176. passwordAllowChangeTime

This attribute specifies the length of time that must pass before the user is allowed to change his password.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Values	Any integer
Default Value	
Syntax	DirectoryString
Example	passwordAllowChangeTime: 5h

3.1.1.177. passwordChange (Password Change)

Indicates whether users may change their passwords.

This can be abbreviated to `pwdAllowUserChange`.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	passwordChange: on

3.1.1.178. passwordCheckSyntax (Check Password Syntax)

This attribute sets whether the password syntax is checked before the password is saved. The password syntax checking mechanism checks that the password meets or exceeds the password minimum length requirement and that the string does not contain any trivial words, such as the user's name or user ID or any attribute value stored in the **uid**, **cn**, **sn**, **givenName**, **ou**, or **mail** attributes of the user's directory entry.

Password syntax includes several different categories for checking:

- The length of string or tokens to use to compare when checking for trivial words in the password (for example, if the token length is three, then no string of three sequential characters in the user's UID, name, email address, or other parameters can be used in the password)
- Minimum number of number characters (0-9)
- Minimum number of uppercase ASCII alphabetic characters
- Minimum number of lowercase ASCII alphabetic characters
- Minimum number of special ASCII characters, such as !@#\$
- Minimum number of 8-bit characters
- Minimum number of character categories required per password; a category can be upper- or lower-case letters, special characters, digits, or 8-bit characters

This can be abbreviated to **pwdCheckSyntax**.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off

Parameter	Description
Syntax	DirectoryString
Example	passwordCheckSyntax: off

3.1.1.179. passwordDictCheck

If set to **on**, the **passwordDictCheck** parameter checks the password against the **CrackLib** dictionary. Directory Server rejects the password if the new password contains a dictionary word.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	passwordDictCheck: off

3.1.1.180. passwordExp (Password Expiration)

Indicates whether user passwords expire after a given number of seconds. By default, user passwords do not expire. Once password expiration is enabled, set the number of seconds after which the password expires using the **passwordMaxAge** attribute.

For more information on password policies, see the "Managing User Accounts" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	passwordExp: on

3.1.1.181. passwordExpirationTime

This attribute specifies the length of time that passes before the user's password expires.

Parameter	Description
Entry DN	cn=config
Valid Values	Any date, in integers
Default Value	none
Syntax	GeneralizedTime
Example	passwordExpirationTime: 202009011953

3.1.1.182. passwordExpWarned

This attribute indicates that a password expiration warning has been sent to the user.

Parameter	Description
Entry DN	cn=config
Valid Values	true false
Default Value	none
Syntax	DirectoryString
Example	passwordExpWarned: true

3.1.1.183. passwordGraceLimit (Password Expiration)

This attribute is only applicable if password expiration is enabled. After the user's password has expired, the server allows the user to connect for the purpose of changing the password. This is called a *grace login*. The server allows only a certain number of attempts before completely locking out the user. This attribute is the number of grace logins allowed. A value of **0** means the server does not allow grace logins.

Parameter	Description
Entry DN	cn=config
Valid Values	0 (off) to any reasonable integer
Default Value	0
Syntax	Integer

Parameter	Description
Example	passwordGraceLimit: 3

3.1.1.184. passwordHistory (Password History)

Enables password history. Password history refers to whether users are allowed to reuse passwords. By default, password history is disabled, and users can reuse passwords. If this attribute is set to **on**, the directory stores a given number of old passwords and prevents users from reusing any of the stored passwords. Set the number of old passwords the Directory Server stores using the **passwordInHistory** attribute.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	passwordHistory: on

3.1.1.185. passwordInHistory (Number of Passwords to Remember)

Indicates the number of passwords the Directory Server stores in history. Passwords that are stored in history cannot be reused by users. By default, the password history feature is disabled, meaning that the Directory Server does not store any old passwords, and so users can reuse passwords. Enable password history using the **passwordHistory** attribute.

To prevent users from rapidly cycling through the number of passwords that are tracked, use the **passwordMinAge** attribute.

This can be abbreviated to **pwdInHistory**.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to 24 passwords
Default Value	6

Parameter	Description
Syntax	Integer
Example	passwordInHistory: 7

3.1.1.186. passwordIsGlobalPolicy (Password Policy and Replication)

This attribute controls whether password policy attributes are replicated.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	passwordIsGlobalPolicy: off

3.1.1.187. passwordLegacyPolicy

Enables legacy password behavior. Older LDAP clients expected to receive an error to lock a user account once the maximum failure limit was exceeded. For example, if the limit were three failures, then the account was locked at the fourth failed attempt. Newer clients, however, expect to receive the error message when the failure limit is reached. For example, if the limit is three failures, then the account should be locked at the third failed attempt.

Because locking the account when the failure limit is exceeded is the older behavior, it is considered legacy behavior. It is enabled by default, but can be disabled to allow the new LDAP clients to receive the error at the expected time.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	passwordLegacyPolicy: on

3.1.1.188. passwordLockout (Account Lockout)

Indicates whether users are locked out of the directory after a given number of failed bind attempts. By default, users are not locked out of the directory after a series of failed bind attempts. If account lockout is enabled, set the number of failed bind attempts after which the user is locked out using the **passwordMaxFailure** attribute.

This can be abbreviated to **pwdLockOut**.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	passwordLockout: off

3.1.1.189. passwordLockoutDuration (Lockout Duration)

Indicates the amount of time in seconds during which users are locked out of the directory after an account lockout. The account lockout feature protects against hackers who try to break into the directory by repeatedly trying to guess a user's password. Enable and disable the account lockout feature using the **passwordLockout** attribute.

This can be abbreviated to **pwdLockoutDuration**.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to the maximum 32 bit integer value (2147483647) in seconds
Default Value	3600
Syntax	Integer
Example	passwordLockoutDuration: 3600

3.1.1.190. passwordMaxAge (Password Maximum Age)

Indicates the number of seconds after which user passwords expire. To use this attribute, password expiration has to be enabled using the **passwordExp** attribute.

This can be abbreviated to **pwdMaxAge**.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to the maximum 32 bit integer value (2147483647) in seconds
Default Value	8640000 (100 days)
Syntax	Integer
Example	passwordMaxAge: 100

3.1.1.191. passwordBadWords

The **passwordBadWords** parameter defines a comma-separated list of strings that users are not allowed to use in a password.

Note that Directory Server does a case-insensitive comparison of the strings.

Parameter	Description
Entry DN	cn=config
Valid Values	Any string
Default Value	""
Syntax	DirectoryString
Example	passwordBadWords: example

3.1.1.192. passwordMaxClassChars

If you set the **passwordMaxClassChars** parameter to a value higher than **0**, Directory Server prevents setting a password that has more consecutive characters from the same category than the value set in the parameter. If enabled, Directory Server checks for consecutive characters of the following categories:

- digits
- alpha characters

- lower case
- upper case

For example, if you set **passwordMaxClassChars** to **3**, passwords containing, for example, **jdif** or **1947** are not allowed.

Parameter	Description
Entry DN	cn=config
Valid Range	0 (disabled) to maximum 32-bit integer (2147483647)
Default Value	0
Syntax	Integer
Example	passwordMaxClassChars: 0

3.1.1.193. passwordMaxFailure (Maximum Password Failures)

Indicates the number of failed bind attempts after which a user is locked out of the directory. By default, account lockout is disabled. Enable account lockout by modifying the **passwordLockout** attribute.

This can be abbreviated to **pwdMaxFailure**.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to maximum integer bind failures
Default Value	3
Syntax	Integer
Example	passwordMaxFailure: 3

3.1.1.194. passwordMaxRepeats (Password Syntax)

Maximum number of times the same character can appear sequentially in the password. Zero (**0**) is off. Integer values reject any password which used a character more than that number of times; for example, **1** rejects characters that are used more than once (**aa**) and **2** rejects characters used more than twice (**aaa**).

Parameter	Description
Entry DN	cn=config
Valid Range	0 to 64
Default Value	0
Syntax	Integer
Example	passwordMaxRepeats: 1

3.1.1.195. passwordMaxSeqSets

If you set the **passwordMaxSeqSets** parameter to a value higher than **0**, Directory Server rejects passwords with duplicate monotonic sequences exceeding the length set in the parameter. For example, if you set **passwordMaxSeqSets** to **2**, setting the password to **azXYZ_XYZ-g** is not allowed, because **XYZ** appears twice in the password.

Parameter	Description
Entry DN	cn=config
Valid Range	0 (disabled) to the maximum 32 bit integer value (2147483647)
Default Value	0
Syntax	Integer
Example	passwordMaxSeqSets: 0

3.1.1.196. passwordMaxSequence

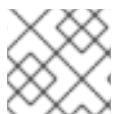
If you set the **passwordMaxSequence** parameter to a value higher than **0**, Directory Server rejects new passwords with a monotonic sequence longer than the value set in **passwordMaxSequence**. For example, if you set the parameter to **3**, Directory Server rejects passwords containing strings such as **1234** or **dcba**.

Parameter	Description
Entry DN	cn=config
Valid Range	0 (disabled) to the maximum 32 bit integer value (2147483647)
Default Value	0

Parameter	Description
Syntax	Integer
Example	passwordMaxSequence: 0

3.1.1.197. passwordMin8Bit (Password Syntax)

This sets the minimum number of 8-bit characters the password must contain.



NOTE

The 7-bit checking for **userPassword** must be disabled to use this.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to 64
Default Value	0
Syntax	Integer
Example	passwordMin8Bit: 0

3.1.1.198. passwordMinAge (Password Minimum Age)

Indicates the number of seconds that must pass before a user can change their password. Use this attribute in conjunction with the **passwordInHistory** (number of passwords to remember) attribute to prevent users from quickly cycling through passwords so that they can use their old password again. A value of zero (**0**) means that the user can change the password immediately.

This can be abbreviated to **pwdMaxFailure**.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to valid maximum integer
Default Value	0
Syntax	Integer

Parameter	Description
Example	passwordMinAge: 150

3.1.1.199. passwordMinAlphas (Password Syntax)

This attribute sets the minimum number of alphabetic characters password must contain.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to 64
Default Value	0
Syntax	Integer
Example	passwordMinAlphas: 4

3.1.1.200. passwordMinCategories (Password Syntax)

This sets the minimum number of character categories that are represented in the password. The categories are:

- Lowercase alphabetic characters
- Uppercase alphabetic characters
- Numbers
- Special ASCII characters, such as \$ and punctuation marks
- 8-bit characters

For example, if the value of this attribute were set to **2**, and the user tried to change the password to **aaaaaa**, the server would reject the password because it contains only lower case characters, and therefore contains characters from only one category. A password of **aAaAaA** would pass because it contains characters from two categories, uppercase and lowercase.

The default is **3**, which means that if password syntax checking is enabled, valid passwords have to have three categories of characters.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to 5

Parameter	Description
Default Value	0
Syntax	Integer
Example	passwordMinCategories: 2

3.1.1.201. PasswordMinDigits (Password Syntax)

This sets the minimum number of digits a password must contain.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to 64
Default Value	0
Syntax	Integer
Example	passwordMinDigits: 3

3.1.1.202. passwordMinLength (Password Minimum Length)

This attribute specifies the minimum number of characters that must be used in Directory Server user password attributes. In general, shorter passwords are easier to crack. Directory Server enforces a minimum password of eight characters. This is long enough to be difficult to crack but short enough that users can remember the password without writing it down.

This can be abbreviated to **pwdMinLength**.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Range	2 to 512 characters
Default Value	8
Syntax	Integer
Example	passwordMinLength: 8

3.1.1.203. PasswordMinLowers (Password Syntax)

This attribute sets the minimum number of lower case letters password must contain.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to 64
Default Value	0
Syntax	Integer
Example	passwordMinLowers: 1

3.1.1.204. PasswordMinSpecials (Password Syntax)

This attribute sets the minimum number of *special*, or not alphanumeric, characters a password must contain.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to 64
Default Value	0
Syntax	Integer
Example	passwordMinSpecials: 1

3.1.1.205. PasswordMinTokenLength (Password Syntax)

This attribute sets the smallest attribute value length that is used for *trivial* words checking. For example, if the **PasswordMinTokenLength** is set to **3**, then a **givenName** of **DJ** does not result in a policy that rejects **DJ** from being in the password, but the policy rejects a password containing the **givenName** of **Bob**.

Directory Server checks the minimum token length against values in the following attributes:

- **uid**
- **cn**
- **sn**
- **givenName**

- **mail**
- **ou**

If Directory Server should check additional attributes, you can set them in the **passwordUserAttributes** parameter. For details, see [Section 3.1.1.210, “passwordUserAttributes”](#).

Parameter	Description
Entry DN	cn=config
Valid Range	1 to 64
Default Value	3
Syntax	Integer
Example	passwordMinTokenLength: 3

3.1.1.206. PasswordMinUppers (Password Syntax)

This sets the minimum number of uppercase letters password must contain.

Parameter	Description
Entry DN	cn=config
Valid Range	0 to 64
Default Value	0
Syntax	Integer
Example	passwordMinUppers: 2

3.1.1.207. passwordMustChange (Password Must Change)

Indicates whether users must change their passwords when they first bind to the Directory Server or when the password has been reset by the Manager DN.

This can be abbreviated to **pwdMustChange**.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config

Parameter	Description
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	passwordMustChange: off

3.1.1.208. passwordPalindrome

If you enable the **passwordPalindrome** parameter, Directory Server rejects a password if the new password contains a palindrome.

A palindrome is a string which reads the same forward as backward, such as **abc11cba**.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	passwordPalindrome: off

3.1.1.209. passwordResetFailureCount (Reset Password Failure Count After)

Indicates the amount of time in seconds after which the password failure counter resets. Each time an invalid password is sent from the user's account, the password failure counter is incremented. If the **passwordLockout** attribute is set to **on**, users are locked out of the directory when the counter reaches the number of failures specified by the **passwordMaxFailure** attribute (within **600** seconds by default). After the amount of time specified by the **passwordLockoutDuration** attribute, the failure counter is reset to zero (**0**).

This can be abbreviated to **pwdFailureCountInterval**.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config

Parameter	Description
Valid Range	1 to the maximum 32 bit integer value (2147483647) in seconds
Default Value	600
Syntax	Integer
Example	passwordResetFailureCount: 600

3.1.1.210. passwordUserAttributes

By default, if you set a minimum token length in the **passwordMinTokenLength** parameter, Directory Server checks the tokens only against certain attributes. For details, see [Section 3.1.1.205, "PasswordMinTokenLength \(Password Syntax\)"](#).

The **passwordUserAttributes** parameter enables you to set a comma-separated list of additional attributes that Directory Server should check.

Parameter	Description
Entry DN	cn=config
Valid Values	Any string
Default Value	""
Syntax	DirectoryString
Example	passwordUserAttributes: telephoneNumber, I

3.1.1.211. passwordSendExpiringTime

When a client requests the password expiring control, Directory Server returns the "time to expire" value only if the password is within the warning period. To provide compatibility with existing clients that always expect this value to be returned – regardless if the password expiration time is within the warning period – the **passwordSendExpiringTime** parameter can be set to **on**.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off

Parameter	Description
Syntax	DirectoryString
Example	passwordSendExpiringTime: off

3.1.1.212. passwordStorageScheme (Password Storage Scheme)

This attribute sets the method used to encrypt user passwords stored in **userPassword** attributes. For further details, such as recommended strong password storage schemes, see [Section 4.1.43, "Password Storage Schemes"](#).



NOTE

Red Hat recommends not setting this attribute. If the value is not set, Directory Server automatically uses the strongest supported password storage scheme available. If a future Directory Server update changes the default value to increase security, passwords will be automatically encrypted using the new storage scheme if a user sets a password.

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	See Section 4.1.43, "Password Storage Schemes" .
Default Value	PBKDF2_SHA256
Syntax	DirectoryString
Example	passwordStorageScheme: PBKDF2_SHA256

3.1.1.213. passwordTPRDelayExpireAt

The **passwordTPRDelayExpireAt** attribute is part of the password policy. After the administrator sets a temporary password to a user account, **passwordTPRDelayExpireAt** defines the time in seconds before the temporary password expires.

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	-1 (disabled) to the maximum 32 bit integer value (2147483647)

Parameter	Description
Default Value	-1
Syntax	Integer
Example	passwordTPRDelayExpireAt: 3600

3.1.1.214. passwordTPRDelayValidFrom

The **passwordTPRDelayValidFrom** attribute is part of the password policy. After the administrator sets a temporary password to a user account, **passwordTPRDelayValidFrom** defines the time in seconds before a temporary password can be used.

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	-1 (disabled) to the maximum 32 bit integer value (2147483647)
Default Value	-1
Syntax	Integer
Example	passwordTPRDelayValidFrom: 60

3.1.1.215. passwordTPRMaxUse

The **passwordTPRMaxUse** attribute is part of the password policy. The attribute sets the number of times a user can authenticate successfully or not before the temporary password expires. If the authentication is successful, Directory Server only allows the user to change the password before other operations are possible. If the user does not change the password, the operation is terminated. The counter of the number of authentication attempts is increased regardless whether the authentication was successful or not.

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	cn=config
Valid Values	-1 (disabled) to the maximum 32 bit integer value (2147483647)

Parameter	Description
Default Value	-1
Syntax	Integer
Example	passwordTPRMaxUse: 5

3.1.1.216. passwordTrackUpdateTime

Sets whether to record a separate timestamp specifically for the last time that the password for an entry was changed. If this is enabled, then it adds the **pwdUpdateTime** operational attribute to the user account entry (separate from other update times, like **modifyTime**).

Using this timestamp can make it easier to synchronize password changes between different LDAP stores, such as Active Directory.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	passwordTrackUpdateTime: off

3.1.1.217. passwordUnlock (Unlock Account)

Indicates whether users are locked out of the directory for a specified amount of time or until the administrator resets the password after an account lockout. The account lockout feature protects against hackers who try to break into the directory by repeatedly trying to guess a user's password. If this **passwordUnlock** attribute is set to **off** and the operational attribute **accountUnlockTime** has a value of **0**, then the account is locked indefinitely.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Values	on off

Parameter	Description
Default Value	on
Syntax	DirectoryString
Example	passwordUnlock: off

3.1.1.218. passwordWarning (Send Warning)

Indicates the number of seconds before a user's password is due to expire that the user receives a password expiration warning control on their next LDAP operation. Depending on the LDAP client, the user may also be prompted to change their password at the time the warning is sent.

This can be abbreviated to **pwdExpireWarning**.

For more information on password policies, see the "Managing User Authentication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config
Valid Range	1 to the maximum 32 bit integer value (2147483647) in seconds
Default Value	86400 (1 day)
Syntax	Integer
Example	passwordWarning: 86400

3.1.1.219. retryCountResetTime

The **retryCountResetTime** attribute contains the date and time in UTC-format after which the **passwordRetryCount** attribute will be reset to **0**.

Parameter	Description
Entry DN	cn=config
Valid Range	Any valid time stamp in UTC format
Default Value	none
Syntax	Generalized Time

Parameter	Description
Example	retryCountResetTime: 20190618094419Z

3.1.2. cn=changelog5,cn=config

Multi-supplier replication changelog configuration entries are stored under the **cn=changelog5** entry. The **cn=changelog5,cn=config** entry is an instance of the **extensibleObject** object class.

The **cn=changelog5** entry must contain the following object classes:

- **top**
- **extensibleObject**



NOTE

Two different types of changelogs are maintained by Directory Server. The first type, which is stored here and referred to as the *changelog*, is used by multi-supplier replication; the second changelog, which is actually a plug-in and referred to as the *retro changelog*, is for compatibility with some legacy applications. See [Section 4.1.48, “Retro Changelog Plug-in”](#) for further information about the Retro Changelog Plug-in.

3.1.2.1. cn

This required attribute sets the relative distinguished name (RDN) of a changelog entry.

Parameter	Description
Entry DN	cn=changelog5,cn=config
Valid Values	Any string
Default Value	changelog5
Syntax	DirectoryString
Example	cn=changelog5

3.1.2.2. nsslapd-changelogcompactdb-interval

The Berkeley database does not reuse free pages unless the database is explicitly compacted. The compact operation returns the unused pages to the file system and the database file size shrinks. This parameter defines the interval in seconds when the changelog database is compacted. Note that compacting the database is resource-intensive, and thus should not be done too frequently.

This setting does not require a server restart to take effect.

Parameter	Description
Entry DN	cn=changelog5,cn=config
Valid Values	0 (no compaction) to 2147483647 seconds
Default Value	2592000 (30 days)
Syntax	Integer
Example	nsslapd-changelogcompactdb-interval: 2592000

3.1.2.3. nsslapd-changelogdir

This required attribute specifies the name of the directory in which the changelog entry is created. Whenever a changelog configuration entry is created, it must contain a valid directory; otherwise, the operation is rejected. The GUI proposes by default that this entry be stored in `/var/lib/dirsrv/slapd-instance/changelogdb/`.



WARNING

If the **cn=changelog5** entry is removed, the directory specified in the **nsslapd-changelogdir** parameter, including any subdirectories, are removed, with all of their contents.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=changelog5,cn=config
Valid Values	Any valid path to the directory storing the changelog
Default Value	None
Syntax	DirectoryString
Example	nsslapd-changelogdir: <code>/var/lib/dirsrv/slapd-<i>instance</i>/changelogdb/</code>

3.1.2.4. nsslapd-changelogmaxage (Max Changelog Age)

When synchronizing with a consumer, each update is stored in the changelog with a time stamp. The **nsslapd-changelogmaxage** parameter sets the maximum age of a record stored in the changelog.

Older records, that were successfully transferred to all replicas, are removed automatically. If the **nsslapd-changelogmaxage** and **nsslapd-changelogmaxentries** parameters are not set, all records are kept.



NOTE

The file size of the replication changelog is not automatically reduced if you set a lower value in the **nsslapd-changelogmaxentries** parameter. For further details, see the corresponding sections in the [Red Hat Directory Administration Guide](#).

The **nsslapd-changelogmaxage** parameter additionally sets the maximum age of entries in the retro changelog. The size of the retro changelog is automatically reduced when you set a lower value.

The trim operation is executed in intervals set in the **nsslapd-changelogtrim-interval** parameter.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=changelog5,cn=config
Valid Range	0 (meaning that entries are not removed according to their age) to maximum 32-bit integer (2147483647)
Default Value	0
Syntax	DirectoryString <i>IntegerAgeID</i> where <i>AgeID</i> is s for seconds, m for minutes, h for hours, d for days, and w for weeks
Example	nsslapd-changelogmaxage: 30d

3.1.2.5. nsslapd-changelogmaxentries (Max Changelog Records)

When synchronizing with a consumer, each update is stored in the changelog. The **nsslapd-changelogmaxentries** parameter sets the maximum number of records stored in the changelog. The oldest records, that were successfully transferred to all replicas and exceeding this number, are removed automatically. If the **nsslapd-changelogmaxentries** and **nsslapd-changelogmaxage** parameters are not set, all records are kept.



NOTE

The file size of the replication changelog is not automatically reduced if you set a lower value in the **nsslapd-changelogmaxentries** parameter. For further details, see the corresponding sections in the [Red Hat Directory Administration Guide](#).

The trim operation is executed in intervals set in the **nsslapd-changelogtrim-interval** parameter.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=changelog5,cn=config
Valid Range	0 (meaning that the only maximum limit is the disk size) to maximum 32-bit integer (2147483647)
Default Value	0
Syntax	Integer
Example	nsslapd-changelogmaxentries: 5000

3.1.2.6. nsslapd-changelogmaxconcurrentwrites (Max Concurrent Rewrites)

This attribute specifies the value used to initialize the new semaphore that controls the concurrent writes to the changelog. For information on the changelog, see [Section 3.1.2.3, “nsslapd-changelogdir”](#).

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=changelog5,cn=config
Valid Range	Maximum number of concurrent changelog writes
Default Value	2
Syntax	DirectoryString
Example	nsslapd-changelogmaxconcurrentwrites: 4

3.1.2.7. nsslapd-changelogtrim-interval (Replication Changelog Trimming Interval)

Directory Server repeatedly runs a trimming process on the changelog. To change the time between two runs, update the **nsslapd-changelogtrim-interval** parameter and set the interval in seconds.

This setting does not require a server restart to take effect.

Parameter	Description
Entry DN	cn=changelog5,cn=config
Valid Range	0 to the maximum 32 bit integer value (2147483647)
Default Value	300 (5 minutes)

Parameter	Description
Syntax	DirectoryString
Example	nsslapd-changelogtrim-interval: 300

3.1.2.8. nsslapd-encryptionalgorithm (Encryption Algorithm)

This attribute specifies the encryption algorithm used to encrypt the changelog. To enable the changelog encryption, the server certificate must be installed on the directory server. For information on the changelog, see [Section 3.1.2.3, “nsslapd-changelogdir”](#).

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=changelog5,cn=config
Valid Range	AES or 3DES
Default Value	None
Syntax	DirectoryString
Example	nsslapd-encryptionalgorithm: AES

3.1.2.9. nsSymmetricKey

This attribute stores the internally-generated symmetric key. For information on the changelog, see [Section 3.1.2.3, “nsslapd-changelogdir”](#).

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=changelog5,cn=config
Valid Range	Base 64-encoded key
Default Value	None
Syntax	DirectoryString
Example	None

3.1.3. Changelog Attributes

The changelog attributes contain the changes logged in the changelog.

3.1.3.1. changes

This attribute contains the changes made to the entry for add and modify operations in LDIF format.

OID	2.16.840.1.113730.3.1.8
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	Changelog Internet Draft

3.1.3.2. changeLog

This attribute contains the distinguished name of the entry which contains the set of entries comprising the server's changelog.

OID	2.16.840.1.113730.3.1.35
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Changelog Internet Draft

3.1.3.3. changeNumber

This attribute is always present. It contains an integer which uniquely identifies each change made to a directory entry. This number is related to the order in which the change occurred. The higher the number, the later the change.

OID	2.16.840.1.113730.3.1.5
Syntax	Integer
Multi- or Single-Valued	Multi-valued
Defined in	Changelog Internet Draft

3.1.3.4. changeTime

This attribute defines a time, in a **YYMMDDHHMMSS** format, when the entry was added.

OID	2.16.840.1.113730.3.1.77
-----	--------------------------

Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

3.1.3.5. changeType

This attribute specifies the type of LDAP operation, **add**, **delete**, **modify**, or **modrdn**. For example:

changeType: modify

OID	2.16.840.1.113730.3.1.7
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Changelog Internet Draft

3.1.3.6. deleteOldRdn

In the case of **modrdn** operations, this attribute specifies whether the old RDN was deleted.

A value of zero (**0**) will delete the old RDN. Any other non-zero value will keep the old RDN. (Non-zero values can be negative or positive integers.)

OID	2.16.840.1.113730.3.1.10
Syntax	Boolean
Multi- or Single-Valued	Multi-valued
Defined in	Changelog Internet Draft

3.1.3.7. filterInfo

This is used by the changelog for processing replication.

OID	2.16.840.1.113730.3.1.206
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued

Defined in	Directory Server
------------	------------------

3.1.3.8. newRdn

In the case of **modrdn** operations, this attribute specifies the new RDN of the entry.

OID	2.16.840.1.113730.3.1.9
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Changelog Internet Draft

3.1.3.9. newSuperior

In the case of **modrdn** operations, this attribute specifies the new parent (superior) entry for the moved entry.

OID	2.16.840.1.113730.3.1.11
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Changelog Internet Draft

3.1.3.10. targetDn

This attribute contains the DN of the entry that was affected by the LDAP operation. In the case of a **modrdn** operation, the **targetDn** attribute contains the DN of the entry before it was modified or moved.

OID	2.16.840.1.113730.3.1.6
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Changelog Internet Draft

3.1.4. cn=encryption

Encryption related attributes are stored under the **cn=encryption,cn=config** entry. The **cn=encryption,cn=config** entry is an instance of the **nsslapdEncryptionConfig** object class.

3.1.4.1. allowWeakCipher

This attribute controls whether weak ciphers are allowed or rejected. The default depends on the value set in the **nsSSL3Ciphers** parameter.

Ciphers are considered weak, if:

- They are exportable.
Exportable ciphers are labeled **EXPORT** in the cipher name. For example, in **TLS_RSA_EXPORT_WITH_RC4_40_MD5**.
- They are symmetrical and weaker than the 3DES algorithm.
Symmetrical ciphers use the same cryptographic keys for both encryption and decryption.
- The key length is shorter than 128 bits.

The server has to be restarted for changes to this attribute to take effect.

Entry DN	cn=encryption,cn=config
Valid Values	on off
Default Value	off , if the value in the nsSSL3Ciphers parameter is set to +all or default . on , if the value in the nsSSL3Ciphers parameter contains a user-specific cipher list.
Syntax	DirectoryString
Example	allowWeakCipher: on

3.1.4.2. allowWeakDHParam

The network security services (NSS) libraries linked with Directory Server requires minimum of 2048-bit Diffie-Hellman (DH) parameters. However, some clients connecting to Directory Server, such as Java 1.6 and 1.7 clients, only support 1024-bit DH parameters. The **allowWeakDHParam** parameter allows you to enable support for weak 1024-bit DH parameters in Directory Server.

The server has to be restarted for changes to this attribute to take effect.

Parameter	Description
Entry DN	cn=encryption,cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString

Parameter	Description
Example	allowWeakDHParam: off

3.1.4.3. nsSSL3Ciphers

This attribute specifies the set of TLS encryption ciphers Directory Server uses during encrypted communications.

The value set in this parameter influences the default value of the **allowWeakCipher** parameter. For details, see [Section 3.1.4.1, "allowWeakCipher"](#).

Parameter	Description
Entry DN	cn=encryption,cn=config
Valid Values	<p>Comma separated list of NSS supported ciphers. Additionally, the following parameters are possible:</p> <ul style="list-style-type: none"> * default: Enables the default ciphers advertised by NSS except weak ciphers. For further information, see List supported cipher suites for SSL connections * +all: All ciphers are enabled. This includes weak ciphers, if the allowWeakCipher parameter is enabled. * -all: All ciphers are disabled.
Default Value	default
Syntax	<p>DirectoryString</p> <p>Use the plus (+) symbol to enable or minus (-) symbol to disable, followed by the ciphers. Blank spaces are not allowed in the list of ciphers.</p> <p>To enable all ciphers – except rsa_null_md5, which must be specifically called – specify +all.</p>
Example	<pre>nsSSL3Ciphers: +TLS_RSA_AES_128_SHA,+TLS_RSA_AES_256_SHA, +TLS_RSA_WITH_AES_128_GCM_SHA256,- RSA_NULL_SHA</pre>

For details how to list all supported ciphers, see the corresponding section in the [Red Hat Directory Server Administration Guide](#).

3.1.4.4. nsSSLActivation

This attribute shows whether an TLS cipher family is enabled for a given security module.

Entry DN	<code>cn=encryptionType,cn=encryption,cn=config</code>
Valid Values	on off
Default Value	
Syntax	DirectoryString
Example	<code>nsSSLActivation: on</code>

3.1.4.5. nsSSLClientAuth

This attribute shows how the Directory Server enforces client authentication. It accepts the following values:

- **off** – the Directory Server will not accept client authentication
- **allowed** (default) – the Directory Server will accept client authentication, but not require it
- **required** – all clients must use client authentication.



IMPORTANT

The Directory Server Console does not support client authentication. Therefore, if the **nsSSLClientAuth** attribute is set to **required**, the Console cannot be used to manage the instance.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	<code>cn=config</code>
Valid Values	off allowed required
Default Value	allowed
Syntax	DirectoryString
Example	<code>nsSSLClientAuth: allowed</code>

3.1.4.6. nsSSLEnabledCiphers

Directory Server generates the multi-valued **nsSSLEnabledCiphers** attribute automatically. The attribute is read-only and displays the ciphers Directory Server currently uses. The list might not be the same as you set in the **nsSSL3Ciphers** attribute. For example, if you set weak ciphers in the

nsSSL3Ciphers attribute, but **allowWeakCipher** is disabled, the **nsSSLEnabledCiphers** attribute neither lists the weak ciphers nor does Directory Server use them.

Parameter	Description
Entry DN	cn=config
Valid Values	The values of this attribute are auto-generated and read-only.
Default Value	
Syntax	DirectoryString
Example	nsSSLClientAuth: TLS_RSA_WITH_AES_256_CBC_SHA::AES::SHA1::25 6

3.1.4.7. nsSSLPersonalitySSL

This attribute contains the certificate name to use for SSL.

Entry DN	cn=encryption,cn=config
Valid Values	A certificate nickname
Default Value	
Syntax	DirectoryString
Example:	nsSSLPersonalitySSL: Server-Cert

3.1.4.8. nsSSLSessionTimeout

This attribute sets the lifetime duration of a TLS connection. The minimum timeout value is **5** seconds. If a smaller value is set, then it is automatically replaced by **5** seconds. A value greater than the maximum value in the valid range below is replaced by the maximum value in the range.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=encryption,cn=config
Valid Range	5 seconds to 24 hours

Parameter	Description
Default Value	0, which means use the maximum value in the valid range above.
Syntax	Integer
Example	nsSSLSessionTimeout: 5

3.1.4.9. nsSSLSupportedCiphers

This attribute contains the supported ciphers for the server.

Entry DN	cn=encryption,cn=config
Valid Values	A specific family, cipher, and strength string
Default Value	
Syntax	DirectoryString
Example:	nsSSLSupportedCiphers: TLS_RSA_WITH_AES_256_CBC_SHA::AES::SHA1::25 6

3.1.4.10. nsSSLToken

This attribute contains the name of the token (security module) used by the server.

Entry DN	cn=encryption,cn=config
Valid Values	A module name
Default Value	
Syntax	DirectoryString
Example:	nsSSLToken: internal (software)

3.1.4.11. nsTLS1

Enables TLS version 1. The ciphers used with TLS are defined in the **nsSSL3Ciphers** attribute.

If the **sslVersionMin** and **sslVersionMax** parameters are set in conjunction with **nsTLS1**, Directory Server selects the most secure settings from these parameters.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=encryption,cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsTLS1: on

3.1.4.12. nsTLSAllowClientRenegotiation

Directory Server uses the **SSL_OptionSet()** network security services (NSS) function with the **SSL_ENABLE_RENEGOTIATION** option to control the TLS renegotiation behavior of NSS.

The **nsTLSAllowClientRenegotiation** attribute controls which values Directory Server passes to the **SSL_ENABLE_RENEGOTIATION** option:

- If you set **nsTLSAllowClientRenegotiation: on**, Directory Server passes **SSL_RENEGOTIATE_REQUIRE_XTN** to the **SSL_ENABLE_RENEGOTIATION** option. In this case, NSS allows secure renegotiations attempts using [RFC 5746](#).
- If you set **nsTLSAllowClientRenegotiation: off**, Directory Server passes **SSL_RENEGOTIATE_NEVER** to the **SSL_ENABLE_RENEGOTIATION** option. In this case, NSS denies all renegotiations attempts, even secure ones.

For further details about the NSS TLS renegotiation behavior, see the [The RFC 5746 implementation in NSS \(Network Security Services\)](#) section in the *Is Red Hat affected by TLS renegotiation MITM attacks (CVE-2009-3555)?* article.

The service must be restarted for changes to this attribute to take effect.

Parameter	Description
Entry DN	cn=encryption,cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsTLSAllowClientRenegotiation: on

3.1.4.13. sslVersionMin

The **sslVersionMin** parameter sets the minimum version of the TLS protocol Directory Server uses. However, by default, Directory Server sets this parameter automatically based on the system-wide crypto policy. If you set the crypto policy profile in the **/etc/crypto-policies/config** file to:

- **DEFAULT, FUTURE, or FIPS**, Directory Server sets **sslVersionMin** to **TLS1.2**
- **LEGACY**, Directory Server sets **sslVersionMin** to **TLS1.0**

Alternatively, you can manually set **sslVersionMin** to higher value than the one defined in the crypto policy.

The service must be restarted for changes to this attribute to take effect.

Entry DN	cn=encryption,cn=config
Valid Values	TLS protocol versions, such as TLS1.2
Default Value	Depends on the system-wide crypto policy profile you set.
Syntax	DirectoryString
Example:	sslVersionMin: TLS1.2

3.1.4.14. **sslVersionMax**

Sets the maximum version of the TLS protocol to be used. By default this value is set to the newest available protocol version in the NSS library installed on the system.

The server has to be restarted for changes to this attribute to go into effect.

If the **sslVersionMin** and **sslVersionMax** parameters are set in conjunction with **nsTLS1**, Directory Server selects the most secure settings from these parameters.

Entry DN	cn=encryption,cn=config
Valid Values	TLS protocol version such as TLS1.0
Default Value	Newest available protocol version in the NSS library installed on the system
Syntax	DirectoryString
Example:	sslVersionMax: TLS1.2

3.1.5. **cn=features**

There are not attributes for the **cn=features** entry itself. This entry is only used as a parent container entry, with the **nsContainer** object class.

The child entries contain an **oid** attribute to identify the feature and the **directoryServerFeature** object class, plus optional identifying information about the feature, such as specific ACLs. For example:

```
dn: oid=2.16.840.1.113730.3.4.9,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid: 2.16.840.1.113730.3.4.9
cn: VLV Request Control
aci: (targetattr != "aci")(version 3.0; acl "VLV Request Control"; allow( read, search, compare, proxy )
userdn = "ldap:///all");
creatorsName: cn=server,cn=plugins,cn=config
modifiersName: cn=server,cn=plugins,cn=config
createTimestamp: 20200129132357Z
modifyTimestamp: 20200129132357Z
```

3.1.5.1. oid

The **oid** attribute contains an object identifier assigned to a directory service feature. **oid** is used as the naming attribute for these directory features.

OID	2.16.840.1.113730.3.1.215
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

3.1.6. cn=mapping tree

- Configuration attributes for suffixes, replication, and Windows synchronization are stored under **cn=mapping tree,cn=config**. Configuration attributes related to suffixes are found under the suffix subentry **cn=suffix, cn=mapping tree,cn=config**. For example, a **suffix** is the root entry in the directory tree, such as **dc=example,dc=com**.
- Replication configuration attributes are stored under **cn=replica,cn=suffix, cn=mapping tree,cn=config**.
- Replication agreement attributes are stored under **cn=replicationAgreementName, cn=replica,cn=suffix, cn=mapping tree,cn=config**.
- Windows synchronization agreement attributes are stored under **cn=syncAgreementName, cn=replica,cn=suffix, cn=mapping tree,cn=config**.

3.1.7. Suffix Configuration Attributes under cn=suffix_DN

Suffix configurations are stored under the **cn="suffix_DN",cn=mapping tree,cn=config** entry. These entries are instances of the **nsMappingTree** object class. The **extensibleObject** object class enables entries that belong to it to hold any user attribute. For suffix configuration attributes to be taken into account by the server, these object classes, in addition to the **top** object class, must be present in the entry.

You must write the suffix DN in quotes because it contains characters such as equals signs (=), commas (,), and space characters. By using quotes, the DN appears correctly as a value in another DN. For example: **cn="dc=example,dc=com",cn=mapping tree,cn=config**

For further details, see the corresponding section in the [Directory Server Administration Guide](#).

3.1.7.1. cn

This mandatory attribute sets the relative distinguished name (RDN) of a new suffix.

Parameter	Description
Entry DN	<code>cn=suffix_DN,cn=mapping tree,cn=config</code>
Valid Values	Any valid LDAP DN
Default Value	
Syntax	DirectoryString
Example	<code>cn: dn=example,dc=com</code>

3.1.7.2. nsslapd-backend

This parameter sets the name of the database or database link used to process requests. It is multi-valued, with one database or database link per value. This attribute is required when the value of the **nsslapd-state** attribute is set to **backend** or **referral on update**.

Set the value to the name of the back-end database entry instance under **cn=ldbm database,cn=plugins,cn=config**. For example: **o=userroot,cn=ldbm database,cn=plugins,cn=config**

Parameter	Description
Entry DN	<code>cn=suffix_DN,cn=mapping tree,cn=config</code>
Valid Values	Any valid partition name
Default Value	
Syntax	DirectoryString
Example	<code>nsslapd-backend: userRoot</code>

3.1.7.3. nsslapd-distribution-function

The **nsslapd-distribution-function** parameter sets the name of the custom distribution function. You must set this attribute when you set more than one database in the **nsslapd-backend** attribute.

For further details about the custom distribution function, see the corresponding section in the [Directory Server Administration Guide](#).

Parameter	Description
Entry DN	<code>cn=suffix_DN,cn=mapping tree,cn=config</code>
Valid Values	Any valid distribution function
Default Value	
Syntax	DirectoryString
Example	<code>nsslapd-distribution-plugin: distribution_function_name</code>

3.1.7.4. nsslapd-distribution-plugin

The **nsslapd-distribution-plugin** sets the shared library to be used with the custom distribution function. You must set this attribute when you set more than one database in the **nsslapd-backend** attribute.

For further details about the custom distribution function, see the corresponding section in the [Directory Server Administration Guide](#).

Parameter	Description
Entry DN	<code>cn=suffix_DN,cn=mapping tree,cn=config</code>
Valid Values	Any valid distribution plug-in
Default Value	
Syntax	DirectoryString
Example	<code>nsslapd-distribution-plugin: /path/to/shared/library</code>

3.1.7.5. nsslapd-parent

If you want to create a sub suffix, use the **nsslapd-parent** attribute to define the parent suffix.

If the attribute is not set, the new suffix is created as a root suffix.

Parameter	Description
Entry DN	<code>cn=suffix_DN,cn=mapping tree,cn=config</code>
Valid Values	Any valid partition name

Parameter	Description
Default Value	
Syntax	DirectoryString
Example	nsslapd-parent-suffix: dc=example,dc=com

3.1.7.6. nsslapd-referral

This attribute sets the LDAP URL of the referral to be returned by the suffix. You can add the **nsslapd-referral** attribute multiple times to set multiple referral URLs.

You must set this attribute if you set the **nsslapd-state** parameter to **referral** or **on update**.

Parameter	Description
Entry DN	cn=suffix_DN,cn=mapping tree,cn=config
Valid Values	Any valid LDAP URL
Default Value	
Syntax	DirectoryString
Example	nsslapd-referral: ldap://example.com/

3.1.7.7. nsslapd-state

This parameter determines how a suffix handles operations. The attribute takes the following values:

- **backend**: The back-end database processes all operations.
- **disabled**: The database is not available for processing operations. The server returns a **No such search object** error in response to requests made by client applications.
- **referral**: Directory Server returns a referral URL for requests to this suffix.
- **referral on update**: The database is used for all operations. Only for update requests is a referral sent.

Parameter	Description
Entry DN	cn=suffix_DN,cn=mapping tree,cn=config
Valid Values	backend disabled referral referral on update
Default Value	backend

Parameter	Description
Syntax	DirectoryString
Example	nsslapd-state: backend

3.1.8. Replication Attributes under `cn=replica,cn=suffixDN,cn=mapping tree,cn=config`

Replication configuration attributes are stored under **`cn=replica,cn=suffix, cn=mapping tree,cn=config`**. The **`cn=replica`** entry is an instance of the **`nsDS5Replica`** object class. For replication configuration attributes to be taken into account by the server, this object class (in addition to the **`top`** object class) must be present in the entry. For further information about replication, see the "Managing Replication" chapter in the *Red Hat Directory Server Administration Guide*.

The **`cn=replica,cn=suffix,cn=mapping tree,cn=config`** entry must contain the following object classes:

- **`top`**
- **`extensibleObject`**
- **`nsds5replica`**

3.1.8.1. `cn`

Sets the naming attribute for the replica. The **`cn`** attribute must be set to **`replica`**.

Parameter	Description
Entry DN	<code>cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	The value must be set to <code>replica</code> .
Default Value	<code>replica</code>
Syntax	DirectoryString
Example	<code>cn=replica</code>

3.1.8.2. `nsds5DebugReplicaTimeout`

This attribute gives an alternate timeout period to use when the replication is run with debug logging. This can set only the time or both the time and the debug level:

```
nsds5debugreplicatimeout: seconds[:debuglevel]
```

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	Any numeric string
Default Value	
Syntax	DirectoryString
Example	nsds5debugreplicatimeout: 60:8192

3.1.8.3. nsDS5Flags

This attribute sets replica properties that were previously defined in flags. At present only one flag exists, which sets whether the log changes.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	0 1 * 0: The replica does not write to the changelog; this is the default for consumers. * 1: The replica writes to the changelog; this is the default for hubs and suppliers.
Default Value	0
Syntax	Integer
Example	nsDS5Flags: 0

3.1.8.4. nsDS5ReplConflict

Although this attribute is not in the **cn=replica** entry, it is used in conjunction with replication. This multi-valued attribute is included on entries that have a change conflict that cannot be resolved automatically by the synchronization process. To check for replication conflicts requiring administrator intervention, perform an LDAP search for (**nsDS5ReplConflict=***). For example:

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -s sub -b
dc=example,dc=com "(!(objectclass=nsTombstone)(nsDS5ReplConflict=*))" dn nsDS5ReplConflict
nsUniqueId
```

Using the search filter "**(objectclass=nsTombstone)**" also shows tombstone (deleted) entries. The value of the **nsDS5ReplConflict** contains more information about which entries are in conflict, usually by referring to them by their **nsUniqueId**. It is possible to search for a tombstone entry by its **nsUniqueId**.

For example:

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -s sub -b
dc=example,dc=com "(|(objectclass=nsTombstone)(nsUniqueID=66a2b699-1dd211b2-807fa9c3-
a58714648))"
```

3.1.8.5. nsDS5ReplicaAutoReferral

This attribute sets whether the Directory Server follows configured referrals for the database.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	on off
Default Value	
Syntax	DirectoryString
Example	nsDS5ReplicaAutoReferral: on

3.1.8.6. nsState

This attribute stores information on the state of the clock. It is designed only for internal use to ensure that the server cannot generate a change sequence number (**csn**) inferior to existing ones required for detecting backward clock errors.

3.1.8.7. nsDS5ReplicaAbortCleanRUV

This read-only attribute specifies whether the background task that removes old RUV entries for obsolete or missing suppliers is being aborted. See [Section 3.1.16.13, “cn=abort cleanallruv”](#) for more information about this task. A value of **0** means that the task is inactive, and a value of **1** means that the task is active.

This attribute is present to allow the abort task to be resumed after a server restart. When the task completes, the attribute is deleted.

The server ignores the modify request if this value is set manually.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	0 1
Default Value	None
Syntax	Integer

Parameter	Description
Example	nsDS5ReplicaAbortCleanRUV:1

3.1.8.8. nsds5ReplicaBackoffMin and nsds5ReplicaBackoffMax

These attributes are used in environments with heavy replication traffic, where updates need to be sent as fast as possible.

By default, if a remote replica is busy, the replication protocol will go into a "back off" state, and it will retry to send its updates at the next interval of the back-off timer. By default, the timer starts at 3 seconds, and has a maximum wait period of 5 minutes. As these default settings maybe not be sufficient under certain circumstances, you can use **nsds5ReplicaBackoffMin** and **nsds5ReplicaBackoffMax** to configure the minimum and maximum wait times.

The configuration settings can be applied while the server is online, and do not require a server restart. If invalid settings are used, then the default values are used instead. The configuration must be handled through CLI tools.

3.1.8.9. nsDS5ReplicaBindDN

This multi-valued attribute specifies the DN to use when binding. Although there can be more than one value in this **cn=replica** entry, there can only be one supplier bind DN per replication agreement. Each value should be the DN of a local entry on the consumer server. If replication suppliers are using client certificate-based authentication to connect to the consumers, configure the certificate mapping on the consumer to map the **subjectDN** in the certificate to a local entry.



IMPORTANT

For security reasons, do not set this attribute to **cn=Directory Manager**.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	Any valid DN
Default Value	
Syntax	DirectoryString
Example	nsDS5ReplicaBindDN: cn=replication manager,cn=config

3.1.8.10. nsDS5ReplicaBindDNGroup

The **nsDS5ReplicaBindDNGroup** attribute specifies a group DN. This group is then expanded and its members, including the members of its subgroups, are added to the **replicaBindDNs** attribute at startup or when the replica object is modified. This extends the current functionality provided by the

nsDS5ReplicaBindDN attribute, as it allows to set a group DN.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	Any valid group DN
Default Value	
Syntax	DirectoryString
Example	nsDS5ReplicaBindDNGroup: cn=sample_group,ou=groups,dc=example,dc=com

3.1.8.11. nsDS5ReplicaBindDNGroupCheckInterval

Directory Server checks for any changes in the groups specified in the **nsDS5ReplicaBindDNGroup** attribute and automatically rebuilds the list for the **replicaBindDN** parameter accordingly. These operations have a negative effect on performance and are therefore performed only at a specified interval set in the **nsDS5ReplicaBindDNGroupCheckInterval** attribute.

This attribute accepts the following values:

- **-1**: Disables the dynamic check at runtime. The administrator must restart the instance when the **nsDS5ReplicaBindDNGroup** attribute changes.
- **0**: Directory Server rebuilds the lists immediately after the groups are changed.
- Any positive 32-bit integer value: Minimum number of seconds that are required to pass since the last rebuild.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	-1 to maximum 32-bit integer (2147483647)
Default Value	-1
Syntax	Integer
Example	nsDS5ReplicaBindDNGroupCheckInterval: 0

3.1.8.12. nsDS5ReplicaChangeCount

This read-only attribute shows the total number of entries in the changelog and whether they still remain to be replicated. When the changelog is purged, only the entries that are still to be replicated remain.

See [Section 3.1.8.18, “nsDS5ReplicaPurgeDelay”](#) and [Section 3.1.8.23, “nsDS5ReplicaTombstonePurgeInterval”](#) for more information about purge operation properties.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	-1 to maximum 32-bit integer (2147483647)
Default Value	
Syntax	Integer
Example	nsDS5ReplicaChangeCount: 675

3.1.8.13. nsDS5ReplicaCleanRUV

This read-only attribute specifies whether the background task that removes old RUV entries for obsolete or missing suppliers is active. See [Section 3.1.16.12, “cn=cleanallruv”](#) for more information about this task. A value of **0** means that the task is inactive, and a value of **1** means that the task is active.

This attribute is present to allow the cleanup task to be resumed after a server restart. When the task completes, the attribute is deleted.

The server ignores the modify request if this value is set manually.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	0 1
Default Value	None
Syntax	Integer
Example	nsDS5ReplicaCleanRUV: 0

3.1.8.14. nsDS5Replicaid

This attribute sets the unique ID for suppliers in a given replication environment.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config

Parameter	Description
Valid Range	For suppliers: 1 to 65534 For consumers and hubs: 65535
Default Value	
Syntax	Integer
Example	nsDS5ReplicaId:1

3.1.8.15. nsDS5ReplicaLegacyConsumer

If this attribute is absent or has a value of **false**, then it means that the replica is not a legacy consumer.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	true false
Default Value	false
Syntax	DirectoryString
Example	nsDS5ReplicaLegacyConsumer: false

3.1.8.16. nsDS5ReplicaName

This attribute specifies the name of the replica with a unique identifier for internal operations. If it is not specified, this unique identifier is allocated by the server when the replica is created.



NOTE

It is recommended that the server be permitted to generate this name. However, in certain circumstances, for example, in replica role changes (supplier to hub etc.), this value needs to be specified. Otherwise, the server will not use the correct changelog database, and replication fails.

This attribute is destined for internal use only.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	

Parameter	Description
Default Value	
Syntax	DirectoryString (a UID identifies the replica)
Example	nsDS5ReplicaName: 66a2b699-1dd211b2-807fa9c3-a58714648

3.1.8.17. nsds5ReplicaProtocolTimeout

When stopping the server, disabling replication, or removing a replication agreement, there is a timeout on how long to wait before stopping replication when the server is under load. The **nsds5ReplicaProtocolTimeout** attribute can be used to configure this timeout and its default value is 120 seconds.

There may be scenarios where a timeout of 2 minutes is too long, or not long enough. For example, a particular replication agreement may need more time before ending a replication session during a shutdown.

This attribute can be added to the main replication configuration entry for a back end:

Parameter	Description
Entry DN	cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=map ping tree,cn=config
Valid Range	0 to maximum 32-bit integer (2147483647) in seconds
Default value	120
Syntax	Integer
Example	nsds5ReplicaProtocolTimeout: 120

The **nsds5ReplicaProtocolTimeout** attribute can also be added to a replication agreement. The replication agreement protocol timeout overrides the timeout set in the main replica configuration entry. This allows different timeouts for different replication agreements. If a replication session is in progress, a new timeout will abort that session and allow the server to shutdown.

3.1.8.18. nsDS5ReplicaPurgeDelay

This attribute controls the maximum age of deleted entries (tombstone entries) and state information.

The Directory Server stores tombstone entries and state information so that when a conflict occurs in a multi-supplier replication process, the server resolves the conflicts based on the timestamp and replica ID stored in the change sequence numbers.

An internal Directory Server housekeeping operation periodically removes tombstone entries which are

older than the value of this attribute (in seconds). State information which is older than the **nsDS5ReplicaPurgeDelay** value is removed when an entry which contains the state information is modified.

Not every tombstone and state information may be removed because, with multi-supplier replication, the server may need to keep a small number of the latest updates to prime replication, even if they are older than the value of the attribute.

This attribute specifies the interval, in seconds, to perform internal purge operations on an entry. When setting this attribute, ensure that the purge delay is longer than the longest replication cycle in the replication policy to preserve enough information to resolve replication conflicts and to prevent the copies of data stored in different servers from diverging.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	0 (keep forever) to maximum 32-bit integer (2147483647)
Default Value	604800 [1 week (60x60x24x7)]
Syntax	Integer
Example	nsDS5ReplicaPurgeDelay: 604800

3.1.8.19. nsDS5ReplicaReapActive

This read-only attribute specifies whether the background task that removes old tombstones (deleted entries) from the database is active. See [Section 3.1.8.23, "nsDS5ReplicaTombstonePurgeInterval"](#) for more information about this task. A value of **0** means that the task is inactive, and a value of **1** means that the task is active. The server ignores the modify request if this value is set manually.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	0 1
Default Value	
Syntax	Integer
Example	nsDS5ReplicaReapActive: 0

3.1.8.20. nsDS5ReplicaReferral

This multi-valued attribute specifies the user-defined referrals. This should only be defined on a consumer. User referrals are only returned when a client attempts to modify data on a read-only

consumer. This optional referral overrides the referral that is automatically configured by the consumer by the replication protocol.

The URL can use the format **ldap[s]://host_name:port_number** or **ldap[s]://IP_address:port_number**, with an IPv4 or IPv6 address.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	Any valid LDAP URL
Default Value	
Syntax	DirectoryString
Example	nsDS5ReplicaReferral: ldap://server.example.com:389

3.1.8.21. nsDS5ReplicaReleaseTimeout

This attribute, when used on suppliers and hubs in multi-supplier scenarios, determines a timeout period (in seconds) after which a supplier will release a replica. This is useful in situations when problems such as a slow network connection causes one supplier to acquire access to a replica and hold it for a long time, preventing all other suppliers from accessing it and sending updates. If this attribute is set, replicas are released by suppliers after the specified period, resulting in improved replication performance.

Setting this attribute to **0** disables the timeout. Any other value determines the length of the timeout in seconds.



IMPORTANT

Avoid setting this attribute to values between **1** and **30**. In most scenarios, short timeouts decrease the replication performance.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	0 to maximum 32-bit integer (2147483647) in seconds
Default Value	60
Syntax	Integer
Example	nsDS5ReplicaReleaseTimeout: 60

3.1.8.22. nsDS5ReplicaRoot

This attribute sets the DN at the root of a replicated area. This attribute must have the same value as the suffix of the database being replicated and cannot be modified.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	Suffix of the database being replicated, which is the suffix DN
Default Value	
Syntax	DirectoryString
Example	nsDS5ReplicaRoot: "dc=example,dc=com"

3.1.8.23. nsDS5ReplicaTombstonePurgeInterval

This attribute specifies the time interval in seconds between purge operation cycles.

Periodically, the server runs an internal housekeeping operation to purge old update and state information from the changelog and the main database. See [Section 3.1.8.18, “nsDS5ReplicaPurgeDelay”](#).

When setting this attribute, remember that the purge operation is time-consuming, especially if the server handles many delete operations from clients and suppliers.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Range	0 to maximum 32-bit integer (2147483647) in seconds
Default Value	86400 (1 day)
Syntax	Integer
Example	nsDS5ReplicaTombstonePurgeInterval: 86400

3.1.8.24. nsDS5ReplicaType

Defines the type of replication relationship that exists between this replica and the others.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	0 1 2 3 * 0 means unknown * 1 means primary (not yet used) * 2 means consumer (read-only) * 3 consumer/supplier (updateable)
Default Value	
Syntax	Integer
Example	nsDS5ReplicaType: 2

3.1.8.25. nsds5Task

This attribute launches a replication task, such as dumping the database contents to an LDIF file or removing obsolete suppliers from the replication topology.

You can set the **nsds5Task** attribute to one of the following values:

- **cl2ldif**: Exports the changelog to an LDIF file in the `/var/lib/dirsrv/slapd-instance_name/changelogdb/` directory.
- **ldif2cl**: Imports the changelog from an LDIF file stored in the `/var/lib/dirsrv/slapd-instance_name/changelogdb/` directory.
- **cleanruv**: Removes a Replica Update Vector (RUV) from the suppliers where you run the operation.
- **cleanallruv**: Removes RUVs from all servers in a replication topology.

You do not have to restart the server for this setting to take effect.

Parameter	Description
Entry DN	cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	* cl2ldif * ldif2cl * cleanruv * cleanallruv

Parameter	Description
Default Value	
Syntax	DirectoryString
Example	nsds5Task: cleanallruv

3.1.9. Replication Attributes under **cn=ReplicationAgreementName,cn=replica,cn=suffixName,cn=mapping tree,cn=config**

The replication attributes that concern the replication agreement are stored under **cn=ReplicationAgreementName, cn=replica, cn=suffixDN, cn=mapping tree, cn=config**. The **cn=ReplicationAgreementName** entry is an instance of the **nsDS5ReplicationAgreement** object class. Replication agreements are configured only on supplier replicas.

3.1.9.1. cn

This attribute is used for naming. Once this attribute has been set, it cannot be modified. This attribute is required for setting up a replication agreement.

Parameter	Description
Entry DN	<i>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</i>
Valid Values	Any valid cn
Default Value	
Syntax	DirectoryString
Example	cn: SupplierAtoSupplierB

3.1.9.2. description

Free form text description of the replication agreement. This attribute can be modified.

Parameter	Description
Entry DN	<i>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</i>
Valid Values	Any string
Default Value	

Parameter	Description
Syntax	DirectoryString
Example	description: Replication Agreement between Server A and Server B.

3.1.9.3. nsDS5ReplicaBindDN

This attribute sets the DN to use when binding to the consumer during replication. The value of this attribute must be the same as the one in **cn=replica** on the consumer replica. This may be empty if certificate-based authentication is used, in which case the DN used is the subject DN of the certificate, and the consumer must have appropriate client certificate mapping enabled. This can also be modified.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	Any valid DN (can be empty if client certificates are used)
Default Value	
Syntax	DirectoryString
Example	nsDS5ReplicaBindDN: cn=replication manager,cn=config

3.1.9.4. nsDS5ReplicaBindMethod

This attribute sets the method for the server to use to bind to the consumer server.

The **nsDS5ReplicaBindMethod** supports the following values:

- Empty or **SIMPLE**: The server uses password-based authentication. When using this bind method, additionally, set the **nsds5ReplicaBindDN** and **nsds5ReplicaCredentials** parameters to provide a user name and password.
- **SSLCLIENTAUTH**: Enables certificate-based authentication between the supplier and consumer. For this, the consumer server must have a certificate mapping configured to map the supplier's certificate to the replication manager entry.
- **SASL/GSSAPI**: Enables Kerberos authentication using SASL. This requires that the supplier server have a Kerberos keytab, and the consumer server a SASL mapping entry configured to map the supplier's Kerberos principal to the replication manager entry.
For further details, see the following sections in the *Red Hat Directory Server Administration Guide*:
 - [About the KDC Server and Keytabs](#)

- [Configuring SASL Identity Mapping from the Console](#)
- **SASL/DIGEST-MD5:** Enables password-based authentication using SASL with the **DIGEST-MD5** mechanism. When using this bind method, additionally, set the **nsds5ReplicaBindDN** and **nsds5ReplicaCredentials** parameters to provide a user name and password.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	SIMPLE SSLCLIENTAUTH SASL/GSSAPI SASL/DIGEST
Default Value	SIMPLE
Syntax	DirectoryString
Example	<code>nsDS5ReplicaBindMethod: SIMPLE</code>

3.1.9.5. nsds5ReplicaBootstrapBindDN

The **nsds5ReplicaBootstrapBindDN** parameter sets the fall-back bind distinguished name (DN) that Directory Server uses when the supplier fails to bind to a consumer due to an **LDAP_INVALID_CREDENTIALS (err=49)**, **LDAP_INAPPROPRIATE_AUTH (err=48)**, or **LDAP_NO SUCH OBJECT (err=32)** error.

In these cases, Directory Server uses the information from the **nsds5ReplicaBootstrapBindDN**, **nsds5ReplicaBootstrapCredentials**, **nsds5ReplicaBootstrapBindMethod**, and **nsds5ReplicaBootstrapTransportInfo** parameters to establish the connection. If the server also fails to establish the connection using these bootstrap settings, the server stops trying to connect.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	Any valid DN
Default Value	
Syntax	DirectoryString
Example	<code>nsds5ReplicaBootstrapBindDN: cn=replication manager,cn=config</code>

3.1.9.6. nsds5ReplicaBootstrapBindMethod

The **nsds5ReplicaBootstrapBindMethod** parameter sets the password for the fall-back login

mechanism that Directory Server uses when the supplier fails to bind to a consumer due to an **LDAP_INVALID_CREDENTIALS (err=49)**, **LDAP_INAPPROPRIATE_AUTH (err=48)**, or **LDAP_NO_SUCH_OBJECT (err=32)** error.

In these cases, Directory Server uses the information from the **nsds5ReplicaBootstrapBindDN**, **nsds5ReplicaBootstrapCredentials**, **nsds5ReplicaBootstrapBindMethod**, and **nsds5ReplicaBootstrapTransportInfo** parameters to establish the connection. If the server also fails to establish the connection using these bootstrap settings, the server stops trying to connect.

Parameter	Description
Entry DN	<i>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</i>
Valid Values	SIMPLE SSLCLIENTAUTH SASL/GSSAPI SASL/DIGEST
Default Value	
Syntax	DirectoryString
Example	nsds5ReplicaBootstrapBindMethod: SIMPLE

3.1.9.7. nsds5ReplicaBootstrapCredentials

The **nsds5ReplicaBootstrapCredentials** parameter sets the password for the fall-back bind distinguished name (DN) that Directory Server uses when the supplier fails to bind to a consumer due to an **LDAP_INVALID_CREDENTIALS (err=49)**, **LDAP_INAPPROPRIATE_AUTH (err=48)**, or **LDAP_NO_SUCH_OBJECT (err=32)** error.

In these cases, Directory Server uses the information from the **nsds5ReplicaBootstrapBindDN**, **nsds5ReplicaBootstrapCredentials**, **nsds5ReplicaBootstrapBindMethod**, and **nsds5ReplicaBootstrapTransportInfo** parameters to establish the connection. If the server also fails to establish the connection using these bootstrap settings, the server stops trying to connect.

Directory Server automatically hashes the password using the AES reversible password encryption algorithm when you set the parameter in clear text.

Parameter	Description
Entry DN	<i>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</i>
Valid Values	Any valid string.
Default Value	
Syntax	DirectoryString
Example	nsds5ReplicaBootstrapCredentials: password

3.1.9.8. nsds5ReplicaBootstrapTransportInfo

The **nsds5ReplicaBootstrapTransportInfo** parameter sets the encryption method for the connection to and from the replica for the fall-back connection that Directory Server uses when the supplier fails to bind to a consumer due to an **LDAP_INVALID_CREDENTIALS (err=49)**, **LDAP_INAPPROPRIATE_AUTH (err=48)**, or **LDAP_NO_SUCH_OBJECT (err=32)** error.

In these cases, Directory Server uses the information from the **nsds5ReplicaBootstrapBindDN**, **nsds5ReplicaBootstrapCredentials**, **nsds5ReplicaBootstrapBindMethod**, and **nsds5ReplicaBootstrapTransportInfo** parameters to establish the connection. If the server also fails to establish the connection using these bootstrap settings, the server stops trying to connect.

The attribute takes the following values:

- **TLS**: The connection uses the **StartTLS** command to start the encryption.
- **SSL**: The connection uses LDAPS with TLS encryption.
- **LDAP**: The connection is not encrypted.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	TLS SSL LDAP
Default Value	
Syntax	DirectoryString
Example	<code>nsds5ReplicaBootstrapTransportInfo: SSL</code>

3.1.9.9. nsDS5ReplicaBusyWaitTime

This attribute sets the amount of time in seconds a supplier should wait after a consumer sends back a busy response before making another attempt to acquire access. The default value is three (3) seconds. If the attribute is set to a negative value, Directory Server sends the client a message and an **LDAP_UNWILLING_TO_PERFORM** error code.

The **nsDS5ReplicaBusyWaitTime** attribute works in conjunction with the **nsDS5ReplicaSessionPauseTime** attribute. The two attributes are designed so that the **nsDS5ReplicaSessionPauseTime** interval is always at least one second longer than the interval specified for **nsDS5ReplicaBusyWaitTime**. The longer interval gives waiting suppliers a better chance to gain consumer access before the previous supplier can re-access the consumer.

Set the **nsDS5ReplicaBusyWaitTime** attribute at any time by using **changetype:modify** with the **replace** operation. The change takes effect for the next update session if one is already in progress.

Parameter	Description
-----------	-------------

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	Any valid integer
Default Value	3
Syntax	Integer
Example	<code>nsDS5ReplicaBusyWaitTime: 3</code>

3.1.9.10. nsDS5ReplicaChangesSentSinceStartup

This read-only attribute shows the number of changes sent to this replica since the server started. The actual value in the attribute is stored as a binary blob; in the Directory Server Console, this value is a ratio, in the form `replica_id:changes_sent/changes_skipped`. For example, for 100 changes sent and no changes skipped for replica 7, the attribute value is displayed in the Console as 7:100/0.

In the command line, the attribute value is shown in a binary form. For example:

```
nsds5replicaChangesSentSinceStartup:: MToxLzAg
```

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Range	0 to maximum 32-bit integer (2147483647)
Default Value	
Syntax	Integer
Example	<code>nsds5replicaChangesSentSinceStartup:: MToxLzAg</code>

3.1.9.11. nsDS5ReplicaCredentials

This attribute sets the credentials for the bind DN specified in the **nsDS5ReplicaBindDN** attribute. Directory Server uses this password to connect to the consumer.

The example below shows the encrypted value, as stored in the `/etc/dirsrv/slappd-instance_name/dse.ldif` file and not the actual password. To set a value, set it in clear text, for example **nsDS5ReplicaCredentials: password**. Directory Server then encrypts the password using the AES reversible password encryption schema when it stores the value.

When you use certificate-based authentication, this attribute does not have a value set.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	Any valid password
Default Value	
Syntax	DirectoryString {AES-Base64-algorithm-id}encoded_password
Example	<code>nsDS5ReplicaCredentials: {AES-TUhNRONT...}VoglUB8GG5A...</code>

3.1.9.12. nsds5ReplicaEnabled

This attribute sets whether a replication agreement is active, meaning whether replication is occurring per that agreement. The default is **on**, so that replication is enabled.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	<code>nsds5ReplicaEnabled: off</code>

3.1.9.13. nsds5ReplicaFlowControlPause

This parameters sets the time in milliseconds to pause after reaching the number of entries and updates set in the **nsds5ReplicaFlowControlWindow** parameter is reached. Updating both the **nsds5ReplicaFlowControlWindow** and **nsds5ReplicaFlowControlPause** parameters enables you to fine-tune the replication throughput. For further details, see [Section 3.1.9.14, "nsds5ReplicaFlowControlWindow"](#).

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	<code>cn=replication_agreement_name,cn=replica,cn=suffix_DN,cn=mapping tree,cn=config</code>

Parameter	Description
Valid Values	0 to maximum 64-bit long
Default Value	2000
Syntax	Integer
Example	nsds5ReplicaFlowControlPause: 2000

3.1.9.14. nsds5ReplicaFlowControlWindow

This attribute sets the maximum number of entries and updates sent by a supplier, which are not acknowledged by the consumer. After reaching the limit, the supplier pauses the replication agreement for the time set in the **nsds5ReplicaFlowControlPause** parameter. Updating both the **nsds5ReplicaFlowControlWindow** and **nsds5ReplicaFlowControlPause** parameters enables you to fine-tune the replication throughput.

Update this setting if the supplier sends entries and updates faster than the consumer can import or update, and acknowledge the data. In this case, the following message is logged in the supplier's error log file:

Total update flow control gives time (2000 msec) to the consumer before sending more entries [msgid sent: xxx, rcv: yyy]
If total update fails you can try to increase nsds5ReplicaFlowControlPause and/or decrease nsds5ReplicaFlowControlWindow in the replica agreement configuration

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	<code>cn=replication_agreement_name,cn=replica,cn=suffix_DN,cn=mapping tree,cn=config</code>
Valid Values	0 to maximum 64-bit long
Default Value	1000
Syntax	Integer
Example	nsds5ReplicaFlowControlWindow: 1000

3.1.9.15. nsDS5ReplicaHost

This attribute sets the host name for the remote server containing the consumer replica. Once this attribute has been set, it cannot be modified.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	Any valid host server name
Default Value	
Syntax	DirectoryString
Example	<code>nsDS5ReplicaHost: ldap2.example.com</code>

3.1.9.16. nsDS5ReplicaLastInitEnd

This optional, read-only attribute states when the initialization of the consumer replica ended.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	<code>YYYYMMDDhhmmssZ</code> is the date/time in Generalized Time form at which the connection was opened. This value gives the time in relation to Greenwich Mean Time. The hours are set with a 24-hour clock. The Z at the end indicates that the time is relative to Greenwich Mean Time.
Default Value	
Syntax	GeneralizedTime
Example	<code>nsDS5ReplicaLastInitEnd: 20200504121603Z</code>

3.1.9.17. nsDS5ReplicaLastInitStart

This optional, read-only attribute states when the initialization of the consumer replica started.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>

Parameter	Description
Valid Values	YYYYMMDDhhmmssZ is the date/time in Generalized Time form at which the connection was opened. This value gives the time in relation to Greenwich Mean Time. The hours are set with a 24-hour clock. The Z at the end indicates that the time is relative to Greenwich Mean Time.
Default Value	
Syntax	GeneralizedTime
Example	nsDS5ReplicaLastInitStart: 20200503030405

3.1.9.18. nsDS5ReplicaLastInitStatus

This optional, read-only attribute provides status for the initialization of the consumer. There is typically a numeric code followed by a short string explaining the status. Zero (**0**) means success.

Parameter	Description
Entry DN	<i>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</i>
Valid Values	0 (Consumer Initialization Succeeded), followed by any other status message.
Default Value	
Syntax	String
Example	nsDS5ReplicaLastInitStatus: 0 Consumer Initialization Succeeded

3.1.9.19. nsDS5ReplicaLastUpdateEnd

This read-only attribute states when the most recent replication schedule update ended.

Parameter	Description
Entry DN	<i>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</i>

Parameter	Description
Valid Values	YYYYMMDDhhmmssZ is the date/time in Generalized Time form at which the connection was opened. This value gives the time in relation to Greenwich Mean Time. The hours are set with a 24-hour clock. The Z at the end indicates that the time is relative to Greenwich Mean Time.
Default Value	
Syntax	GeneralizedTime
Example	nsDS5ReplicaLastUpdateEnd: 20200502175801Z

3.1.9.20. nsDS5ReplicaLastUpdateStart

This read-only attribute states when the most recent replication schedule update started.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	YYYYMMDDhhmmssZ is the date/time in Generalized Time form at which the connection was opened. This value gives the time in relation to Greenwich Mean Time. The hours are set with a 24-hour clock. The Z at the end indicates that the time is relative to Greenwich Mean Time.
Default Value	
Syntax	GeneralizedTime
Example	nsDS5ReplicaLastUpdateStart: 20200504122055Z

3.1.9.21. nsds5replicaLastUpdateStatus

In the read-only **nsds5replicaLastUpdateStatus** attribute of each replication agreement, Directory Server displays the latest status of the agreement. For a list of status, see [Appendix B, Replication Agreement Status](#).

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>

Parameter	Description
Valid Values	See Appendix B, Replication Agreement Status .
Default Value	
Syntax	DirectoryString
Example	nsds5replicaLastUpdateStatus: Error (0) Replica acquired successfully: Incremental update succeeded

3.1.9.22. nsDS5ReplicaPort

This attribute sets the port number for the remote server containing the replica. Once this attribute has been set, it cannot be modified.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	Port number for the remote server containing the replica
Default Value	
Syntax	Integer
Example	nsDS5ReplicaPort:389

3.1.9.23. nsDS5ReplicaReapActive

This read-only attribute specifies whether the background task that removes old tombstones (deleted entries) from the database is active. See [Section 3.1.8.23, "nsDS5ReplicaTombstonePurgeInterval"](#) for more information about this task. A value of zero (**0**) means that the task is inactive, and a value of **1** means that the task is active. If this value is set manually, the server ignores the modify request.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	0 1
Default Value	

Parameter	Description
Syntax	Integer
Example	nsDS5ReplicaReapActive: 0

3.1.9.24. nsDS5BeginReplicaRefresh

Initializes the replica. This attribute is absent by default. However, if this attribute is added with a value of **start**, then the server initializes the replica and removes the attribute value. To monitor the status of the initialization procedure, poll for this attribute. When initialization is finished, the attribute is removed from the entry, and the other monitoring attributes can be used for detailed status inquiries.

Parameter	Description
Entry DN	<i>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</i>
Valid Values	stop start
Default Value	
Syntax	DirectoryString
Example	nsDS5BeginReplicaRefresh: start

3.1.9.25. nsDS5ReplicaRoot

This attribute sets the DN at the root of a replicated area. This attribute must have the same value as the suffix of the database being replicated and cannot be modified.

Parameter	Description
Entry DN	<i>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</i>
Valid Values	Suffix of the database being replicated - same as suffixDN above
Default Value	
Syntax	DirectoryString
Example	nsDS5ReplicaRoot: "dc=example,dc=com"

3.1.9.26. nsDS5ReplicaSessionPauseTime

This attribute sets the amount of time in seconds a supplier should wait between update sessions. The default value is **0**. If the attribute is set to a negative value, Directory Server sends the client a message and an **LDAP_UNWILLING_TO_PERFORM** error code.

The **nsDS5ReplicaSessionPauseTime** attribute works in conjunction with the **nsDS5ReplicaBusyWaitTime** attribute. The two attributes are designed so that the **nsDS5ReplicaSessionPauseTime** interval is always at least one second longer than the interval specified for **nsDS5ReplicaBusyWaitTime**. The longer interval gives waiting suppliers a better chance to gain consumer access before the previous supplier can re-access the consumer.

- If either attribute is specified but not both, **nsDS5ReplicaSessionPauseTime** is set automatically to **1** second more than **nsDS5ReplicaBusyWaitTime**.
- If both attributes are specified, but **nsDS5ReplicaSessionPauseTime** is less than or equal to **nsDS5ReplicaBusyWaitTime**, **nsDS5ReplicaSessionPauseTime** is set automatically to **1** second more than **nsDS5ReplicaBusyWaitTime**.

When setting the values, ensure that the **nsDS5ReplicaSessionPauseTime** interval is at least **1** second longer than the interval specified for **nsDS5ReplicaBusyWaitTime**. Increase the interval as needed until there is an acceptable distribution of consumer access among the suppliers.

Set the **nsDS5ReplicaSessionPauseTime** attribute at any time by using **changetype:modify** with the **replace** operation. The change takes effect for the next update session if one is already in progress.

If Directory Server has to reset the value of **nsDS5ReplicaSessionPauseTime** automatically, the value is changed internally only. The change is not visible to clients, and it is not saved to the configuration file. From an external viewpoint, the attribute value appears as originally set.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	Any valid integer
Default Value	0
Syntax	Integer
Example	<code>nsDS5ReplicaSessionPauseTime: 0</code>

3.1.9.27. nsds5ReplicaStripAttrs

Fractional replication allows a list of attributes which are removed from replication updates (**nsDS5ReplicatedAttributeList**). However, a change to an excluded attribute still triggers a modify event and generates an empty replication update.

The **nsds5ReplicaStripAttrs** attribute adds a list of attributes which cannot be sent in an empty replication event and are stripped from the update sequence. This logically includes operational attributes like **modifiersName**.

If a replication event is *not* empty, the stripped attributes are replicated. These attributes are removed from updates only if the event would otherwise be empty.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Range	A space-separated list of any supported directory attribute
Default Value	
Syntax	DirectoryString
Example	<code>nsds5ReplicaStripAttrs: modifiersname modifytimestamp</code>

3.1.9.28. nsDS5ReplicatedAttributeList

This allowed attribute specifies any attributes that are *not* replicated to a consumer server. Fractional replication allows databases to be replicated across slow connections or to less secure consumers while still protecting sensitive information. By default, all attributes are replicated, and this attribute is not present. For more information on fractional replication, see the "Managing Replication" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Range	
Default Value	
Syntax	DirectoryString
Example	<code>nsDS5ReplicatedAttributeList: (objectclass=*) \$ EXCLUDE accountlockout memberof</code>

3.1.9.29. nsDS5ReplicatedAttributeListTotal

This allowed attribute specifies any attributes that are *not* replicated to a consumer server during a total update.

Fractional replication only replicates specified attributes. This improves the overall network performance. However, there may be times when administrators want to restrict some attributes using fractional replication during an incremental update but allow those attributes to be replicated during a total update (or vice versa).

By default, all attributes are replicated. **nsDS5ReplicatedAttributeList** sets the incremental replication list; if only **nsDS5ReplicatedAttributeList** is set, then this list applies to total updates as well.

nsDS5ReplicatedAttributeListTotal sets the list of attributes to exclude only from a total update.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Range	
Default Value	
Syntax	DirectoryString
Example	<code>nsDS5ReplicatedAttributeListTotal: (objectclass=*) \$EXCLUDE accountlockout</code>

3.1.9.30. nsDS5ReplicaTimeout

This allowed attribute specifies the number of seconds outbound LDAP operations waits for a response from the remote replica before timing out and failing. If the server writes **Warning: timed out waiting** messages in the error log file, then increase the value of this attribute.

Find out the amount of time the operation actually lasted by examining the access log on the remote machine, and then set the **nsDS5ReplicaTimeout** attribute accordingly to optimize performance.

Parameter	Description
Entry DN	<code>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Range	0 to maximum 32-bit integer value (2147483647) in seconds
Default Value	120
Syntax	Integer
Example	<code>nsDS5ReplicaTimeout: 120</code>

3.1.9.31. nsDS5ReplicaTransportInfo

This attribute sets the type of transport used for transporting data to and from the replica. This attribute cannot be modified once it is set.

The attribute takes the following values:

- **StartTLS**: The connection uses encryption using the **StartTLS** command.
- **LDAPS**: The connection uses TLS encryption.

- **LDAP:** The connection uses the unencrypted LDAP protocol. This value is also used, if the **nsDS5ReplicaTransportInfo** attribute is not set.

Parameter	Description
Entry DN	<i>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</i>
Valid Values	StartTLS LDAPS LDAP
Default Value	absent
Syntax	DirectoryString
Example	nsDS5ReplicaTransportInfo: StartTLS

3.1.9.32. nsDS5ReplicaUpdateInProgress

This read-only attribute states whether or not a replication update is in progress.

Parameter	Description
Entry DN	<i>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</i>
Valid Values	true false
Default Value	
Syntax	DirectoryString
Example	nsDS5ReplicaUpdateInProgress: true

3.1.9.33. nsDS5ReplicaUpdateSchedule

This multi-valued attribute specifies the replication schedule and can be modified. Changes made to this attribute take effect immediately. Modifying this value can be useful to pause replication and resume it later. For example, if this value to **0000-0001 0**, this in effect causes the server to stop sending updates for this replication agreement. The server continues to store them for replay later. If the value is later changed back to **0000-2359 0123456**, this makes replication immediately resume and sends all pending changes.

Parameter	Description
Entry DN	<i>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</i>

Parameter	Description
Valid Range	Time schedule presented as XXXX-YYYY 0123456, where XXXX is the starting hour, YYYY is the finishing hour, and the numbers 0123456 are the days of the week starting with Sunday.
Default Value	0000-2359 0123456 (all the time)
Syntax	Integer
Example	nsDS5ReplicaUpdateSchedule: 0000-2359 0123456

3.1.9.34. nsDS5ReplicaWaitForAsyncResults

In a replication environment, the **nsDS5ReplicaWaitForAsyncResults** parameter sets the time in milliseconds for which a supplier waits if the consumer is not ready before resending data.

Note that if you set the parameter to **0**, the default value is used.

Parameter	Description
Entry DN	<i>cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</i>
Valid Range	0 to maximum 32-bit integer (2147483647)
Default Value	100
Syntax	Integer
Example	nsDS5ReplicaWaitForAsyncResults: 100

3.1.9.35. nsDS50ruv

This attribute stores the last replica update vector (RUV) read from the consumer of this replication agreement. It is always present and must not be changed.

3.1.9.36. nsruvReplicaLastModified

This attribute contains the most recent time that an entry in the replica was modified and the changelog was updated.

3.1.9.37. nsds5ReplicaProtocolTimeout

When stopping the server, disabling replication, or removing a replication agreement, there is a timeout on how long to wait before stopping replication when the server is under load. The

nsds5ReplicaProtocolTimeout attribute can be used to configure this timeout and its default value is 120 seconds.

There may be scenarios where a timeout of 2 minutes is too long, or not long enough. For example, a particular replication agreement may need more time before ending a replication session during a shutdown.

This attribute can be added to the main replication configuration entry for a back end:

Parameter	Description
Entry DN	cn=replica,cn=dc\3Dexample\2Cdc\3Dcom,cn=map ping tree,cn=config
Valid Range	0 to maximum 32-bit integer (2147483647) in seconds
Default value	120
Syntax	Integer
Example	nsds5ReplicaProtocolTimeout: 120

The **nsds5ReplicaProtocolTimeout** attribute can also be added to a replication agreement. The replication agreement protocol timeout overrides the timeout set in the main replica configuration entry. This allows different timeouts for different replication agreements. If a replication session is in progress, a new timeout will abort that session and allow the server to shutdown.

3.1.10. Synchronization Attributes under **cn=syncAgreementName,cn=WindowsReplica,cn=suffixName,cn=mapping tree,cn=config**

The synchronization attributes that concern the synchronization agreement are stored under **cn=syncAgreementName, cn=WindowsReplica, cn=suffixDN, cn=mapping tree, cn=config**. The **cn=syncAgreementName** entry is an instance of the **nsDSWindowsReplicationAgreement** object class. For synchronization agreement configuration attributes to be taken into account by the server, this object class (in addition to the **top** object class) must be present in the entry. Synchronization agreements are configured only on databases that are enabled to synchronize with Windows Active Directory servers.

Table 3.6. List of Attributes Shared Between Replication and Synchronization Agreements

cn	nsDS5ReplicaLastUpdateEnd
description	nsDS5ReplicaLastUpdateStart
nsDS5ReplicaBindDN (the Windows sync manager ID)	nsDS5ReplicaLastUpdateStatus
nsDS5ReplicaBindMethod	nsDS5ReplicaPort

nsDS5ReplicaBusyWaitTime	nsDS5ReplicaRoot
nsDS5ReplicaChangesSentSinceStartup	nsDS5ReplicaSessionPauseTime
nsDS5ReplicaCredentials (the Windows sync manager password)	nsDS5ReplicaTimeout
nsDS5ReplicaHost (the Windows host)	nsDS5ReplicaTransportInfo
nsDS5ReplicaLastInitEnd	nsDS5ReplicaUpdateInProgress
nsDS5ReplicaLastInitStart	nsDS5ReplicaUpdateSchedule
nsDS5ReplicaLastInitStatus	nsDS5Oruv
winSyncMoveAction	winSyncInterval
nsds5ReplicaStripAttrs	

3.1.10.1. nsds7DirectoryReplicaSubtree

The suffix or DN of the Directory Server subtree that is being synchronized.

Parameter	Description
Entry DN	<code>cn=syncAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	Any valid suffix or subsuffix
Default Value	
Syntax	DirectoryString
Example	<code>nsDS7DirectoryReplicaSubtree:ou=People,dc=example,dc=com</code>

3.1.10.2. nsds7DirsyncCookie

This string is created by Active Directory DirSync and gives the state of the Active Directory Server at the time of the last synchronization. The old cookie is sent to Active Directory with each Directory Server update; a new cookie is returned along with the Windows directory data. This means only entries which have changed since the last synchronization are retrieved.

Parameter	Description
Entry DN	<code>cn=syncAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	Any string
Default Value	
Syntax	DirectoryString
Example	<code>nsDS7DirsyncCookie::khDKJFBZsjBDSCkjsdhIU74DJJVBXDhfvjmfvbjhzxj</code>

3.1.10.3. nsds7NewWinGroupSyncEnabled

This attribute sets whether a new group created in the Windows sync peer is automatically synchronized by creating a new group on the Directory Server.

Parameter	Description
Entry DN	<code>cn=syncAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	on off
Default Value	
Syntax	DirectoryString
Example	<code>nsDS7NewWinGroupSyncEnabled: on</code>

3.1.10.4. nsds7NewWinUserSyncEnabled

This attribute sets whether a new entry created in the Windows sync peer is automatically synchronized by creating a new entry on the Directory Server.

Parameter	Description
Entry DN	<code>cn=syncAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	on off
Default Value	

Parameter	Description
Syntax	DirectoryString
Example	nsDS7NewWinUserSyncEnabled: on

3.1.10.5. nsds7WindowsDomain

This attribute sets the name of the Windows domain to which the Windows sync peer belongs.

Parameter	Description
Entry DN	<code>cn=syncAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	Any valid domain name
Default Value	
Syntax	DirectoryString
Example	nsDS7WinndowsDomain: DOMAINWORLD

3.1.10.6. nsds7WindowsReplicaSubtree

The suffix or DN of the Windows subtree that is being synchronized.

Parameter	Description
Entry DN	<code>cn=syncAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	Any valid suffix or subsuffix
Default Value	
Syntax	DirectoryString
Example	nsDS7WindowsReplicaSubtree: <code>cn=Users,dc=domain,dc=com</code>

3.1.10.7. oneWaySync

This attribute sets which direction to perform synchronization. This can either be from the Active Directory server to the Directory Server or from the Directory Server to the Active Directory server.

If this attribute is absent (the default), then the synchronization agreement is *bi-directional*, so changes made in both domains are synchronized.

Parameter	Description
Entry DN	cn=syncAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	toWindows fromWindows null
Default Value	
Syntax	DirectoryString
Example	oneWaySync: fromWindows

3.1.10.8. winSyncInterval

This attribute sets how frequently, in seconds, the Directory Server polls the Windows sync peer to look for changes in the Active Directory entries. If this entry is not set, the Directory Server checks the Windows server every five (5) minutes, meaning the default value is **300** (300 seconds).

This value can be set lower to write Active Directory changes over to the Directory Server faster or raised if the directory searches are taking too long.

Parameter	Description
Entry DN	cn=syncAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Valid Values	1 to the maximum 32-bit integer value (2147483647)
Default Value	300
Syntax	Integer
Example	winSyncInterval: 600

3.1.10.9. winSyncMoveAction

The synchronization process starts at the root DN to begin evaluating entries for synchronization. Entries are correlated based on the **samAccount** in the Active Directory and the **uid** attribute in Directory Server. The synchronization plug-in notes if a previously synced entry (based on the **samAccount/uid** relationship) is removed from the synced subtree either because it is deleted or moved, then the synchronization plug-in recognizes that the entry is no longer to be synced.

The **winSyncMoveAction** attribute for the synchronization agreement sets instructions on how to handle these moved entries:

- **none** takes no action, so if a synced Directory Server entry exists, it may be synced over to or create an Active Directory entry *within* scope. If no synced Directory Server entry exists, nothing happens at all (this is the default behavior).
- **unsync** removes any sync-related attributes (**ntUser** or **ntGroup**) from the Directory Server entry but otherwise leaves the Directory Server entry intact. The Active Directory and Directory Server entries exist in tandem.



IMPORTANT

There is a risk when unsyncing entries that the Active Directory entry may be deleted at a later time, and the Directory Server entry will be left intact. This can create data inconsistency issues, especially if the Directory Server entry is ever used to recreate the entry on the Active Directory side later.

- **delete** deletes the corresponding entry on the Directory Server side, regardless of whether it was ever synced with Active Directory (this was the default behavior in 9.0).



IMPORTANT

You almost never want to delete a Directory Server entry without deleting the corresponding Active Directory entry. This option is available only for compatibility with Directory Server 9.0 systems.

Parameter	Description
Entry DN	<code>cn=syncAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config</code>
Valid Values	none delete unsync
Default Value	none
Syntax	DirectoryString
Example	<code>winSyncMoveAction: unsync</code>

3.1.11. **cn=monitor**

Information used to monitor the server is stored under **cn=monitor**. This entry and its children are read-only; clients cannot directly modify them. The server updates this information automatically. This section describes the **cn=monitor** attributes. The only attribute that can be changed by a user to set access control is the **aci** attribute.

If the **nsslapd-counters** attribute in **cn=config** is set to **on** (the default setting), then all of the counters kept by the Directory Server instance increment using 64-bit integers, even on 32-bit machines or with a 32-bit version of Directory Server. For the **cn=monitor** entry, the 64-bit integers are used with the **opsinitiated**, **opscompleted**, **entriessent**, and **bytessent** counters.



NOTE

The **nsslapd-counters** attribute enables 64-bit support for these specific database and server counters. The counters which use 64-bit integers are not configurable; the 64-bit integers are either enabled for all the allowed counters or disabled for all allowed counters.

connection

This attribute lists open connections and associated status and performance related information and values. These are given in the following format:

connection: A:YYYYMMDDhhmmssZ:B:C:D:E:F:G:H:I:IP_address

For example:

**connection: 69:20200604081953Z:6086:6086:-
:cn=proxy,ou=special_users,dc=example,dc=test:0:11:27:7448846:ip=192.0.2.1**

- **A** is the connection number, which is the number of the slot in the connection table associated with this connection. This is the number logged as **slot=A** in the access log message when this connection was opened, and usually corresponds to the file descriptor associated with the connection. The attribute **dTableSize** shows the total size of the connection table.
- **YYYYMMDDhhmmssZ** is the date and time, in GeneralizedTime form, at which the connection was opened. This value gives the time in relation to Greenwich Mean Time.
- **B** is the number of operations received on this connection.
- **C** is the number of completed operations.
- **D** is **r** if the server is in the process of reading BER from the network, empty otherwise. This value is usually empty (as in the example).
- **E** this is the bind DN. This may be empty or have value of **NULLDN** for anonymous connections.
- **F** is the connection maximum threads state: **1** is in max threads, **0** is not.
- **G** is the number of times this thread has hit the maximum threads value.
- **H** is the number of operations attempted that were blocked by the maximum number of threads.
- **I** is the connection ID as reported in the logs as **conn=connection_ID**.
- **IP_address** is the IP address of the LDAP client.



NOTE

B and C for the initiated and completed operations should ideally be equal.

currentConnections

This attribute shows the number of currently open and active Directory Server connections.

totalConnections

This attribute shows the total number of Directory Server connections. This number includes connections that have been opened and closed since the server was last started in addition to the **currentConnections**.

dTableSize

This attribute shows the size of the Directory Server connection table. Each connection is associated with a slot in this table, and usually corresponds to the file descriptor used by this connection. See [Section 3.1.1.61, “nsslapd-conntablesize”](#) for more information.

readWaiters

This attribute shows the number of connections where some requests are pending and not currently being serviced by a thread in Directory Server.

opsInitiated

This attribute shows the number of Directory Server operations initiated.

opsCompleted

This attribute shows the number of Directory Server operations completed.

entriesSent

This attribute shows the number of entries sent by Directory Server.

bytesSent

This attribute shows the number of bytes sent by Directory Server.

currentTime

This attribute shows the current time, given in Greenwich Mean Time (indicated by **generalizedTime** syntax **Z** notation; for example, **20200202131102Z**).

startTime

This attribute shows the Directory Server start time given in Greenwich Mean Time, indicated by **generalizedTime** syntax **Z** notation. For example, **20200202131102Z**.

version

This attribute shows the Directory Server vendor, version, and build number. For example, **Red Hat/11.3.1 B2020.274.08**.

threads

This attribute shows the number of threads used by the Directory Server. This should correspond to **nsslapd-threadnumber** in **cn=config**.

nbackEnds

This attribute shows the number of Directory Server database back ends.

backendMonitorDN

This attribute shows the DN for each Directory Server database backend. For further information on monitoring the database, see the following sections:

- Section 4.4.9, “Database Attributes under `cn=attributeName,cn=encrypted attributes,cn=database_name,cn=ldbm database,cn=plugins,cn=config`”
- Section 4.4.5, “Database Attributes under `cn=database,cn=monitor,cn=ldbm database,cn=plugins,cn=config`”
- Section 4.5.4, “Database Link Attributes under `cn=monitor,cn=database instance name,cn=chaining database,cn=plugins,cn=config`”

3.1.12. `cn=replication`

This entry has no attributes. When configuring legacy replication, those entries are stored under this **`cn=replication`** node, which serves as a placeholder.

3.1.13. `cn=sasl`

Entries which contain SASL mapping configurations are stored under **`cn=mapping,cn=sasl,cn=config`**. The **`cn=sasl`** entry is an instance of the **`nsContainer`** object class. Each mapping underneath it is an instance of the **`nsSaslMapping`** object class.

3.1.13.1. `nsSaslMapBaseDNTemplate`

This attribute contains the search base DN template used in SASL identity mapping.

Parameter	Description
Entry DN	<code>cn=mapping_name,cn=mapping,cn=sasl,cn=config</code>
Valid Values	Any valid DN
Default Value	
Syntax	IA5String
Example	<code>nsSaslMapBaseDNTemplate: ou=People,dc=example,dc=com</code>

3.1.13.2. `nsSaslMapFilterTemplate`

This attribute contains the search filter template used in SASL identity mapping.

Parameter	Description
Entry DN	<code>cn=mapping_name,cn=mapping,cn=sasl,cn=config</code>
Valid Values	Any string
Default Value	

Parameter	Description
Syntax	IA5String
Example	nsSaslMapFilterTemplate: (cn=\1)

3.1.13.3. nsSaslMapPriority

Directory Server enables you to set multiple simple authentication and security layer (SASL) mappings. If SASL fallback is enabled by the **nsslapd-sasl-mapping-fallback** parameter, you can set the **nsSaslMapPriority** attribute to prioritize the individual SASL mappings.

This setting does not require a server restart to take effect.

Parameter	Description
Entry DN	cn=map <i>ping_name</i> ,cn=mapping,cn=sasl,cn=config
Valid Values	1 (highest priority) - 100 (lowest priority)
Default Value	100
Syntax	Integer
Example	nsSaslMapPriority: 100

3.1.13.4. nsSaslMapRegexString

This attribute contains a regular expression used to map SASL identity strings.

Parameter	Description
Entry DN	cn=map <i>ping_name</i> ,cn=mapping,cn=sasl,cn=config
Valid Values	Any valid regular expression
Default Value	
Syntax	IA5String
Example	nsSaslMapRegexString: \(.*\)

3.1.14. cn=SNMP

SNMP configuration attributes are stored under **cn=SNMP, cn=config**. The **cn=SNMP** entry is an instance of the **nsSNMP** object class.

3.1.14.1. nssnmpenabled

This attribute sets whether SNMP is enabled.

Parameter	Description
Entry DN	cn=SNMP,cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nssnmpenabled: off

3.1.14.2. nssnmporganization

This attribute sets the organization to which the Directory Server belongs.

Parameter	Description
Entry DN	cn=SNMP,cn=config
Valid Values	Organization name
Default Value	
Syntax	DirectoryString
Example	nssnmporganization: Red Hat, Inc.

3.1.14.3. nssnmplocation

This attribute sets the location within the company or organization where the Directory Server resides.

Parameter	Description
Entry DN	cn=SNMP,cn=config
Valid Values	Location
Default Value	
Syntax	DirectoryString

Parameter	Description
Example	nssnmplocation: B14

3.1.14.4. nssnmpcontact

This attribute sets the email address of the person responsible for maintaining the Directory Server.

Parameter	Description
Entry DN	cn=SNMP,cn=config
Valid Values	Contact email address
Default Value	
Syntax	DirectoryString
Example	nssnmpcontact: jerome@example.com

3.1.14.5. nssnmpdescription

Provides a unique description of the Directory Server instance.

Parameter	Description
Entry DN	cn=SNMP,cn=config
Valid Values	Description
Default Value	
Syntax	DirectoryString
Example	nssnmpdescription: Employee directory instance

3.1.14.6. nssnmpmasterhost

nssnmpmasterhost is deprecated. This attribute is deprecated with the introduction of **net-snmp**. The attribute still appears in **dse.ldif** but without a default value.

Parameter	Description
Entry DN	cn=SNMP,cn=config

Parameter	Description
Valid Values	machine host name or localhost
Default Value	<blank>
Syntax	DirectoryString
Example	nssnmpmasterhost: localhost

3.1.14.7. nssnmpmasterport

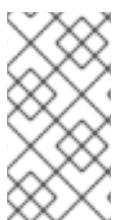
The **nssnmpmasterport** attribute was deprecated with the introduction of **net-snmp**. The attribute still appears in **dse.ldif** but without a default value.

Parameter	Description
Entry DN	cn=SNMP,cn=config
Valid Values	Operating system dependent port number. See the operating system documentation for further information.
Default Value	<blank>
Syntax	Integer
Example	nssnmpmasterport: 199

3.1.15. SNMP Statistic Attributes

Table 3.7, "SNMP Statistic Attributes" contains read-only attributes which list the statistics available for LDAP and SNMP clients. Unless otherwise noted, the value for the given attribute is the number of requests received by the server or results returned by the server since startup. Some of these attributes are not used by or are not applicable to the Directory Server but are still required to be present by SNMP clients.

If the **nsslapd-counters** attribute in **cn=config** is set to **on** (the default setting), then all of the counters kept by the Directory Server instance increment using 64-bit integers, even on 32-bit machines or with a 32-bit version of Directory Server. All of the SNMP statistics attributes use the 64-bit integers, if it is configured.



NOTE

The **nsslapd-counters** attribute enables 64-bit integers for these specific database and server counters. The counters which use 64-bit integers are not configurable; 64-bit integers are either enabled for all the allowed counters or disabled for all allowed counters.

Table 3.7. SNMP Statistic Attributes

Attribute	Description
AnonymousBinds	This shows the number of anonymous bind requests.
UnAuthBinds	This shows the number of unauthenticated (anonymous) binds.
SimpleAuthBinds	This shows the number of LDAP simple bind requests (DN and password).
StrongAuthBinds	This shows the number of LDAP SASL bind requests, for all SASL mechanisms.
BindSecurityErrors	This shows the number of times an invalid password was given in a bind request.
InOps	This shows the total number of all requests received by the server.
ReadOps	Not used. This value is always 0 .
CompareOps	This shows the number of LDAP compare requests.
AddEntryOps	This shows the number of LDAP add requests.
RemoveEntryOps	This shows the number of LDAP delete requests.
ModifyEntryOps	This shows the number of LDAP modify requests.
ModifyRDNOps	This shows the number of LDAP modify RDN (modrdn) requests.
ListOps	Not used. This value is always 0 .
SearchOps	This shows the number of LDAP search requests.
OneLevelSearchOps	This shows the number of one-level search operations.
WholeSubtreeSearchOps	This shows the number of subtree-level search operations.
Referrals	This shows the number of LDAP referrals returned.
Chainings	Not used. This value is always 0 .

Attribute	Description
SecurityErrors	This shows the number of errors returned that were security related, such as invalid passwords, unknown or invalid authentication methods, or stronger authentication required.
Errors	This shows the number of errors returned.
Connections	This shows the number of currently open connections.
ConnectionSeq	This shows the total number of connections opened, including both currently open and closed connections.
BytesRecv	This shows the number of bytes received.
BytesSent	This shows the number of bytes sent.
EntriesReturned	This shows the number of entries returned as search results.
ReferralsReturned	This provides information on referrals returned as search results (continuation references).
MasterEntries	Not used. This value is always 0 .
CopyEntries	Not used. This value is always 0 .
CacheEntries ^[a]	If the server has only one database back end, this is the number of entries cached in the entry cache. If the server has more than one database back end, this value is 0 , and see the monitor entry for each one for more information.
CacheHits	If the server has only one database back end, this is the number of entries returned from the entry cache, rather than from the database, for search results. If the server has more than one database back end, this value is 0 , and see the monitor entry for each one for more information.
SlaveHits	Not used. This value is always 0 .

[a] **CacheEntries** and **CacheHits** are updated every ten (10) seconds. Red Hat strongly encourages using the database back end specific monitor entries for this and other database information.

3.1.16. cn=tasks

Some core Directory Server tasks can be initiated by editing a directory entry using LDAP tools. These task entries are contained in **cn=tasks**. Each task can be invoked by updating an entry such as the following:

```
dn: cn=task_id,cn=task_type,cn=tasks,cn=config  
...
```

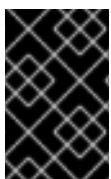
In Red Hat Directory Server deployments before Directory Server 8.0, many Directory Server tasks were managed by the Administration Server. These tasks were moved to the core Directory Server configuration in version 8.0 and are invoked and administered by Directory Server under the **cn=tasks** entry.

The following tasks are managed under the **cn=tasks** entry:

- [Section 3.1.16.2, "cn=import"](#)
- [Section 3.1.16.3, "cn=export"](#)
- [Section 3.1.16.4, "cn=backup"](#)
- [Section 3.1.16.5, "cn=restore"](#)
- [Section 3.1.16.6, "cn=index"](#)
- [Section 3.1.16.7, "cn=schema reload task"](#)
- [Section 3.1.16.8, "cn=memberof task"](#)
- [Section 3.1.16.9, "cn=fixup linked attributes"](#)
- [Section 3.1.16.10, "cn=syntax validate"](#)
- [Section 3.1.16.11, "cn=USN tombstone cleanup task"](#)
- [Section 3.1.16.12, "cn=cleanallruv"](#)
- [Section 3.1.16.13, "cn=abort cleanallruv"](#)
- [Section 3.1.16.14, "cn=automember rebuild membership"](#)
- [Section 3.1.16.15, "cn=automember export updates"](#)
- [Section 3.1.16.16, "cn=automember map updates"](#)

The common attributes for these tasks are listed in [Section 3.1.16.1, "Task Invocation Attributes for Entries under cn=tasks"](#).

The **cn=tasks** entry itself has no attributes and serves as the parent and container entry for the individual task entries.



IMPORTANT

Task entries are not permanent configuration entries. They only exist in the configuration file for as long as the task operation is running or until the **ttl** period expires. Then, the entry is deleted automatically by the server.

3.1.16.1. Task Invocation Attributes for Entries under cn=tasks

Five tasks which administer Directory Server instances have configuration entries which initiate and identify individual operations. These task entries are instances of the same object class, **extensibleObject**, and have certain common attributes which describe the state and behavior of Directory Server tasks. The task types can be import, export, backup, restore, index, schema reload, and memberof.

cn

The **cn** attribute identifies a new task operation to initiate. The **cn** attribute value can be anything, as long as it defines a new task.

Parameter	Description
Entry DN	<code>cn=task_name,cn=task_type,cn=tasks,cn=config</code>
Valid Values	Any string
Default Value	
Syntax	DirectoryString
Example	<code>cn: example task entry name</code>

nsTaskStatus

This attribute contains changing information about the status of the task, such as cumulative statistics or its current output message. The entire contents of the attribute may be updated periodically for as long as the process is running.

This attribute value is set by the server and *should not* be edited.

Parameter	Description
Entry DN	<code>cn=task_name,cn=task_type,cn=tasks,cn=config</code>
Valid Values	Any string
Default Value	
Syntax	case-exact string
Example	<code>nsTaskStatus: Loading entries....</code>

nsTaskLog

This entry contains all of the log messages for the task, including both warning and information messages. New messages are appended to the end of the entry value, so this attribute value grows larger, without erasing the original contents, by default.

Successful task operations, which have an **nsTaskExitCode** of **0**, are only recorded in the **nsTaskLog** attribute. Any non-zero response, which indicates an error, may be recorded in the error log as an error, but the error message is only recorded in the **nsTaskLog** attribute. For this reason, use the information in the **nsTaskLog** attribute to find out what errors actually occurred.

This attribute value is set by the server and *should not* be edited.

Parameter	Description
Entry DN	<code>cn=task_name,cn=task_type,cn=tasks,cn=config</code>
Valid Values	Any string
Default Value	
Syntax	Case-exact string
Example	nsTaskLog: example...

nsTaskExitCode

This attribute contains the exit code for the task. This attribute only exists after the task is completed and any value is only valid if the task is complete. The result code can be any LDAP exit code, as listed in [Section 7.4, “LDAP Result Codes”](#), but only a **0** value equals success; any other result code is an error.

This attribute value is set by the server and *should not* be edited.

Parameter	Description
Entry DN	<code>cn=task_name,cn=task_type,cn=tasks,cn=config</code>
Valid Values	0 (success) to 97 ^[a]
Default Value	
Syntax	Integer
Example	nsTaskExitCode: 0

[a] Any response other than **0** is an error.

nsTaskCurrentItem

This attribute shows the number of subtask which the task operation has completed, assuming the task can be broken down into subtasks. If there is only one task, then **nsTaskCurrentItem** is **0** while the task is running, and **1** when the task is complete. In this way, the attribute is analogous to a progress bar. When the **nsTaskCurrentItem** attribute has the same value as **nsTaskTotalItems**, then the task is completed.

This attribute value is set by the server and *should not* be edited.

Parameter	Description
Entry DN	<code>cn=task_name,cn=task_type,cn=tasks,cn=config</code>
Valid Values	0 to the maximum 32 bit integer value (2147483647)
Default Value	
Syntax	Integer
Example	<code>nsTaskCurrentItem: 148</code>

nsTaskTotalItems

This attribute shows the total number of subtasks that must be completed for the task operation. When the **nsTaskCurrentItem** attribute has the same value as **nsTaskTotalItems**, then the task is completed.

This attribute value is set by the server and *should not* be edited.

Parameter	Description
Entry DN	<code>cn=task_name,cn=task_type,cn=tasks,cn=config</code>
Valid Values	0 to the maximum 32 bit integer value (2147483647)
Default Value	
Syntax	Integer
Example	<code>nsTaskTotalItems: 152</code>

nsTaskCancel

This attribute allows a task to be aborted while in progress. This attribute can be modified by users.

Parameter	Description
Entry DN	<code>cn=task_name,cn=task_type,cn=tasks,cn=config</code>
Valid Values	true false
Default Value	
Syntax	Case-insensitive string

Parameter	Description
Example	nsTaskCancel: true

ttl

This attribute sets the amount of time (in seconds) the task entry will remain in the DSE after the task has finished or aborted. Setting a **ttl** attribute allows the task entry to be polled for new status information without missing the exit code. Setting the **ttl** attribute to **0** means that the entry is not cached.

Parameter	Description
Entry DN	<code>cn=task_name,cn=task_type,cn=tasks,cn=config</code>
Valid Values	0 (cannot be cached) to the maximum 32 bit integer value (2147483647)
Default Value	
Syntax	DirectoryString
Example	ttl: 120

3.1.16.2. cn=import

An LDIF file or multiple LDIF files can be imported through the command line by creating a special task entry which defines the parameters of the task and initiates the task. As soon as the task is complete, the task entry is removed from the directory.

The **cn=import** entry is a container entry for import task operations. The **cn=import** entry itself has no attributes, but each of the task entries within this entry, such as **cn=task_ID**, **cn=import**, **cn=tasks**, **cn=config**, uses the following attributes to define the import task.

An import task entry under **cn=import** must contain the LDIF file to import (in the **nsFilename** attribute) and the name of the instance into which to import the file (in the **nsInstance** attribute). Additionally, it must contain a unique **cn** to identify the task. For example:

```
dn: cn=example import,cn=import,cn=tasks,cn=config
objectclass: extensibleObject
cn: example import
nsFilename: /home/files/example.ldif
nsInstance: userRoot
```

As the import operation runs, the task entry will contain all of the server-generated task attributes listed in [Section 3.1.16.1, “Task Invocation Attributes for Entries under cn=tasks”](#).

There are some optional attributes which can be used to refine the import operation, similar to the options for the **ldif2db** and **ldif2db.pl** scripts:

- **nsIncludeSuffix**, which is analogous to the **-s** option to specify the suffix to import

- [nsExcludeSuffix](#), analogous to the **-x** option to specify a suffix or subtree to exclude from the import
- [nsImportChunkSize](#), analogous to the **-c** option to override starting a new pass during the import and merge the chunks
- [nsImportIndexAttrs](#), which sets whether to import attribute indexes (with no corollary in the script options)
- [nsUniqueldGenerator](#), analogous to the **-g** option to generate unique ID numbers for the entries
- [nsUniqueldGeneratorNamespace](#), analogous to the **-G** option to generate a unique, name-based ID for the entries

nsFilename

The **nsFilename** attribute contains the path and filenames of the LDIF files to import into the Directory Server instance. To import multiple files, add multiple instances of this attribute. For example:

```
nsFilename: file1.ldif
nsFilename: file2.ldif
```

Parameter	Description
Entry DN	<code>cn=task_name,cn=import,cn=tasks,cn=config</code>
Valid Values	Any string
Default Value	
Syntax	Case-exact string, multi-valued
Example	<code>nsFilename: /home/jsmith/example.ldif</code>

nsInstance

This attribute supplies the name of the database instance into which to import the files, such as **userRoot** or **slapd-example**.

Parameter	Description
Entry DN	<code>cn=task_name,cn=import,cn=tasks,cn=config</code>
Valid Values	The name of a Directory Server instance database (any string)
Default Value	
Syntax	Case-exact string

Parameter	Description
Example	nsInstance: userRoot

nsIncludeSuffix

This attribute identifies a specific suffix or subtree to import from the LDIF file.

Parameter	Description
Entry DN	cn=task_name,cn=import,cn=tasks,cn=config
Valid Values	Any DN
Default Value	
Syntax	DN, multi-valued
Example	nsIncludeSuffix: ou=people,dc=example,dc=com

nsExcludeSuffix

This attribute identifies suffixes or subtrees in the LDIF file to exclude from the import.

Parameter	Description
Entry DN	cn=task_name,cn=import,cn=tasks,cn=config
Valid Values	Any DN
Default Value	
Syntax	DN, multi-valued
Example	nsExcludeSuffix: ou=machines,dc=example,dc=com

nsImportChunkSize

This attribute defines the number of chunks to have during the import operation, which overrides the server's detection during the import of when to start a new pass and merges the chunks.

Parameter	Description
Entry DN	cn=task_name,cn=import,cn=tasks,cn=config
Valid Values	0 to the maximum 32 bit integer value (2147483647)

Parameter	Description
Default Value	0
Syntax	Integer
Example	nsImportChunkSize: 10

nsImportIndexAttrs

This attribute sets whether to index the attributes that are imported into database instance.

Parameter	Description
Entry DN	cn=task_name,cn=import,cn=tasks,cn=config
Valid Values	true false
Default Value	true
Syntax	Case-insensitive string
Example	nsImportIndexAttrs: true

nsUniqueIdGenerator

This sets whether to generate a unique ID for the imported entries. By default, this attribute generates time-based IDs.

Parameter	Description
Entry DN	cn=task_name,cn=import,cn=tasks,cn=config
Valid Values	none (no unique ID) empty (time-based ID) deterministic <i>namespace</i> (name-based ID)
Default Value	empty
Syntax	Case-insensitive string
Example	nsUniqueIdGenerator:

nsUniqueIdGeneratorNamespace

This attribute defines how to generate name-based IDs; the attribute sets the namespace to use to generate the IDs. This option is useful to import the same LDIF file into two Directory Server instances when the entries need to have the same IDs.

Parameter	Description
Entry DN	<code>cn=task_name,cn=import,cn=tasks,cn=config</code>
Valid Values	Any string
Default Value	
Syntax	Case-insensitive string
Example	<code>nsUniqueIdGeneratorNamespace: example</code>

3.1.16.3. **cn=export**

A database or multiple databases can be exported through the command line by creating a special task entry which defines the parameters of the task and initiates the task. As soon as the task is complete, the task entry is removed from the directory.

The **cn=export,cn=tasks,cn=config** entry is a container for export task operations. These tasks are stored within this container and named **cn=task_name,cn=export,cn=tasks,cn=config**.

While the export operation is running, the task entry contains all of the server-generated task attributes listed in [Section 3.1.16.1, “Task Invocation Attributes for Entries under cn=tasks”](#).

You can create export tasks manually or use the **db2ldif.pl** command. The following table displays the **db2ldif.pl** command-line options and their corresponding attributes:

db2ldif.pl option	Task attribute	Description
-a	nsFilename	Sets the path to the exported LDIF file.
-C	nsUseld2Entry	If enabled, use only the main database file only.
-M	nsUseOneFile	If enabled, store output in multiple files.
-n	nsInstance	Sets the database name.
-N	nsPrintKey	Enables you to suppress printing the sequence number.
-r	nsExportReplica	If set, the export will include attributes to initialize a replica.
-s	nsIncludeSuffix	Sets the suffix to include in the exported file.

db2ldif.pl option	Task attribute	Description
-u	nsDumpUniqId	Enables you not to export the unique ID.
-U	nsNoWrap	If set, long lines are not wrapped.
-x	nsExcludeSuffix	Sets the suffix to exclude in the exported file.

nsFilename

The **nsFilename** attribute contains the path and filenames of the LDIF files to which to export the Directory Server instance database.

Parameter	Description
Entry DN	<code>cn=task_name,cn=export,cn=tasks,cn=config</code>
Valid Values	Any string
Default Value	
Syntax	Case-exact string, multi-valued
Example	nsFilename: /home/jsmith/example.ldif

nsInstance

This attribute supplies the name of the database instance from which to export the database, such as **userRoot** or **userRoot**.

Parameter	Description
Entry DN	<code>cn=task_name,cn=export,cn=tasks,cn=config</code>
Valid Values	The name of a Directory Server instance (any string)
Default Value	
Syntax	Case-exact string, multi-valued
Example	nsInstance: userRoot

nsIncludeSuffix

This attribute identifies a specific suffix or subtree to export to an LDIF file.

Parameter	Description
Entry DN	<code>cn=task_name,cn=export,cn=tasks,cn=config</code>
Valid Values	Any DN
Default Value	
Syntax	DN, multi-valued
Example	<code>nsIncludeSuffix: ou=people,dc=example,dc=com</code>

nsExcludeSuffix

This attribute identifies suffixes or subtrees in the database to exclude from the exported LDIF file.

Parameter	Description
Entry DN	<code>cn=task_name,cn=export,cn=tasks,cn=config</code>
Valid Values	Any DN
Default Value	
Syntax	DN, multi-valued
Example	<code>nsExcludeSuffix: ou=machines,dc=example,dc=com</code>

nsUseOneFile

This attribute sets whether to export all Directory Server instances to a single LDIF file or separate LDIF files.

Parameter	Description
Entry DN	<code>cn=task_name,cn=export,cn=tasks,cn=config</code>
Valid Values	true false
Default Value	true
Syntax	Case-insensitive string
Example	<code>nsUseOneFile: true</code>

nsExportReplica

This attribute identifies whether the exported database will be used in replication. For replicas, the proper attributes and settings will be included with the entry to initialize the replica automatically.

Parameter	Description
Entry DN	<code>cn=task_name,cn=export,cn=tasks,cn=config</code>
Valid Values	true false
Default Value	false
Syntax	Case-insensitive string
Example	<code>nsExportReplica: true</code>

nsPrintKey

This attribute sets whether to print the entry ID number as the entry is processed by the export task.

Parameter	Description
Entry DN	<code>cn=task_name,cn=export,cn=tasks,cn=config</code>
Valid Values	true false
Default Value	true
Syntax	Case-insensitive string
Example	<code>nsPrintKey: false</code>

nsUseld2Entry

The **nsUseld2Entry** attribute uses the main database index, **id2entry**, to define the exported LDIF entries.

Parameter	Description
Entry DN	<code>cn=task_name,cn=export,cn=tasks,cn=config</code>
Valid Values	true false
Default Value	false
Syntax	Case-insensitive string
Example	<code>nsUseld2Entry: true</code>

nsNoWrap

This attribute sets whether to wrap long lines in the LDIF file.

Parameter	Description
Entry DN	<code>cn=task_name,cn=export,cn=tasks,cn=config</code>
Valid Values	true false
Default Value	false
Syntax	Case-insensitive string
Example	<code>nsNoWrap: false</code>

nsDumpUniqId

This attribute sets that the unique IDs for the exported entries are not exported.

Parameter	Description
Entry DN	<code>cn=task_name,cn=export,cn=tasks,cn=config</code>
Valid Values	true false
Default Value	true
Syntax	Case-insensitive string
Example	<code>nsDumpUniqId: true</code>

3.1.16.4. cn=backup

A database can be backed up through the command line by creating a special task entry which defines the parameters of the task and initiates the task. As soon as the task is complete, the task entry is removed from the directory.

The **cn=backup** entry is a container entry for backup task operations. The **cn=backup** entry itself has no attributes, but each of the task entries within this entry, such as **cn=task_ID**, **cn=backup**, **cn=tasks**, **cn=config**, uses the following attributes to define the backup task.

A backup task entry under **cn=backup** must contain the location of the directory to which to copy the archive copy (in the **nsArchiveDir** attribute) and the type of database being backed up (in the **nsDatabaseType** attribute). Additionally, it must contain a unique **cn** to identify the task. For example:

```
dn: cn=example backup,cn=backup,cn=tasks,cn=config
objectclass: extensibleObject
cn: example backup
```

```
nsArchiveDir: /export/backups/
nsDatabaseType: ldbm database
```

As the backup operation runs, the task entry will contain all of the server-generated task attributes listed in [Section 3.1.16.1, “Task Invocation Attributes for Entries under cn=tasks”](#).

nsArchiveDir

This attribute gives the location of the directory to which to write the backup.

The backup directory here should usually be the same as the one configured in the **nsslapd-bakdir** attribute.

If this attribute is not included with the **cn=backup** task, the task will fail with an LDAP object class violation error (65).

Parameter	Description
Entry DN	<code>cn=task_name,cn=backup,cn=tasks,cn=config</code>
Valid Values	Any local directory location
Default Value	
Syntax	Case-exact string
Example	<code>nsArchiveDir: /export/backups</code>

nsDatabaseType

This attribute gives the kind of database being archived. Setting the database types signals what kind of backup plug-in the Directory Server should use to archive the database.

Parameter	Description
Entry DN	<code>cn=task_name,cn=backup,cn=tasks,cn=config</code>
Valid Values	<code>ldbm database</code>
Default Value	<code>ldbm database</code>
Syntax	Case-exact string
Example	<code>nsDatabaseType: ldbm database</code>

3.1.16.5. cn=restore

A database can be restored through the command line by creating a special task entry which defines the parameters of the task and initiates the task. As soon as the task is complete, the task entry is removed from the directory.

The **cn=restore** entry is a container entry for task operations to restore a database. The **cn=restore** entry itself has no attributes, but each of the task entries within this entry, such as **cn=task_ID**, **cn=restore**, **cn=tasks**, **cn=config**, uses the following attributes to define the restore task.

A restore task entry under **cn=restore** must contain the location of the directory from which to retrieve the archive copy (in the **nsArchiveDir** attribute) and the type of database being restored (in the **nsDatabaseType** attribute). Additionally, it must contain a unique **cn** to identify the task. For example:

```
dn: cn=example restore,cn=restore,cn=tasks,cn=config
objectclass: extensibleObject
cn: example restore
nsArchiveDir: /export/backups/
nsDatabaseType: ldbm database
```

As the restore operation runs, the task entry will contain all of the server-generated task attributes listed in [Section 3.1.16.1, “Task Invocation Attributes for Entries under cn=tasks”](#).

nsArchiveDir

This attribute gives the location of the directory to which to write the backup.

Parameter	Description
Entry DN	<code>cn=task_name,cn=restore,cn=tasks,cn=config</code>
Valid Values	Any local directory location
Default Value	
Syntax	Case-exact string
Example	<code>nsArchiveDir: /export/backups</code>

nsDatabaseType

This attribute gives the kind of database being archived. Setting the database types signals what kind of backup plug-in the Directory Server should use to archive the database.

Parameter	Description
Entry DN	<code>cn=task_name,cn=restore,cn=tasks,cn=config</code>
Valid Values	<code>ldbm database</code>
Default Value	<code>ldbm database</code>
Syntax	Case-exact string
Example	<code>nsDatabaseType: ldbm database</code>

3.1.16.6. cn=index

Directory attributes can be indexed through the command line by creating a special task entry which defines the parameters of the task and initiates the task. As soon as the task is complete, the task entry is removed from the directory.

The **cn=index** entry is a container entry for index task operations. The **cn=index** entry itself has no attributes, but each of the task entries within this entry, such as **cn=task_ID**, **cn=index**, **cn=tasks**, **cn=config**, uses the following attributes to define the backup task.

An index task entry under **cn=index** can create a standard index by identifying the attribute to be indexed and the type of index to create, both defined in the **nsIndexAttribute** attribute.

Alternatively, the index task can be used to generate virtual list view (VLV) indexes for an attribute using the **nsIndexVLVAttribute** attribute. This is the same as running the **vlvindex** script.

For example:

```
dn: cn=example presence index,cn=index,cn=tasks,cn=config
objectclass: top
objectclass: extensibleObject
cn: example presence index
nsInstance: userRoot
nsIndexAttribute: cn:pres

dn: cn=example VLV index,cn=index,cn=tasks,cn=config
objectclass: extensibleObject
cn: example VLV index
nsIndexVLVAttribute: "by MCC ou=people,dc=example,dc=com"
```

As the index operation runs, the task entry will contain all of the server-generated task attributes listed in [Section 3.1.16.1, “Task Invocation Attributes for Entries under cn=tasks”](#).

nsIndexAttribute

This attribute gives the name of the attribute to index and the types of indexes to apply. The format of the attribute value is the attribute name and a comma-separated list of index types, enclosed in double quotation marks. For example:

```
nsIndexAttribute: attribute:index1,index2
```

Parameter	Description
Entry DN	<code>cn=task_name,cn=index,cn=tasks,cn=config</code>
Valid Values	* Any attribute * The index type, which can be pres (presence), eq (equality), approx (approximate), and sub (substring)
Default Value	
Syntax	Case-insensitive string, multi-valued

Parameter	Description
Example	* nsIndexAttribute: cn:pres,eq * nsIndexAttribute: description:sub

nsIndexVLVAttribute

This attribute gives the name of the target entry for a VLV index. A virtual list view is based on a browsing index entry (as described in the *Administration Guide*), which defines the virtual list base DN, scope, and filter. The **nsIndexVLVAttribute** value is the browsing index entry, and the VLV creation task is run according to the browsing index entry parameters.

Parameter	Description
Entry DN	cn=task_name,cn=index,cn=tasks,cn=config
Valid Values	RDN of the subentry of the VLV entry definition
Default Value	
Syntax	DirectoryString
Example	nsIndexVLVAttribute: "browsing index sort identifier"

3.1.16.7. cn=schema reload task

The directory schema is loaded when the directory instance is started or restarted. Any changes to the directory schema, including adding custom schema elements, are not loaded automatically and available to the instance until the server is restarted or by initiating a schema reload task.

Custom schema changes can be reloaded dynamically, without having to restart the Directory Server instance. This is done by initiating a schema reload task through creating a new task entry under the **cn=tasks** entry.

The custom schema file can be located in any directory; if not specified with the **schemadir** attribute, the server reloads the schema from the default **/etc/dirsrv/slappd-instance/schema** directory.



IMPORTANT

Any schema loaded from another directory must be copied into the schema directory or the schema will be lost when the server.

The schemad reload task is initiated though the command line by creating a special task entry which defines the parameters of the task and initiates the task. As soon as the task is complete, the task entry is removed from the directory. For example:

```
dn: cn=example schema reload,cn=schema reload task,cn=tasks,cn=config
objectclass: extensibleObject
```

```
cn:example schema reload
schemadir: /export/schema
```

The **cn=schema reload task** entry is a container entry for schema reload operations. The **cn=schema reload task** entry itself has no attributes, but each of the task entries within this entry, such as **cn=task_ID**, **cn=schema reload task**, **cn=tasks**, **cn=config**, uses the schema reload attributes to define the individual reload task.

cn

The **cn** attribute identifies a new task operation to initiate. The **cn** attribute value can be anything, as long as it defines a new task.

Parameter	Description
Entry DN	<code>cn=task_name,cn=schema reload task,cn=tasks,cn=config</code>
Valid Values	Any string
Default Value	
Syntax	DirectoryString
Example	cn: example reload task ID

schemadir

This contains the full path to the directory containing the custom schema file.

Parameter	Description
Entry DN	<code>cn=task_name,cn=schema reload task,cn=tasks,cn=config</code>
Valid Values	Any local directory path
Default Value	<code>/etc/dirsrv/schema</code>
Syntax	DirectoryString
Example	<code>schemadir: /export/schema/</code>

3.1.16.8. cn=memberof task

The **memberOf** attribute is created and managed by the Directory Server automatically to display group membership on the members' user entries. When the **member** attribute on a group entry is changed, all of the members' associated directory entries are automatically updated with their corresponding **memberOf** attributes.

The **cn=memberof task** (and the related **fixup-memberof.pl** script) is used to create the initial **memberOf** attributes on the member's user entries in the directory. After the **memberOf** attributes are created, then the MemberOf Plug-in manages the **memberOf** attributes automatically.

The **memberOf** update task must give the DN of the entry or subtree to run the update task against (set in the **basedn** attribute). Optionally, the task can include a filter to identify the members' user entries to update (set in the **filter** attribute). For example:

```
dn: cn=example memberOf,cn=memberof task,cn=tasks,cn=config
objectclass: extensibleObject
cn:example memberOf
basedn: ou=people,dc=example,dc=com
filter: (objectclass=groupofnames)
```

When the task is complete, the task entry is removed from the directory.

The **cn=memberof task** entry is a container entry for **memberOf** update operations. The **cn=memberof task** entry itself has no attributes, but each of the task entries beneath this entry, such as **cn=task_ID, cn=memberof task, cn=tasks, cn=config**, uses its attributes to define the individual update task.

basedn

This attribute gives the base DN to use to search for the user entries to update the **memberOf** attribute.

Parameter	Description
Entry DN	<code>cn=task_name,cn=memberof task,cn=tasks,cn=config</code>
Valid Values	Any DN
Default Value	
Syntax	DN
Example	<code>basedn: ou=people,dc=example,dc=com</code>

filter

This attribute gives an optional LDAP filter to use to select which user entries to update the **memberOf** attribute. Each member of a group has a corresponding user entry in the directory.

Parameter	Description
Entry DN	<code>cn=task_name,cn=memberof task,cn=tasks,cn=config</code>
Valid Values	Any LDAP filter
Default Value	<code>(objectclass=*)</code>
Syntax	DirectoryString

Parameter	Description
Example	filter: (l=Sunnyvale)

3.1.16.9. cn=fixup linked attributes

The Directory Server has a Linked Attributes Plug-in which allows one attribute, set in one entry, to update another attribute in another entry automatically. Both entries have DNs for values. The DN value in the first entry points to the entry for the plug-in to update; the attribute in the second entry contains a DN back-pointer to the first entry.

This is similar to the way that the MemberOf Plug-in uses the **member** attribute in group entries to set **memberOf** attribute in user entries. With linked attributes, any attribute can be defined as a "link," and then another attribute is "managed" in affected entries.

The **cn=fixup linked attributes** (and the related **fixup-linkedattrs.pl** script) creates the managed attributes – based on link attributes that already exist in the database – in the user entries once the linking plug-in instance is created. After the linked and managed attributes are set, the Linked Attributes Plug-in maintains the managed attributes dynamically, as users change the link attributes.

The linked attributes update task can specify which linked attribute plug-in instance to update, set in the optional **linkdn** attribute. If this attribute is not set on the task entry, then all configured linked attributes are updated.

```
dn: cn=example,cn=fixup linked attributes,cn=tasks,cn=config
objectclass: extensibleObject
cn:example
linkdn: cn=Example Link,cn=Linked Attributes,cn=plugins,cn=config
```

When the task is complete, the task entry is removed from the directory.

The **cn=fixup linked attributes** entry is a container entry for any linked attribute update operation. The **cn=fixup linked attributes** entry itself has no attributes related to individual tasks, but each of the task entries beneath this entry, such as **cn=task_ID**, **cn=fixup linked attributes**, **cn=tasks**, **cn=config**, uses its attributes to define the individual update task.

linkdn

Each linked-managed attribute pair is configured in a linked attributes plug-in instance. The **linkdn** attribute sets the specific linked attribute plug-in used to update the entries by giving the plug-in instance DN. For example:

```
linkdn: cn=Manager Attributes,cn=Linked Attributes,cn=plugins,cn=config
```

If no plug-in instance is given, then all linked attributes are updated.

Parameter	Description
Entry DN	<code>cn=task_name,cn=fixup linked attributes,cn=tasks,cn=config</code>

Parameter	Description
Valid Values	A DN (for an instance of the Linked Attributes plug-in)
Default Value	None
Syntax	DN
Example	linkdn: cn=Manager Links,cn=Linked Attributes,cn=plugins,cn=config

3.1.16.10. **cn=syntax validate**

Syntax validation checks every modification to attributes to make sure that the new value has the required syntax for that attribute type. Attribute syntaxes are validated against the definitions in [RFC 4514](#).

Syntax validation is enabled by default. However, syntax validation only audits *changes* to attribute values, such as when an attribute is added or modified. It does not validate the syntax of *existing* attribute values.

Validation of the existing syntax can be done with the syntax validation task. This task checks entries under a specified subtree (in the `basedn` attribute) and, optionally, only entries which match a specified filter (in the `filter` attribute).

```
dn: cn=example,cn=syntax validate,cn=tasks,cn=config
objectclass: extensibleObject
cn:example
basedn: ou=people,dc=example,dc=com
filter: "(objectclass=inetorgperson)"
```

When the task is complete, the task entry is removed from the directory.

If syntax validation is disabled or if a server is migrated, then there may be data in the server which does not conform to attribute syntax requirements. The syntax validation task can be run to evaluate those existing attribute values before enabling syntax validation.

The **cn=syntax validate** entry is a container entry for any syntax validation operation. The **cn=syntax validate** entry itself has no attributes that are specific to any task. Each of the task entries beneath this entry, such as **cn=task_ID**, **cn=syntax validate**, **cn=tasks**, **cn=config**, uses its attributes to define the individual update task.

basedn

Gives the subtree against which to run the syntax validation task. For example:

```
basedn: ou=people,dc=example,dc=com
```

Parameter	Description
Entry DN	<code>cn=task_name,cn=syntax validate,cn=tasks,cn=config</code>
Valid Values	Any DN
Default Value	None
Syntax	DN
Example	<code>basedn: dc=example,dc=com</code>

filter

Contains an optional LDAP filter which can be used to identify specific entries beneath the given **basedn** against which to run the syntax validation task. If this attribute is not set on the task, then every entry within the **basedn** is audited. For example:

```
filter: "(objectclass=person)"
```

Parameter	Description
Entry DN	<code>cn=task_name,cn=syntax validate,cn=tasks,cn=config</code>
Valid Values	Any LDAP filter
Default Value	<code>"(objectclass=*)"</code>
Syntax	DirectoryString
Example	<code>filter: "(objectclass=*)"</code>

3.1.16.11. cn=USN tombstone cleanup task

If the USN Plug-in is enabled, then *update sequence numbers* (USNs) are set on every entry whenever a directory write operation, like add or modify, occurs on that entry. This is reflected in the **entryUSN** operational attribute. This USN is set even when an entry is deleted, and the tombstone entries are maintained by the Directory Server instance.

The **cn=USN tombstone cleanup task** (and the related **usn-tombstone-cleanup.pl** script) deletes the tombstone entries maintained by the instance according to the back end database (in the **backend** attribute) or the suffix (in the **suffix** attribute). Optionally, only a subset of tombstone entries can be deleted by specifying a maximum USN to delete (in the **max_usn_to_delete** attribute), which preserves the most recent tombstone entries.

```
dn: cn=example,cn=USN tombstone cleanup task,cn=tasks,cn=config
objectclass: extensibleObject
cn:example
```

```
backend: userroot
max_usn_to_delete: 500
```



IMPORTANT

This task can only be launched if replication is *not* enabled. Replication maintains its own tombstone store, and these tombstone entries cannot be deleted by the USN Plug-in; they must be maintained by the replication processes. Thus, Directory Server prevents users from running the cleanup task for replicated databases.

Attempting to create this task entry for a replicated back end will return this error in the command line:

`ldap_add: DSA is unwilling to perform`

In the error log, there is a more explicit message that the suffix cannot have tombstone removed because it is replicated.

`[...] usn-plugin - Suffix dc=example,dc=com is replicated. Unwilling to perform cleaning up tombstones.`

When the task is complete, the task entry is removed from the directory.

The **cn=USN tombstone cleanup task** entry is a container entry for all USN tombstone delete operations. The **cn=USN tombstone cleanup task** entry itself has no attributes related to any individual task, but each of the task entries beneath this entry, such as **cn=task_ID**, **cn=USN tombstone cleanup task**, **cn=tasks**, **cn=config**, uses its attributes to define the individual update task.

backend

This gives the Directory Server instance back end, or database, to run the cleanup operation against. If the back end is not specified, then the suffix must be specified.

Parameter	Description
Entry DN	<code>cn=task_name,cn=USN tombstone cleanup task,cn=tasks,cn=config</code>
Valid Values	Database name
Default Value	None
Syntax	DirectoryString
Example	<code>backend: userroot</code>

max_usn_to_delete

This gives the highest USN value to delete when removing tombstone entries. All tombstone entries up to and including that number are deleted. Tombstone entries with higher USN values (that means more recent entries) are not deleted.

Parameter	Description
Entry DN	cn=task_name,cn=USN tombstone cleanup task,cn=tasks,cn=config
Valid Values	Any integer
Default Value	None
Syntax	Integer
Example	max_usn_to_delete: 500

suffix

This gives the suffix or subtree in the Directory Server to run the cleanup operation against. If the suffix is not specified, then the back end must be given.

Parameter	Description
Entry DN	cn=task_name,cn=USN tombstone cleanup task,cn=tasks,cn=config
Valid Values	Any subtree DN
Default Value	None
Syntax	DN
Example	suffix: dc=example,dc=com

3.1.16.12. cn=cleanallruv

Information about the replication topology – all of the suppliers which are supplying updates to each other and other replicas within the same replication group – is contained in a set of metadata called the *replica update vector (RUV)*. The RUV contains information about the supplier like its ID and URL, its latest change state number for changes made on the local server, and the CSN of the first change. Both suppliers and consumers store RUV information, and they use it to control replication updates.

When one supplier is removed from the replication topology, it may remain in another replica's RUV. When the other replica is restarted, it can record errors in its log that the replication plug-in does not recognize the (removed) supplier.

```
[09/Sep/2020:09:03:43 -0600] NSMMReplicationPlugin - ruv_compare_ruv: RUV [changelog max RUV] does not contain element [{replica 55 ldap://server.example.com:389} 4e6a27ca000000370000 4e6a27e8000000370000] which is present in RUV [database RUV]
```

.....

[09/Sep/2020:09:03:43 -0600] NSMMReplicationPlugin - replica_check_for_data_reload: Warning: for replica dc=example,dc=com there were some differences between the changelog max RUV and the database RUV. If there are obsolete elements in the database RUV, you should remove them using the CLEANRUV task. If they are not obsolete, you should check their status to see why there are no changes from those servers in the changelog.

When the supplier is permanently removed from the topology, then any lingering metadata about that supplier should be purged from every other supplier's RUV entry.

The **cn=cleanallruv** task propagates through all servers in the replication topology and removes the RUV entries associated with the specified missing or obsolete supplier.

When the task is complete, the task entry is removed from the directory.

The **cn=cleanallruv** entry is a container entry for all clean RUV operations. The **cn=cleanallruv** entry itself has no attributes related to any individual task, but each of the task entries beneath this entry, such as **cn=task_ID,cn=cleanallruv, cn=tasks, cn=config**, uses its attributes to define the individual update task.

Each clean RUV task must specify the replica ID number of the replica RUV entries to remove, the based DN of the replicated database, and whether remaining updates from the missing supplier should be applied before removing the RUV data.

```
dn: cn=clean 55,cn=cleanallruv,cn=tasks,cn=config
objectclass: extensibleObject
replica-base-dn: dc=example,dc=com
replica-id: 55
replica-force-cleaning: no
cn: clean 55
```

replica-base-dn

This gives the Directory Server base DN associated with the replicated database. This is the base DN for the replicated suffix.

Parameter	Description
Entry DN	<code>cn=task_name,cn=cleanallruv,cn=tasks,cn=config</code>
Valid Values	Directory suffix DN
Default Value	None
Syntax	DirectoryString
Example	<code>replica-base-dn: dc=example,dc=com</code>

replica-id

This gives the replica ID (defined in the **nsDS5ReplicaId** attribute for the replica configuration entry) of the replica *to be removed* from the replication topology.

Parameter	Description
Entry DN	<code>cn=task_name,cn=cleanallruv,cn=tasks,cn=config</code>
Valid Values	0 to 65534
Default Value	None
Syntax	Integer
Example	replica-id: 55

replica-force-cleaning

This sets whether any outstanding updates from the replica to be removed should be applied (**no**) or whether the clean RUV operation should force-continue and lose any remaining updates (**yes**).

Parameter	Description
Entry DN	<code>cn=task_name,cn=cleanallruv,cn=tasks,cn=config</code>
Valid Values	no yes
Default Value	None
Syntax	DirectoryString
Example	replica-force-cleaning: no

3.1.16.13. cn=abort cleanallruv

The [Section 3.1.16.12, “cn=cleanallruv”](#) task can take several minutes to propagate among all servers in the replication topology, even longer if the task processes all updates first. For performance or other maintenance considerations, it is possible to terminate a clean RUV task, and that termination is also propagated across all servers in the replication topology.

The termination task is an instance of the **cn=abort cleanallruv** entry.

When the task is complete, the task entry is removed from the directory.

The **cn=abort cleanallruv** entry is a container entry for all clean RUV operations. The **cn=abort cleanallruv** entry itself has no attributes related to any individual task, but each of the task entries beneath this entry, such as **cn=task_ID, cn=abort cleanallruv, cn=tasks, cn=config**, uses its attributes to define the individual update task.

Each clean RUV task must specify the replica ID number of the replica RUV entries to *which are currently being removed*, the based DN of the replicated database, and whether the terminate task should complete when it has completed on all servers in the topology or just locally.

```
dn: cn=abort 55,cn=abort cleanallruv,cn=tasks,cn=config
objectclass: extensibleObject
replica-base-dn: dc=example,dc=com
replica-id: 55
replica-certify-all: yes
cn: abort 55
```

replica-base-dn

This gives the Directory Server base DN associated with the replicated database. This is the base DN for the replicated suffix.

Parameter	Description
Entry DN	cn= <i>task_name</i> ,cn=abort cleanallruv,cn=tasks,cn=config
Valid Values	Directory suffix DN
Default Value	None
Syntax	DirectoryString
Example	replica-base-dn: dc=example,dc=com

replica-id

This gives the replica ID (defined in the **nsDS5ReplicaId** attribute for the replica configuration entry) of the replica *in the process of being removed* from the replication topology.

Parameter	Description
Entry DN	cn= <i>task_name</i> ,cn=abort cleanallruv,cn=tasks,cn=config
Valid Values	0 to 65534
Default Value	None
Syntax	Integer
Example	replica-id: 55

replica-certify-all

This sets whether the task should complete successfully on *all* servers in the replication topology before completing the task locally (**yes**) or whether the task should show complete as soon as it completes locally (**no**).

Parameter	Description
Entry DN	<code>cn=task_name,cn=abort cleanallruv,cn=tasks,cn=config</code>
Valid Values	no yes
Default Value	None
Syntax	DirectoryString
Example	<code>replica-certify-all: yes</code>

3.1.16.14. **cn=automember rebuild membership**

The Auto Member Plug-in only runs when new entries are added to the directory. The plug-in ignores existing entries or entries which are edited to match an automembership rule.

The **cn=automember rebuild membership** task runs the current automembership rules against *existing* entries to update or rebuild group membership. All configured automembership rules are run against the identified entries (though not all rules may apply to a given entry).

basedn

This gives the Directory Server base DN to use to search for user entries. The entries in the specified DN are then updated according to the automembership rules.

Parameter	Description
Entry DN	<code>cn=task_name,cn=automember rebuild membership,cn=tasks,cn=config</code>
Valid Values	Directory suffix DN
Default Value	None
Syntax	DirectoryString
Example	<code>basedn: dc=example,dc=com</code>

filter

This attribute gives an LDAP filter to use to identify which user entries to update according to the configured automembership rules.

Parameter	Description
Entry DN	<code>cn=task_name,cn=automember rebuild membership,cn=tasks,cn=config</code>
Valid Values	Any LDAP filter
Default Value	None
Syntax	DirectoryString
Example	<code>filter: (uid=*)</code>

scope

This attribute gives an LDAP search scope to use when searching the given base DN.

Parameter	Description
Entry DN	<code>cn=task_name,cn=automember rebuild membership,cn=tasks,cn=config</code>
Valid Values	sub base one
Default Value	None
Syntax	DirectoryString
Example	<code>scope: sub</code>

3.1.16.15. cn=automember export updates

This task runs against *existing entries* in the directory and exports the results of what users would have been added to what groups, based on the rules. This is useful for testing existing rules against existing users to see how your real deployment are performing.

The automembership-related changes are *not* executed. The proposed changes are written to a specified LDIF file.

basedn

This gives the Directory Server base DN to use to search for user entries. A test-run of the automembership rules will be run against the identified entries.

Parameter	Description
Entry DN	<code>cn=task_name,cn=automember export updates,cn=tasks,cn=config</code>

Parameter	Description
Valid Values	Directory suffix DN
Default Value	None
Syntax	DirectoryString
Example	basedn: dc=example,dc=com

filter

This attribute gives an LDAP filter to use to identify which user entries to test-run the automembership rules.

Parameter	Description
Entry DN	cn=task_name,cn=automember export updates,cn=tasks,cn=config
Valid Values	Any LDAP filter
Default Value	None
Syntax	DirectoryString
Example	filter: (uid=*)

scope

This attribute gives an LDAP search scope to use when searching the given base DN.

Parameter	Description
Entry DN	cn=task_name,cn=automember export updates,cn=tasks,cn=config
Valid Values	sub base one
Default Value	None
Syntax	DirectoryString
Example	scope: sub

ldif

This attribute sets the full path and filename of an LDIF file to which to write the proposed changes from the test-run of the automembership rules. This file must be local to the system from which the task is initiated.

Parameter	Description
Entry DN	<code>cn=task_name,cn=automember export updates,cn=tasks,cn=config</code>
Valid Values	Local path and filename
Default Value	None
Syntax	DirectoryString
Example	<code>ldif: /tmp/automember-results.ldif</code>

3.1.16.16. `cn=automember map updates`

This task runs against entries within an LDIF file (new entries or, potentially, test entries) and then writes the proposed changes to those user entries to an LDIF file. This can be very useful for testing a new rule, before applying it to (real) new or existing user entries.

The automembership-related changes are *not* executed. The proposed changes are written to a specified LDIF file.

`ldif_in`

This attribute sets the full path and filename of an LDIF file from which to import entries to test with the configured automembership rules. These entries are not imported into the directory and the changes are not performed. The entries are loaded and used by the test-run only.

This file must be local to the system from which the task is initiated.

Parameter	Description
Entry DN	<code>cn=task_name,cn=automember map updates,cn=tasks,cn=config</code>
Valid Values	Local path and filename
Default Value	None
Syntax	DirectoryString
Example	<code>ldif_in: /tmp/automember-test-users.ldif</code>

`ldif_out`

This attribute sets the full path and filename of an LDIF file to which to write the proposed changes from the test-run of the automembership rules. This file must be local to the system from which the task is initiated.

Parameter	Description
Entry DN	<code>cn=task_name,cn=automember map updates,cn=tasks,cn=config</code>
Valid Values	Local path and filename
Default Value	None
Syntax	DirectoryString
Example	<code>ldif_out:/tmp/automember-results.ldif</code>

3.1.16.17. `cn=des2aes`

This task searches for all reversible password entries in the specified user database which are encoded using the outdated **DES** cipher, and converts them to the more secure **AES** cipher.

Previously, this task was being performed automatically on all suffixes during Directory Server startup. However, since the search for DES passwords was typically unindexed, it could take a very long time to perform on suffixes containing large amounts of entries, which in turn caused Directory Server to time out and fail to start. For that reason, the search is now performed only on **cn=config**, but to convert passwords in any other database you must run this task manually.

`suffix`

This multivalued attribute specifies a suffix to check for DES passwords and convert them to AES. If this attribute is omitted then all the back ends/suffixes are checked.

Parameter	Description
Entry DN	<code>cn=task_name,cn=des2aes,cn=tasks,cn=config</code>
Valid Values	Directory suffix DN
Default Value	None
Syntax	DirectoryString
Example	<code>suffix: dc=example,dc=com</code>

3.1.17. `cn=uniqueid generator`

The unique ID generator configuration attributes are stored under **cn=uniqueid generator,cn=config**. The **cn=uniqueid generator** entry is an instance of the **extensibleObject** object class.

nsstate

This attribute saves the state of the unique ID generator across server restarts. This attribute is maintained by the server. Do not edit it.

Parameter	Description
Entry DN	cn=uniqueid generator,cn=config
Valid Values	
Default Value	
Syntax	DirectoryString
Example	nsstate: AbId0c3oMIDUntiLCyYNGgAAAAAAAAAA

3.1.18. Root DSE Configuration Parameters

3.1.18.1. nsslapd-return-default-opattr

Directory Server does not display the operational attributes in Root DSE searches. For example, if you are running the **ldapsearch** utility with the **-s base -b ""** parameters, only the user attributes are displayed. For clients expecting operational attributes in Root DSE search output, you can enable this behavior to provide backward compatibility:

1. Stop the Directory Server instance.
2. Edit the **/etc/dirsrv/slapd-*instance_name*/dse.ldif** file and add the following parameters to the **dn:** section:

```
nsslapd-return-default-opattr: supportedsaslmechanisms
nsslapd-return-default-opattr: nsBackendSuffix
nsslapd-return-default-opattr: subschemasubentry
nsslapd-return-default-opattr: supportedldapversion
nsslapd-return-default-opattr: supportedcontrol
nsslapd-return-default-opattr: ref
nsslapd-return-default-opattr: vendorname
nsslapd-return-default-opattr: vendorVersion
nsslapd-return-default-opattr: supportedextension
nsslapd-return-default-opattr: namingcontexts
```

3. Start the Directory Server instance.

Parameter	Description
Entry DN	Root DSE

Parameter	Description
Valid Values	supportedsaslmechanisms nsBackendSuffix subschemasubentry supportedldapversion supportedcontrol ref vendorname vendorVersion
Default Value	
Syntax	DirectoryString
Example	nsslapd-return-default-opattr: supportedsaslmechanisms

3.2. CONFIGURATION OBJECT CLASSES

Many configuration entries simply use the **extensibleObject** object class, but some require other object classes. These configuration object classes are listed here.

3.2.1. changeLogEntry (Object Class)

This object class is used for entries which store changes made to the Directory Server entries.

To configure Directory Server to maintain a changelog that is compatible with the changelog implemented in Directory Server 4.1x, enable the Retro Changelog Plug-in. Each entry in the changelog has the **changeLogEntry** object class.

This object class is defined in Changelog Internet Draft.

Superior Class

top

OID

2.16.840.1.113730.3.2.1

Table 3.8. Required Attributes

objectClass	Defines the object classes for the entry.
Section 3.1.3.3, "changeNumber"	Contains a number assigned arbitrarily to the changelog.
Section 3.1.3.4, "changeTime"	The time at which a change took place.
Section 3.1.3.5, "changeType"	The type of change performed on an entry.
Section 3.1.3.10, "targetDn"	The distinguished name of an entry added, modified or deleted on a supplier server.

Table 3.9. Allowed Attributes

Section 3.1.3.1, "changes"	Changes made to the Directory Server.
Section 3.1.3.6, "deleteOldRdn"	A flag that defines whether the old Relative Distinguished Name (RDN) of the entry should be kept as a distinguished attribute of the entry or should be deleted.
Section 3.1.3.8, "newRdn"	New RDN of an entry that is the target of a modRDN or modDN operation.
Section 3.1.3.9, "newSuperior"	Name of the entry that becomes the immediate superior of the existing entry when processing a modDN operation.

3.2.2. directoryServerFeature (Object Class)

This object class is used specifically for entries which identify a feature of the directory service. This object class is defined by Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.40

Table 3.10. Required Attributes

Attribute	Definition
objectClass	Gives the object classes assigned to the entry.

Table 3.11. Allowed Attributes

Attribute	Definition
cn	Specifies the common name of the entry.
multiLineDescription	Gives a text description of the entry.
oid	Specifies the OID of the feature.

3.2.3. nsBackendInstance (Object Class)

This object class is used for the Directory Server back end, or database, instance entry. This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.109

Table 3.12. Required Attributes

Attribute	Definition
objectClass	Defines the object classes for the entry.
cn	Gives the common name of the entry.

3.2.4. nsChangelog4Config (Object Class)

In order for Directory Server 11.3 to replicate between Directory Server 4.x servers, the Directory Server 11.3 instance must have a special changelog configured. This object class defines the configuration for the retro changelog.

This object class is defined for the Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.82

Table 3.13. Allowed Attributes

Attribute	Definition
cn (common Name)	Gives the common name of the entry.

3.2.5. nsDS5Replica (Object Class)

This object class is for entries which define a replica in database replication. Many of these attributes are set within the back end and cannot be modified.

Information on the attributes for this object class are listed with the core configuration attributes in chapter 2 of the *Directory Server Configuration, Command, and File Reference*.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.108

Table 3.14. Required Attributes

objectClass	Defines the object classes for the entry.
nsDS5Replicaid	Specifies the unique ID for suppliers in a replication environment.
nsDS5ReplicaRoot	Specifies the suffix DN at the root of a replicated area.

Table 3.15. Allowed Attributes

cn	Gives the name for the replica.
nsDS5Flags	Specifies information that has been previously set in flags.
nsDS5ReplicaAutoReferral	Sets whether the server will follow configured referrals for the Directory Server database.
nsDS5ReplicaBindDN	Specifies the DN to use when a supplier server binds to a consumer.
nsDS5ReplicaChangeCount	Gives the total number of entries in the changelog and whether they have been replicated.
nsDS5ReplicaLegacyConsumer	Specifies whether the replica is a legacy consumer.
nsDS5ReplicaName	Specifies the unique ID for the replica for internal operations.
nsDS5ReplicaPurgeDelay	Specifies the time in seconds before the changelog is purged.
nsDS5ReplicaReferral	Specifies the URLs for user-defined referrals.
nsDS5ReplicaReleaseTimeout	Specifies a timeout after which a supplier will release a replica, whether or not it has finished sending its updates.
nsDS5ReplicaTombstonePurgeInterval	Specifies the time interval in seconds between purge operation cycles.
nsDS5ReplicaType	Defines the type of replica, such as a read-only consumer.
nsDS5Task	Launches a replication task, such as dumping the database contents to LDIF; this is used internally by the Directory Server supplier.

nsState	Stores information on the clock so that proper change sequence numbers are generated.
---------	---

3.2.6. nsDS5ReplicationAgreement (Object Class)

Entries with the **nsDS5ReplicationAgreement** object class store the information set in a replication agreement. Information on the attributes for this object class are in chapter 2 of the *Directory Server Configuration, Command, and File Reference*.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.103

Table 3.16. Required Attributes

objectClass	Defines the object classes for the entry.
cn	Used for naming the replication agreement.

Table 3.17. Allowed Attributes

description	Contains a free text description of the replication agreement.
nsDS5BeginReplicaRefresh	Initializes a replica manually.
nsds5debugreplicatimeout	Gives an alternate timeout period to use when the replication is run with debug logging.
nsDS5ReplicaBindDN	Specifies the DN to use when a supplier server binds to a consumer.
nsDS5ReplicaBindMethod	Specifies the method (SSL or simple authentication) to use for binding.
nsDS5ReplicaBusyWaitTime	Specifies the amount of time in seconds a supplier should wait after a consumer sends back a busy response before making another attempt to acquire access.
nsDS5ReplicaChangesSentSinceStartup	The number of changes sent to this replica since the server started.

nsDS5ReplicaCredentials	Specifies the password for the bind DN.
nsDS5ReplicaHost	Specifies the host name for the consumer replica.
nsDS5ReplicaLastInitEnd	States when the initialization of the consumer replica ended.
nsDS5ReplicaLastInitStart	States when the initialization of the consumer replica started.
nsDS5ReplicaLastInitStatus	The status for the initialization of the consumer.
nsDS5ReplicaLastUpdateEnd	States when the most recent replication schedule update ended.
nsDS5ReplicaLastUpdateStart	States when the most recent replication schedule update started.
nsDS5ReplicaLastUpdateStatus	Provides the status for the most recent replication schedule updates.
nsDS5ReplicaPort	Specifies the port number for the remote replica.
nsDS5ReplicaRoot	Specifies the suffix DN at the root of a replicated area.
nsDS5ReplicaSessionPauseTime	Specifies the amount of time in seconds a supplier should wait between update sessions.
nsDS5ReplicatedAttributeList	Specifies any attributes that will not be replicated to a consumer server.
nsDS5ReplicaTimeout	Specifies the number of seconds outbound LDAP operations will wait for a response from the remote replica before timing out and failing.
nsDS5ReplicaTransportInfo	Specifies the type of transport used for transporting data to and from the replica.
nsDS5ReplicaUpdateInProgress	States whether a replication schedule update is in progress.
nsDS5ReplicaUpdateSchedule	Specifies the replication schedule.
nsDS50ruv	Manages the internal state of the replica using the replication update vector.
nsruvReplicaLastModified	Contains the most recent time that an entry in the replica was modified and the changelog was updated.

nsds5ReplicaStripAttrs	With fractional replication, an update to an excluded attribute still triggers a replication event, but that event is empty. This attribute sets attributes to strip from the replication update. This prevents changes to attributes like internalModifyTimestamp from triggering an empty replication update.
------------------------	--

3.2.7. nsDSWindowsReplicationAgreement (Object Class)

Stores the synchronization attributes that concern the synchronization agreement. Information on the attributes for this object class are in chapter 2 of the *Red Hat Directory Server Configuration, Command, and File Reference*.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.503

Table 3.18. Required Attributes

objectClass	Defines the object classes for the entry.
cn	Gives the name of the synchronization agreement.

Table 3.19. Allowed Attributes

description	Contains a text description of the synchronization agreement.
nsDS5BeginReplicaRefresh	Initiates a manual synchronization.
nsds5debugreplicatimeout	Gives an alternate timeout period to use when the synchronization is run with debug logging.
nsDS5ReplicaBindDN	Specifies the DN to use when the Directory Server binds to the Windows server.
nsDS5ReplicaBindMethod	Specifies the method (SSL or simple authentication) to use for binding.
nsDS5ReplicaBusyWaitTime	Specifies the amount of time in seconds the Directory Server should wait after the Windows server sends back a busy response before making another attempt to acquire access.

nsDS5ReplicaChangesSentSinceStartup	Shows the number of changes sent since the Directory Server started.
nsDS5ReplicaCredentials	Specifies the credentials for the bind DN.
nsDS5ReplicaHost	Specifies the host name for the Windows domain controller of the Windows server being synchronized.
nsDS5ReplicaLastInitEnd	States when the last total update (resynchronization) of the Windows server ended.
nsDS5ReplicaLastInitStart	States when the last total update (resynchronization) of the Windows server started.
nsDS5ReplicaLastInitStatus	The status for the total update (resynchronization) of the Windows server.
nsDS5ReplicaLastUpdateEnd	States when the most recent update ended.
nsDS5ReplicaLastUpdateStart	States when the most recent update started.
nsDS5ReplicaLastUpdateStatus	Provides the status for the most recent updates.
nsDS5ReplicaPort	Specifies the port number for the Windows server.
nsDS5ReplicaRoot	Specifies the root suffix DN of the Directory Server.
nsDS5ReplicaSessionPauseTime	Specifies the amount of time in seconds the Directory Server should wait between update sessions.
nsDS5ReplicaTimeout	Specifies the number of seconds outbound LDAP operations will wait for a response from the Windows server before timing out and failing.
nsDS5ReplicaTransportInfo	Specifies the type of transport used for transporting data to and from the Windows server.
nsDS5ReplicaUpdateInProgress	States whether an update is in progress.
nsDS5ReplicaUpdateSchedule	Specifies the synchronization schedule.
nsDS50rv	Manages the internal state of the Directory Server sync peer using the replication update vector (RUV).
nsds7DirectoryReplicaSubtree	Specifies the Directory Server suffix (root or sub) that is synced.

nsds7DirsyncCookie	Contains a cookie set by the sync service that functions as an RUV.
nsds7NewWinGroupSyncEnabled	Specifies whether new Windows group accounts are automatically created on the Directory Server.
nsds7NewWinUserSyncEnabled	Specifies whether new Windows user accounts are automatically created on the Directory Server.
nsds7WindowsDomain	Identifies the Windows domain being synchronized; analogous to nsDS5ReplicaHost in a replication agreement.
nsds7WindowsReplicaSubtree	Specifies the Windows server suffix (root or sub) that is synced.
nsruvReplicaLastModified	Contains the most recent time that an entry in the Directory Server sync peer was modified and the changelog was updated.
winSyncInterval	Sets how frequently, in seconds, the Directory Server polls the Windows server for updates to write over. If this is not set, the default is 300 , which is 300 seconds or five (5) minutes.
winSyncMoveAction	Sets how the sync plug-in handles corresponding entries that are discovered in Active Directory outside of the synced subtree. The sync process can ignore these entries (none, the default) or it can assume that the entries were moved intentionally to remove them from synchronization, and it can then either delete the corresponding Directory Server entry (delete) or remove the synchronization attributes and no longer sync the entry (unsync).

3.2.8. nsEncryptionConfig

The **nsEncryptionConfig** object class stores the configuration information for allowed encryption options, such as protocols and cipher suites. This is defined in the Administrative Services.

Superior Class

top

OID

nsEncryptionConfig-oid

Table 3.20. Required Attributes

Attribute	Definition
objectClass	Defines the object classes for the entry.
cn (commonName)	Gives the common name of the device.

Table 3.21. Allowed Attributes

Attribute	Definition
nsSSL3SessionTimeout	Sets the timeout period for an SSLv3 cipher session.
nsSSLClientAuth	Sets how the server handles client authentication. There are three possible values: allow, disallow, or require.
nsSSLSessionTimeout	Sets the timeout period for a cipher session.
nsSSLSupportedCiphers	Contains a list of all ciphers available to be used with secure connections to the server.
nsTLS1	Sets whether TLS version 1 is enabled for the server.

3.2.9. nsEncryptionModule

The **nsEncryptionModule** object class stores the encryption module information. This is defined in the Administrative Services.

Superior Class

top

OID

nsEncryptionModule-oid

Table 3.22. Required Attributes

Attribute	Definition
objectClass	Defines the object classes for the entry.
cn (commonName)	Gives the common name of the device.

Table 3.23. Allowed Attributes

Attribute	Definition
nsSSLActivation	Sets whether to enable a cipher family.

Attribute	Definition
nsSSLPersonalitySSL	Contains the name of the certificate used by the server for SSL.
nsSSLToken	Identifies the security token used by the server.

3.2.10. nsMappingTree (Object Class)

A mapping tree maps a suffix to the back end. Each mapping tree entry uses the **nsMappingTree** object class. This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.110

Table 3.24. Required Attributes

Attribute	Definition
objectClass	Gives the object classes assigned to the entry.
cn	Gives the common name of the entry.

3.2.11. nsSaslMapping (Object Class)

This object class is used for entries which contain an identity mapping configuration for mapping SASL attributes to the Directory Server attributes.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.317

Table 3.25. Required Attributes

objectClass	Defines the object classes for the entry.
cn	Gives the name of the SASL mapping entry.
Section 3.1.13.1, "nsSaslMapBaseDNTemplate"	Contains the search base DN template.
Section 3.1.13.2, "nsSaslMapFilterTemplate"	Contains the search filter template.

Section 3.1.13.4, "nsSaslMapRegexString"	Contains a regular expression to match SASL identity strings.
--	---

3.2.12. nsslapdConfig (Object Class)

The **nsslapdConfig** object class defines the configuration object, **cn=config**, for the Directory Server instance.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.39

Table 3.26. Required Attributes

Attribute	Definition
objectClass	Gives the object classes assigned to the entry.

Table 3.27. Allowed Attributes

Attribute	Definition
cn	Gives the common name of the entry.

3.2.13. passwordPolicy (Object Class)

Both local and global password policies take the **passwordPolicy** object class. This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.13

Table 3.28. Required Attributes

Attribute	Definition
objectClass	Gives the object classes assigned to the entry.

Table 3.29. Allowed Attributes

Attribute	Definition
Section 3.1.1.190, "passwordMaxAge (Password Maximum Age)"	Sets the number of seconds after which user passwords expire.
Section 3.1.1.180, "passwordExp (Password Expiration)"	Identifies whether the user's password expires after an interval given by the passwordMaxAge attribute.
Section 3.1.1.202, "passwordMinLength (Password Minimum Length)"	Sets the minimum number of characters that must be used in passwords.
Section 3.1.1.185, "passwordInHistory (Number of Passwords to Remember)"	Sets the number of passwords the directory stores in the history.
Section 3.1.1.177, "passwordChange (Password Change)"	Identifies whether or not users is allowed to change their own password.
Section 3.1.1.218, "passwordWarning (Send Warning)"	Sets the number of seconds before a warning message is sent to users whose password is about to expire.
Section 3.1.1.188, "passwordLockout (Account Lockout)"	Identifies whether or not users are locked out of the directory after a given number of failed bind attempts.
Section 3.1.1.193, "passwordMaxFailure (Maximum Password Failures)"	Sets the number of failed bind attempts after which a user will be locked out of the directory.
Section 3.1.1.217, "passwordUnlock (Unlock Account)"	Identifies whether a user is locked out until the password is reset by an administrator or whether the user can log in again after a given lockout duration. The default is to allow a user to log back in after the lockout period.
Section 3.1.1.189, "passwordLockoutDuration (Lockout Duration)"	Sets the time, in seconds, that users will be locked out of the directory.
Section 3.1.1.178, "passwordCheckSyntax (Check Password Syntax)"	Identifies whether the password syntax is checked by the server before the password is saved.
Section 3.1.1.207, "passwordMustChange (Password Must Change)"	Identifies whether or not to change their passwords when they first login to the directory or after the password is reset by the Directory Manager.
Section 3.1.1.212, "passwordStorageScheme (Password Storage Scheme)"	Sets the type of encryption used to store Directory Server passwords.
Section 3.1.1.198, "passwordMinAge (Password Minimum Age)"	Sets the number of seconds that must pass before a user can change their password.

Attribute	Definition
Section 3.1.1.209, "passwordResetFailureCount (Reset Password Failure Count After)"	Sets the time, in seconds, after which the password failure counter will be reset. Each time an invalid password is sent from the user's account, the password failure counter is incremented.
Section 3.1.1.183, "passwordGraceLimit (Password Expiration)"	Sets the number of grace logins permitted when a user's password is expired.
Section 3.1.1.201, "PasswordMinDigits (Password Syntax)"	Sets the minimum number of numeric characters (0 through 9) which must be used in the password.
Section 3.1.1.199, "passwordMinAlphas (Password Syntax)"	Sets the minimum number of alphabetic characters that must be used in the password.
Section 3.1.1.206, "PasswordMinUppers (Password Syntax)"	Sets the minimum number of upper case alphabetic characters, A to Z, which must be used in the password.
Section 3.1.1.203, "PasswordMinLowers (Password Syntax)"	Sets the minimum number of lower case alphabetic characters, a to z, which must be used in the password.
Section 3.1.1.204, "PasswordMinSpecials (Password Syntax)"	Sets the minimum number of special ASCII characters, such as !@#\$., which must be used in the password.
Section 3.1.1.197, "passwordMin8Bit (Password Syntax)"	Sets the minimum number of 8-bit characters used in the password.
Section 3.1.1.194, "passwordMaxRepeats (Password Syntax)"	Sets the maximum number of times that the same character can be used in row.
Section 3.1.1.200, "passwordMinCategories (Password Syntax)"	Sets the minimum number of categories which must be used in the password.
Section 3.1.1.205, "PasswordMinTokenLength (Password Syntax)"	Sets the length to check for trivial words.
Section 3.1.1.214, "passwordTPRDelayValidFrom"	Sets a delay when temporary passwords become valid.
Section 3.1.1.213, "passwordTPRDelayExpireAt"	Sets the number of seconds a temporary password is valid.

Attribute	Definition
Section 3.1.1.215, "passwordTPRMaxUse" Sets the maximum number off attempts a temporary password can be used	

3.3. ROOT DSE ATTRIBUTES

The attributes in this section are used to define the root directory server entry (DSE) for the server instance. The information defined in the DSE relates to the actual configuration of the server instance, such as the controls, mechanisms, or features supported in that version of the server software. It also contains information specific to the instance, like its build number and installation date.

The DSE is a special entry, outside the normal DIT, and can be returned by searching with a null search base. For example:

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x -s base -b ""
"objectclass=*" 
```

3.3.1. dataversion

This attribute contains a timestamp which shows the most recent edit time for any data in the directory.

```
dataversion: 020090923175302020090923175302 
```

OID	
Syntax	GeneralizedTime
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

3.3.2. defaultNamingContext

Corresponds to the naming context, out of all configured naming contexts, which clients should use by default.

OID	
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

3.3.3. lastusn

The USN Plug-in assigns a sequence number to every entry whenever a write operation – add, modify, delete, and modrdn – is performed for that entry. The USN is assigned in the **entryUSN** operational attribute for the entry.

The USN Plug-in has two modes: local and global.

In local mode, each database maintained for a server instance has its own instance of the USN Plug-in with a separate USN counter per back end database. The most recent USN assigned for any entry in the database is displayed in the **lastusn** attribute. When the USN Plug-in is set to local mode, the **lastUSN** attribute shows both the database which assigned the USN and the USN:

lastusn;database_name:USN

For example:

lastusn;example1: 213
lastusn;example2: 207

In global mode, when the database uses a shared USN counter, the **lastUSN** value shows the latest USN assigned by any database:

lastusn: 420



NOTE

This attribute does not count internal server operations. Only normal write operations in the back end database – add, modify, delete, and modrdn – increment the USN count.

Syntax	Integer
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

3.3.4. namingContexts

Corresponds to a naming context the server is controlling or shadowing. When the Directory Server does not control any information (such as when it is an LDAP gateway to a public X.500 directory), this attribute is absent. When the Directory Server believes it contains the entire directory, the attribute has a single value, and that value is the empty string (indicating the null DN of the root). This attribute permits a client contacting a server to choose suitable base objects for searching.

OID	1.3.6.1.4.1.1466.101.120.5
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

3.3.5. netscapemdsuffix

This attribute contains the DN for the top suffix of the directory tree for machine data maintained in the server. The DN itself points to an LDAP URL. For example:

```
cn=ldap://dc=server_name,dc=example,dc=com:389
```

OID	2.16.840.1.113730.3.1.212
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

3.3.6. supportedControl

The values of this attribute are the object identifiers (OIDs) that identify the controls supported by the server. When the server does not support controls, this attribute is absent.

OID	1.3.6.1.4.1.1466.101.120.13
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

3.3.7. supportedExtension

The values of this attribute are the object identifiers (OIDs) that identify the extended operations supported by the server. When the server does not support extended operations, this attribute is absent.

OID	1.3.6.1.4.1.1466.101.120.7
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

3.3.8. supportedFeatures

This attribute contains features supported by the current version of Red Hat Directory Server.

OID	1.3.6.1.4.1.4203.1.3.5
-----	------------------------

Syntax	OID
Multi- or Single-Valued	Multi-valued
Defined in	RFC 3674

3.3.9. supportedLDAPVersion

This attribute identifies the versions of the LDAP protocol implemented by the server.

OID	1.3.6.1.4.1.1466.101.120.15
Syntax	Integer
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

3.3.10. supportedSASLMechanisms

This attribute identifies the names of the SASL mechanisms supported by the server. When the server does not support SASL attributes, this attribute is absent.

OID	1.3.6.1.4.1.1466.101.120.14
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

3.3.11. vendorName

This attribute contains the name of the server vendor.

OID	1.3.6.1.1.4
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	RFC 3045

3.3.12. vendorVersion

This attribute shows the vendor's version number for the server.

OID	1.3.6.1.1.5
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	RFC 3045

3.4. LEGACY ATTRIBUTES

The attributes were standard with Directory Server 4.x and older. This are still included with the schema for compatibility, but are not for current versions of the Directory Server.

3.4.1. Legacy Server Attributes

These attributes were originally used to configure the server instance entries for Directory Server 4.x and older servers.

3.4.1.1. LDAPServer (Object Class)

This object class identifies the LDAP server information. It is defined by Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.35

Table 3.30. Required Attributes

Attribute	Definition
objectClass	Gives the object classes assigned to the entry.
cn	Specifies the common name of the entry.

Table 3.31. Allowed Attributes

Attribute	Definition
description	Gives a text description of the entry.
l (localityName)	Gives the city or geographical location of the entry.
ou (organizationalUnitName)	Gives the organizational unit or division to which the account belongs.

Attribute	Definition
seeAlso	Contains a URL to another entry or site with related information.
generation	Store the server generation string.
changeLogMaximumAge	Specifies changelog maximum age.
changeLogMaximumSize	Specifies maximum changelog size.

3.4.1.2. changeLogMaximumAge

This sets the maximum age for the changelog maintained by the server.

OID	2.16.840.1.113730.3.1.200
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

3.4.1.3. changeLogMaximumConcurrentWrites

This attribute sets the maximum number of concurrent writes that can be written to the changelog.

OID	2.16.840.1.113730.3.1.205
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

3.4.1.4. changeLogMaximumSize

This attribute sets the maximum size for the changelog.

OID	2.16.840.1.113730.3.1.201
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued

Defined in	Directory Server
------------	------------------

3.4.1.5. generation

This attribute contains a byte vector that uniquely identifies that specific server and version. This number distinguishes between servers during replication.

OID	2.16.840.1.113730.3.1.612
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

3.4.1.6. nsSynchUniqueAttribute

This attribute is used for Windows synchronization.

OID	2.16.840.1.113730.3.1.407
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

3.4.1.7. nsSynchUserIDFormat

This attribute is used for Windows synchronization.

OID	2.16.840.1.113730.3.1.406
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

CHAPTER 4. PLUG-IN IMPLEMENTED SERVER FUNCTIONALITY REFERENCE

This chapter contains reference information on Red Hat Directory Server plug-ins.

The configuration for each part of Directory Server plug-in functionality has its own separate entry and set of attributes under the subtree **cn=plugins,cn=config**.

```
dn: cn=Telephone Syntax,cn=plugins,cn=config
objectclass: top
objectclass: nsSlapdPlugin
objectclass: extensibleObject
cn: Telephone Syntax
nsslapd-pluginPath: libsyntax-plugin
nsslapd-pluginInitfunc: tel_init
nsslapd-pluginType: syntax
nsslapd-pluginEnabled: on
```

Some of these attributes are common to all plug-ins while others may be particular to a specific plug-in. Check which attributes are currently being used by a given plug-in by performing an **ldapsearch** on the **cn=config** subtree.

All plug-ins are instances of the **nsSlapdPlugin** object class, which in turn inherits from the **extensibleObject** object class. For plug-in configuration attributes to be taken into account by the server, both of these object classes (in addition to the **top** object class) must be present in the entry, as shown in the following example:

```
dn:cn=ACL Plugin,cn=plugins,cn=config
objectclass:top
objectclass:nsSlapdPlugin
objectclass:extensibleObject
```

4.1. SERVER PLUG-IN FUNCTIONALITY REFERENCE

The following tables provide a quick overview of the plug-ins provided with Directory Server, along with their configurable options, configurable arguments, default setting, dependencies, general performance-related information, and further reading. These tables assist in weighing plug-in performance gains and costs and choose the optimal settings for the deployment. The *Further Information* section cross-references further reading, where this is available.

4.1.1. 7-bit Check Plug-in

Plug-in Parameter	Description
Plug-in ID	NS7bitAtt
DN of Configuration Entry	cn=7-bit check,cn=plugins,cn=config
Description	Checks certain attributes are 7-bit clean
Type	preoperation

Plug-in Parameter	Description
Configurable Options	on
off	Default Setting
on	Configurable Arguments
List of attributes (uid mail userpassword) followed by "," and then suffixes on which the check is to occur.	Dependencies
Database	Performance-Related Information
None	Further Information

4.1.2. ACL Plug-in

Plug-in Parameter	Description
Plug-in ID	acl
DN of Configuration Entry	cn=ACL Plugin,cn=plugins,cn=config
Description	ACL access check plug-in
Type	accesscontrol
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
Database	Performance-Related Information
Access control incurs a minimal performance hit. Leave this plug-in enabled since it is the primary means of access control for the server.	Further Information

4.1.3. ACL Preoperation Plug-in

Plug-in Parameter	Description
Plug-in ID	acl
DN of Configuration Entry	cn=ACL preoperation,cn=plugins,cn=config
Description	ACL access check plug-in
Type	preoperation
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
Database	Performance-Related Information
Access control incurs a minimal performance hit. Leave this plug-in enabled since it is the primary means of access control for the server.	Further Information

4.1.4. Account Policy Plug-in

Plug-in Parameter	Description
Plug-in ID	none
DN of Configuration Entry	cn=Account Policy Plugin,cn=plugins,cn=config
Description	Defines a policy to lock user accounts after a certain expiration period or inactivity period.
Type	object
Configurable Options	on
off	Default Setting
off	Configurable Arguments
A pointer to a configuration entry which contains the global account policy settings.	Dependencies

Plug-in Parameter	Description
Database	Performance-Related Information
None	Further Information

4.1.5. Account Usability Plug-in

Plug-in Parameter	Description
Plug-in ID	acctusability
DN of Configuration Entry	cn=Account Usability Plugin,cn=plugins,cn=config
Description	Checks the authentication status, or usability, of an account without actually authenticating as the given user
Type	preoperation
Configurable Options	on
off	Default Setting
on	Dependencies
Database	Performance-Related Information

4.1.6. AD DN Plug-in

Plug-in Parameter	Description
Plug-in ID	addn
DN of Configuration Entry	cn=addn,cn=plugins,cn=config
Description	Enables the usage of Active Directory-formatted user names, such as user_name and user_name@domain , for bind operations.
Type	preoperation
Configurable Options	on
off	Default Setting

Plug-in Parameter	Description
off	Configurable Arguments
addn_default_domain : Sets the default domain that is automatically appended to user names without domain.	Dependencies
None	Performance-Related Information

4.1.7. Attribute Uniqueness Plug-in

Plug-in Parameter	Description
Plug-in ID	NSUniqueAttr
DN of Configuration Entry	cn=Attribute Uniqueness,cn=plugins,cn=config
Description	Checks that the values of specified attributes are unique each time a modification occurs on an entry. For example, most sites require that a user ID and email address be unique.
Type	preoperation
Configurable Options	on
off	Default Setting
off	Configurable Arguments
To check for UID attribute uniqueness in all listed subtrees, enter uid "DN" "DN" However, to check for UID attribute uniqueness when adding or updating entries with the requiredObjectClass , enter attribute="uid" MarkerObjectclass = "ObjectName" and, optionally requiredObjectClass = "ObjectName" . This starts checking for the required object classes from the parent entry containing the <i>ObjectClass</i> as defined by the MarkerObjectClass attribute.	Dependencies
Database	Performance-Related Information

Plug-in Parameter	Description
<p>Directory Server provides the UID Uniqueness Plug-in by default. To ensure unique values for other attributes, create instances of the Attribute Uniqueness Plug-in for those attributes. See the "Using the Attribute Uniqueness Plug-in" section in the <i>Red Hat Directory Server Administration Guide</i> for more information about the Attribute Uniqueness Plug-in.</p> <p>The UID Uniqueness Plug-in is off by default due to operation restrictions that need to be addressed before enabling the plug-in in a multi-supplier replication environment. Turning the plug-in on may slow down Directory Server performance.</p>	<p>Further Information</p>

4.1.8. Auto Membership Plug-in

Plug-in Parameter	Description
Plug-in ID	Auto Membership
DN of Configuration Entry	cn=Auto Membership,cn=plugins,cn=config
Description	Container entry for automember definitions. Automember definitions search new entries and, if they match defined LDAP search filters and regular expression conditions, add the entry to a specified group automatically.
Type	preoperation
Configurable Options	on
off	Default Setting
off	Configurable Arguments
None for the main plug-in entry. The definition entry must specify an LDAP scope, LDAP filter, default group, and member attribute format. The optional regular expression child entry can specify inclusive and exclusive expressions and a different target group.	Dependencies
Database	Performance-Related Information
None.	Further Information

4.1.9. Binary Syntax Plug-in


WARNING

Binary syntax is deprecated. Use Octet String syntax instead.

Plug-in Parameter	Description
Plug-in ID	bin-syntax
DN of Configuration Entry	cn=Binary Syntax,cn=plugins,cn=config
Description	Syntax for handling binary data.
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.10. Bit String Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	bitstring-syntax
DN of Configuration Entry	cn=Bit String Syntax,cn=plugins,cn=config
Description	Supports bit string syntax values and related matching rules from RFC 4517 .
Type	syntax

Plug-in Parameter	Description
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.11. Bitwise Plug-in

Plug-in Parameter	Description
Plug-in ID	bitwise
DN of Configuration Entry	cn=Bitwise Plugin,cn=plugins,cn=config
Description	Matching rule for performing bitwise operations against the LDAP server
Type	matchingrule
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.12. Boolean Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	boolean-syntax
DN of Configuration Entry	cn=Boolean Syntax,cn=plugins,cn=config
Description	Supports boolean syntax values (TRUE or FALSE) and related matching rules from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.13. Case Exact String Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	ces-syntax
DN of Configuration Entry	cn=Case Exact String Syntax,cn=plugins,cn=config
Description	Supports case-sensitive matching or Directory String, IA5 String, and related syntaxes. This is not a case-exact syntax; this plug-in provides case-sensitive matching rules for different string syntaxes.
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies

Plug-in Parameter	Description
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.14. Case Ignore String Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	directorystring-syntax
DN of Configuration Entry	cn=Case Ignore String Syntax,cn=plugins,cn=config
Description	Supports case-insensitive matching rules for Directory String, IA5 String, and related syntaxes. This is not a case-insensitive syntax; this plug-in provides case-sensitive matching rules for different string syntaxes.
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.15. Chaining Database Plug-in

Plug-in Parameter	Description
Plug-in ID	chaining database
DN of Configuration Entry	cn=Chaining database,cn=plugins,cn=config

Plug-in Parameter	Description
Description	Enables back end databases to be linked
Type	database
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
There are many performance related tuning parameters involved with the chaining database. See the "Maintaining Database Links" section in the <i>Red Hat Directory Server Administration Guide</i> .	Further Information

4.1.16. Class of Service Plug-in

Plug-in Parameter	Description
Plug-in ID	cos
DN of Configuration Entry	cn=Class of Service,cn=plugins,cn=config
Description	Allows for sharing of attributes between entries
Type	object
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
* Type: Database	Performance-Related Information
* Named: State Change Plug-in	
* Named: Views Plug-in	

Plug-in Parameter	Description
Do not modify the configuration of this plug-in. Leave this plug-in running at all times.	Further Information

4.1.17. Content Synchronization Plug-in

Plug-in Parameter	Description
Plug-in ID	content-sync-plugin
DN of Configuration Entry	cn=Content Synchronization,cn=plugins,cn=config
Description	Enables support for the SyncRepl protocol in Directory Server according to RFC 4533 .
Type	object
Configurable Options	on
off	Default Setting
off	Configurable Arguments
None	Dependencies
Retro Changelog Plug-in	Performance-Related Information
If you know which back end or subtree clients access to synchronize data, limit the scope of the Retro Changelog plug-in accordingly.	Further Information

4.1.18. Country String Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	countrystring-syntax
DN of Configuration Entry	cn=Country String Syntax,cn=plugins,cn=config
Description	Supports country naming syntax values and related matching rules from RFC 4517 .
Type	syntax

Plug-in Parameter	Description
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.19. Delivery Method Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	delivery-syntax
DN of Configuration Entry	cn=Delivery Method Syntax,cn=plugins,cn=config
Description	Supports values that are lists of preferred deliver methods and related matching rules from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.20. deref Plug-in

Plug-in Parameter	Description
Plug-in ID	Dereference
DN of Configuration Entry	cn=deref,cn=plugins,cn=config
Description	For dereference controls in directory searches
Type	preoperation
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
Database	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.21. Distinguished Name Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	dn-syntax
DN of Configuration Entry	cn=Distinguished Name Syntax,cn=plugins,cn=config
Description	Supports DN value syntaxes and related matching rules from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information

Plug-in Parameter	Description
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.22. Distributed Numeric Assignment Plug-in

Plug-in Information	Description
Plug-in ID	Distributed Numeric Assignment
Configuration Entry DN	cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Description	Distributed Numeric Assignment plugin
Type	preoperation
Configurable Options	on
off	Default Setting
off	Configurable Arguments
	Dependencies
Database	Performance-Related Information
None	Further Information

4.1.23. Enhanced Guide Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	enhancedguide-syntax
DN of Configuration Entry	cn=Enhanced Guide Syntax,cn=plugins,cn=config
Description	Supports syntaxes and related matching rules for creating complex criteria, based on attributes and filters, to build searches; from RFC 4517 .
Type	syntax

Plug-in Parameter	Description
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.24. Facsimile Telephone Number Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	facsimile-syntax
DN of Configuration Entry	cn=Facsimile Telephone Number Syntax,cn=plugins,cn=config
Description	Supports syntaxes and related matching rules for fax numbers; from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.25. Fax Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	fax-syntax
DN of Configuration Entry	cn=Fax Syntax,cn=plugins,cn=config
Description	Supports syntaxes and related matching rules for storing images of faxed objects; from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.26. Generalized Time Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	time-syntax
DN of Configuration Entry	cn=Generalized Time Syntax,cn=plugins,cn=config
Description	Supports syntaxes and related matching rules for dealing with dates, times and time zones; from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies

Plug-in Parameter	Description
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.27. Guide Syntax Plug-in



WARNING	
	This syntax is deprecated. Use Enhanced Guide syntax instead.

Plug-in Parameter	Description
Plug-in ID	guide-syntax
DN of Configuration Entry	cn=Guide Syntax,cn=plugins,cn=config
Description	Syntax for creating complex criteria, based on attributes and filters, to build searches
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.28. HTTP Client Plug-in

Plug-in Parameter	Description
Plug-in ID	http-client
DN of Configuration Entry	cn=HTTP Client,cn=plugins,cn=config
Description	HTTP client plug-in
Type	preoperation
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
Database	Performance-Related Information
	Further Information

4.1.29. Integer Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	int-syntax
DN of Configuration Entry	cn=Integer Syntax,cn=plugins,cn=config
Description	Supports integer syntaxes and related matching rules from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information

Plug-in Parameter	Description
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.30. Internationalization Plug-in

Plug-in Parameter	Description
Plug-in ID	orderingrule
DN of Configuration Entry	cn=Internationalization Plugin,cn=plugins,cn=config
Description	Enables internationalized strings to be ordered in the directory
Type	matchingrule
Configurable Options	on
off	Default Setting
on	Configurable Arguments
The Internationalization Plug-in has one argument, which must not be modified, which specifies the location of the /etc/dirsrv/config/slapd-collations.conf file. This file stores the collation orders and locales used by the Internationalization Plug-in.	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.31. JPEG Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	jpeg-syntax

Plug-in Parameter	Description
DN of Configuration Entry	cn=JPEG Syntax,cn=plugins,cn=config
Description	Supports syntaxes and related matching rules for JPEG image data; from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.32. ldbm database Plug-in

Plug-in Parameter	Description
Plug-in ID	ldbm-backend
DN of Configuration Entry	cn=ldbm database,cn=plugins,cn=config
Description	Implements local databases
Type	database
Configurable Options	
Default Setting	on
Configurable Arguments	None
Dependencies	* Syntax * matchingRule
Performance-Related Information	See Section 4.4, “Database Plug-in Attributes” for further information on database configuration.

Plug-in Parameter	Description
Further Information	See the "Configuring Directory Databases" chapter in the <i>Red Hat Directory Server Administration Guide</i> .

4.1.33. Linked Attributes Plug-in

Plug-in Parameter	Description
Plug-in ID	Linked Attributes
DN of Configuration Entry	cn=Linked Attributes,cn=plugins,cn=config
Description	Container entry for linked-managed attribute configuration entries. Each configuration entry under the container links one attribute to another, so that when one entry is updated (such as a manager entry), then any entry associated with that entry (such as a custom directReports attribute) are automatically updated with a user-specified corresponding attribute.
Type	preoperation
Configurable Options	on
off	Default Setting
off	Configurable Arguments
None for the main plug-in entry. Each plug-in instance has three possible attributes: * linkType, which sets the primary attribute for the plug-in to monitor *managedType, which sets the attribute which will be managed dynamically by the plug-in whenever the attribute in linkType is modified * linkScope, which restricts the plug-in activity to a specific subtree within the directory tree	Dependencies
Database	Performance-Related Information
Any attribute set in linkType must only allow values in a DN format. Any attribute set inmanagedType must be multi-valued.	Further Information

4.1.34. Managed Entries Plug-in

Plug-in Information	Description
Plug-in ID	Managed Entries
Configuration Entry DN	cn=Managed Entries,cn=plugins,cn=config
Description	Container entry for automatically generated directory entries. Each configuration entry defines a target subtree and a template entry. When a matching entry in the target subtree is created, then the plug-in automatically creates a new, related entry based on the template.
Type	preoperation
Configurable Options	on
off	Default Setting
off	Configurable Arguments
None for the main plug-in entry. Each plug-in instance has four possible attributes: * originScope, which sets the search base * originFilter, which sets the search base for matching entries * managedScope, which sets the subtree under which to create new managed entries * managedTemplate, which is the template entry used to create the managed entries	Dependencies
Database	Performance-Related Information
None	Further Information

4.1.35. MemberOf Plug-in

Plug-in Information	Description
Plug-in ID	memberOf
Configuration Entry DN	cn=memberOf Plugin,cn=plugins,cn=config

Plug-in Information	Description
Description	Manages the memberOf attribute on user entries, based on the member attributes in the group entry.
Type	postoperation
Configurable Options	on
off	Default Setting
off	Configurable Arguments
* memberOfAttr sets the attribute to generate in people's entries to show their group membership. * memberOfGroupAttr sets the attribute to use to identify group member's DNs.	Dependencies
Database	Performance-Related Information
None	Further Information

4.1.36. Multi-master Replication Plug-in

Plug-in Parameter	Description
Plug-in ID	replication-multimaster
DN of Configuration Entry	cn=Multimaster Replication plugin,cn=plugins,cn=config
Description	Enables replication between two current Directory Servers
Type	object
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies

Plug-in Parameter	Description
* <i>Named</i> : ldbm database	Performance-Related Information
* <i>Named</i> : DES	
* <i>Named</i> : Class of Service	
	Further Information

4.1.37. Name and Optional UID Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	nameoptuid-syntax
DN of Configuration Entry	cn=Name And Optional UID Syntax,cn=plugins,cn=config
Description	Supports syntaxes and related matching rules to store and search for a DN with an optional unique ID; from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.38. Numeric String Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	numstr-syntax
DN of Configuration Entry	cn=Numeric String Syntax,cn=plugins,cn=config

Plug-in Parameter	Description
Description	Supports syntaxes and related matching rules for strings of numbers and spaces; from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.39. Octet String Syntax Plug-in



NOTE

Use the Octet String syntax instead of Binary, which is deprecated.

Plug-in Parameter	Description
Plug-in ID	octetstring-syntax
DN of Configuration Entry	cn=Octet String Syntax,cn=plugins,cn=config
Description	Supports octet string syntaxes and related matching rules from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies

Plug-in Parameter	Description
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.40. OID Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	oid-syntax
DN of Configuration Entry	cn=OID Syntax,cn=plugins,cn=config
Description	Supports object identifier (OID) syntaxes and related matching rules from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.41. PAM Pass Through Auth Plug-in

Plug-in Parameter	Description
Plug-in ID	pam_passthruauth
DN of Configuration Entry	cn=PAM Pass Through Auth,cn=plugins,cn=config
Description	Enables pass-through authentication for PAM, meaning that a PAM service can use the Directory Server as its user authentication store.

Plug-in Parameter	Description
Type	preoperation
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
Database	Performance-Related Information
	Further Information

4.1.42. Pass Through Authentication Plug-in

Plug-in Parameter	Description
Plug-in ID	passthruauth
DN of Configuration Entry	cn=Pass Through Authentication,cn=plugins,cn=config
Description	Enables <i>pass-through authentication</i> , the mechanism which allows one directory to consult another to authenticate bind requests.
Type	preoperation
Configurable Options	on
off	Default Setting
off	Configurable Arguments
ldap://example.com:389/o=example	Dependencies
Database	Performance-Related Information
Pass-through authentication slows down bind requests a little because they have to make an extra hop to the remote server. See the "Using Pass-through Authentication" chapter in the <i>Red Hat Directory Server Administration Guide</i> .	Further Information

4.1.43. Password Storage Schemes

Directory Server implements the password storage schemes as plug-ins. However, the **cn=Password Storage Schemes,cn=plugins,cn=config** entry itself is just a container, not a plug-in entry. All password storage scheme plug-ins are stored as a subentry of this container.

To display all password storage schemes plug-ins, enter:

```
# ldapsearch -D "cn=Directory Manager" -W -p 389 -h server.example.com -x \
-b "cn=Password Storage Schemes,cn=plugins,cn=config" -s sub "(objectclass=*)" dn
```



WARNING

Red Hat recommends not disabling the password scheme plug-ins nor to change the configurations of the plug-ins to prevent unpredictable authentication behavior.

Strong Password Storage Schemes

Red Hat recommends using only the following strong password storage schemes (strongest first):

- **PBKDF2_SHA256** (default)

The password-based key derivation function 2 (PBKDF2) was designed to expend resources to counter brute force attacks. PBKDF2 supports a variable number of iterations to apply the hashing algorithm. Higher iterations improve security but require more hardware resources. In Directory Server, the **PBKDF2_SHA256** scheme is implemented using 30,000 iterations to apply the SHA256 algorithm. This value is hard-coded and will be increased in future versions of Directory Server without requiring interaction by an administrator.



NOTE

The network security service (NSS) database in Red Hat Enterprise Linux 6 does not support PBKDF2. Therefore you cannot use this password scheme in a replication topology with Directory Server 9.

- **SSHA512**

The salted secure hashing algorithm (SSHA) implements an enhanced version of the secure hashing algorithm (SHA), that uses a randomly generated salt to increase the security of the hashed password. **SSHA512** implements the hashing algorithm using 512 bits.

Weak Password Storage Schemes

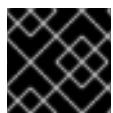
Besides the recommended strong password storage schemes, Directory Server supports the following weak schemes for backward compatibility:

AES	CLEAR	CRYPT
CRYPT-MD5	CRYPT-SHA256	CRYPT-SHA512

DES	MD5	NS-MTA-MD5
		[a]
SHA [b]	SHA256	SHA384
SHA512	SMD5	SSHA
SSHA256	SSHA384	

[a] Directory Server only supports authentication using this scheme. You can no longer use it to encrypt passwords.

[b] 160 bit

**IMPORTANT**

Only continue using a weak scheme over a short time frame, as it increases security risks.

4.1.44. Posix Winsync API Plug-in

Plug-in Parameter	Description
Plug-in ID	posix-winsync-plugin
DN of Configuration Entry	cn=Posix Winsync API,cn=plugins,cn=config
Description	Enables and configures Windows synchronization for Posix attributes set on Active Directory user and group entries.
Type	preoperation
Configurable Arguments	* on
off	Default Setting
* memberUID mapping (groups)	
* converting and sorting memberUID values in lower case (groups)	
* memberOf fix-up tasks with sync operations	
* use Windows 2003 Posix schema	
off	Configurable Arguments

Plug-in Parameter	Description
None	Dependencies

4.1.45. Postal Address String Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	postaladdress-syntax
DN of Configuration Entry	cn=Postal Address Syntax,cn=plugins,cn=config
Description	Supports postal address syntaxes and related matching rules from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.46. Printable String Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	printablestring-syntax
DN of Configuration Entry	cn=Printable String Syntax,cn=plugins,cn=config
Description	Supports syntaxes and matching rules for alphanumeric and select punctuation strings (for strings which conform to printable strings as defined in RFC 4517).
Type	syntax

Plug-in Parameter	Description
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.47. Referential Integrity Postoperation Plug-in

Plug-in Parameter	Description
Plug-in ID	referint
DN of Configuration Entry	cn=Referential Integrity Postoperation,cn=plugins,cn=config
Description	Enables the server to ensure referential integrity
Type	postoperation
Configurable Options	All configuration and on
off	Default Setting
off	Configurable Arguments
When enabled, the post-operation Referential Integrity Plug-in performs integrity updates on the member , uniqueMember , owner , and seeAlso attributes immediately after a delete or rename operation. The plug-in can be configured to perform integrity checks on all other attributes. For details, see the corresponding section in the <i>Directory Server Administration Guide</i> .	Dependencies
Database	Performance-Related Information

Plug-in Parameter	Description
The Referential Integrity Plug-in should be enabled only on one supplier in a multi-supplier replication environment to avoid conflict resolution loops. When enabling the plug-in on chained servers, be sure to analyze the performance resource and time needs as well as integrity needs; integrity checks can be time consuming and demanding on memory and CPU. All attributes specified must be indexed for both presence and equality.	Further Information

4.1.48. Retro Changelog Plug-in

Plug-in Parameter	Description
Plug-in ID	retrocl
DN of Configuration Entry	cn=Retro Changelog Plugin,cn=plugins,cn=config
Description	Used by LDAP clients for maintaining application compatibility with Directory Server 4.x versions. Maintains a log of all changes occurring in the Directory Server. The retro changelog offers the same functionality as the changelog in the 4.x versions of Directory Server. This plug-in exposes the cn=changelog suffix to clients, so that clients can use this suffix with or without persistent search for simple sync applications.
Type	object
Configurable Options	on
off	Default Setting
off	Configurable Arguments
See Section 4.16, “Retro Changelog Plug-in Attributes ” for further information on the two configuration attributes for this plug-in.	Dependencies
* Type: Database	Performance-Related Information
* Named: Class of Service	
May slow down Directory Server update performance.	Further Information

Plug-in Parameter	Description
-------------------	-------------

4.1.49. Roles Plug-in

Plug-in Parameter	Description
Plug-in ID	roles
DN of Configuration Entry	cn=Roles Plugin,cn=plugins,cn=config
Description	Enables the use of roles in the Directory Server
Type	object
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
* Type: Database	Performance-Related Information
* Named: State Change Plug-in	
* Named: Views Plug-in	
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.50. RootDN Access Control Plug-in

Plug-in Parameter	Description
Plug-in ID	rootdn-access-control
DN of Configuration Entry	cn=RootDN Access Control,cn=plugins,cn=config
Description	Enables and configures access controls to use for the root DN entry.

Plug-in Parameter	Description
Type	internalpreoperation
Configurable Options	on
off	Default Setting
off	Configurable Attributes
* rootdn-open-time and rootdn-close-time for time-based access controls * rootdn-days-allowed for day-based access controls * rootdn-allow-host, rootdn-deny-host, rootdn-allow-ip, and rootdn-deny-ip for host-based access controls	Dependencies
None	Further Information

4.1.51. Schema Reload Plug-in

Plug-in Information	Description
Plug-in ID	schemareload
Configuration Entry DN	cn=Schema Reload,cn=plugins,cn=config
Description	Task plug-in to reload schema files
Type	object
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
	Further Information

4.1.52. Space Insensitive String Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	none
DN of Configuration Entry	cn=Space Insensitive String Syntax,cn=plugins,cn=config
Description	Syntax for handling space-insensitive values
Type	syntax
Configurable Options	on
off	Default Setting
off	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.53. State Change Plug-in

Plug-in Parameter	Description
Plug-in ID	statechange
DN of Configuration Entry	cn=State Change Plugin,cn=plugins,cn=config
Description	Enables state-change-notification service
Type	postoperation
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information

Plug-in Parameter	Description
	Further Information

4.1.54. Syntax Validation Task Plug-in

Plug-in Parameter	Description
Plug-in ID	none
DN of Configuration Entry	cn=Syntax Validation Task,cn=plugins,cn=config
Description	Enables syntax validation for attribute values
Type	object
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
	Further Information

4.1.55. Telephone Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	tele-syntax
DN of Configuration Entry	cn=Telephone Syntax,cn=plugins,cn=config
Description	Supports telephone number syntaxes and related matching rules from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting

Plug-in Parameter	Description
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.56. Teletex Terminal Identifier Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	teletextermid-syntax
DN of Configuration Entry	cn=Teletex Terminal Identifier Syntax,cn=plugins,cn=config
Description	Supports international telephone number syntaxes and related matching rules from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.57. Telex Number Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	telex-syntax

Plug-in Parameter	Description
DN of Configuration Entry	cn=Telex Number Syntax,cn=plugins,cn=config
Description	Supports syntaxes and related matching rules for the telex number, country code, and answerback code of a telex terminal; from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
None	Performance-Related Information
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.58. URI Syntax Plug-in

Plug-in Parameter	Description
Plug-in ID	none
DN of Configuration Entry	cn=URI Syntax,cn=plugins,cn=config
Description	Supports syntaxes and related matching rules for unique resource identifiers (URIs), including unique resource locators (URLs); from RFC 4517 .
Type	syntax
Configurable Options	on
off	Default Setting
off	Configurable Arguments
None	Dependencies
None	Performance-Related Information

Plug-in Parameter	Description
Do not modify the configuration of this plug-in. If enabled, Red Hat recommends leaving this plug-in running at all times.	Further Information

4.1.59. USN Plug-in

Plug-in Parameter	Description
Plug-in ID	USN
DN of Configuration Entry	cn=USN,cn=plugins,cn=config
Description	Sets an update sequence number (USN) on an entry, for every entry in the directory, whenever there is a modification, including adding and deleting entries and modifying attribute values.
Type	object
Configurable Options	on
off	Default Setting
off	Configurable Arguments
None	Dependencies
Database	Performance-Related Information
For replication, it is recommended that the entryUSN configuration attribute be excluded using fractional replication.	Further Information

4.1.60. Views Plug-in

Plug-in Parameter	Description
Plug-in ID	views
DN of Configuration Entry	cn=Views,cn=plugins,cn=config
Description	Enables the use of views in the Directory Server databases.

Plug-in Parameter	Description
Type	object
Configurable Options	on
off	Default Setting
on	Configurable Arguments
None	Dependencies
* Type: Database	Performance-Related Information
* Named: State Change Plug-in	
Do not modify the configuration of this plug-in. Red Hat recommends leaving this plug-in running at all times.	Further Information

4.2. LIST OF ATTRIBUTES COMMON TO ALL PLUG-INS

This list provides a brief attribute description, the entry DN, valid range, default value, syntax, and an example for each attribute.

4.2.1. nsslapdPlugin (Object Class)

Each Directory Server plug-in belongs to the **nsslapdPlugin** object class.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.41

Table 4.1. Required Attributes

Attribute	Definition
objectClass	Gives the object classes assigned to the entry.
cn	Gives the common name of the entry.
Section 4.2.8, “nsslapd-pluginPath”	Identifies the plugin library name (without the library suffix).

Attribute	Definition
Section 4.2.7, "nsslapd-pluginInitfunc"	Identifies an initialization function of the plugin.
Section 4.2.10, "nsslapd-pluginType"	Identifies the type of plugin.
Section 4.2.6, "nsslapd-pluginId"	Identifies the plugin ID.
Section 4.2.12, "nsslapd-pluginVersion"	Identifies the version of plugin.
Section 4.2.11, "nsslapd-pluginVendor"	Identifies the vendor of plugin.
Section 4.2.4, "nsslapd-pluginDescription"	Identifies the description of the plugin.
Section 4.2.5, "nsslapd-pluginEnabled"	Identifies whether or not the plugin is enabled.
Section 4.2.9, "nsslapd-pluginPrecedence"	Sets the priority for the plug-in in the execution order.

4.2.2. nsslapd-logAccess

This attribute enables you to log search operations run by the plug-in to the file set in the **nsslapd-accesslog** parameter in **cn=config**.

Plug-in Parameter	Description
Entry DN	cn=plug-in name,cn=plugins,cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-logAccess: Off

4.2.3. nsslapd-logAudit

This attribute enables you to log and audit modifications to the database originated from the plug-in.

Successful modification events are logged in the audit log, if the **nsslapd-auditlog-logging-enabled** parameter is enabled in **cn=config**. To log failed modification database operations by a plug-in, enable the **nsslapd-auditfaillog-logging-enabled** attribute in **cn=config**.

Plug-in Parameter	Description
Entry DN	cn=plug-in name,cn=plugins,cn=config

Plug-in Parameter	Description
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-logAudit: Off

4.2.4. nsslapd-pluginDescription

This attribute provides a description of the plug-in.

Plug-in Parameter	Description
Entry DN	cn=plug-in name,cn=plugins,cn=config
Valid Values	
Default Value	None
Syntax	DirectoryString
Example	nsslapd-pluginDescription: acl access check plug-in

4.2.5. nsslapd-pluginEnabled

This attribute specifies whether the plug-in is enabled. This attribute can be changed over protocol but will only take effect when the server is next restarted.

Plug-in Parameter	Description
Entry DN	cn=plug-in name,cn=plugins,cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-pluginEnabled: on

4.2.6. nsslapd-pluginId

This attribute specifies the plug-in ID.

Plug-in Parameter	Description
Entry DN	cn=plug-in name,cn=plugins,cn=config
Valid Values	Any valid plug-in ID
Default Value	None
Syntax	DirectoryString
Example	nsslapd-pluginId: chaining database

4.2.7. nsslapd-pluginInitfunc

This attribute specifies the plug-in function to be initiated.

Plug-in Parameter	Description
Entry DN	cn=plug-in name,cn=plugins,cn=config
Valid Values	Any valid plug-in function
Default Value	None
Syntax	DirectoryString
Example	nsslapd-pluginInitfunc: NS7bitAttr_Init

4.2.8. nsslapd-pluginPath

This attribute specifies the full path to the plug-in.

Plug-in Parameter	Description
Entry DN	cn=plug-in name,cn=plugins,cn=config
Valid Values	Any valid path
Default Value	None
Syntax	DirectoryString

Plug-in Parameter	Description
Example	nsslapd-pluginPath: uid-plugin

4.2.9. nsslapd-pluginPrecedence

This attribute sets the precedence or priority for the execution order of a plug-in. Precedence defines the execution order of plug-ins, which allows more complex environments or interactions since it can enable a plug-in to wait for a completed operation before being executed. This is more important for pre-operation and post-operation plug-ins.

Plug-ins with a value of 1 have the highest priority and are run first; plug-ins with a value of 99 have the lowest priority. The default is 50.

Plug-in Parameter	Description
Entry DN	cn=plug-in name,cn=plugins,cn=config
Valid Values	1 to 99
Default Value	50
Syntax	Integer
Example	nsslapd-pluginPrecedence: 3

4.2.10. nsslapd-pluginType

This attribute specifies the plug-in type. See [Section 4.3.5, “nsslapd-plugin-depends-on-type”](#) for further information.

Plug-in Parameter	Description
Entry DN	cn=plug-in name,cn=plugins,cn=config
Valid Values	Any valid plug-in type
Default Value	None
Syntax	DirectoryString
Example	nsslapd-pluginType: preoperation

4.2.11. nsslapd-pluginVendor

This attribute specifies the vendor of the plug-in.

Plug-in Parameter	Description
Entry DN	cn=plug-in name,cn=plugins,cn=config
Valid Values	Any approved plug-in vendor
Default Value	Red Hat, Inc.
Syntax	DirectoryString
Example	nsslapd-pluginVendor: Red Hat, Inc.

4.2.12. nsslapd-pluginVersion

This attribute specifies the plug-in version.

Plug-in Parameter	Description
Entry DN	cn=plug-in name,cn=plugins,cn=config
Valid Values	Any valid plug-in version
Default Value	Product version number
Syntax	DirectoryString
Example	nsslapd-pluginVersion: 11.3

4.3. ATTRIBUTES ALLOWED BY CERTAIN PLUG-INS

4.3.1. nsslapd-dynamic-plugins

Directory Server supports dynamic plug-ins that can be enabled without restarting the server. The **nsslapd-dynamic-plugins** attribute specifies whether the server is configured to allow for dynamic plug-ins. By default, dynamic plug-ins are disabled.

Some plug-ins cannot be configured as dynamic, and they require the server to be restarted.

Plug-in Parameter	Description
Entry DN	cn=config
Valid Values	on off
Default Value	off

Plug-in Parameter	Description
Syntax	DirectoryString
Example	nsslapd-dynamic-plugins: on

4.3.2. nsslapd-pluginConfigArea

Some plug-in entries are container entries, and multiple instances of the plug-in are created beneath this container in **cn=plugins,cn=config**. However, the **cn=plugins,cn=config** is not replicated, which means that the plug-in configurations beneath those container entries must be configured manually, in some way, on every Directory Server instance.

The **nsslapd-pluginConfigArea** attribute points to another container entry, in the main database area, which contains the plug-in instance entries. This container entry can be in a replicated database, which allows the plug-in configuration to be replicated.

Plug-in Parameter	Description
Entry DN	<i>cn=plug-in name,cn=plugins,cn=config</i>
Valid Values	Any valid DN
Default Value	
Syntax	DN
Example	nsslapd-pluginConfigArea: cn=managed entries container,ou=containers,dc=example,dc=com

4.3.3. nsslapd-pluginLoadNow

This attribute specifies whether to load all of the symbols used by a plug-in immediately (**true**), as well as all symbols references by those symbols, or to load the symbol the first time it is used (**false**).

Plug-in Parameter	Description
Entry DN	<i>cn=plug-in name,cn=plugins,cn=config</i>
Valid Values	true false
Default Value	false
Syntax	DirectoryString
Example	nsslapd-pluginLoadNow: false

4.3.4. nsslapd-pluginLoadGlobal

This attribute specifies whether the symbols in dependent libraries are made visible locally (**false**) or to the executable and to all shared objects (**true**).

Plug-in Parameter	Description
Entry DN	cn=plug-in name,cn=plugins,cn=config
Valid Values	true false
Default Value	false
Syntax	DirectoryString
Example	nsslapd-pluginLoadGlobal: false

4.3.5. nsslapd-plugindepends-on-type

Multi-valued attribute used to ensure that plug-ins are called by the server in the correct order. Takes a value which corresponds to the type number of a plug-in, contained in the attribute **nsslapd-pluginType**. See [Section 4.2.10, “nsslapd-pluginType”](#) for further information. All plug-ins with a type value which matches one of the values in the following valid range will be started by the server prior to this plug-in. The following postoperation Referential Integrity Plug-in example shows that the database plug-in will be started prior to the postoperation Referential Integrity Plug-in.

Plug-in Parameter	Description
Entry DN	cn=referential integrity postoperation,cn=plugins,cn=config
Valid Values	database
Default Value	
Syntax	DirectoryString
Example	nsslapd-plugindepends-on-type: database

4.3.6. nsslapd-plugindepends-on-named

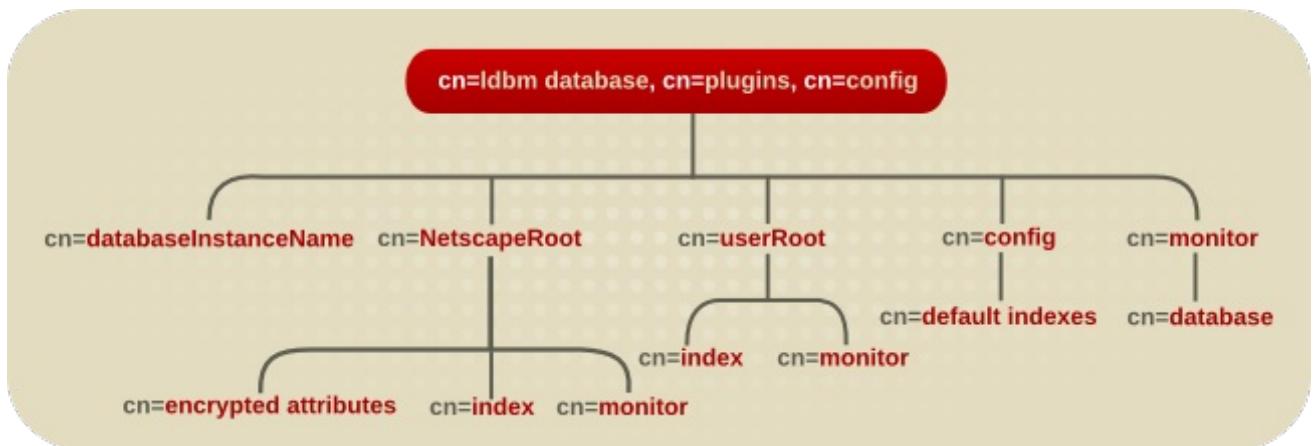
Multi-valued attribute used to ensure that plug-ins are called by the server in the correct order. Takes a value which corresponds to the **cn** value of a plug-in. The plug-in with a **cn** value matching one of the following values will be started by the server prior to this plug-in. If the plug-in does not exist, the server fails to start. The following postoperation Referential Integrity Plug-in example shows that the Views plug-in is started before Roles. If Views is missing, the server is not going to start.

Plug-in Parameter	Description
Entry DN	cn=referential integrity postoperation,cn=plugins,cn=config
Valid Values	Class of Service
Default Value	
Syntax	DirectoryString
Example	* nsslapd-plugin-depends-on-named: Views * nsslapd-pluginId: roles

4.4. DATABASE PLUG-IN ATTRIBUTES

The database plug-in is also organized in an information tree, as shown in [Figure 4.1, “Database Plug-in”](#).

Figure 4.1. Database Plug-in



All plug-in technology used by the database instances is stored in the **cn=ldbmdatabase** plug-in node. This section presents the additional attribute information for each of the nodes in bold in the **cn=ldbmdatabase,cn=plugins,cn=config** information tree.

4.4.1. Database Attributes under **cn=config,cn=ldbmdatabase,cn=plugins,cn=config**

This section covers global configuration attributes common to all instances are stored in the **cn=config,cn=ldbmdatabase,cn=plugins,cn=config** tree node.

4.4.1.1. **nsslapd-backend-implement**

The **nsslapd-backend-implement** parameter defines the database back end Directory Server uses.



IMPORTANT

Directory Server currently only supports the Berkeley Database (BDB). Therefore, you cannot set this parameter to a different value.

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	bdb
Default Value	bdb
Syntax	Directory String
Example	nsslapd-backend-implement: bdb

4.4.1.2. nsslapd-backend-opt-level

This parameter can trigger experimental code to improve write performance.

Possible values:

- **0:** Disables the parameter.
- **1:** The replication update vector is not written to the database during the transaction
- **2:** Changes the order of taking the back end lock and starts the transaction
- **4:** Moves code out of the transaction.

All parameters can be combined. For example **7** enables all optimisation features.



WARNING

This parameter is experimental. *Never change its value unless you are specifically told to do so by the Red Hat support.*

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	0 1 2 4
Default Value	0
Syntax	Integer
Example	nsslapd-backend-opt-level: 0

4.4.1.3. nsslapd-directory

This attribute specifies absolute path to database instance. If the database instance is manually created then this attribute must be included, something which is set by default (and modifiable) in the Directory Server Console. Once the database instance is created, do not modify this path as any changes risk preventing the server from accessing data.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	Any valid absolute path to the database instance
Default Value	
Syntax	DirectoryString
Example	nsslapd-directory: /var/lib/dirsrv/slapd- <i>instance</i> /db

4.4.1.4. nsslapd-exclude-from-export

This attribute contains a space-separated list of names of attributes to exclude from an entry when a database is exported. This mainly is used for some configuration and operational attributes which are specific to a server instance.

Do not remove any of the default values for this attribute, since that may affect server performance.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	Any valid attribute
Default Value	entrydn entryid dncomp parentid numSubordinates entryusn
Syntax	DirectoryString
Example	nsslapd-exclude-from-export: entrydn entryid dncomp parentid numSubordinates entryusn

4.4.1.5. nsslapd-db-transaction-wait

If you enable the **nsslapd-db-transaction-wait** parameter, Directory Server does not start the transaction and waits until lock resources are available.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-db-transaction-wait: off

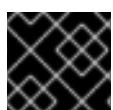
4.4.1.6. nsslapd-db-private-import-mem

The **nsslapd-db-private-import-mem** parameter manages whether or not Directory Server uses private memory for allocation of regions and mutexes for a database import.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-db-private-import-mem: on

4.4.1.7. nsslapd-db-deadlock-policy

The **nsslapd-db-deadlock-policy** parameter sets the **libdb** library-internal deadlock policy.



IMPORTANT

Only change this parameter if instructed by Red Hat Support.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	0-9
Default Value	0
Syntax	DirectoryString

Parameter	Description
Example	nsslapd-db-deadlock-policy: 0

4.4.1.8. nsslapd-idl-switch

The **nsslapd-idl-switch** parameter sets the IDL format Directory Server uses. Note that Red Hat no longer supports the old IDL format.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	new old
Default Value	new
Syntax	Directory String
Example	nsslapd-idl-switch: new

4.4.1.9. nsslapd-idlistscanlimit

This performance-related attribute, present by default, specifies the number of entry IDs that are searched during a search operation. Attempting to set a value that is not a number or is too big for a 32-bit signed integer returns an **LDAP_UNWILLING_TO_PERFORM** error message, with additional error information explaining the problem. It is advisable to keep the default value to improve search performance.

For further details, see the corresponding sections in the:

- [Directory Server Performance Tuning Guide](#)
- [Directory Server Administration Guide](#)

This parameter can be changed while the server is running, and the new value will affect subsequent searches.

The corresponding user-level attribute is **nsIDListScanLimit**.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	100 to the maximum 32-bit integer value (2147483647) entry IDs
Default Value	4000

Parameter	Description
Syntax	Integer
Example	nsslapd-idlistscanlimit: 4000

4.4.1.10. nsslapd-lookthroughlimit

This performance-related attribute specifies the maximum number of entries that the Directory Server will check when examining candidate entries in response to a search request. The Directory Manager DN, however, is, by default, unlimited and overrides any other settings specified here. It is worth noting that binder-based resource limits work for this limit, which means that if a value for the operational attribute **nsLookThroughLimit** is present in the entry as which a user binds, the default limit will be overridden. Attempting to set a value that is not a number or is too big for a 32-bit signed integer returns an **LDAP_UNWILLING_TO_PERFORM** error message with additional error information explaining the problem.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	-1 to maximum 32-bit integer in entries (where -1 is unlimited)
Default Value	5000
Syntax	Integer
Example	nsslapd-lookthroughlimit: 5000

4.4.1.11. nsslapd-mode

This attribute specifies the permissions used for newly created index files.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	Any four-digit octal number. However, mode 0600 is recommended. This allows read and write access for the owner of the index files (which is the user as whom the ns-slapd runs) and no access for other users.
Default Value	600
Syntax	Integer

Parameter	Description
Example	nsslapd-mode: 0600

4.4.1.12. nsslapd-pagedidlistscanlimit

This performance-related attribute specifies the number of entry IDs that are searched, specifically, for a search operation using the simple paged results control.

This attribute works the same as the **nsslapd-idlistscanlimit** attribute, except that it only applies to searches with the simple paged results control.

If this attribute is not present or is set to zero, then the **nsslapd-idlistscanlimit** is used to paged searches as well as non-paged searches.

The corresponding user-level attribute is **nsPagedIDListScanLimit**.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	-1 to maximum 32-bit integer in entries (where -1 is unlimited)
Default Value	0
Syntax	Integer
Example	nsslapd-pagedidlistscanlimit: 5000

4.4.1.13. nsslapd-pagedlookthroughlimit

This performance-related attribute specifies the maximum number of entries that the Directory Server will check when examining candidate entries for a search which uses the simple paged results control.

This attribute works the same as the **nsslapd-lookthroughlimit** attribute, except that it only applies to searches with the simple paged results control.

If this attribute is not present or is set to zero, then the **nsslapd-lookthroughlimit** is used to paged searches as well as non-paged searches.

The corresponding user-level attribute is **nsPagedLookThroughLimit**.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	-1 to maximum 32-bit integer in entries (where -1 is unlimited)

Parameter	Description
Default Value	0
Syntax	Integer
Example	nsslapd-pagedlookthroughlimit: 25000

4.4.1.14. nsslapd-rangelookthroughlimit

This performance-related attribute specifies the maximum number of entries that the Directory Server will check when examining candidate entries in response to a range search request.

Range searches use operators to set a bracket to search for and return an entire subset of entries within the directory. For example, this searches for every entry modified at or after midnight on January 1:

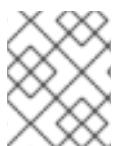
(modifyTimestamp>=20200101010101Z)

The nature of a range search is that it must evaluate every single entry within the directory to see if it is within the range given. Essentially, a range search is always an all IDs search.

For most users, the look-through limit kicks in and prevents range searches from turning into an all IDs search. This improves overall performance and speeds up range search results. However, some clients or administrative users like Directory Manager may not have a look-through limit set. In that case, a range search can take several minutes to complete or even continue indefinitely.

The **nsslapd-rangelookthroughlimit** attribute sets a separate range look-through limit that applies to all users, including Directory Manager.

This allows clients and administrative users to have high look-through limits while still allowing a reasonable limit to be set on potentially performance-impaired range searches.



NOTE

Unlike other resource limits, this applies to searches by any user, including the Directory Manager, regular users, and other LDAP clients.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	-1 to maximum 32-bit integer in entries (where -1 is unlimited)
Default Value	5000
Syntax	Integer
Example	nsslapd-rangelookthroughlimit: 5000

4.4.1.15. nsslapd-search-bypass-filter-test

If you enable the **nsslapd-search-bypass-filter-test** parameter, Directory Server bypasses filter checks when it builds candidate lists during a search. If you set the parameter to **verify**, Directory Server evaluates the filter against the search candidate entries.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	on off verify
Default Value	on
Syntax	Directory String
Example	nsslapd-search-bypass-filter-test: on

4.4.1.16. nsslapd-search-use-vlv-index

The **nsslapd-search-use-vlv-index** enables and disables virtual list view (VLV) searches.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	on off
Default Value	on
Syntax	Directory String
Example	nsslapd-search-use-vlv-index: on

4.4.1.17. nsslapd-subtree-rename-switch

Every directory entry is stored as a key in an entry index file. The index key maps the current entry DN to its meta entry in the index. This mapping is done either by the RDN of the entry or by the full DN of the entry.

When a subtree entry is allowed to be renamed (meaning, an entry with children entries, effectively renaming the whole subtree), its entries are stored in the **entryrdn.db** index, which associates parent and child entries by an assigned ID rather than their DN. If subtree rename operations are not allowed, then the **entryrdn.db** index is disabled and the **entrydn.db** index is used, which simply uses full DNs, with the implicit parent-child relationships.

Parameter	Description
Entry DN	cn=config,cn=ldb database,cn=plugins,cn=config
Valid Values	off on
Default Value	on
Syntax	DirectoryString
Example	nsslapd-subtree-rename-switch: on

4.4.2. Database Attributes under **cn=bdb,cn=config,cn=ldb database,cn=plugins,cn=config**

This section covers global configuration attributes common to all instances are stored in the **cn=bdb,cn=config,cn=ldb database,cn=plugins,cn=config** tree node.

4.4.2.1. **nsslapd-cache-autosize**

This performance tuning-related attribute sets the percentage of free memory that is used in total for the database and entry cache. For example, if the value is set to **10**, 10% of the system's free RAM is used for both caches. If this value is set to a value greater than **0**, auto-sizing is enabled for the database and entry cache.

For optimized performance, Red Hat recommends not to disable auto-sizing. However, in certain situations it can be necessary to disable auto-sizing. In this case, set the **nsslapd-cache-autosize** attribute to **0** and manually set:

- the database cache in the **nsslapd-dbcache size** attribute.
- the entry cache in the **nsslapd-cachememsize** attribute.

For further details about auto-sizing, see the corresponding section in the *Red Hat Directory Server Performance Tuning Guide*.



NOTE

If the **nsslapd-cache-autosize** and **nsslapd-cache-autosize-split** attribute are both set to high values, such as **100**, Directory Server fails to start. To fix the problem, set both parameters to more reasonable values. For example:

nsslapd-cache-autosize: 10
nsslapd-cache-autosize-split: 40

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldb database,cn=plugins,cn=config

Parameter	Description
Valid Range	0 to 100. If 0 is set, the default value is used instead.
Default Value	10
Syntax	Integer
Example	nsslapd-cache-autosize: 10

4.4.2.2. nsslapd-cache-autosize-split

This performance tuning-related attribute sets the percentage of RAM that is used for the database cache. The remaining percentage is used for the entry cache. For example, if the value is set to **40**, the database cache uses 40%, and the entry cache the remaining 60% of the free RAM reserved in the **nsslapd-cache-autosize** attribute.

For further details about auto-sizing, see the corresponding section in the [Red Hat Directory Server Performance Tuning Guide](#).



NOTE

If the **nsslapd-cache-autosize** and **nsslapd-cache-autosize-split** attribute are both set to high values, such as **100**, Directory Server fails to start. To fix the problem, set both parameters to more reasonable values. For example:

```
nsslapd-cache-autosize: 10
nsslapd-cache-autosize-split: 40
```

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 to 99. If 0 is set, the default value is used instead.
Default Value	40
Syntax	Integer
Example	nsslapd-cache-autosize-split: 40

4.4.2.3. nsslapd-db-checkpoint-interval

This sets the amount of time in seconds after which the Directory Server sends a checkpoint entry to the database transaction log. The database transaction log contains a sequential listing of all recent database operations and is used for database recovery only. A checkpoint entry indicates which

database operations have been physically written to the directory database. The checkpoint entries are used to determine where in the database transaction log to begin recovery after a system failure. The **nsslapd-db-checkpoint-interval** attribute is absent from **dse.Idif**. To change the checkpoint interval, add the attribute to **dse.Idif**. This attribute can be dynamically modified using **Idapmodify**. For further information on modifying this attribute, see the "Tuning Directory Server Performance" chapter in the *Red Hat Directory Server Administration Guide*.

This attribute is provided only for system modification/diagnostics and should be changed only with the guidance of Red Hat Technical Support or Red Hat Consulting. Inconsistent settings of this attribute and other configuration attributes may cause the Directory Server to be unstable.

For more information on database transaction logging, see the "Monitoring Server and Database Activity" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	10 to 300 seconds
Default Value	60
Syntax	Integer
Example	nsslapd-db-checkpoint-interval: 120

4.4.2.4. nsslapd-db-circular-logging

This attribute specifies circular logging for the transaction log files. If this attribute is switched off, old transaction log files are not removed and are kept renamed as old log transaction files. Turning circular logging off can severely degrade server performance and, as such, should only be modified with the guidance of Red Hat Technical Support or Red Hat Consulting.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-db-circular-logging: on

4.4.2.5. nsslapd-db-compactdb-interval

The Berkeley database does not reuse free pages unless the database is explicitly compacted. The

compact operation returns the unused pages to the file system and the database file size shrinks. This parameter defines the interval in seconds when the database is compacted. Note that compacting the database is resource-intensive, and thus should not be done too frequently.

This setting does not require a server restart to take effect.

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	0 (no compaction) to 2147483647 seconds
Default Value	2592000 (30 days)
Syntax	Integer
Example	nsslapd-compactdb-interval: 2592000

4.4.2.6. nsslapd-db-debug

This attribute specifies whether additional error information is to be reported to Directory Server. To report error information, set the parameter to **on**. This parameter is meant for troubleshooting; enabling the parameter may slow down the Directory Server.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-db-debug: off

4.4.2.7. nsslapd-db-durable-transactions

This attribute sets whether database transaction log entries are immediately written to the disk. The database transaction log contains a sequential listing of all recent database operations and is used for database recovery only. With durable transactions enabled, every directory change will always be physically recorded in the log file and, therefore, able to be recovered in the event of a system failure. However, the durable transactions feature may also slow the performance of the Directory Server. When durable transactions is disabled, all transactions are logically written to the database transaction log but may not be physically written to disk immediately. If there were a system failure before a directory change was physically written to disk, that change would not be recoverable. The **nsslapd-db-durable-transactions** attribute is absent from **dse.Idif**. To disable durable transactions, add the attribute to **dse.Idif**.

This attribute is provided only for system modification/diagnostics and should be changed only with the guidance of Red Hat Technical Support or Red Hat Consulting. Inconsistent settings of this attribute and other configuration attributes may cause the Directory Server to be unstable.

For more information on database transaction logging, see the "Monitoring Server and Database Activity" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsslapd-db-durable-transactions: on

4.4.2.8. nsslapd-db-home-directory

To move the database to another physical location for performance reasons, use this parameter to specify the home directory.

This situation will occur only for certain combinations of the database cache size, the size of physical memory, and kernel tuning attributes. In particular, this situation should not occur if the database cache size is less than 100 megabytes.

- The disk is heavily used (more than 1 megabyte per second of data transfer).
- There is a long service time (more than 100ms).
- There is mostly write activity.

If these are all true, use the **nsslapd-db-home-directory** attribute to specify a subdirectory of a **tmpfs** type filesystem.

The directory referenced by the **nsslapd-db-home-directory** attribute must be a subdirectory of a filesystem of type tmpfs (such as **/tmp**). However, Directory Server does not create the subdirectory referenced by this attribute. This directory must be created either manually or by using a script. Failure to create the directory referenced by the **nsslapd-db-home-directory** attribute will result in Directory Server being unable to start.

Also, if there are multiple Directory Servers on the same machine, their **nsslapd-db-home-directory** attributes must be configured with different directories. Failure to do so will result in the databases for both directories becoming corrupted.

The use of this attribute causes internal Directory Server database files to be moved to the directory referenced by the attribute. It is possible, but unlikely, that the server will no longer start after the files have been moved because not enough memory can be allocated. This is a symptom of an overly large database cache size being configured for the server. If this happens, reduce the size of the database cache size to a value where the server will start again.

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	Any valid directory name in a tmpfs filesystem, such as /tmp
Default Value	
Syntax	DirectoryString
Example	nsslapd-db-home-directory: /tmp/slapd-phonebook

4.4.2.9. nsslapd-db-idx-divisor

This attribute specifies the index block size in terms of the number of blocks per database page. The block size is calculated by dividing the database page size by the value of this attribute. A value of **1** makes the block size exactly equal to the page size. The default value of **0** sets the block size to the page size minus an estimated allowance for internal database overhead. For the majority of installations, the default value should not be changed unless there are specific tuning needs.

Before modifying the value of this attribute, export all databases using the **db2ldif** script. Once the modification has been made, reload the databases using the **ldif2db** script.



WARNING

This parameter should only be used by very advanced users.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 to 8
Default Value	0
Syntax	Integer
Example	nsslapd-db-idx-divisor: 2

4.4.2.10. nsslapd-db-locks

Lock mechanisms in Directory Server control how many copies of Directory Server processes can run at the same time. The **nsslapd-db-locks** parameter sets the maximum number of locks.

Only set this parameter to a higher value if Directory Server runs out of locks and logs **libdb: Lock table is out of available locks** error messages. If you set a higher value without a need, this increases the size of the `/var/lib/dirsrv/slappd-instance_name/db_db.*` files without any benefit. For more information about monitoring the logs and determining a realistic value, see the corresponding section in the *Directory Server Performance Tuning Guide*.

The service must be restarted for changes to this attribute to take effect.

Parameter	Description
Entry DN	<code>cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config</code>
Valid Range	0 - 2147483647
Default Value	10000
Syntax	Integer
Example	<code>nsslapd-db-locks: 10000</code>

4.4.2.11. nsslapd-db-locks-monitoring-enable

Running out of database locks can lead to data corruption. With the **nsslapd-db-locks-monitoring-enable** parameter, you can enable or disable database lock monitoring. If the parameter is enabled, which is the default, Directory Server terminates all searches if the number of active database locks is higher than the percentage threshold configured in **nsslapd-db-locks-monitoring-threshold**. If an issue occurs, the administrator can increase the number of database locks in the **nsslapd-db-locks** parameter.

Restart the service for changes to this attribute to take effect.

Parameter	Description
Entry DN	<code>cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config</code>
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	<code>nsslapd-db-locks-monitoring-enable: on</code>

4.4.2.12. nsslapd-db-locks-monitoring-pause

If monitoring of database locks is enabled in the [nsslapd-db-locks-monitoring-enable](#) parameter, **nsslapd-db-locks-monitoring-pause** defines the interval in milliseconds that the monitoring thread sleeps between the checks.

If you set this parameter to a too high value, the server can run out of database locks before the monitoring check happens. However, setting a too low value can slow down the server.

You do not have to restart the server for this setting to take effect.

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	0 - 2147483647 (value in milliseconds)
Default Value	500
Syntax	DirectoryString
Example	nsslapd-db-locks-monitoring-pause: 500

4.4.2.13. nsslapd-db-locks-monitoring-threshold

If monitoring of database locks is enabled in the [nsslapd-db-locks-monitoring-enable](#) parameter, **nsslapd-db-locks-monitoring-threshold** sets the maximum percentage of used database locks before Directory Server terminates searches to avoid further lock exhaustion.

Restart the service for changes to this attribute to take effect.

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	70 - 95
Default Value	90
Syntax	DirectoryString
Example	nsslapd-db-locks-monitoring-threshold: 90

4.4.2.14. nsslapd-db-logbuf-size

This attribute specifies the log information buffer size. Log information is stored in memory until the buffer fills up or the transaction commit forces the buffer to be written to disk. Larger buffer sizes can significantly increase throughput in the presence of long running transactions, highly concurrent applications, or transactions producing large amounts of data. The log information buffer size is the transaction log size divided by four.

The **nsslapd-db-logbuf-size** attribute is only valid if the **nsslapd-db-durable-transactions** attribute is set to **on**.

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	32K to maximum 32-bit integer (limited to the amount of memory available on the machine)
Default Value	32K
Syntax	Integer
Example	nsslapd-db-logbuf-size: 32K

4.4.2.15. nsslapd-db-logdirectory

This attribute specifies the path to the directory that contains the database transaction log. The database transaction log contains a sequential listing of all recent database operations. Directory Server uses this information to recover the database after an instance shut down unexpectedly.

By default, the database transaction log is stored in the same directory as the directory database. To update this parameter, you must manually update the **/etc/dirsrv/slappd-instance_name/dse.ldif** file. For details, see the *Changing the Transaction Log Directory* section in the [Red Hat Directory Server Administration Guide](#).

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	Any valid path
Default Value	
Syntax	DirectoryString
Example	nsslapd-db-logdirectory: /var/lib/dirsrv/slappd-instance_name/db/

4.4.2.16. nsslapd-db-logfile-size

This attribute specifies the maximum size of a single file in the log in bytes. By default, or if the value is set to **0**, a maximum size of 10 megabytes is used. The maximum size is an unsigned 4-byte value.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 to unsigned 4-byte integer
Default Value	10MB
Syntax	Integer
Example	nsslapd-db-logfile-size: 10 MB

4.4.2.17. nsslapd-db-page-size

This attribute specifies the size of the pages used to hold items in the database in bytes. The minimum size is 512 bytes, and the maximum size is 64 kilobytes. If the page size is not explicitly set, Directory Server defaults to a page size of 8 kilobytes. Changing this default value can have a significant performance impact. If the page size is too small, it results in extensive page splitting and copying, whereas if the page size is too large it can waste disk space.

Before modifying the value of this attribute, export all databases using the **db2ldif** script. Once the modification has been made, reload the databases using the **ldif2db** script.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	512 bytes to 64 kilobytes
Default Value	8KB
Syntax	Integer
Example	nsslapd-db-page-size: 8KB

4.4.2.18. nsslapd-db-spin-count

This attribute specifies the number of times that test-and-set mutexes should spin without blocking.



WARNING

Never touch this value unless you are very familiar with the inner workings of Berkeley DB or are specifically told to do so by Red Hat support.

The default value of **0** causes BDB to calculate the actual value by multiplying the number of available CPU cores (as reported by the **nproc** utility or the **sysconf(_SC_NPROCESSORS_ONLN)** call) by **50**. For example, with a processor with 8 logical cores, leaving this attribute set to **0** is equivalent to setting it to **400**. It is not possible to turn spinning off entirely – if you want to minimize the amount of times test-and-set mutexes will spin without blocking, set this attribute to **1**.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 to 2147483647 (2^31-1)
Default Value	0
Syntax	Integer
Example	nsslapd-db-spin-count: 0

4.4.2.19. nsslapd-db-transaction-batch-max-wait

If [Section 4.4.2.21, “nsslapd-db-transaction-batch-val”](#) is set, the flushing of transactions is done by a separate thread when the set batch value is reached. However if there are only a few updates, this process might take too long. This parameter controls when transactions should be flushed latest, independently of the batch count. The values is defined in milliseconds.



WARNING

This parameter is experimental. *Never change its value unless you are specifically told to do so by the Red Hat support.*

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 - 2147483647 (value in milliseconds)
Default Value	50
Syntax	Integer
Example	nsslapd-db-transaction-batch-max-wait: 50

4.4.2.20. nsslapd-db-transaction-batch-min-wait

If [Section 4.4.2.21, “nsslapd-db-transaction-batch-val”](#) is set, the flushing of transactions is done by a separate thread when the set batch value is reached. However if there are only a few updates, this process might take too long. This parameter controls when transactions should be flushed earliest, independently of the batch count. The values is defined in milliseconds.



WARNING

This parameter is experimental. *Never change its value unless you are specifically told to do so by the Red Hat support.*

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 - 2147483647 (value in milliseconds)
Default Value	50
Syntax	Integer
Example	nsslapd-db-transaction-batch-min-wait: 50

4.4.2.21. nsslapd-db-transaction-batch-val

This attribute specifies how many transactions will be batched before being committed. This attribute can improve update performance when full transaction durability is not required. This attribute can be dynamically modified using **ldapmodify**. For further information on modifying this attribute, see the “Tuning Directory Server Performance” chapter in the *Red Hat Directory Server Administration Guide*.



WARNING

Setting this value will reduce data consistency and may lead to loss of data. This is because if there is a power outage before the server can flush the batched transactions, those transactions in the batch will be lost.

Do not set this value unless specifically requested to do so by Red Hat support.

If this attribute is not defined or is set to a value of **0**, transaction batching will be turned off, and it will be impossible to make remote modifications to this attribute using LDAP. However, setting this attribute to a value greater than **0** causes the server to delay committing transactions until the number of queued transactions is equal to the attribute value. A value greater than **0** also allows modifications to this

attribute remotely using LDAP. A value of **1** for this attribute allows modifications to the attribute setting remotely using LDAP, but results in no batching behavior. A value of **1** at server startup is therefore useful for maintaining normal durability while also allowing transaction batching to be turned on and off remotely when required. Remember that the value for this attribute may require modifying the **nsslapd-db-logbuf-size** attribute to ensure sufficient log buffer size for accommodating the batched transactions.



NOTE

The **nsslapd-db-transaction-batch-val** attribute is only valid if the **nsslapd-db-durable-transaction** attribute is set to **on**.

For more information on database transaction logging, see the "Monitoring Server and Database Activity" chapter in the *Red Hat Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 to 30
Default Value	0 (or turned off)
Syntax	Integer
Example	nsslapd-db-transaction-batch-val: 5

4.4.2.22. nsslapd-db-trickle-percentage

This attribute sets that at least the specified percentage of pages in the shared-memory pool are clean by writing dirty pages to their backing files. This is to ensure that a page is always available for reading in new information without having to wait for a write.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	0 to 100
Default Value	40
Syntax	Integer
Example	nsslapd-db-trickle-percentage: 40

4.4.2.23. nsslapd-db-verbose

This attribute specifies whether to record additional informational and debugging messages when searching the log for checkpoints, doing deadlock detection, and performing recovery. This parameter is meant for troubleshooting, and enabling the parameter may slow down the Directory Server.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-db-verbose: off

4.4.2.24. nsslapd-import-cache-autosize

This performance tuning-related attribute automatically sets the size of the import cache (**importCache**) to be used during the command-line-based import process of LDIF files to the database (the **Idif2db** operation).

In Directory Server, the import operation can be run as a server task or exclusively on the command-line. In the task mode, the import operation runs as a general Directory Server operation. The **nsslapd-import-cache-autosize** attribute enables the import cache to be set automatically to a predetermined size when the import operation is run on the command-line. The attribute can also be used by Directory Server during the task mode import for allocating a specified percentage of free memory for import cache.

By default, the **nsslapd-import-cache-autosize** attribute is enabled and is set to a value of **-1**. This value autosizes the import cache for the **Idif2db** operation only, automatically allocating fifty percent (50%) of the free physical memory for the import cache. The percentage value (50%) is hard-coded and cannot be changed.

Setting the attribute value to **50 (nsslapd-import-cache-autosize: 50)** has the same effect on performance during an **Idif2db** operation. However, such a setting will have the same effect on performance when the import operation is run as a Directory Server task. The **-1** value autosizes the import cache just for the **Idif2db** operation and not for any, including import, general Directory Server tasks.



NOTE

The purpose of a **-1** setting is to enable the **Idif2db** operation to benefit from free physical memory but, at the same time, not compete for valuable memory with the entry cache, which is used for general operations of the Directory Server.

Setting the **nsslapd-import-cache-autosize** attribute value to **0** turns off the import cache autosizing feature – that is, no autosizing occurs during either mode of the import operation. Instead, Directory Server uses the **nsslapd-import-cachesize** attribute for import cache size, with a default value of **20000000**.

There are three caches in the context of Directory Server: database cache, entry cache, and import

cache. The import cache is only used during the import operation. The **nsslapd-cache-autosize** attribute, which is used for autosizing the entry cache and database cache, is used during the Directory Server operations only and not during the **ldif2db** command-line operation; the attribute value is the percentage of free physical memory to be allocated for the entry cache and database cache.

If both the autosizing attributes, **nsslapd-cache-autosize** and **nsslapd-import-cache-autosize**, are enabled, ensure that their sum is less than 100.

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	-1, 0 (turns import cache autosizing off) to 100
Default Value	-1 (turns import cache autosizing on for ldif2db only and allocates 50% of the free physical memory to import cache)
Syntax	Integer
Example	nsslapd-import-cache-autosize: -1

4.4.2.25. nsslapd-db cachesize

This performance tuning-related attribute specifies the database index cache size, in bytes. This is one of the most important values for controlling how much physical RAM the directory server uses.

This is not the entry cache. This is the amount of memory the Berkeley database back end will use to cache the indexes (the **.db** files) and other files. This value is passed to the Berkeley DB API function **set_cachesize**. If automatic cache resizing is activated, this attribute is overridden when the server replaces these values with its own guessed values at a later stage of the server startup.

For more technical information on this attribute, see the cache size section of the Berkeley DB reference guide at
https://docs.oracle.com/cd/E17076_04/html/programmer_reference/general_am_conf.html#am_conf_ca

Attempting to set a value that is not a number or is too big for a 32-bit signed integer returns an **LDAP_UNWILLING_TO_PERFORM** error message with additional error information explaining the problem.



NOTE

Do not set the database cache size manually. Red Hat recommends to use the database cache auto-sizing feature for optimized performance. For further see the corresponding section in the [Red Hat Directory Server Performance Tuning Guide](#).

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=bdb,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Range	500 kilobytes to 4 gigabytes for 32-bit platforms and 500 kilobytes to $2^{64}-1$ for 64-bit platforms
Default Value	
Syntax	Integer
Example	nsslapd-dbcachesize: 10000000

4.4.2.26. nsslapd-dbncache

This attribute can split the LDBM cache into equally sized separate pieces of memory. It is possible to specify caches that are large enough so that they cannot be allocated contiguously on some architectures; for example, some systems limit the amount of memory that may be allocated contiguously by a process. If **nsslapd-dbncache** is **0** or **1**, the cache will be allocated contiguously in memory. If it is greater than **1**, the cache will be broken up into **ncache**, equally sized separate pieces of memory.

To configure a dbcache size larger than 4 gigabytes, add the **nsslapd-dbncache** attribute to **cn=config,cn=ldbm database,cn=plugins,cn=config** between the **nsslapd-dbcachesize** and **nsslapd-db-logdirectory** attribute lines.

Set this value to an integer that is one-quarter (1/4) the amount of memory in gigabytes. For example, for a 12 gigabyte system, set the **nsslapd-dbncache** value to **3**; for an 8 gigabyte system, set it to **2**.

This attribute is provided only for system modification/diagnostics and should be changed only with the guidance of Red Hat technical support or Red Hat professional services. Inconsistent settings of this attribute and other configuration attributes may cause the Directory Server to be unstable.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	1 to 4
Default Value	1
Syntax	Integer
Example	nsslapd-dbncache: 1

4.4.2.27. nsslapd-search-bypass-filter-test

If you enable the **nsslapd-search-bypass-filter-test** parameter, Directory Server bypasses filter checks when it builds candidate lists during a search. If you set the parameter to **verify**, Directory Server evaluates the filter against the search candidate entries.

Parameter	Description
Entry DN	cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	on off verify
Default Value	on
Syntax	Directory String
Example	nsslapd-search-bypass-filter-test: on

4.4.3. Database Attributes under **cn=monitor,cn=ldbm database,cn=plugins,cn=config**

Global read-only attributes containing database statistics for monitoring activity on the databases are stored in the **cn=monitor,cn=ldbm database,cn=plugins,cn=config** tree node. For more information on these entries, see the "Monitoring Server and Database Activity" chapter in the *Red Hat Directory Server Administration Guide*.

dbcachehits

This attribute shows the requested pages found in the database.

dbcachetries

This attribute shows the total cache lookups.

dbcachehitratio

This attribute shows the percentage of requested pages found in the database cache (hits/tries).

dbcachepagein

This attribute shows the pages read into the database cache.

dbcachepageout

This attribute shows the pages written from the database cache to the backing file.

dbcacheroevict

This attribute shows the clean pages forced from the cache.

dbcacherwevict

This attribute shows the dirty pages forced from the cache.

normalizedDNcachetries

Total number of cache lookups since the instance was started.

normalizedDNcachehits

Normalized DNs found within the cache.

normalizedDNcachemisses

Normalized DNs not found within the cache.

normalizedDNcachehitratio

Percentage of the normalized DNs found in the cache.

currentNormalizedDNCachesize

Current size of the normalized DN cache in bytes.

maxNormalizedDNCachesize

Current value of the **nsslapd-ndn-cache-max-size** parameter. For details how to update this setting, see [Section 3.1.1.128, “nsslapd-ndn-cache-max-size”](#).

currentNormalizedDNCacheCount

Number of normalized cached DNs.

4.4.4. Database Attributes under *cn=database_name,cn=ldbmcn=plugins,cn=config*

The **cn=database_name** subtree contains all the configuration data for the user-defined database.

The **cn=userRoot** subtree is called *userRoot* by default. However, this is not hard-coded and, given the fact that there are going to be multiple database instances, this name is changed and defined by the user as and when new databases are added. The **cn=userRoot** database referenced can be any user database.

The following attributes are common to databases, such as **cn=userRoot**.

4.4.4.1. nsslapd-cachesize

This attribute has been deprecated. To resize the entry cache, use nsslapd-cachememsize.

This performance tuning-related attribute specifies the cache size in terms of the number of entries it can hold. However, this attribute is deprecated in favor of the **nsslapd-cachememsize** attribute, which sets an absolute allocation of RAM for the entry cache size, as described in [Section 4.4.4.2, “nsslapd-cachememsize”](#).

Attempting to set a value that is not a number or is too big for a 32-bit signed integer (on 32-bit systems) returns an **LDAP_UNWILLING_TO_PERFORM** error message with additional error information explaining the problem.

The server has to be restarted for changes to this attribute to go into effect.

**NOTE**

The performance counter for this setting goes to the highest 64-bit integer, even on 32-bit systems, but the setting itself is limited on 32-bit systems to the highest 32-bit integer because of how the system addresses memory.

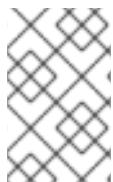
Parameter	Description
Entry DN	<code>cn=database_name,cn=ldbm database,cn=plugins,cn=config</code>
Valid Range	1 to $2^{32}-1$ on 32-bit systems or $2^{63}-1$ on 64-bit systems or -1, which means limitless
Default Value	-1
Syntax	Integer
Example	<code>nsslapd-cachesize: -1</code>

4.4.4.2. nsslapd-cachememsize

This performance tuning-related attribute specifies the size, in bytes, for the available memory space for the entry cache. The simplest method is limiting cache size in terms of memory occupied. Activating automatic cache resizing overrides this attribute, replacing these values with its own guessed values at a later stage of the server startup.

Attempting to set a value that is not a number or is too big for a 32-bit signed integer (on 32-bit systems) returns an **LDAP_UNWILLING_TO_PERFORM** error message with additional error information explaining the problem.

The performance counter for this setting goes to the highest 64-bit integer, even on 32-bit systems, but the setting itself is limited on 32-bit systems to the highest 32-bit integer because of how the system addresses memory.



NOTE

Do not set the database cache size manually. Red Hat recommends to use the entry cache auto-sizing feature for optimized performance. For further see the corresponding section in the [Red Hat Directory Server Performance Tuning Guide](#).

Parameter	Description
Entry DN	<code>cn=database_name,cn=ldbm database,cn=plugins,cn=config</code>
Valid Range	500 kilobytes to $2^{64}-1$ on 64-bit systems
Default Value	209715200 (200 MiB)
Syntax	Integer
Example	<code>nsslapd-cachememsize: 209715200</code>

4.4.4.3. nsslapd-directory

This attribute specifies the path to the database instance. If it is a relative path, it starts from the path specified by **nsslapd-directory** in the global database entry **cn=config,cn=ldbm database,cn=plugins,cn=config**. The database instance directory is named after the instance name and located in the global database directory, by default. After the database instance has been created, do not modify this path, because any changes risk preventing the server from accessing data.

Parameter	Description
Entry DN	<code>cn=database_name,cn=ldbm database,cn=plugins,cn=config</code>
Valid Values	Any valid path to the database instance
Default Value	
Syntax	DirectoryString
Example	<code>nsslapd-directory: /var/lib/dirsrv/slapd-instance/db/userRoot</code>

4.4.4.4. nsslapd-dncachememsize

This performance tuning-related attribute specifies the size, in bytes, for the available memory space for the DN cache. The DN cache is similar to the entry cache for a database, only its table stores only the entry ID and the entry DN. This allows faster lookups for rename and moddn operations.

The simplest method is limiting cache size in terms of memory occupied.

Attempting to set a value that is not a number or is too big for a 32-bit signed integer (on 32-bit systems) returns an **LDAP_UNWILLING_TO_PERFORM** error message with additional error information explaining the problem.



NOTE

The performance counter for this setting goes to the highest 64-bit integer, even on 32-bit systems, but the setting itself is limited on 32-bit systems to the highest 32-bit integer because of how the system addresses memory.

Parameter	Description
Entry DN	<code>cn=database_name,cn=ldbm database,cn=plugins,cn=config</code>
Valid Range	500 kilobytes to $2^{32}-1$ on 32-bit systems and to $2^{64}-1$ on 64-bit systems
Default Value	10,485,760 (10 megabytes)

Parameter	Description
Syntax	Integer
Example	nsslapd-dncachememsize: 10485760

4.4.4.5. nsslapd-readonly

This attribute specifies read-only mode for a single back-end instance. If this attribute has a value of **off**, then users have all read, write, and execute permissions allowed by their access permissions.

Parameter	Description
Entry DN	cn=database_name,cn=ldbmcn=plugins,cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-readonly: off

4.4.4.6. nsslapd-require-index

When switched to **on**, this attribute allows one to refuse unindexed searches. This performance-related attribute avoids saturating the server with erroneous searches.

Parameter	Description
Entry DN	cn=database_name,cn=ldbmcn=plugins,cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsslapd-require-index: off

4.4.4.7. nsslapd-require-internalop-index

When a plug-in modifies data, it has a write lock on the database. On large databases, if a plug-in then executes an unindexed search, the plug-in can use all database locks and corrupt the database or the

server becomes unresponsive. To avoid this problem, you can reject internal unindexed searches by enabling the **nsslapd-require-internalop-index** parameter.

Parameter	Description
Entry DN	<code>cn=database_name,cn=ldbmdatabase,cn=plugins,cn=config</code>
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	<code>nsslapd-require-internalop-index: off</code>

4.4.4.8. nsslapd-suffix

This attribute specifies the suffix of the *database link*. This is a single-valued attribute because each database instance can have only one suffix. Previously, it was possible to have more than one suffix on a single database instance, but this is no longer the case. As a result, this attribute is single-valued to enforce the fact that each database instance can only have one suffix entry. Any changes made to this attribute after the entry has been created take effect only after the server containing the database link is restarted.

Parameter	Description
Entry DN	<code>cn=database_name,cn=ldbmdatabase,cn=plugins,cn=config</code>
Valid Values	Any valid DN
Default Value	
Syntax	DirectoryString
Example	<code>nsslapd-suffix: o=Example</code>

4.4.4.9. vlvBase

This attribute sets the base DN for which the browsing or virtual list view (VLV) index is created.

For more information on VLV indexes, see the indexing chapter in the *Administration Guide*.

Parameter	Description
Entry DN	<code>cn=index_name,cn=userRoot,cn=ldbmdatabase,cn=plugins,cn=config</code>

Parameter	Description
Valid Values	Any valid DN
Default Value	
Syntax	DirectoryString
Example	vlvBase: ou=People,dc=example,dc=com

4.4.4.10. vlvEnabled

The **vlvEnabled** attribute provides status information about a specific VLV index, and Directory Server sets this attribute at run time. Although **vlvEnabled** is shown in the configuration, you cannot modify this attribute.

For more information on VLV indexes, see the indexing chapter in the *Administration Guide*.

Parameter	Description
Entry DN	<code>cn=index_name,cn=userRoot,cn=ldbm database,cn=plugins,cn=config</code>
Valid Values	0 (disabled) 1 (enabled)
Default Value	1
Syntax	DirectoryString
Example	vlvEnabled: 0

4.4.4.11. vlvFilter

The browsing or virtual list view (VLV) index is created by running a search according to a filter and including entries which match that filter in the index. The filter is specified in the **vlvFilter** attribute.

For more information on VLV indexes, see the indexing chapter in the *Administration Guide*.

Parameter	Description
Entry DN	<code>cn=index_name,cn=userRoot,cn=ldbm database,cn=plugins,cn=config</code>
Valid Values	Any valid LDAP filter
Default Value	

Parameter	Description
Syntax	DirectoryString
Example	vlvFilter: (

4.4.4.12. vlvIndex (Object Class)

A *browsing index* or *virtual list view (VLV) index* dynamically generates an abbreviated index of entry headers that makes it much faster to visually browse large indexes. A VLV index definition has two parts: one which defines the index and one which defines the search used to identify entries to add to the index. The **vlvIndex** object class defines the index entry.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.42

Table 4.2. Required Attributes

Attribute	Definition
objectClass	Defines the object classes for the entry.
cn	Gives the common name of the entry.
Section 4.4.4.15, "vlvSort"	Identifies the attribute list that the browsing index (virtual list view index) is sorted on.

Table 4.3. Allowed Attributes

Attribute	Definition
Section 4.4.4.10, "vlvEnabled"	Stores the availability of the browsing index.
Section 4.4.4.16, "vlvUses"	Contains the count the browsing index is used.

4.4.4.13. vlvScope

This attribute sets the scope of the search to run for entries in the browsing or virtual list view (VLV) index.

For more information on VLV indexes, see the indexing chapter in the *Administration Guide*.

Parameter	Description
Entry DN	<code>cn=index_name,cn=userRoot,cn=ldbmcn=plugins,cn=config</code>
Valid Values	* 1 (one-level or children search) * 2 (subtree search)
Default Value	
Syntax	Integer
Example	<code>vlvScope: 2</code>

4.4.4.14. vlvSearch (Object Class)

A *browsing index* or *virtual list view (VLV) index* dynamically generates an abbreviated index of entry headers that makes it much faster to visually browse large indexes. A VLV index definition has two parts: one which defines the index and one which defines the search used to identify entries to add to the index. The **vlvSearch** object class defines the search filter entry.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.38

Table 4.4. Required Attributes

Attribute	Definition
<code>objectClass</code>	Defines the object classes for the entry.
Section 4.4.4.9, "vlvBase"	Identifies base DN the browsing index is created.
Section 4.4.4.13, "vlvScope"	Identifies the scope to define the browsing index.
Section 4.4.4.11, "vlvFilter"	Identifies the filter string to define the browsing index.

Table 4.5. Allowed Attributes

Attribute	Definition
<code>multiLineDescription</code>	Gives a text description of the entry.

4.4.4.15. vlvSort

This attribute sets the sort order for returned entries in the browsing or virtual list view (VLV) index.



NOTE

The entry for this attribute is a **vlvIndex** entry beneath the **vlvSearch** entry.

For more information on VLV indexes, see the indexing chapter in the *Administration Guide*.

Parameter	Description
Entry DN	<code>cn=index_name,cn=index_name,cn=userRoot,cn=ldbm database,cn=plugins,cn=config</code>
Valid Values	Any Directory Server attributes, in a space-separated list
Default Value	
Syntax	DirectoryString
Example	<code>vlvSort: cn givenName o ou sn</code>

4.4.4.16. vlvUses

The **vlvUses** attribute contains the count the browsing index uses, and Directory Server sets this attribute at run time. Although **vlvUses** is shown in the configuration, you cannot modify this attribute.

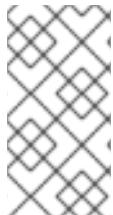
For more information on VLV indexes, see the indexing chapter in the *Administration Guide*.

Parameter	Description
Entry DN	<code>cn=index_name,cn=userRoot,cn=ldbm database,cn=plugins,cn=config</code>
Valid Values	N/A
Default Value	
Syntax	DirectoryString
Example	<code>vlvUses: 800</code>

4.4.5. Database Attributes under `cn=database,cn=monitor,cn=ldbm database,cn=plugins,cn=config`

The attributes in this tree node entry are all read-only, database performance counters. All of the values for these attributes are 32-bit integers, except for **entrycachehits** and **entrycachetries**.

If the **nsslapd-counters** attribute in **cn=config** is set to **on**, then some of the counters kept by the Directory Server instance increment using 64-bit integers, even on 32-bit machines or with a 32-bit version of Directory Server. For the database monitoring, the **entrycachehits** and **entrycachetries** counters use 64-bit integers.



NOTE

The **nsslapd-counters** attribute enables 64-bit support for these specific database and server counters. The counters which use 64-bit integers are not configurable; the 64-bit integers are either enabled for all the allowed counters or disabled for all allowed counters.

nsslapd-db-abort-rate

This attribute shows the number of transactions that have been aborted.

nsslapd-db-active-txns

This attribute shows the number of transactions that are currently active.

nsslapd-db-cache-hit

This attribute shows the requested pages found in the cache.

nsslapd-db-cache-try

This attribute shows the total cache lookups.

nsslapd-db-cache-region-wait-rate

This attribute shows the number of times that a thread of control was forced to wait before obtaining the region lock.

nsslapd-db-cache-size-bytes

This attribute shows the total cache size in bytes.

nsslapd-db-clean-pages

This attribute shows the clean pages currently in the cache.

nsslapd-db-commit-rate

This attribute shows the number of transactions that have been committed.

nsslapd-db-deadlock-rate

This attribute shows the number of deadlocks detected.

nsslapd-db-dirty-pages

This attribute shows the dirty pages currently in the cache.

nsslapd-db-hash-buckets

This attribute shows the number of hash buckets in buffer hash table.

nsslapd-db-hash-elements-examine-rate

This attribute shows the total number of hash elements traversed during hash table lookups.

nsslapd-db-hash-search-rate

This attribute shows the total number of buffer hash table lookups.

nsslapd-db-lock-conflicts

This attribute shows the total number of locks not immediately available due to conflicts.

nsslapd-db-lock-region-wait-rate

This attribute shows the number of times that a thread of control was forced to wait before obtaining the region lock.

nsslapd-db-lock-request-rate

This attribute shows the total number of locks requested.

nsslapd-db-lockers

This attribute shows the number of current lockers.

nsslapd-db-log-bytes-since-checkpoint

This attribute shows the number of bytes written to this log since the last checkpoint.

nsslapd-db-log-region-wait-rate

This attribute shows the number of times that a thread of control was forced to wait before obtaining the region lock.

nsslapd-db-log-write-rate

This attribute shows the number of megabytes and bytes written to this log.

nsslapd-db-longest-chain-length

This attribute shows the longest chain ever encountered in buffer hash table lookups.

nsslapd-db-page-create-rate

This attribute shows the pages created in the cache.

nsslapd-db-page-read-rate

This attribute shows the pages read into the cache.

nsslapd-db-page-ro-evict-rate

This attribute shows the clean pages forced from the cache.

nsslapd-db-page-rw-evict-rate

This attribute shows the dirty pages forced from the cache.

nsslapd-db-page-trickle-rate

This attribute shows the dirty pages written using the **memp_trickle** interface.

nsslapd-db-page-write-rate

This attribute shows the pages read into the cache.

nsslapd-db-pages-in-use

This attribute shows all pages, clean or dirty, currently in use.

nsslapd-db-txn-region-wait-rate

This attribute shows the number of times that a thread of control was forced to wait before obtaining the region lock.

currentdncachecount

This attribute shows the number of DNs currently present in the DN cache.

currentdncachesize

This attribute shows the total size, in bytes, of DNs currently present in the DN cache.

maxdncachesize

This attribute shows the maximum size, in bytes, of DNs that can be maintained in the database DN cache.

4.4.6. Database Attributes under **cn=monitor,cn=userRoot,cn=ldbm database,cn=plugins,cn=config**

The attributes in this tree node entry are all read-only, database performance counters.

If the **nsslapd-counters** attribute in **cn=config** is set to **on**, then some of the counters kept by the Directory Server instance increment using 64-bit integers, even on 32-bit machines or with a 32-bit version of Directory Server. For database monitoring, the **entrycachehits** and **entrycachetries** counters use 64-bit integers.



NOTE

The **nsslapd-counters** attribute enables 64-bit support for these specific database and server counters. The counters which use 64-bit integers are not configurable; the 64-bit integers are either enabled for all the allowed counters or disabled for all allowed counters.

dbfilename-number

This attribute gives the name of the file and provides a sequential integer identifier (starting at 0) for the file. All associated statistics for the file are given this same numerical identifier.

dbfilecachehit-number

This attribute gives the number of times that a search requiring data from this file was performed and that the data were successfully obtained from the cache. The number in this attribute's name corresponds to the one in **dbfilename**.

dbfilecachemiss-number

This attribute gives the number of times that a search requiring data from this file was performed and that the data could not be obtained from the cache. The number in this attribute's name corresponds to the one in **dbfilename**.

dbfilepagein-number

This attribute gives the number of pages brought to the cache from this file. The number in this attributes name corresponds to the one in **dbfilename**.

dbfilepageout-number

This attribute gives the number of pages for this file written from cache to disk. The number in this attributes name corresponds to the one in **dbfilename**.

currentDNCacheCount

Number of cached DNs.

currentDNCacheSize

Current size of the DN cache in bytes.

DNCacheHitRatio

Percentage of the DNs found in the cache.

DNCacheHits

DNs found within the cache.

DNCacheMisses

DNs not found within the cache.

DNCacheTries

Total number of cache lookups since the instance was started.

maxDNCacheSize

Current value of the **nsslapd-ndn-cache-max-size** parameter. For details how to update this setting, see [Section 3.1.1.128, “nsslapd-ndn-cache-max-size”](#).

4.4.7. Database Attributes under cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config

The set of default indexes is stored here. Default indexes are configured per back end in order to optimize Directory Server functionality for the majority of setup scenarios. All indexes, except system-essential ones, can be removed, but care should be taken so as not to cause unnecessary disruptions. For further information on indexes, see the "Managing Indexes" chapter in the *Red Hat Directory Server Administration Guide*.

4.4.7.1. cn

This attribute provides the name of the attribute to index.

Parameter	Description
Entry DN	cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config

Parameter	Description
Valid Values	Any valid index cn
Default Value	None
Syntax	DirectoryString
Example	cn: aci

4.4.7.2. nsIndex

This object class defines an index in the back end database. This object is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.44

Table 4.6. Required Attributes

Attribute	Definition
objectClass	Defines the object classes for the entry.
cn	Gives the common name of the entry.
Section 4.4.7.5, "nsSystemIndex"	Identify whether or not the index is a system defined index.

Table 4.7. Allowed Attributes

Attribute	Definition
description	Gives a text description of the entry.
Section 4.4.7.3, "nsIndexType"	Identifies the index type.
Section 4.4.7.4, "nsMatchingRule"	Identifies the matching rule.

4.4.7.3. nsIndexType

This optional, multi-valued attribute specifies the type of index for Directory Server operations and takes the values of the attributes to be indexed. Each required index type has to be entered on a separate line.

Parameter	Description
Entry DN	cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	<ul style="list-style-type: none"> * pres = presence index * eq = equality index * approx = approximate index * sub = substring index * matching rule = international index * index browse = browsing index
Default Value	
Syntax	DirectoryString
Example	nsIndexType: eq

4.4.7.4. nsMatchingRule

This optional, multi-valued attribute specifies the ordering matching rule name or OID used to match values and to generate index keys for the attribute. This is most commonly used to ensure that equality and range searches work correctly for languages other than English (7-bit ASCII).

This is also used to allow range searches to work correctly for integer syntax attributes that do not specify an ordering matching rule in their schema definition. **uidNumber** and **gidNumber** are two commonly used attributes that fall into this category.

For example, for a **uidNumber** that uses integer syntax, the rule attribute could be **nsMatchingRule: integerOrderingMatch**.



NOTE

Any change to this attribute will not take effect until the change is saved and the index is rebuilt using **db2index**, which is described in more detail in the "Managing Indexes" chapter of the *Red Hat Directory Server Administration Guide*).

Parameter	Description
Entry DN	cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	Any valid collation order object identifier (OID)
Default Value	None

Parameter	Description
Syntax	DirectoryString
Example	nsMatchingRule: 2.16.840.1.113730.3.3.2.3.1 (For Bulgarian)

4.4.7.5. nsSystemIndex

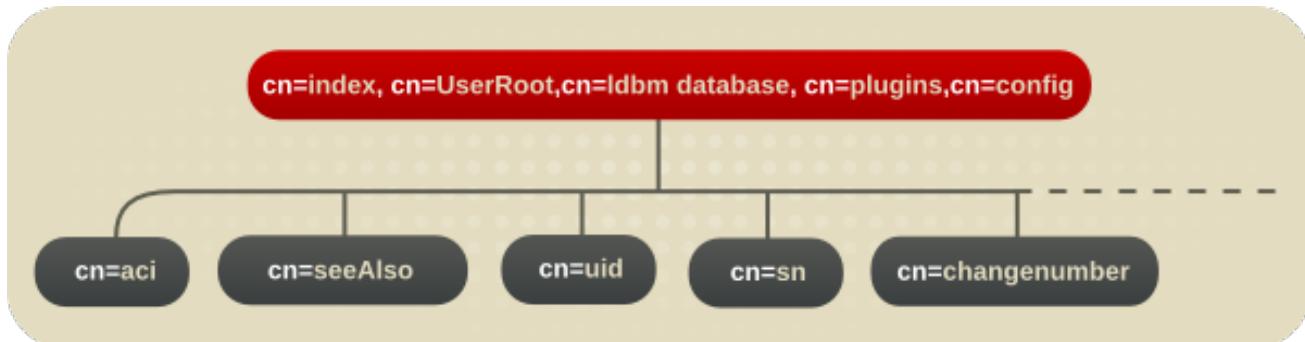
This mandatory attribute specifies whether the index is a *system index*, an index which is vital for Directory Server operations. If this attribute has a value of **true**, then it is system-essential. System indexes should not be removed, as this will seriously disrupt server functionality.

Parameter	Description
Entry DN	cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config
Valid Values	true false
Default Value	
Syntax	DirectoryString
Example	nssystemindex: true

4.4.8. Database Attributes under **cn=index,cn=database_name,cn=ldbm database,cn=plugins,cn=config**

In addition to the set of default indexes that are stored under **cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config**, custom indexes can be created for user-defined back end instances; these are stored under **cn=index,cn=database_name,cn=ldbm database,cn=plugins,cn=config**. Each indexed attribute represents a subentry under the **cn=config** information tree nodes, as shown in the following diagram:

Figure 4.2. Indexed Attribute Representing a Subentry



For example, the index file for the **aci** attribute under **o=UserRoot** appears in the Directory Server as follows:

```
dn:cn=aci,cn=index,cn=UserRoot,cn=ldbm database,cn=plugins,cn=config
objectclass:top
objectclass:nsIndex
cn:aci
nsSystemIndex:true
nsIndexType:pres
```

These entries share all of the indexing attributes listed for the default indexes in [Section 4.4.7, "Database Attributes under cn=default indexes,cn=config,cn=ldbm database,cn=plugins,cn=config"](#). For further information about indexes, see the "Managing Indexes" chapter in the *Red Hat Directory Server Administration Guide*.

4.4.8.1. nsIndexIDListScanLimit

This multi-valued parameter defines a search limit for certain indices or to use no ID list. For further information, see the corresponding section in the [Directory Server Performance Tuning Guide](#).

Parameter	Description
Entry DN	cn=attribute_name,cn=index,cn=database_name,cn=dbm database,cn=plugins,cn=config
Valid Values	See the corresponding section in the Directory Server Performance Tuning Guide .
Default Value	
Syntax	DirectoryString
Example	nsIndexIDListScanLimit: limit=0 type=eq values/inetorgperson

4.4.8.2. nsSubStrBegin

By default, for a search to be indexed, the search string must be at least three characters long, without counting any wildcard characters. For example, the string **abc** would be an indexed search while **ab*** would not be. Indexed searches are significantly faster than unindexed searches, so changing the minimum length of the search key is helpful to increase the number of indexed searches.

This substring length can be edited based on the position of any wildcard characters. The **nsSubStrBegin** attribute sets the required number of characters for an indexed search for the beginning of a search string, before the wildcard. For example:

```
abc*
```

If the value of this attribute is changed, then the index must be regenerated using **db2index**.

Parameter	Description

Parameter	Description
Entry DN	<code>cn=attribute_name,cn=index,cn=database_name,cn=dbm database,cn=plugins,cn=config</code>
Valid Values	Any integer
Default Value	3
Syntax	Integer
Example	<code>nsSubStrBegin: 2</code>

4.4.8.3. nsSubStrEnd

By default, for a search to be indexed, the search string must be at least three characters long, without counting any wildcard characters. For example, the string **abc** would be an indexed search while **ab*** would not be. Indexed searches are significantly faster than unindexed searches, so changing the minimum length of the search key is helpful to increase the number of indexed searches.

This substring length can be edited based on the position of any wildcard characters. The **nsSubStrEnd** attribute sets the required number of characters for an indexed search for the end of a search string, after the wildcard. For example:

`*xyz`

If the value of this attribute is changed, then the index must be regenerated using **db2index**.

Parameter	Description
Entry DN	<code>cn=attribute_name,cn=index,cn=database_name,cn=dbm database,cn=plugins,cn=config</code>
Valid Values	Any integer
Default Value	3
Syntax	Integer
Example	<code>nsSubStrEnd: 2</code>

4.4.8.4. nsSubStrMiddle

By default, for a search to be indexed, the search string must be at least three characters long, without counting any wildcard characters. For example, the string **abc** would be an indexed search while **ab*** would not be. Indexed searches are significantly faster than unindexed searches, so changing the minimum length of the search key is helpful to increase the number of indexed searches.

This substring length can be edited based on the position of any wildcard characters. The **nsSubStrMiddle** attribute sets the required number of characters for an indexed search where a wildcard is used in the middle of a search string. For example:

ab^{*}z

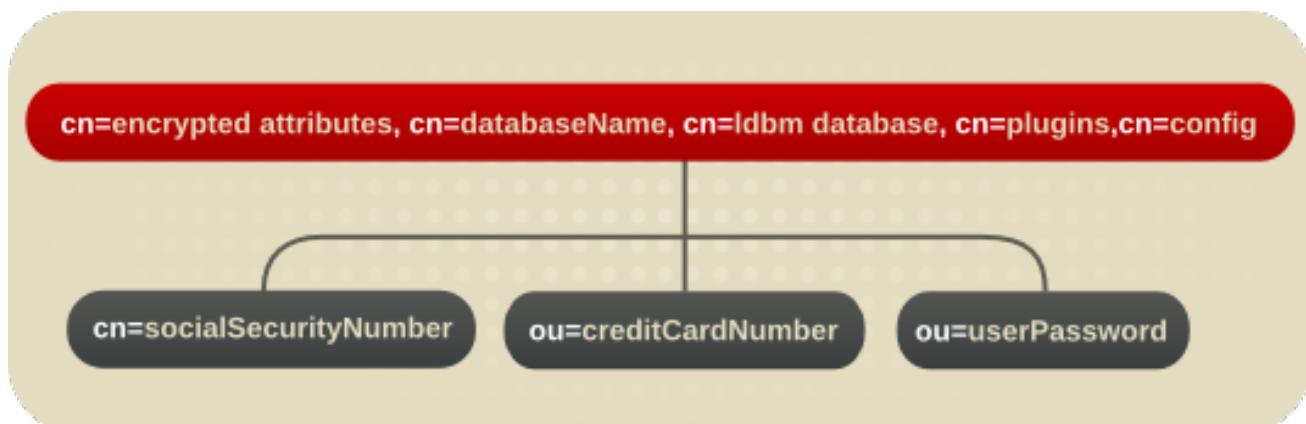
If the value of this attribute is changed, then the index must be regenerated using **db2Index**.

Parameter	Description
Entry DN	<code>cn=attribute_name,cn=index,cn=database_name,cn=dbm database,cn=plugins,cn=config</code>
Valid Values	Any integer
Default Value	3
Syntax	Integer
Example	<code>nsSubStrMiddle: 3</code>

4.4.9. Database Attributes under `cn=attributeName,cn=encrypted attributes,cn=database_name,cn=ldbm database,cn=plugins,cn=config`

The **nsAttributeEncryption** object class allows selective encryption of attributes within a database. Extremely sensitive information such as credit card numbers and government identification numbers may not be protected enough by routine access control measures. Normally, these attribute values are stored in CLEAR within the database; encrypting them while they are stored adds another layer of protection. This object class has one attribute, **nsEncryptionAlgorithm**, which sets the encryption cipher used per attribute. Each encrypted attribute represents a subentry under the above **cn=config** information tree nodes, as shown in the following diagram:

Figure 4.3. Encrypted Attributes under the `cn=config` Node



For example, the database encryption file for the **userPassword** attribute under **o=UserRoot** appears in the Directory Server as follows:

`dn:cn=userPassword,cn=encrypted attributes,o=UserRoot,cn=ldbm database,
cn=plugins,cn=config
objectclass:top`

```
objectclass:nsAttributeEncryption
cn:userPassword
nsEncryptionAlgorithm:AES
```

To configure database encryption, see the "Database Encryption" section of the "Configuring Directory Databases" chapter in the *Red Hat Directory Server Administration Guide*. For more information about indexes, see the "Managing Indexes" chapter in the *Red Hat Directory Server Administration Guide*.

4.4.9.1. nsAttributeEncryption (Object Class)

This object class is used for core configuration entries which identify and encrypt selected attributes within a Directory Server database.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.316

Table 4.8. Required Attributes

objectClass	Defines the object classes for the entry.
cn	Specifies the attribute being encrypted using its common name.
Section 4.4.9.2, "nsEncryptionAlgorithm"	The encryption cipher used.

4.4.9.2. nsEncryptionAlgorithm

nsEncryptionAlgorithm selects the cipher used by **nsAttributeEncryption**. The algorithm can be set per encrypted attribute.

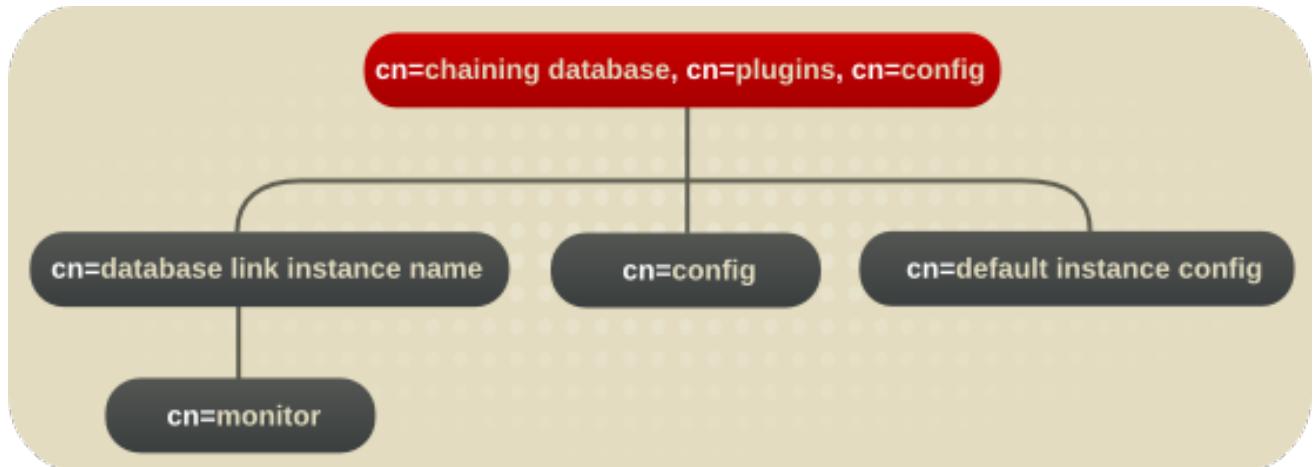
Parameter	Description
Entry DN	cn=attributeName,cn=encrypted attributes,cn=database_name,cn=ldbm database,cn=plugins,cn=config
Valid Values	The following are supported ciphers: * Advanced Encryption Standard Block Cipher (AES) * Triple Data Encryption Standard Block Cipher (3DES)
Default Value	

Parameter	Description
Syntax	DirectoryString
Example	nsEncryptionAlgorithm: AES

4.5. DATABASE LINK PLUG-IN ATTRIBUTES (CHAINING ATTRIBUTES)

The database link plug-in attributes are also organized in an information tree, as shown in the following diagram:

Figure 4.4. Database Link Plug-in



All plug-in technology used by the database link instances is stored in the **cn=chaining database** plug-in node. This section presents the additional attribute information for the three nodes marked in bold in the **cn=chaining database,cn=plugins,cn=config** information tree in [Figure 4.4, “Database Link Plug-in”](#).

4.5.1. Database Link Attributes under **cn=config,cn=chaining database,cn=plugins,cn=config**

This section covers global configuration attributes common to all instances are stored in the **cn=config,cn=chaining database,cn=plugins,cn=config** tree node.

4.5.1.1. nsActiveChainingComponents

This attribute lists the components using chaining. A component is any functional unit in the server. The value of this attribute overrides the value in the global configuration attribute. To disable chaining on a particular database instance, use the value **None**. This attribute also allows the components used to chain to be altered. By default, no components are allowed to chain, which explains why this attribute will probably not appear in a list of **cn=config,cn=chaining database,cn=config** attributes, as LDAP considers empty attributes to be non-existent.

Parameter	Description
Entry DN	cn=config,cn=chaining database,cn=plugins,cn=config

Parameter	Description
Valid Values	Any valid component entry
Default Value	None
Syntax	DirectoryString
Example	nsActiveChainingComponents: cn=uid uniqueness,cn=plugins,cn=config

4.5.1.2. nsMaxResponseDelay

This error detection, performance-related attribute specifies the maximum amount of time it can take a remote server to respond to an LDAP operation request made by a database link before an error is suspected. Once this delay period has been met, the database link tests the connection with the remote server.

Parameter	Description
Entry DN	cn=config,cn=chaining database,cn=plugins,cn=config
Valid Values	Any valid delay period in seconds
Default Value	60 seconds
Syntax	Integer
Example	nsMaxResponseDelay: 60

4.5.1.3. nsMaxTestResponseDelay

This error detection, performance-related attribute specifies the duration of the test issued by the database link to check whether the remote server is responding. If a response from the remote server is not returned before this period has passed, the database link assumes the remote server is down, and the connection is not used for subsequent operations.

Parameter	Description
Entry DN	cn=config,cn=chaining database,cn=plugins,cn=config
Valid Values	Any valid delay period in seconds
Default Value	15 seconds

Parameter	Description
Syntax	Integer
Example	nsMaxTestResponseDelay: 15

4.5.1.4. nsTransmittedControls

This attribute, which can be both a global (and thus dynamic) configuration or an instance (that is, **cn=database link instance, cn=chaining database,cn=plugins,cn=config**) configuration attribute, allows the controls the database link forwards to be altered. The following controls are forwarded by default by the database link:

- Managed DSA (OID: 2.16.840.1.113730.3.4.2)
- Virtual list view (VLV) (OID: 2.16.840.1.113730.3.4.9)
- Server side sorting (OID: 1.2.840.113556.1.4.473)
- Loop detection (OID: 1.3.6.1.4.1.1466.29539.12)

Other controls, such as dereferencing and simple paged results for searches, can be added to the list of controls to forward.

Parameter	Description
Entry DN	cn=config,cn=chaining database,cn=plugins,cn=config
Valid Values	Any valid OID or the above listed controls forwarded by the database link
Default Value	None
Syntax	Integer
Example	nsTransmittedControls: 1.2.840.113556.1.4.473

4.5.2. Database Link Attributes under **cn=default instance config,cn=chaining database,cn=plugins,cn=config**

Default instance configuration attributes for instances are housed in the **cn=default instance config,cn=chaining database,cn=plugins,cn=config** tree node.

4.5.2.1. nsAbandonedSearchCheckInterval

This attribute shows the number of seconds that pass before the server checks for abandoned operations.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	0 to maximum 32-bit integer (2147483647) seconds
Default Value	1
Syntax	Integer
Example	nsAbandonedSearchCheckInterval: 10

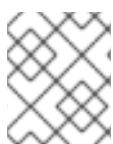
4.5.2.2. nsBindConnectionsLimit

This attribute shows the maximum number of TCP connections the database link establishes with the remote server.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	1 to 50 connections
Default Value	3
Syntax	Integer
Example	nsBindConnectionsLimit: 3

4.5.2.3. nsBindRetryLimit

Contrary to what the name suggests, this attribute does not specify the number of times a database link retries to bind with the remote server but the number of times it *tries* to bind with the remote server. A value of **1** here indicates that the database link only attempts to bind once.



NOTE

Retries only occur for connection failures and not for other types of errors, such as invalid bind DNs or bad passwords.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config

Parameter	Description
Valid Range	0 to 5
Default Value	3
Syntax	Integer
Example	nsBindRetryLimit: 3

4.5.2.4. nsBindTimeout

This attribute shows the amount of time before the bind attempt times out. There is no real valid range for this attribute, except reasonable patience limits.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	0 to 60 seconds
Default Value	15
Syntax	Integer
Example	nsBindTimeout: 15

4.5.2.5. nsCheckLocalACI

Reserved for advanced use only. This attribute controls whether ACIs are evaluated on the database link as well as the remote data server. Changes to this attribute only take effect once the server has been restarted.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsCheckLocalACI: on

4.5.2.6. nsConcurrentBindLimit

This attribute shows the maximum number of concurrent bind operations per TCP connection.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	1 to 25 binds
Default Value	10
Syntax	Integer
Example	nsConcurrentBindLimit: 10

4.5.2.7. nsConcurrentOperationsLimit

This attribute specifies the maximum number of concurrent operations allowed.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	1 to 50 operations
Default Value	2
Syntax	Integer
Example	nsConcurrentOperationsLimit: 5

4.5.2.8. nsConnectionLife

This attribute specifies connection lifetime. Connections between the database link and the remote server can be kept open for an unspecified time or closed after a specific period of time. It is faster to keep the connections open, but it uses more resources. When the value is **0** and a list of failover servers is provided in the **nsFarmServerURL** attribute, the main server is never contacted after failover to the alternate server.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config

Parameter	Description
Valid Range	0 to limitless seconds (where 0 means forever)
Default Value	0
Syntax	Integer
Example	nsConnectionLife: 0

4.5.2.9. nsOperationConnectionsLimit

This attribute shows the maximum number of LDAP connections the database link establishes with the remote server.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	1 to n connections
Default Value	20
Syntax	Integer
Example	nsOperationConnectionsLimit: 10

4.5.2.10. nsProxiedAuthorization

Reserved for advanced use only. If you disable proxied authorization, binds for chained operations are executed as the user set in the **nsMultiplexorBindDn** attribute.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Values	on off
Default Value	on
Syntax	DirectoryString
Example	nsProxiedAuthorization: on

4.5.2.11. nsReferralOnScopedSearch

This attribute controls whether referrals are returned by scoped searches. This attribute can be used to optimize the directory because returning referrals in response to scoped searches is more efficient. A referral is returned to all the configured farm servers.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	nsReferralOnScopedSearch: off

4.5.2.12. nsSizeLimit

This attribute shows the default size limit for the database link in bytes.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	-1 (no limit) to maximum 32-bit integer (2147483647) entries
Default Value	2000
Syntax	Integer
Example	nsSizeLimit: 2000

4.5.2.13. nsTimeLimit

This attribute shows the default search time limit for the database link.

Parameter	Description
Entry DN	cn=default instance config,cn=chaining database,cn=plugins,cn=config
Valid Range	-1 to maximum 32-bit integer (2147483647) seconds

Parameter	Description
Default Value	3600
Syntax	Integer
Example	nsTimeLimit: 3600

4.5.3. Database Link Attributes under `cn=database_link_name,cn=chaining, database,cn=plugins,cn=config`

This information node stores the attributes concerning the server containing the data. A *farm server* is a server which contains data on databases. This attribute can contain optional servers for failover, separated by spaces. For cascading chaining, this URL can point to another database link.

4.5.3.1. nsBindMechanism

This attribute sets a bind mechanism for the farm server to connect to the remote server. A farm server is a server containing data in one or more databases. This attribute configures the connection type, either standard, TLS, or SASL.

- *empty*. This performs simple authentication and requires the **nsMultiplexorBindDn** and **nsMultiplexorCredentials** attributes to give the bind information.
- *EXTERNAL*. This uses an TLS certificate to authenticate the farm server to the remote server. Either the farm server URL must be set to the secure URL (**ldaps**) or the **nsUseStartTLS** attribute must be set to **on**. Additionally, the remote server must be configured to map the farm server's certificate to its bind identity. Certificate mapping is described in the *Administration Guide*.
- *DIGEST-MD5*. This uses SASL with DIGEST-MD5 encryption. As with simple authentication, this requires the **nsMultiplexorBindDn** and **nsMultiplexorCredentials** attributes to give the bind information.
- *GSSAPI*. This uses Kerberos-based authentication over SASL. The farm server must be connected over the standard port, meaning the URL has **ldap**, because the Directory Server does not support SASL/GS-API over TLS. The farm server must be configured with a Kerberos keytab, and the remote server must have a defined SASL mapping for the farm server's bind identity. Setting up Kerberos keytabs and SASL mappings is described in the *Administration Guide*.

Parameter	Description
Entry DN	<code>cn=database_link_name,cn=chaining, database,cn=plugins,cn=config</code>

Parameter	Description
Valid Values	* empty * EXTERNAL * DIGEST-MD5 * GSSAPI
Default Value	empty
Syntax	DirectoryString
Example	nsBindMechanism: GSSAPI

4.5.3.2. nsFarmServerURL

This attribute gives the LDAP URL of the remote server. A farm server is a server containing data in one or more databases. This attribute can contain optional servers for failover, separated by spaces. If using cascading changing, this URL can point to another database link.

Parameter	Description
Entry DN	cn=database_link_name,cn=chaining database,cn=plugins,cn=config
Valid Values	Any valid remote server LDAP URL
Default Value	
Syntax	DirectoryString
Example	nsFarmServerURL: ldap://farm1.example.com farm2.example.com:389 farm3.example.com:1389/

4.5.3.3. nsMultiplexorBindDN

This attribute gives the DN of the administrative entry used to communicate with the remote server. The *multiplexor* is the server that contains the database link and communicates with the farm server. This bind DN cannot be the Directory Manager, and, if this attribute is not specified, the database link binds as **anonymous**.

Parameter	Description
Entry DN	cn=database_link_name,cn=chaining database,cn=plugins,cn=config

Parameter	Description
Valid Values	
Default Value	DN of the multiplexor
Syntax	DirectoryString
Example	nsMultiplexerBindDN: cn=proxy manager

4.5.3.4. nsMultiplexorCredentials

Password for the administrative user, given in plain text. If no password is provided, it means that users can bind as **anonymous**. The password is encrypted in the configuration file. The example below is what is shown, not what is typed.

Parameter	Description
Entry DN	cn=database_link_name,cn=chaining database,cn=plugins,cn=config
Valid Values	Any valid password, which will then be encrypted using the DES reversible password encryption schema
Default Value	
Syntax	DirectoryString
Example	nsMultiplexerCredentials: {DES} 9Eko69APCJff

4.5.3.5. nshoplimit

This attribute specifies the maximum number of times a database is allowed to chain; that is, the number of times a request can be forwarded from one database link to another.

Parameter	Description
Entry DN	cn=database_link_name,cn=chaining database,cn=plugins,cn=config
Valid Range	1 to an appropriate upper limit for the deployment
Default Value	10
Syntax	Integer

Parameter	Description
Example	nsHopLimit: 3

4.5.3.6. nsUseStartTLS

This attribute sets whether to use Start TLS to initiate a secure, encrypted connection over an insecure port. This attribute can be used if the **nsBindMechanism** attribute is set to **EXTERNAL** but the farm server URL set to the standard URL (**ldap**) or if the **nsBindMechanism** attribute is left empty.

Parameter	Description
Entry DN	<code>cn=database_link_name,cn=chaining database,cn=plugins,cn=config</code>
Valid Values	off on
Default Value	off
Syntax	DirectoryString
Example	nsUseStartTLS: on

4.5.4. Database Link Attributes under `cn=monitor,cn=database instance name,cn=chaining database,cn=plugins,cn=config`

Attributes used for monitoring activity on the instances are stored in the **cn=monitor,cn=database instance name,cn=chaining database,cn=plugins,cn=config** information tree.

nsAddCount

This attribute gives the number of add operations received.

nsDeleteCount

This attribute gives the number of delete operations received.

nsModifyCount

This attribute gives the number of modify operations received.

nsRenameCount

This attribute gives the number of rename operations received.

nsSearchBaseCount

This attribute gives the number of base level searches received.

nsSearchOneLevelCount

This attribute gives the number of one-level searches received.

nsSearchSubtreeCount

This attribute gives the number of subtree searches received.

nsAbandonCount

This attribute gives the number of abandon operations received.

nsBindCount

This attribute gives the number of bind requests received.

nsUnbindCount

This attribute gives the number of unbinds received.

nsCompareCount

This attribute gives the number of compare operations received.

nsOperationConnectionCount

This attribute gives the number of open connections for normal operations.

nsOpenBindConnectionCount

This attribute gives the number of open connections for bind operations.

4.6. PAM PASS THROUGH AUTH PLUG-IN ATTRIBUTES

Local PAM configurations on Unix systems can leverage an external authentication store for LDAP users. This is a form of pass-through authentication which allows the Directory Server to use the externally-stored user credentials for directory access.

PAM pass-through authentication is configured in child entries beneath the PAM Pass Through Auth Plug-in container entry. All of the possible configuration attributes for PAM authentication (defined in the **60pam-plugin.ldif** schema file) are available to a child entry; the child entry must be an instance of the PAM configuration object class.

Example 4.1. Example PAM Pass Through Auth Configuration Entries

```
dn: cn=PAM Pass Through Auth,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
objectClass: pamConfig
cn: PAM Pass Through Auth
nsslapd-pluginPath: libpam-passthru-plugin
nsslapd-pluginInitfunc: pam_passthruauth_init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: on
nsslapd-pluginLoadGlobal: true
nsslapd-plugindepends-on-type: database
nsslapd-pluginId: pam_passthruauth
nsslapd-pluginVersion: 9.0.0
nsslapd-pluginVendor: Red Hat
nsslapd-pluginDescription: PAM pass through authentication plugin
```

```

dn: cn=Example PAM Config,cn=PAM Pass Through Auth,cn=plugins,cn=config
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
objectClass: pamConfig
cn: Example PAM Config
pamMissingSuffix: ALLOW
pamExcludeSuffix: cn=config
pamIDMapMethod: RDN ou=people,dc=example,dc=com
pamIDMapMethod: ENTRY ou=engineering,dc=example,dc=com
pamIDAttr: customPamUid
pamFilter: (manager=uid=bjensen,ou=people,dc=example,dc=com)
pamFallback: FALSE
pamSecure: TRUE
pamService: ldapserver

```

The PAM configuration, at a minimum, must define a mapping method (a way to identify what the PAM user ID is from the Directory Server entry), the PAM server to use, and whether to use a secure connection to the service.

```

pamIDMapMethod: RDN
pamSecure: FALSE
pamService: ldapserver

```

The configuration can be expanded for special settings, such as to exclude or specifically include subtrees or to map a specific attribute value to the PAM user ID.

4.6.1. pamConfig (Object Class)

This object class is used to define the PAM configuration to interact with the directory service. This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.318

Allowed Attributes

- [Section 4.6.2, "pamExcludeSuffix"](#)
- [Section 4.6.7, "pamIncludeSuffix"](#)
- [Section 4.6.8, "pamMissingSuffix"](#)
- [Section 4.6.4, "pamFilter"](#)
- [Section 4.6.5, "pamIDAttr"](#)
- [Section 4.6.6, "pamIDMapMethod"](#)
- [Section 4.6.3, "pamFallback"](#)

- [Section 4.6.9, "pamSecure"](#)
- [Section 4.6.10, "pamService"](#)
- **nsslapd-pluginConfigArea**

4.6.2. pamExcludeSuffix

This attribute specifies a suffix to exclude from PAM authentication.

OID	2.16.840.1.113730.3.1.2068
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

4.6.3. pamFallback

Sets whether to fallback to regular LDAP authentication if PAM authentication fails.

OID	2.16.840.1.113730.3.1.2072
Syntax	Boolean
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

4.6.4. pamFilter

Sets an LDAP filter to use to identify specific entries within the included suffixes for which to use PAM pass-through authentication. If not set, all entries within the suffix are targeted by the configuration entry.

OID	2.16.840.1.113730.3.1.2131
Syntax	Boolean
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

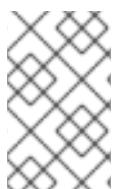
4.6.5. pamIDAttr

This attribute contains the attribute name which is used to hold the PAM user ID.

OID	2.16.840.1.113730.3.1.2071
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

4.6.6. pamIDMapMethod

Gives the method to use to map the LDAP bind DN to a PAM identity.



NOTE

Directory Server user account inactivation is only validated using the ENTRY mapping method. With RDN or DN, a Directory Server user whose account is inactivated can still bind to the server successfully.

OID	2.16.840.1.113730.3.1.2070
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

4.6.7. pamIncludeSuffix

This attribute sets a suffix to include for PAM authentication.

OID	2.16.840.1.113730.3.1.2067
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

4.6.8. pamMissingSuffix

Identifies how to handle missing include or exclude suffixes. The options are ERROR (which causes the bind operation to fail); ALLOW, which logs an error but allows the operation to proceed; and IGNORE, which allows the operation and does not log any errors.

OID	2.16.840.1.113730.3.1.2069
-----	----------------------------

Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

4.6.9. pamSecure

Requires secure TLS connection for PAM authentication.

OID	2.16.840.1.113730.3.1.2073
Syntax	Boolean
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

4.6.10. pamService

Contains the service name to pass to PAM. This assumes that the service specified has a configuration file in the **/etc/pam.d/** directory.



IMPORTANT

The **pam_fprintd.so** module cannot be in the configuration file referenced by the **pamService** attribute of the **PAM Pass-Through Authentication Plug-in** configuration. Using the PAM **pam_fprintd.so** module causes the Directory Server to hit the max file descriptor limit and can cause the Directory Server process to abort.



IMPORTANT

The **pam_fprintd.so** module cannot be in the configuration file referenced by the **pamService** attribute of the PAM Pass-Through Authentication Plug-in configuration. Using the PAM **fprintd** module causes the Directory Server to hit the max file descriptor limit and can cause the Directory Server process to abort.

OID	2.16.840.1.113730.3.1.2074
Syntax	IA5String
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

4.7. ACCOUNT POLICY PLUG-IN ATTRIBUTES

Account policies can be set that automatically lock an account after a certain amount of time has elapsed. This can be used to create temporary accounts that are only valid for a preset amount of time or to lock users which have been inactive for a certain amount of time.

The Account Policy Plug-in itself only accept one argument, which points to a plug-in configuration entry.

```
dn: cn=Account Policy Plugin,cn=plugins,cn=config
...
nsslapd-pluginarg0: cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
```

The account policy configuration entry defines, for the entire server, what attributes to use for account policies. Most of the configuration defines attributes to use to evaluate account policies and expiration times, but the configuration also defines what object class to use to identify subtree-level account policy definitions.

```
dn: cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
objectClass: top
objectClass: extensibleObject
cn: config

... attributes for evaluating accounts ...
alwaysRecordLogin: yes
stateattrname: lastLoginTime
altstateattrname: createTimestamp

... attributes for account policy entries ...
specattrname: acctPolicySubentry
limitattrname: accountInactivityLimit
```

Once the plug-in is configured globally, account policy entries can be created within the user subtrees, and then these policies can be applied to users and to roles through classes of service.

Example 4.2. Account Policy Definition

```
dn: cn=AccountPolicy,dc=example,dc=com
objectClass: top
objectClass: ldapsubentry
objectClass: extensibleObject
objectClass: accountpolicy
# 86400 seconds per day * 30 days = 2592000 seconds
accountInactivityLimit: 2592000
cn: AccountPolicy
```

Any entry, both individual users and roles or CoS templates, can be an account policy subentry. Every account policy subentry has its creation and login times tracked against any expiration policy.

Example 4.3. User Account with Account Policy

```
dn: uid=scarter,ou=people,dc=example,dc=com
...
lastLoginTime: 20060527001051Z
acctPolicySubentry: cn=AccountPolicy,dc=example,dc=com
```

4.7.1. altstateattrname

Account expiration policies are based on some timed criteria for the account. For example, for an inactivity policy, the primary criteria may be the last login time, **lastLoginTime**. However, there may be instances where that attribute does not exist on an entry, such as a user who never logged into his account. The **altstateattrname** attribute provides a backup attribute for the server to reference to evaluate the expiration time.

Parameter	Description
Entry DN	cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
Valid Range	Any time-based entry attribute
Default Value	None
Syntax	DirectoryString
Example	altstateattrname: createTimeStamp

4.7.2. alwaysRecordLogin

By default, only entries which have an account policy directly applied to them – meaning, entries with the **acctPolicySubentry** attribute – have their login times tracked. If account policies are applied through classes of service or roles, then the **acctPolicySubentry** attribute is on the template or container entry, not the user entries themselves.

The **alwaysRecordLogin** attribute sets that every entry records its last login time. This allows CoS and roles to be used to apply account policies.

Parameter	Description
Entry DN	cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
Valid Range	yes no
Default Value	no
Syntax	DirectoryString
Example	alwaysRecordLogin: no

4.7.3. alwaysRecordLoginAttr

The **Account Policy** plug-in uses the attribute name set in the **alwaysRecordLoginAttr** parameter to store the time of the last successful login in this attribute in the user's directory entry. For further information, see the corresponding section in the *Directory Server Administration Guide*.

Parameter	Description
Entry DN	cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
Valid Range	Any valid attribute name
Default Value	stateAttrName
Syntax	DirectoryString
Example	alwaysRecordLoginAttr: lastLoginTime

4.7.4. limitattrname

The account policy entry in the user directory defines the time limit for the account lockout policy. This time limit can be set in any time-based attribute, and a policy entry could have multiple time-based attributes in it. The attribute within the policy to use for the account inactivation limit is defined in the **limitattrname** attribute in the Account Policy Plug-in, and it is applied globally to all account policies.

Parameter	Description
Entry DN	cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
Valid Range	Any time-based entry attribute
Default Value	None
Syntax	DirectoryString
Example	limitattrname: accountInactivityLimit

4.7.5. specattrname

There are really two configuration entries for an account policy: the global settings in the plug-in configuration entry and then user- or subtree-level settings in an entry within the user directory. An account policy can be set directly on a user entry or it can be set as part of a CoS or role configuration. The way that the plug-in identifies which entries are account policy configuration entries is by identifying a specific attribute on the entry which flags it as an account policy. This attribute in the plug-in configuration is **specattrname**; its will usually be set to **acctPolicySubentry**.

Parameter	Description
Entry DN	cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
Valid Range	Any time-based entry attribute
Default Value	None
Syntax	DirectoryString
Example	specattrname: acctPolicySubentry

4.7.6. stateattrname

Account expiration policies are based on some timed criteria for the account. For example, for an inactivity policy, the primary criteria may be the last login time, **lastLoginTime**. The primary time attribute used to evaluate an account policy is set in the **stateattrname** attribute.

Parameter	Description
Entry DN	cn=config,cn=Account Policy Plugin,cn=plugins,cn=config
Valid Range	Any time-based entry attribute
Default Value	None
Syntax	DirectoryString
Example	stateattrname: lastLoginTime

4.8. AD DN PLUG-IN ATTRIBUTES

The **AD DN** plug-in supports multiple domain configurations. Create one configuration entry for each domain. For details, see the corresponding section in the *Red Hat Directory Server Administration Guide*.

4.8.1. cn

Sets the domain name of the configuration entry. The plug-in uses the domain name from the authenticating user name to select the corresponding configuration entry.

Parameter	Description
Entry DN	<i>cn=domain_name,cn=addn,cn=plugins,cn=config</i>
Valid Entry	Any string

Parameter	Description
Default Value	None
Syntax	DirectoryString
Example	cn: example.com

4.8.2. addn_base

Sets the base DN under which Directory Server searches the user's DN.

Parameter	Description
Entry DN	<code>cn=domain_name,cn=addn,cn=plugins,cn=config</code>
Valid Entry	Any valid DN
Default Value	None
Syntax	DirectoryString
Example	<code>addn_base: ou=People,dc=example,dc=com</code>

4.8.3. addn_filter

Sets the search filter. Directory Server replaces the **%s** variable automatically with the non-domain part of the authenticating user. For example, if the user name in the bind is **user_name@example.com**, the filter searches the corresponding DN which is **(&(objectClass=account)(uid=user_name))**.

Parameter	Description
Entry DN	<code>cn=domain_name,cn=addn,cn=plugins,cn=config</code>
Valid Entry	Any valid DN
Default Value	None
Syntax	DirectoryString
Example	<code>addn_filter: (&(objectClass=account)(uid=%s))</code>

4.9. AUTO MEMBERSHIP PLUG-IN ATTRIBUTES

Automembership essentially allows a static group to act like a dynamic group. Different automembership definitions create searches that are automatically run on all new directory entries. The

automembership rules search for and identify matching entries – much like the dynamic search filters – and then explicitly add those entries as members to the specified static group.

The Auto Membership Plug-in itself is a container entry. Each automember definition is a child of the Auto Membership Plug-in. The automember definition defines the LDAP search base and filter to identify entries and a default group to add them to.

```
dn: cn=Hostgroups,cn=Auto Membership Plugin,cn=plugins,cn=config
objectclass: autoMemberDefinition
cn: Hostgroups
autoMemberScope: dc=example,dc=com
autoMemberFilter: objectclass=ipHost
autoMemberDefaultGroup: cn=systems,cn=hostgroups,ou=groups,dc=example,dc=com
autoMemberGroupingAttr: member:dn
```

Each automember definition can have its own child entry that defines additional conditions for assigning the entry to group. Regular expressions can be used to include or exclude entries and assign them to specific groups based on those conditions.

```
dn: cn=webservers,cn=Hostgroups,cn=Auto Membership Plugin,cn=plugins,cn=config
objectclass: autoMemberRegexRule
description: Group for webservers
cn: webservers
autoMemberTargetGroup: cn=webservers,cn=hostgroups,dc=example,dc=com
autoMemberInclusiveRegex: fqdn=~www\.[0-9]+\.\example\.com
```

If the entry matches the main definition and not any of the regular expression conditions, then it uses the group in the main definition. If it matches a regular expression condition, then it is added to the regular expression condition group.

4.9.1. **autoMemberDefaultGroup**

This attribute sets a default or fallback group to add the entry to as a member. If only the definition entry is used, then this is the group to which all matching entries are added. If regular expression conditions are used, then this group is used as a fallback if an entry which matches the LDAP search filter do not match any of the regular expressions.

Parameter	Description
Entry DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	Any existing Directory Server group
Default Value	None
Single- or Multi-Valued	Single
Syntax	DirectoryString
Example	autoMemberDefaultGroup: cn=hostgroups,ou=groups,dc=example,dc=com

4.9.2. autoMemberDefinition (Object Class)

This attribute identifies the entry as an automember definition. This entry must be a child of the Auto Membership Plug-in, **cn=Auto Membership Plugin,cn=plugins,cn=config**.

Allowed Attributes

- autoMemberScope
- autoMemberFilter
- autoMemberDefaultGroup
- autoMemberGroupingAttr

4.9.3. autoMemberExclusiveRegex

This attribute sets a single regular expression to use to identify entries to *exclude*. If an entry matches the exclusion condition, then it is *not* included in the group. Multiple regular expressions could be used, and if an entry matches any one of those expressions, it is excluded in the group.

The format of the expression is a Perl-compatible regular expression (PCRE). For more information on PCRE patterns, see [the pcresyntax\(3\) man page](#).



NOTE

Exclude conditions are evaluated first and take precedence over include conditions.

Parameter	Description
Entry DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	Any regular expression
Default Value	None
Single- or Multi-Valued	Multi-valued
Syntax	DirectoryString
Example	autoMemberExclusiveRegex: fqdn=^www\.\web[0-9]+\.\example\.com

4.9.4. autoMemberFilter

This attribute sets a standard LDAP search filter to use to search for matching entries.

Parameter	Description
Entry DN	cn=Auto Membership Plugin,cn=plugins,cn=config

Parameter	Description
Valid Range	Any valid LDAP search filter
Default Value	None
Single- or Multi-Valued	Single
Syntax	DirectoryString
Example	autoMemberFilter:objectclass=ntUser

4.9.5. autoMemberGroupingAttr

This attribute gives the name of the member attribute in the group entry and the attribute in the object entry that supplies the member attribute value, in the format `group_member_attr:entry_attr`.

This structures how the Automembership Plug-in adds a member to the group, depending on the group configuration. For example, for a **groupOfUniqueNames** user group, each member is added as a **uniqueMember** attribute. The value of **uniqueMember** is the DN of the user entry. In essence, each group member is identified by the attribute-value pair of **uniqueMember: user_entry_DN**. The member entry format, then, is **uniqueMember:dn**.

Parameter	Description
Entry DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	Any Directory Server attribute
Default Value	None
Single- or Multi-Valued	Single
Syntax	DirectoryString
Example	autoMemberGroupingAttr: member:dn

4.9.6. autoMemberInclusiveRegex

This attribute sets a single regular expression to use to identify entries to *include*. Multiple regular expressions could be used, and if an entry matches any one of those expressions, it is included in the group (assuming it does not match an exclude expression).

The format of the expression is a Perl-compatible regular expression (PCRE). For more information on PCRE patterns, see [the pcresyntax\(3\) man page](#).

Parameter	Description
Entry DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	Any regular expression
Default Value	None
Single- or Multi-Valued	Multi-valued
Syntax	DirectoryString
Example	autoMemberInclusiveRegex: fqdn=^www\.\web[0-9]+\.\example\.com

4.9.7. autoMemberProcessModifyOps

By default, the Directory Server invokes the Automembership plug-in for add and modify operations. With this setting, the plug-in changes groups when you add a group entry to a user or modify a group entry of a user. If you set the **autoMemberProcessModifyOps** to **off**, Directory Server only invokes the Automembership plug-in when you add a group entry to a user. In this case, if an administrator changes a user entry, and that entry impacts what Automembership groups the user belongs to, the plug-in does not remove the user from the old group and only adds the new group. To update the old group, you must then manually run a fix-up task.

Parameter	Description
Entry DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Values	on off
Default Value	on
Single- or Multi-Valued	Single
Syntax	DirectoryString
Example	autoMemberProcessModifyOps: on

4.9.8. autoMemberRegexRule (Object Class)

This attribute identifies the entry as a regular expression rule. This entry must be a child of an automember definition (**objectclass: autoMemberDefinition**).

Allowed Attributes

- autoMemberInclusiveRegex

- autoMemberExclusiveRegex
- autoMemberTargetGroup

4.9.9. autoMemberScope

This attribute sets the subtree DN to search for entries. This is the search base.

Parameter	Description
Entry DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	Any Directory Server subtree
Default Value	None
Single- or Multi-Valued	Single
Syntax	DirectoryString
Example	autoMemberScope: dc=example,dc=com

4.9.10. autoMemberTargetGroup

This attribute sets which group to add the entry to as a member, if it meets the regular expression conditions.

Parameter	Description
Entry DN	cn=Auto Membership Plugin,cn=plugins,cn=config
Valid Range	Any Directory Server group
Default Value	None
Single- or Multi-Valued	Single
Syntax	DirectoryString
Example	autoMemberTargetGroup: cn=webservers,cn=hostgroups,ou=groups,dc=example,dc=com

4.10. DISTRIBUTED NUMERIC ASSIGNMENT PLUG-IN ATTRIBUTES

The Distributed Numeric Assignment Plug-in manages ranges of numbers and assigns unique numbers within that range to entries. By breaking number assignments into ranges, the Distributed Numeric Assignment Plug-in allows multiple servers to assign numbers without conflict. The plug-in also

manages the ranges assigned to servers, so that if one instance runs through its range quickly, it can request additional ranges from the other servers.

Distributed numeric assignment can be configured to work with single attribute types or multiple attribute types, and is only applied to specific suffixes and specific entries within the subtree.

Distributed numeric assignment is handled per-attribute and is only applied to specific suffixes and specific entries within the subtree.

4.10.1. dnaPluginConfig (Object Class)

This object class is used for entries which configure the DNA Plug-in and numeric ranges to assign to entries.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.324

Allowed Attributes

- dnaType
- dnaPrefix
- dnaNextValue
- dnaMaxValue
- dnaInterval
- dnaMagicRegen
- dnaFilter
- dnaScope
- dnaSharedCfgDN
- dnaThreshold
- dnaNextRange
- dnaRangeRequestTimeout
- cn

4.10.2. dnaFilter

This attribute sets an LDAP filter to use to search for and identify the entries to which to apply the distributed numeric assignment range.

The **dnaFilter** attribute is required to set up distributed numeric assignment for an attribute.

Parameter	Description
Entry DN	<code>cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</code>
Valid Range	Any valid LDAP filter
Default Value	None
Syntax	DirectoryString
Example	<code>dnaFilter: (objectclass=person)</code>

4.10.3. dnaInterval

This attribute sets an interval to use to increment through numbers in a range. Essentially, this skips numbers at a predefined rate. If the interval is **3** and the first number in the range is **1**, the next number used in the range is **4**, then **7**, then **10**, incrementing by three for every new number assignment.

In a replication environment, the **dnaInterval** enables multiple servers to share the same range. However, when you configure different servers that share the same range, set the **dnaInterval** and **dnaNextVal** parameters accordingly so that the different servers do not generate the same values. You must also consider this if you add new servers to the replication topology.

Parameter	Description
Entry DN	<code>cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</code>
Valid Range	Any integer
Default Value	1
Syntax	Integer
Example	<code>dnaInterval: 1</code>

4.10.4. dnaMagicRegen

This attribute sets a user-defined value that instructs the plug-in to assign a new value for the entry. The magic value can be used to assign new unique numbers to existing entries or as a standard setting when adding new entries.

The magic entry should be outside of the defined range for the server so that it cannot be triggered by accident. Note that this attribute does not have to be a number when used on a DirectoryString or other character type. However, in most cases the DNA plug-in is used on attributes which only accept integer values, and in such cases the **dnamagicregen** value must also be an integer.

Parameter	Description
Entry DN	<code>cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</code>
Valid Range	Any string
Default Value	None
Syntax	DirectoryString
Example	<code>dnaMagicRegen: -1</code>

4.10.5. dnaMaxValue

This attribute sets the maximum value that can be assigned for the range. The default is **-1**, which is the same as setting the highest 64-bit integer.

Parameter	Description
Entry DN	<code>cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</code>
Valid Range	1 to the maximum 32-bit integer on 32-bit systems and to the maximum 64-bit integer on 64-bit systems; -1 is unlimited
Default Value	-1
Syntax	Integer
Example	<code>dna.MaxValue: 1000</code>

4.10.6. dnaNextRange

This attribute defines the next range to use when the current range is exhausted. This value is automatically set when range is transferred between servers, but it can also be manually set to add a range to a server if range requests are not used.

The **dnaNextRange** attribute should be set explicitly only if a separate, specific range has to be assigned to other servers. Any range set in the **dnaNextRange** attribute must be unique from the available range for the other servers to avoid duplication. If there is no request from the other servers and the server where **dnaNextRange** is set explicitly has reached its set **dna.MaxValue**, the next set of values (part of the **dnaNextRange**) is allocated from this deck.

The **dnaNextRange** allocation is also limited by the **dnaThreshold** attribute that is set in the DNA configuration. Any range allocated to another server for **dnaNextRange** cannot violate the threshold for the server, even if the range is available on the deck of **dnaNextRange**.

**NOTE**

If the **dnaNextRange** attribute is handled internally if it is not set explicitly. When it is handled automatically, the **dna.MaxValue** attribute serves as upper limit for the next range.

The attribute sets the range in the format *lower_range–upper_range*.

Parameter	Description
Entry DN	<code>cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</code>
Valid Range	1 to the maximum 32-bit integer on 32-bit systems and to the maximum 64-bit integer on 64-bit systems for the lower and upper ranges
Default Value	None
Syntax	DirectoryString
Example	<code>dnaNextRange: 100-500</code>

4.10.7. dnaNextValue

This attribute gives the next available number which can be assigned. After being initially set in the configuration entry, this attribute is managed by the Distributed Numeric Assignment Plug-in.

The **dnaNextValue** attribute is required to set up distributed numeric assignment for an attribute.

Parameter	Description
Entry DN	<code>cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</code>
Valid Range	1 to the maximum 32-bit integer on 32-bit systems and to the maximum 64-bit integer on 64-bit systems
Default Value	-1
Syntax	Integer
Example	<code>dnaNextValue: 1</code>

4.10.8. dnaPrefix

This attribute defines a prefix that can be prepended to the generated number values for the attribute. For example, to generate a user ID such as **user1000**, the **dnaPrefix** setting would be **user**.

dnaPrefix can hold any kind of string. However, some possible values for **dnaType** (such as **uidNumber** and **gidNumber**) require only integer values. To use a prefix string, consider using a custom attribute for **dnaType** which allows strings.

Parameter	Description
Entry DN	<code>cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</code>
Valid Range	Any string
Default Value	None
Example	<code>dnaPrefix: id</code>

4.10.9. dnaRangeRequestTimeout

One potential situation with the Distributed Numeric Assignment Plug-in is that one server begins to run out of numbers to assign. The **dnaThreshold** attribute sets a threshold of available numbers in the range, so that the server can request an additional range from the other servers before it is unable to perform number assignments.

The **dnaRangeRequestTimeout** attribute sets a timeout period, in seconds, for range requests so that the server does not stall waiting on a new range from one server and can request a range from a new server.

For range requests to be performed, the **dnaSharedCfgDN** attribute must be set.

Parameter	Description
Entry DN	<code>cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</code>
Valid Range	1 to the maximum 32-bit integer on 32-bit systems and to the maximum 64-bit integer on 64-bit systems
Default Value	10
Syntax	Integer
Example	<code>dnaRangeRequestTimeout: 15</code>

4.10.10. dnaScope

This attribute sets the base DN to search for entries to which to apply the distributed numeric assignment. This is analogous to the base DN in an **Idapsearch**.

Parameter	Description
Entry DN	<code>cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</code>
Valid Range	Any Directory Server entry
Default Value	None
Syntax	DirectoryString
Example	<code>dnaScope: ou=people,dc=example,dc=com</code>

4.10.11. dnaSharedCfgDN

This attribute defines a shared identity that the servers can use to transfer ranges to one another. This entry is replicated between servers and is managed by the plug-in to let the other servers know what ranges are available. This attribute must be set for range transfers to be enabled.



NOTE

The shared configuration entry must be configured in the replicated subtree, so that the entry can be replicated to the servers. For example, if the **ou=People,dc=example,dc=com** subtree is replicated, then the configuration entry must be in that subtree, such as **ou=UID Number Ranges, ou=People,dc=example,dc=com**.

The entry identified by this setting must be manually created by the administrator. The server will automatically contain a sub-entry beneath it to transfer ranges.

Parameter	Description
Entry DN	<code>cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</code>
Valid Range	Any DN
Default Value	None
Syntax	DN
Example	<code>dnaSharedCfgDN: cn=range transfer user,cn=config</code>

4.10.12. dnaThreshold

One potential situation with the Distributed Numeric Assignment Plug-in is that one server begins to run out of numbers to assign, which can cause problems. The Distributed Numeric Assignment Plug-in allows the server to request a new range from the available ranges on other servers.

So that the server can recognize when it is reaching the end of its assigned range, the **dnaThreshold** attribute sets a threshold of remaining available numbers in the range. When the server hits the threshold, it sends a request for a new range.

For range requests to be performed, the **dnaSharedCfgDN** attribute must be set.

Parameter	Description
Entry DN	cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	1 to the maximum 32-bit integer on 32-bit systems and to the maximum 64-bit integer on 64-bit systems
Default Value	100
Syntax	Integer
Example	dnaThreshold: 100

4.10.13. dnaType

This attribute sets which attributes have unique numbers being generated for them. In this case, whenever the attribute is added to the entry with the magic number, an assigned value is automatically supplied.

This attribute is required to set a distributed numeric assignment for an attribute.

If the **dnaPrefix** attribute is set, then the prefix value is prepended to whatever value is generated by **dnaType**. The **dnaPrefix** value can be any kind of string, but some reasonable values for **dnaType** (such as **uidNumber** and **gidNumber**) require only integer values. To use a prefix string, consider using a custom attribute for **dnaType** which allows strings.

Parameter	Description
Entry DN	cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Valid Range	Any Directory Server attribute
Default Value	None
Example	dnaType: uidNumber

4.10.14. dnaSharedConfig (Object Class)

This object class is used to configure the shared configuration entry that is replicated between suppliers that are all using the same DNA Plug-in configuration for numeric assignments.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.325

Allowed Attributes

- dnaHostname
- dnaPortNum
- dnaSecurePortNum
- dnaRemainingValues

4.10.15. dnaHostname

This attribute identifies the host name of a server in a shared range, as part of the DNA range configuration for that specific host in multi-supplier replication. Available ranges are tracked by host and the range information is replicated among all suppliers so that if any supplier runs low on available numbers, it can use the host information to contact another supplier and request a new range.

Parameter	Description
Entry DN	<code>cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</code>
Syntax	DirectoryString
Valid Range	Any valid host name
Default Value	None
Example	<code>dnahostname: ldap1.example.com</code>

4.10.16. dnaPortNum

This attribute gives the standard port number to use to connect to the host identified in **dnaHostname**.

Parameter	Description
Entry DN	<code>cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</code>
Syntax	Integer

Parameter	Description
Valid Range	0 to 65535
Default Value	389
Example	dnaPortNum: 389

4.10.17. dnaRemainingValues

This attribute contains the number of values that are remaining and available to a server to assign to entries.

Parameter	Description
Entry DN	dnaHostname= <i>host_name</i> +dnaPortNum= <i>port_number</i> ,ou=ranges,dc=example,dc=com
Syntax	Integer
Valid Range	Any integer
Default Value	None
Example	dnaRemainingValues: 1000

4.10.18. dnaRemoteBindCred

Specifies the Replication Manager's password. If you set a bind method in the **dnaRemoteBindMethod** attribute that requires authentication, additionally set the **dnaRemoteBindDN** and **dnaRemoteBindCred** parameter for every server in the replication deployment in the plug-in configuration entry under the **cn=config** entry.

Set the parameter in plain text. The value is automatically AES-encrypted before it is stored.

A server restart is required for the change to take effect.

Parameter	Description
Entry DN	cn=DNA_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config
Syntax	DirectoryString {AES} encrypted_password
Valid Values	Any valid AES-encrypted password.
Default Value	

Parameter	Description
Example	dnaRemoteBindCred: {AES-TUhNRONTcUdTSWIzRFFFkRUQm1NRVVHQ1NxR1NJYjNEUVGRERBNEJDUmxB0Uk0WXpjM1l5MHdaVE5rTXpZNA0KTnkxaE9XSmhORGRoT0MwMk1ESmpNV014TUFBQ0FRSUNBU0F3Q2dZSUtvWklodmNOQWdj0hRWUpZSVpJQVdVRA0KQkFFcUJCQk5KbUFDUWFOMHITWdsUVp3QjBJOQ==}bBR3On6cBmw0DdhcRx826g==

4.10.19. dnaRemoteBindDN

Specifies the Replication Manager DN. If you set a bind method in the **dnaRemoteBindMethod** attribute that requires authentication, additionally set the **dnaRemoteBindDN** and **dnaRemoteBindCred** parameter for every server in the replication deployment in the plug-in configuration under the **cn=config** entry.

A server restart is required for the change to take effect.

Parameter	Description
Entry DN	<i>cn=DN_A_config_entry,cn=Distributed Numeric Assignment Plugin,cn=plugins,cn=config</i>
Syntax	DirectoryString
Valid Values	Any valid Replication Manager DN.
Default Value	
Example	dnaRemoteBindDN: cn=replication manager,cn=config

4.10.20. dnaRemoteBindMethod

Specifies the remote bind method. If you set a bind method in this attribute that requires authentication, additionally set the **dnaRemoteBindDN** and **dnaRemoteBindCred** parameter for every server in the replication deployment in the plug-in configuration entry under the **cn=config** entry.

A server restart is required for the change to take effect.

Parameter	Description
Entry DN	<i>dnaHostname=host_name+dnaPortNum=port_number,ou=ranges,dc=example,dc=com</i>
Syntax	DirectoryString

Parameter	Description
Valid Values	SIMPLE SSL SASL/GSSAPI SASL/DIGEST-MD5
Default Value	
Example	dnaRemoteBindMethod: SIMPLE

4.10.21. dnaRemoteConnProtocol

Specifies the remote connection protocol.

A server restart is required for the change to take effect.

Parameter	Description
Entry DN	<code>dnaHostname=host_name+dnsPortNum=port_number,ou=ranges,dc=example,dc=com</code>
Syntax	DirectoryString
Valid Values	LDAP, SSL, or TLS
Default Value	
Example	dnaRemoteConnProtocol: LDAP

4.10.22. dnaSecurePortNum

This attribute gives the secure (TLS) port number to use to connect to the host identified in **dnaHostname**.

Parameter	Description
Entry DN	<code>dnaHostname=host_name+dnsPortNum=port_number,ou=ranges,dc=example,dc=com</code>
Syntax	Integer
Valid Range	0 to 65535
Default Value	636
Example	dnaSecurePortNum: 636

4.11. LINKED ATTRIBUTES PLUG-IN ATTRIBUTES

Many times, entries have inherent relationships to each other (such as managers and employees, document entries and their authors, or special groups and group members). While attributes exist that reflect these relationships, these attributes have to be added and updated on each entry manually. That can lead to a whimsically inconsistent set of directory data, where these entry relationships are unclear, outdated, or missing.

The Linked Attributes Plug-in allows one attribute, set in one entry, to update another attribute in another entry automatically. The first attribute has a DN value, which points to the entry to update; the second entry attribute also has a DN value which is a back-pointer to the first entry. The link attribute which is set by users and the dynamically-updated "managed" attribute in the affected entries are both defined by administrators in the Linked Attributes Plug-in instance.

Conceptually, this is similar to the way that the MemberOf Plug-in uses the **member** attribute in group entries to set **memberOf** attribute in user entries. Only with the Linked Attributes Plug-in, all of the link/managed attributes are user-defined and there can be multiple instances of the plug-in, each reflecting different link-managed relationships.

There are a couple of caveats for linking attributes:

- Both the link attribute and the managed attribute must have DNs as values. The DN in the link attribute points to the entry to add the managed attribute to. The managed attribute contains the linked entry DN as its value.
- The managed attribute must be multi-valued. Otherwise, if multiple link attributes point to the same managed entry, the managed attribute value would not be updated accurately.

4.11.1. linkScope

This restricts the scope of the plug-in, so it operates only in a specific subtree or suffix. If no scope is given, then the plug-in will update any part of the directory tree.

Parameter	Description
Entry DN	<code>cn=plugin_instance,cn=Linked Attributes,cn=plugins,cn=config</code>
Valid Range	Any DN
Default Value	None
Syntax	DN
Example	<code>linkScope: ou=People,dc=example,dc=com</code>

4.11.2. linkType

This sets the user-managed attribute. This attribute is modified and maintained by users, and then when this attribute value changes, the linked attribute is automatically updated in the targeted entries.

Parameter	Description
Entry DN	<code>cn=plugin_instance,cn=Linked Attributes,cn=plugins,cn=config</code>
Valid Range	Any Directory Server attribute
Default Value	None
Syntax	DirectoryString
Example	<code>linkType: directReport</code>

4.11.3. managedType

This sets the managed, or plug-in maintained, attribute. This attribute is managed dynamically by the Linked Attributes Plug-in instance. Whenever a change is made to the managed attribute, then the plug-in updates all of the linked attributes on the targeted entries.

Parameter	Description
Entry DN	<code>cn=plugin_instance,cn=Linked Attributes,cn=plugins,cn=config</code>
Valid Range	Any Directory Server attribute
Default Value	None
Syntax	DN
Example	<code>managedType: manager</code>

4.12. MANAGED ENTRIES PLUG-IN ATTRIBUTES

In some unique circumstances, it is useful to have an entry created automatically when another entry is created. For example, this can be part of Posix integration by creating a specific group entry when a new user is created. Each instance of the Managed Entries Plug-in identifies two areas:

- The scope of the plug-in, meaning the subtree and the search filter to use to identify entries which require a corresponding managed entry
- A template entry that defines what the managed entry should look like

4.12.1. managedBase

This attribute sets the subtree under which to create the managed entries. This can be any entry in the directory tree.

Parameter	Description
Entry DN	<code>cn=instance_name,cn=Managed Entries Plugin,cn=plugins,cn=config</code>
Valid Values	Any Directory Server subtree
Default Value	None
Syntax	DirectoryString
Example	<code>managedBase: ou=groups,dc=example,dc=com</code>

4.12.2. managedTemplate

This attribute identifies the template entry to use to create the managed entry. This entry can be located anywhere in the directory tree; however, it is recommended that this entry is in a replicated suffix so that all suppliers and consumers in replication are using the same template.

The attributes used to create the managed entry template are described in the [Red Hat Directory Server Configuration, Command, and File Reference](#).

Parameter	Description
Entry DN	<code>cn=instance_name,cn=Managed Entries Plugin,cn=plugins,cn=config</code>
Valid Values	Any Directory Server entry of the mepTemplateEntry object class
Default Value	None
Syntax	DirectoryString
Example	<code>managedTemplate: cn=My Template,ou=Templates,dc=example,dc=com</code>

4.12.3. originFilter

This attribute sets the search filter to use to search for and identify the entries within the subtree which require a managed entry. The filter allows the managed entries behavior to be limited to a specific type of entry or subset of entries. The syntax is the same as a regular search filter.

Parameter	Description
Entry DN	<code>cn=instance_name,cn=Managed Entries Plugin,cn=plugins,cn=config</code>

Parameter	Description
Valid Values	Any valid LDAP filter
Default Value	None
Syntax	DirectoryString
Example	originFilter: objectclass=posixAccount

4.12.4. originScope

This attribute sets the scope of the search to use to see which entries the plug-in monitors. If a new entry is created within the scope subtree, then the Managed Entries Plug-in creates a new managed entry that corresponds to it.

Parameter	Description
Entry DN	cn= <i>instance_name</i> ,cn=Managed Entries Plugin,cn=plugins,cn=config
Valid Values	Any Directory Server subtree
Default Value	None
Syntax	DirectoryString
Example	originScope: ou=people,dc=example,dc=com

4.13. MEMBEROF PLUG-IN ATTRIBUTES

Group membership is defined within group entries using attributes such as **member**. Searching for the **member** attribute makes it easy to list all of the members for the group. However, group membership is not reflected in the member's user entry, so it is impossible to tell to what groups a person belongs by looking at the user's entry.

The MemberOf Plug-in synchronizes the group membership in group members with the members' individual directory entries by identifying changes to a specific member attribute (such as **member**) in the group entry and then working back to write the membership changes over to a specific attribute in the members' user entries.

4.13.1. cn

Sets the name of the plug-in instance.

Parameter	Description
Entry DN	cn=MemberOf Plugin,cn=plugins,cn=config
Valid Values	Any valid string
Default Value	
Syntax	DirectoryString
Example	cn: Example MemberOf Plugin Instance

4.13.2. memberOfAllBackends

This attribute specifies whether to search the local suffix for user entries or all available suffixes. This can be desirable in directory trees where users may be distributed across multiple databases so that group membership is evaluated comprehensively and consistently.

Parameter	Description
Entry DN	cn=MemberOf Plugin,cn=plugins,cn=config
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	memberOfAllBackends: on

4.13.3. memberOfAttr

This attribute specifies the attribute in the user entry for the Directory Server to manage to reflect group membership. The MemberOf Plug-in generates the value of the attribute specified here in the directory entry for the member. There is a separate attribute for every group to which the user belongs.

Parameter	Description
Entry DN	cn=MemberOf Plugin,cn=plugins,cn=config
Valid Range	Any Directory Server attribute
Default Value	memberOf
Syntax	DirectoryString

Parameter	Description
Example	memberOfAttr: memberOf

4.13.4. memberOfAutoAddOC

To enable the **memberOf** plug-in to add the **memberOf** attribute to a user, the user object must contain an object class that allows this attribute. If an entry does not have an object class that allows the **memberOf** attribute then the **memberOf** plugin will automatically add the object class listed in the **memberOfAutoAddOC** parameter.

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	cn=MemberOf Plugin,cn=plugins,cn=config
Valid Values	Any Directory Server object class
Default Value	nsMemberOf
Syntax	DirectoryString
Example	memberOfAutoAddOC: nsMemberOf

4.13.5. memberOfEntryScope

If you configured several back ends or multiple-nested suffixes, the multi-valued **memberOfEntryScope** parameter enables you to set what suffixes the **MemberOf** plug-in works on. If the parameter is not set, the plug-in works on all suffixes. The value set in the **memberOfEntryScopeExcludeSubtree** parameter has a higher priority than values set in **memberOfEntryScope**.

For further details, see the corresponding section in the *Directory Server Administration Guide*.

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	cn=MemberOf Plugin,cn=plugins,cn=config
Valid Range	Any Directory Server entry DN.
Default Value	
Syntax	DirectoryString

Parameter	Description
Example	memberOfEntryScope: ou=people,dc=example,dc=com

4.13.6. memberOfEntryScopeExcludeSubtree

If you configured several back ends or multiple-nested suffixes, the multi-valued **memberOfEntryScopeExcludeSubtree** parameter enables you to set what suffixes the **MemberOf** plug-in excludes. The value set in the **memberOfEntryScopeExcludeSubtree** parameter has a higher priority than values set in **memberOfEntryScope**. If the scopes set in both parameters overlap, the **MemberOf** plug-in only works on the non-overlapping directory entries.

For further details, see the corresponding section in the *Directory Server Administration Guide*.

This setting does not require restarting the server to take effect.

Parameter	Description
Entry DN	cn=memberOf Plugin,cn=plugins,cn=config
Valid Range	Any Directory Server entry DN.
Default Value	
Syntax	DirectoryString
Example	memberOfEntryScopeExcludeSubtree: ou=sample,dc=example,dc=com

4.13.7. memberOfGroupAttr

This attribute specifies the attribute in the group entry to use to identify the DNs of group members. By default, this is the **member** attribute, but it can be any membership-related attribute that contains a DN value, such as **uniqueMember** or **member**.



NOTE

Any attribute can be used for the **memberOfGroupAttr** value, but the MemberOf Plug-in only works if the value of the target attribute contains the DN of the member entry. For example, the **member** attribute contains the DN of the member's user entry:

member: uid=jsmith,ou=People,dc=example,dc=com

Some member-related attributes do not contain a DN, like the **memberURL** attribute. That attribute will not work as a value for **memberOfGroupAttr**. The **memberURL** value is a URL, and a non-DN value cannot work with the MemberOf Plug-in.

Parameter	Description
Entry DN	cn=MemberOf Plugin,cn=plugins,cn=config
Valid Range	Any Directory Server attribute
Default Value	member
Syntax	DirectoryString
Example	memberOfGroupAttr: member

4.14. ATTRIBUTE UNIQUENESS PLUG-IN ATTRIBUTES

The **Attribute Uniqueness** plug-in ensures that the value of an attribute is unique across the directory or subtree.

4.14.1. cn

Sets the name of the **Attribute Uniqueness** plug-in configuration record. You can use any string, but Red Hat recommends naming the configuration record ***attribute_name* Attribute Uniqueness**.

Parameter	Description
Entry DN	cn=attribute_uniqueness_configuration_record_name, cn=plugins,cn=config
Valid Values	Any valid string
Default Value	None
Syntax	DirectoryString
Example	cn: mail Attribute Uniqueness

4.14.2. uniqueness-attribute-name

Sets the name of the attribute whose values must be unique. This attribute is multi-valued.

Parameter	Description
Entry DN	cn=attribute_uniqueness_configuration_record_name, cn=plugins,cn=config
Valid Values	Any valid attribute name

Parameter	Description
Default Value	None
Syntax	DirectoryString
Example	uniqueness-attribute-name: mail

4.14.3. uniqueness-subtrees

Sets the DN under which the plug-in checks for uniqueness of the attribute's value. This attribute is multi-valued.

Parameter	Description
Entry DN	<code>cn=attribute_uniqueness_configuration_record_name, cn=plugins,cn=config</code>
Valid Values	Any valid subtree DN
Default Value	None
Syntax	DirectoryString
Example	uniqueness-subtrees: ou=Sales,dc=example,dc=com

4.14.4. uniqueness-across-all-subtrees

If enabled (**on**), the plug-in checks that the attribute is unique across all subtrees set. If you set the attribute to **off**, uniqueness is only enforced within the subtree of the updated entry.

Parameter	Description
Entry DN	<code>cn=attribute_uniqueness_configuration_record_name, cn=plugins,cn=config</code>
Valid Values	on off
Default Value	off
Syntax	DirectoryString
Example	uniqueness-across-all-subtrees: off

4.14.5. uniqueness-top-entry-oc

Directory Server searches this object class in the parent entry of the updated object. If it was not found, the search continues at the next higher level entry up to the root of the directory tree. If the object class was found, Directory Server verifies that the value of the attribute set in **uniqueness-attribute-name** is unique in this subtree.

Parameter	Description
Entry DN	<code>cn=attribute_uniqueness_configuration_record_name, cn=plugins,cn=config</code>
Valid Values	Any valid object class
Default Value	None
Syntax	DirectoryString
Example	<code>uniqueness-top-entry-oc: nsContainer</code>

4.14.6. uniqueness-subtree-entries-oc

Optionally, when using the **uniqueness-top-entry-oc** parameter, you can configure that the **Attribute Uniqueness** plug-in only verifies if an attribute is unique, if the entry contains the object class set in this parameter.

Parameter	Description
Entry DN	<code>cn=attribute_uniqueness_configuration_record_name, cn=plugins,cn=config</code>
Valid Values	Any valid object class
Default Value	None
Syntax	DirectoryString
Example	<code>uniqueness-subtree-entries-oc: inetOrgPerson</code>

4.15. POSIX WINSYNC API PLUG-IN ATTRIBUTES

By default, Posix-related attributes are not synchronized between Active Directory and Red Hat Directory Server. On Linux systems, system users and groups are identified as Posix entries, and LDAP Posix attributes contain that required information. However, when Windows users are synced over, they have **ntUser** and **ntGroup** attributes automatically added which identify them as Windows accounts, but no Posix attributes are synced over (even if they exist on the Active Directory entry) and no Posix attributes are added on the Directory Server side.

The Posix Winsync API Plug-in synchronizes POSIX attributes between Active Directory and Directory Server entries.



NOTE

All POSIX attributes (such as **uidNumber**, **gidNumber**, and **homeDirectory**) are synchronized between Active Directory and Directory Server entries. However, if a new POSIX entry or POSIX attributes are added to an existing entry in the Directory Server, *only the POSIX attributes are synchronized over to the Active Directory corresponding entry*. The POSIX object class (**posixAccount** for users and **posixGroup** for groups) is not added to the Active Directory entry.

This plug-in is disabled by default and must be enabled before any Posix attributes will be synchronized from the Active Directory entry to the Directory Server entry.

4.15.1. posixWinsyncCreateMemberOfTask

This attribute sets whether to run the memberOf fix-up task immediately after a sync run in order to update group memberships for synced users. This is disabled by default because the memberOf fix-up task can be resource-intensive and cause performance issues if it is run too frequently.

Parameter	Description
Entry DN	cn=Posix Winsync API Plugin,cn=plugins,cn=config
Valid Range	true false
Default Value	false
Example	posixWinsyncCreateMemberOfTask: false

4.15.2. posixWinsyncLowerCaseUID

This attribute sets whether to store (and, if necessary, convert) the UID value in the **memberUID** attribute in lower case.

Parameter	Description
Entry DN	cn=Posix Winsync API Plugin,cn=plugins,cn=config
Valid Range	true false
Default Value	false
Example	posixWinsyncLowerCaseUID: false

4.15.3. posixWinsyncMapMemberUID

This attribute sets whether to map the **memberUID** attribute in an Active Directory group to the **uniqueMember** attribute in a Directory Server group.

Parameter	Description
Entry DN	cn=Posix Winsync API Plugin,cn=plugins,cn=config
Valid Range	true false
Default Value	true
Example	posixWinsyncMapMemberUID: false

4.15.4. posixWinsyncMapNestedGrouping

The **posixWinsyncMapNestedGrouping** parameter manages if nested groups are updated when **memberUID** attributes in an Active Directory POSIX group change. Updating nested groups is supported up a depth of five levels.

Parameter	Description
Entry DN	cn=Posix Winsync API Plugin,cn=plugins,cn=config
Valid Range	true false
Default Value	false
Example	posixWinsyncMapNestedGrouping: false

4.15.5. posixWinsyncMsSFUSchema

This attribute sets whether to the older Microsoft System Services for Unix 3.0 (msSFU30) schema when syncing Posix attributes from Active Directory. By default, the Posix Winsync API Plug-in uses Posix schema for modern Active Directory servers: 2005, 2008, and later versions. There are slight differences between the modern Active Directory Posix schema and the Posix schema used by Windows Server 2003 and older Windows servers. If an Active Directory domain is using the older-style schema, then the older-style schema can be used instead.

Parameter	Description
Entry DN	cn=Posix Winsync API Plugin,cn=plugins,cn=config
Valid Range	true false
Default Value	false
Example	posixWinsyncMsSFUSchema: true

4.16. RETRO CHANGELOG PLUG-IN ATTRIBUTES

Two different types of changelogs are maintained by Directory Server. The first type, referred to as simply a *changelog*, is used by multi-supplier replication, and the second changelog, a plug-in referred to as the *retro changelog*, is intended for use by LDAP clients for maintaining application compatibility with Directory Server 4.x versions.

This Retro Changelog Plug-in is used to record modifications made to a supplier server. When the supplier server's directory is modified, an entry is written to the Retro Changelog that contains both of the following:

- A number that uniquely identifies the modification. This number is sequential with respect to other entries in the changelog.
- The modification action; that is, exactly how the directory was modified.

It is through the Retro Changelog Plug-in that the changes performed to the Directory Server are accessed using searches to **cn=changelog** suffix.

4.16.1. isReplicated

This optional attribute sets a flag to indicate on a change in the changelog whether the change is newly made on that server or whether it was replicated over from another server.

Parameter	Description
OID	2.16.840.1.113730.3.1.2085
Entry DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
Valid Values	true false
Default Value	None
Syntax	Boolean
Example	isReplicated: true

4.16.2. nsslapd-attribute

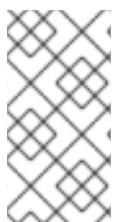
This attribute explicitly specifies another Directory Server attribute which must be included in the retro changelog entries.

Many operational attributes and other types of attributes are commonly excluded from the retro changelog, but these attributes may need to be present for a third-party application to use the changelog data. This is done by listing the attribute in the retro changelog plug-in configuration using the **nsslapd-attribute** parameter.

It is also possible to specify an optional alias for the specified attribute within the **nsslapd-attribute** value.

nsslapd-attribute: *attribute:alias*

Using an alias for the attribute can help avoid conflicts with other attributes in an external server or application which may use the retro changelog records.



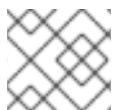
NOTE

Setting the value of the **nsslapd-attribute** attribute to **isReplicated** is a way of indicating, in the retro changelog entry itself, whether the modification was done on the local server (that is, whether the change is an original change) or whether the change was replicated over to the server.

Parameter	Description
Entry DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
Valid Values	Any valid directory attribute (standard or custom)
Default Value	None
Syntax	DirectoryString
Example	nsslapd-attribute: nsUniqueId: uniqueID

4.16.3. nsslapd-changelogdir

This attribute specifies the name of the directory in which the changelog database is created the first time the plug-in is run. By default, the database is stored with all the other databases under **/var/lib/dirsrv/slapd-*instance*/changelogdb**.



NOTE

For performance reasons, store this database on a different physical disk.

The server has to be restarted for changes to this attribute to go into effect.

Parameter	Description
Entry DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
Valid Values	Any valid path to the directory
Default Value	None
Syntax	DirectoryString
Example	nsslapd-changelogdir: /var/lib/dirsrv/slapd- <i>instance</i> /changelogdb

4.16.4. nsslapd-changelogmaxage (Max Changelog Age)

This attribute specifies the maximum age of any entry in the changelog. The changelog contains a record for each directory modification and is used when synchronizing consumer servers. Each record contains a timestamp. Any record with a timestamp that is older than the value specified in this attribute is removed. If this attribute is absent, there is no age limit on changelog records, which is the default behavior since this attribute is not present by default.



NOTE

Expired changelog records will not be removed if there is an agreement that has fallen behind further than the maximum age.

Parameter	Description
Entry DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
Valid Range	0 (meaning that entries are not removed according to their age) to the maximum 32 bit integer value (2147483647)
Default Value	0
Syntax	DirectoryString Integer AgeID AgeID is s for seconds, m for minutes, h for hours, d for days, or w for weeks.
Example	nsslapd-changelogmaxage: 30d

4.16.5. nsslapd-exclude-attrs

The **nsslapd-exclude-attrs** parameter stores an attribute name to exclude from the retro changelog database. To exclude multiple attributes, add one **nsslapd-exclude-attrs** parameter for each attribute to exclude.

Parameter	Description
Entry DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
Valid Values	Any valid attribute name
Default Value	None
Syntax	DirectoryString
Example	nsslapd-exclude-attrs: example

4.16.6. nsslapd-exclude-suffix

The **nsslapd-exclude-suffix** parameter stores a suffix to exclude from the retro changelog database. You can add the parameter multiple times to exclude multiple suffixes.

Parameter	Description
Entry DN	cn=Retro Changelog Plugin,cn=plugins,cn=config
Valid Values	Any valid attribute name
Default Value	None
Syntax	DirectoryString
Example	<code>nsslapd-exclude-suffix: ou=demo,dc=example,dc=com</code>

4.17. ROOTDN ACCESS CONTROL PLUG-IN ATTRIBUTES

The root DN, cn=Directory Manager, is a special user entry that is defined outside the normal user database. Normal access control rules are not applied to the root DN, but because of the powerful nature of the root user, it can be beneficial to apply some kind of access control rules to the root user.

The RootDN Access Control Plug-in sets normal access controls – host and IP address restrictions, time-of-day restrictions, and day of week restrictions – on the root user.

This plug-in is disabled by default.

4.17.1. rootdn-allow-host

This sets what hosts, by fully-qualified domain name, the root user is allowed to use to access the Directory Server. Any hosts not listed are implicitly denied.

Wild cards are allowed.

This attribute can be used multiple times to specify multiple hosts, domains, or subdomains.

Parameter	Description
Entry DN	cn=RootDN Access Control Plugin,cn=plugins,cn=config
Valid Range	Any valid host name or domain, including asterisks (*) for wildcards
Default Value	None
Syntax	DirectoryString
Example	<code>rootdn-allow-host: *.example.com</code>

4.17.2. rootdn-allow-ip

This sets what IP addresses, either IPv4 or IPv6, for machines the root user is allowed to use to access the Directory Server. Any IP addresses not listed are implicitly denied.

Wild cards are allowed.

This attribute can be used multiple times to specify multiple addresses, domains, or subnets.

Parameter	Description
Entry DN	cn=RootDN Access Control Plugin,cn=plugins,cn=config
Valid Range	Any valid IPv4 or IPv6 address, including asterisks (*) for wildcards
Default Value	None
Syntax	DirectoryString
Example	rootdn-allow-ip: 192.168..

4.17.3. rootdn-close-time

This sets part of a time period or range when the root user is allowed to access the Directory Server. This sets when the time-based access ends, when the root user is no longer allowed to access the Directory Server.

This is used in conjunction with the **rootdn-open-time** attribute.

Parameter	Description
Entry DN	cn=RootDN Access Control Plugin,cn=plugins,cn=config
Valid Range	Any valid time, in a 24-hour format
Default Value	None
Syntax	Integer
Example	rootdn-close-time: 1700

4.17.4. rootdn-days-allowed

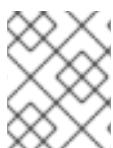
This gives a comma-separated list of what days the root user is allowed to use to access the Directory Server. Any days listed are implicitly denied. This can be used with **rootdn-close-time** and **rootdn-open-time** to combine time-based access and days-of-week or it can be used by itself (with all

hours allowed on allowed days).

Parameter	Description
Entry DN	cn=RootDN Access Control Plugin,cn=plugins,cn=config
Valid Values	* Sun * Mon * Tue * Wed * Thu * Fri * Sat
Default Value	None
Syntax	DirectoryString
Example	rootdn-days-allowed: Mon, Tue, Wed, Thu, Fri

4.17.5. rootdn-deny-ip

This sets what IP addresses, either IPv4 or IPv6, for machines the root user is *not* allowed to use to access the Directory Server. Any IP addresses not listed are implicitly allowed.



NOTE

Deny rules supercede allow rules, so if an IP address is listed in both the **rootdn-allow-ip** and **rootdn-deny-ip** attributes, it is denied access.

Wild cards are allowed.

This attribute can be used multiple times to specify multiple addresses, domains, or subnets.

Parameter	Description
Entry DN	cn=RootDN Access Control Plugin,cn=plugins,cn=config
Valid Range	Any valid IPv4 or IPv6 address, including asterisks (*) for wildcards

Parameter	Description
Default Value	None
Syntax	DirectoryString
Example	rootdn-deny-ip: 192.168.0.0

4.17.6. rootdn-open-time

This sets part of a time period or range when the root user is allowed to access the Directory Server. This sets when the time-based access *begins*.

This is used in conjunction with the **rootdn-close-time** attribute.

Parameter	Description
Entry DN	cn=RootDN Access Control Plugin,cn=plugins,cn=config
Valid Range	Any valid time, in a 24-hour format
Default Value	None
Syntax	Integer
Example	rootdn-open-time: 0800

CHAPTER 5. DIRECTORY ENTRY SCHEMA REFERENCE

5.1. ABOUT DIRECTORY SERVER SCHEMA

This chapter provides an overview of some of the basic concepts of the directory schema and lists the files in which the schema is described. It describes object classes, attributes, and object identifiers (OIDs) and briefly discusses extending server schema and schema checking.

5.1.1. Schema Definitions

The directory schema is a set of rules that defines how data can be stored in the directory. Directory information is stored discrete entries, and each entry is comprised of a set of attributes and their values. The kind of identity being described in the entry is defined in the entry's object classes. An object class specifies the kind of object the entry describes through the defined set of attributes for the object class.

Basically, the schema files are lists of the kinds of entries that can be created (the *object classes*) and the ways that those entries can be described (the *attributes*). The schema *defines* what the object classes and attributes are. The schema also defines the format that the attribute values contain (the attribute's *syntax*) and whether there can only be a single instance of that attribute.

Additional schema files can be added to the Directory Server configuration and loaded in the server, so the schema is customizable and can be extended as required.

For more detailed information about object classes, attributes, and how the Directory Server uses the schema, see the *Deployment Guide*.



WARNING

The Directory Server fails to start if the schema definitions contain too few or too many characters. Use exactly one space in those places where the LDAP standards allow the use of zero or many spaces; for example, the place between the NAME keyword and the name of an attribute type.

5.1.1.1. Object Classes

In LDAP, an object class defines the set of attributes that can be used to define an entry. The LDAP standard provides object classes for many common types of entries, such as people (**person** and **inetOrgPerson**), groups (**groupOfUniqueNames**), locations (**locality**), organizations and divisions (**organization** and **organizationalUnit**), and equipment (**device**).

In a schema file, an object class is identified by the **objectclasses** line, then followed by its OID, name, a description, its direct superior object class (an object class which is required to be used in conjunction with the object class and which shares its attributes with this object class), and the list of required (**MUST**) and allowed (**MAY**) attributes.

This is shown in [Example 5.1, “person Object Class Schema Entry”](#).

Example 5.1. person Object Class Schema Entry

```
objectClasses: ( 2.5.6.6 NAME 'person' DESC 'Standard LDAP objectclass' SUP top MUST ( sn $ cn ) MAY ( description $ seeAlso $ telephoneNumber $ userPassword ) X-ORIGIN 'RFC 2256' )
```

5.1.1.1. Required and Allowed Attributes

Every object class defines a number of required attributes and of allowed attributes. Required attributes must be present in entries using the specified object class, while allowed attributes are permissible and available for the entry to use, but are not required for the entry to be valid.

As in [Example 5.1, “person Object Class Schema Entry”](#), the **person** object class requires the **cn**, **sn**, and **objectClass** attributes and allows the **description**, **seeAlso**, **telephoneNumber**, and **userPassword** attributes.



NOTE

All entries require the **objectClass** attribute, which lists the object classes assigned to the entry.

5.1.1.2. Object Class Inheritance

An entry can have more than one object class. For example, the entry for a person is defined by the **person** object class, but the same person may also be described by attributes in the **inetOrgPerson** and **organizationalPerson** object classes.

Additionally, object classes can be hierarchical. An object class can inherit attributes from another class, in addition to its own required and allowed attributes. The second object class is the *superior* object class of the first.

The server’s object class structure determines the list of required and allowed attributes for a particular entry. For example, a user’s entry has to have the **inetOrgPerson** object class. In that case, the entry must also include the superior object class for **inetOrgPerson**, **organizationalPerson**, and the superior object class for **organizationalPerson**, which is **person**:

```
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
```

When the **inetOrgPerson** object class is assigned to an entry, the entry automatically inherits the required and allowed attributes from the superior object classes.

5.1.1.2. Attributes

Directory entries are composed of attributes and their values. These pairs are called *attribute-value assertions* or AVAs. Any piece of information in the directory is associated with a descriptive attribute. For instance, the **cn** attribute is used to store a person’s full name, such as **cn: John Smith**.

Additional attributes can supply additional information about John Smith:

```
givenname: John
surname: Smith
mail: jsmith@example.com
```

In a schema file, an attribute is identified by the **attributetypes** line, then followed by its OID, name, a description, syntax (allowed format for its value), optionally whether the attribute is single- or multi-valued, and where the attribute is defined.

This is shown in [Example 5.2, “description Attribute Schema Entry”](#).

Example 5.2. description Attribute Schema Entry

```
attributetypes: ( 2.5.4.13 NAME 'description' DESC 'Standard LDAP attribute type' SYNTAX
1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN 'RFC 2256' )
```

Some attributes can be abbreviated. These abbreviations are listed as part of the attribute definition:

```
attributetypes: ( 2.5.4.3 NAME ( 'cn' 'commonName' ) ...
```

5.1.1.2.1. Directory Server Attribute Syntaxes

The attribute’s syntax defines the format of the values which the attribute allows; as with other schema elements, the syntax is defined for an attribute using the syntax’s OID in the schema file entry. In the Directory Server Console, the syntax is referenced by its friendly name.

The Directory Server uses the attribute’s syntax to perform sorting and pattern matching on entries.

For more information about LDAP attribute syntaxes, see [RFC 4517](#).

Table 5.1. Supported LDAP Attribute Syntaxes

Name	OID	Definition
Binary	1.3.6.1.4.1.1466.115.121.1.5	<i>Deprecated. Use Octet string instead.</i>
Bit String	1.3.6.1.4.1.1466.115.121.1.6	For values which are bitstrings, such as '010111101'B .
Boolean	1.3.6.1.4.1.1466.115.121.1.7	For attributes with only two allowed values, TRUE or FALSE.
Country String	1.3.6.1.4.1.1466.115.121.1.11	For values which are limited to exactly two printable string characters; for example, US for the United States.
DN	1.3.6.1.4.1.1466.115.121.1.12	For values which are distinguished names (DNs).

Name	OID	Definition
Delivery Method	1.3.6.1.4.1.1466.115.121.1.14	For values which are contained a preferred method of delivering information or contacting an entity. The different values are separated by a dollar sign (\$). For example: [literal,subs="+quotes,verbatim"] telephone \$ physical
Directory String	1.3.6.1.4.1.1466.115.121.1.15	For values which are valid UTF-8 strings. These values are not necessarily case-insensitive. Both case-sensitive and case-insensitive matching rules are available for Directory String and related syntaxes.
Enhanced Guide	1.3.6.1.4.1.1466.115.121.1.21	For values which contain complex search parameters based on attributes and filters.
Facsimile	1.3.6.1.4.1.1466.115.121.1.22	For values which contain fax numbers.
Fax	1.3.6.1.4.1.1466.115.121.1.23	For values which contain the images of transmitted faxes.
Generalized Time	1.3.6.1.4.1.1466.115.121.1.24	For values which are encoded as printable strings. The time zone must be specified. It is strongly recommended to use GMT time.
Guide	1.3.6.1.4.1.1466.115.121.1.25	<i>Obsolete.</i> For values which contain complex search parameters based on attributes and filters.
IA5 String	1.3.6.1.4.1.1466.115.121.1.26	For values which are valid strings. These values are not necessarily case-insensitive. Both case-sensitive and case-insensitive matching rules are available for IA5 String and related syntaxes.
Integer	1.3.6.1.4.1.1466.115.121.1.27	For values which are whole numbers.
JPEG	1.3.6.1.4.1.1466.115.121.1.28	For values which contain image data.

Name	OID	Definition
Name and Optional UID	1.3.6.1.4.1.1466.115.121.1.34	For values which contain a combination value of a DN and (optional) unique ID.
Numeric String	1.3.6.1.4.1.1466.115.121.1.36	For values which contain a string of both numerals and spaces.
OctetString	1.3.6.1.4.1.1466.115.121.1.40	For values which are binary; this replaces the binary syntax.
Object Class Description	1.3.6.1.4.1.1466.115.121.1.37	For values which contain object class definitions.
OID	1.3.6.1.4.1.1466.115.121.1.38	For values which contain OID definitions.
Postal Address	1.3.6.1.4.1.1466.115.121.1.41	<p>For values which are encoded in the format postal-address = dstring * ("\$" dstring). For example:</p> <pre>[literal,subs="+quotes,verbatim"] ... 1234 Main St.\$Raleigh, NC 12345\$USA</pre> <p>Each <i>dstring</i> component is encoded as a DirectoryString value. Backslashes and dollar characters, if they occur, are quoted, so that they will not be mistaken for line delimiters. Many servers limit the postal address to 6 lines of up to thirty characters.</p>
Printable String	1.3.6.1.4.1.1466.115.121.1.44	For values which contain printable strings.
Space-Insensitive String	2.16.840.1.113730.3.7.1	For values which contain space-insensitive strings.
TelephoneNumber	1.3.6.1.4.1.1466.115.121.1.50	For values which are in the form of telephone numbers. It is recommended to use telephone numbers in international form.
Teletex Terminal Identifier	1.3.6.1.4.1.1466.115.121.1.51	For values which contain an international telephone number.

Name	OID	Definition
Telex Number	1.3.6.1.4.1.1466.115.121.1.52	For values which contain a telex number, country code, and answerback code of a telex terminal.
URI		For values in the form of a URL, introduced by a string such as http:// , https:// , ftp:// , ldap:// , and ldaps:// . The URI has the same behavior as IA5 String. See RFC 4517 for more information on this syntax.

5.1.1.2.2. Single- and Multi-Valued Attributes

By default, most attributes are multi-valued. This means that an entry can contain the same attribute multiple times, with different values. For example:

```
dn: uid=jsmith,ou=marketing,ou=people,dc=example,dc=com
ou: marketing
ou: people
```

The **cn**, **tel**, and **objectclass** attributes, for example, all can have more than one value. Attributes that are single-valued – that is, only one instance of the attribute can be specified – are specified in the schema as only allowing a single value. For example, **uidNumber** can only have one possible value, so its schema entry has the term **SINGLE-VALUE**. If the attribute is multi-valued, there is no value expression.

5.1.2. Default Directory Server Schema Files

Template schema definitions for Directory Server are stored in the **/etc/dirsrv/schema** directory. These default schema files are used to generate the schema files for new Directory Server instances. Each server instance has its own instance-specific schema directory in **/etc/dirsrv/slappd-instance/schema**. The schema files in the instance directory are used only by that instance.

To modify the directory schema, create new attributes and new object classes in the instance-specific schema directory. Because the default schema is used for creating new instances and each individual instance has its own schema files, it is possible to have slightly different schema for each instance, matching the use of each instance.

Any custom attributes added using the Directory Server Console or LDAP commands are stored in the **99user.ldif** file; other custom schema files can be added to the **/etc/dirsrv/slappd-instance/schema** directory for each instance. Do not make any modifications with the standard files that come with Red Hat Directory Server.

For more information about how the Directory Server stores information and suggestions for planning directory schema, see the *Deployment Guide*.

Table 5.2. Schema Files

Schema File	Purpose
00core.ldif	Recommended core schema from the X.500 and LDAP standards (RFCs). This schema is used by the Directory Server itself for the instance configuration and to start the server instance.
01core389.ldif	Recommended core schema from the X.500 and LDAP standards (RFCs). This schema is used by the Directory Server itself for the instance configuration and to start the server instance.
02common.ldif	Standard-related schema from RFC 2256, LDAPv3, and standard schema defined by Directory Server which is used to configure entries.
05rfc2927.ldif	Schema from RFC 2927, "MIME Directory Profile for LDAP Schema."
05rfc4523.ldif	Schema definitions for X.509 certificates.
05rfc4524.ldif	Cosine LDAP/X.500 schema.
06inetorgperson.ldif	inetorgperson schema elements from RFC 2798, RFC 2079, and part of RFC 1274.
10rfc2307.ldif	Schema from RFC 2307, "An Approach for Using LDAP as a Network Information Service."
20subscriber.ldif	Common schema element for Directory Server-Nortel subscriber interoperability.
25java-object.ldif	Schema from RFC 2713, "Schema for Representing Java Objects in an LDAP Directory."
28pilot.ldif	Schema from the pilot RFCs, especially RFC 1274, that are no longer recommended for use in new deployments.
30ns-common.ldif	Common schema.
50ns-admin.ldif	Schemas used by the Administration Server.
50ns-certificate.ldif	Schemas used by Red Hat Certificate System.
50ns-directory.ldif	Schema used by legacy Directory Server 4.x servers.
50ns-mail.ldif	Schema for mail servers.

Schema File	Purpose
50ns-value.ldif	Schema for value items in Directory Server.
50ns-web.ldif	Schema for web servers.
60autofs.ldif	Object classes for automount configuration; this is one of several schema files used for NIS servers.
60eduperson.ldif	Schema elements for education-related people and organization entries.
60mozilla.ldif	Schema elements for Mozilla-related user profiles.
60nss-ldap.ldif	Schema elements for GSS-API service names.
60pam-plugin.ldif	Schema elements for integrating directory services with PAM modules.
60pureftpd.ldif	Schema elements for defining FTP user accounts.
60rfc2739.ldif	Schema elements for calendars and vCard properties.
60rfc3712.ldif	Schema elements for configuring printers.
60sabayon.ldif	Schema elements for defining sabayon user entries.
60sudo.ldif	Schema elements for defining sudo users and roles.
60trust.ldif	Schema elements for defining trust relationships for NSS or PAM.
99user.ldif	Custom schema elements added through the Directory Server Console.

5.1.3. Object Identifiers (OIDs)

All schema elements have object identifiers (OIDs) assigned to them, including attributes and object classes. An OID is a sequence of integers, usually written as a dot-separated string. All custom attributes and classes must conform to the X.500 and LDAP standards.



WARNING

If an OID is not specified for a schema element, Directory Server automatically uses ***ObjectClass_name-oid*** and ***attribute_name-oid***. However, using text OIDs instead of numeric OIDs can lead to problems with clients, server interoperability, and server behavior, assigning a numeric OID is strongly recommended.

OIDs can be built on. The base OID is a root number which is used for every schema element for an organization, and then schema elements can be incremented from there. For example, a base OID could be **1**. The company then uses **1.1** for attributes, so every new attribute has an OID of **1.1.x**. It uses **1.2** for object classes, so every new object class has an OID of **1.2.x**.

For Directory Server-defined schema elements, the base OIDs are as follows:

- The Netscape base OID is **2.16.840.1.113730**.
- The Directory Server base OID is **2.16.840.1.113730.3**.
- All Netscape-defined attributes have the base OID **2.16.840.1.113370.3.1**.
- All Netscape-defined object classes have the base OID **2.16.840.1.113730.3.2**.

For more information about OIDs or to request a prefix, go to the Internet Assigned Number Authority (IANA) website at <http://www.iana.org/>.

5.1.4. Extending the Schema

The Directory Server schema includes hundreds of object classes and attributes that can be used to meet most of directory requirements. This schema can be extended with new object classes and attributes that meet evolving requirements for the directory service in the enterprise by creating custom schema files.

When adding new attributes to the schema, a new object class should be created to contain them. Adding a new attribute to an existing object class can compromise the Directory Server's compatibility with existing LDAP clients that rely on the standard LDAP schema and may cause difficulties when upgrading the server.

For more information about extending server schema, see the *Deployment Guide*.

5.1.5. Schema Checking

Schema checking means that the Directory Server checks every entry when it is created, modified, or in a database imported using LDIF to make sure that it complies with the schema definitions in the schema files. Schema checking verifies three things:

- Object classes and attributes used in the entry are defined in the directory schema.
- Attributes required for an object class are contained in the entry.
- Only attributes allowed by the object class are contained in the entry.

You should run Directory Server with schema checking turned on. For information on enabling schema checking, see the *Administration Guide*.

5.1.6. Syntax Validation

Syntax validation means that the Directory Server checks that the value of an attribute matches the required syntax for that attribute. For example, syntax validation will confirm that a new **telephoneNumber** attribute actually has a valid telephone number for its value.

With its basic configuration, syntax validation (like schema checking) will check any directory modification to make sure the attribute value matches the required syntax and will reject any modifications that violate the syntax. Optionally, syntax validation can be configured to log warning messages about syntax violations, and either reject the change or allow the modification process to succeed.

All syntaxes are validated against [RFC 4514](#), except for DNs. By default, DNs are validated against [RFC 1779](#) or [RFC 2253](#), which are less strict than [RFC 4514](#). Strict validation for DNs has to be explicitly configured.

This feature checks all attribute syntaxes listed in [Table 5.1, “Supported LDAP Attribute Syntaxes”](#), with the exception of binary syntaxes (which cannot be verified) and non-standard syntaxes, which do not have a defined required format. The *unvalidated* syntaxes are as follows:

- Fax (binary)
- OctetString (binary)
- JPEG (binary)
- Binary (non-standard)
- Space Insensitive String (non-standard)
- URI (non-standard)

When syntax validation is enabled, *new* attribute values are checked whenever an attribute is added or modified to an entry. (This does not include *replication* changes, since the syntax would have been checked on the supplier server.) It is also possible to check *existing* attribute values for syntax violations by running the **syntax-validation.pl** script.

For information on options for syntax validation, see the *Administration Guide*.

5.2. ENTRY ATTRIBUTE REFERENCE

The attributes listed in this reference are manually assigned or available to directory entries. The attributes are listed in alphabetical order with their definition, syntax, and OID.

5.2.1. abstract

The **abstract** attribute contains an abstract for a document entry.

OID	0.9.2342.19200300.102.1.9
Syntax	DirectoryString

Multi- or Single-Valued	Multi-valued
Defined in	Internet White Pages Pilot

5.2.2. accessTo

This attribute defines what specific hosts or servers a user is allowed to access.

OID	5.3.6.1.1.1.1
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	nss_ldap/pam_ldap

5.2.3. accountInactivityLimit

The **accountInactivityLimit** attribute sets the time period, in seconds, from the last login time of an account before that account is locked for inactivity.

OID	1.3.6.1.4.1.11.1.3.2.1.3
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

5.2.4. acctPolicySubentry

The **acctPolicySubentry** attribute identifies any entry which belongs to an account policy (specifically, an account lockout policy). The value of this attribute points to the account policy which is applied to the entry.

This can be set on an individual user entry or on a CoS template entry or role entry.

OID	1.3.6.1.4.1.11.1.3.2.1.2
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

5.2.5. administratorContactInfo

This attribute contains the contact information for the LDAP or server administrator.

OID	2.16.840.1.113730.3.1.74
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.6. adminRole

This attribute contains the role assigned to the user identified in the entry.

OID	2.16.840.1.113730.3.1.601
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape Administration Services

5.2.7. adminUrl

This attribute contains the URL of the Administration Server.

OID	2.16.840.1.113730.3.1.75
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.8. aliasedObjectName

The **aliasedObjectName** attribute is used by the Directory Server to identify alias entries. This attribute contains the DN (distinguished name) for the entry for which this entry is the alias. For example:

aliasedObjectName: uid=jdoe,ou=people,dc=example,dc=com

OID	2.5.4.1
Syntax	DN

Multi- or Single-Valued	Single-valued
Defined in	RFC 2256

5.2.9. associatedDomain

The **associatedDomain** attribute contains the DNS domain associated with the entry in the directory tree. For example, the entry with the distinguished name **c=US,o=Example Corporation** has the associated domain of **EC.US**. These domains should be represented in RFC 822 order.

associatedDomain:US

OID	0.9.2342.19200300.100.1.37
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.10. associatedName

The **associatedName** identifies an organizational directory tree entry associated with a DNS domain. For example:

associatedName: c=us

OID	0.9.2342.19200300.100.1.38
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.11. attributeTypes

This attribute is used in a schema file to identify an attribute defined within the subschema.

OID	2.5.21.5
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued

Defined in	RFC 2252
------------	--------------------------

5.2.12. audio

The **audio** attribute contains a sound file using a binary format. This attribute uses a **u-law** encoded sound data. For example:

`audio:: AAAAAA==`

OID	0.9.2342.19200300.100.1.55
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.13. authorCn

The **authorCn** attribute contains the common name of the document's author. For example:

`authorCn: John Smith`

OID	0.9.2342.19200300.102.1.11
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Internet White Pages Pilot

5.2.14. authorityRevocationList

The **authorityRevocationList** attribute contains a list of revoked CA certificates. This attribute should be requested and stored in a binary format, like **authorityRevocationList;binary**. For example:

`authorityrevocationlist;binary:: AAAAAA==`

OID	2.5.4.38
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.15. authorSn

The **authorSn** attribute contains the last name or family name of the author of a document entry. For example:

authorSn: Smith

OID	0.9.2342.19200300.102.1.12
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Internet White Pages Pilot

5.2.16. automountInformation

This attribute contains information used by the autofs automounter.



NOTE

The **automountInformation** attribute is defined in **60autofs.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **60autofs.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.33
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.17. bootFile

This attribute contains the boot image file name.



NOTE

The **bootFile** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.24
-----	----------------

Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2307

5.2.18. bootParameter

This attribute contains the value for **rpc.bootparamd**.



NOTE

The **bootParameter** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-*instance*/schema** directory.

OID	1.3.6.1.1.1.23
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2307

5.2.19. buildingName

The **buildingName** attribute contains the building name associated with the entry. For example:

buildingName: 14

OID	0.9.2342.19200300.100.1.48
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.20. businessCategory

The **businessCategory** attribute identifies the type of business in which the entry is engaged. The attribute value should be a broad generalization, such as a corporate division level. For example:

businessCategory: Engineering

OID	2.5.4.15
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.21. c (countryName)

The **countryName**, or **c**, attribute contains the two-character country code to represent the country names. The country codes are defined by the ISO. For example:

```
countryName: GB
c: US
```

OID	2.5.4.6
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	RFC 2256

5.2.22. cACertificate

The **cACertificate** attribute contains a CA certificate. The attribute should be requested and stored binary format, such as **cACertificate;binary**. For example:

```
cACertificate;binary:: AAAAAA==
```

OID	2.5.4.37
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.23. carLicense

The **carLicense** attribute contains an entry's automobile license plate number. For example:

```
carLicense: 6ABC246
```

OID	2.16.840.1.113730.3.1.1
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2798

5.2.24. certificateRevocationList

The **certificateRevocationList** attribute contains a list of revoked user certificates. The attribute value is to be requested and stored in binary form, as **certificateACertificate;binary**. For example:

```
certificateRevocationList;binary:: AAAAAA==
```

OID	2.5.4.39
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.25. cn (commonName)

The **commonName** attribute contains the name of an entry. For user entries, the **cn** attribute is typically the person's full name. For example:

```
commonName: John Smith
cn: Bill Anderson
```

With the **LDAPReplica** or **LDAPServerobject** object classes, the **cn** attribute value has the following format:

```
cn: replicater.example.com:17430/dc%3Dexample%2Cdc%3Com
```

OID	2.5.4.3
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.26. co (friendlyCountryName)

The **friendlyCountryName** attribute contains a country name; this can be any string. Often, the **country** is used with the ISO-designated two-letter country code, while the **co** attribute contains a readable country name. For example:

```
friendlyCountryName: Ireland
co: Ireland
```

OID	0.9.2342.19200300.100.1.43
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.27. cosAttribute

The **cosAttribute** contains the name of the attribute for which to generate a value for the CoS. There can be more than one **cosAttribute** value specified. This attribute is used by all types of CoS definition entries.

OID	2.16.840.1.113730.3.1.550
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.28. cosIndirectSpecifier

The **cosIndirectSpecifier** specifies the attribute values used by an indirect CoS to identify the template entry.

OID	2.16.840.1.113730.3.1.577
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

5.2.29. cosPriority

The **cosPriority** attribute specifies which template provides the attribute value when CoS templates compete to provide an attribute value. This attribute represents the global priority of a template. A priority of zero is the highest priority.

OID	2.16.840.1.113730.3.1.569
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

5.2.30. cosSpecifier

The **cosSpecifier** attribute contains the attribute value used by a classic CoS, which, along with the template entry's DN, identifies the template entry.

OID	2.16.840.1.113730.3.1.551
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

5.2.31. cosTargetTree

The **cosTargetTree** attribute defines the subtrees to which the CoS schema applies. The values for this attribute for the schema and for multiple CoS schema may overlap their target trees arbitrarily.

OID	2.16.840.1.113730.3.1.552
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

5.2.32. cosTemplateDn

The **cosTemplateDn** attribute contains the DN of the template entry which contains a list of the shared attribute values. Changes to the template entry attribute values are automatically applied to all the entries within the scope of the CoS. A single CoS might have more than one template entry associated with it.

OID	2.16.840.1.113730.3.1.553
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued

Defined in	Directory Server
------------	------------------

5.2.33. crossCertificatePair

The value for the **crossCertificatePair** attribute must be requested and stored in binary format, such as **certificateCertificateRepair;binary**. For example:

crossCertificatePair;binary:: AAAAAA==

OID	2.5.4.40
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.34. dc (domainComponent)

The **dc** attribute contains one component of a domain name. For example:

dc: example
domainComponent: example

OID	0.9.2342.19200300.100.1.25
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	RFC 2247

5.2.35. deltaRevocationList

The **deltaRevocationList** attribute contains a certificate revocation list (CRL). The attribute value is requested and stored in binary format, such as **deltaRevocationList;binary**.

OID	2.5.4.53
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.36. departmentNumber

The **departmentNumber** attribute contains an entry's department number. For example:

departmentNumber: 2604

OID	2.16.840.1.113730.3.1.2
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2798

5.2.37. description

The **description** attribute provides a human-readable description for an entry. For **person** or **organization** object classes, this can be used for the entry's role or work assignment. For example:

description: Quality control inspector for the ME2873 product line.

OID	2.5.4.13
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.38. destinationIndicator

The **destinationIndicator** attribute contains the city and country associated with the entry. This attribute was once required to provide public telegram service and is generally used in conjunction with the **registeredAddress** attribute. For example:

destinationIndicator: Stow, Ohio, USA

OID	2.5.4.27
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.39. displayName

The **displayName** attribute contains the preferred name of a person to use when displaying that person's entry. This is especially useful for showing the preferred name for an entry in a one-line summary list. Since other attribute types, such as **cn**, are multi-valued, they cannot be used to display a preferred name. For example:

displayName: John Smith

OID	2.16.840.1.113730.3.1.241
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	RFC 2798

5.2.40. dITRedirect

The **dITRedirect** attribute indicates that the object described by one entry now has a newer entry in the directory tree. This attribute may be used when an individual's place of work changes, and the individual acquires a new organizational DN.

dITRedirect: cn=jsmith,dc=example,dc=com

OID	0.9.2342.19200300.100.1.54
Syntax	DN
Defined in	RFC 1274

5.2.41. dmdName

The **dmdName** attribute value specifies a directory management domain (DMD), the administrative authority that operates the Directory Server.

OID	2.5.4.54
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	RFC 2256

5.2.42. dn (distinguishedName)

The **dn** attribute contains an entry's distinguished name. For example:

dn: uid=Barbara Jensen,ou=Quality Control,dc=example,dc=com

OID	2.5.4.49
Syntax	DN
Defined in	RFC 2256

5.2.43. dNSRecord

The **dNSRecord** attribute contains DNS resource records, including type A (Address), type MX (Mail Exchange), type NS (Name Server), and type SOA (Start of Authority) resource records. For example:

dNSRecord: IN NS ns.uu.net

OID	0.9.2342.19200300.100.1.26
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Internet Directory Pilot

5.2.44. documentAuthor

The **documentAuthor** attribute contains the DN of the author of a document entry. For example:

documentAuthor: uid=Barbara.Jensen,ou=People,dc=example,dc=com

OID	0.9.2342.19200300.100.1.14
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.45. documentIdentifier

The **documentIdentifier** attribute contains a unique identifier for a document. For example:

documentIdentifier: L3204REV1

OID	0.9.2342.19200300.100.1.11
Syntax	DirectoryString

Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.46. documentLocation

The **documentLocation** attribute contains the location of the original version of a document. For example:

documentLocation: Department Library

OID	0.9.2342.19200300.100.1.15
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.47. documentPublisher

The **documentPublisher** attribute contains the person or organization who published a document. For example:

documentPublisher: Southeastern Publishing

OID	0.9.2342.19200300.100.1.56
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	RFC 1274

5.2.48. documentStore

The **documentStore** attribute contains information on where the document is stored.

OID	0.9.2342.19200300.102.1.10
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Internet White Pages Pilot

5.2.49. **documentTitle**

The **documentTitle** attribute contains a document's title. For example:

documentTitle: Red Hat Directory Server Administrator Guide

OID	0.9.2342.19200300.100.1.12
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.50. **documentVersion**

The **documentVersion** attribute contains the current version number for the document. For example:

documentVersion: 1.1

OID	0.9.2342.19200300.100.1.13
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.51. **drink (favouriteDrink)**

The **favouriteDrink** attribute contains a person's favorite beverage. This can be shortened to **drink**. For example:

favouriteDrink: iced tea
drink: cranberry juice

OID	0.9.2342.19200300.100.1.5
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.52. **dSAQuality**

The **dSAQuality** attribute contains the rating of the directory system agents' (DSA) quality. This attribute allows a DSA manager to indicate the expected level of availability of the DSA. For example:

dSAQuality: high

OID	0.9.2342.19200300.100.1.49
Syntax	Directory-String
Multi- or Single-Valued	Single-valued
Defined in	RFC 1274

5.2.53. employeeNumber

The **employeeNumber** attribute contains the employee number for the person. For example:

employeeNumber: 3441

OID	2.16.840.1.113730.3.1.3
Syntax	Directory-String
Multi- or Single-Valued	Single-valued
Defined in	RFC 2798

5.2.54. employeeType

The **employeeType** attribute contains the employment type for the person. For example:

employeeType: Full time

OID	2.16.840.1.113730.3.1.4
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2798

5.2.55. enhancedSearchGuide

The **enhancedSearchGuide** attribute contains information used by an X.500 client to construct search filters. For example:

enhancedSearchGuide: (uid=bjensen)

OID	2.5.4.47
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2798

5.2.56. fax (facsimileTelephoneNumber)

The **facsimileTelephoneNumber** attribute contains the entry's facsimile number; this attribute can be abbreviated as **fax**. For example:

**facsimileTelephoneNumber: +1 415 555 1212
fax: +1 415 555 1212**

OID	2.5.4.23
Syntax	TelephoneNumber
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.57. gecos

The **gecos** attribute is used to determine the GECOS field for the user. This is comparable to the **cn** attribute, although using a **gecos** attribute allows additional information to be embedded in the GECOS field aside from the common name. This field is also useful if the common name stored in the directory is not the user's full name.

gecos: John Smith



NOTE

The **gecos** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.2
Syntax	DirectoryString

Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.58. generationQualifier

The **generationQualifier** attribute contains the generation qualifier for a person's name, which is usually appended as a suffix to the name. For example:

generationQualifier:III

OID	2.5.4.44
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.59. gidNumber

The **gidNumber** attribute contains a unique numeric identifier for a group entry or to identify the group for a user entry. This is analogous to the group number in Unix.

gidNumber: 100



NOTE

The **gidNumber** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.1
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.60. givenName

The **givenName** attribute contains an entry's given name, which is usually the first name. For example:

givenName: Rachel

OID	2.5.4.42
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.61. **homeDirectory**

The **homeDirectory** attribute contains the path to the user's home directory.

homeDirectory: /home/jsmith



NOTE

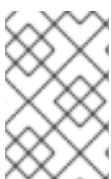
The **homeDirectory** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.3
Syntax	IA5String
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.62. **homePhone**

The **homePhone** attribute contains the entry's residential phone number. For example:

homePhone: 415-555-1234



NOTE

Although RFC 1274 defines both **homeTelephoneNumber** and **homePhone** as names for the residential phone number attribute, Directory Server only implements the **homePhone** name.

OID	0.9.2342.19200300.100.1.20
Syntax	TelephoneNumber
Multi- or Single-Valued	Multi-valued

Defined in	RFC 1274
------------	--------------------------

5.2.63. homePostalAddress

The **homePostalAddress** attribute contains an entry's home mailing address. Since this attribute generally spans multiple lines, each line break has to be represented by a dollar sign (\$). To represent an actual dollar sign (\$) or backslash (\) in the attribute value, use the escaped hex values \24 and \5c, respectively. For example:

homePostalAddress: 1234 Ridgeway Drive\$Santa Clara, CA\$99555

To represent the following string:

The dollar (\$) value can be found in the c:\cost file.

The entry value is:

The dollar (\24) value can be found\$in the c:\c5cost file.

OID	0.9.2342.19200300.100.1.39
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.64. host

The **host** contains the host name of a computer. For example:

host: labcontroller01

OID	0.9.2342.19200300.100.1.9
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.65. houseldentifier

The **houseldentifier** contains an identifier for a specific building at a location. For example:

houseldentifier: B105

OID	2.5.4.51
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.66. inetDomainBaseDN

This attribute identifies the base DN of user subtree for a DNS domain.

OID	2.16.840.1.113730.3.1.690
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	Subscriber interoperability

5.2.67. inetDomainStatus

This attribute shows the current status of the domain. A domain has a status of **active**, **inactive**, or **deleted**.

OID	2.16.840.1.113730.3.1.691
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Subscriber interoperability

5.2.68. inetSubscriberAccountId

This attribute contains the a unique attribute used to link the user entry for the subscriber to a billing system.

OID	2.16.840.1.113730.3.1.694
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Subscriber interoperability

5.2.69. **inetSubscriberChallenge**

The **inetSubscriberChallenge** attribute contains some kind of question or prompt, the challenge phrase, which is used to confirm the identity of the user in the **subscriberIdentity** attribute. This attribute is used in conjunction with the **inetSubscriberResponse** attribute, which contains the response to the challenge.

OID	2.16.840.1.113730.3.1.695
Syntax	IA5String
Multi- or Single-Valued	Single-valued
Defined in	Subscriber interoperability

5.2.70. **inetSubscriberResponse**

The **inetSubscriberResponse** attribute contains the answer to the challenge question in the **inetSubscriberChallenge** attribute to verify the user in the **subscriberIdentity** attribute.

OID	2.16.840.1.113730.3.1.696
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Subscriber interoperability

5.2.71. **inetUserHttpURL**

This attribute contains the web addresses associated with the user.

OID	2.16.840.1.113730.3.1.693
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Subscriber interoperability

5.2.72. **inetUserStatus**

This attribute shows the current status of the user (subscriber). A user has a status of **active**, **inactive**, or **deleted**.

OID	2.16.840.1.113730.3.1.692
Syntax	DirectoryString
Multi- or Single-Valued	Single-Valued
Defined in	Subscriber interoperability

5.2.73. info

The **info** attribute contains any general information about an object. Avoid using this attribute for specific information and rely instead on specific, possibly custom, attribute types. For example:

info: not valid

OID	0.9.2342.19200300.100.1.4
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.74. initials

The **initials** contains a person's initials; this does not contain the entry's surname. For example:

initials: BAJ

Directory Server and Active Directory handle the **initials** attribute differently. The Directory Server allows a practically unlimited number of characters, while Active Directory has a restriction of six characters. If an entry is synced with a Windows peer and the value of the **initials** attribute is longer than six characters, then the value is automatically truncated to six characters when it is synchronized. There is no information written to the error log to indicate that synchronization changed the attribute value, either.

OID	2.5.4.43
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.75. installationTimeStamp

This contains the time that the server instance was installed.

OID	2.16.840.1.113730.3.1.73
Syntax	DirectoryString
Multi- or Single-Valued	Multi-Valued
Defined in	Netscape Administration Services

5.2.76. internationalISDNNumber

The **internationalISDNNumber** attribute contains the ISDN number of a document entry. This attribute uses the internationally recognized format for ISDN addresses given in CCITT Rec. E. 164.

OID	2.5.4.25
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.77. ipHostNumber

This contains the IP address for a server.



NOTE

The **ipHostNumber** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.19
Syntax	DirectoryString
Multi- or Single-Valued	Multi-Valued
Defined in	RFC 2307

5.2.78. ipNetmaskNumber

This contains the IP netmask for the server.

**NOTE**

The **ipHostNumber** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	2.16.840.1.113730.3.1.73
Syntax	DirectoryString
Multi- or Single-Valued	Multi-Valued
Defined in	RFC 2307

5.2.79. ipNetworkNumber

This identifies the IP network.

**NOTE**

The **ipNetworkNumber** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.20
Syntax	DirectoryString
Multi- or Single-Valued	Single-Valued
Defined in	RFC 2307

5.2.80. ipProtocolNumber

This attribute identifies the IP protocol version number.

**NOTE**

The **ipProtocolNumber** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.17
-----	----------------

Syntax	Integer
Multi- or Single-Valued	Single-Valued
Defined in	RFC 2307

5.2.81. ipServicePort

This attribute gives the port used by the IP service.



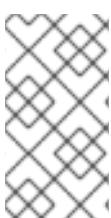
NOTE

The **ipServicePort** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-instance/schema** directory.

OID	1.3.6.1.1.1.15
Syntax	Integer
Multi- or Single-Valued	Single-Valued
Defined in	RFC 2307

5.2.82. ipServiceProtocol

This identifies the protocol used by the IP service.



NOTE

The **ipServiceProtocol** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-instance/schema** directory.

OID	1.3.6.1.1.1.16
Syntax	DirectoryString
Multi- or Single-Valued	Multi-Valued
Defined in	RFC 2307

5.2.83. janetMailbox

The **janetMailbox** contains a JANET email address, usually for users located in the United Kingdom who do not use RFC 822 email address. Entries with this attribute must also contain the **rfc822Mailbox** attribute.

OID	0.9.2342.19200300.100.1.46
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.84. jpegPhoto

The **jpegPhoto** attribute contains a JPEG photo, a binary value. For example:

```
jpegPhoto:: AAAAAA==
```

OID	0.9.2342.19200300.100.1.60
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2798

5.2.85. keyWords

The **keyWord** attribute contains keywords associated with the entry. For example:

```
keyWords: directory LDAP X.500
```

OID	0.9.2342.19200300.102.1.7
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Internet White Pages Pilot

5.2.86. knowledgeInformation

This attribute is no longer used.

OID	2.5.4.2
-----	---------

Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.87. I (localityName)

The **localityName**, or **I**, attribute contains the county, city, or other geographical designation associated with the entry. For example:

localityName: Santa Clara
I: Santa Clara

OID	2.5.4.7
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.88. labeledURI

The **labeledURI** contains a Uniform Resource Identifier (URI) which is related, in some way, to the entry. Values placed in the attribute should consist of a URI (currently only URLs are supported), optionally followed by one or more space characters and a label.

labeledURI: http://home.example.com
labeledURI: http://home.example.com Example website

OID	1.3.6.1.4.1.250.1.57
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2709

5.2.89. loginShell

The **loginShell** attribute contains the path to a script that is launched automatically when a user logs into the domain.

loginShell: c:\scripts\jsmith.bat

**NOTE**

The **loginShell** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.4
Syntax	IA5String
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.90. macAddress

This attribute gives the MAC address for a server or piece of equipment.

**NOTE**

The **macAddress** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.22
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2307

5.2.91. mail

The **mail** attribute contains a user's primary email address. This attribute value is retrieved and displayed by whitepage applications. For example:

mail: jsmith@example.com

OID	0.9.2342.19200300.100.1.3
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued

Defined in	RFC 1274
------------	--------------------------

5.2.92. mailAccessDomain

This attribute lists the domain which a user can use to access the messaging server.

OID	2.16.840.1.113730.3.1.12
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.93. mailAlternateAddress

The **mailAlternateAddress** attribute contains additional email addresses for a user. This attribute does not reflect the default or primary email address; that email address is set by the **mail** attribute.

For example:

```
mailAlternateAddress: jsmith@example.com
mailAlternateAddress: smith1701@alt.com
```

OID	2.16.840.1.113730.3.1.13
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.94. mailAutoReplyMode

This attribute sets whether automatic replies are enabled for the messaging server.

OID	2.16.840.1.113730.3.1.14
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.95. mailAutoReplyText

This attribute stores the text to used in an auto-reply email.

OID	2.16.840.1.113730.3.1.15
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.96. mailDeliveryOption

This attribute defines the mail delivery mechanism to use for the mail user.

OID	2.16.840.1.113730.3.1.16
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.97. mailEnhancedUniqueMember

This attribute contains the DN of a unique member of a mail group.

OID	2.16.840.1.113730.3.1.31
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.98. mailForwardingAddress

This attribute contains an email address to which to forward a user's email.

OID	2.16.840.1.113730.3.1.17
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.99. mailHost

The **mailHost** attribute contains the host name of a mail server. For example:

mailHost: mail.example.com

OID	2.16.840.1.113730.3.1.18
Syntax	DirctyString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.100. mailMessageStore

This identifies the location of a user's email box.

OID	2.16.840.1.113730.3.1.19
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.101. mailPreferenceOption

The **mailPreferenceOption** defines whether a user should be included on a mailing list, both electronic and physical. There are three options.

0	Does not appear in mailing lists.
1	Add to any mailing lists.
2	Added only to mailing lists which the provider views as relevant to the user interest.

If the attribute is absent, then the default is to assume that the user is not included on any mailing list. This attribute should be interpreted by anyone using the directory to derive mailing lists and its value respected. For example:

mailPreferenceOption: 0

OID	0.9.2342.19200300.100.1.47
-----	----------------------------

Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	RFC 1274

5.2.102. mailProgramDeliveryInfo

This attribute contains any commands to use for programmed mail delivery.

OID	2.16.840.1.113730.3.1.20
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.103. mailQuota

This attribute sets the amount of disk space allowed for a user's mail box.

OID	2.16.840.1.113730.3.1.21
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.104. mailRoutingAddress

This attribute contains the routing address to use when forwarding the emails received by the user to another messaging server.

OID	2.16.840.1.113730.3.1.24
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.105. manager

The **manager** contains the distinguished name (DN) of the manager for the person. For example:

manager: cn=Bill Andersen,ou=Quality Control,dc=example,dc=com

OID	0.9.2342.19200300.100.1.10
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.106. member

The **member** attribute contains the distinguished names (DNs) of each member of a group. For example:

member: cn=John Smith,dc=example,dc=com

OID	2.5.4.31
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.107. memberCertificateDescription

This attribute is a multi-valued attribute where each value is a description, a pattern, or a filter matching the subject DN of a certificate, usually a certificate used for TLS client authentication.

memberCertificateDescription matches any certificate that contains a subject DN with the same attribute-value assertions (AVAs) as the description. The description may contain multiple **ou** AVAs. A matching DN must contain those same **ou** AVAs, in the same order, although it may be interspersed with other AVAs, including other **ou** AVAs. For any other attribute type (not **ou**), there should be at most one AVA of that type in the description. If there are several, all but the last are ignored.

A matching DN must contain that same AVA but no other AVA of the same type nearer the root (later, syntactically).

AVAs are considered the same if they contain the same attribute description (case-insensitive comparison) and the same attribute value (case-insensitive comparison, leading and trailing whitespace ignored, and consecutive whitespace characters treated as a single space).

To be considered a member of a group with the following **memberCertificateDescription** value, a certificate needs to include **ou=x**, **ou=A**, and **dc=example**, but not **dc=company**.

memberCertificateDescription: {ou=x,ou=A,dc=company,dc=example}

To match the group's requirements, a certificate's subject DNs must contain the same **ou** attribute types in the same order as defined in the **memberCertificateDescription** attribute.

OID	2.16.840.1.113730.3.1.199
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.108. memberNisNetgroup

This attribute merges the attribute values of another netgroup into the current one by listing the name of the merging netgroup.



NOTE

The **memberNisNetgroup** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.13
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2307

5.2.109. memberOf

This attribute contains the name of a group to which the user is a member.

memberOf is the default attribute generated by the MemberOf Plug-in on the user entry of a group member. This attribute is automatically synchronized to the listed **member** attributes in a group entry, so that displaying group membership for entries is managed by Directory Server.



NOTE

This attribute is only synchronized between group entries and the corresponding members' user entries if the MemberOf Plug-in is enabled and is configured to use this attribute.

OID	1.2.840.113556.1.2.102
-----	------------------------

Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Delegated Administrator

5.2.110. memberUid

The **memberUid** attribute contains the login name of the member of a group; this can be different than the DN identified in the **member** attribute.

memberUID: jsmith



NOTE

The **memberUID** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.12
Syntax	IA5String
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.111. memberURL

This attribute identifies a URL associated with each member of a group. Any type of labeled URL can be used.

memberURL: ldap://cn=jsmith,ou=people,dc=example,dc=com

OID	2.16.840.1.113730.3.1.198
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.112. mepManagedBy

This attribute contains a pointer in an automatically-generated entry that points back to the DN of the originating entry. This attribute is set by the Managed Entries Plug-in and cannot be modified manually.

OID	2.16.840.1.113730.3.1.2086
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

5.2.113. mepManagedEntry

This attribute contains a pointer to an automatically-generated entry which corresponds to the current entry. This attribute is set by the Managed Entries Plug-in and cannot be modified manually.

OID	2.16.840.1.113730.3.1.2087
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

5.2.114. mepMappedAttr

This attribute sets an attribute in the Managed Entries template entry which must exist in the generated entry. The *mapping* means that some value of the originating entry is used to supply the given attribute. The values of these attributes will be tokens in the form *attribute: \$attr*. For example:

mepMappedAttr: gidNumber: \$gidNumber

As long as the syntax of the expanded token of the attribute does not violate the required attribute syntax, then other terms and strings can be used in the attribute. For example:

mepMappedAttr: cn: Managed Group for \$cn

OID	2.16.840.1.113730.3.1.2089
Syntax	OctetString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.115. mepRDNAAttr

This attribute sets which attribute to use as the naming attribute in the automatically-generated entry created by the Managed Entries Plug-in. Whatever attribute *type* is given in the naming attribute should be present in the managed entries template entry as a **mepMappedAttr**.

OID	2.16.840.1.113730.3.1.2090
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

5.2.116. mepStaticAttr

This attribute sets an attribute with a defined value that must be added to the automatically-generated entry managed by the Managed Entries Plug-in. This value will be used for every entry generated by that instance of the Managed Entries Plug-in.

mepStaticAttr: posixGroup

OID	2.16.840.1.113730.3.1.2088
Syntax	OctetString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.117. grpAddHeader

This attribute contains information about the header in the messages.

OID	2.16.840.1.113730.3.1.781
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.118. grpAllowedBroadcaster

This attribute sets whether to allow the user to send broadcast messages.

OID	2.16.840.1.113730.3.1.22
-----	--------------------------

Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.119. mgrpAllowedDomain

This attribute sets the domains for the mail group.

OID	2.16.840.1.113730.3.1.23
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.120. mgrpApprovePassword

This attribute sets whether a user must approve a password used to access their email.

OID	mgrpApprovePassword-oid
Syntax	IA5String
Multi- or Single-Valued	Single-valued
Defined in	Netscape Messaging Server

5.2.121. mgrpBroadcasterPolicy

This attribute defines the policy for broadcasting emails.

OID	2.16.840.1.113730.3.1.788
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.122. mgrpDeliverTo

This attribute contains information about the delivery destination for email.

OID	2.16.840.1.113730.3.1.25
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.123. mgrpErrorsTo

This attribute contains information about where to deliver error messages for the messaging server.

OID	2.16.840.1.113730.3.1.26
Syntax	IA5String
Multi- or Single-Valued	Single-valued
Defined in	Netscape Messaging Server

5.2.124. mgrpModerator

This attribute contains the contact name for the mailing list moderator.

OID	2.16.840.1.113730.3.1.33
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.125. mgrpMsgMaxSize

This attribute sets the maximum size allowed for email messages.

OID	2.16.840.1.113730.3.1.32
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape Messaging Server

5.2.126. mgrpMsgRejectAction

This attribute defines what actions the messaging server should take for rejected messages.

OID	2.16.840.1.113730.3.1.28
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.127. **mgrpMsgRejectText**

This attribute sets the text to use for rejection notifications.

OID	2.16.840.1.113730.3.1.29
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.128. **mgrpNoDuplicateChecks**

This attribute defines whether the messaging server checks for duplicate emails.

OID	2.16.840.1.113730.3.1.789
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape Messaging Server

5.2.129. **mgrpRemoveHeader**

This attribute sets whether the header is removed in reply messages.

OID	2.16.840.1.113730.3.1.801
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.130. grpRFC822MailMember

This attribute identifies the member of a mail group.

OID	2.16.840.1.113730.3.1.30
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.131. mobile

The **mobile**, or **mobileTelephoneNumber**, contains the entry's mobile or cellular phone number. For example:

mobileTelephoneNumber: 415-555-4321

OID	0.9.2342.19200300.100.1.41
Syntax	TelephoneNumber
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.132. mozillaCustom1

This attribute is used by Mozilla Thunderbird to manage a shared address book.

OID	1.3.6.1.4.1.13769.4.1
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.133. mozillaCustom2

This attribute is used by Mozilla Thunderbird to manage a shared address book.

OID	1.3.6.1.4.1.13769.4.2
Syntax	DirectoryString

Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.134. mozillaCustom3

This attribute is used by Mozilla Thunderbird to manage a shared address book.

OID	1.3.6.1.4.1.13769.4.3
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.135. mozillaCustom4

This attribute is used by Mozilla Thunderbird to manage a shared address book.

OID	1.3.6.1.4.1.13769.4.4
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.136. mozillaHomeCountryName

This attribute sets the country used by Mozilla Thunderbird in a shared address book.

OID	1.3.6.1.4.1.13769.3.6
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.137. mozillaHomeLocalityName

This attribute sets the city used by Mozilla Thunderbird in a shared address book.

OID	1.3.6.1.4.1.13769.3.3
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.138. mozillaHomePostalCode

This attribute sets the postal code used by Mozilla Thunderbird in a shared address book.

OID	1.3.6.1.4.1.13769.3.5
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.139. mozillaHomeState

This attribute sets the state or province used by Mozilla Thunderbird in a shared address book.

OID	1.3.6.1.4.1.13769.3.4
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.140. mozillaHomeStreet

This attribute sets the street address used by Mozilla Thunderbird in a shared address book.

OID	1.3.6.1.4.1.13769.3.1
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.141. mozillaHomeStreet2

This attribute contains the second line of a street address used by Mozilla Thunderbird in a shared address book.

OID	1.3.6.1.4.1.13769.3.2
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.142. mozillaHomeUrl

This attribute contains a URL used by Mozilla Thunderbird in a shared address book.

OID	1.3.6.1.4.1.13769.3.7
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.143. mozillaNickname (xmozillanickname)

This attribute contains a nickname used by Mozilla Thunderbird for a shared address book.

OID	1.3.6.1.4.1.13769.2.1
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Mozilla Address Book

5.2.144. mozillaSecondEmail (xmozillasecondemail)

This attribute contains an alternate or secondary email address for an entry in a shared address book for Mozilla Thunderbird.

OID	1.3.6.1.4.1.13769.2.2
Syntax	IA5String
Multi- or Single-Valued	Single-valued

Defined in	Mozilla Address Book
------------	----------------------

5.2.145. mozillaUseHtmlMail (xmozillausehtmlmail)

This attribute sets an email type preference for an entry in a shared address book in Mozilla Thunderbird.

OID	1.3.6.1.4.1.13769.2.3
Syntax	Boolean
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.146. mozillaWorkStreet2

This attribute contains a street address for a workplace or office for an entry in Mozilla Thunderbird's shared address book.

OID	1.3.6.1.4.1.13769.3.8
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.147. mozillaWorkUrl

This attribute contains a URL for a work site in an entry in a shared address book in Mozilla Thunderbird.

OID	1.3.6.1.4.1.13769.3.9
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Mozilla Address Book

5.2.148. multiLineDescription

This attribute contains a description of an entry which spans multiple lines in the LDIF file.

OID	1.3.6.1.4.1.250.1.2
-----	---------------------

Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Internet White Pages Pilot

5.2.149. name

The **name** attribute identifies the attribute supertype which can be used to form string attribute types for naming.

It is unlikely that values of this type will occur in an entry. LDAP server implementations that do not support attribute subtyping do not need to recognize this attribute in requests. Client implementations should not assume that LDAP servers are capable of performing attribute subtyping.

OID	2.5.4.41
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.150. netscapeReversiblePassword

This attribute contains the password for HTTP Digest/MD5 authentication.

OID	2.16.840.1.113730.3.1.812
Syntax	OctetString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Web Server

5.2.151. NisMapEntry

This attribute contains the information for a NIS map to be used by Network Information Services.



NOTE

This attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-instance/schema** directory.

OID	1.3.6.1.1.1.27
Syntax	IA5String
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

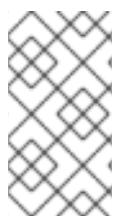
5.2.152. nisMapName

This attribute contains the name of a mapping used by a NIS server.

OID	1.3.6.1.1.1.26
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2307

5.2.153. nisNetgroupTriple

This attribute contains information on a netgroup used by a NIS server.



NOTE

This attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.14
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2307

5.2.154. nsAccessLog

This entry identifies the access log used by a server.

OID	nsAccessLog-oid
-----	-----------------

Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.155. nsAdminAccessAddresses

This attribute contains the IP address of the Administration Server used by the instance.

OID	nsAdminAccessAddresses-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.156. nsAdminAccessHosts

This attribute contains the host name of the Administration Server.

OID	nsAdminAccessHosts-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.157. nsAdminAccountInfo

This attribute contains other information about the Administration Server account.

OID	nsAdminAccountInfo-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.158. nsAdminCacheLifetime

This sets the length of time to store the cache used by the Directory Server.

OID	nsAdminCacheLifetime-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.159. nsAdminCgiWaitPid

This attribute defines the wait time for Administration Server CGI process IDs.

OID	nsAdminCgiWaitPid-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.160. nsAdminDomainName

This attribute contains the name of the administration domain containing the Directory Server instance.

OID	nsAdminDomainName-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.161. nsAdminEnableEnduser

This attribute sets whether to allow end user access to admin services.

OID	nsAdminEnableEnduser-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.162. nsAdminEndUserHTMLIndex

This attribute sets whether to allow end users to access the HTML index of admin services.

OID	nsAdminEndUserHTMLIndex-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.163. nsAdminGroupName

This attribute gives the name of the admin guide.

OID	nsAdminGroupName-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.164. nsAdminOneACLDir

This attribute gives the directory path to the directory containing access control lists for the Administration Server.

OID	nsAdminOneACLDir-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.165. nsAdminSIEDN

This attribute contains the DN of the server instance entry (SIE) for the Administration Server.

OID	nsAdminSIEDN-oid
Syntax	DN
Multi- or Single-Valued	Multi-valued

Defined in	Netscape Administration Services
------------	----------------------------------

5.2.166. nsAdminUsers

This attribute gives the path and name of the file which contains the information for the Administration Server admin user.

OID	nsAdminUsers-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.167. nsAIMid

This attribute contains the AOL Instant Messaging user ID for the user.

OID	2.16.840.1.113730.3.2.300
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.168. nsBaseDN

This contains the base DN used in the Directory Server's server instance definition entry.

OID	nsBaseDN-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.169. nsBindDN

This attribute contains the bind DN defined in the Directory Server SIE.

OID	nsBindDN-oid
-----	--------------

Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.170. nsBindPassword

This attribute contains the password used by the bind DN defined in **nsBindDN**.

OID	nsBindPassword-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.171. nsBuildNumber

This defines, in the Directory Server SIE, the build number of the server instance.

OID	nsBuildNumber-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.172. nsBuildSecurity

This defines, in the Directory Server SIE, the build security level.

OID	nsBuildSecurity-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.173. nsCertConfig

This attribute defines the configuration for the Red Hat Certificate System.

OID	nsCertConfig-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Certificate System

5.2.174. nsClassname

OID	nsClassname-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.175. nsConfigRoot

This attribute contains the root DN of the configuration directory.

OID	nsConfigRoot-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.176. nscpAIMScreenname

This attribute gives the AIM screen name of a user.

OID	1.3.6.1.4.1.13769.2.4
Syntax	TelephoneString
Multi- or Single-Valued	Multi-valued
Defined in	Mozilla Address Book

5.2.177. nsDefaultAcceptLanguage

This attribute contains the language codes which are accepted for HTML clients.

OID	nsDefaultAcceptLanguage-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.178. nsDefaultObjectClass

This attribute stores object class information in a container entry.

OID	nsDefaultObjectClass-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.179. nsDeleteclassname

OID	nsDeleteclassname-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.180. nsDirectoryFailoverList

This attribute contains a list of Directory Servers to use for failover.

OID	nsDirectoryFailoverList-oid
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.181. nsDirectoryInfoRef

This attribute refers to a DN of an entry with information about the server.

OID	nsDirectoryInfoRef-oid
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.182. nsDirectoryURL

This attribute contains the Directory Server URL.

OID	nsDirectoryURL-oid
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.183. nsDisplayName

This attribute contains a display name.

OID	nsDisplayName-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.184. nsErrorLog

This attribute identifies the error log used by the server.

OID	nsErrorLog-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.185. nsExecRef

This attribute contains the path or location of an executable which can be used to perform server tasks.

OID	nsExecRef-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.186. nsExpirationDate

This attribute contains the expiration date of an application.

OID	nsExpirationDate-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.187. nsGroupRDNComponent

This attribute defines the attribute to use for the RDN of a group entry.

OID	nsGroupRDNComponent-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.188. nsHardwarePlatform

This attribute indicates the hardware on which the server is running. The value of this attribute is the same as the output from **uname -m**. For example:

nsHardwarePlatform:i686

OID	nsHardwarePlatform-oid
Syntax	DirectoryString

Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.189. nsHelpRef

This attribute contains a reference to an online help file.

OID	nsHelpRef-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.190. nsHostLocation

This attribute contains information about the server host.

OID	nsHostLocation-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.191. nsICQid

This attribute contains an ICQ ID for the user.

OID	2.16.840.1.113730.3.1.2014
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.192. nsInstalledLocation

This attribute contains the installation directory for Directory Servers which are version 7.1 or older.

OID	nsInstalledLocation-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.193. nsJarfilename

This attribute gives the jar file name used by the Console.

OID	nsJarfilename-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.194. nsLdapSchemaVersion

This gives the version number of the LDAP directory schema.

OID	nsLdapSchemaVersion-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.195. nsLicensedFor

The **nsLicensedFor** attribute identifies the server the user is licensed to use. Administration Server expects each **nsLicenseUser** entry to contain zero or more instances of this attribute. Valid keywords for this attribute include the following:

- **slapd** for a licensed Directory Server client.
- **mail** for a licensed mail server client.
- **news** for a licensed news server client.
- **cal** for a licensed calendar server client.

For example:

nsLicensedFor: slapd

OID	2.16.840.1.113730.3.1.36
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Administration Server

5.2.196. nsLicenseEndTime

Reserved for future use.

OID	2.16.840.1.113730.3.1.38
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Administration Server

5.2.197. nsLicenseStartTime

Reserved for future use.

OID	2.16.840.1.113730.3.1.37
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Administration Server

5.2.198. nsLogSuppress

This attribute sets whether to suppress server logging.

OID	nsLogSuppress-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.199. nsmsgDisallowAccess

This attribute defines access to a messaging server.

OID	nsmsgDisallowAccess-oid
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.200. nsmsgNumMsgQuota

This attribute sets a quota for the number of messages which will be kept by the messaging server.

OID	nsmsgNumMsgQuota-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.201. nsMSNid

This attribute contains the MSN instant messaging ID for the user.

OID	2.16.840.1.113730.3.1.2016
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.202. nsNickName

This attribute gives a nickname for an application.

OID	nsNickName-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued

Defined in	Netscape
------------	----------

5.2.203. nsNYR

OID	nsNYR-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Administration Services

5.2.204. nsOsVersion

This attribute contains the version number of the operating system for the host on which the server is running.

OID	nsOsVersion-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.205. nsPidLog

OID	nsPidLog-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.206. nsPreference

This attribute stores the Console preference settings.

OID	nsPreference-oid
Syntax	DirectoryString

Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.207. nsProductName

This contains the name of the product, such as Red Hat Directory Server or Administration Server.

OID	nsProductName-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.208. nsProductVersion

This contains the version number of the Directory Server or Administration Server.

OID	nsProductVersion-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.209. nsRevisionNumber

This attribute contains the revision number of the Directory Server or Administration Server.

OID	nsRevisionNumber-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.210. nsSecureServerPort

This attribute contains the TLS port for the Directory Server.

**NOTE**

This attribute does not *configure* the TLS port for the Directory Server. This is configured in **nsslapd-secureport** configuration attribute in the Directory Server's **dse.ldif** file. Configuration attributes are described in the *Configuration, Command, and File Reference*.

OID	nsSecureServerPort-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.211. nsSerialNumber

This attribute contains a serial number or tracking number assigned to a specific server application, such as Red Hat Directory Server or Administration Server.

OID	nsSerialNumber-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.212. nsServerAddress

This attribute contains the IP address of the server host on which the Directory Server is running.

OID	nsServerAddress-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.213. nsServerCreationClassname

This attribute gives the class name to use when creating a server.

OID	nsServerCreationClassname-oid
-----	-------------------------------

Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.214. nsServerID

This contains the server's instance name. For example:

nsServerID: slapd-example

OID	nsServerID-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.215. nsServerMigrationClassname

This attribute contains the name of the class to use when migrating a server.

OID	nsServerMigrationClassname-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.216. nsServerPort

This attribute contains the standard LDAP port for the Directory Server.



NOTE

This attribute does not *configure* the standard port for the Directory Server. This is configured in **nsslapd-port** configuration attribute in the Directory Server's **dse.ldif** file. Configuration attributes are described in the *Configuration, Command, and File Reference*.

OID	nsServerPort-oid
-----	------------------

Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.217. nsServerSecurity

This shows whether the Directory Server requires a secure TLS or SSL connection.

OID	nsServerSecurity-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.218. nsSNMPContact

This attribute contains the contact information provided by the SNMP.

OID	2.16.840.1.113730.3.1.235
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.219. nsSNMPDescription

This contains a description of the SNMP service.

OID	2.16.840.1.113730.3.1.236
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.220. nsSNMPEnabled

This attribute shows whether SNMP is enabled for the server.

OID	2.16.840.1.113730.3.1.232
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.221. nsSNMPLocation

This attribute shows the location provided by the SNMP service.

OID	2.16.840.1.113730.3.1.234
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.222. nsSNMPMasterHost

This attribute shows the host name for the SNMP master agent.

OID	2.16.840.1.113730.3.1.237
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.223. nsSNMPMasterPort

This attribute shows the port number for the SNMP subagent.

OID	2.16.840.1.113730.3.1.238
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.224. nsSNMPOrganization

This attribute contains the organization information provided by SNMP.

OID	2.16.840.1.113730.3.1.233
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.225. nsSuiteSpotUser

This attribute has been obsoleted.

This attribute identifies the Unix user who installed the server.

OID	nsSuiteSpotUser-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.226. nsTaskLabel

OID	nsTaskLabel-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.227. nsUniqueAttribute

This sets a unique attribute for the server preferences.

OID	nsUniqueAttribute-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.228. nsUserIDFormat

This attribute sets the format to use to generate the **uid** attribute from the **givenname** and **sn** attributes.

OID	nsUserIDFormat-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.229. nsUserRDNComponent

This attribute sets the attribute type to set the RDN for user entries.

OID	nsUserRDNComponent-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.230. nsValueBin

OID	2.16.840.1.113730.3.1.247
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	Netscape servers – value item

5.2.231. nsValueCES

OID	2.16.840.1.113730.3.1.244
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Netscape servers – value item

5.2.232. nsValueCIS

OID	2.16.840.1.113730.3.1.243
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape servers – value item

5.2.233. nsValueDefault

OID	2.16.840.1.113730.3.1.250
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape servers – value item

5.2.234. nsValueDescription

OID	2.16.840.1.113730.3.1.252
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape servers – value item

5.2.235. nsValueDN

OID	2.16.840.1.113730.3.1.248
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Netscape servers – value item

5.2.236. nsValueFlags

OID	2.16.840.1.113730.3.1.251
-----	---------------------------

Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape servers – value item

5.2.237. nsValueHelpURL

OID	2.16.840.1.113730.3.1.254
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Netscape servers – value item

5.2.238. nsValueInt

OID	2.16.840.1.113730.3.1.246
Syntax	Integer
Multi- or Single-Valued	Multi-valued
Defined in	Netscape servers – value item

5.2.239. nsValueSyntax

OID	2.16.840.1.113730.3.1.253
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape servers – value item

5.2.240. nsValueTel

OID	2.16.840.1.113730.3.1.245
Syntax	TelephoneString
Multi- or Single-Valued	Multi-valued

Defined in	Netscape servers – value item
------------	-------------------------------

5.2.241. nsValueType

OID	2.16.840.1.113730.3.1.249
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape servers – value item

5.2.242. nsVendor

This contains the name of the server vendor.

OID	nsVendor-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape

5.2.243. nsViewConfiguration

This attribute stores the view configuration used by Console.

OID	nsViewConfiguration-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.244. nsViewFilter

This attribute sets the attribute-value pair which is used to identify entries belonging to the view.

OID	2.16.840.1.113730.3.1.3023
Syntax	IA5String

Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.245. nsWellKnownJarfiles

OID	nsWellKnownJarfiles-oid
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.246. nswmExtendedUserPrefs

This attribute is used to store user preferences for accounts in a messaging server.

OID	2.16.840.1.113730.3.1.520
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.247. nsYIMid

This attribute contains the Yahoo instant messaging user name for the user.

OID	2.16.840.1.113730.3.1.2015
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

5.2.248. ntGroupAttributes

This attribute points to a binary file which contains information about the group. For example:

```
ntGroupAttributes:: lyEvYmluL2tzaAoKlwojIGRIZmF1bHQgdmFsdWUKlwpIPSJgaG9zdG5hb
```

OID	2.16.840.1.113730.3.1.536
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.249. ntGroupCreateNewGroup

The **ntGroupCreateNewGroup** attribute is used by Windows Sync to determine whether the Directory Server should create new group entry when a new group is created on a Windows server. **true** creates the new entry; **false** ignores the Windows entry.

OID	2.16.840.1.113730.3.1.45
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.250. ntGroupDeleteGroup

The **ntGroupDeleteGroup** attribute is used by Windows Sync to determine whether the Directory Server should delete a group entry when the group is deleted on a Windows sync peer server. **true** means the account is deleted; **false** ignores the deletion.

OID	2.16.840.1.113730.3.1.46
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.251. ntGroupDomainId

The **ntGroupDomainID** attribute contains the domain ID string for a group.

ntGroupDomainId: DS HR Group

OID	2.16.840.1.113730.3.1.44
Syntax	DirectoryString

Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.252. ntGroupId

The **ntGroupId** attribute points to a binary file which identifies the group. For example:

ntGroupId: IOUnHNjjRgghghREgfvItrGHyuTYhjIOhTYtyHJuSDwOopKLhjGbnGFtr

OID	2.16.840.1.113730.3.1.110
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.253. ntGroupType

In Active Directory, there are two major types of groups: security and distribution. Security groups are most similar to groups in Directory Server, since security groups can have policies configured for access controls, resource restrictions, and other permissions. Distribution groups are for mailing distribution. These are further broken down into global and local groups. The Directory Server *ntGroupType* supports all four group types:

The **ntGroupType** attribute identifies the type of Windows group. The valid values are as follows:

- **-21483646** for global/security
- **-21483644** for domain local/security
- **2** for global/distribution
- **4** for domain local/distribution

This value is set automatically when the Windows groups are synchronized. To determine the type of group, you must manually configure it when the group gets created. By default, Directory Server groups do not have this attribute and are synchronized as global/security groups.

ntGroupType: -21483646

OID	2.16.840.1.113730.3.1.47
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued

Defined in	Netscape NT Synchronization
------------	-----------------------------

5.2.254. ntUniqueId

The **ntUniqueId** attribute contains a generated number used for internal server identification and operation. For example:

ntUniqueId: 352562404224a44ab040df02e4ef500b

OID	2.16.840.1.113730.3.1.111
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.255. ntUserAcctExpires

This attribute indicates when the entry's Windows account will expire. This value is stored as a string in GMT format. For example:

ntUserAcctExpires: 20081015203415

OID	2.16.840.1.113730.3.1.528
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.256. ntUserAuthFlags

This attribute contains authorization flags set for the Windows account.

OID	2.16.840.1.113730.3.1.60
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.257. ntUserBadPwCount

This attribute sets the number of bad password failures are allowed before an account is locked.

OID	2.16.840.1.113730.3.1.531
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.258. ntUserCodePage

The **ntUserCodePage** attribute contains the code page for the user's language of choice. For example:

ntUserCodePage: AAAAAA==

OID	2.16.840.1.113730.3.1.533
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.259. ntUserComment

This attribute contains a text description or note about the user entry.

OID	2.16.840.1.113730.3.1.522
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.260. ntUserCountryCode

This attribute contains the two-character country code for the country where the user is located.

OID	2.16.840.1.113730.3.1.532
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued

Defined in	Netscape NT Synchronization
------------	-----------------------------

5.2.261. ntUserCreateNewAccount

The **ntUserCreateNewAccount** attribute is used by Windows Sync to determine whether the Directory Server should create a new user entry when a new user is created on a Windows server. **true** creates the new entry; **false** ignores the Windows entry.

OID	2.16.840.1.113730.3.1.42
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.262. ntUserDeleteAccount

The **ntUserDeleteAccount** attribute IS Used by Windows Sync to determine whether a Directory Server entry will be automatically deleted when the user is deleted from the Windows sync peer server. **true** means the user entry is deleted; **false** ignores the deletion.

OID	2.16.840.1.113730.3.1.43
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.263. ntUserDomainId

The **ntUserDomainId** attribute contains the Windows domain login ID. For example:

ntUserDomainId: jsmith

OID	2.16.840.1.113730.3.1.41
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.264. ntUserFlags

This attribute contains additional flags set for the Windows account.

OID	2.16.840.1.113730.3.1.523
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.265. ntUserHomeDir

The **ntUserHomeDir** attribute contains an ASCII string representing the Windows user's home directory. This attribute can be null. For example:

```
ntUserHomeDir: c:\jsmith
```

OID	2.16.840.1.113730.3.1.521
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.266. ntUserHomeDirDrive

This attribute contains information about the drive on which the user's home directory is stored.

OID	2.16.840.1.113730.3.1.535
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.267. ntUserLastLogoff

The **ntUserLastLogoff** attribute contains the time of the last logoff. This value is stored as a string in GMT format.

If security logging is turned on, then this attribute is updated on synchronization only if some other aspect of the user's entry has changed.

```
ntUserLastLogoff: 20201015203415Z
```

OID	2.16.840.1.113730.3.1.527
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.268. ntUserLastLogon

The **ntUserLastLogon** attribute contains the time that the user last logged into the Windows domain. This value is stored as a string in GMT format. If security logging is turned on, then this attribute is updated on synchronization only if some other aspect of the user's entry has changed.

ntUserLastLogon: 20201015203415Z

OID	2.16.840.1.113730.3.1.526
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.269. ntUserLogonHours

The **ntUserLogonHours** attribute contains the time periods that a user is allowed to log onto the Active Directory domain. This attribute corresponds to the **logonHours** attribute in Active Directory.

OID	2.16.840.1.113730.3.1.530
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.270. ntUserLogonServer

The **ntUserLogonServer** attribute defines the Active Directory server to which the user's logon request is forwarded.

OID	2.16.840.1.113730.3.1.65
Syntax	DirectoryString

Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.271. ntUserMaxStorage

The **ntUserMaxStorage** attribute contains the maximum amount of disk space available for the user.

ntUserMaxStorage: 4294967295

OID	2.16.840.1.113730.3.1.529
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.272. ntUserNumLogons

This attribute shows the number of successful logons to the Active Directory domain for the user.

OID	2.16.840.1.113730.3.1.64
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.273. ntUserParms

The **ntUserParms** attribute contains a Unicode string reserved for use by applications.

OID	2.16.840.1.113730.3.1.62
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.274. ntUserPasswordExpired

This attribute shows whether the password for the Active Directory account has expired.

OID	2.16.840.1.113730.3.1.68
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.275. ntUserPrimaryGroupId

The **ntUserPrimaryGroupId** attribute contains the group ID of the primary group to which the user belongs.

OID	2.16.840.1.113730.3.1.534
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.276. ntUserPriv

This attribute shows the type of privileges allowed for the user.

OID	2.16.840.1.113730.3.1.59
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.277. ntUserProfile

The **ntUserProfile** attribute contains the path to a user's profile. For example:

ntUserProfile: c:\jsmith\profile.txt

OID	2.16.840.1.113730.3.1.67
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued

Defined in	Netscape NT Synchronization
------------	-----------------------------

5.2.278. ntUserScriptPath

The **ntUserScriptPath** attribute contains the path to an ASCII script used by the user to log into the domain.

ntUserScriptPath: c:\jstorm\lscript.bat

OID	2.16.840.1.113730.3.1.524
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.279. ntUserUniqueId

The **ntUserUniqueId** attribute contains a unique numeric ID for the Windows user.

OID	2.16.840.1.113730.3.1.66
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.280. ntUserUnitsPerWeek

The **ntUserUnitsPerWeek** attribute contains the total amount of time that the user has spent logged into the Active Directory domain.

OID	2.16.840.1.113730.3.1.63
Syntax	Binary
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.281. ntUserUsrComment

The **ntUserUsrComment** attribute contains additional comments about the user.

OID	2.16.840.1.113730.3.1.61
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.282. ntUserWorkstations

The **ntUserWorkstations** attribute contains a list of names, in ASCII strings, of work stations which the user is allowed to log in to. There can be up to eight work stations listed, separated by commas. Specify **null** to permit users to log on from any workstation. For example:

ntUserWorkstations: firefly

OID	2.16.840.1.113730.3.1.525
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape NT Synchronization

5.2.283. o (organizationName)

The **organizationName**, or **o**, attribute contains the organization name. For example:

organizationName: Example Corporation
o: Example Corporation

OID	2.5.4.10
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.284. objectClass

The **objectClass** attribute identifies the object classes used for an entry. For example:

objectClass: person

OID	2.5.4.0
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.285. objectClasses

This attribute is used in a schema file to identify an object class allowed by the subschema definition.

OID	2.5.21.6
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

5.2.286. obsoletedByDocument

The **obsoletedByDocument** attribute contains the distinguished name of a document which obsoletes the current document entry.

OID	0.9.2342.19200300.102.1.4
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Internet White Pages Pilot

5.2.287. obsoletesDocument

The **obsoletesDocument** attribute contains the distinguished name of a documented which is obsoleted by the current document entry.

OID	0.9.2342.19200300.102.1.3
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Internet White Pages Pilot

5.2.288. oncRpcNumber

The **oncRpcNumber** attribute contains part of the RPC map and stores the RPC number for UNIX RPCs.



NOTE

The **oncRpcNumber** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.18
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.289. organizationalStatus

The **organizationalStatus** identifies the person's category within an organization.

organizationalStatus: researcher

OID	0.9.2342.19200300.100.1.45
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.290. otherMailbox

The **otherMailbox** attribute contains values for email types other than X.400 and RFC 822.

otherMailbox: internet \$ jsmith@example.com

OID	0.9.2342.19200300.100.1.22
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued

Defined in

[RFC 1274](#)

5.2.291. ou (organizationalUnitName)

The **organizationalUnitName**, or **ou**, contains the name of an organizational division or a subtree within the directory hierarchy.

```
  organizationalUnitName: Marketing
  ou: Marketing
```

OID	2.5.4.11
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.292. owner

The **owner** attribute contains the DN of the person responsible for an entry. For example:

```
  owner: cn=John Smith,ou=people,dc=example,dc=com
```

OID	2.5.4.32
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.293. pager

The **pagerTelephoneNumber**, or **pager**, attribute contains a person's pager phone number.

```
  pagerTelephoneNumber: 415-555-6789
  pager: 415-555-6789
```

OID	0.9.2342.19200300.100.1.42
Syntax	TelephoneNumber
Multi- or Single-Valued	Multi-valued

Defined in	RFC 1274
------------	--------------------------

5.2.294. parentOrganization

The **parentOrganization** attribute identifies the parent organization of an organization or organizational unit.

OID	1.3.6.1.4.1.1466.101.120.41
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	Netscape

5.2.295. personalSignature

The **personalSignature** attribute contains the entry's signature file, in binary format.

personalSignature:: AAAAAA==

OID	0.9.2342.19200300.100.1.53
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.296. personalTitle

The **personalTitle** attribute contains a person's honorific, such as **Ms.**, **Dr.**, **Prof.**, and **Rev.**

personalTitle: Mr.

OID	0.9.2342.19200300.100.1.40
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.297. photo

The **photo** attribute contains a photo file, in a binary format.

photo:: AAAAAA==

OID	0.9.2342.19200300.100.1.7
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.298. physicalDeliveryOfficeName

The **physicalDeliveryOffice** contains the city or town in which a physical postal delivery office is located.

physicalDeliveryOfficeName: Raleigh

OID	2.5.4.19
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.299. postalAddress

The **postalAddress** attribute identifies the entry's mailing address. This field is intended to include multiple lines. When represented in LDIF format, each line should be separated by a dollar sign (\$).

To represent an actual dollar sign (\$) or backslash (\) within the entry text, use the escaped hex values \24 and \5c respectively. For example, to represent the string:

The dollar (\$) value can be found
in the c:\cost file.

provide the string:

The dollar (\24) value can be found\$in the c:\5ccost file.

OID	2.5.4.16
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued

Defined in	RFC 2256
------------	--------------------------

5.2.300. postalCode

The **postalCode** contains the zip code for an entry located within the United States.

postalCode: 44224

OID	2.5.4.17
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.301. postOfficeBox

The **postOfficeBox** attribute contains the postal address number or post office box number for an entry's physical mailing address.

postOfficeBox: 1234

OID	2.5.4.18
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.302. preferredDeliveryMethod

The **preferredDeliveryMethod** contains an entry's preferred contact or delivery method. For example:

preferredDeliveryMethod: telephone

OID	2.5.4.28
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.303. preferredLanguage

The **preferredLanguage** attribute contains a person's preferred written or spoken language. The value should conform to the syntax for HTTP Accept-Language header values.

OID	2.16.840.1.113730.3.1.39
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	RFC 2798

5.2.304. preferredLocale

A *locale* identifies language-specific information about how users of a specific region, culture, or custom expect data to be presented, including how data of a given language is interpreted and how data is to be sorted. Directory Server supports three locales for American English, Japanese, and German.

The **preferredLocale** attribute sets which locale is preferred by a user.

OID	1.3.6.1.4.1.1466.101.120.42
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape

5.2.305. preferredTimeZone

The **preferredTimeZone** attribute sets the time zone to use for the user entry.

OID	1.3.6.1.4.1.1466.101.120.43
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Netscape

5.2.306. presentationAddress

The **presentationAddress** attribute contains the OSI presentation address for an entry. This attribute includes the OSI Network Address and up to three selectors, one each for use by the transport, session, and presentation entities. For example:

presentationAddress: TELEX+00726322+RFC-1006+02+130.59.2.1

OID	2.5.4.29
Syntax	IA5String
Multi- or Single-Valued	Single-valued
Defined in	RFC 2256

5.2.307. protocolInformation

The **protocolInformation** attribute, used together with the **presentationAddress** attribute, provides additional information about the OSO network service.

OID	2.5.4.48
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.308. pwdReset

When an administrator changes the password of a user, Directory Server sets the **pwdReset** operational attribute in the user's entry to **true**. Applications can use this attribute to identify if a password of a user has been reset by an administrator.



NOTE

The **pwdReset** attribute is an operational attribute and, therefore, users cannot edit it.

OID	1.3.6.1.4.1.1466.115.121.1.7
Syntax	Boolean
Multi- or Single-Valued	Single-valued
Defined in	RFC draft-behera-ldap-password-policy

5.2.309. ref

The **ref** attribute is used to support LDAPv3 smart referrals. The value of this attribute is an LDAP URL:

ldap: host_name:port_number/subtree_dn

The port number is optional.

For example:

```
ref: ldap://server.example.com:389/ou=People,dc=example,dc=com
```

OID	2.16.840.1.113730.3.1.34
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	LDAPv3 Referrals Internet Draft

5.2.310. registeredAddress

This attribute contains a postal address for receiving telegrams or expedited documents. The recipient's signature is usually required on delivery.

OID	2.5.4.26
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.311. roleOccupant

This attribute contains the distinguished name of the person acting in the role defined in the **organizationalRole** entry.

```
roleOccupant: uid=bjensen,dc=example,dc=com
```

OID	2.5.4.33
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.312. roomNumber

This attribute specifies the room number of an object. The **cn** attribute should be used for naming room objects.

roomNumber: 230

OID	0.9.2342.19200300.100.1.6
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.313. searchGuide

The **searchGuide** attribute specifies information for suggested search criteria when using the entry as the base object in the directory tree for a search operation. When constructing search filters, use the **enhancedSearchGuide** attribute instead.

OID	2.5.4.14
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.314. secretary

The **secretary** attribute identifies an entry's secretary or administrative assistant.

secretary: cn=John Smith,dc=example,dc=com

OID	0.9.2342.19200300.100.1.21
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.315. seeAlso

The **seeAlso** attribute identifies another Directory Server entry that may contain information related to this entry.

seeAlso: cn=Quality Control Inspectors,ou=manufacturing,dc=example,dc=com

OID	2.5.4.34
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.316. serialNumber

The **serialNumber** attribute contains the serial number of a device.

serialNumber: 555-1234-AZ

OID	2.5.4.5
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.317. serverHostName

The **serverHostName** attribute contains the host name of the server on which the Directory Server is running.

OID	2.16.840.1.113730.3.1.76
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Red Hat Administration Services

5.2.318. serverProductName

The **serverProductName** attribute contains the name of the server product.

OID	2.16.840.1.113730.3.1.71
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued

Defined in	Red Hat Administration Services
------------	---------------------------------

5.2.319. serverRoot

This attribute is obsolete.

This attribute shows the installation directory (server root) of Directory Servers version 7.1 or older.

OID	2.16.840.1.113730.3.1.70
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Administration Services

5.2.320. serverVersionNumber

The **serverVersionNumber** attribute contains the server version number.

OID	2.16.840.1.113730.3.1.72
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Red Hat Administration Services

5.2.321. shadowExpire

The **shadowExpire** attribute contains the date that the shadow account expires. The format of the date is in the number days since EPOCH, in UTC. To calculate this on the system, run a command like the following, using **-d** for the current date and **-u** to specify UTC:

```
$ echo date -u -d 20100108 +%s /24/60/60 |bc
```

```
14617
```

The result (14617 in the example) is then the value of **shadowExpire**.

```
shadowExpire: 14617
```

**NOTE**

The **shadowExpire** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-*instance*/schema** directory.

OID	1.3.6.1.1.1.10
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.322. shadowFlag

The **shadowFlag** attribute identifies what area in the shadow map stores the flag values.

shadowFlag: 150

**NOTE**

The **shadowFlag** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-*instance*/schema** directory.

OID	1.3.6.1.1.1.11
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.323. shadowInactive

The **shadowInactive** attribute sets how long, in days, the shadow account can be inactive.

shadowInactive: 15

**NOTE**

The **shadowInactive** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-*instance*/schema** directory.

OID	1.3.6.1.1.1.9
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.324. shadowLastChange

The **shadowLastChange** attribute contains the number of days between January 1, 1970 and the day when the user password was last set. For example, if an account's password was last set on Nov 4, 2016, the **shadowLastChange** attribute is set to **0**

The following exceptions are existing:

- When the **passwordMustChange** parameter is enabled in the **cn=config** entry, new accounts have **0** set in the **shadowLastChange** attribute.
- When you create an account without password, the **shadowLastChange** attribute is not added.

The **shadowLastChange** attribute is automatically updated for accounts synchronized from Active Directory.



NOTE

The **shadowLastChange** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-instance/schema** directory.

OID	1.3.6.1.1.1.5
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.325. shadowMax

The **shadowMax** attribute sets the maximum number of days that a shadow password is valid.

shadowMax: 10

**NOTE**

The **shadowMax** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.7
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.326. shadowMin

The **shadowMin** attribute sets the minimum number of days that must pass between changing the shadow password.

shadowMin: 3

**NOTE**

The **shadowMin** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.6
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.327. shadowWarning

The **shadowWarning** attribute sets how many days in advance of password expiration to send a warning to the user.

shadowWarning: 2

**NOTE**

The **shadowWarning** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.8
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.328. singleLevelQuality

The **singleLevelQuality** specifies the purported data quality at the level immediately below in the directory tree.

OID	0.9.2342.19200300.100.1.50
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	RFC 1274

5.2.329. sn (surname)

The **surname**, or **sn**, attribute contains an entry's *surname*, also called a last name or family name.

```
surname: Jensen
sn: Jensen
```

OID	2.5.4.4
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.330. st (stateOrProvinceName)

The **stateOrProvinceName**, or **st**, attributes contains the entry's state or province.

stateOrProvinceName: California
st: California

OID	2.5.4.8
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.331. street

The **streetAddress**, or **street**, attribute contains an entry's street name and residential address.

streetAddress: 1234 Ridgeway Drive
street: 1234 Ridgeway Drive

OID	2.5.4.9
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.332. subject

The **subject** attribute contains information about the subject matter of the document entry.

subject: employee option grants

OID	0.9.2342.19200300.102.1.8
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Internet White Pages Pilot

5.2.333. subtreeMaximumQuality

The **subtreeMaximumQuality** attribute specifies the purported maximum data quality for a directory subtree.

OID	0.9.2342.19200300.100.1.52
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	RFC 1274

5.2.334. subtreeMinimumQuality

The **subtreeMinimumQuality** specifies the purported minimum data quality for a directory subtree.

OID	0.9.2342.19200300.100.1.51
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	RFC 1274

5.2.335. supportedAlgorithms

The **supportedAlgorithms** attribute contains algorithms which are requested and stored in a binary form, such as **supportedAlgorithms;binary**.

supportedAlgorithms:: AAAAAA==

OID	2.5.4.52
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.336. supportedApplicationContext

This attribute contains the identifiers of OSI application contexts.

OID	2.5.4.30
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued

Defined in	RFC 2256
------------	--------------------------

5.2.337. telephoneNumber

The **telephoneNumber** contains an entry's phone number. For example:

telephoneNumber: 415-555-2233

OID	2.5.4.20
Syntax	TelephoneNumber
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.338. teletexTerminalIdentifier

The **teletexTerminalIdentifier** attribute contains an entry's teletex terminal identifier. The first printable string in the example is the encoding of the first portion of the teletex terminal identifier to be encoded, and the subsequent 0 or more octet strings are subsequent portions of the teletex terminal identifier:

teletex-id = ttx-term 0*("\$" ttx-param)
ttx-term = printablestring
ttx-param = ttx-key ":" ttx-value
ttx-key = "graphic" / "control" / "misc" / "page" / "private"
ttx-value = octetstring

OID	2.5.4.22
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.339. telexNumber

This attribute defines the telex number of the entry. The format of the telex number is as follows:

actual-number "\$" country "\$" answerback

- *actual-number* is the syntactic representation of the number portion of the telex number being encoded.
- *country* is the TELEX country code.
- *answerback* is the answerback code of a TELEX terminal.

OID	2.5.4.21
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.340. title

The **title** attribute contains a person's title within the organization.

title: Senior QC Inspector

OID	2.5.4.12
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.341. ttl (TimeToLive)

The **TimeToLive**, or **ttl**, attribute contains the time, in seconds, that cached information about an entry should be considered valid. Once the specified time has elapsed, the information is considered out of date. A value of zero (**0**) indicates that the entry should not be cached.

TimeToLive: 120
ttl: 120

OID	1.3.6.1.4.250.1.60
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	LDAP Caching Internet Draft

5.2.342. uid (userID)

The **userID**, more commonly **uid**, attribute contains the entry's unique user name.

userID: jsmith
uid: jsmith

OID	0.9.2342.19200300.100.1.1
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.343. uidNumber

The **uidNumber** attribute contains a unique numeric identifier for a user entry. This is analogous to the user number in Unix.

uidNumber: 120



NOTE

The **uidNumber** attribute is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

OID	1.3.6.1.1.1.0
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	RFC 2307

5.2.344. uniquelIdentifier

This attribute identifies a specific item used to distinguish between two entries when a distinguished name has been reused. This attribute is intended to detect any instance of a reference to a distinguished name that has been deleted. This attribute is assigned by the server.

uniquelIdentifier:: AAAAAA==

OID	0.9.2342.19200300.100.1.44
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.345. uniqueMember

The **uniqueMember** attribute identifies a group of names associated with an entry where each name was given a **uniqueIdentifier** to ensure its uniqueness. A value for the **uniqueMember** attribute is a DN followed by the **uniqueIdentifier**.

OID	2.5.4.50
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.346. updatedByDocument

The **updatedByDocument** attribute contains the distinguished name of a document that is an updated version of the document entry.

OID	0.9.2342.19200300.102.1.6
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Internet White Pages Pilot

5.2.347. updatesDocument

The **updatesDocument** attribute contains the distinguished name of a document for which this document is an updated version.

OID	0.9.2342.19200300.102.1.5
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Internet White Pages Pilot

5.2.348. userCertificate

This attribute is stored and requested in the binary form, as **userCertificate;binary**.

```
userCertificate;binary:: AAAAAA==
```

OID	2.5.4.36
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.349. userClass

This attribute specifies a category of computer user. The semantics of this attribute are arbitrary. The **organizationalStatus** attribute makes no distinction between computer users and other types of users users and may be more applicable.

userClass: intern

OID	0.9.2342.19200300.100.1.8
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

5.2.350. userPassword

This attribute identifies the entry's password and encryption method in the format *{encryption method}encrypted password*. For example:

userPassword: {sha}FTSLQhxXpA05

Transferring cleartext passwords is strongly discouraged where the underlying transport service cannot guarantee confidentiality. Transferring in cleartext may result in disclosure of the password to unauthorized parties.

OID	2.5.4.35
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.351. userPKCS12

This attribute provides a format for the exchange of personal identity information. The attribute is stored and requested in binary form, as **userPKCS12;binary**. The attribute values are PFX PDUs stored as binary data.

OID	2.16.840.1.113730.3.1.216
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2798

5.2.352. userSMIMECertificate

The **userSMIMECertificate** attribute contains certificates which can be used by mail clients for S/MIME. This attribute requests and stores data in a binary format. For example:

```
userSMIMECertificate;binary:: AAAAAA==
```

OID	2.16.840.1.113730.3.1.40
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2798

5.2.353. vacationEndDate

This attribute shows the ending date of the user's vacation period.

OID	2.16.840.1.113730.3.1.708
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.354. vacationStartDate

This attribute shows the start date of the user's vacation period.

OID	2.16.840.1.113730.3.1.707
Syntax	DirectoryString

Multi- or Single-Valued	Multi-valued
Defined in	Netscape Messaging Server

5.2.355. x121Address

The **x121Address** attribute contains a user's X.121 address.

OID	2.5.4.24
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.2.356. x500UniqueId

Reserved for future use. An X.500 identifier is a binary method of identification useful for differentiating objects when a distinguished name has been reused.

x500UniqueId:: AAAAAA==

OID	2.5.4.45
Syntax	Binary
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2256

5.3. ENTRY OBJECT CLASS REFERENCE

This reference is an alphabetical list of the object classes accepted by the default schema. It gives a definition of each object class and lists its required and allowed attributes. The object classes listed are available to support entry information.

The required attributes listed for an object class must be present in the entry when that object class is added to the directory's **ldif** file. If an object class has a superior object class, both of these object classes with all required attributes must be present in the entry. If required attributes are not listed in the **ldif** file, than the server will not restart.



NOTE

The LDAP RFCs and X.500 standards allow for an object class to have more than one superior object class. This behavior is not currently supported by Directory Server.

5.3.1. account

The **account** object class defines entries for computer accounts. This object class is defined in [RFC 1274](#).

Superior Class

top

OID

0.9.2342.19200300.100.4.5

Table 5.3. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes for the entry.
Section 5.2.342, "uid (userID)"	Gives the defined account's user ID.

Table 5.4. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.64, "host"	Gives the host name for the machine on which the account resides.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the account belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the account belongs.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.

5.3.2. accountpolicy

The **accountpolicy** object class defines entries for account inactivation or expiration policies. This is used for a user directory configuration entry, which works in conjunction with the Account Policy Plug-in configuration.

Superior Class

top

OID

1.3.6.1.4.1.11.1.3.2.2.1

Table 5.5. Allowed Attributes

Attribute	Definition
Section 5.2.3, "accountInactivityLimit"	Sets the period, in seconds, from the last login time of an account before that account is locked for inactivity.

5.3.3. alias

The **alias** object class points to other directory entries. This object class is defined in [RFC 2256](#).

**NOTE**

Aliasing entries is not supported in Red Hat Directory Server.

Superior Class

top

OID

2.5.6.1

Table 5.6. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.
Section 5.2.8, "aliasedObjectName"	Gives the distinguished name of the entry for which this entry is an alias.

5.3.4. bootableDevice

The **bootableDevice** object class points to a device with boot parameters. This object class is defined in [RFC 2307](#).

**NOTE**

This object class is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-instance/schema** directory.

Superior Class

top

OID

1.3.6.1.1.2.12

Table 5.7. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the device.

Table 5.8. Allowed Attributes

Attribute	Definition
Section 5.2.17, "bootFile"	Gives the boot image file.
Section 5.2.18, "bootParameter"	Gives the parameters used by the boot process for the device.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the device belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the device belongs.
Section 5.2.292, "owner"	Gives the DN (distinguished name) of the person responsible for the device.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.316, "serialNumber"	Contains the serial number of the device.

5.3.5. cacheObject

The **cacheObject** is an object that contains the time to live (**ttl**) attribute type. This object class is defined in the LDAP Caching Internet Draft.

Superior Class

top

OID

1.3.6.1.4.1.250.3.18

Table 5.9. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.

Table 5.10. Allowed Attributes

Attribute	Definition
Section 5.2.341, "ttl (TimeToLive)"	The time that the object remains (lives) in the cache.

5.3.6. cosClassicDefinition

The **cosClassicDefinition** object class defines a class of service template entry using the entry's DN (distinguished name), given in the [Section 5.2.32, "cosTemplateDn"](#) attribute, and the value of one of the target attributes, specified in the [Section 5.2.30, "cosSpecifier"](#) attribute.

This object class is defined in [RFC 1274](#).

Superior Class

cosSuperDefinition

OID

2.16.840.1.113730.3.2.100

Table 5.11. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.27, "cosAttribute"	Provides the name of the attribute for which the CoS generates a value. There can be more than one cosAttribute value specified.

Table 5.12. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.30, "cosSpecifier"	Specifies the attribute value used by a classic CoS, which, along with the template entry's DN, identifies the template entry.
Section 5.2.32, "cosTemplateDn"	Provides the DN of the template entry which is associated with the CoS definition.

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.7. cosDefinition

The **cosDefinition** object class defines which class of service is being used; this object class provide compatibility with the DS4.1 CoS Plug-in.

This object class is defined in [RFC 1274](#).

Superior Class

top

OID

2.16.840.1.113730.3.2.84

Table 5.13. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.14. Allowed Attributes

Attribute	Definition
Section 6.2, "aci"	Evaluates what rights are granted or denied when the Directory Server receives an LDAP request from a client.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.27, "cosAttribute"	Provides the name of the attribute for which the CoS generates a value. There can be more than one cosAttribute value specified.
Section 5.2.30, "cosSpecifier"	Specifies the attribute value used by a classic CoS, which, along with the template entry's DN, identifies the template entry.
Section 5.2.31, "cosTargetTree"	Defines the subtrees in the directory to which the CoS schema applies.
Section 5.2.32, "cosTemplateDn"	Provides the DN of the template entry which is associated with the CoS definition.
Section 5.2.342, "uid (userID)"	Gives the user ID for the entry.

5.3.8. cosIndirectDefinition

The **cosIndirectDefinition** defines the template entry using the value of one of the target entry's attributes. The attribute of the target entry is specified in the [Section 5.2.28, "cosIndirectSpecifier"](#) attribute.

This object class is defined by Directory Server.

Superior Class

cosSuperDefinition

OID

2.16.840.1.113730.3.2.102

Table 5.15. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.27, "cosAttribute"	Provides the name of the attribute for which the CoS generates a value. There can be more than one cosAttribute value specified.

Table 5.16. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.28, "cosIndirectSpecifier"	Specifies the attribute value used by an indirect CoS to identify the template entry.
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.9. cosPointerDefinition

This object class identifies the template entry associated with the CoS definition using the template entry's DN value. The DN of the template entry is specified in the [Section 5.2.28, "cosIndirectSpecifier"](#) attribute.

This object class is defined by Directory Server.

Superior Class

cosSuperDefinition

OID

2.16.840.1.113730.3.2.101

Table 5.17. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.27, "cosAttribute"	Provides the name of the attribute for which the CoS generates a value. There can be more than one cosAttribute value specified.

Table 5.18. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.32, "cosTemplateDn"	Provides the DN of the template entry which is associated with the CoS definition.
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.10. cosSuperDefinition

All CoS definition object classes inherit from the **cosSuperDefinition** object class.

This object class is defined by Directory Server.

Superior Class

LDAPsubentry

OID

2.16.840.1.113730.3.2.99

Table 5.19. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.27, "cosAttribute"	Provides the name of the attribute for which the CoS generates a value. There can be more than one cosAttribute value specified.

Table 5.20. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.11. cosTemplate

The **cosTemplate** object class contains a list of the shared attribute values for the CoS.

This object class is defined by Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.128

Table 5.21. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.22. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.29, "cosPriority"	Specifies which template provides the attribute value when CoS templates compete to provide an attribute value.

5.3.12. country

The **country** object class defines entries which represent countries. This object class is defined in [RFC 2256](#).

Superior Class

top

OID

2.5.6.2

Table 5.23. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.21, "c (countryName)"	Contains the two-character code representing country names, as defined by ISO, in the directory.

Table 5.24. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.313, "searchGuide"	Specifies information for suggested search criteria when using the entry as the base object in the directory tree for a search.

5.3.13. dcObject

The **dcObject** object class allows domain components to be defined for an entry. This object class is defined as auxiliary because it is commonly used in combination with another object class, such as **o (organization)**, **ou (organizationalUnit)**, or **l (locality)**.

For example:

```
dn: dc=example,dc=com
objectClass: top
objectClass: organizationalUnit
objectClass: dcObject
dc: example
ou: Example Corporation
```

This object class is defined in [RFC 2247](#).

Superior Class

top

OID

1.3.6.1.4.1.1466.344

Table 5.25. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.34, "dc (domainComponent)"	Contains one component of a domain name.

5.3.14. device

The **device** object class stores information about network devices, such as printers, in the directory. This object class is defined in [RFC 2247](#).

Superior Class

top

OID

2.5.6.14

Table 5.26. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the device.
Section 5.2.25, "cn (commonName)"	Gives the common name of the device.

Table 5.27. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the device belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the device belongs.
Section 5.2.292, "owner"	Gives the DN (distinguished name) of the person responsible for the device.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.316, "serialNumber"	Contains the serial number of the device.

5.3.15. document

The **document** object class defines directory entries that represent documents. [RFC 1247](#).

Superior Class

top

OID

0.9.2342.19200300.100.4.6

Table 5.28. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.45, "documentIdentifier"	Gives the unique ID for the document.

Table 5.29. Allowed Attributes

Attribute	Definition
Section 5.2.1, "abstract"	Contains the abstract for the document.
Section 5.2.12, "audio"	Stores a sound file in binary format.
Section 5.2.13, "authorCn"	Gives the author's common name or given name.
Section 5.2.15, "authorSn"	Gives the author's surname.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.40, "dITRedirect"	Contains the DN (distinguished name) of the entry to use as a redirect for the document entry.
Section 5.2.44, "documentAuthor"	Contains the DN (distinguished name) of the author.
Section 5.2.46, "documentLocation"	Gives the location of the original document.
Section 5.2.47, "documentPublisher"	Identifies the person or organization that published the document.
Section 5.2.48, "documentStore"	
Section 5.2.49, "documentTitle"	Contains the title of the document.
Section 5.2.50, "documentVersion"	Gives the version number of the document.
Section 5.2.73, "info"	Contains information about the document.
Section 5.2.84, "jpegPhoto"	Stores a JPG image.
Section 5.2.85, "keyWords"	Contains keywords related to the document.

Attribute	Definition
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 6.13, "lastModifiedBy"	Gives the DN (distinguished name) of the last user which modified the document entry.
Section 6.14, "lastModifiedTime"	Gives the time of the last modification.
Section 5.2.105, "manager"	Gives the DN (distinguished name) of the entry's manager.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the document belongs.
Section 5.2.286, "obsoletedByDocument"	Gives the DN (distinguished name) of another document entry which <i>obsoletes</i> this document.
Section 5.2.287, "obsoletesDocument"	Gives the DN (distinguished name) of another document entry which is <i>obsoleted by</i> this document.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the document belongs.
Section 5.2.297, "photo"	Stores a photo of the document in binary format.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.332, "subject"	Describes the subject of the document.
Section 5.2.344, "uniqueIdentifier"	Distinguishes between two entries when a distinguished name has been reused.
Section 5.2.346, "updatedByDocument"	Gives the DN (distinguished name) of another document entry which <i>updates</i> this document.
Section 5.2.347, "updatesDocument"	Gives the DN (distinguished name) of another document entry which is <i>updated by</i> this document.

5.3.16. documentSeries

The **documentSeries** object class defines an entry that represents a series of documents. This object class is defined in [RFC 1274](#).

Superior Class

top

OID

0.9.2342.19200300.100.4.9

Table 5.30. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.31. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.87, "l (localityName)"	Gives the place where the document series is physically located.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the document series belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the series belongs.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.337, "telephoneNumber"	Gives the telephone number of the person responsible for the document series.

5.3.17. domain

The **domain** object class defines directory entries that represent DNS domains. Use the [Section 5.2.34, "dc \(domainComponent\)"](#) attribute to name entries of this object class.

This object class is also used for Internet domain names, such as **example.com**.

The **domain** object class can only be used for a directory entry which does *not* correspond to an organization, organizational unit, or any other object which has an object class defined for it. object for which an object class has been defined.

This object class is defined in [RFC 2252](#).

Superior Class

top

OID

0.9.2342.19200300.100.4.13

Table 5.32. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.34, "dc (domainComponent)"	Contains one component of a domain name.

Table 5.33. Allowed Attributes

Attribute	Definition
Section 5.2.10, "associatedName"	Gives the name of an entry within the organizational directory tree which is associated with a DNS domain.
Section 5.2.20, "businessCategory"	Gives the type of business in which this domain is engaged.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.38, "destinationIndicator"	Gives the country and city associated with the entry; this was once required to provide public telegram service.
Section 5.2.56, "fax (facsimileTelephoneNumber)"	Gives the fax number for the domain.
Section 5.2.76, "internationalISDNNumber"	Gives the ISDN number for the domain.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the entry belongs.
Section 5.2.298, "physicalDeliveryOfficeName"	Gives a location where physical deliveries can be made.
Section 5.2.301, "postOfficeBox"	Gives the post office box number for the domain.
Section 5.2.299, "postalAddress"	Contains the mailing address for the domain.
Section 5.2.300, "postalCode"	Gives the postal code for the domain, such as the zip code in the United States.
Section 5.2.302, "preferredDeliveryMethod"	Shows the person's preferred method of contact or message delivery.
Section 5.2.310, "registeredAddress"	Gives a postal address suitable to receive expedited documents when the recipient must verify delivery.

Attribute	Definition
Section 5.2.313, "searchGuide"	Specifies information for suggested search criteria when using the entry as the base object in the directory tree for a search.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.330, "st (stateOrProvinceName)"	Gives the state or province where the domain is located.
Section 5.2.331, "street"	Gives the street name and address number for the domain's physical location.
Section 5.2.337, "telephoneNumber"	Gives the phone number for the domain.
Section 5.2.338, "teletexTerminalIdentifier"	Gives the ID for a domain's teletex terminal.
Section 5.2.339, "telexNumber"	Gives the telex number for the domain.
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.
Section 5.2.355, "x121Address"	Gives the X.121 address for the domain.

5.3.18. domainRelatedObject

The **domainRelatedObject** object class defines entries that represent DNS or NRS domains which are equivalent to an X.500 domain, such as an organization or organizational unit.

This object class is defined in [RFC 1274](#).

Superior Class

top

OID

0.9.2342.19200300.100.4.17

Table 5.34. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.9, "associatedDomain"	Specifies a DNS domain associated with an object in the directory tree.

5.3.19. dSA

The **dSA** object class defines entries that represent DSAs.

This object class is defined in [RFC 1274](#).

Superior Class

top

OID

2.5.6.13

Table 5.35. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.306, "presentationAddress"	Contains the entry's OSI presentation address.

Table 5.36. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.86, "knowledgeInformation"	
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the entry belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.336, "supportedApplicationContext"	Contains the identifiers of OSI application contexts.

5.3.20. extensibleObject

When present in an entry, **extensibleObject** permits the entry to hold optionally any attribute. The allowed attribute list of this class is implicitly the set of all attributes known to the server.

This object class is defined in [RFC 2252](#).

Superior Class

top

OID

1.3.6.1.4.1.1466.101.120.111

Table 5.37. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Allowed Attributes

All attributes known to the server.

5.3.21. friendlyCountry

The **friendlyCountry** object class defines country entries within the directory. This object class allows more friendly names than the **country** object class.

This object class is defined in [RFC 1274](#).

Superior Class

top

OID

0.9.2342.19200300.100.4.18

Table 5.38. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.26, "co (friendlyCountryName)"	Stores the human-readable country name.
Section 5.2.21, "c (countryName)"	Contains the two-character code representing country names, as defined by ISO, in the directory.

Table 5.39. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.313, "searchGuide"	Specifies information for suggested search criteria when using the entry as the base object in the directory tree for a search.

5.3.22. groupOfCertificates

The **groupOfCertificates** object class describes a set of X.509 certificates. Any certificate that matches one of the [Section 5.2.107, "memberCertificateDescription"](#) values is considered a member of the group.

Superior Class

[top](#)

OID

2.16.840.1.113730.3.2.31

Table 5.40. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.41. Allowed Attributes

Attribute	Definition
Section 5.2.20, "businessCategory"	Gives the type of business in which the group is engaged.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.107, "memberCertificateDescription"	Contains the values used to determine if a particular certificate is a member of this group.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the entry belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.
Section 5.2.292, "owner"	Contains the DN (distinguished name) of the person responsible for the group.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.

5.3.23. groupOfMailEnhancedUniqueNames

The **groupOfMailEnhancedUniqueNames** object class is used for a mail group which must have unique members. This object class is defined for Netscape Messaging Server.

Superior Class

[top](#)

OID

2.16.840.1.113730.3.2.5

Table 5.42. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.43. Allowed Attributes

Attribute	Definition
Section 5.2.20, "businessCategory"	Gives the type of business in which the group is engaged.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.97, "mailEnhancedUniqueMember"	Contains a unique DN value to identify a member of the mail group.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the entry belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.
Section 5.2.292, "owner"	Contains the DN (distinguished name) of the person responsible for the group.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.

5.3.24. groupOfNames

The **groupOfNames** object class contains entries for a group of names. This object class is defined in [RFC 2256](#).

**NOTE**

The definition for this object class in Directory Server differs from the standard definition. In the standard definition, [Section 5.2.106, "member"](#) is a required attribute, while in Directory Server it is an allowed attribute. Directory Server, therefore, allows a group to have no members.

Superior Class

top

OID

2.5.6.9

Table 5.44. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.45. Allowed Attributes

Attribute	Definition
Section 5.2.20, "businessCategory"	Gives the type of business in which the entry is engaged.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.106, "member"	Contains the DN (distinguished name) of a group member.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the entry belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.
Section 5.2.292, "owner"	Contains the DN (distinguished name) of the person responsible for the group.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.

5.3.25. groupOfUniqueNames

The **groupOfUniqueNames** object class defines a group which contains unique names.

**NOTE**

The definition for this object class in Directory Server differs from the standard definition. In the standard definition, [Section 5.2.345, "uniqueMember"](#) is a required attribute, while in Directory Server it is an allowed attribute. Directory Server, therefore, allows a group to have no members.

This object class is defined in [RFC 2256](#).

Superior Class

top

OID

2.5.6.17

Table 5.46. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.47. Allowed Attributes

Attribute	Definition
Section 5.2.20, "businessCategory"	Gives the type of business in which the entry is engaged.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the entry belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.
Section 5.2.292, "owner"	Contains the DN (distinguished name) of the person responsible for the group.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.345, "uniqueMember"	Contains the DN (distinguished name) of a member of the group; this DN must be unique.

5.3.26. groupOfURLs

The **groupOfURLs** object class is an auxiliary object class for the **groupOfUniqueNames** and **groupOfNames** object classes. This group consists of a list of labeled URLs.

Superior Class

top

OID

2.16.840.1.113730.3.2.33

Table 5.48. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.49. Allowed Attributes

Attribute	Definition
Section 5.2.20, "businessCategory"	Gives the type of business in which the group is engaged.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.111, "memberURL"	Contains a URL associated with each member of the group.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the entry belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.
Section 5.2.292, "owner"	Contains the DN (distinguished name) of the person responsible for the group.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.

5.3.27. **ieee802Device**

The **ieee802Device** object class points to a device with a MAC address. This object class is defined in RFC 2307.



NOTE

This object class is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-instance/schema** directory.

Superior Class

top

OID

1.3.6.1.1.2.11

Table 5.50. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the device.

Table 5.51. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.90, "macAddress"	Gives the MAC address of the device.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the device belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the device belongs.
Section 5.2.292, "owner"	Gives the DN (distinguished name) of the person responsible for the device.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.316, "serialNumber"	Contains the serial number of the device.

5.3.28. inetAdmin

The **inetAdmin** object class is a marker for an administrative group or user. This object class is defined for the Netscape Delegated Administrator.

Superior Class

top

OID

2.16.840.1.113730.3.2.112

Table 5.52. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.53. Allowed Attributes

Attribute	Definition
Section 5.2.6, "adminRole"	Identifies a role to which the administrative user belongs.
Section 5.2.109, "memberOf"	Contains a group name to which the administrative user belongs. This is dynamically managed by the MemberOf Plug-in.

5.3.29. inetDomain

The **inetDomain** object class is a auxiliary class for virtual domain nodes. This object class is defined for the Netscape Delegated Administrator.

Superior Class

top

OID

2.16.840.1.113730.3.2.129

Table 5.54. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.55. Allowed Attributes

Attribute	Definition
Section 5.2.66, "inetDomainBaseDN"	Defines the base DN of the user subtree for a DNS domain.
Section 5.2.67, "inetDomainStatus"	Gives the status of the domain. The status can be active, inactive, or deleted.

5.3.30. inetOrgPerson

The **inetOrgPerson** object class defines entries representing people in an organization's enterprise network. This object class inherits the [Section 5.2.25, "cn \(commonName\)"](#) and [Section 5.2.329, "sn \(surname\)"](#) attributes from the **person** object class.

This object class is defined in [RFC 2798](#).

Superior Class

person

OID

2.16.840.1.113730.3.2.2

Table 5.56. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.329, "sn (surname)"	Gives the person's family name or last name.

Table 5.57. Allowed Attributes

Attribute	Definition
Section 5.2.12, "audio"	Stores a sound file in binary format.
Section 5.2.20, "businessCategory"	Gives the type of business in which the entry is engaged.
Section 5.2.23, "carLicense"	Gives the license plate number of the person's vehicle.
Section 5.2.36, "departmentNumber"	Gives the department for which the person works.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.38, "destinationIndicator"	Gives the country and city associated with the entry; this was once required to provide public telegram service.
Section 5.2.39, "displayName"	Shows the preferred name of a person to use when displaying entries.
Section 5.2.53, "employeeNumber"	Contains the person's employee number.
Section 5.2.54, "employeeType"	Shows the person's type of employment (for example, full time).
Section 5.2.56, "fax (facsimileTelephoneNumber)"	Contains the person's fax number.
Section 5.2.60, "givenName"	Contains the person's first name.
Section 5.2.62, "homePhone"	Gives the person's home phone number.
Section 5.2.63, "homePostalAddress"	Gives the person's home mailing address.

Attribute	Definition
Section 5.2.74, "initials"	Gives the person's initials.
Section 5.2.76, "internationalISDNNumber"	Contains the ISDN number for the entry.
Section 5.2.84, "jpegPhoto"	Stores a JPG image.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.88, "labeledURI"	Contains a URL which is relevant to the entry.
Section 5.2.91, "mail"	Contains the person's email address.
Section 5.2.105, "manager"	Contains the DN (distinguished name) of the direct supervisor of the person entry.
Section 5.2.131, "mobile"	Gives the person's mobile phone number.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the entry belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.
Section 5.2.293, "pager"	Gives the person's pager number.
Section 5.2.297, "photo"	Stores a photo of a person, in binary format.
Section 5.2.298, "physicalDeliveryOfficeName"	Gives a location where physical deliveries can be made.
Section 5.2.301, "postOfficeBox"	Gives the post office box number for the entry.
Section 5.2.299, "postalAddress"	Contains the mailing address for the entry.
Section 5.2.300, "postalCode"	Gives the postal code for the entry, such as the zip code in the United States.
Section 5.2.302, "preferredDeliveryMethod"	Shows the person's preferred method of contact or message delivery.
Section 5.2.303, "preferredLanguage"	Gives the person's preferred written or spoken language.
Section 5.2.310, "registeredAddress"	Gives a postal address suitable to receive expedited documents when the recipient must verify delivery.
Section 5.2.312, "roomNumber"	Gives the room number where the person is located.

Attribute	Definition
Section 5.2.314, "secretary"	Contains the DN (distinguished name) of the person's secretary or administrative assistant.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.330, "st (stateOrProvinceName)"	Gives the state or province where the entry is located.
Section 5.2.331, "street"	Gives the street name and number for the person's physical location.
Section 5.2.337, "telephoneNumber"	Gives the telephone number for the entry.
Section 5.2.338, "teletexTerminalIdentifier"	Gives the identifier for the person's teletex terminal.
Section 5.2.339, "telexNumber"	Gives the telex number associated with the entry.
Section 5.2.340, "title"	Shows the person's job title.
Section 5.2.342, "uid (userID)"	Contains the person's user ID (usually his logon ID).
Section 5.2.348, "userCertificate"	Stores a user's certificate in cleartext (not used).
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.
Section 5.2.352, "userSMIMECertificate"	Stores the person's certificate in binary form so it can be used by S/MIME clients.
Section 5.2.355, "x121Address"	Gives the X.121 address for the person.
Section 5.2.356, "x500UniquelIdentifier"	Reserved for future use.

5.3.31. **inetSubscriber**

The **inetSubscriber** object class is used for general user account management. This object class is defined for the Netscape subscriber interoperability.

Superior Class

top

OID

2.16.840.1.113730.3.2.134

Table 5.58. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.59. Allowed Attributes

Attribute	Definition
Section 5.2.68, "inetSubscriberAccountId"	Contains a unique attribute linking the subscriber to a billing system.
Section 5.2.69, "inetSubscriberChallenge"	Contains some kind of question or prompt, the challenge phrase, which is used to confirm the identity of the user.
Section 5.2.70, "inetSubscriberResponse"	Contains the answer to the challenge question.

5.3.32. **inetUser**

The **inetUser** object class is an auxiliary class which must be present in an entry in order to deliver subscriber services. This object class is defined for the Netscape subscriber interoperability.

Superior Class

top

OID

2.16.840.1.113730.3.2.130

Table 5.60. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.61. Allowed Attributes

Attribute	Definition
Section 5.2.71, "inetUserHttpURL"	Contains web addresses associated with the user.
Section 5.2.72, "inetUserStatus"	Gives the status of the user. The status can be active, inactive, or deleted.
Section 5.2.109, "memberOf"	Contains a group name to which the user belongs. This is dynamically managed by the MemberOf Plug-in.

Attribute	Definition
Section 5.2.342, "uid (userID)"	Contains the person's user ID (usually his logon ID).
Section 5.2.350, "userPassword"	Stores the password with which the user can use to access the user account.

5.3.33. ipHost

The **ipHost** object class stores IP information about a host. This object class is defined in [RFC 2307](#).



NOTE

This object class is defined in **10rfc2307.Idif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.Idif** file and copy the **10rfc2307bis.Idif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-instance/schema** directory.

Superior Class

top

OID

1.3.6.1.1.2.6

Table 5.62. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the device.
Section 5.2.77, "ipHostNumber"	Contains the IP address of the device or host.

Table 5.63. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.105, "manager"	Contains the DN (distinguished name) of the maintainer or supervisor of the entry.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the device belongs.

Attribute	Definition
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the device belongs.
Section 5.2.292, "owner"	Gives the DN (distinguished name) of the person responsible for the device.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.316, "serialNumber"	Contains the serial number of the device.

5.3.34. ipNetwork

The **ipNetwork** object class stores IP information about a network. This object class is defined in [RFC 2307](#).



NOTE

This object class is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-instance/schema** directory.

Superior Class

top

OID

1.3.6.1.1.2.7

Table 5.64. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the device.
Section 5.2.79, "ipNetworkNumber"	Contains the IP number for the network.

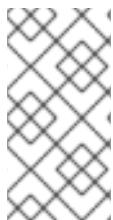
Table 5.65. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.

Attribute	Definition
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.105, "manager"	Contains the DN (distinguished name) of the maintainer or supervisor of the entry.
Section 5.2.78, "ipNetmaskNumber"	Contains the IP netmask for the network.

5.3.35. ipProtocol

The **ipProtocol** object class shows the IP protocol version. This object class is defined in [RFC 2307](#).



NOTE

This object class is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-instance/schema** directory.

Superior Class

top

OID

1.3.6.1.1.2.4

Table 5.66. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the device.
Section 5.2.80, "ipProtocolNumber"	Contains the IP protocol number for the network.

Table 5.67. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.36. ipService

The **ipService** object class stores information about the IP service. This object class is defined in [RFC 2307](#).

**NOTE**

This object class is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-instance/schema** directory.

Superior Class

top

OID

1.3.6.1.1.2.3

Table 5.68. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the device.
Section 5.2.81, "ipServicePort"	Gives the port number used by the IP service.
Section 5.2.82, "ipServiceProtocol"	Contains the IP protocol number for the service.

Table 5.69. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.37. labeledURIObject

This object class can be added to existing directory objects to allow URI values to be included. Using this object class does not preclude including the [Section 5.2.88, "labeledURI"](#) attribute type directly in other object classes as appropriate.

This object class is defined in [RFC 2079](#).

Superior Class

top

OID

1.3.6.1.4.1.250.3.15

Table 5.70. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.71. Allowed Attributes

Attribute	Definition
Section 5.2.88, "labeledURI"	Gives a URI which is relevant to the entry's object.

5.3.38. locality

The **locality** object class defines entries that represent localities or geographic areas.

This object class is defined in [RFC 2256](#).

Superior Class

top

OID

2.5.6.3

Table 5.72. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.73. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.313, "searchGuide"	Specifies information for suggested search criteria when using the entry as the base object in the directory tree for a search.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.330, "st (stateOrProvinceName)"	Gives the state or province associated with the locality.

Attribute	Definition
Section 5.2.331, "street"	Gives a street and number associated with the locality.

5.3.39. mailGroup

The **mailGroup** object class defines the mail attributes for a group. This object is defined in the schema for the Netscape Messaging Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.4

Table 5.74. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.75. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.91, "mail"	Stores email addresses for the group.
Section 5.2.93, "mailAlternateAddress"	Contains secondary email addresses for the group.
Section 5.2.99, "mailHost"	Contains the host name of the mail server.
Section 5.2.292, "owner"	Contains the DN (distinguished name) of the person responsible for the group.

5.3.40. mailRecipient

The **mailRecipient** object class defines a mail account for a user. This object is defined in the schema for the Netscape Messaging Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.3

Table 5.76. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.77. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.91, "mail"	Stores email addresses for the group.
Section 5.2.92, "mailAccessDomain"	Contains the domain from which the user can access the messaging server.
Section 5.2.93, "mailAlternateAddress"	Contains secondary email addresses for the group.
Section 5.2.94, "mailAutoReplyMode"	Specifies whether autoreply mode for the account is enabled.
Section 5.2.95, "mailAutoReplyText"	Contains the text use for automatic reply emails.
Section 5.2.96, "mailDeliveryOption"	Specifies the mail delivery mechanism to be used for the mail user.
Section 5.2.98, "mailForwardingAddress"	Specifies the mail delivery mechanism to use for the mail user.
Section 5.2.99, "mailHost"	Contains the host name of the mail server.
Section 5.2.100, "mailMessageStore"	Specifies the location of the user's mail box.
Section 5.2.102, "mailProgramDeliveryInfo"	Specifies the commands used for programmed mail delivery.
Section 5.2.103, "mailQuota"	Specifies the disk space allowed for the user's mail box.
Section 5.2.104, "mailRoutingAddress"	Contains a routing address to use when forwarding the mail from this entry's account to another messaging server.
Section 5.2.148, "multiLineDescription"	Contains a text description of the entry which spans more than one line.

Attribute	Definition
Section 5.2.342, "uid (userID)"	Gives the defined account's user ID.
Section 5.2.350, "userPassword"	Stores the password with which the entry can access the account.

5.3.41. mepManagedEntry

The **mepManagedEntry** object class identifies an entry which was generated by an instance of the Managed Entries Plug-in. This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.319

Table 5.78. Allowed Attributes

Attribute	Definition
Section 5.2.112, "mepManagedBy"	Gives the DN of the originating entry which corresponds to the managed entry.

5.3.42. mepOriginEntry

The **mepOriginEntry** object class identifies an entry which is within a subtree that is monitored by an instance of the Managed Entries Plug-in *and* which has had a managed entry created by the plug-in, for which this is the originating entry. This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.320

Table 5.79. Allowed Attributes

Attribute	Definition
Section 5.2.113, "mepManagedEntry"	Gives the DN of the managed entry entry which was created by the Managed Entries Plug-in instance and which corresponds to this originating entry.

5.3.43. mepTemplateEntry

The **mepTemplateEntry** object class identifies an entry which is used as a template by an instance of the Managed Entries Plug-in to create the managed entries. This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.321

Table 5.80. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.114, "mepMappedAttr"	Contains an attribute-token pair that the plug-in uses to create an attribute in the managed entry with a value taken from the originating entry.
Section 5.2.115, "mepRDNAttr"	Specifies which attribute to use as the naming attribute in the managed entry.
Section 5.2.116, "mepStaticAttr"	Contains an attribute-value pair that will be used, with that specified value, in the managed entry.

5.3.44. netscapeCertificateServer

The **netscapeCertificateServer** object class stores information about a Netscape certificate server. This object is defined in the schema for the Netscape Certificate Management System.

Superior Class

top

OID

2.16.840.1.113730.3.2.18

Table 5.81. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

5.3.45. netscapeDirectoryServer

The **netscapeDirectoryServer** object class stores information about a Directory Server instance. This object is defined in the schema for the Netscape Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.23

Table 5.82. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

5.3.46. NetscapeLinkedOrganization

NetscapeLinkedOrganization is an auxiliary object class. This object is defined in the schema for the Netscape server suite.

Superior Class

top

OID

1.3.6.1.4.1.1466.101.120.141

Table 5.83. Allowed Attributes

Attribute	Definition
Section 5.2.294, "parentOrganization"	Identifies the parent organization for the linked organization defined for the server suite.

5.3.47. netscapeMachineData

The **netscapeMachineData** object class distinguishes between machine data and non-machine data. This object is defined in the schema for the Netscape Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.32

5.3.48. NetscapePreferences

NetscapePreferences is an auxiliary object class which stores the user preferences. This object is defined by Netscape.

Superior Class

top

OID

1.3.6.1.4.1.1466.101.120.142

Table 5.84. Required Attributes

Attribute	Definition
Section 5.2.303, "preferredLanguage"	Gives the person's preferred written or spoken language.
Section 5.2.304, "preferredLocale"	Gives the person's preferred locale. A locale setting defines cultural or national settings like date formats and currencies.
Section 5.2.305, "preferredTimeZone"	Gives the person's preferred time zone.

5.3.49. netscapeReversiblePasswordObject

netscapeReversiblePasswordObject is an auxiliary object class to store a password. This object is defined in the schema for the Netscape Web Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.154

Table 5.85. Allowed Attributes

Attribute	Definition
Section 5.2.150, "netscapeReversiblePassword"	Contains a password used for HTTP Digest/MD5 authentication.

5.3.50. netscapeServer

The **netscapeServer** object class contains instance-specific information about a Netscape server and its installation.

Superior Class

top

OID

2.16.840.1.113730.3.2.10

Table 5.86. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.87. Allowed Attributes

Attribute	Definition
Section 5.2.5, "administratorContactInfo"	Contains the contact information for the server administrator.
Section 5.2.7, "adminUrl"	Contains the URL for the Administration Server used by the instance.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.75, "installationTimeStamp"	Contains the time that the server instance was installed.
Section 5.2.317, "serverHostName"	Contains the host name of the server on which the Directory Server instance is running.
Section 5.2.318, "serverProductName"	Contains the product name of the server type.
Section 5.2.319, "serverRoot"	Specifies the top directory where the server product is installed.
Section 5.2.320, "serverVersionNumber"	Contains the product version number.
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.

5.3.51. netscapeWebServer

The **netscapeWebServer** object class identifies an installed Netscape Web Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.29

Table 5.88. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.214, "nsServerID"	Contains the server's name or ID.

Table 5.89. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.216, "nsServerPort"	Contains the server's port number.

5.3.52. newPilotPerson

The **newPilotPerson** object class is a subclass of the **person** to allow additional attributes to be assigned to entries of the **person** object class. This object class inherits the [Section 5.2.25, "cn \(commonName\)"](#) and [Section 5.2.329, "sn \(surname\)"](#) attributes from the **person** object class.

This object class is defined in Internet White Pages Pilot.

Superior Class

person

OID

0.9.2342.19200300.100.4.4

Table 5.90. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.329, "sn (surname)"	Gives the person's family name or last name.

Table 5.91. Allowed Attributes

Attribute	Definition
Section 5.2.20, "businessCategory"	Gives the type of business in which the entry is engaged.
Section 5.2.37, "description"	Gives a text description of the entry.

Attribute	Definition
Section 5.2.51, "drink (favouriteDrink)"	Gives the person's favorite drink.
Section 5.2.62, "homePhone"	Gives the person's home phone number.
Section 5.2.63, "homePostalAddress"	Gives the person's home mailing address.
Section 5.2.83, "janetMailbox"	Gives the person's email address; this is primarily for use in Great Britain or organizations which do no use RFC 822 mail addresses.
Section 5.2.91, "mail"	Contains the person's email address.
Section 5.2.101, "mailPreferenceOption"	Indicates the user's preference for including his name on mailing lists (electronic or physical).
Section 5.2.131, "mobile"	Gives the person's mobile phone number.
Section 5.2.289, "organizationalStatus"	Gives the common job category for a person's function.
Section 5.2.290, "otherMailbox"	Contains values for electronic mailbox types other than X.400 and RFC 822.
Section 5.2.293, "pager"	Gives the person's pager number.
Section 5.2.295, "personalSignature"	Contains the person's signature file.
Section 5.2.296, "personalTitle"	Gives the person's honorific.
Section 5.2.302, "preferredDeliveryMethod"	Shows the person's preferred method of contact or message delivery.
Section 5.2.312, "roomNumber"	Gives the room number where the person is located.
Section 5.2.314, "secretary"	Contains the DN (distinguished name) of the person's secretary or administrative assistant.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.337, "telephoneNumber"	Gives the telephone number for the entry.
Section 5.2.342, "uid (userID)"	Contains the person's user ID (usually his logon ID).
Section 5.2.349, "userClass"	Describes the type of computer user this entry is.

Attribute	Definition
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.

5.3.53. nisMap

This object class points to a NIS map.

This object class is defined in [RFC 2307](#), which defines object classes and attributes to use LDAP as a network information service.



NOTE

This object class is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-instance/schema** directory.

Superior Class

top

OID

1.3.6.1.1.2.13

Table 5.92. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.152, "nisMapName"	Contains the NIS map name.

Table 5.93. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.54. nisNetgroup

This object class contains a netgroup used within a NIS domain. Adding this object class allows administrators to use netgroups to control login and service authentication in NIS.

This object class is defined in [RFC 2307](#), which defines object classes and attributes to use LDAP as a network information service.



NOTE

This object class is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

Superior Class

top

OID

1.3.6.1.1.2.8

Table 5.94. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.95. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.108, "memberNisNetgroup"	Merges the attribute values of another netgroup into the current one by listing the name of the merging netgroup.
Section 5.2.153, "nisNetgroupTriple"	Contains a user name (,bobby,example.com) or a machine name (shellserver1,,example.com).

5.3.55. nisObject

This object class contains information about an object in a NIS domain.

This object class is defined in [RFC 2307](#), which defines object classes and attributes to use LDAP as a network information service.



NOTE

This object class is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-*instance*/schema** directory.

Superior Class

top

OID

1.3.6.1.1.2.10

Table 5.96. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.151, "NisMapEntry"	Identifies the NIS map entry.
Section 5.2.152, "nisMapName"	Contains the name of the NIS map.

Table 5.97. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.56. nsAdminConfig

This object class stores the configuration parameters for the Administration Server. This object is defined for the Administration Services.

Superior Class

nsConfig

OID

nsAdminConfig-oid

Table 5.98. Allowed Attributes

Attribute	Definition
Section 5.2.155, "nsAdminAccessAddresses"	Identifies the Administration Server IP addresses.
Section 5.2.156, "nsAdminAccessHosts"	Contains the Administration Server host name or a list of Administration Server host names.
Section 5.2.158, "nsAdminCacheLifetime"	Notes the length of the cache timeout period.
Section 5.2.159, "nsAdminCgiWaitPid"	Contains the PID of the CGI process the server is waiting for.

Attribute	Definition
Section 5.2.161, "nsAdminEnableEnduser"	Sets whether to allow or disallow end user access to the Administration Server web services pages.
Section 5.2.164, "nsAdminOneACLDir"	Contains the path of the local ACL directory for the Administration Server.
Section 5.2.166, "nsAdminUsers"	Points to the file which contains the admin user info.

5.3.57. nsAdminConsoleUser

This object class stores the configuration parameters for the Administration Server. This object is defined for the Administration Services.

Superior Class

top

OID

nsAdminConsoleUser-oid

Table 5.99. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.100. Allowed Attributes

Attribute	Definition
Section 5.2.206, "nsPreference"	Stores preference information for console settings.

5.3.58. nsAdminDomain

This object class stores user information to access Admin Console. This object is defined for the Administration Services.

Superior Class

organizationalUnit

OID

nsAdminDomain-oid

Table 5.101. Allowed Attributes

Attribute	Definition
Section 5.2.160, "nsAdminDomainName"	Identifies the administration domain for the servers.

5.3.59. nsAdminGlobalParameters

This object class stores the configuration parameters for the Administration Server. This object is defined for the Administration Services.

Superior Class

top

OID

nsAdminGlobalParameters-oid

Table 5.102. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.103. Allowed Attributes

Attribute	Definition
Section 5.2.162, "nsAdminEndUserHTMLIndex"	Sets whether to allow or disallow end-user access to the HTML index pages.
Section 5.2.202, "nsNickName"	Gives the nickname for the application.

5.3.60. nsAdminGroup

This object class stores group information for administrator users in the Administration Server. This object is defined for the Administration Services.

Superior Class

top

OID

nsAdminGroup-oid

Table 5.104. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.105. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.163, "nsAdminGroupName"	Contains the name for the admin group.
Section 5.2.165, "nsAdminSIEDN"	Shows the DN of the server instance entry (SIE) for the Administration Server instance.
Section 5.2.175, "nsConfigRoot"	Gives the full path to the Administration Server instance's configuration directory.

5.3.61. nsAdminObject

This object class contains information about an object used by Administration Server, such as a task. This object is defined for the Administration Services.

Superior Class

top

OID

nsAdminObject-oid

Table 5.106. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.107. Allowed Attributes

Attribute	Definition
Section 5.2.174, "nsClassname"	Contains the class name associated with the task or resource editor for the Administration Server.

Attribute	Definition
Section 5.2.193, "nsJarfilename"	Gives the name of the JAR file used by the Administration Server Console to access the object.

5.3.62. nsAdminResourceEditorExtension

This object class contains an extension used by the Console Resource Editor. This object is defined for the Administration Services.

Superior Class

nsAdminObject

OID

nsAdminResourceEditorExtension-oid

Table 5.108. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.109. Allowed Attributes

Attribute	Definition
Section 5.2.157, "nsAdminAccountInfo"	Contains information about the Administration Server account.
Section 5.2.179, "nsDeleteclassname"	Contains the name of a class to be deleted.

5.3.63. nsAdminServer

This object class defines the Administration Server instance. This object is defined for the Administration Services.

Superior Class

top

OID

nsAdminServer-oid

Table 5.110. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.214, "nsServerID"	Contains the Directory Server ID, such as slapd-example .

Table 5.111. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.64. nsAIMpresence

nsAIMpresence is an auxiliary object class which defines the status of an AOL instance messaging account. This object is defined for the Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.300

Table 5.112. Allowed Attributes

Attribute	Definition
Section 5.2.167, "nsAIMid"	Contains the AIM user ID for the entry.
Section 6.23, "nsAIMStatusGraphic"	Contains a pointer to the graphic image which indicates the AIM account's status.
Section 6.24, "nsAIMStatusText"	Contains the text to indicate the AIM account's status.

5.3.65. nsApplication

nsApplication defines an application or server entry. This is defined by Netscape.

Superior Class

top

OID

nsApplication-oid

Table 5.113. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.114. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.75, "installationTimeStamp"	Contains the time that the server instance was installed.
Section 5.2.171, "nsBuildNumber"	Contains the build number for the server instance.
Section 5.2.172, "nsBuildSecurity"	Contains the level of security used to make the build.
Section 5.2.186, "nsExpirationDate"	Contains the date that the license for the application expires.
Section 5.2.192, "nsInstalledLocation"	For servers which are version 7.1 or older, shows the installation directory for the server.
Section 5.2.194, "nsLdapSchemaVersion"	Gives the version of the LDAP schema files used by the Directory Server.
Section 5.2.202, "nsNickName"	Gives the nickname for the application.
Section 5.2.207, "nsProductName"	Gives the name of the server product.
Section 5.2.208, "nsProductVersion"	Shows the version number of the server product.
Section 5.2.209, "nsRevisionNumber"	Contains the revision number (minor version) for the product.
Section 5.2.211, "nsSerialNumber"	Gives the serial number assigned to the server product.
Section 5.2.215, "nsServerMigrationClassname"	Gives the class to use to migrate a server instance.
Section 5.2.213, "nsServerCreationClassname"	Gives the class to use to create a server instance.
Section 5.2.242, "nsVendor"	Contains the name of the vendor who designed the server.

5.3.66. nsCertificateServer

The **nsCertificateServer** object class stores information about a Red Hat Certificate System instance. This object is defined in the schema for the Certificate System.

Superior Class

top

OID

nsCertificateServer-oid

Table 5.115. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.214, "nsServerID"	Contains the server's name or ID.

Table 5.116. Allowed Attributes

Attribute	Definition
Section 5.2.173, "nsCertConfig"	Contains configuration settings for a Red Hat Certificate System instance.
Section 5.2.216, "nsServerPort"	Contains the server's port number.
Section 5.2.317, "serverHostName"	Contains the host name of the server on which the Directory Server instance is running.

5.3.67. nsComplexRoleDefinition

Any role that is not a simple role is, by definition, a complex role.

This object class is defined by Directory Server.

Superior Class

nsRoleDefinition

OID

2.16.840.1.113730.3.2.95

Table 5.117. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.118. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.68. nsContainer

Some entries do not define any specific entity, but they create a defined space within the directory tree as a parent entry for similar or related child entries. These are *container entries*, and they are identified by the **nsContainer** object class.

Superior Class

top

OID

2.16.840.1.113730.3.2.104

Table 5.119. Required Attributes

Attribute	Definition
objectClass	Defines the object classes for the entry.
cn	Gives the common name of the entry.

5.3.69. nsCustomView

The **nsCustomView** object class defines information about custom views of the Directory Server data in the Directory Server Console. This is defined for Administration Services.

Superior Class

nsAdminObject

OID

nsCustomView-oid

Table 5.120. Allowed Attributes

Attribute	Definition
Section 5.2.183, "nsDisplayName"	Contains the name of the custom view setting profile.

5.3.70. nsDefaultObjectClasses

nsDefaultObjectClasses sets default object classes to use when creating a new object of a certain type within the directory. This is defined for Administration Services.

Superior Class

top

OID

nsDefaultObjectClasses-oid

Table 5.121. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the device.

Table 5.122. Allowed Attributes

Attribute	Definition
Section 5.2.178, "nsDefaultObjectClass"	Contains an object class to assign by default to an object type.

5.3.71. nsDirectoryInfo

nsDirectoryInfo contains information about a directory instance. This is defined for Administration Services.

Superior Class

top

OID

nsDirectoryInfo-oid

Table 5.123. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the device.

Table 5.124. Allowed Attributes

Attribute	Definition

Attribute	Definition
Section 5.2.169, "nsBindDN"	Contains the bind DN defined for the server in its server instance entry.
Section 5.2.170, "nsBindPassword"	Contains the password for the bind identity in the SIE.
Section 5.2.180, "nsDirectoryFailoverList"	Contains a list of URLs of other Directory Server instances to use for failover support if the instance in nsDirectoryURL is unavailable.
Section 5.2.181, "nsDirectoryInfoRef"	Contains a reference to a distinguished name (DN) in the directory.
Section 5.2.182, "nsDirectoryURL"	Contains a URL to access the Directory Server instance.

5.3.72. nsDirectoryServer

nsDirectoryServer is the defining object class for a Directory Server instance. This is defined for the Directory Server.

Superior Class

top

OID

nsDirectoryServer-oid

Table 5.125. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.
Section 5.2.214, "nsServerID"	Contains the server's name or ID.

Table 5.126. Allowed Attributes

Attribute	Definition
Section 5.2.168, "nsBaseDN"	Contains the base DN for the server instance.
Section 5.2.169, "nsBindDN"	Contains the bind DN defined for the server in its server instance entry.

Attribute	Definition
Section 5.2.170, "nsBindPassword"	Contains the password for the bind identity in the SIE.
Section 5.2.210, "nsSecureServerPort"	Contains the server's TLS port number.
Section 5.2.216, "nsServerPort"	Contains the server's port number.
Section 5.2.317, "serverHostName"	Contains the host name of the server on which the Directory Server instance is running.

5.3.73. nsFilteredRoleDefinition

The **nsFilteredRoleDefinition** object class defines how entries are assigned to the role, depending upon the attributes contained by each entry.

This object class is defined in Directory Server.

Superior Class

nsComplexRoleDefinition

OID

2.16.840.1.113730.3.2.97

Table 5.127. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 6.39, "nsRoleFilter"	Specifies the filter used to identify entries in the filtered role.

Table 5.128. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.74. nsGlobalParameters

The **nsGlobalParameters** object class contains global preference settings.

This object class is defined in Administrative Services.

Superior Class

top

OID

nsGlobalParameters-oid

Table 5.129. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.130. Allowed Attributes

Attribute	Definition
Section 5.2.187, "nsGroupRDNComponent"	Defines the default attribute type used in the RDN of the group entry.
Section 5.2.227, "nsUniqueAttribute"	Defines a unique attribute in the preferences.
Section 5.2.228, "nsUserIDFormat"	Sets the format to generate the user ID from the givenname and sn attributes.
Section 5.2.229, "nsUserRDNComponent"	Sets the attribute type to use as the naming component in the user DN.
nsNYR	Not used.
nsWellKnownJarfiles	Not used.

5.3.75. nsHost

The **nsHost** object class stores information about the server host.

This object class is defined in Administrative Services.

Superior Class

top

OID

nsHost-oid

Table 5.131. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.132. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.188, "nsHardwarePlatform"	Identifies the hardware platform for the host on which the Directory Server instance is running. This is the same information as running uname -m .
Section 5.2.190, "nsHostLocation"	Gives the location of the server host.
Section 5.2.204, "nsOsVersion"	Contains the operating system version of the server host.
Section 5.2.317, "serverHostName"	Contains the host name of the server on which the Directory Server instance is running.

5.3.76. nsICQpresence

nsICQpresence is an auxiliary object class which defines the status of an ICQ messaging account. This object is defined for the Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.301

Table 5.133. Allowed Attributes

Attribute	Definition
Section 5.2.191, "nsICQid"	Contains the ICQ user ID for the entry.
Section 6.28, "nsICQStatusGraphic"	Contains a pointer to the graphic image which indicates the ICQ account's status.

Attribute	Definition
Section 6.29, "nsICQStatusText"	Contains the text to indicate the ICQ account's status.

5.3.77. nsLicenseUser

The **nsLicenseUser** object class tracks licenses for servers that are licensed on a per-client basis. **nsLicenseUser** is intended to be used with the **inetOrgPerson** object class. You can manage the contents of this object class through the **Users and Groups** area of the Administration Server.

This object class is defined in the Administration Server schema.

Superior Class

top

OID

2.16.840.1.113730.3.2.7

Table 5.134. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.135. Allowed Attributes

Attribute	Definition
Section 5.2.195, "nsLicensedFor"	Identifies the server that the user is licensed to use.
Section 5.2.196, "nsLicenseEndTime"	Reserved for future use.
Section 5.2.197, "nsLicenseStartTime"	Reserved for future use.

5.3.78. nsManagedRoleDefinition

The **nsManagedRoleDefinition** object class specifies the member assignments of a role to an explicit, enumerated list of members.

This object class is defined in Directory Server.

Superior Class

nsComplexRoleDefinition

OID

2.16.840.1.113730.3.2.96

Table 5.136. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.137. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.79. nsMessagingServerUser

nsICQpresence is an auxiliary object class that describes a messaging server user. This object class is defined for Netscape Messaging Server.

Superior Class

top

OID

2.16.840.113730.3.2.37

Table 5.138. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes for the entry.

Table 5.139. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.92, "mailAccessDomain"	Contains the domain from which the user can access the messaging server.
Section 5.2.93, "mailAlternateAddress"	Contains secondary email addresses for the group.
Section 5.2.94, "mailAutoReplyMode"	Specifies whether autoreply mode for the account is enabled.
Section 5.2.95, "mailAutoReplyText"	Contains the text use for automatic reply emails.

Attribute	Definition
Section 5.2.96, "mailDeliveryOption"	Specifies the mail delivery mechanism to be used for the mail user.
Section 5.2.98, "mailForwardingAddress"	Specifies the mail delivery mechanism to use for the mail user.
Section 5.2.100, "mailMessageStore"	Specifies the location of the user's mail box.
Section 5.2.102, "mailProgramDeliveryInfo"	Specifies the commands used for programmed mail delivery.
Section 5.2.103, "mailQuota"	Specifies the disk space allowed for the user's mail box.
Section 5.2.199, "nsmsgDisallowAccess"	Sets limits on the mail protocols available to the user.
Section 5.2.200, "nsmsgNumMsgQuota"	Specifies the number of messages allowed for the user's mail box.
Section 5.2.246, "nswmExtendedUserPrefs"	Stores the extended preferences for the user.
Section 5.2.353, "vacationEndDate"	Contains the end date for a vacation period.
Section 5.2.354, "vacationStartDate"	Contains the start date for a vacation period.

5.3.80. nsMSNpresence

nsMSNpresence is an auxiliary object class which defines the status of an MSN instance messaging account. This object is defined for the Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.303

Table 5.140. Allowed Attributes

Attribute	Definition
Section 5.2.201, "nsMSNid"	Contains the MSN user ID for the entry.

5.3.81. nsNestedRoleDefinition

The **nsNestedRoleDefinition** object class specifies one or more roles, of any type, are included as members within the role.

This object class is defined in Directory Server.

Superior Class

nsComplexRoleDefinition

OID

2.16.840.1.113730.3.2.98

Table 5.141. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 6.38, "nsRoleDn"	Specifies the roles assigned to an entry.

Table 5.142. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.82. nsResourceRef

The **nsNestedRoleDefinition** object class configures a resource reference.

This object class is defined in the Administration Services.

Superior Class

top

OID

nsResourceRef-oid

Table 5.143. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.144. Allowed Attributes

Attribute	Definition
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.

5.3.83. nsRoleDefinition

All role definition object classes inherit from the **nsRoleDefinition** object class.

This object class is defined by Directory Server.

Superior Class

LDAPsubentry

OID

2.16.840.1.113730.3.2.93

Table 5.145. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.146. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.84. nsSimpleRoleDefinition

Roles containing this object class are called simple roles because they have a deliberately limited flexibility, which makes it easy to:

- Enumerate the members of a role.
- Determine whether a given entry possesses a particular role.
- Enumerate all the roles possessed by a given entry.
- Assign a particular role to a given entry.
- Remove a particular role from a given entry.

This object class is defined by Directory Server.

Superior Class

nsRoleDefinition

OID

2.16.840.1.113730.3.2.94

Table 5.147. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.148. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.85. nsSNMP

This object class defines the configuration for the SNMP plug-in object used by the Directory Server.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.41

Table 5.149. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.220, "nsSNMPEnabled"	Sets whether SNMP is enabled for the Directory Server instance.

Table 5.150. Allowed Attributes

Attribute	Definition
Section 5.2.218, "nsSNMPContact"	Contains the contact information provided by the SNMP agent.

Attribute	Definition
Section 5.2.219, "nsSNMPDescription"	Contains a text description of the SNMP setup.
Section 5.2.221, "nsSNMPLocation"	Contains the location information or configuration for the SNMP agent.
Section 5.2.222, "nsSNMPMasterHost"	Contains the host name for the server where the SNMP master agent is located.
Section 5.2.223, "nsSNMPMasterPort"	Contains the port to access the SNMP subagent.
Section 5.2.224, "nsSNMPOrganization"	Contains the organization name or information provided by the SNMP service.

5.3.86. nsTask

This object class defines the configuration for tasks performed by the Directory Server.

This object class is defined for the Administrative Services.

Superior Class

top

OID

nsTask-oid

Table 5.151. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.152. Allowed Attributes

Attribute	Definition
Section 5.2.185, "nsExecRef"	Contains a reference to the program which will perform the task.
Section 5.2.189, "nsHelpRef"	Contains a reference to an online (HTML) help file associated with the task window.
Section 5.2.198, "nsLogSuppress"	Sets whether to suppress logging for the task.

Attribute	Definition
Section 5.2.226, "nsTaskLabel"	Contains a label associated with the task in the Console.

5.3.87. nsTaskGroup

This object class defines the information for a group of tasks in the Console.

This object class is defined for the Administrative Services.

Superior Class

top

OID

nsTaskGroup-oid

Table 5.153. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.154. Allowed Attributes

Attribute	Definition
Section 5.2.226, "nsTaskLabel"	Contains a label associated with the task in the Console.

5.3.88. nsTopologyCustomView

This object class configures the topology views used for the profile in the Console.

This object class is defined for the Administrative Services.

Superior Class

nsCustomView

OID

nsTopologyCustomView-oid

Table 5.155. Required Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.156. Allowed Attributes

Attribute	Definition
Section 5.2.243, "nsViewConfiguration"	Contains the view configuration to use in the Console.

5.3.89. nsTopologyPlugin

This object class configures the topology plug-in used to set views in the Console.

This object class is defined for the Administrative Services.

Superior Class

nsAdminObject

OID

nsTopologyPlugin-oid

5.3.90. nsValueItem

This object class defines a value item object configuration, which is used to specify information that is dependent on the value type of an entry. A value item relates to the allowed attribute value syntax for an entry attribute, such as binary or case-sensitive string.

This object class is defined in Netscape Servers - Value Item.

Superior Class

top

OID

2.16.840.1.113730.3.2.45

Table 5.157. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.158. Allowed Attributes

Attribute	Definition
Section 5.2.230, "nsValueBin"	Contains information or operations related to the binary value type.
Section 5.2.231, "nsValueCES"	Contains information or operations related to the case-exact string (CES) value type.
Section 5.2.232, "nsValueCIS"	Contains information or operations related to the case-insensitive (CIS) value type.
Section 5.2.233, "nsValueDefault"	Sets the default value type to use for an attribute or configuration parameter.
Section 5.2.234, "nsValueDescription"	Gives a text description of the value item setting.
Section 5.2.235, "nsValueDN"	Contains information or operations related to the DN value type.
Section 5.2.236, "nsValueFlags"	Sets flags for the value item object.
Section 5.2.237, "nsValueHelpURL"	Contains a reference to an online (HTML) help file associated with the value item object.
Section 5.2.238, "nsValueInt"	Contains information or operations related to the integer value type.
Section 5.2.239, "nsValueSyntax"	Defines the syntax to use for the value item object.
Section 5.2.240, "nsValueTel"	Contains information or operations related to the telephone string value type.
Section 5.2.241, "nsValueType"	Sets which value type to apply.

5.3.91. nsView

This object class is used for a view entry in the directory tree.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.304

Table 5.159. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.160. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.244, "nsViewFilter"	Identifies the filter used by the view plug-in.

5.3.92. nsYIMpresence

nsYIMpresence is an auxiliary object class which defines the status of a Yahoo instance messaging account. This object is defined for the Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.302

Table 5.161. Allowed Attributes

Attribute	Definition
Section 5.2.247, "nsYIMid"	Contains the Yahoo user ID for the entry.
Section 6.45, "nsYIMStatusGraphic"	Contains a pointer to the graphic image which indicates the Yahoo account's status.
Section 6.46, "nsYIMStatusText"	Contains the text to indicate the Yahoo account's status.

5.3.93. ntGroup

The **ntGroup** object class holds data for a group entry stored in a Windows Active Directory server. Several Directory Server attributes correspond directly to or are mapped to match Windows group attributes. When you create a new group in the Directory Server that is to be synchronized with a Windows server group, Directory Server attributes are assigned to the Windows entry. These attributes may then be added, modified, or deleted in the entry through either directory service.

This object class is defined in Netscape NT Synchronization.

Superior Class

[top](#)**OID**

2.16.840.1.113730.3.2.9

Table 5.162. Required Object Classes

Object Class	Definition
Section 5.3.39, "mailGroup"	Allows the mail attribute to be synchronized between Windows and Directory Server groups.

Table 5.163. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.263, "ntUserDomainId"	Contains the Windows domain login ID for the group account.

Table 5.164. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry; this corresponds to the Windows name field.
Section 5.2.37, "description"	Gives a text description of the entry; corresponds to the Windows comment field.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.106, "member"	Specifies the members of the group.
Section 5.2.249, "ntGroupCreateNewGroup"	Specifies whether a Windows account should be created when an entry is created in the Directory Server.
Section 5.2.250, "ntGroupDeleteGroup"	Specifies whether a Windows account should be deleted when an entry is deleted in the Directory Server.
Section 5.2.251, "ntGroupDomainId"	Gives the domain ID string for the group.
Section 5.2.253, "ntGroupType"	Defines what kind of Windows domain group the entry is.

Attribute	Definition
Section 5.2.254, "ntUniqueId"	Contains a generated ID number used by the server for operations and identification.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.

5.3.94. ntUser

The **ntUser** entry holds data for a user entry stored in a Windows Active Directory server. Several Directory Server attributes correspond directly to or are mapped to match Windows user account fields. When you create a new person entry in the Directory Server that is to be synchronized with a Windows server, Directory Server attributes are assigned to Windows user account fields. These attributes may then be added, modified, or deleted in the entry through either directory service.

This object class is defined in Netscape NT Synchronization.

Superior Class

top

OID

2.16.840.1.113730.3.2.8

Table 5.165. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry; this corresponds to the Windows name field.
Section 5.2.263, "ntUserDomainId"	Contains the Windows domain login ID for the user account.

Table 5.166. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry; corresponds to the Windows comment field.

Attribute	Definition
Section 5.2.38, "destinationIndicator"	Gives the country and city associated with the entry; this was once required to provide public telegram service.
Section 5.2.56, "fax (facsimileTelephoneNumber)"	Gives the fax number for the user.
Section 5.2.60, "givenName"	Contains the person's first name.
Section 5.2.62, "homePhone"	Gives the person's home phone number.
Section 5.2.63, "homePostalAddress"	Gives the person's home mailing address.
Section 5.2.74, "initials"	Gives the person's initials.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.91, "mail"	Contains the person's email address.
Section 5.2.105, "manager"	Contains the DN (distinguished name) of the direct supervisor of the person entry.
Section 5.2.131, "mobile"	Gives the person's mobile phone number.
Section 5.2.255, "ntUserAcctExpires"	Identifies when the user's Windows account will expire.
Section 5.2.258, "ntUserCodePage"	Gives the user's code page.
Section 5.2.261, "ntUserCreateNewAccount"	Specifies whether a Windows account should be created when this entry is created in the Directory Server.
Section 5.2.262, "ntUserDeleteAccount"	Specifies whether a Windows account should be deleted when this entry is deleted in the Directory Server.
Section 5.2.265, "ntUserHomeDir"	Gives the path to the user's home directory.
Section 5.2.267, "ntUserLastLogoff"	Gives the time of the user's last logoff from the Windows server.
Section 5.2.268, "ntUserLastLogon"	Gives the time of the user's last logon to the Windows server.

Attribute	Definition
Section 5.2.271, "ntUserMaxStorage"	Shows the maximum disk space available to the user in the Windows server.
Section 5.2.273, "ntUserParms"	Contains a Unicode string reserved for use by applications.
Section 5.2.277, "ntUserProfile"	Contains the path to the user's Windows profile.
Section 5.2.278, "ntUserScriptPath"	Contains the path to the user's Windows login script.
Section 5.2.282, "ntUserWorkstations"	Contains a list of Windows workstations from which the user is allowed to log into the Windows domain.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the entry belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.
Section 5.2.293, "pager"	Gives the person's pager number.
Section 5.2.299, "postalAddress"	Contains the mailing address for the entry.
Section 5.2.300, "postalCode"	Gives the postal code for the entry, such as the zip code in the United States.
Section 5.2.301, "postOfficeBox"	Gives the post office box number for the entry.
Section 5.2.310, "registeredAddress"	Gives a postal address suitable to receive expedited documents when the recipient must verify delivery.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.329, "sn (surname)"	Gives the person's family name or last name.
Section 5.2.330, "st (stateOrProvinceName)"	Gives the state or province where the person is located.
Section 5.2.331, "street"	Gives the street name and address number for the person's physical location.
Section 5.2.337, "telephoneNumber"	Gives the telephone number for the entry.
Section 5.2.338, "teletexTerminalIdentifier"	Gives the identifier for the person's teletex terminal.
Section 5.2.339, "telexNumber"	Gives the telex number associated with the entry.

Attribute	Definition
Section 5.2.340, "title"	Shows the person's job title.
Section 5.2.348, "userCertificate"	Stores a user's certificate in cleartext (not used).
Section 5.2.355, "x121Address"	Gives the X.121 address for the entry.

5.3.95. oncRpc

The **oncRpc** object class defines an abstraction of an Open Network Computing Remote Procedure Call (ONC RPC). This object class is defined in [RFC 2307](#).



NOTE

This object class is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slapd-instance/schema** directory.

Superior Class

top

OID

1.3.6.1.1.2.5

Table 5.167. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Defines the object classes for the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.288, "oncRpcNumber"	Contains part of the RPC map and stores the RPC number for UNIX RPCs.

Table 5.168. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.

5.3.96. organization

The **organization** attributes defines entries that represent organizations. An organization is generally assumed to be a large, relatively static grouping within a larger corporation or enterprise.

This object class is defined in [RFC 2256](#).

Superior Class

top

OID

2.5.6.4

Table 5.169. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the entry belongs.

Table 5.170. Allowed Attributes

Attribute	Definition
Section 5.2.20, "businessCategory"	Gives the type of business in which the entry is engaged.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.38, "destinationIndicator"	Gives the country and city associated with the entry; this was once required to provide public telegram service.
Section 5.2.56, "fax (facsimileTelephoneNumber)"	Contains the fax number for the entry.
Section 5.2.76, "internationalISDNNumber"	Contains the ISDN number for the entry.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.298, "physicalDeliveryOfficeName"	Gives a location where physical deliveries can be made.
Section 5.2.299, "postalAddress"	Contains the mailing address for the entry.
Section 5.2.300, "postalCode"	Gives the postal code for the entry, such as the zip code in the United States.
Section 5.2.301, "postOfficeBox"	Gives the post office box number for the entry.
Section 5.2.302, "preferredDeliveryMethod"	Shows the preferred method of contact or message delivery for the entry.

Attribute	Definition
Section 5.2.310, "registeredAddress"	Gives a postal address suitable to receive expedited documents when the recipient must verify delivery.
Section 5.2.313, "searchGuide"	Specifies information for suggested search criteria when using the entry as the base object in the directory tree for a search.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.330, "st (stateOrProvinceName)"	Gives the state or province where the person is located.
Section 5.2.331, "street"	Gives the street name and number for the person's physical location.
Section 5.2.337, "telephoneNumber"	Gives the telephone number of the person responsible for the organization.
Section 5.2.338, "teletexTerminalIdentifier"	Gives the ID for an entry's teletex terminal.
Section 5.2.339, "telexNumber"	Gives the telex number associated with the entry.
Section 5.2.350, "userPassword"	Gives the password with which the entry can bind to the directory.
Section 5.2.355, "x121Address"	Gives the X.121 address for the entry.

5.3.97. organizationalPerson

The **organizationalPerson** object class defines entries for people employed or affiliated with the organization. This object class inherits the [Section 5.2.25, "cn \(commonName\)"](#) and [Section 5.2.329, "sn \(surname\)"](#) attributes from the **person** object class.

This object class is defined in [RFC 2256](#).

Superior Class

person

OID

2.5.6.7

Table 5.171. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.329, "sn (surname)"	Gives the person's family name or last name.

Table 5.172. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.38, "destinationIndicator"	Gives the country and city associated with the entry; this was once required to provide public telegram service.
Section 5.2.56, "fax (facsimileTelephoneNumber)"	Contains the fax number for the entry.
Section 5.2.76, "internationalISDNNumber"	Contains the ISDN number for the entry.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.
Section 5.2.298, "physicalDeliveryOfficeName"	Gives a location where physical deliveries can be made.
Section 5.2.299, "postalAddress"	Contains the mailing address for the entry.
Section 5.2.300, "postalCode"	Gives the postal code for the entry, such as the zip code in the United States.
Section 5.2.301, "postOfficeBox"	Gives the post office box number for the entry.
Section 5.2.302, "preferredDeliveryMethod"	Shows the person's preferred method of contact or message delivery.
Section 5.2.310, "registeredAddress"	Gives a postal address suitable to receive expedited documents when the recipient must verify delivery.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.

Attribute	Definition
Section 5.2.330, "st (stateOrProvinceName)"	Gives the state or province where the person is located.
Section 5.2.331, "street"	Gives the street name and number for the person's physical location.
Section 5.2.337, "telephoneNumber"	Gives the telephone number for the entry.
Section 5.2.338, "teletexTerminalIdentifier"	Gives the ID for an entry's teletex terminal.
Section 5.2.339, "telexNumber"	Gives the telex number associated with the entry.
Section 5.2.340, "title"	Shows the person's job title.
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.
Section 5.2.355, "x121Address"	Gives the X.121 address for the entry.

5.3.98. organizationalRole

The **organizationalRole** object class is used to define entries for roles held by people within an organization.

This object class is defined in [RFC 2256](#).

Superior Class

top

OID

2.5.6.8

Table 5.173. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.174. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.

Attribute	Definition
Section 5.2.38, "destinationIndicator"	Gives the country and city associated with the entry; this was once required to provide public telegram service.
Section 5.2.56, "fax (facsimileTelephoneNumber)"	Contains the fax number for the entry.
Section 5.2.76, "internationalISDNNumber"	Contains the ISDN number for the entry.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.
Section 5.2.298, "physicalDeliveryOfficeName"	Gives a location where physical deliveries can be made.
Section 5.2.299, "postalAddress"	Contains the mailing address for the entry.
Section 5.2.300, "postalCode"	Gives the postal code for the entry, such as the zip code in the United States.
Section 5.2.301, "postOfficeBox"	Gives the post office box number for the entry.
Section 5.2.302, "preferredDeliveryMethod"	Shows the role's preferred method of contact or message delivery.
Section 5.2.310, "registeredAddress"	Gives a postal address suitable to receive expedited documents when the recipient must verify delivery.
Section 5.2.311, "roleOccupant"	Contains the DN (distinguished name) of the person in the role.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.330, "st (stateOrProvinceName)"	Gives the state or province where the entry is located.
Section 5.2.331, "street"	Gives the street name and number for the role's physical location.
Section 5.2.337, "telephoneNumber"	Gives the telephone number for the entry.
Section 5.2.338, "teletexTerminalIdentifier"	Gives the ID for an entry's teletex terminal.
Section 5.2.339, "telexNumber"	Gives the telex number associated with the entry.

Attribute	Definition
Section 5.2.355, "x121Address"	Gives the X.121 address for the entry.

5.3.99. organizationalUnit

The **organizationalUnit** object class defines entries that represent *organizational units*, generally understood to be a relatively static grouping within a larger organization.

This object class is defined in [RFC 2256](#).

Superior Class

top

OID

2.5.6.5

Table 5.175. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.

Table 5.176. Allowed Attributes

Attribute	Definition
Section 5.2.20, "businessCategory"	Gives the type of business in which the entry is engaged.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.38, "destinationIndicator"	Gives the country and city associated with the entry; this was once required to provide public telegram service.
Section 5.2.56, "fax (facsimileTelephoneNumber)"	Contains the fax number for the entry.
Section 5.2.76, "internationalISDNNumber"	Contains the ISDN number for the entry.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.298, "physicalDeliveryOfficeName"	Gives a location where physical deliveries can be made.

Attribute	Definition
Section 5.2.299, "postalAddress"	Contains the mailing address for the entry.
Section 5.2.300, "postalCode"	Gives the postal code for the entry, such as the zip code in the United States.
Section 5.2.301, "postOfficeBox"	Gives the post office box number for the entry.
Section 5.2.302, "preferredDeliveryMethod"	Gives the preferred method of being contacted.
Section 5.2.310, "registeredAddress"	Gives a postal address suitable to receive expedited documents when the recipient must verify delivery.
Section 5.2.313, "searchGuide"	Specifies information for suggested search criteria when using the entry as the base object in the directory tree for a search.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.330, "st (stateOrProvinceName)"	Gives the state or province where the person is located.
Section 5.2.331, "street"	Gives the street name and number for the role's physical location.
Section 5.2.337, "telephoneNumber"	Gives the telephone number for the entry.
Section 5.2.338, "teletexTerminalIdentifier"	Gives the ID for an entry's teletex terminal.
Section 5.2.339, "telexNumber"	Gives the telex number associated with the entry.
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.
Section 5.2.355, "x121Address"	Gives the X.121 address for the entry.

5.3.100. person

The **person** object class represents entries for generic people. This is the base object class for the **organizationalPerson** object class.

This object class is defined in [RFC 2256](#).

Superior Class

[top](#)

OID

2.5.6.6

Table 5.177. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.329, "sn (surname)"	Gives the person's family name or last name.

Table 5.178. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.337, "telephoneNumber"	Gives the telephone number for the entry.
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.

5.3.101. pilotObject

The **pilotObject** is a subclass to allow additional attributes to be assigned to entries of all other object classes.

This object class is defined in [RFC 1274](#).

Superior Class

top

OID

0.9.2342.19200300.100.4.3

Table 5.179. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.180. Allowed Attributes

Attribute	Definition
Section 5.2.12, "audio"	Stores a sound file in a binary format.
Section 5.2.40, "dITRedirect"	Contains the DN (distinguished name) of the entry to use as a redirect for the entry.
Section 5.2.73, "info"	Contains information about the entry.
Section 5.2.84, "jpegPhoto"	Stores a JPG image.
Section 6.13, "lastModifiedBy"	Gives the DN (distinguished name) of the last user which modified the document entry.
Section 6.14, "lastModifiedTime"	Gives the time the object was most recently modified.
Section 5.2.105, "manager"	Gives the DN (distinguished name) of the entry's manager.
Section 5.2.297, "photo"	Stores a photo of the document in binary format.
Section 5.2.344, "uniqueIdentifier"	Distinguishes between two entries when a distinguished name has been reused.

5.3.102. pilotOrganization

The **pilotOrganization** object class is a subclass used to add attributes to **organization** and **organizationalUnit** object class entries.

This object class is defined in [RFC 1274](#).

Superior Class

top

OID

0.9.2342.19200300.100.4.20

Table 5.181. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the entry belongs.
Section 5.2.291, "ou (organizationalUnitName)"	Gives the organizational unit or division to which the entry belongs.

Table 5.182. Allowed Attributes

Attribute	Definition
Section 5.2.19, "buildingName"	Gives the name of the building where the entry is located.
Section 5.2.20, "businessCategory"	Gives the type of business in which the entry is engaged.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.38, "destinationIndicator"	Gives the country and city associated with the entry; this was once required to provide public telegram service.
Section 5.2.56, "fax (facsimileTelephoneNumber)"	Contains the fax number for the entry.
Section 5.2.76, "internationalISDNNumber"	Contains the ISDN number for the entry.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.298, "physicalDeliveryOfficeName"	Gives a location where physical deliveries can be made.
Section 5.2.299, "postalAddress"	Contains the mailing address for the entry.
Section 5.2.300, "postalCode"	Gives the postal code for the entry, such as the zip code in the United States.
Section 5.2.301, "postOfficeBox"	Gives the post office box number for the entry.
Section 5.2.302, "preferredDeliveryMethod"	Gives the preferred method of being contacted.
Section 5.2.310, "registeredAddress"	Gives a postal address suitable to receive expedited documents when the recipient must verify delivery.
Section 5.2.313, "searchGuide"	Specifies information for suggested search criteria when using the entry as the base object in the directory tree for a search.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.330, "st (stateOrProvinceName)"	Gives the state or province where the person is located.
Section 5.2.331, "street"	Gives the street name and address number for the person's physical location.

Attribute	Definition
Section 5.2.337, "telephoneNumber"	Gives the telephone number for the entry.
Section 5.2.338, "teletexTerminalIdentifier"	Gives the ID for an entry's teletex terminal.
Section 5.2.339, "telexNumber"	Gives the telex number associated with the entry.
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.
Section 5.2.355, "x121Address"	Gives the X.121 address for the entry.

5.3.103. pkiCA

The **pkiCA** auxiliary object class contains required or available certificates that are configured for a certificate authority. This object class is defined in [RFC 4523](#), which defines object classes and attributes for LDAP to use to manage X.509 certificates and related certificate services.

Superior Class

top

OID

2.5.6.22

Table 5.183. Allowed Attributes

Attribute	Definition
Section 5.2.14, "authorityRevocationList"	Contains a list of revoked CA certificates.
Section 5.2.22, "cACertificate"	Contains a CA certificate.
Section 5.2.24, "certificateRevocationList"	Contains a list of certificates that have been revoked.
Section 5.2.33, "crossCertificatePair"	Contains a pair of certificates that are used to cross-certify a pair of CAs in a FBKA-style bridge CA configuration.

5.3.104. pkiUser

The **pkiUser** auxiliary object class contains required certificates for a user or client that connects to a certificate authority or element in the public key infrastructure. This object class is defined in [RFC 4523](#), which defines object classes and attributes for LDAP to use to manage X.509 certificates and related certificate services.

Superior Class

[top](#)**OID**

2.5.6.21

Table 5.184. Allowed Attributes

Attribute	Definition
Section 5.2.348, "userCertificate"	Stores a user's certificate, usually in binary form.

5.3.105. posixAccount

The **posixAccount** object class defines network accounts which use POSIX attributes. This object class is defined in [RFC 2307](#), which defines object classes and attributes to use LDAP as a network information service.

**NOTE**

This object class is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-instance/schema** directory.

Superior Class[top](#)**OID**

1.3.6.1.1.2.0

Table 5.185. Required Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.59, "gidNumber"	Contains a unique numeric identifier for a group entry or to identify the group for a user entry, analogous to the group number in Unix.
Section 5.2.61, "homeDirectory"	Contains the path to the user's home directory.
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.342, "uid (userID)"	Gives the defined account's user ID.
Section 5.2.343, "uidNumber"	Contains a unique numeric identifier for a user entry, analogous to the user number in Unix.

Table 5.186. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.57, "gecos"	Used to determine the GECOS field for the user; this is based on a common name, with additional information embedded.
Section 5.2.89, "loginShell"	Contains the path to a script that is launched automatically when a user logs into the domain.
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.

5.3.106. posixGroup

The **posixGroup** object class defines a group of network accounts which use POSIX attributes. This object class is defined in [RFC 2307](#), which defines object classes and attributes to use LDAP as a network information service.

Superior Class

top

OID

1.3.6.1.1.2.2

Table 5.187. Required Attributes

Attribute	Definition
Section 5.2.59, "gidNumber"	Contains the path to a script that is launched automatically when a user logs into the domain.
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.188. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.110, "memberUid"	Gives the login name of the group member; this possibly may not be the same as the member's DN.
Section 5.2.350, "userPassword"	Contains the login name of the member of a group.

5.3.107. referral

The **referral** object class defines an object which supports LDAPv3 smart referrals. This object class is defined in LDAPv3 referrals Internet Draft.

Superior Class

top

OID

2.16.840.1.113730.3.2.6

Table 5.189. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 5.190. Allowed Attributes

Attribute	Definition
Section 5.2.309, "ref"	Contains information for an LDAPv3 smart referral.

5.3.108. residentialPerson

The **residentialPerson** object class manages a person's residential information.

This object class is defined in [RFC 2256](#).

Superior Class

top

OID

2.5.6.10

Table 5.191. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.329, "sn (surname)"	Gives the person's family name or last name.

Table 5.192. Allowed Attributes

Attribute	Definition
Section 5.2.20, "businessCategory"	Gives the type of business in which the entry is engaged.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.38, "destinationIndicator"	Gives the country and city associated with the entry; this was once required to provide public telegram service.
Section 5.2.56, "fax (facsimileTelephoneNumber)"	Contains the fax number for the entry.
Section 5.2.76, "internationalISDNNumber"	Contains the ISDN number for the entry.
Section 5.2.298, "physicalDeliveryOfficeName"	Gives a location where physical deliveries can be made.
Section 5.2.299, "postalAddress"	Contains the mailing address for the entry.
Section 5.2.300, "postalCode"	Gives the postal code for the entry, such as the zip code in the United States.
Section 5.2.301, "postOfficeBox"	Gives the post office box number for the entry.
Section 5.2.302, "preferredDeliveryMethod"	Shows the person's preferred method of contact or message delivery.
Section 5.2.310, "registeredAddress"	Gives a postal address suitable to receive expedited documents when the recipient must verify delivery.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.330, "st (stateOrProvinceName)"	Gives the state or province where the person is located.
Section 5.2.331, "street"	Gives the street name and address number for the person's physical location.
Section 5.2.337, "telephoneNumber"	Gives the telephone number for the entry.
Section 5.2.338, "teletexTerminalIdentifier"	Gives the ID for an entry's teletex terminal.
Section 5.2.339, "telexNumber"	Gives the telex number associated with the entry.
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.

Attribute	Definition
Section 5.2.355, "x121Address"	Gives the X.121 address for the entry.

5.3.109. RFC822LocalPart

The **RFC822LocalPart** object class defines entries that represent the local part of RFC 822 mail addresses. The directory treats this part of an RFC822 address as a domain.

This object class is defined by the Internet Directory Pilot.

Superior Class

domain

OID

0.9.2342.19200300.100.4.14

Table 5.193. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.34, "dc (domainComponent)"	Contains one component of a domain name.

Table 5.194. Allowed Attributes

Attribute	Definition
Section 5.2.10, "associatedName"	Gives the name of an entry within the organizational directory tree which is associated with a DNS domain.
Section 5.2.20, "businessCategory"	Gives the type of business in which the entry is engaged.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.38, "destinationIndicator"	Gives the country and city associated with the entry; this was once required to provide public telegram service.
Section 5.2.56, "fax (facsimileTelephoneNumber)"	Contains the fax number for the entry.
Section 5.2.76, "internationalISDNNumber"	Contains the ISDN number for the entry.

Attribute	Definition
Section 5.2.87, "l (localityName)"	Gives the city or geographical location of the entry.
Section 5.2.283, "o (organizationName)"	Gives the organization to which the account belongs.
Section 5.2.298, "physicalDeliveryOfficeName"	Gives a location where physical deliveries can be made.
Section 5.2.299, "postalAddress"	Contains the mailing address for the entry.
Section 5.2.300, "postalCode"	Gives the postal code for the entry, such as the zip code in the United States.
Section 5.2.301, "postOfficeBox"	Gives the post office box number for the entry.
Section 5.2.302, "preferredDeliveryMethod"	Shows the person's preferred method of contact or message delivery.
Section 5.2.310, "registeredAddress"	Gives a postal address suitable to receive expedited documents when the recipient must verify delivery.
Section 5.2.313, "searchGuide"	Specifies information for suggested search criteria when using the entry as the base object in the directory tree for a search.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.329, "sn (surname)"	Gives the person's family name or last name.
Section 5.2.330, "st (stateOrProvinceName)"	Gives the state or province where the person is located.
Section 5.2.331, "street"	Gives the street name and address number for the person's physical location.
Section 5.2.337, "telephoneNumber"	Gives the telephone number for the entry.
Section 5.2.338, "teletexTerminalIdentifier"	Gives the identifier for the person's teletex terminal.
Section 5.2.339, "telexNumber"	Gives the telex number associated with the entry.
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.
Section 5.2.355, "x121Address"	Gives the X.121 address for the entry.

5.3.110. room

The **room** object class stores information in the directory about rooms.

Superior Class

top

OID

0.9.2342.19200300.100.4.7

Table 5.195. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.25, "cn (commonName)"	Gives the common name of the entry.

Table 5.196. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the room.
Section 5.2.312, "roomNumber"	Contains the room's number.
Section 5.2.315, "seeAlso"	Contains a URL to another entry or site with related information.
Section 5.2.337, "telephoneNumber"	Gives the telephone number for the entry.

5.3.111. shadowAccount

The **shadowAccount** object class allows the LDAP directory to be used as a shadow password service. Shadow password services relocate the password files on a host to a shadow file with tightly restricted access.

This object class is defined in [RFC 2307](#), which defines object classes and attributes to use LDAP as a network information service.



NOTE

This object class is defined in **10rfc2307.ldif** in the Directory Server. To use the updated RFC 2307 schema, remove the **10rfc2307.ldif** file and copy the **10rfc2307bis.ldif** file from the **/usr/share/dirsrv/data** directory to the **/etc/dirsrv/slappd-instance/schema** directory.

Superior Class

top

OID

1.3.6.1.1.2.1

Table 5.197. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.342, "uid (userID)"	Gives the defined account's user ID.

Table 5.198. Allowed Attributes

Attribute	Definition
Section 5.2.37, "description"	Gives a text description of the entry.
Section 5.2.321, "shadowExpire"	Contains the date that the shadow account expires.
Section 5.2.322, "shadowFlag"	Identifies what area in the shadow map stores the flag values.
Section 5.2.323, "shadowInactive"	Sets how long the shadow account can be inactive.
Section 5.2.324, "shadowLastChange"	Contains the time and date of the last modification to the shadow account.
Section 5.2.325, "shadowMax"	Sets the maximum number of days that a shadow password is valid.
Section 5.2.326, "shadowMin"	Sets the minimum number of days that must pass between changing the shadow password.
Section 5.2.327, "shadowWarning"	Sets how many days in advance of password expiration to send a warning to the user.
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.

5.3.112. simpleSecurityObject

The **simpleSecurityObject** object class allows an entry to contain the **userPassword** attribute when an entry's principal object classes do not allow a password attribute. Reserved for future use.

This object class is defined in [RFC 1274](#).

Superior Class

top

OID

0.9.2342.19200300.100.4.19

Table 5.199. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.350, "userPassword"	Stores the password with which the entry can bind to the directory.

5.3.113. strongAuthenticationUser

The **strongAuthenticationUser** object class stores a user's certificate in the directory.

This object class is defined in [RFC 2256](#).

Superior Class

top

OID

2.5.6.15

Table 5.200. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.
Section 5.2.348, "userCertificate"	Stores a user's certificate, usually in binary form.

CHAPTER 6. OPERATIONAL ATTRIBUTES AND OBJECT CLASSES

Operational attributes are attributes used to perform directory operations and are available for every entry in the directory, regardless of whether they are defined for the object class of the entry.

Operational attributes are only returned in an **ldapsearch** operation if specifically requested. To return all operational attributes of an object, specify **+**.

Operational attributes are created and managed by Directory Server on entries, such as the time the entry is created or modified and the creator's name. These attributes can be set on any entry, regardless of other attributes or object classes on the entry.

6.1. ACCOUNTUNLOCKTIME

The **accountUnlockTime** attribute contains the date and time in GMT-format at which the account will become unlocked. A value of **0** means that the account must be unlocked by an administrator.

OID	2.16.840.1.113730.3.1.95
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

6.2. ACI

This attribute is used by the Directory Server to evaluate what rights are granted or denied when it receives an LDAP request from a client.

OID	2.16.840.1.113730.3.1.55
Syntax	IA5String
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

6.3. ALTSERVER

The values of this attribute are URLs of other servers which may be contacted when this server becomes unavailable. If the server does not know of any other servers which could be used, this attribute is absent. This information can be cached in case the preferred LDAP server later becomes unavailable.

OID	1.3.6.1.4.1.1466.101.120.6
Syntax	IA5String

Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

6.4. CREATETIMESTAMP

This attribute contains the date and time that the entry was initially created.

OID	2.5.18.1
Syntax	GeneralizedTime
Multi- or Single-Valued	Single-valued
Defined in	RFC 1274

6.5. CREATORSNAME

This attribute contains the name of the user which created the entry.

OID	2.5.18.3
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	RFC 1274

6.6. DITCONTENTRULES

This attribute defines the DIT content rules which are in force within a subschema. Each value defines one DIT content rule. Each value is tagged by the object identifier of the structural object class to which it pertains.

OID	2.5.21.2
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

6.7. DITSTRUCTURERULES

This attribute defines the DIT structure rules which are in force within a subschema. Each value defines one DIT structure rule.

OID	2.5.21.1
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

6.8. ENTRYUSN

When the USN Plug-in is enabled, the server automatically assigns an *update sequence number* to entries every time a write operation (add, modify, modrdn, or delete) is performed. The USN is stored in the **entryUSN** operational attribute on the entry; the **entryUSN**, then, shows the number for the most recent change on any entry.



NOTE

The **entryUSN** attribute increments only with operations performed by LDAP clients. It does not count internal operations.

By default, the **entryUSN** is unique per back end database instance, so entries in other databases may have the same USN. The **nsslapd-entryusn-global** parameter changes the assignment of USNs from local to global, that is, from being counted on a single database to being counted for all databases in the topology. The parameter is turned off by default.

A corresponding entry, **lastusn**, is kept in the root DSE entry, which shows the most recently- assigned USN. In *local* mode, **lastusn** shows the most recently- assigned USN per back end database. In *global* mode, **lastusn** shows the most recently assigned USN for the entire topology.

OID	2.16.840.1.113730.3.1.606
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.9. INTERNALCREATORSNAME

For entries which were created by a plug-in or by the server, rather than a Directory Server user, this attribute records what internal user (by plug-in DN) created the entry.

The **internalCreatorsname** attributes always show a plug-in as the identity. This plug-in could be an additional plug-in, such as the MemberOf Plug-in. If the change is made by the core Directory Server, then the plug-in is the database plug-in, **cn=ldbm database,cn=plugins,cn=config**.

OID	2.16.840.1.113730.3.1.2114
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.10. INTERNALMODIFIERSNAME

If an entry is edited by a plug-in or by the server, rather than a Directory Server user, this attribute records what internal user (by plug-in DN) modified the entry.

The **internalModifiersname** attributes always show a plug-in as the identity. This plug-in could be an additional plug-in, such as the MemberOf Plug-in. If the change is made by the core Directory Server, then the plug-in is the database plug-in, **cn=ldbm database, cn=plugins, cn=config**.

OID	2.16.840.1.113730.3.1.2113
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.11. HASSUBORDINATES

This attribute indicates whether the entry has subordinate entries.

OID	1.3.6.1.4.1.1466.115.121.1.7
Syntax	Boolean
Multi- or Single-Valued	Single-valued
Defined in	numSubordinates Internet Draft

6.12. LASTLOGINTIME

The **lastLoginTime** attribute contains a timestamp of the last time that the given account authenticated to the directory, in the format **YYYYMMDDHHMMSSZ**. For example:

lastLoginTime: 20200527001051Z

This is used to evaluate account lockout policies based on account inactivity.

OID	2.16.840.1.113719.1.1.4.1.35
Syntax	GeneralizedTime
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.13. LASTMODIFIEDBY

The **lastModifiedBy** attribute contains the distinguished name (DN) of the user who last edited the entry. For example:

lastModifiedBy: cn=Barbara Jensen,ou=Engineering,dc=example,dc=com

OID	0.9.2342.19200300.100.1.24
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

6.14. LASTMODIFIEDTIME

The **lastModifiedTime** attribute contains the time, in UTC format, an entry was last modified. For example:

lastModifiedTime: Thursday, 22-Sep-93 14:15:00 GMT

OID	0.9.2342.19200300.100.1.23
Syntax	DirectString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 1274

6.15. LDAPSUBENTRY

These entries hold operational data. This object class is defined in the LDAP Subentry Internet Draft.

Superior Class

top

OID

2.16.840.1.113719.2.142.6.1.1

Table 6.1. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 6.2. Allowed Attributes

Attribute	Definition
Section 5.2.25, "cn (commonName)"	Specifies the common name of the entry.

6.16. LDAPSNTAXES

This attribute identifies the syntaxes implemented, with each value corresponding to one syntax.

OID	1.3.6.1.4.1.1466.101.120.16
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

6.17. MATCHINGRULES

This attribute defines the matching rules used within a subschema. Each value defines one matching rule.

OID	2.5.21.4
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

6.18. MATCHINGRULEUSE

This attribute indicates the attribute types to which a matching rule applies in a subschema.

OID	2.5.21.8
-----	----------

Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

6.19. MODIFYTIMESTAMP

This attribute contains the date and time that the entry was most recently modified.

OID	2.5.18.2
Syntax	GeneralizedTime
Multi- or Single-Valued	Single-valued
Defined in	RFC 1274

6.20. MODIFIERSNAME

This attribute contains the name of the user which last modified the entry.

OID	2.5.18.4
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	RFC 1274

6.21. NAMEFORMS

This attribute defines the name forms used in a subschema. Each value defines one name form.

OID	2.5.21.7
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	RFC 2252

6.22. NSACCOUNTLOCK

This attribute shows whether the account is active or inactive.

OID	2.16.840.1.113730.3.1.610
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

6.23. NSAIMSTATUSGRAPHIC

This attribute contains a path pointing to the graphic which illustrates the AIM user status.

OID	2.16.840.1.113730.3.1.2018
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.24. NSAIMSTATUSTEXT

This attribute contains the text which indicates the current AIM user status.

OID	2.16.840.1.113730.3.1.2017
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.25. NSBACKENDSUFFIX

This contains the suffix used by the back end.

OID	2.16.840.1.113730.3.1.803
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

6.26. NSCPENTRYDN

This attribute contains the (former) entry DN for a tombstone entry.

OID	2.16.840.1.113730.3.1.545
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.27. NSDS5REPLCONFLICT

This attribute is included on entries that have a change conflict that cannot be resolved automatically by the synchronization or replication process. The value of the **nsDS5RepConflict** contains information about which entries are in conflict, usually by referring to them by their **nsUniqueId** for both current entries and tombstone entries.

OID	2.16.840.1.113730.3.1.973
Syntax	DirectoryString
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

6.28. NSICQSTATUSGRAPHIC

This attribute contains a path pointing to the graphic which illustrates the ICQ user status.

OID	2.16.840.1.113730.3.1.2022
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.29. NSICQSTATUSTEXT

This attribute contains the text for the current ICQ user status.

OID	2.16.840.1.113730.3.1.2021
-----	----------------------------

Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.30. NSIDLETIMEOUT

This attribute identifies the user-based connection idle timeout period, in seconds.

OID	2.16.840.1.113730.3.1.573
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.31. NSIDLISTSCANLIMIT

This attribute specifies the number of entry IDs that are searched during a search operation. Keep the default value to improve search performance. For a more detailed explanation of the effect of ID lists on search performance, see the "Overview of the Searching Algorithm" section of the "Managing Indexes" chapter in the *Red Hat Directory Server Administration Guide*.

OID	2.16.840.1.113730.3.1.2106
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.32. NSLOOKTHROUGHLIMIT

This attribute sets the maximum number of entries for that user through which the server is allowed to look during a search operation. This attribute is configured in the server itself and applied to a user when he initiates a search.

OID	2.16.840.1.113730.3.1.570
Syntax	Integer
Multi- or Single-Valued	Single-valued

Defined in	Directory Server
------------	------------------

6.33. NSPAGEDIDLISTSCANLIMIT

This attribute specifies the number of entry IDs that are searched, specifically, for a search operation using the simple paged results control. This attribute works the same as the **nsIDListScanLimit** attribute, except that it only applies to searches with the simple paged results control.

If this attribute is not present or is set to zero, then the **nsIDListScanLimit** is used to paged searches as well as non-paged searches.

OID	2.16.840.1.113730.3.1.2109
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.34. NSPAGEDLOOKTHROUGHLIMIT

This attribute specifies the maximum number of entries that the Directory Server will check when examining candidate entries for a search which uses the simple paged results control. This attribute works the same as the **nsLookThroughLimit** attribute, except that it only applies to searches with the simple paged results control.

If this attribute is not present or is set to zero, then the **nsLookThroughLimit** is used to paged searches as well as non-paged searches.

OID	2.16.840.1.113730.3.1.2108
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.35. NSPAGEDSIZELIMIT

This attribute sets the maximum number of entries to return from a search operation *specifically which uses the simple paged results control*. This overrides the **nsSizeLimit** attribute for paged searches.

If this value is set to zero, then the **nsSizeLimit** attribute is used for paged searches as well as non-paged searches for the user, or the global configuration settings are used.

OID	2.16.840.1.113730.3.1.2107
-----	----------------------------

Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.36. NSPARENTUNIQUEID

For tombstone (deleted) entries stored in replication, the **nsParentUniqueId** attribute contains the DN or entry ID for the parent of the original entry.

OID	2.16.840.1.113730.3.1.544
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.37. NSROLE

This attribute is a computed attribute that is not stored with the entry itself. It identifies to which roles an entry belongs.

OID	2.16.840.1.113730.3.1.574
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

6.38. NSROLEDN

This attribute contains the distinguished name of all roles that apply to an entry. Membership of a managed role is granted upon an entry by adding the role's DN to the entry's **nsRoleDN** attribute. For example:

```
dn: cn=staff,ou=employees,dc=example,dc=com
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsSimpleRoleDefinition
objectclass: nsManagedRoleDefinition

dn: cn=userA,ou=users,ou=employees,dc=example,dc=com
objectclass: top
objectclass: person
```

```

sn: uA
userpassword: secret
nsroledn: cn=staff,ou=employees,dc=example,dc=com

```

A nested role specifies containment of one or more roles of any type. In that case, **nsRoleDN** defines the DN of the contained roles. For example:

```

dn: cn=everybody,ou=employees,dc=example,dc=com
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsNestedRoleDefinition
nsroledn: cn=manager,ou=employees,dc=example,dc=com
nsroledn: cn=staff,ou=employees,dc=example,dc=com

```

OID	2.16.840.1.113730.3.1.575
Syntax	DN
Multi- or Single-Valued	Multi-valued
Defined in	Directory Server

6.39. NSROLEFILTER

This attribute sets the filter identifies entries which belong to the role.

OID	2.16.840.1.113730.3.1.576
Syntax	IA5String
Multi- or Single-Valued	Single-valued
Defined in	RFC 2252

6.40. NSSCHEMACSN

This attribute is one of the subschema DSE attribute types.

OID	2.5.21.82.16.840.1.113730.3.1.804
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.41. NSSIZELIMIT

This attribute shows the default size limit for a database or database link in bytes.

OID	2.16.840.1.113730.3.1.571
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.42. NSTIMELIMIT

This attribute shows the default search time limit for a database or database link.

OID	2.16.840.1.113730.3.1.572
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.43. NSTOMBSTONE (OBJECT CLASS)

Tombstone entries are entries which have been deleted from Directory Server. For replication and restore operations, these deleted entries are saved so that they can be resurrected and replaced if necessary. Each tombstone entry has the **nsTombstone** object class, automatically.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.113

Table 6.3. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

Table 6.4. Allowed Attributes

Attribute	Definition
Section 6.36, "nsParentUniqueId"	Identifies the unique ID of the parent entry of the original entry.
Section 6.26, "nscpEntryDN"	Identifies the original entry DN in a tombstone entry.

6.44. NSUNIQUEID

This attribute identifies or assigns a unique ID to a server entry.

OID	2.16.840.1.113730.3.1.542
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.45. NSYIMSTATUSGRAPHIC

This attribute contains a path pointing to the graphic which illustrates the Yahoo IM user status.

OID	2.16.840.1.113730.3.1.2020
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.46. NSYIMSTATUSTEXT

This attribute contains the text for the current Yahoo IM user status.

OID	2.16.840.1.113730.3.1.2019
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.47. NUMSUBORDINATES

This attribute indicates how many immediate subordinates an entry has. For example, **numSubordinates=0** in a leaf entry.

OID	1.3.1.1.4.1.453.16.2.103
Syntax	Integer
Multi- or Single-Valued	Single-valued
Defined in	numSubordinates Internet Draft

6.48. PASSWORDGRACEUSERTIME

This attribute counts the number of attempts the user has made with the expired password.

OID	2.16.840.1.113730.3.1.998
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.49. PASSWORDRETRYCOUNT

This attribute counts the number of consecutive failed attempts at entering the correct password.

OID	2.16.840.1.113730.3.1.93
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.50. PWDPOLICYSUBENTRY

This attribute value points to the entry DN of the new password policy.

OID	2.16.840.1.113730.3.1.997
Syntax	DirectoryString
Multi- or Single-Valued	Single-valued

Defined in	Directory Server
------------	------------------

6.51. PWDUPDATETIME

This attribute value stores the time of the most recent password change for the account.

OID	2.16.840.1.113730.3.1.2133
Syntax	GeneralizedTime
Multi- or Single-Valued	Single-valued
Defined in	Directory Server

6.52. SUBSCHEMASUBENTRY

This attribute contains the DN of an entry that contains schema information. For example:

subschemaSubentry: cn=schema

OID	2.5.18.10
Syntax	DN
Multi- or Single-Valued	Single-valued
Defined in	RFC 2252

6.53. GLUE (OBJECT CLASS)

The **glue** object class defines an entry in a special state: resurrected due to a replication conflict.

This object class is defined by Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.30

Table 6.5. Required Attributes

Attribute	Definition
Section 5.2.284, "objectClass"	Gives the object classes assigned to the entry.

6.54. PASSWORDOBJECT (OBJECT CLASS)

This object class is used for entries which store password information for a user in the directory.

This object class is defined in Directory Server.

Superior Class

top

OID

2.16.840.1.113730.3.2.12

Table 6.6. Required Attributes

Section 5.2.284, "objectClass"	Defines the object classes for the entry.
--	---

Table 6.7. Allowed Attributes

Section 6.1, "accountUnlockTime"	Refers to the amount of time that must pass after an account lockout before the user can bind to the directory again.
Section 3.1.1.176, "passwordAllowChangeTime"	Specifies the length of time that must pass before users are allowed to change their passwords.
Section 3.1.1.181, "passwordExpirationTime"	Specifies the length of time that passes before the user's password expires.
Section 3.1.1.182, "passwordExpWarned"	Indicates that a password expiration warning has been sent to the user.
Section 6.48, "passwordGraceUserTime"	Specifies the number of login attempts that are allowed to a user after the password has expired.
Section 3.1.1.184, "passwordHistory (Password History)"	Contains the history of the user's previous passwords.
Section 6.49, "passwordRetryCount"	Counts the number of consecutive failed attempts at entering the correct password.
Section 6.50, "pwdpolicysubentry"	Points to the entry DN of the new password policy.
Section 3.1.1.219, "retryCountResetTime"	Specifies the length of time that passes before the passwordRetryCount attribute is reset.

6.55. SUBSCHEMA (OBJECT CLASS)

This identifies an auxiliary object class subentry which administers the subschema for the subschema administrative area. It holds the operational attributes representing the policy parameters which express the subschema.

This object class is defined in RFC 2252.

Superior Class

top

OID

2.5.20.1

Table 6.8. Required Attributes

Section 5.2.284, "objectClass"	Defines the object classes for the entry.
--	---

Table 6.9. Allowed Attributes

Section 5.2.11, "attributeTypes"	Attribute types used within a subschema.
Section 6.6, "dITContentRules"	Defines the DIT content rules which are in force within a subschema.
Section 6.7, "dITStructureRules"	Defines the DIT structure rules which are in force within a subschema.
Section 6.18, "matchingRuleUse"	Indicates the attribute types to which a matching rule applies in a subschema.
Section 6.17, "matchingRules"	Defines the matching rules used within a subschema.
Section 6.21, "nameForms"	Defines the name forms used in a subschema.
Section 5.2.285, "objectClasses"	Defines the object classes used in a subschema.

CHAPTER 7. LOG FILE REFERENCE

Red Hat Directory Server (Directory Server) provides logs to help monitor directory activity. Monitoring helps quickly detecting and remedying failures and, where done proactively, anticipating and resolving potential problems before they result in failure or poor performance. Part of monitoring the directory effectively is understanding the structure and content of the log files.

This chapter does not provide an exhaustive list of log messages. However, the information presented in this chapter serves as a good starting point for common problems and for better understanding the information in the access, error, and audit logs.

Logs are kept per Directory Server instances and are located in the **/var/log/dirsrv/slappd-*instance*** directory.

7.1. ACCESS LOG REFERENCE

The Directory Server access log contains detailed information about client connections to the directory. A connection is a sequence of requests from the same client with the following structure:

- Connection record, which provides the connection index and the IP address of the client
- Bind record
- Bind result record
- Sequence of operation request and operation result pairs of records, or individual records in the case of connection, closed, and abandon records
- Unbind record
- Closed record

The following is an example access log entry:

```
[23/Jun/2020:16:30:27.388006333 -0400] conn=20 op=5 SRCH base="dc=example,dc=com"
scope=2 filter="(&(objectClass=top)(objectClass=ldapsubentry)(objectClass=passwordpolicy))"
attrs="distinguishedName"
```

Apart from connection, closed, and abandon records, which appear individually, all records appear in pairs, consisting of a request for service record followed by a **RESULT** record:

```
[23/Jun/2020:16:30:27.390881301 -0400] conn=20 op=5 RESULT err=0 tag=101 nentries=0
wtime=0.000035342 optime=0.002877749 etime=0.002911121
```

The **RESULT** message contains the following performance-related entries:

- **wtime**: The amount of time the operation was waiting in the work queue before a worker thread picked up the operation
- **optime**: The amount of time it took for the actual operation to perform the task
- **etime**: The elapsed time, which covers the time the operation was received by the server to when the server sent back the result to the client



NOTE

The **wtime** and **optime** values provide useful information about how the server handles load and processes operations. Due to the timing of when Directory Server gathers these statistics, the sum of the **wtime** and **optime** values are slightly greater than the **etime** value. However, this very small difference is negligible.

The access logs have different levels of logging, set in the **nsslapd-accesslog-level** attribute. The following sections provide an overview of the default access logging content, log levels, and the content logged at different logging levels:

- [Section 7.1.1, "Access Logging Levels"](#)
- [Section 7.1.2, "Default Access Logging Content"](#)
- [Section 7.1.3, "Access Log Content for Additional Access Logging Levels"](#)

Note that you cannot change the format of the access log.

7.1.1. Access Logging Levels

Different levels of access logging generate different amounts of detail and record different kinds of operations. The log level is set in the instance's [Section 3.1.1.2, "nsslapd-accesslog-level \(Access Log Level\)" configuration attribute](#). The default level of logging is level 256, which logs access to an entry, but there are five different log levels available:

- 0 = No access logging.
- 4 = Logging for internal access operations.
- 256 = Logging for access to an entry.
- 512 = Logging for access to an entry and referrals.

This levels are additive, so to enable several different kinds of logging, add the values of those levels together. For example, to log internal access operations, entry access, and referrals, set the value of **nsslapd-accesslog-level** to **516 (512+4)**.

7.1.2. Default Access Logging Content

This section describes the access log content in detail based on the default access logging level extract shown below.

Example 7.1. Example Access Log

```
[21/Apr/2020:11:39:51 -0700] conn=11 fd=608 slot=608 connection from 207.1.153.51 to
192.18.122.139
[21/Apr/2020:11:39:51 -0700] conn=11 op=0 BIND dn="cn=Directory Manager" method=128
version=3
[21/Apr/2020:11:39:51 -0700] conn=11 op=0 RESULT err=0 tag=97 nentries=0 etime=0
[21/Apr/2020:11:39:51 -0700] conn=11 op=1 SRCH base="dc=example,dc=com" scope=2 filter=
"(mobile=+1 123 456-7890)"
[21/Apr/2020:11:39:51 -0700] conn=11 op=1 RESULT err=0 tag=101 nentries=1 etime=3 notes=U
[21/Apr/2020:11:39:51 -0700] conn=11 op=2 UNBIND
[21/Apr/2020:11:39:51 -0700] conn=11 op=2 fd=608 closed - U1
```

```
[21/Apr/2020:11:39:52 -0700] conn=12 fd=634 slot=634 connection from 207.1.153.51 to
192.18.122.139
[21/Apr/2020:11:39:52 -0700] conn=12 op=0 BIND dn="cn=Directory Manager" method=128
version=3
[21/Apr/2020:11:39:52 -0700] conn=12 op=0 RESULT err=0 tag=97 nentries=0 etime=0
[21/Apr/2020:11:39:52 -0700] conn=12 op=1 SRCH base="dc=example,dc=com" scope=2 filter="
(uid=bjensen)"
[21/Apr/2020:11:39:52 -0700] conn=12 op=2 ABANDON targetop=1 msgid=2 nentries=0 etime=0
[21/Apr/2020:11:39:52 -0700] conn=12 op=3 UNBIND
[21/Apr/2020:11:39:52 -0700] conn=12 op=3 fd=634 closed - U1
[21/Apr/2020:11:39:53 -0700] conn=13 fd=659 slot=659 connection from 207.1.153.51 to
192.18.122.139
[21/Apr/2020:11:39:53 -0700] conn=13 op=0 BIND dn="cn=Directory Manager" method=128
version=3
[21/Apr/2020:11:39:53 -0700] conn=13 op=0 RESULT err=0 tag=97 nentries=0 etime=0
[21/Apr/2020:11:39:53 -0700] conn=13 op=1 EXT oid="2.16.840.1.113730.3.5.3"
[21/Apr/2020:11:39:53 -0700] conn=13 op=1 RESULT err=0 tag=120 nentries=0 etime=0
[21/Apr/2020:11:39:53 -0700] conn=13 op=2 ADD dn="cn=Sat Apr 21 11:39:51 MET DST
2020,dc=example,dc=com"
[21/Apr/2020:11:39:53 -0700] conn=13 op=2 RESULT err=0 tag=105 nentries=0 etime=0
csn=3b4c8cfb000000030000
[21/Apr/2020:11:39:53 -0700] conn=13 op=3 EXT oid="2.16.840.1.113730.3.5.5"
[21/Apr/2020:11:39:53 -0700] conn=13 op=3 RESULT err=0 tag=120 nentries=0 etime=0
[21/Apr/2020:11:39:53 -0700] conn=13 op=4 UNBIND
[21/Apr/2020:11:39:53 -0700] conn=13 op=4 fd=659 closed - U1
[21/Apr/2020:11:39:55 -0700] conn=14 fd=700 slot=700 connection from 207.1.153.51 to
192.18.122.139
[21/Apr/2020:11:39:55 -0700] conn=14 op=0 BIND dn="" method=sasl version=3 mech=DIGEST-
MD5
[21/Apr/2020:11:39:55 -0700] conn=14 op=0 RESULT err=14 tag=97 nentries=0 etime=0, SASL
bind in progress
[21/Apr/2020:11:39:55 -0700] conn=14 op=1 BIND dn="uid=jdoe,dc=example,dc=com"
method=sasl version=3 mech=DIGEST-MD5
[21/Apr/2020:11:39:55 -0700] conn=14 op=1 RESULT err=0 tag=97 nentries=0 etime=0
dn="uid=jdoe,dc=example,dc=com"
[21/Apr/2020:11:39:55 -0700] conn=14 op=2 UNBIND
[21/Apr/2020:11:39:53 -0700] conn=14 op=2 fd=700 closed - U1
```

Connection Number

Every external LDAP request is listed with an incremental connection number, in this case **conn=11**, starting at **conn=0** immediately after server startup.

```
[21/Apr/2020:11:39:51 -0700] conn=11 fd=608 slot=608 connection from 207.1.153.51 to
192.18.122.139
```

Internal LDAP requests are not recorded in the access log by default. To activate the logging of internal access operations, specify access logging level **4** on the [Section 3.1.1.2, “nsslapd-accesslog-level \(Access Log Level\)” configuration attribute](#).

File Descriptor

Every connection from an external LDAP client to Directory Server requires a file descriptor or socket descriptor from the operating system, in this case **fd=608**. **fd=608** indicates that it was file descriptor number 608 out of the total pool of available file descriptors which was used.

[21/Apr/2020:11:39:51 -0700] conn=11 **fd=608** slot=608 connection from 207.1.153.51 to 192.18.122.139

Slot Number

The slot number, in this case **slot=608**, is a legacy part of the access log which has the same meaning as file descriptor. Ignore this part of the access log.

[21/Apr/2020:11:39:51 -0700] conn=11 fd=608 **slot=608** connection from 207.1.153.51 to 192.18.122.139

Operation Number

To process a given LDAP request, Directory Server will perform the required series of operations. For a given connection, all operation request and operation result pairs are given incremental operation numbers beginning with **op=0** to identify the distinct operations being performed.

[21/Apr/2020:11:39:51 -0700] conn=11 **op=0** RESULT err=0 tag=97 nentries=0 etime=0

In [Section 7.1.2, "Default Access Logging Content"](#), we have **op=0** for the bind operation request and result pair, then **op=1** for the LDAP search request and result pair, and so on. The entry **op=-1** in the access log generally means that the LDAP request for this connection was not issued by an external LDAP client but, instead, initiated internally.

Method Type

The method number, in this case **method=128**, indicates which LDAPv3 bind method was used by the client.

[21/Apr/2020:11:39:51 -0700] conn=11 op=0 BIND dn="cn=Directory Manager" **method=128** version=3

There are three possible bind method values:

- **0** for authentication
- **128** for simple bind with user password
- **sasl** for SASL bind using external authentication mechanism

Version Number

The version number, in this case **version=3**, indicates the LDAP version number (either LDAPv2 or LDAPv3) that the LDAP client used to communicate with the LDAP server.

[21/Apr/2020:11:39:51 -0700] conn=11 op=0 BIND dn="cn=Directory Manager" method=128 **version=3**

Error Number

The error number, in this case **err=0**, provides the LDAP result code returned from the LDAP operation performed. The LDAP error number **0** means that the operation was successful. For a more comprehensive list of LDAP result codes, see [Section 7.4, "LDAP Result Codes"](#).

[21/Apr/2020:11:39:51 -0700] conn=11 op=0 RESULT **err=0** tag=97 nentries=0 etime=0

Tag Number

The tag number, in this case **tag=97**, indicates the type of result returned, which is almost always a reflection of the type of operation performed. The tags used are the BER tags from the LDAP protocol.

```
[21/Apr/2020:11:39:51 -0700] conn=11 op=0 RESULT err=0 tag=97 nentries=0 etime=0
```

Table 7.1. Commonly-Used Tags

Tag	Description
tag=97	Result from a client bind operation.
tag=100	The actual entry being searched for.
tag=101	Result from a search operation.
tag=103	Result from a modify operation.
tag=105	Result from an add operation.
tag=107	Result from a delete operation.
tag=109	Result from a moddn operation.
tag=111	Result from a compare operation.
tag=115	Search reference when the entry on which the search was performed holds a referral to the required entry. Search references are expressed in terms of a referral.
tag=120	Result from an extended operation.
tag=121	Result from an intermediate operation.



NOTE

tag=100 and **tag=115** are not result tags as such, and so it is unlikely that they will be recorded in the access log.

Number of Entries

nentries shows the number of entries, in this case **nentries=0**, that were found matching the LDAP client's request.

```
[21/Apr/2020:11:39:51 -0700] conn=11 op=0 RESULT err=0 tag=97 nentries=0 etime=0
```

Elapsed Time

etime shows the elapsed time, in this case **etime=3**, or the amount of time (in seconds) that it took the Directory Server to perform the LDAP operation.

```
[21/Apr/2020:11:39:51 -0700] conn=11 op=1 RESULT err=0 tag=101 nentries=1 etime=3 notes=U
```

An **etime** value of **0** means that the operation actually took 0 nanoseconds to perform.

LDAP Request Type

The LDAP request type indicates the type of LDAP request being issued by the LDAP client. Possible values are:

- **SRCH** for search
- **MOD** for modify
- **DEL** for delete
- **ADD** for add
- **MODDN** for moddn
- **EXT** for extended operation
- **ABANDON** for abandon operation

If the LDAP request resulted in sorting of entries, then the message **SORT serialno** will be recorded in the log, followed by the number of candidate entries that were sorted. For example:

```
[04/May/2020:15:51:46 -0700] conn=114 op=68 SORT serialno (1)
```

The number enclosed in parentheses specifies the number of candidate entries that were sorted, which in this case is **1**.

LDAP Response Type

The LDAP response type indicates the LDAP response being issued by the LDAP client. There are three possible values:

- **RESULT**
- **ENTRY**
- **REFERRAL**, an LDAP referral or search reference

Search Indicators

Directory Server provides additional information on searches in the **notes** field of log entries. For example:

```
[21/Apr/2016:11:39:51 -0700] conn=11 op=1 RESULT err=0 tag=101 nentries=1 etime=3 notes=U
```

The following search indicators exist:

Paged Search Indicator:**notes=P**

LDAP clients with limited resources can control the rate at which an LDAP server returns the results of a search operation. When the search performed used the LDAP control extension for simple

paging of search results, Directory Server logs the **notes=P** paged search indicator. This indicator is informational and no further actions are required.

For more details, see [RFC 2696](#).

Unindexed Search Indicators:**notes=A** and **notes=U**

When attributes are not indexed, Directory Server must search them in the database directly. This procedure is more resource-intensive than searching the index file.

The following unindexed search indicators can be logged:

- **notes=A**

All candidate attributes in the filter were unindexed and a full table scan was required. This can exceed the value set in the **nsslapd-lookthroughlimit** parameter.

- **notes=U**

This state is set in the following situations:

- At least one of the search terms is unindexed.

- The limit set in the **nsslapd-idlistscanlimit** parameter was reached during the search operation. For details, see [Section 4.4.1.9, “nsslapd-idlistscanlimit”](#).

Unindexed searches occur in the following scenarios:

- The **nsslapd-idlistscanlimit** parameter’s value was reached within the index file used for the search.

- No index file existed.

- The index file was not configured in the way required by the search.

To optimize future searches, add frequently searched unindexed attributes to the index. For details, see the corresponding section in the [Directory Server Administration Guide](#).



NOTE

An unindexed search indicator is often accompanied by a large **etime** value, as unindexed searches are generally more time consuming.

Beside a single value, the **notes** field can have the following value combinations: **notes=P,A** and **notes=U,P**.

VLV-Related Entries

When a search involves virtual list views (VLVs), appropriate entries are logged in the access log file. Similar to the other entries, VLV-specific entries show the request and response information side by side:

VLV RequestInformation ResponseInformation

RequestInformation has the following form:

beforeCount:afterCount:index:contentCount

If the client uses a position-by-value VLV request, the format for the first part, the request information would be *beforeCount: afterCount: value*.

ResponseInformation has the following form:

targetPosition:contentCount (resultCode)

The example below highlights the VLV-specific entries:

```
[07/May/2020:11:43:29 -0700] conn=877 op=8530 SRCH base="(ou=People)" scope=2 filter=
(uid=*)"
[07/May/2020:11:43:29 -0700] conn=877 op=8530 SORT uid
[07/May/2020:11:43:29 -0700] conn=877 op=8530 VLV 0:5:0210 10:5397 (0)
[07/May/2020:11:43:29 -0700] conn=877 op=8530 RESULT err=0 tag=101 nentries=1 etime=0
```

In the above example, the first part, **0:5:0210**, is the VLV request information:

- The beforeCount is **0**.
- The afterCount is **5**.
- The value is **0210**.

The second part, **10:5397 (0)**, is the VLV response information:

- The targetPosition is **10**.
- The contentCount is **5397**.
- The (resultCode) is **(0)**.

Search Scope

The entry **scope=n** defines the scope of the search performed, and **n** can have a value of **0**, **1**, or **2**.

- **0** for base search
- **1** for one-level search
- **2** for subtree search

Extended Operation OID

An extended operation OID, such as **EXT oid="2.16.840.1.113730.3.5.3"** or **EXT oid="2.16.840.1.113730.3.5.5"** in [Example 7.1, "Example Access Log"](#), provides the OID of the extended operation being performed. [Table 7.2, "LDAPv3 Extended Operations Supported by Directory Server"](#) provides a partial list of LDAPv3 extended operations and their OIDs supported in Directory Server.

Table 7.2. LDAPv3 Extended Operations Supported by Directory Server

Extended Operation Name	Description	OID
Directory Server Start Replication Request	Sent by a replication initiator to indicate that a replication session is requested.	2.16.840.1.113730.3.5.3

Extended Operation Name	Description	OID
Directory Server Replication Response	Sent by a replication responder in response to a Start Replication Request Extended Operation or an End Replication Request Extended Operation.	2.16.840.1.113730.3.5.4
Directory Server End Replication Request	Sent to indicate that a replication session is to be terminated.	2.16.840.1.113730.3.5.5
Directory Server Replication Entry Request	Carries an entry, along with its state information (csn and UniquelIdentifier) and is used to perform a replica initialization.	2.16.840.1.113730.3.5.6
Directory Server Bulk Import Start	Sent by the client to request a bulk import together with the suffix being imported to and sent by the server to indicate that the bulk import may begin.	2.16.840.1.113730.3.5.7
Directory Server Bulk Import Finished	Sent by the client to signal the end of a bulk import and sent by the server to acknowledge it.	2.16.840.1.113730.3.5.8

Change Sequence Number

The change sequence number, in this case **csn=3b4c8cfb000000030000**, is the replication change sequence number, indicating that replication is enabled on this particular naming context.

Abandon Message

The abandon message indicates that an operation has been aborted.

```
[21/Apr/2020:11:39:52 -0700] conn=12 op=2 ABANDON targetop=1 msgid=2 nentries=0 etime=0
```

nentries=0 indicates the number of entries sent before the operation was aborted, **etime=0** value indicates how much time (in seconds) had elapsed, and **targetop=1** corresponds to an operation value from a previously initiated operation (that appears earlier in the access log).

There are two possible log **ABANDON** messages, depending on whether the message ID succeeds in locating which operation was to be aborted. If the message ID succeeds in locating the operation (the **targetop**) then the log will read as above. However, if the message ID does not succeed in locating the operation or if the operation had already finished prior to the **ABANDON** request being sent, then the log will read as follows:

```
[21/Apr/2020:11:39:52 -0700] conn=12 op=2 ABANDON targetop=NOTFOUND msgid=2
```

targetop=NOTFOUND indicates the operation to be aborted was either an unknown operation or already complete.

Message ID

The message ID, in this case **msgid=2**, is the LDAP operation identifier, as generated by the LDAP SDK client. The message ID may have a different value than the operation number but identifies the same operation. The message ID is used with an **ABANDON** operation and tells the user which client operation is being abandoned.

```
[21/Apr/2020:11:39:52 -0700] conn=12 op=2 ABANDON targetop=NOTFOUND msgid=2
```



NOTE

The Directory Server operation number starts counting at 0, and, in the majority of LDAP SDK/client implementations, the message ID number starts counting at 1, which explains why the message ID is frequently equal to the Directory Server operation number plus 1.

SASL Multi-Stage Bind Logging

In Directory Server, logging for multi-stage binds is explicit. Each stage in the bind process is logged. The error codes for these SASL connections are really return codes. In [Example 7.1, "Example Access Log"](#), the SASL bind is currently in progress so it has a return code of **err=14**, meaning the connection is still open, and there is a corresponding progress statement, **SASL bind in progress**.

```
[21/Apr/2020:11:39:55 -0700] conn=14 op=0 BIND dn="" method=sasl version=3 mech=DIGEST-MD5
```

```
[21/Apr/2020:11:39:55 -0700] conn=14 op=0 RESULT err=14 tag=97 nentries=0 etime=0, SASL bind in progress
```

In logging a SASL bind, the **sasl** method is followed by the LDAP [Version Number](#) and the SASL mechanism used, as shown below with the GSS-API mechanism.

```
[21/Apr/2020:12:57:14 -0700] conn=32 op=0 BIND dn="" method=sasl version=3 mech=GSSAPI
```



NOTE

The authenticated DN (the DN used for access control decisions) is now logged in the BIND result line as opposed to the bind request line, as was previously the case:

```
[21/Apr/2020:11:39:55 -0700] conn=14 op=1 RESULT err=0 tag=97 nentries=0 etime=0 dn="uid=jdoe,dc=example,dc=com"
```

For SASL binds, the DN value displayed in the bind request line is not used by the server and, as a consequence, is not relevant. However, given that the authenticated DN is the DN which, for SASL binds, must be used for audit purposes, it is essential that this be clearly logged. Having this authenticated DN logged in the bind result line avoids any confusion as to which DN is which.

7.1.3. Access Log Content for Additional Access Logging Levels

This section presents the additional access logging levels available in the Directory Server access log.

In [Example 7.2, "Access Log Extract with Internal Access Operations Level \(Level 4\)"](#), access logging level **4**, which logs internal operations, is enabled.

Example 7.2. Access Log Extract with Internal Access Operations Level (Level 4)

```
[12/Jul/2020:16:45:46 +0200] conn=Internal op=-1 SRCH
base="cn=\22dc=example,dc=com\22,cn=mapping tree,cn=config"scope=0
filter="objectclass=nsMappingTree"attrs="nsslapd-referral" options=persistent
[12/Jul/2020:16:45:46 +0200] conn=Internal op=-1 RESULT err=0 tag=48 nentries=1etime=0
[12/Jul/2020:16:45:46 +0200] conn=Internal op=-1 SRCH
base="cn=\22dc=example,dc=com\22,cn=mapping tree,cn=config"scope=0
filter="objectclass=nsMappingTree" attrs="nsslapd-state"
[12/Jul/2020:16:45:46 +0200] conn=Internal op=-1 RESULT err=0 tag=48 nentries=1etime=0
```

Access log level **4** enables logging for internal operations, which log search base, scope, filter, and requested search attributes, in addition to the details of the search being performed.

In the following example, access logging level **768** is enabled (512 + 256), which logs access to entries and referrals. In this extract, six entries and one referral are returned in response to the search request, which is shown on the first line.

```
[12/Jul/2020:16:43:02 +0200] conn=306 fd=60 slot=60 connection from 127.0.0.1 to 127.0.0.1
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 SRCH base="dc=example,dc=com" scope=2 filter="(
description=*)" attrs=ALL
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 ENTRY dn="ou=Special
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 ENTRY dn="cn=Accounting
Managers,ou=groups,dc=example,dc=com"
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 ENTRY dn="cn=HR
Managers,ou=groups,dc=example,dc=com"
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 ENTRY dn="cn=QA
Managers,ou=groups,dc=example,dc=com"
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 ENTRY dn="cn=PD
Managers,ou=groups,dc=example,dc=com"
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 ENTRY dn="ou=Red Hat
Servers,dc=example,dc=com"
[12/Jul/2020:16:43:02 +0200] conn=306 op=0 REFERRAL
```

Connection Description

The connection description, in this case **conn=Internal**, indicates that the connection is an internal connection. The operation number **op=-1** also indicates that the operation was initiated internally.

```
[12/Jul/2020:16:45:46 +0200] conn=Internal op=-1 ENTRY
dn="cn=\22dc=example,dc=com\22,cn=mapping tree,cn=config"
```

Options Description

The options description (**options=persistent**) indicates that a persistent search is being performed, as distinguished from a regular search operation. Persistent searches can be used as a form of monitoring and configured to return changes to given configurations as changes occur.

Both log levels **512** and **4** are enabled for this example, so both internal access operations and entry access and referrals being logged.

```
[12/Jul/2020:16:45:46 +0200] conn=Internal op=-1 SRCH
base="cn=\22dc=example,dc=com\22,cn=mapping tree,cn=config"scope=0
filter="objectclass=nsMappingTree"attrs="nsslapd-referral" options=persistent
```

7.1.4. Common Connection Codes

A connection code is a code that is added to the **closed** log message to provide additional information related to the connection closure.

Table 7.3. Common Connection Codes

Connection Code	Description
A1	Client aborts the connection.
B1	Corrupt BER tag encountered. If BER tags, which encapsulate data being sent over the wire, are corrupt when they are received, a B1 connection code is logged to the access log. BER tags can be corrupted due to physical layer network problems or bad LDAP client operations, such as an LDAP client aborting before receiving all request results.
B2	BER tag is longer than the nsslapd-maxbersize attribute value. For further information about this configuration attribute, see Section 3.1.116, "nsslapd-maxbersize (Maximum Message Size)".
B3	Corrupt BER tag encountered.
B4	Server failed to flush data response back to client.
P2	Closed or corrupt connection has been detected.
T1	Client does not receive a result within the specified idletimeout period. For further information about this configuration attribute, see Section 3.1.95, "nsslapd-idletimeout (Default Idle Timeout)".
T2	Server closed connection after ioblocktimeout period was exceeded. For further information about this configuration attribute, see Section 3.1.98, "nsslapd-ioblocktimeout (IO Block Time Out)".
U1	Connection closed by server after client sends an unbind request. The server will always close the connection when it sees an unbind request.

7.2. ERROR LOG REFERENCE

The Directory Server error log records messages for Directory Server transactions and operations. These may be error messages for failed operations, but it also contains general information about the processes of Directory Server and LDAP tasks, such as server startup messages, logins and searches of the directory, and connection information.

7.2.1. Error Log Logging Levels

The error log can record different amounts of *detail* for operations, as well as different *kinds* of information depending on the type of error logging enabled.

The logging level is set in the [Section 3.1.1.78, “nsslapd-errorlog-level \(Error Log Level\)”](#) configuration attribute. The default log level is **16384**, which included critical error messages and standard logged messages, like LDAP results codes and startup messages. As with access logging, error logging levels are additive. To enable both replication logging (**8192**) and plug-in logging (**65536**), set the log level to **73728** (**8192 + 65536**).



NOTE

Enabling high levels of debug logging can significantly erode server performance. Debug log levels, such as replication (**8192**) should only be enabled for troubleshooting, not for daily operations.

Table 7.4. Error Log Levels

Setting	Console Name	Description
1	Trace function calls	Logs a message when the server enters and exits a function.
2	Packeting handlings	Logs debug information for packets processed by the server.
4	Heavy trace output	Logs when the server enters and exits a function, with additional debugging messages.
8	Connection management	Logs the current connection status, including the connection methods used for a SASL bind.
16	Packets sent/received	Print out the numbers of packets sent and received by the server.
32	Search filter processing	Logs all of the functions called by a search operation.
64	Config file processing	Prints any .conf configuration files used with the server, line by line, when the server is started. By default, only slapd-collations.conf is available and processed.

Setting	Console Name	Description
128	Access control list processing	
2048	Log entry parsing.	Logs schema parsing debugging information.
4096	Housekeeping	Housekeeping thread debugging.
8192	Replication	Logs detailed information about every replication-related operation, including updates and errors, which is important for debugging replication problems.
16384	Default	Default level of logging used for critical errors and other messages that are always written to the error log, such as server startup messages. Messages at this level are always included in the error log, regardless of the log level setting.
32768	Entry cache	Database entry cache debugging.
65536	Plug-ins	Writes an entry to the log file when a server plug-in calls slapi-log-error , so this is used for server plug-in debugging.
262144	Access control summary	Summarizes information about access to the server, much less verbose than level 128 . This value is recommended for use when a summary of access control processing is needed. Use 128 for very detailed processing messages.

7.2.2. Error Log Content

The format of the error log differs compared from that of the access log:

Log entries written by the server

Entries that the server writes to the file, use the following format:

time_stamp - severity_level - function_name - message

For example:

```
[24/Mar/2017:11:31:38.781466443 +0100] - ERR - no_diskspace - No enough space left on device (/var/lib/dirsrv/slapd-instance_name/db) (40009728 bytes); at least 145819238 bytes space is needed for db region files
```

Log entries written by plug-ins

Entries that plug-ins write to the file, use the following format:

```
time_stamp - severity_level - plug-in_name - function_name - message
```

For example:

```
[24/Mar/2017:11:42:17.628363848 +0100] - ERR - NSMMReplicationPlugin - multimaster_extop_StartNSDS50ReplicationRequest - conn=19 op=3 repl="o=example.com": Excessive clock skew from supplier RUV
```

Error log entries contain the following columns:

- Time stamp: The format can differ depending on your local settings. If high-resolution time stamps are enabled in the **nsslapd-logging-hr-timestamps-enabled** attribute in the **cn=config** entry (default), the time stamp is exact to the nanosecond.
- Severity level: The following severity levels are used:
 - **EMERG**: This level is logged when the server fails to start.
 - **ALERT**: The server is in a critical state and possible action must be taken.
 - **CRIT**: Severe error.
 - **ERR**: General error.
 - **WARNING**: A warning message, that is not necessarily an error.
 - **NOTICE**: A normal, but significant condition occurred. For example, this is logged for expected behavior.
 - **INFO**: Informational messages, such as startup, shutdown, import, export, backup, restore.
 - **DEBUG**: Debug-level messages. This level is also used by default when using a verbose logging level, such as **Trace function calls** (1), **Access control list processing** (128), and **Replication** (8192). For a list of error log levels, see [Table 7.4, "Error Log Levels"](#).

You can use the severity levels to filter your log entries. For example, to display only log entries using the **ERR** severity:

```
# grep ERR /var/log/dirsrv/slapd-instance_name/errors
[24/Mar/2017:11:31:38.781466443 +0100] - ERR - no_diskspace - No enough space left on device (/var/lib/dirsrv/slapd-instance_name/db) (40009728 bytes); at least 145819238 bytes space is needed for db region files
[24/Mar/2017:11:31:38.815623298 +0100] - ERR - ldbm_back_start - Failed to init database, err=28 No space left on device
[24/Mar/2017:11:31:38.828591835 +0100] - ERR - plugin_dependency_startall - Failed to start database plugin ldbm database
...
```

- Plug-in name: If a plug-in logged the entry, this column displays the name of the plug-in. If the server logged the entry, this column does not appear.
- Function name: Functions that the operation or the plug-in called.
- Message: The output that the operation or plug-in returned. This message contains additional information, such as LDAP error codes and connection information.

7.2.3. Error Log Content for Other Log Levels

The different log levels return not only different levels of detail, but also information about different types of server operations. Some of these are summarized here, but there are many more combinations of logging levels possible.

Replication logging is one of the most important diagnostic levels to implement. This logging level records all operations related to replication and Windows synchronization, including processing modifications on a supplier and writing them to the changelog, sending updates, and changing replication agreements.

Whenever a replication update is prepared or sent, the error log identifies the replication or synchronization agreement being specified, the consumer host and port, and the current replication task.

`[timestamp] NSMMReplicationPlugin - agmt="name" (consumer_host:consumer_port): current_task`

For example:

`[09/Jan/2020:13:44:48 -0500] NSMMReplicationPlugin - agmt="cn=example2" (alt:13864): {replicageneration} 4949df6e000000010000`

{replicageneration} means that the new information is being sent, and **4949df6e000000010000** is the change sequence number of the entry being replicated.

[Example 7.3, “Replication Error Log Entry”](#) shows the complete process of sending a single entry to a consumer, from adding the entry to the changelog to releasing the consumer after replication is complete.

Example 7.3. Replication Error Log Entry

```
[29/May/2017:14:15:30.539817639 +0200] - DEBUG - _csngen_adjust_local_time - gen state before 592c103d0000:1496059964:0:1
[29/May/2017:14:15:30.562983285 +0200] - DEBUG - _csngen_adjust_local_time - gen state after 592c10e20000:1496060129:0:1
[29/May/2017:14:15:30.578828393 +0200] - DEBUG - NSMMReplicationPlugin - rvu_add_csn_inprogress - Successfully inserted csn 592c10e2000000020000 into pending list
[29/May/2017:14:15:30.589917123 +0200] - DEBUG - NSMMReplicationPlugin - changelog program - _cl5GetDBFileByReplicaName - found DB object 0x558ddfe1f720 for database /var/lib/dirsrv/slapd-supplier_2/changelogdb/d3de3e8d-446611e7-a89886da-6a37442d_592c0e0b000000010000.db
[29/May/2017:14:15:30.600044236 +0200] - DEBUG - NSMMReplicationPlugin - changelog program - cl5WriteOperationTxn - Successfully written entry with csn (592c10e2000000020000)
[29/May/2017:14:15:30.615923352 +0200] - DEBUG - NSMMReplicationPlugin - changelog program - _cl5GetDBFileByReplicaName - found DB object 0x558ddfe1f720 for database /var/lib/dirsrv/slapd-supplier_2/changelogdb/d3de3e8d-446611e7-a89886da-6a37442d_592c0e0b000000010000.db
```

```
[29/May/2017:14:15:30.627443305 +0200] - DEBUG - NSMMReplicationPlugin -  
csnplCommitALL: committing all csns for csn 592c10e2000000020000  
[29/May/2017:14:15:30.632713657 +0200] - DEBUG - NSMMReplicationPlugin -  
csnplCommitALL: processing data csn 592c10e2000000020000  
[29/May/2017:14:15:30.652621188 +0200] - DEBUG - NSMMReplicationPlugin - ruv_update_ruv -  
Successfully committed csn 592c10e2000000020000  
[29/May/2017:14:15:30.669666453 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_run -  
agmt="cn=meTo_localhost:39001" (localhost:39001): State: wait_for_changes ->  
wait_for_changes  
[29/May/2017:14:15:30.685259483 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_run -  
agmt="cn=meTo_localhost:39001" (localhost:39001): State: wait_for_changes ->  
ready_to_acquire_replica  
[29/May/2017:14:15:30.689906327 +0200] - DEBUG - NSMMReplicationPlugin - conn_connect -  
agmt="cn=meTo_localhost:39001" (localhost:39001) - Trying non-secure slapi_ldap_init_ext  
[29/May/2017:14:15:30.700259799 +0200] - DEBUG - NSMMReplicationPlugin - conn_connect -  
agmt="cn=meTo_localhost:39001" (localhost:39001) - binddn = cn=replrep, cn=config, passwd =  
{AES-  
TUhNR0NTcUdTSWIzRFFFRkRUQm1NRVHQ1NxR1NJYjNEUVGRERBNEJDUmIZVFUzTnp  
RMk55MDBaR1ZtTXpobQ0KTWkxaE9XTTRPREpoTIMwME1EaGpabVUxWmdBQ0FRSUNBU0F  
3Q2dZSUtvWklodmNOQWdj0hRWUpZSVpJQVdVRA0KQkFFcUJCRGhwMnNLcEZ2ZWE2RzEw  
WG10OU41Tg==}+36owal7oTmvWhxRzUqX5w==  
[29/May/2017:14:15:30.712287531 +0200] - DEBUG - NSMMReplicationPlugin -  
conn_cancel_linger - agmt="cn=meTo_localhost:39001" (localhost:39001) - No linger to cancel on  
the connection  
[29/May/2017:14:15:30.736779494 +0200] - DEBUG - _csngen_adjust_local_time - gen state  
before 592c10e20001:1496060129:0:1  
[29/May/2017:14:15:30.741909244 +0200] - DEBUG - _csngen_adjust_local_time - gen state  
after 592c10e30000:1496060130:0:1  
[29/May/2017:14:15:30.880287041 +0200] - DEBUG - NSMMReplicationPlugin - acquire_replica -  
agmt="cn=meTo_localhost:39001" (localhost:39001): Replica was successfully acquired.  
[29/May/2017:14:15:30.897500049 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_run -  
agmt="cn=meTo_localhost:39001" (localhost:39001): State: ready_to_acquire_replica ->  
sending_updates  
[29/May/2017:14:15:30.914417773 +0200] - DEBUG - csngen_adjust_time - gen state before  
592c10e30001:1496060130:0:1  
[29/May/2017:14:15:30.926341721 +0200] - DEBUG - NSMMReplicationPlugin - changelog  
program - _cl5GetDBFile - found DB object 0x558ddfe1f720 for database /var/lib/dirsrv/slapd-  
supplier_2/changelogdb/d3de3e8d-446611e7-a89886da-6a37442d_592c0e0b000000010000.db  
[29/May/2017:14:15:30.943094471 +0200] - DEBUG - NSMMReplicationPlugin - changelog  
program - _cl5PositionCursorForReplay - (agmt="cn=meTo_localhost:39001" (localhost:39001)):  
Consumer RUV:  
[29/May/2017:14:15:30.949395331 +0200] - DEBUG - NSMMReplicationPlugin -  
agmt="cn=meTo_localhost:39001" (localhost:39001): {replicageneration}  
592c0e0b000000010000  
[29/May/2017:14:15:30.961118175 +0200] - DEBUG - NSMMReplicationPlugin -  
agmt="cn=meTo_localhost:39001" (localhost:39001): {replica 1 ldap://localhost:39001}  
592c0e17000000010000 592c0e1a000100010000 00000000  
[29/May/2017:14:15:30.976680025 +0200] - DEBUG - NSMMReplicationPlugin -  
agmt="cn=meTo_localhost:39001" (localhost:39001): {replica 2 ldap://localhost:39002}  
592c103c000000020000 592c103c000000020000 00000000  
[29/May/2017:14:15:30.990404183 +0200] - DEBUG - NSMMReplicationPlugin - changelog  
program - _cl5PositionCursorForReplay - (agmt="cn=meTo_localhost:39001" (localhost:39001)):  
Supplier RUV:  
[29/May/2017:14:15:31.001242624 +0200] - DEBUG - NSMMReplicationPlugin -  
agmt="cn=meTo_localhost:39001" (localhost:39001): {replicageneration}  
592c0e0b000000010000
```

[29/May/2017:14:15:31.017406105 +0200] - DEBUG - NSMMReplicationPlugin - agmt="cn=meTo_localhost:39001" (localhost:39001): {replica 2 ldap://localhost:39002} 592c103c000000020000 592c10e2000000020000 592c10e1 [29/May/2017:14:15:31.028803190 +0200] - DEBUG - NSMMReplicationPlugin - agmt="cn=meTo_localhost:39001" (localhost:39001): {replica 1 ldap://localhost:39001} 592c0e1a000100010000 592c0e1a000100010000 00000000 [29/May/2017:14:15:31.040172464 +0200] - DEBUG - agmt="cn=meTo_localhost:39001" (localhost:39001) - clcache_get_buffer - found thread private buffer cache 0x558ddf870f00 [29/May/2017:14:15:31.057495165 +0200] - DEBUG - agmt="cn=meTo_localhost:39001" (localhost:39001) - clcache_get_buffer - _pool is 0x558ddfe294d0 _pool->pl_busy_lists is 0x558ddfab84c0 _pool->pl_busy_lists->bl_buffers is 0x558ddf870f00 [29/May/2017:14:15:31.063015498 +0200] - DEBUG - agmt="cn=meTo_localhost:39001" (localhost:39001) - clcache_initial_anchorcsn - agmt="cn=meTo_localhost:39001" (localhost:39001) - (cscb 0 - state 0) - csnPrevMax () csnMax (592c10e2000000020000) csnBuf (592c103c000000020000) csnConsumerMax (592c103c000000020000) [29/May/2017:14:15:31.073252305 +0200] - DEBUG - clcache_initial_anchorcsn - anchor is now: 592c103c000000020000 [29/May/2017:14:15:31.089915209 +0200] - DEBUG - NSMMReplicationPlugin - changelog program - agmt="cn=meTo_localhost:39001" (localhost:39001): CSN 592c103c000000020000 found, position set for replay [29/May/2017:14:15:31.095825439 +0200] - DEBUG - agmt="cn=meTo_localhost:39001" (localhost:39001) - clcache_get_next_change - load=1 rec=1 csn=592c10e2000000020000 [29/May/2017:14:15:31.100123762 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_result_threadmain - Starting [29/May/2017:14:15:31.115749709 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_result_threadmain - Read result for message_id 0 [29/May/2017:14:15:31.125866330 +0200] - DEBUG - NSMMReplicationPlugin - replay_update - agmt="cn=meTo_localhost:39001" (localhost:39001): Sending add operation (dn="cn=user,ou=People,dc=example,dc=com" csn=592c10e2000000020000) [29/May/2017:14:15:31.142339398 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_result_threadmain - Read result for message_id 0 [29/May/2017:14:15:31.160456597 +0200] - DEBUG - NSMMReplicationPlugin - replay_update - agmt="cn=meTo_localhost:39001" (localhost:39001): Consumer successfully sent operation with csn 592c10e2000000020000 [29/May/2017:14:15:31.172399536 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_result_threadmain - Read result for message_id 0 [29/May/2017:14:15:31.188857336 +0200] - DEBUG - agmt="cn=meTo_localhost:39001" (localhost:39001) - clcache_adjust_anchorcsn - agmt="cn=meTo_localhost:39001" (localhost:39001) - (cscb 0 - state 1) - csnPrevMax (592c10e2000000020000) csnMax (592c10e2000000020000) csnBuf (592c10e2000000020000) csnConsumerMax (592c10e2000000020000) [29/May/2017:14:15:31.199605024 +0200] - DEBUG - agmt="cn=meTo_localhost:39001" (localhost:39001) - clcache_load_buffer - rc=-30988 [29/May/2017:14:15:31.210800816 +0200] - DEBUG - NSMMReplicationPlugin - send_updates - agmt="cn=meTo_localhost:39001" (localhost:39001): No more updates to send (cl5GetNextOperationToReplay) [29/May/2017:14:15:31.236214134 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_waitfor_async_results - 0 5 [29/May/2017:14:15:31.246755544 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_result_threadmain - Read result for message_id 0 [29/May/2017:14:15:31.277705986 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_result_threadmain - Read result for message_id 0 [29/May/2017:14:15:31.303530336 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_result_threadmain - Read result for message_id 5 [29/May/2017:14:15:31.318259308 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_result_threadmain - Result 1, 0, 0, 5, (null)

```
[29/May/2017:14:15:31.335263462 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_result_threadmain - Read result for message_id 5
[29/May/2017:14:15:31.364551307 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_waitfor_async_results - 5 5
[29/May/2017:14:15:31.376301820 +0200] - DEBUG - NSMMReplicationPlugin -
repl5_inc_result_threadmain exiting
[29/May/2017:14:15:31.393707037 +0200] - DEBUG - agmt="cn=meTo_localhost:39001"
(localhost:39001) - clcache_return_buffer - session end: state=5 load=1 sent=1 skipped=0
skipped_new_rid=0 skipped_csn_gt_cons_maxcsn=0 skipped_up_to_date=0
skipped_csn_gt_ruv=0 skipped_csn_covered=0
[29/May/2017:14:15:31.398134114 +0200] - DEBUG - NSMMReplicationPlugin -
consumer_connection_extension_acquire_exclusive_access - conn=4 op=3 Acquired consumer
connection extension
[29/May/2017:14:15:31.423099625 +0200] - DEBUG - NSMMReplicationPlugin -
multimaster_extop_StartNSDS50ReplicationRequest - conn=4 op=3 repl="dc=example,dc=com":
Begin incremental protocol
[29/May/2017:14:15:31.438899389 +0200] - DEBUG - csngen_adjust_time - gen state before
592c10e30001:1496060130:0:1
[29/May/2017:14:15:31.443800884 +0200] - DEBUG - csngen_adjust_time - gen state after
592c10e40001:1496060130:1:1
[29/May/2017:14:15:31.454123488 +0200] - DEBUG - NSMMReplicationPlugin -
replica_get_exclusive_access - conn=4 op=3 repl="dc=example,dc=com": Acquired replica
[29/May/2017:14:15:31.469698781 +0200] - DEBUG - NSMMReplicationPlugin - release_replica -
agmt="cn=meTo_localhost:39001" (localhost:39001): Successfully released consumer
[29/May/2017:14:15:31.475096195 +0200] - DEBUG - NSMMReplicationPlugin -
conn_start_linger -agmt="cn=meTo_localhost:39001" (localhost:39001) - Beginning linger on the
connection
[29/May/2017:14:15:31.485281588 +0200] - DEBUG - NSMMReplicationPlugin - repl5_inc_run -
agmt="cn=meTo_localhost:39001" (localhost:39001): State: sending_updates ->
wait_for_changes
[29/May/2017:14:15:31.495865065 +0200] - DEBUG - NSMMReplicationPlugin -
multimaster_extop_StartNSDS50ReplicationRequest - conn=4 op=3 repl="dc=example,dc=com":
StartNSDS90ReplicationRequest: response=0 rc=0
[29/May/2017:14:15:31.501617765 +0200] - DEBUG - NSMMReplicationPlugin -
consumer_connection_extension_relinquish_exclusive_access - conn=4 op=3 Relinquishing
consumer connection extension
[29/May/2017:14:15:31.716627741 +0200] - DEBUG - NSMMReplicationPlugin -
consumer_connection_extension_acquire_exclusive_access - conn=4 op=4 Acquired consumer
connection extension
[29/May/2017:14:15:31.735431913 +0200] - DEBUG - NSMMReplicationPlugin -
replica_relinquish_exclusive_access - conn=4 op=4 repl="dc=example,dc=com": Released replica
held by locking_purl=conn=4 id=3
[29/May/2017:14:15:31.745841821 +0200] - DEBUG - NSMMReplicationPlugin -
consumer_connection_extension_relinquish_exclusive_access - conn=4 op=4 Relinquishing
consumer connection extension
```

Plug-in logging records every the name of the plug-in and all of the functions called by the plug-in. This has a simple format:

```
[timestamp] Plugin_name - message
[timestamp] - function - message
```

The information returned can be hundreds of lines long as every step is processed. The precise information recorded depends on the plug-in itself. For example, the ACL Plug-in includes a connection

and operation number, as shown in [Example 7.4, “Example ACL Plug-in Error Log Entry with Plug-in Logging”](#).

Example 7.4. Example ACL Plug-in Error Log Entry with Plug-in Logging

```
[29/May/2017:14:38:19.133878244 +0200] - DEBUG - get_filter_internal - ==>
[29/May/2017:14:38:19.153942547 +0200] - DEBUG - get_filter_internal - PRESENT
[29/May/2017:14:38:19.177908064 +0200] - DEBUG - get_filter_internal - <= 0
[29/May/2017:14:38:19.193547449 +0200] - DEBUG - slapi_vattr_filter_test_ext_internal - =>
[29/May/2017:14:38:19.198121765 +0200] - DEBUG - slapi_vattr_filter_test_ext_internal - <=
[29/May/2017:14:38:19.214342752 +0200] - DEBUG - slapi_vattr_filter_test_ext_internal -
PRESENT
[29/May/2017:14:38:19.219886104 +0200] - DEBUG - NSACLPlugin - acl_access_allowed -
conn=15 op=1 (main): Allow search on entry(cn=replication,cn=config): root user
[29/May/2017:14:38:19.230152526 +0200] - DEBUG - slapi_vattr_filter_test_ext_internal - <= 0
[29/May/2017:14:38:19.240971955 +0200] - DEBUG - NSACLPlugin -
acl_read_access_allowed_on_entry - Root access (read) allowed on
entry(cn=replication,cn=config)
[29/May/2017:14:38:19.246456160 +0200] - DEBUG - cos-plugin - cos_cache_vattr_types -
Failed to get class of service reference
[29/May/2017:14:38:19.257200851 +0200] - DEBUG - NSACLPlugin - Root access (read) allowed
on entry(cn=replication,cn=config)
[29/May/2017:14:38:19.273534025 +0200] - DEBUG - NSACLPlugin - Root access (read) allowed
on entry(cn=replication,cn=config)
[29/May/2017:14:38:19.289474926 +0200] - DEBUG - slapi_filter_free - type 0x87
```



NOTE

[Example 7.4, “Example ACL Plug-in Error Log Entry with Plug-in Logging”](#) shows both plug-in logging and search filter processing (log level 65696).

Many other kinds of logging have similar output to the plug-in logging level, only for different kinds of internal operations. Heavy trace output (**4**), access control list processing (**128**), schema parsing (**2048**), and housekeeping (**4096**) all record the functions called by the different operations being performed. In this case, the difference is not in the format of what is being recorded, but what operations it is being recorded for.

The configuration file processing goes through any **.conf** file, printing every line, whenever the server starts up. This can be used to debug any problems with files outside of the server’s normal configuration. By default, only **slapd-collations.conf** file, which contains configurations for international language sets, is available.

Example 7.5. Config File Processing Log Entry

```
[29/May/2017:15:26:48.897935879 +0200] - DEBUG - collation_read_config - Reading config file
/etc/dirsrv/slapd-supplier_1/slapd-collations.conf
[29/May/2017:15:26:48.902606586 +0200] - DEBUG - collation-plugin - collation_read_config -
line 16: collation "" "" "" 1 3 2.16.840.1.113730.3.3.2.0.1 default
[29/May/2017:15:26:48.918493657 +0200] - DEBUG - collation-plugin - collation_read_config -
line 17: collation ar "" "" 1 3 2.16.840.1.113730.3.3.2.1.1 ar
[29/May/2017:15:26:48.932550086 +0200] - DEBUG - collation-plugin - collation_read_config -
line 18: collation be "" "" 1 3 2.16.840.1.113730.3.3.2.2.1 be be-BY
...
...
```

There are two levels of ACI logging, one for debug information and one for summary. Both of these ACI logging levels records some extra information that is not included with other types of plug-ins or error logging, including [Connection Number](#) and [Operation Number](#) information. Show the name of the plugin, the bind DN of the user, the operation performed or attempted, and the ACI which was applied. The debug level shows the series of functions called in the course of the bind and any other operations, as well.

[Example 7.6, "Access Control Summary Logging"](#) shows the summary access control log entry.

Example 7.6. Access Control Summary Logging

```
[29/May/2017:15:34:52.742034888 +0200] - DEBUG - NSACLPlugin - acllist_init_scan - Failed to
find root for base: cn=features,cn=config
[29/May/2017:15:34:52.761702767 +0200] - DEBUG - NSACLPlugin - acllist_init_scan - Failed to
find root for base: cn=config
[29/May/2017:15:34:52.771907825 +0200] - DEBUG - NSACLPlugin - acl_access_allowed - #####
conn=6 op=1 binddn="cn=user,ou=people,dc=example,dc=com"
[29/May/2017:15:34:52.776327012 +0200] - DEBUG - NSACLPlugin - ***** RESOURCE
INFO STARTS *****
[29/May/2017:15:34:52.786397852 +0200] - DEBUG - NSACLPlugin - Client DN:
cn=user,ou=people,dc=example,dc=com
[29/May/2017:15:34:52.797004451 +0200] - DEBUG - NSACLPlugin - resource
type:256(search target_DN )
[29/May/2017:15:34:52.807135945 +0200] - DEBUG - NSACLPlugin - Slapi_Entry DN:
cn=features,cn=config
[29/May/2017:15:34:52.822877838 +0200] - DEBUG - NSACLPlugin - ATTR: objectClass
[29/May/2017:15:34:52.827250828 +0200] - DEBUG - NSACLPlugin - rights:search
[29/May/2017:15:34:52.831603634 +0200] - DEBUG - NSACLPlugin - ***** RESOURCE
INFO ENDS *****
[29/May/2017:15:34:52.847183276 +0200] - DEBUG - NSACLPlugin - acl__scan_for_acis - Num
of ALLOW Handles:0, DENY handles:0
[29/May/2017:15:34:52.857857195 +0200] - DEBUG - NSACLPlugin -
print_access_control_summary - conn=6 op=1 (main): Deny search on
entry(cn=features,cn=config).attr(objectClass) to cn=user,ou=people,dc=example,dc=com: no aci
matched the resource
```

7.3. AUDIT LOG REFERENCE

The audit log records *changes made* to the server instance. Unlike the error and access log, the audit log does not record access to the server instance, so searches against the database are not logged.

The audit log is formatted differently than the access and error logs and is like a time-stamped LDIF file. The operations recorded in the audit log are formatted as LDIF statements:

```
timestamp: date
dn: modified_entry
changetype: action
action:attribute
attribute:new_value
-
replace: modifiersname
```

```

modifiersname: dn
-
replace: modifytimestamp
modifytimestamp: date
-
```

LDIF files and formats are described in more detail in the "LDAP Data Interchange Format" appendix of the *Administration Guide*.

Several different kinds of audit entries are shown in [Example 7.7, "Audit Log Content"](#).

Example 7.7. Audit Log Content

```

... modifying an entry ...
time: 20200108181429
dn: uid=scarter,ou=people,dc=example,dc=com
changetype: modify
replace: userPassword
userPassword: {SSHA}8EcJhJolgBgY/E5j8JiVoj6W3BLyj9Za/rCPOw==

replace: modifiersname
modifiersname: cn=Directory Manager

replace: modifytimestamp
modifytimestamp: 20200108231429Z
-

... sending a replication update ...
time: 20200109131811
dn: cn=example2,cn=replica,cn="dc=example,dc=com",cn=mapping tree,cn=config
changetype: modify
replace: nsds5BeginReplicaRefresh
nsds5BeginReplicaRefresh: start

replace: modifiersname
modifiersname: cn=Directory Manager

replace: modifytimestamp
modifytimestamp: 20200109181810Z
-
```

Note that you cannot change to format or set a log level for the audit log.

7.4. LDAP RESULT CODES

Directory Server uses the following LDAP result codes:

Table 7.5. LDAP Result Codes

Decimal Values	Hex Values	Constants
0	0x00	LDAP_SUCCESS

Decimal Values	Hex Values	Constants
1	0x01	LDAP_OPERATIONS_ERROR
2	0x02	LDAP_PROTOCOL_ERROR
3	0x03	LDAP_TIMELIMIT_EXCEEDED
4	0x04	LDAP_SIZELIMIT_EXCEEDED
5	0x05	LDAP_COMPARE_FALSE
6	0x06	LDAP_COMPARE_TRUE
7	0x07	LDAP_AUTH_METHOD_NOT_SUPPORTED
LDAP_STRONG_AUTH_NOT_SUPPORTED		
8	0x08	LDAP_STRONG_AUTH_REQUIRED
9	0x09	LDAP_PARTIAL_RESULTS
10	0x0a	LDAP_REFERRAL [a]
11	0x0b	LDAP_ADMINLIMIT_EXCEEDED
12	0x0c	LDAP_UNAVAILABLE_CRITICAL_EXTENSION
13	0x0d	LDAP_CONFIDENTIALITY_REQUIRED
14	0x0e	LDAP_SASL_BIND_IN_PROGRESS
16	0x10	LDAP_NO_SUCH_ATTRIBUTE
17	0x11	LDAP_UNDEFINED_TYPE
18	0x12	LDAP_INAPPROPRIATE_MATCHING
19	0x13	LDAP_CONSTRAINT_VIOLATION

Decimal Values	Hex Values	Constants
20	0x14	LDAP_TYPE_OR_VALUE_EXISTS
21	0x15	LDAP_INVALID_SYNTAX
32	0x20	LDAP_NO_SUCH_OBJECT
33	0x21	LDAP_ALIAS_PROBLEM
34	0x22	LDAP_INVALID_DN_SYNTAX
35	0x23	LDAP_IS_LEAF [b]
36	0x24	LDAP_ALIAS_DEREF_PROBLEM
48	0x30	LDAP_INAPPROPRIATE_AUTH
49	0x31	LDAP_INVALID_CREDENTIALS
50	0x32	LDAP_INSUFFICIENT_ACCESS
51	0x33	LDAP_BUSY
52	0x34	LDAP_UNAVAILABLE
53	0x35	LDAP_UNWILLING_TO_PERFORM
54	0x36	LDAP_LOOP_DETECT
60	0x3c	LDAP_SORT_CONTROL_MISSING
61	0x3d	LDAP_INDEX_RANGE_ERROR
64	0x40	LDAP_NAMING_VIOLATION
65	0x41	LDAP_OBJECT_CLASS_VIOLATION
66	0x42	LDAP_NOT_ALLOWED_ON_NONLEAF
67	0x43	LDAP_NOT_ALLOWED_ON_RDN

Decimal Values	Hex Values	Constants
68	0x44	LDAP_ALREADY_EXISTS
69	0x45	LDAP_NO_OBJECT_CLASS_MO DS
70	0x46	LDAP_RESULTS_TOO_LARGE [c]
71	0x47	LDAP_AFFECTS_MULTIPLE_DS AS
76	0x4C	LDAP_VIRTUAL_LIST_VIEW_ERR OR
80	0x50	LDAP_OTHER
81	0x51	LDAP_SERVER_DOWN
82	0x52	LDAP_LOCAL_ERROR
83	0x53	LDAP_ENCODING_ERROR
84	0x54	LDAP_DECODING_ERROR
85	0x55	LDAP_TIMEOUT
86	0x56	LDAP_AUTH_UNKNOWN
87	0x57	LDAP_FILTER_ERROR
88	0x58	LDAP_USER_CANCELLED
89	0x59	LDAP_PARAM_ERROR
90	0x5A	LDAP_NO_MEMORY
91	0x5B	LDAP_CONNECT_ERROR
92	0x5C	LDAP_NOT_SUPPORTED
93	0x5D	LDAP_CONTROL_NOT_FOUND
94	0x5E	LDAP_MORE_RESULTS_TO_RET URN

Decimal Values	Hex Values	Constants
95	0x5F	LDAP_MORE_RESULTS_TO_RETURRN
96	0x60	LDAP_CLIENT_LOOP
97	0x61	LDAP_REFERRAL_LIMIT_EXCEEDED
118	0x76	LDAP_CANCELLED

[a] LDAPv3
 [b] Not used in LDAPv3
 [c] Reserved for CLDAP

7.5. REPLACING LOG FILES WITH A NAMED PIPE

Many administrators want to do some special configuration or operation with logging data, like configuring an access log to record only certain events. This is not possible using the standard Directory Server log file configuration attributes, but it is possible by sending the log data to a named pipe, and then using another script to process the data. Using a named pipe for the log simplifies these special tasks, like:

- Logging certain events, like failed bind attempts or connections from specific users or IP addresses
- Logging entries which match a specific regular expression pattern
- Keeping the log to a certain length (logging only the last number of lines)
- Sending a notification, such as an email, when an event occurs

Replacing a log file with a pipe improves performance, especially on servers with a high rate of operations.

The named pipe is different than using a script to extract data from the logs because of how data are handled in the log buffer.

If a log is buffered, server performance is good, but important data are not written to disk (the log file) as soon as the event occurs. If the server is having a problem with crashing, it may crash before the data is written to disk – and there is no data for the script to extract.

If a log is not buffered^[1], the writes are flushed to disk with each operation, causing a lot of disk I/O and performance degradation.

Replacing the log disk file with a pipe has the benefits of buffering, since the script that reads from the pipe can buffer the incoming log data in memory (which is not possible with a simple script).

The usage and option details for the script is covered in [Section 9.4, “ds-logpipe.py”](#). The basic format is: **ds-logpipe.py**/*path/to/named_pipe*--**user***pipe_user*--**maxlines***number*--**serverpid***file.pid*--**serverpid***PID*--**servertimeout***seconds*--**plugin**=/*path/to/plugin.py***pluginfile.arg**=*value*

7.5.1. Using the Named Pipe for Logging

The Directory Server instance can use a named pipe for its logging simply by running the named pipe log script and giving the name of the pipe. (If the server is already running, then the log has to be reopened, but there is no configuration required otherwise.)

```
# ds-logpipe.py /var/log/dirsrv/slapd-example/access
```

Running the **ds-logpipe.py** in this way has the advantage of being simple to implement and not requiring any Directory Server configuration changes. This is useful for fast debugging or monitoring, especially if you are looking for a specific type of event.

If the Directory Server instance will frequently or permanently use the named pipe rather than a real file for logging, then it is possible to reconfigure the instance to create the named pipe and use it for logging (as it does by default for the log files).

Three things need to be configured for the log configuration for the instance:

- The log file to use has to be changed to the pipe (**nsslapd-*log**, where the * can be access, error, or audit^[2], depending on the log type being configured)
- Buffering should be disabled because the script already buffers the log entries (**nsslapd-*log-logbuffering**)
- Log rotation should be disabled so that the server does not attempt to rotate the named pipe (**nsslapd-*log-maxlogsperdir**, **nsslapd-*log-logexpirationtime**, and **nsslapd-*log-logrotationtime**)

These configuration changes can be made in the Directory Server Console or using **ldapmodify**.

For example, this switches the access log to **access.pipe**:

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x

dn: cn=config
changetype: modify
replace: nsslapd-accesslog
nsslapd-accesslog: /var/log/dirsrv/slapd-instance/access.pipe

-
replace: nsslapd-accesslog-logbuffering
nsslapd-accesslog-logbuffering: off

-
replace: nsslapd-accesslog-maxlogsperdir
nsslapd-accesslog-maxlogsperdir: 1

-
replace: nsslapd-accesslog-logexpirationtime
nsslapd-accesslog-logexpirationtime: -1

-
replace: nsslapd-accesslog-logrotationtime
nsslapd-accesslog-logrotationtime: -1
```

**NOTE**

Making these changes causes the server to close the current log file and switch to the named pipe immediately. This can be very helpful for debugging a running server and sifting the log output for specific messages.

7.5.2. Starting the Named Pipe with the Server

The named pipe can be started and shut down along with the Directory Server instance by editing the instance's init script configuration file.

**NOTE**

The named pipe script has to be specifically configured in the instance's **dse.ldif** file before it can be called at server startup.

1. Open the instance configuration file for the server system.

/etc/sysconfig/dirsrv-*instance_name*

**WARNING**

Do not edit the **/etc/sysconfig/dirsrv** file.

2. At the end of the file, there will be a line that reads:

Put custom instance specific settings below here.

Below that line, insert the **ds-logpipe.py** command to launch when the server starts. For example:

```
# only keep the last 1000 lines of the error log
python /usr/bin/ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe -m 1000 -u dirsrv -s
/var/run/dirsrv/slapd-example.pid > /var/log/dirsrv/slapd-example/errors &

# only log failed binds
python /usr/bin/ds-logpipe.py /var/log/dirsrv/slapd-example/access.pipe -u dirsrv -s
/var/run/dirsrv/slapd-example.pid --plugin=/usr/share/dirsrv/data/failedbinds.py
failedbinds.logfile=/var/log/dirsrv/slapd-example/access.failedbinds &
```

**NOTE**

The **-s** option both specifies the .pid file for the server to write its PID to and sets the script to start and stop with the server process.

7.5.3. Using Plug-ins with the Named Pipe Log

A plug-in can be called to read the log data from the named pipe and perform some operation on it. There are some considerations with using plug-ins with the named pipe log script:

- The plug-in function is called for every line read from the named pipe.
- The plug-in function must be a Python script and must end in **.py**.
- Any plug-in arguments are passed in the command line to the named pipe log script.
- A pre-operation function can be specified for when the plug-in is loaded.
- A post-operation function can be called for when the script exits.

7.5.3.1. Loading Plug-ins with the Named Pipe Log Script

There are two options with **ds-logpipe.py** to use for plug-ins:

- The **--plugin** option gives the path to the plug-in file (which must be a Python script and must end in **.py**).
- The **plugin.arg** option passes plug-in arguments to the named pipe log script. The plug-in file name (without the **.py** extension) is *plugin* and any argument allowed in that plug-in can be *arg*.

For example:

```
ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe --plugin=/usr/share/dirsrv/data/example-funct.py example-funct.regex="warning" > warnings.txt
```

If there are more than one values passed for the same argument, then they are converted into a list of values in the plug-in dict. For example, this script gives two values for **arg1**:

```
--plugin=/path/to/pluginname.py pluginname.arg1=foo pluginname.arg1=bar pluginname.arg2=baz
```

In the plug-in, this is converted to:

```
{'arg1': ['foo', 'bar'],
 'arg2': 'baz'}
```

This is a Python **dict** object with two keys. The first key is the string **arg1**, and its value is a Python list object with two elements, the strings **foo** and **bar**. The second key is the string **arg2**, and its value is the string **baz**. If an argument has only a single value, it is left as a simple string. Multiple values for a single argument name are converted into a list of strings.

7.5.3.2. Writing Plug-ins to Use with the Named Pipe Log Script

The **ds-logpipe.py** command expects up to three functions in any plug-in: **plugin ()**, **pre ()**, and **post ()**.

Any plug-in used with the **ds-logpipe.py** command must specify the **plugin** function.

The **plugin ()** function is performed against every line in the log data, while the **pre ()** and **post ()** functions are run when the script is started and stopped, respectively.

Each function can have any arguments defined for it, and these arguments can then be passed to the script using the *plugin.arg* option. Additionally, each function can have its own return values and actions defined for it.

Example 7.8. Simple Named Pipe Log Plug-in

```
def pre(myargs):
    retval = True
    myarg = myargs['argname']
    if isinstance(myarg, list): # handle list of values
    else: # handle single value
        if bad_problem:
            retval = False
    return retval

def plugin(line):
    retval = True
    # do something with line
    if something_is_bogus:
        retval = False
    return retval

def post(): # no arguments
    # do something
    # no return value
```

[1] Server performance suffers when log buffering is disabled on the access log, when the log level is changed on the error log, or with audit logging.

[2] The audit log is not enabled by default, so this log has to be enabled before a named pipe can be used to replace it.

CHAPTER 8. CONFIGURATION FILE REFERENCE

Most Directory Server feature you configure are in the **cn=config** entry in the directory. However, for certain features, Directory Server reads settings from configuration files. This chapter describe these files and their settings.

8.1. CERTMAP.CONF

If you set up certificate-based authentication, the **/etc/dirsrv/slappd-*instance_name*/certmap.conf** file manages how Directory Server dynamically maps a certificate to a user entry.

The **/etc/dirsrv/slappd-*instance_name*/certmap.conf** file uses the following format:

```
certmap alias_name      certificate_issuer_DN
alias_name:parameter_name  value
```

You can specify individual settings for different certificate issuer Distinguished Names (DN). For issuer DNs that do not have a separate configuration, the settings from the **default** entry will be used. The following is the required minimum configuration for the **default** entry:

```
certmap default  default
```

Additionally, you can set all available parameters for the **default** entry. Directory Server will use them if they are not specified in individual configurations for issuer DNs.

Example 8.1. Configuration for the**default** Entry and a Specific Issuer DN

The following configuration sets individual settings for certificates having the **o=Example Inc.,c=US** issuer DN set. Other certificates will use the settings from the **default** entry.

```
certmap default  default
default:DNC parms  dc
default:FilterComps  mail, cn
default:VerifyCert  on

certmap example  o=Example Inc.,c=US
example:DNC parms
```

You can set the following parameters:

DNC parms

The **DNC parms** parameter determines how Directory Server generates the base DN used to search for a user in the directory:

- If attributes in the **subject** field of the certificate match the base DN, set the **DNC parms** parameter to these attributes. Separate multiple attribute with commas. However, the order of the attributes in the **DNC parms** parameter must match the order in the subject of the certificate.

For example, if your certificate's subject is

e=user_name@example.com,cn=user_name,o=Example Inc.,c=US, and you want Directory Server to use **cn=user_name,o=Example Inc.,c=US** as base DN when searching for the user, set the **DNC parms** parameter to **cn, o, c**.



IMPORTANT

The values of attributes set in the **DNComps** parameter must be unique in the database.

- Set the parameter to an empty value if the base DN cannot be generated from the **subject** field of the certificate. In this situation, Directory Server searches the for user in the entire directory using a filter generated from the setting in the **FilterComps** parameter.
For example, if the certificate's subject is **e=user_name@example.com,cn=user_name,o=Example Inc.,c=US**, but Directory Server stores its data in the **dc=example,dc=com** entry, Directory Server cannot generate a valid base DN from the subject of the certificate, because the required components are not part of the subject. In this case, set **DNComps** to an empty string to search for the user in the entire directory.
- Comment out or do not set this parameter, if either the **subject** field of the certificate matches exactly the DN of the user in Directory Server or if you want to use the setting from the **CmapLdapAttr** parameter.
Alternatively, set the **nsslapd-certmap-basedn** parameter in the **cn=config** entry to use a hard-coded base DN.

FilterComps

This parameter sets which attributes from the **subject** field of the certificate Directory Server uses to generate the search filter to locate the user:

- Set this parameter to a comma-separated list of attributes used in the certificate's subject. Directory Server will use these attributes in an **AND** operation in the filter.



NOTE

Certificate Subjects use the **e** attribute for the email address, which does not exist in the default Directory Server schema. For this reason, Directory Server automatically maps this attribute to the **mail** attribute. This means, if you use the **mail** attribute in the **FilterComps** parameter, Directory Server reads the value of the **e** attribute from the subject of the certificate.

For example, if the subject of a certificate is

e=user_name@example.com,cn=user_name,dc=example,dc=com,o=Example Inc.,c=US and you want to dynamically generate the **(&(mail=username@domain)(cn=user_name))** filter, set the **FilterComps** parameter to **mail,cn**.

- If the parameter is commented out or set to an empty value, the **(objectclass=*)** filter will be used.

verifycert

Directory Server always verifies if the certificate has been issued by a trusted Certificate Authority (CA). However, if you additionally set the **verifycert** parameter to **on**, Directory Server additionally verifies that the certificate matches the Distinguished Encoding Rules (DER)-formatted certificate stored in the **userCertificate** binary attribute of the user.

If you do not set this parameter, **verifycert** is disabled.

CmapLdapAttr

If your user entries contain an attribute that stores the subject DN of the user certificate, set the

CmapLdapAttr to this attribute name. Directory Server will use this attribute and the subject DN to locate the user. In this case the no filter is generated based on the attributes in the **FilterComps** parameter.

library

Sets the path name to a shared library or Dynamic Link Library (DLL) file. Use this setting only if you create your own properties using the certificate API. This parameter is deprecated and will be removed in a future release.

InitFn

Sets the name of the **init** function, if you use a custom library. Use this setting only if you create your own properties using the certificate API. This parameter is deprecated and will be removed in a future release.



IMPORTANT

When Directory Server searches the matching user, the search must return exactly one entry. If the search returns multiple entries, Directory Server logs a **multiple matches** error and authentication fails.

For further details, see the corresponding section in the *Directory Server Administration Guide*.

CHAPTER 9. COMMAND-LINE UTILITIES

This chapter contains reference information on command-line utilities used with Red Hat Directory Server (Directory Server). These command-line utilities make it easy to perform administration tasks on the Directory Server.

9.1. DS-REPLCHECK

The **ds-replcheck** utility compares two Directory Server instances or LDIF-formatted files to identify if they are synchronized. For further details, see the *Comparing Two Directory Server Instances* section in the [Red Hat Directory Server Administration Guide](#).

For details about the syntax and command-line options, see the **ds-replcheck(1)** man page.

9.2. LDIF

ldif automatically formats LDIF files and creates base-64 encoded attribute values. Base-64 encoding makes it possible to represent binary data, such as a JPEG image, in LDIF. Base-64 encoded data is represented using a double colon (::) symbol. For example:

jpegPhoto:: encoded data

In addition to binary data, other values that must be base-64 encoded can be identified with other symbols, including the following:

- Any value that begins with a space.
- Any value that begins with a single colon (:).
- Any value that contains non-ASCII data, including newlines.

The **ldif** command-line utility will take any input and format it with the correct line continuation and appropriate attribute information. The **ldif** utility also senses whether the input requires base-64 encoding.

For details about the syntax and command-line options, see the **ldif(5)** man page.

9.3. DBSCAN

The **dbscan** tool analyzes and extracts information from a Directory Server database file. There are four kinds of database files that can be scanned with **dbscan**:

- **id2entry.db**, the main database file for a user database
- **entryrdn.db** for a user database
- secondary index files for a user database, like **cn.db**
- **numeric_string.db** for the changelog in **/var/lib/dirsrv/slappd-instance/changelogdb**

See [Section 2.2.2, “Database Files”](#) for more information on database files.

Database files use the **.db2**, **.db3**, **.db4**, and **.db** extensions in their filename, depending on the version of Directory Server.

For details about the syntax and command-line options, see the **dbscan(1)** man page.

Examples

The following are command-line examples of different situations using **dbscan** to examine the Directory Server databases.

Example 9.1. Dumping the Entry File

```
dbscan -f /var/lib/dirsrv/slapd-instance/db/userRoot/id2entry.db
```

Example 9.2. Displaying the Index Keys in cn.db

```
dbscan -f /var/lib/dirsrv/slapd-instance/db/userRoot/cn.db
```

Example 9.3. Displaying the Index Keys and the Count of Entries with the Key in mail.db

```
# dbscan -r -f /var/lib/dirsrv/slapd-instance/db/userRoot/mail.db
```

Example 9.4. Displaying the Index Keys and the All IDs with More Than 20 IDs in sn.db

```
# dbscan -r -G 20 -f /var/lib/dirsrv/slapd-instance/db/userRoot/sn.db
```

Example 9.5. Displaying the Summary of objectclass.db

```
# dbscan -s -f /var/lib/dirsrv/slapd-instance/db/userRoot/objectclass.db
```

Example 9.6. Displaying VLV Index File Contents

```
# dbscan -r -f /var/lib/dirsrv/slapd-instance/db/userRoot/vlv#bymccoupeopledcpeopleledccom.db
```

Example 9.7. Displaying the Changelog File Contents

```
# dbscan -f /var/lib/dirsrv/slapd-instance/changelogdb/c1a2fc02-1d11b2-8018afa7-fdce000_424c8a000f00.db
```

Example 9.8. Dumping the Index File uid.db with Raw Mode

```
# dbscan -R -f /var/lib/dirsrv/slapd-instance/db/userRoot/uid.db
```

Example 9.9. Displaying the entryID with the Common Name Key "=hr managers"

In this example, the common name key is **=hr managers**, and the equals sign (=) means the key is an equality index.

```
# dbscan -k "=hr managers" -r -f /var/lib/dirsrv/slappd-instance/db/userRoot/cn.db
=hr%20managers 7
```

Example 9.10. Displaying an Entry with the entry ID of 7

```
# dbscan -K 7 -f /var/lib/dirsrv/slappd-instance/db/userRoot/id2entry.db

id 7 dn: cn=HR Managers,ou=groups,dc=example,dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: HR Manager
ou: groups
description: People who can manage HR entries
creatorsName: cn=Directory Manager
modifiersName: cn=Directory Manager
createTimestamp: 20050408230424Z
modifyTimestamp: 20050408230424Z
nsUniqueId: 8b465f73-1dd211b2-807fd340-d7f40000 parentid: 3
entryid: 7
entrydn: cn=hr managers,ou=groups,dc=example,dc=com
```

Example 9.11. Displaying the Contents of entryrdn Index

```
# dbscan -f /var/lib/dirsrv/slappd-instance/db/userRoot/entryrdn.db -k "dc=example,dc=com"

dc=example,dc=com
ID: 1; RDN: "dc=example,dc=com"; NRDN: "dc=example,dc=com"
C1:dc=example,dc=com
ID: 2; RDN: "cn=Directory Administrators"; NRDN: "cn=directory administrators"
2:cn=directory administrators
ID: 2; RDN: "cn=Directory Administrators"; NRDN: "cn=directory administrators"
P2:cn=directory administrators
ID: 1; RDN: "dc=example,dc=com"; NRDN: "dc=example,dc=com"
C1:dc=example,dc=com
ID: 3; RDN: "ou=Groups"; NRDN: "ou=groups"
3:ou=groups
ID: 3; RDN: "ou=Groups"; NRDN: "ou=groups"
[...]
```

9.4. DS-LOGPIPE.PY

The named pipe log script can replace any of the Directory Server log files (access, errors, and audit) with a named pipe. That pipe can be attached to another script which can process the log data before sending it to output, such as only writing lines that match a certain pattern or are of a certain event type.

Using a named pipe script provides flexibility:

- The error log level can be set very high for diagnosing an issue to create a log of only the last few hundred or thousand log messages, without a performance hit.
- Messages can be filtered to keep only certain events of interest. For example, the named pipe script can record only failed BIND attempts in the access log, and other events are discarded.
- The script can be used to send notifications when events happen, like adding or deleting a user entry or when a specific error occurs.

For details about the syntax and command-line options, see the **ds-logpipe.py(1)** man page.

Examples

The procedures for configuring the server for named pipe logging are covered in [Section 7.5, “Replacing Log Files with a Named Pipe”](#).

The most basic usage of the named pipe log script points to only the named pipe.

Example 9.12. Basic Named Pipe Log Script

```
# ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe
```



NOTE

When the script exits (either because it completes or because it is terminated through a SIGTERM or Ctrl+C), the script dumps the last 1000 lines of the error log to standard output.

The script can be run in the background, and you can interactively monitor the output. In that case, the command **kill -1 %1** can be used to tell the script to dump the last 1000 lines of the buffer to stdout, and continue running in the background.

Example 9.13. Running the Named Pipe Log Script in the Background

```
# ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe &
```

To simply dump the last 1000 lines when the script exits (or is killed or interrupted) and save the output to a file automatically, redirect the script output to a user-defined file.

Example 9.14. Saving the Output from the Named Pipe Log Script

```
# ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe > /etc/dirsrv/myerrors.log 2>&1
```

The named pipe script can be configured to start and stop automatically with the Directory Server

process. This requires the name of the server’s PID file to which to write the script’s PID when the script is running, with the **-s** argument. The PID for the server can be reference either by pointing to the server PID file or by giving the actual process ID number (if the server process is already running).

Example 9.15. Specifying the Server PID

```
# ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe --serverpidfile /var/run/dirsrv/slapd-example.pid
```

A plug-in can be called to read the log data from the named pipe and perform some operation on it.

Example 9.16. Named Pipe Log Script with a Related Plug-in

```
# ds-logpipe.py /var/log/dirsrv/slapd-example/errors.pipe --plugin=/usr/share/dirsrv/data/logregex.py logregex.regex="warning"
```

In [Example 9.16, “Named Pipe Log Script with a Related Plug-in”](#), only log lines containing the string **warning** are stored in the internal buffer and printed when the script exits.

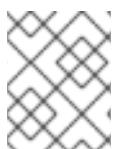
If no plug-in is passed with the script arguments, the script just buffers 1000 log lines (by default) and prints them upon exit. There are two plug-ins provided with the script:

- **logregex.py** keeps only log lines that match the given regular expression. The plug-in argument has the format **logregex.regex=pattern** to specify the string or regular expression to use. There can be multiple **logregex.regex** arguments which are all treated as AND statements. The error log line must match all given arguments. To allow any matching log lines to be records (OR), use a single **logregex.regex** argument with a pipe (|) between the strings or expressions. See the pcre or Python regular expression documentation for more information about regular expressions and their syntax.
- **failedbinds.py** logs only failed BIND attempts, so this plug-in is only used for the access log. This takes the option **failedbinds.logfile=/path/to/access.log**, which is the file that the actual log messages are written to. This plug-in is an example of a complex plug-in that does quite a bit of processing and is a good place to reference to do other types of access log processing.

9.5. DN2RDN

Versions of Directory Server older than 9.0 used the **entrydn** index to help map the entry IDs in the **id2entry.db4** database to the full DNs of the entry. (One side effect of this was that modrdn operations could only be done on leaf entries, because there was no way to identify the children of an entry and update their DNs if the parent DN changed.) When subtree-level renames are allowed, then the ID-to-entry mapping is done using the **entryrdn** index with the **id2entry.db** database.

After an upgrade, instances of Directory Server may still be using the **entrydn** index. The **dn2rdn** tool has one purpose: to convert the entry index mapping from a DN-based format to an RDN-based format, by converting the **entrydn** index to **entryrdn**.



NOTE

The **dn2rdn** tool is in the **/usr/sbin/** directory, since it is always run on the local Directory Server instance.

9.6. PWDHASH

The **pwdhash** utility encrypts a specified plain text password. If a user or the Directory Manager cannot log in, use **pwdhash** to compare the encrypted passwords. You can also use the generated hash to manually reset the Directory Manager's password.

The **pwdhash** utility uses the following storage scheme to encrypt the password:

- If you pass the **-s *storage_scheme*** parameter to **pwdhash**, the specified scheme will be used.
- If you pass the **-D *config_directory*** parameter to **pwdhash**, the scheme set in the **nsslapd-rootpwstoragescheme** attribute will be used.
- If you neither specify the path to a valid Directory Server configuration directory nor pass a scheme to **pwdhash**, the utility uses the Directory Server default storage scheme.

For further details about storage schemes, a list of supported values, and the default settings, see [Section 4.1.43, “Password Storage Schemes”](#).

For details about the syntax and command-line options, see the **pwdhash(1)** man page.

APPENDIX A. TESTING SCRIPTS AVAILABLE WITH DIRECTORY SERVER

Red Hat Directory Server provides two scripts which can be used to test Directory Server performance in different stress or load conditions. The test scripts simulate different environments which allow administrators to assess configuration or machine changes before putting them in production.

Both **ldclt** and **rsearch** are located in the **/usr/bin** directory.

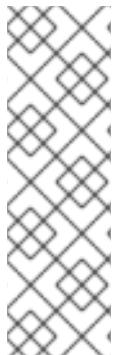
A.1. LDCLT (LOAD STRESS TESTS)

The LDAP client script (**ldclt**) establishes multiple client connections to a server, under user-defined scenarios, to load-test the Directory Server. Client operations include directory adds, searches, modifies, modRDNs, and deletes, as well setup operations like generating LDIF files. Operations can be randomized – binding and unbinding as random users, performing random tasks – to simulate more realistic usage environments for the directory.

The **ldclt** tool measures the completion time of continuously-repeated operations to measure Directory Server performance. Using multiple threads makes it possible to test performance under high loads. Each test performs the same type of LDAP operation, but with different settings (like different user credentials, different attribute types or sizes, and different target subtrees).

Along with defining the LDAP operation variables, administrators can control the thread performance in order to set a specific load on the server.

The **ldclt** tool is specifically intended to be used for automated tests, so its options are extensive, flexible, and easily scripted, even for complex test operations.



NOTE

Remember that **ldclt** is a load test, and therefore uses a significant amount of system resources. The tool uses a minimum of 8 MB of memory. Depending on the numbers of threads, types of operations, and other configuration settings, it can use much more memory.

Depending on the type of operations and the directory data used for those operations, **ldclt** may set its own resource limits. For information on managing system resource limits, see the man pages for **ulimit** and **getrlimit**.

The **ldclt** utility is located in the **/usr/bin** directory.

A.1.1. Syntax

```
ldclt-q-Q-v-V-Emax_errors-bbase_DN-hhost-pport-ttimeout-Dbind_DN-wpassword-oSASL_options-eexecution_params-amax_pending-nnumber_of_threads-iinactivity_times-Nnumber_of_samples-lerror_code-Ttotal_number_of_operations-rlow_range-Rhigh_range-ffilter-sscope-Sconsumer-Psupplier_port-Wwait_time-Zcertificate_file
```

A.1.2. ldclt Options

Table A.1. ldclt Options

Option	Description
-a <i>max_pending_ops</i>	Runs the tool in asynchronous mode with a defined maximum number of pending operations.
-b <i>base_dn</i>	Gives the base DN to use for running the LDAP operation tests. If not given, the default value is dc=example,dc=com .
-D <i>bind_dn</i>	Gives the bind DN for the ldclt utility to use to connect to the server.
-E <i>max_errors</i>	Sets the maximum number of errors that are allowed to occur in test LDAP operations before the tool exits. The default is 1000.
-e <i>execution_params</i>	Specifies the type of operation and other test environment parameters to use for the tests. The possible values for -e are listed in Table A.2, "Execution Parameters" . This option can accept multiple values, in a comma-separated list.
-f <i>filter</i>	Gives an LDAP search filter to use for search testing.
-h	Specifies the host name or IP address of the Directory Server to run tests against. If a host is not specified, ldclt uses the local host.
-I <i>error_code</i>	Tells ldclt to ignore any errors encountered that match a certain response code. For example, -I 89 tells the server to ignore error code 89.
-i <i>inactivity_times</i>	Sets a number of intervals that the tool can be inactive before exiting. By default, this setting is 3, which translates into 30 seconds (each operations interval being 10 seconds long).
-N <i>number_of_samples</i>	Sets the number of iterations to run, meaning how many ten-second test periods to run. By default, this is infinite and the tool only exits when it is manually stopped.
-n <i>number_of_threads</i>	Sets the number of threads to run simultaneously for operations. The default value is 10.

Option	Description
<p>-o <i>SASL_option</i></p>	<p>Tells the tool to connect to the server using SASL and gives the SASL mechanism to use. The format is -o sasl/Option=value. <i>sasl/Option</i> can have one of six values:</p> <ul style="list-style-type: none"> * mech, the SASL authentication mechanism * authid, the user who is binding to the server (Kerberos principal) * authzid, a proxy authorization (ignored by the server since proxy authorization is not supported) * secProp, the security properties * realm, the Kerberos realm * flags <p>The expected values depend on the supported mechanism. The -o can be used multiple times to pass all of the required SASL information for the mechanism. For example:</p> <pre>[literal,subs="+quotes,verbatim"] -o "mech=DIGEST-MD5" -o "authzid=test_user" -o "authid=test_user"</pre>
<p>-P <i>supplier_port</i></p>	<p>Gives the port to use to connect to a supplier server for replication testing. The default, if one is not given, is 16000.</p>
<p>-p <i>port</i></p>	<p>Gives the server port number of the Directory Server instance that is being tested.</p>
<p>-Q</p>	<p>Runs the tool in "super" quiet mode. This ignores any errors that are encountered in operations run by ldcilt.</p>
<p>-q</p>	<p>Runs the tool in quiet mode.</p>
<p>-R <i>number</i></p>	<p>Sets the high number for a range.</p>
<p>-r <i>number</i></p>	<p>Sets the low number of a range.</p>
<p>-S <i>consumer_name</i></p>	<p>Gives the host name of a consumer server to connect to run replication tests.</p>
<p>-s <i>scope</i></p>	<p>Gives the search scope. As with ldapsearch, the values can be subtree, one, or base.</p>

Option	Description
<code>-T <i>ops_per_thread</i></code>	Sets a maximum number of operations allowed per thread.
<code>-t <i>timeout</i></code>	Sets a timeout period for LDAP operations. The default is 30 seconds.
<code>-V</code>	Runs the tool in very verbose mode.
<code>-v</code>	Runs the tool in verbose mode.
<code>-W <i>wait_time</i></code>	Sets a time, in seconds, for the ldclt tool to wait after one operation finishes to start the next operation. The default is 0, which means there is no wait time.
<code>-w <i>password</i></code>	Gives the password to use, with the -D identity, to bind to the Directory Server for testing.
<code>-Z/<i>path/to/cert.db</i></code>	Enables TLS for the test connections and points to the file to use as the certificate database.

The **-e** option sets execution parameters for the **ldclt** test operations. Multiple parameters can be configured, in a comma-separated list. For example:

```
-e add,bindeach,genldif=/var/lib/dirsrv/slapd-instance/ldif/generated.ldif/inetOrgPerson
```

Table A.2. Execution Parameters

Parameter	Description
abandon	Initiates abandon operations for asynchronous search requests.
add	Adds entries to the directory (ldapadd).
append	Appends entries to the end of the LDIF file generated with the genldif option.
ascii	Generates ASCII 7-bit strings.
attreplace= <i>name:mask</i>	Run modify operations that replace an attribute (<i>name</i>) in an existing entry.

Parameter	Description
attrlist=name:name:name	Specifies a list of attributes to return in a search operation.
attrsonly=#	Used with search operations, to set whether to read the attribute values. The possible values are 0 (read values) or 1 (do not read values).
bindeach	Tells the ldclt tool to bind with each operation it attempts.
bindonly	Tells the ldclt tool to only run bind/unbind operations. No other operation is performed.
close	Tells the tool to close the connection rather than perform an unbind operation.
cltcertname=name	Gives the name of the TLS client certificate to use for TLS connections.
commoncounter	Makes all threads opened by the ldclt tool to share the same counter.
counteach	Tells the tool to count each operation, not only successful ones.
delete	Initiates delete operations.
deref	Adds the dereference control to search operations (esearch). With adds, this tells ldclt to add the secretary attribute to new entries, to allow dereference searches.
dontsleeponserverdown	Causes the tool to loop very fast if server down.
emailPerson	This adds the emailPerson object class to generated entries. This is only valid with the add operation (-e add).
esearch	Performs an exact search.
genldif=filename	Generates an LDIF file to use with the operations.
imagesdir=path	Gives a location for images to use with tests.
incr	Enables incremental values.

Parameter	Description
inetOrgPerson	This adds the inetOrgPerson object class to generated entries. This is only valid with the add operation (-e add).
keydbfile= <i>file</i>	Contains the path and file name of the key database to use with TLS connections.
keydbpin= <i>password</i>	Contains the token password to access the key database.
noglobalstats	Tells the tool <i>not</i> to print periodical global statistics.
noloop	Does not loop the incremental numbers.
object= <i>filename</i>	Builds entry objects from an input file.
person	This adds the person object class to generated entries. This is only valid with the add operation (-e add).
random	Tells the ldclt utility to use all random elements, such as random filters and random base DNS.
randomattrlist= <i>name:name:name</i>	Tells the ldclt utility to select random attributes from the given list.
randombase	Tells the ldclt utility to select a random base DN from the directory.
randombaselow= <i>value</i>	Sets the low value for the random generator.
randombasehigh= <i>value</i>	Sets the high value for the random generator.
randombinddn	Tells the ldclt utility to use a random bind DN.
randombinddnfromfile= <i>file</i>	Tells the ldclt utility to use a random bind DN, selected from a file. Each entry in the file must have the appropriate DN–password pair.
randombinddnlow= <i>value</i>	Sets the low value for the random generator.
randombinddnhigh= <i>value</i>	Sets the high value for the random generator.
rdn= <i>attrname:value</i>	Gives an RDN to use as the search filter. This is used instead of the -f filter.

Parameter	Description
referral= <i>value</i>	Sets the referral behavior for operations. There are three options: on (allow referrals), off (disallow referrals), or rebind (attempt to connect again).
smoothshutdown	Tells the IdcIt utility not to shut down its main thread until the worker threads exit.
string	Tells the IdcIt utility to create random strings rather than random numbers.
v2	Tells the IdcIt utility to use LDAPv2 for test operations.
withnewparent	Performs a modRDN operation, renaming an entry with newparent set as an argument.
randomauthid	Uses a random SASL authentication ID.
randomauthidlow= <i>value</i>	Sets the low value for a random SASL authentication ID.
randomauthidhigh= <i>value</i>	Sets the high value for the random SASL authentication ID.

A.1.3. Results from IdcIt

IdcIt continuously runs whatever operation is specified, over the specified number of threads. By default, it prints the performance statistics to the screen every ten (10) seconds.

The results show the average number of operations per thread and per second and then the total number of operations that were run in that ten-second window.

IdcIt[process_id] Average rate: *number_of_ops/thr* (*number_of_ops/sec*), total: *total_number_of_ops*

For example:

IdcIt[22774]: Average rate: 10298.20/thr (15447.30/sec), total: 154473

IdcIt prints cumulative averages and totals every 15 minutes and when the tool is exited.

IdcIt[22774]: Global average rate: 821203.00/thr (16424.06/sec), total: 12318045

IdcIt[22774]: Global number times "no activity" reports: never

IdcIt[22774]: Global no error occurs during this session.

Catch SIGINT - exit...

IdcIt[22774]: Ending at Wed Feb 24 18:39:38 2010

IdcIt[22774]: Exit status 0 - No problem during execution.

Some operations (like adds) and using verbose output options like **-v** or **-V** output additional data to the screen. The kind of information depends on the type of operation, but it generally shows the thread performing the operation and the plug-ins called by the operation. For example:

```
ldcI -b ou=people,dc=example,dc=com -D "cn=Directory Manager" -w secret12 -e
add,person,incr,noloop,commoncounter -r90000 -R99999 -f "cn=testXXXXX" -V

...
ldcI[11176]: T002: After ldap_simple_bind_s (cn=Directory Manager, secret12)
ldcI[11176]: T002: incremental mode:filter="cn=test00009"
ldcI[11176]: T002: tttctx->bufFilter="cn=test00009"
ldcI[11176]: T002: attrs[0]=("objectclass" , "person")
ldcI[11176]: T002: attrs[1]=("cn" , "test00009")
ldcI[11176]: T002: attrs[2]=("sn" , "toto sn")

...
ldcI[11176]: Average rate: 195.00/thr ( 195.00/sec), total: 1950
ldcI[10627]: Global average rate: 238.80/thr (238.80/sec), total: 2388
ldcI[10627]: Global number times "no activity" reports: never
ldcI[10627]: Global no error occurs during this session.
Catch SIGINT - exit...
ldcI[10627]: Ending at Tue Feb 23 11:46:04 2010
ldcI[10627]: Exit status 0 - No problem during execution.
```

Most errors are handled by **ldcI** without interrupting the test. Any fatal errors that are encountered are listed with the tool's exit status and returned in the cumulative total.

Global no error occurs during this session.

Any LDAP operations errors that occur are handled within the thread. A connection error kills the thread without affecting the overall test. The **ldcI** utility does count the number of times each LDAP error is encountered; if the total number of errors that are logged hits more than 1000 (by default), then the script itself will error out.

The way that **ldcI** responds to LDAP errors can be configured. Using the **-E** option sets a different threshold for the script to error out after encountering LDAP errors. Using the **-I** option tells the script to ignore the specified LDAP error codes in all threads. Changing the error exit limit and ignoring certain error codes can allow you to tweak and improve test scripts or test configuration.

A.1.4. Exiting ldcI and ldcI Exit Codes

The **ldcI** command runs indefinitely. The script can stop itself in a handful of situations, like encountering a fatal runtime or initialization error, hitting the limit of LDAP errors, having all threads die, or hitting the operation or time limit.

The statistics for the run are not displayed until the command completes, either through the script exiting or by a user terminating the script. There are two ways to interrupt the **ldcI** script.

- Hitting control-backslash (**kbd:[^\`]**) or **kill -3** prints the current statistics without exiting the script.
- Hitting control-C (**^C**) or **kill -2** exits the script and prints the global statistics.

When the **ldcI** script exits or is interrupted, it returns an exit code along with the statistics and error information.

Table A.3. Idclt Exit Codes

Exit Code	Description
0	Success (no errors).
1	An operation encountered a serious fatal error.
2	There was an error in the parameters passed with the tool.
3	The tool hit the maximum number of LDAP errors.
4	The tool could not bind to the Directory Server instance.
5	The tool could not load the TLS libraries to connect over TLS.
6	There was a multithreading (mutex) error.
7	There was an initialization problem.
8	The tool hit a resource limit, such as a memory allocation error.
99	The script encountered an unknown error.

A.1.5. Usage Scenarios

These provide general examples of using **Idclt** to test Directory Server. Test scripts with more complex examples are available in the **Idclt** source files. This can be downloaded from the 389 Directory Server Project, <https://git.fedorahosted.org/cgit/389/ds.git/tree/ldap/servers/slapd/tools/ldclt/examples>.

Every **Idclt** command requires a set of execution parameters (which varies depending on the type of test) and connection parameters (which are the same for every type of operation). For example:

```
# ldclt -e execution_parameters -h localhost -p 389 -D "cn=Directory Manager" -w secret -b  
"ou=people,dc=example,dc=com"
```

When **Idclt** runs, it first prints all of the configured parameters for that test.

```
Process ID      = 1464  
Host to connect = localhost  
Port number    = 389  
Bind DN        = cn=Directory Manager  
Passwd         = secret  
Referral       = on  
Base DN        = ou=people,dc=example,dc=com  
Filter          = "cn=MrXXX"  
Max times inactive = 3
```

```

Max allowed errors = 1000
Number of samples = -1
Number of threads = 10
Total op. req. = -1
Running mode = 0xa0000009
Running mode = quiet verbose random exact_search
LDAP oper. timeout = 30 sec
Sampling interval = 10 sec
Scope = subtree
Attrsonly = 0
Values range = [0 , 1000000]
Filter's head = "cn=Mr"
Filter's tail =

```

A.1.5.1. Generating LDIFs

The **ldclt** tool itself can be used to generate LDIF files that can be used for testing.



NOTE

When generating an LDIF file, the **ldclt** tool does not attempt to connect to a server or run any operations.

Generating an LDIF file requires a basic template file that the tool uses to create entries (**-e object**), and then a specified output file (**-e genldif**).

The template file can give explicit values for entry attributes or can use variables. If you want a simple way to supply unique values for entry attributes, the **/usr/share/dirsrv/data** directory contains three data files to generate surnames, first names, and organizational units. These lists of values can be used to create test users and directory trees (**dbgen-FamilyNames**, **dbgen-GivenNames**, and **dbgen-OrgUnits**, respectively). These files can be used with the **rndfromfile**, **incrfromfile**, or **incrfromfilenoloop** options.

The basic format of the template file is:

```

# comment

attribute: string | variable=keyword(value)

```

The variable can be any letter from A to H. The possible keywords are listed in [Table A.4, “ldclt Template LDIF File Keywords”](#)

Some variables and keywords can be passed with the **-e object** option and other available parameters (like **rdn**).

```
-e object=inet.txt,rdn='uid:[A=INCRNNOLOOP(0;99999;5)]'
```

Table A.4. ldclt Template LDIF File Keywords

Keyword	Description	Format
---------	-------------	--------

Keyword	Description	Format
RNDN	Generates a random value within the specified range (low – high) and of the given length.	RNDN(low;high;length)
RNDFROMFILE	Pulls a random value from any of the ones available in the specified file.	RNDFROMFILE(filename)
INCRN	Creates sequential values within the specified range (low – high) and of the given length.	INCRN(low;high;length)
INCRNOLOOP	Creates sequential values within the specified range (low – high) and of the given length – without looping through the incremental range.	INCRNOLOOP(low;high;length)
INCRRFROMFILE	Creates values by incrementing through the values in the specified file.	INCRRFROMFILE(filename)
INCRRFROMFILENOLOOP	Creates values by incrementing through the values in the file, without looping back through the values.	INCRRFROMFILENOLOOP(filename)
RNDS	Generates random values of a given length.	RNDS(length)

For example, this template file pulls names from sample files in the **/usr/share/dirsrv/data** and builds other attributes dynamically.

Example A.1. Example Template File

```
objectclass: inetOrgPerson
sn: [B=RNDFROMFILE(/usr/share/dirsrv/data/dbgen-FamilyNames)]
cn: [C=RNDFROMFILE(/usr/share/dirsrv/data/dbgen-GivenNames)] [B]
password: test[A]
description: user id [A]
mail: [C].[B]@example.com
telephonenumber: (555) [RNDN(0;999;3)]-[RNDN(0;9999;4)]
```

The **ldclt** command, then, uses that template to build an LDIF file with 100,000 entries:

```
# ldclt -b "ou=people,dc=csb" -e object=inet.txt,rdn='uid:[A=INCRNNOLOOP(0;99999;5)]' -e
genldif=100Kinet.ldif,commoncounter
```

A.1.5.2. Adding Entries

The **ldclt** tool can add entries that match either of two templates:

- person
- inetorgperson

The **-f** filter sets the format of the naming attribute for the user entries. For example, **-f "cn=MrXXXXXX"** creates a name like **-f "cn=Mr01234"**. Using the **person** or **inetorgperson** parameter with **-f** creates a basic entry.

```
objectclass: person
sn: ex sn
cn: Mr01234
```

More complex entries (which are good for search and modify testing) can be created using the **rdn** parameter and an **object** file. The full range of options for the entries is covered in [Section A.1.5.1, “Generating LDIFs”](#). The **rdn** and **object** parameters provide the format for the entries to add or edit in the directory. The **rdn** execution parameter takes a keyword pattern (as listed in [Table A.4, “ldclt Template LDIF File Keywords”](#)) and draws its entry pool from the entries listed in a text file.

```
-e rdn='uid:[A=INCRNNOLOOP(0;99999;5)]',object=inet.txt
```

The **ldclt** tool creates entries in a numeric sequence. That means that the method of adding those entries and of counting the sequence have to be defined as well. Some possible options for this include:

- **-r** and **-R** to set the numeric range for entries
- **incr** or **random** to set the method of assigning numbers (these are only used with **-f**)
- **-r** and **-R** to set the numeric range for entries
- **noloop**, to stop the add operations when it hits the end of the range rather than looping back

Example A.2. Adding Entries

```
# ldclt -b ou=people,dc=example,dc=com -D "cn=Directory Manager" -w secret -e
add,person,incr,noloop,commoncounter -r0 -R99999 -f "cn=MrXXXXXX" -v -q
```

The **add** operation can also be used to build a directory tree for more complex testing. Whenever an entry is added to the directory that belongs to a non-existent branch, the **ldclt** tool automatically creates that branch entry.



NOTE

The first time that an entry is added that is the child of non-existent branch, the branch entry is added to the directory. However, the entry itself is not added. Subsequent entries will be added to the new branch.

For a branch entry to be added automatically, its naming attribute must be **cn**, **o**, or **ou**.

Example A.3. Creating the Directory Tree

```
# ldclt -b ou=DeptXXX,dc=example,dc=com -D "cn=Directory Manager" -w secret -e
add,person,incr,noloop,commoncounter -r0 -R99999 -f "cn=MrXXXXX" -v -q
```

A.1.5.3. Search Operations

The most basic **ldclt** search test simply looks for all entries within the given base DN. This uses two execution parameters: **esearch** and **random**.

Example A.4. Basic Search Operation

```
# ldclt -h localhost -p 389 -D "cn=Directory Manager" -w secret -b
"ou=people,dc=example,dc=com" -f uid=testXXXXX -e esearch,random -r0 -R99999 -l 32
```



IMPORTANT

A search that returns all entries can use a large amount of memory per thread, as much as 1 GB. **ldclt** is designed to perform searches that return one entry.

The search results can be expanded to return attributes contained in the entries. ([Section A.1.5.1, “Generating LDIFs”](#) has information on generating entries that contain multiple attributes.) To return a specific list of attributes for entries, use the **attrlist** execution parameter and a colon-separated list of attributes.

Example A.5. Searching for a List of Attributes

```
# ldclt -h localhost -p 389 -b "ou=people,dc=example,dc=com" -f uid=XXXXXX -e esearch,random -
r0 -R99999 -l 32 -e attrlist=cn:mail
```

Alternatively, the **ldclt** search operation can return attribute values for attributes randomly selected from the search list. The list is given in the **randomattrlist** execution parameter with a colon-separated list of attributes.

Example A.6. Searching for a List of Random Attributes

```
# ldclt -h localhost -p 389 -b "ou=people,dc=example,dc=com" -f uid=XXXXXX -e esearch,random -
r0 -R99999 -l 32 -e randomattrlist=cn:sn:ou:uid:mail:mobile:description
```

The filter used to match entries can target other entry attributes, not just naming attributes. It depends on the attributes in the generated LDIF.

Example A.7. Searches with Alternate Filters

```
# ldclt -h localhost -p 389 -b "ou=people,dc=example,dc=com" -f mail=XXXXXX@example.com -
e esearch,random -r0 -R99999 -l 32 -e randomattrlist=cn:sn:ou:uid:mail:mobile:description
```

The search operation can also use the RDN-style filter to search for entries. The **rdn** and **object** execution parameters provide the format for the entries to add or edit in the directory. The **rdn** execution parameter takes a keyword pattern (as listed in [Table A.4, “Idclt Template LDIF File Keywords”](#)) and draws its entry pool from the entries listed in a text file.

Example A.8. Searches with RDN Filters

```
# Idclt -h localhost -p 389 -b "ou=people,dc=example,dc=com" -e rdn='mail:[RNDN(0;99999;5)]@example.com',object="inet.txt" -e attrlist=cn:telephonenumber
```

A.1.5.4. Modify Operations

The **attrreplace** execution parameter replaces specific attributes in the entries.

The modify operation uses the RDN filter to search for the entries to update. The **rdn** and **object** parameters provide the format for the entries to add or edit in the directory. The **rdn** execution parameter takes a keyword pattern (as listed in [Table A.4, “Idclt Template LDIF File Keywords”](#)) and draws its entry pool from the entries listed in a text file.

Example A.9. Modify Operation

```
# Idclt -h localhost -p 389 -D "cn=Directory Manager" -w secret -b "ou=people,dc=example,dc=com" -e rdn='uid:[RNDN(0;99999;5)]' -l 32 -e attrreplace='description: random modify XXXXX'
```

A.1.5.5. modrdn Operations

The **Idclt** command supports two kinds of modrdn operations:

- Renaming entries
- Moving an entry to a new parent

The **Idclt** utility creates the new entry name or parent from a randomly-selected DN.

The basic rename operation requires three execution parameters:

- rename
- rdn='pattern'
- object=file

The **rdn** and **object** parameters provide the format for the entries to add or edit in the directory. The **rdn** execution parameter takes a keyword pattern (as listed in [Table A.4, “Idclt Template LDIF File Keywords”](#)) and draws its entry pool from the entries listed in a text file.

Example A.10. Simple Rename Operation

```
# Idclt -h localhost -p 389 -D "cn=Directory Manager" -w secret -b "ou=people,dc=example,dc=com" -l 32 -l 68 -e rename,rdn='uid:[RNDN(0;999;5)]',object="inet.txt"
```

Using the **withnewparent** execution parameter renames the entry and moves it beneath a new parent entry. If the parent entry does not exist, then the **ldcIt** tool creates it.^[3]

Example A.11. Renaming an Entry and Moving to a New Parent

```
# ldcIt -h localhost -p 389 -D "cn=Directory Manager" -w secret12 -b
"ou=DeptXXX,dc=example,dc=com" -l 32 -I 68 -e
rename,withnewparent,rdn='uid:Mr[RNDN(0;99999;5)]',object="inet.txt"
```

A.1.5.6. Delete Operations

The **ldcIt** delete operation is exactly the reverse of the add operation. As with the add, delete operations can remove entries in several different ways:

- Randomly (**-e delete,random**)
- RDN-ranges (**-e delete,rdn=[pattern]**)
- Sequentially (**-e delete,incr**)

Random deletes are configured to occur within the specified range of entries. This requires the following options:

- **-e delete,random**
- **-r** and **-R** for the range bounds
- **-f** for the filter to match the entries

Example A.12. Random Delete Operations

```
# ldcIt -b "ou=people,dc=example,dc=com" -D "cn=Directory Manager" -w secret -e delete,random
-r0 -R99999 -f "uid=XXXXXX" -l 32 -v -q
```

RDN-based deletes use the **rdn** execution parameter with a keyword (as listed in [Table A.4, "ldcIt Template LDIF File Keywords"](#)) and draws its entry pool from the entries listed in a text file. This format requires three execution parameters:

- **-e delete**
- **-e rdn='pattern'**
- **-e object='file'**

Example A.13. RDN-Based Delete Operations

```
# ldcIt -b "ou=people,dc=example,dc=com" -D "cn=Directory Manager" -w secret -e
delete,rdn='uid:[INCRNNOLOOP(0;99999;5)]',object="inet.txt" -l 32 -v -q
```

The last delete operation format is much like the random delete format, only it moves sequentially through the given range, rather than randomly:

- -e delete,incr
- -r and -R for the range bounds
- -f for the filter to match the entries

Example A.14. Sequential Delete Operations

```
# ldclt -b "ou=people,dc=example,dc=com" -D "cn=Directory Manager" -w secret -e delete,incr -r0
-R999999 -f "uid=XXXXXX" -l 32 -v -q
```

A.1.5.7. Bind Operations

By default, each **ldclt** thread binds once to the server and then runs all of its operations in a single session. The **-e bindeach** can be used with any other operation to instruct the **ldclt** tool to bind for each operation and then unbind before initiating the next operation.

```
-e add,bindeach ...
```

To test only bind and unbind operations, use the **-e bindeach,bindonly** execution parameters and no other operation information. For example:

```
# ldclt -h localhost -p 389 -b "ou=people,dc=example,dc=com" -e bindeach,bindonly -e bind_info
```

The bind operation can specify a single user to use for testing by using the **-D** and **-w** user name–password pair in the connection parameters.



NOTE

Use the **-e close** option with the bind parameters to test the affect that dropping connections has on the Directory Server, instead of unbinding cleanly.

Example A.15. Bind Only and Close Tests

```
# ldclt -h localhost -p 389 -D "cn=Directory Manager" -w secret -e bindeach,bindonly,close
```

There are also execution parameters which can be used to select a random bind identity from a given file (**randombinddnfromfile**) or using a DN selected randomly from within a range (**-e randombinddn,randombinddnlow=X,randombinddnhigh=Y**).

Example A.16. Random Binds from Identities in a File

```
# ldclt -h localhost -p 389 -e bindeach,bindonly -e randombinddnfromfile=/tmp/testbind.txt
```

Binding with a random identity is useful if identities have been added from a generated LDIF or using **-e add**, where the accounts were added in a range. The **ldcrt** tool can autogenerate values using X as a variable and incrementing through the specified range.

Example A.17. Random Binds from Random Base DN

```
# ldcrt -h localhost -p 389 -e bindeach,bindonly -D "uid=XXXXXX,dc=example,dc=com" -w testXXXXX -e randombinddn,randombinddnlow=0,randombinddnhigh=99999
```

A.1.5.8. Running Operations on Random Base DNs

Any operation can be run against randomly-selected base DNs. The trio of **randombase** parameters set the range of organizational units to select from. A variable in the **-b** base entry sets the format of the base DN.

```
-b "ou=DeptXXX,dc=example,dc=com" -e randombase,randombaselow=0,randombasehigh=999 ...
```

A.1.5.9. TLS Authentication

Every operation can be run over TLS to test secure authentication and performance for secure connections. There are two parameters required for TLS authentication.

- The connection parameters, **-Z**, which gives the path to the security databases for the Directory Server
- The execution parameters, **cltcertname**, **keydbfile**, and **keydbpin**, which contains the information that the server will prompt to access the TLS databases

For example, this runs bind tests over TLS:

```
# ldcrt -h host -p port -e bindeach,bindonly -Z certPath -e cltcertname=certName,keydbfile=filename,keydbpin=password
```

A.1.5.10. Abandon Operations

The **-e abandon** parameter opens and then cancels operations on the server. This can be run by itself or with other types of operations (like **-e add** or **-e esearch**).

```
# ldcrt -e abandon -h localhost -p 389 -D "cn=Directory Manager" -w secret -v -q -b "ou=people,dc=example,dc=com"
```

A.2. RSEARCH (SEARCH STRESS TESTS)

The **rsearch** utility opens multiple threads that perform the same operation, quickly and repeatedly, in a loop against the specified Directory Server instance, according to the parameters set in the command.

At its simplest, **rsearch** emulates multiple client connections for search operations. With additional options, **rsearch** can be expanded to perform compare, modify, delete, and bind/unbind operations along with search operations.

The tool also tracks the performance of the operations and outputs a running stream of averaged results.



NOTE

The results of **rsearch** tests naturally depend on the performance of the Directory Server and its host machine. Optimize the configuration of the Directory Server and machine first through performance tuning (as in the *Red Hat Directory Server Performance Tuning Guide*).

The **rsearch** utility is located in the **/usr/bin** directory.

A.2.1. Syntax

```
rsearch-Dbind_dn-wpassword-ssuffix-ffilter-hhost-pport-Sscope-b-u-L-N-v-y-q-l-m-M-d-c-ifile_for_filters-BDN_or_uid_file-Aattributes-afile_of_attributes-n-osearch_time_limits-jsample_interval-tthreads-Ttimelimit-V-Cnumber_of_samples-Rreconnect_interval-x-Wpassword-Utext-\? or -H
```

A.2.2. Options

Table A.5. rsearch Options

Option	Description
-A attributes	Contains a list of attributes to be used with the search request. This cannot be used with -a .
-a file_of_attributes	Points to a file which contains a list of attributes to be used with the search request. Each attribute must be on a separate line in the file. For example: [literal,subs="+quotes,verbatim"] attr1 attr2 This cannot be used with -A .
-B DN_or_uid_file	Contains a list of either DNs or UIDs which are used to bind to the server. For DNs, each entry has two lines, one for the DN and one for the UID (which is used as the default password): [literal,subs="+quotes,verbatim"] DN: dn UID: uid The UID files simple has one UID per line: [literal,subs="+quotes,verbatim"] UID: uid1 UID: uid2
-b	Tells the utility to bind before every operation.
-C sample_numbers	Gives the number of samples to take and then exits the utility.

Option	Description
-c	Specifies a compare operation. If this is used, then the -B option must be used.
-D <i>bind_dn</i>	Gives the bind DN for the rsearch utility to use to connect to the server; if no other identity is supplied in a DN file (-B -x), this is the identity used to run tests.
-d	Specifies a delete operation. If this is used, then the -B option must be used.
-f <i>filter</i>	Contains the search filter to be used with search operations.
-h <i>host</i>	Gives the host name of the LDAP server to connect to. The default, if not given, is localhost.
-i <i>file</i>	<p>Refers to a file that contains the names to be appended to the search filter passed with the -f option. The name file is a list, with each name on a separate line. For example:</p> <pre>[literal,subs="+quotes,verbatim"] joe jane</pre> <p>A filter option that can be used with this file is, for example, -f "uid=%s", which results in filters of both "uid=joe" and "uid=jane" randomly being used.</p>
-j <i>sample_interval</i>	Specifies an interval, in seconds, to wait before collecting a sample.
-L	Sets the connection to linger. The connection is discarded when the utility closes.
-l	Logs the utility output.
-M	Specifies a modify operation for an indexed attribute (telephonenumber). This requires the -B option.
-m	Specifies a modify operation for an unindexed attribute (description). This requires the -B option.
-N	Specifies that the tool will only bind to the server, without running any other operation.
-n	Reserved for future use.

Option	Description
-o <i>search_time_limit</i>	Gives the time limit, in seconds, to use for search operations.
-p <i>port</i>	Gives the port to use to connect to the Directory Server instance. If this is not used, the default is 389.
-q	Runs the tool quietly.
-R <i>reconnect_interval</i>	Tells the utility to drop the connection to server and reconnect after the specified number of searches (<i>reconnect_interval</i>).
-S <i>scope</i>	Sets the search scope. The allowed values are 0, 1, and 2, corresponding to one-level, base, and subtree, respectively. The default is 2.
-s <i>suffix</i>	Gives the suffix in the Directory Server against which to run all of the tests.
-T <i>timelimit</i>	Sets a total time limit for the rsearch tests. Once the utility hits that limit, the tool closes.
-t <i>threads</i>	Sets the number of threads for the utility to open. The default is 1.
-U	Passes a filter to use with the bind file. If -x is not used, this option is ignored. The default value is ' (uid=%s) '.
-u	Tells the utility <i>not</i> to unbind from the server, but simply to close the connection.
-V	Shows the running averages of the rsearch results.
-v	Runs the command in verbose mode.
-W	Gives the password to use to bind with identities in the -B file. If this is not given, the default is the UID value.
-x	Tells the utility to use the contents of the -B file for binding. If this is not used, than the -B option is ignored.
-y	Runs the command with no delay between tests.

Option	Description
-\? or -H	Prints the usage for the tool.

A.2.3. Usage Scenarios

The **rsearch** utility can be used to measure the performance of any LDAP operation. The following examples show how to use **rsearch** for a variety of common test scenarios.



NOTE

Even though **rsearch** requires arguments for search parameters like filter and scope, these arguments can be left empty to perform tests for other kinds of LDAP operations.

For example:

```
# rsearch -D "cn=Directory Manager" -w secret -s "" -f ""
```

A.2.3.1. Allowed Configuration Files

Most of the time, the **rsearch** tool uses the information passed in the command line to connect to the server. The **rsearch** tool can accept two different configuration files to use in place of the passed arguments:

- A *DN or UID* file, which contains a list of either UIDs or both DNs and UIDs. The DN/UID file allows **rsearch** to connect using multiple, randomly-selected bind identities. Any operation test can be combined with a bind/unbind test.



WARNING

Random bind identities should not be used with a delete test because the command may attempt to bind with an identity in the DN/UID file that has already been deleted from the directory.

DN/UID files are used with the **-B** option to pass the file and then an operation option (**-c**, **-d**, **-m**, or **-x**).

- A *name* file, which contains a list of names to use as part of the given LDAP filters. The filter in the file can be more complex than the ones specified in the **-f** option. The filter file can be used to run a number of different search tests. For example, having only a few filters means that the tool will begin retrieving results from cache, while using invalid filter can test search failures. It can also test filter performance, such as exact matches, complex filters, or attribute searches. When using a filter file, the **-f** option must be passed with a placeholder value. The placeholder can be used to replace only an attribute value, such as **cn=%s**, which tells the command to pull the attribute value variable from the filter file. The placeholder can also replace the filter itself (**f "%s"**) to supply randomly-selected filters from the file.

The **-i** option pass the name file to use for the search filters. Every line in the file is appended to whatever filter is given with the **-f** option. There are a couple of different ways that these two options can be used together:

- The simplest scenario leaves the **-f** option empty, so it is just a placeholder. In this case, the filters are taken directly from the file passed with the **-i** option.
- Alternatively, the entries in the file could simply be a list of names, and a partial filter can be given for the **-f** option. For example, the name file could have a list of UIDs (jsmith, bjensen, amorrow) and the **-f** filter could be "**uid=**". **rsearch** automatically appends the name to complete the search filter.

A.2.3.2. Results from rsearch

Periodically (every ten seconds by default), **rsearch** returns the current running average for the operations run by the script.

The results first show the number of operations performed *within that interval*. The two ratios in the parenthesis show the total number of operations per second and then the amount of time, in milliseconds, spent on each operation (1 second divided by the total number of operations, multiplied by 1000).

```
date timestamp - Rate: num_ops/thr (ops/sec = num ms/op), total: ops (number thr)
```

For example:

```
# rsearch -D "cn=Directory Manager" -w password -s "ou=people,dc=example,dc=com" -f "objectclass=%s" -i /home/filter.txt
rsearch: 1 threads launched.
```

```
20100209 20:20:40 - Rate: 65961.00/thr (6596.10/sec = 0.1516ms/op), total: 65961 (1 thr)
```

A.2.3.3. Search Testing

The core usage of **rsearch** is search testing. Measuring search performance can be done using only the required arguments with **rsearch**, without any optional arguments:

```
# rsearch -D bind_dn -w password -s suffix -f filter
```

Options can be used to measure specific performance or use a specific environment.

Search filters (in the command line or through a file with the **-i** file) can test different kinds of indexed attributes:

- Filters without wildcards show the performance for exact matches
- Filters with wildcards give performance for substring indexes
- Filters with operators (=, >=, <=, ~=) show the performance for approximate indexes

Example A.18. Basic Search

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "sn=smith"
```

A basic search (which covers caching, since there is only one filter given and multiple search operations) uses the following arguments:

- **-D**, which gives the bind identity
- **-w**, which gives the bind password
- **-s**, which gives the search target (scope)
- **-f**, which gives the search filter

Example A.19. Searches for Specific Attributes

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "sn=%s" -i /home/filter.txt -A givenname,mail,uid
```

Along with the required arguments, this command searches for three specific attributes in the entries, using the **-A** option.

The **-i filter_file** option is required if you use the **%s** variable in the **-f filter** option.

A.2.3.4. Authentication Testing

The **rsearch** utility uses the user DN and password in the (required) **-D** and **-w** arguments to bind to the server. To test authentication performance, these credentials can be left blank, can be passed a list of credentials that are randomly selected, or be set to a special user, like the Directory Manager.

Example A.20. Anonymous Binds

```
# rsearch -D "" -w "" -s "dc=example,dc=com" -f "sn=%s" -i /home/filter.txt
```

The **-D** and **-w** arguments have empty values, so the tool does not have any bind credentials to use to connect to the server. This initiates an anonymous bind.

Example A.21. Random User Authentication

```
# rsearch -D "" -w "" -s "dc=example,dc=com" -f "sn=%s" -i /home/filter.txt -B /home/uids.txt -x
```

Rather than using the credentials in the **-D** and **-w** arguments, the **rsearch** tool can be instructed to pull random bind identities from a list of given UIDs or DNs. This requires two options:

- **-B** points to a file with a list of bind identities. For a UID file, this is simply a list of UIDs, one per line:

```
UID: uid1
UID: uid2
...
...
```

For DNs, each entry has two lines, one for the DN and one for the UID (which is used as the default password):

DN: dn
UID: uid
...

- **-x** forces the tool to use the file from the **-B** argument.

For DNs, the tool uses the DN line for the DN and the UID line as the password. The **-U** option tells the tool to use an attribute other than the UID as the entry naming attribute and **-W** passes a different password (which, by default, is the UID).

```
# rsearch -D "" -w "" -s "dc=example,dc=com" -f "sn=%s" -i /home/filter.txt -B /home/uids.txt -x -U "(cn=*)" -W newpassword
```

A.2.3.5. Modify Operation Testing

rsearch can be used to measure the performance of modify operations on two kinds of attributes: indexed and unindexed. The modify operation is signaled by using either the **-M** or the **-m** option. A list of entries to run modify operations against is passed using the **-B** option.



NOTE

Running a modify operation requires a DN file, which has the format:

DN: dn1
UID: uid1

DN: dn2
UID: uid2
...

Using the **-b** option measures the rate of each set of bind-modify operations. If the **-b** option is not used, then there is only one bind operation, and the test shows the average of all modify operations that are run.

Example A.22. Modify Operations on Unindexed Attributes

```
# rsearch -D "cn=test user,cn=config" -w secret -s "" -f "" -m -B /home/dns.txt -v
```

Modify operations against *unindexed* attributes are done by using the **-m** option. The command performs modify operations on the **description** attribute for each entry selected from the DN file.

The test will run successfully even if the **description** attribute is indexed, so make sure that the attribute is not indexed before running the test.

Example A.23. Modify Operations on Indexed Attributes

```
# rsearch -D "cn=test user,cn=config" -w secret -s "" -f "" -M -B /home/dns.txt -v
```

Modify operations against *indexed* attributes are done by using the **-M** option. The command performs modify operations on the **telephoneNumber** attribute for each entry selected from the DN file.

The test will run successfully even if the **telephoneNumber** attribute is not indexed, so make sure that the attribute is indexed before running the test.

A.2.3.6. Compare Operation Testing

The **ldapcompare** operation can be tested using **rsearch** by passing the **-c** option. The tool runs compare operations against the UID attribute, based on the list of UIDs passed in the **-B** option.



NOTE

Running a compare operation requires a DN file, which has the format:

DN: dn1
UID: uid1

DN: dn2
UID: uid2
...

Example A.24. Compare Operations

```
# rsearch -D "cn=test user,cn=config" -w secret -s "" -f "" -c -B /home/dns.txt -v
```

The **-c** argument tells the command to perform compare operations. This is required. Two other arguments are useful for measuring the performance of compare operations:

- **-B** (without the **-x**), which provides a list of entries that the server can run compare operations for.
- **-v**, which runs **rsearch** in verbose mode and prints the results of each bind attempt and compare operation.

A.2.3.7. Delete Operation Testing

Only one option is required with the delete performance testing: **-d**, which tells the command to run delete operations. As with other operations, the **-B** argument can be used to pass a file which contains a list of entries to be randomly selected and deleted.



NOTE

*Do not use the **-B -x** option pair with delete operations, because the command may attempt to bind to the server with an identity which has already been deleted.*

Example A.25. Delete Operations

```
# rsearch -D "cn=test user,cn=config" -w secret -s "" -f "" -d -B /home/dns.txt
```

If the **-B** argument is used to supply a list of entries available to delete, then it must be a DN file, which has the format:

```
DN: dn1
UID: uid1
```

```
DN: dn2
UID: uid2
...
...
```

A.2.3.8. Changing Time Limits

As with many performance tests, **rsearch** has several time-based metrics:

- The period that operations are run for gathering one round of statistics (by default, ten seconds)
- How long the tool runs (by default, indefinitely)
- How long the tool maintains a connection to the server (by default, indefinitely)

All three time limits can be reset.

Example A.26. Setting the Operations Interval

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "cn=%s" -i
/home/filter.txt -b -j 20
```

The **rsearch** tool prints the results for the operations performed in the immediate interval. The default interval is ten (10) seconds, so every line in the output represents the statistics for the operations run in the preceding ten second. This interval can be changed using the **-j** option.

This resets the test interval to 20 seconds.

Example A.27. Setting the Test Time Limit

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "cn=%s" -i
/home/filter.txt -b -T 600
```

```
...
```

```
20100210 18:36:21 - Rate: 68561.00/thr (6856.10/sec = 0.1459ms/op), total: 68561 (1 thr)
20100210 18:36:31 - Rate: 78016.00/thr (7801.60/sec = 0.1282ms/op), total: 78016 (1 thr)
Final Average rate: 7328.85/sec = 0.1364msec/op, total: 78016
```

Normally, the command runs indefinitely, until the command is interrupted. The **-T** option sets a time limit (in seconds) for the test to run and then exit cleanly. When the tool exits, it prints a final summary of the averages of all test run intervals.

Example A.28. Setting a Reconnect Interval

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "cn=%s" -i
/home/filter.txt -b -R 30
```

The tool usually opens one connection to the server. The reconnect option, **-R**, sets a time interval for the tool to reconnect to the Directory Server.

A.2.3.9. Bind Testing with Any Operation

Bind and unbind rates can be checked with any operation (search, modify, delete, compare) which is measured by **rsearch**. This requires one option, **-b**, which tells the tool to bind to the server with every operation.

Two other attributes can be used with bind testing: **-L** (which sets the tool to linger) and **-N** (which tells the tool to bind and unbind without performing any other operations).

Example A.29. Binding and Unbinding with Every Operation

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "cn=%s" -i
/home/filter.txt -b -L
```

Two options are used to initiate bind and unbind operations for every operation performed by **rsearch**:

- **-b** (required)
- **-L** (recommended)

The **-i filter_file** option is required if you use the **%s** variable in the **-f filter** option.

Example A.30. Testing Anonymous Bind Operations

```
# rsearch -D "" -w "" -s "" -f "" -N -b -L
```

To test the anonymous bind rate, simply use the **-b** option and leave the values for the **-D** and **-w** options empty. The **-N** option ensures that the command only attempts bind and unbind operations.

Example A.31. Testing Random Bind Operations

```
# rsearch -D "" -w "" -s "" -f "" -B /home/uids.txt -x -N -b -L
```

As with anonymous bind operations, the required arguments can be left blank. The **-N** option ensures that the command only attempts bind and unbind operations, while the **-B** and **-x** options supply a list of random bind credentials for the command to select from.

Example A.32. Testing Using a Filter with Bind Operations

```
# rsearch -D "" -w "" -s "" -f "" -B /home/uids.txt -x -U "(uid=*son)" -N -b -L
```

Normally, any identity contained in the bind file (UID or DN) can be used for bind testing. The default filter is "**(uid=%s)**", which every identity entry has. To use only a subset of the identities in the file, the **-U** option can be used to pass an alternate filter.

A.2.3.10. Performing Multi-Threaded Testing

Example A.33. Multiple Threads

```
# rsearch -D "cn=test user,cn=config" -w secret -s "dc=example,dc=com" -f "sn=%s" -i /home/filter.txt -t 5
```

By default, **rsearch** opens one thread for operations. The **-t** option allows a multiple threads to be opened.

[3] As with the add operation, the first time that the parent is referenced by the tool, the parent entry is created, but the entry which prompted the add operation is *not* created.

APPENDIX B. REPLICATION AGREEMENT STATUS

In the read-only **nsds5replicaLastUpdateStatus** attribute of each replication agreement, Directory Server displays the latest status of the agreement. The following is a list of possible statuses:

Disabled agreements

If a replication agreement is disabled, the **nsds5replicaLastUpdateStatus** parameter is no longer updated and can display the following status:

- The replication agreement was already disabled when the server started:
 - Error (0) No replication sessions started since server startup
- The agreement was disabled during run time.
 - Error (0) Replica acquired successfully: agreement disabled

General agreement status

{blank}

- The replication agreement was stopped:
 - Error (0) Replica acquired successfully: Protocol stopped
- An incremental update was started:
 - Error (0) Replica acquired successfully: Incremental update started
- An incremental update succeeded:
 - Error (0) Replica acquired successfully: Incremental update succeeded
- Replication succeeded, but the consumer ended the session to be able to get acquired by another supplier:
 - Error (0) Replica acquired successfully: Incremental update succeeded and yielded

Error messages in the ACQUIRING_REPLICA state

During the first part of a replication session, the supplier acquires the consumer, establishes the connection, binds to the consumer, verifies that the consumer is not already updated by another supplier, and performs additional checks. The following error codes can be displayed in this state:

- Failures during establishing a connection with the consumer:
 - Error (*result_code*) Problem connecting to replica - LDAP error: *ldap_error_message*
 - Error (*result_code*) Problem connecting to replica (SSL not enabled) - LDAP error: *ldap_error_message*

The result code and error message indicates the reason why the connection could not be established.

- An internal error occurred on the consumer:

Error (8) :Failed to acquire replica: Internal error occurred on the remote replica

This error is caused by a failure related to the change sequence number (CSN) generator on the consumer. See the consumer log files for further details.

- The identity used to authenticate to the consumer was neither a valid replication bind distinguished name (DN) nor a member of a bind DN group:

Error (3) :Unable to acquire replica: permission denied. The bind dn does not have permission to supply replication updates to the replica. Will retry later.

- No valid replica was defined for the suffix on the consumer:

Error (6) :Unable to acquire replica: there is no replicated area on the consumer server. Replication is aborting.

- Decoding error of the replication control sent to the consumer:

Error (4) :Unable to acquire replica: the consumer was unable to decode the startReplicationRequest extended operation sent by the supplier. Replication is aborting.

- The replica is currently updated by a different supplier:

Error (1) :Unable to acquire replica: the replica is currently being updated by another supplier.

- The supplier and consumer use the same replica ID:

Error (11) :Unable to aquire replica: the replica has the same Replica ID as this one. Replication is aborting.

The supplier or the consumer is incorrectly configured. Set a unique replica ID in the replication configuration to fix the problem.

- The supplier was set into **backoff** mode:

Error (14) :Unable to acquire replica: the replica instructed us to go into backoff mode. Will retry later.

This state is only displayed when a custom replication hook is implemented.

- Decoding errors of the replication control received from the consumer:

Error (extop_result) :Unable to acquire replica

Error (4) Unable to parse the response to the startReplication extended operation. Replication is aborting.

Error (16) Unable to receive the response for a startReplication extended operation to consumer. Will retry later.

Error (0) Unable to obtain current CSN. " "Replication is aborting.

Error messages in the **SENDING_UPDATES** state

After a replica was successfully acquired, the session starts sending updates. In this state, the following messages can be displayed in the respective steps:

- a. Examining the replica update vector (RUVE):

- The replica has no update vector configured or replication was not enabled on the consumer:

Error (19) : Replica is not initialized

- The consumer was not initialized using the same database generation as the supplier:

Error (19) : Replica has different database generation ID, remote replica may need to be initialized

To fix the problem, initialize either the supplier or the consumer.

- b. Updating the change state number (CSN) generator:

- The time difference between the local and the remove server is too big:

Error (2) : fatal error - too much time skew between replicas

- Directory Server failed to update the CSN generator:

Error (2) : fatal internal error updating the CSN generator

- c. Initial changelog positioning:

- General error in case that the changelog cannot be processed:

Error (15) : Unexpected format encountered in changelog database

This error is logged, for example, if the path to the changelog file does not exist.

- Parsing an entry in the changelog failed:

Error (15) : Unexpected format encountered in changelog database

- Errors related to the database layer of the changelog:

Error (15) : Changelog database was in an incorrect state

Error (15) : Incorrect dbversion found in changelog database

Error (15) : Changelog database error was encountered

For further details, see the **/var/log/dirsrv/slappd-*instance_name*/errors** log file.

- Directory Server failed to allocate memory:

Error (15) : changelog memory allocation error occurred

This error is logged, for example, if the changelog buffer or changelog iterator failed to allocate memory.

- The supplier is ahead of the consumer and wants to send updates, but cannot find the starting point in the changelog:

Error (15) : Data required to update replica has been purged from the changelog. "The replica must be reinitialized.

Error (15) : Changelog data is missing

Directory Server treats these errors as fatal, but they can be resolved if the consumer receives the updates from a different supplier. In this case, it is treated as transient.

d. Sending the next update:

- Creating a result thread failed:

Error (*result_code*) : Failed to create result thread

The result code indicates the reason why the thread was not created.

- General error in case that the changelog cannot be processed:

Error (15) : Invalid parameter passed to cl5GetNextOperationToReplay

This error is logged, for example, if the path to the changelog file does not exist.

- A database error occurred while reading the change log:

Error (15) : Database error occurred while getting the next operation to replay

This event is logged, for example, if Directory Server access a locked database page.

- Directory Server ran out The creation :

Error (15) : Memory allocation error occurred (cl5GetNextOperationToReplay)

e. Sub-entry update:

- The creation of the **replica keep alive** entry failed:

Error (-1) : Agreement is corrupted: missing suffix

General status in the **SEND_UPDATES** state:

- A non-fatal error occurred on the local server while processing the changelog:

Error (18) : Incremental update transient error. Backing off, will retry update later.

See the `/var/log/dirsrv/slapd-instance_name/errors` file for further details.

- A replication connection was disconnected after the connection was established:

Error (16) : Incremental update connection error. Backing off, will retry update later.

- A timeout appeared on an existing replication connection:

Error (17) : Incremental update timeout error. Backing off, will retry update later.

The replication automatically tries to resume later.

GLOSSARY

access control instruction (ACI)

An instruction that grants or denies permissions to entries in the directory.

access control list (ACL)

The mechanism for controlling access to your directory.

access rights

In the context of access control, specify the level of access granted or denied. Access rights are related to the type of operation that can be performed on the directory. The following rights can be granted or denied: read, write, add, delete, search, compare, selfwrite, proxy and all.

account inactivation

Disables a user account, group of accounts, or an entire domain so that all authentication attempts are automatically rejected.

All IDs Threshold

A size limit which is globally applied to every index key managed by the server. When the size of an individual ID list reaches this limit, the server replaces that ID list with an All IDs token.

All IDs token

A mechanism which causes the server to assume that all directory entries match the index key. In effect, the All IDs token causes the server to behave as if no index was available for the search request.

anonymous access

When granted, allows anyone to access directory information without providing credentials, and regardless of the conditions of the bind.

approximate index

Allows for efficient approximate or "sounds-like" searches.

attribute

Holds descriptive information about an entry. Attributes have a label and a value. Each attribute also follows a standard syntax for the type of information that can be stored as the attribute value.

attribute list

A list of required and optional attributes for a given entry type or object class.

authenticating directory server

In pass-through authentication (PTA), the authenticating Directory Server is the Directory Server that contains the authentication credentials of the requesting client. The PTA-enabled host sends PTA requests it receives from clients to the host.

authentication certificate

Digital file that is not transferable and not forgeable and is issued by a third party. Authentication certificates are sent from server to client or client to server in order to verify and authenticate the other party.

base DN

Base distinguished name. A search operation is performed on the base DN, the DN of the entry and all entries below it in the directory tree.

base distinguished name (bind DN)

Distinguished name used to authenticate to Directory Server when performing an operation.

bind distinguished name (bind rule)

In the context of access control, the bind rule specifies the credentials and conditions that a particular user or client must satisfy in order to get access to directory information.

branch entry

An entry that represents the top of a subtree in the directory.

browser

Software, such as Mozilla Firefox, used to request and view World Wide Web material stored as HTML files. The browser uses the HTTP protocol to communicate with the host server.

browsing index

Speeds up the display of entries in the Directory Server Console. Browsing indexes can be created on any branch point in the directory tree to improve display performance.

cascading replication

In a cascading replication scenario, one server, often called the hub supplier, acts both as a consumer and a supplier for a particular replica. It holds a read-only replica and maintains a changelog. It receives updates from the supplier server that holds the supplier copy of the data and in turn supplies those updates to the consumer.

certificate

A collection of data that associates the public keys of a network user with their DN in the directory. The certificate is stored in the directory as user object attributes.

Certificate Authority

Company or organization that sells and issues authentication certificates. You may purchase an authentication certificate from a Certification Authority that you trust. Also known as a CA.

CGI

Common Gateway Interface. An interface for external programs to communicate with the HTTP server. Programs written to use CGI are called CGI programs or CGI scripts and can be written in many of the common programming languages. CGI programs handle forms or perform output parsing that is not done by the server itself.

chaining

A method for relaying requests to another server. Results for the request are collected, compiled, and then returned to the client.

changelog: A changelog is a record that describes the modifications that have occurred on a replica. The supplier server then replays these modifications on the replicas stored on replica servers or on other suppliers, in the case of multi-suppliers replication.

character type

Distinguishes alphabetic characters from numeric or other characters and the mapping of upper-case to lower-case letters.

ciphertext

Encrypted information that cannot be read by anyone without the proper key to decrypt the information.

class definition

Specifies the information needed to create an instance of a particular object and determines how the object works in relation to other objects in the directory.

class of serviceclassic CoS

A classic CoS identifies the template entry by both its DN and the value of one of the target entry's attributes.

clientcode page

An internal table used by a locale in the context of the internationalization plug-in that the operating system uses to relate keyboard keys to character font screen displays.

collation order

Provides language and cultural-specific information about how the characters of a given language are to be sorted. This information might include the sequence of letters in the alphabet or how to compare letters with accents to letters without accents.

consumer

Server containing replicated directory trees or subtrees from a supplier server.

consumer server

In the context of replication, a server that holds a replica that is copied from a different server is called a consumer for that replica.

CoS

A method for sharing attributes between entries in a way that is invisible to applications.

CoS definition entry

Identifies the type of CoS you are using. It is stored as an LDAP subentry below the branch it affects.

CoS template entry

Contains a list of the shared attribute values.

daemon

A background process on a Unix machine that is responsible for a particular system task. Daemon processes do not need human intervention to continue functioning.

DAP

Directory Access Protocol. The ISO X.500 standard protocol that provides client access to the directory.

database link

An implementation of chaining. The database link behaves like a database but has no persistent storage. Instead, it points to data stored remotely.

default index

One of a set of default indexes created per database instance. Default indexes can be modified, although care should be taken before removing them, as certain plug-ins may depend on them.

directory tree

The logical representation of the information stored in the directory. It mirrors the tree model used by most filesystems, with the tree's root point appearing at the top of the hierarchy. Also known as DIT.

Directory Manager

The privileged database administrator, comparable to the root user in UNIX. Access control does not apply to the Directory Manager.

directory service

A database application designed to manage descriptive, attribute-based information about people and resources within an organization.

distinguished name

String representation of an entry's name and location in an LDAP directory.

DNS

Domain Name System. The system used by machines on a network to associate standard IP addresses (such as 198.93.93.10) with host names (such as **www.example.com**). Machines normally get the IP address for a host name from a DNS server, or they look it up in tables maintained on their systems.

DNS alias

A DNS alias is a host name that the DNS server knows points to a different host specifically a DNS CNAME record. Machines always have one real name, but they can have one or more aliases. For example, an alias such as **www.yourdomain.domain** might point to a real machine called **realthing.yourdomain.domain** where the server currently exists.

entry

A group of lines in the LDIF file that contains information about an object.

entry distribution

Method of distributing directory entries across more than one server in order to scale to support large numbers of entries.

entry ID list

Each index that the directory uses is composed of a table of index keys and matching entry ID lists. The entry ID list is used by the directory to build a list of candidate entries that may match the client application's search request.

equality index

Allows you to search efficiently for entries containing a specific attribute value.

file extension

The section of a filename after the period (.) that typically defines the type of file (for example, .GIF and .HTML). In the filename **index.html** the file extension is **html**.

file type

The format of a given file. For example, graphics files are often saved in GIF format, while a text file is usually saved as ASCII text format. File types are usually identified by the file extension (for example, .GIF or .HTML).

filter

A constraint applied to a directory query that restricts the information returned.

filtered role: Allows you to assign entries to the role depending upon the attribute contained by each entry. You do this by specifying an LDAP filter. Entries that match the filter are said to possess the role.

general access

When granted, indicates that all authenticated users can access directory information.

GSS-API

Generic Security Services. The generic access protocol that is the native way for UNIX-based systems to access and authenticate Kerberos services; also supports session encryption.

host name

A name for a machine in the form machine.domain.dom, which is translated into an IP address. For example, **www.example.com** is the machine **www** in the subdomain **example** and **com** domain.

HTML

Hypertext Markup Language. The formatting language used for documents on the World Wide Web. HTML files are plain text files with formatting codes that tell browsers such as the Mozilla Firefox how to display text, position graphics, and form items and to display links to other pages.

HTTP

Hypertext Transfer Protocol. The method for exchanging information between HTTP servers and clients.

HTTPD

An abbreviation for the HTTP daemon or service, a program that serves information using the HTTP protocol. The daemon or service is often called an httpd.

HTTPS

A secure version of HTTP, implemented using the Secure Sockets Layer, SSL.

hub

In the context of replication, a server that holds a replica that is copied from a different server, and, in turn, replicates it to a third server.

IID list scan limit

A size limit which is globally applied to any indexed search operation. When the size of an individual ID list reaches this limit, the server replaces that ID list with an all IDs token.

index key

Each index that the directory uses is composed of a table of index keys and matching entry ID lists.

indirect CoS

An indirect CoS identifies the template entry using the value of one of the target entry's attributes.

international index

Speeds up searches for information in international directories.

IP address

A set of numbers, separated by dots, that specifies the actual location of a machine on the Internet (for example, 198.93.93.10).

ISO

International Standards Organization.

knowledge reference

Pointers to directory information stored in different databases.

LDAP

Lightweight Directory Access Protocol. Directory service protocol designed to run over TCP/IP and across multiple platforms.

LDAPv3

Version 3 of the LDAP protocol, upon which Directory Server bases its schema format.

LDAP client

Software used to request and view LDAP entries from an LDAP Directory Server.

LDAP URL

Provides the means of locating Directory Servers using DNS and then completing the query using LDAP. A sample LDAP URL is **ldap://ldap.example.com**.

LDBM database

A high-performance, disk-based database consisting of a set of large files that contain all of the data assigned to it. The primary data store in Directory Server.

LDIF

LDAP Data Interchange Format. Format used to represent Directory Server entries in text form.

leaf entry

An entry under which there are no other entries. A leaf entry cannot be a branch point in a directory tree.

locale

Identifies the collation order, character type, monetary format and time / date format used to present data for users of a specific region, culture, and custom. This includes information on how data of a given language is interpreted, stored, or collated. The locale also indicates which code page should be used to represent a given language.

managed object

A standard value which the SNMP agent can access and send to the NMS. Each managed object is identified with an official name and a numeric identifier expressed in dot-notation.

managed role

Allows creation of an explicit enumerated list of members.

mapping tree

A data structure that associates the names of suffixes (subtrees) with databases.

matching rule

Provides guidelines for how the server compares strings during a search operation. In an international search, the matching rule tells the server what collation order and operator to use.

MD5

A message digest algorithm by RSA Data Security, Inc., which can be used to produce a short digest of data that is unique with high probability and is mathematically extremely hard to produce; a piece of data that will produce the same message digest.

MD5 signature

A message digest produced by the MD5 algorithm.

MIB

Management Information Base. All data, or any portion thereof, associated with the SNMP network. We can think of the MIB as a database which contains the definitions of all SNMP managed objects. The MIB has a tree-like hierarchy, where the top level contains the most general information about the network and lower levels deal with specific, separate network areas.

MIB namespace

Management Information Base namespace. The means for directory data to be named and referenced. Also called the directory tree.

monetary format

Specifies the monetary symbol used by specific region, whether the symbol goes before or after its value, and how monetary units are represented.

multi-supplier replication

An advanced replication scenario in which two servers each hold a copy of the same read-write replica. Each server maintains a changelog for the replica. Modifications made on one server are automatically replicated to the other server. In case of conflict, a time stamp is used to determine which server holds the most recent version.

multiplexor

The server containing the database link that communicates with the remote server.

n + 1 directory problem

The problem of managing multiple instances of the same information in different directories, resulting in increased hardware and personnel costs.

name collisions

Multiple entries with the same distinguished name.

nested role

Allows the creation of roles that contain other roles.

network management application

Network Management Station component that graphically displays information about SNMP managed devices, such as which device is up or down and which and how many error messages were received.

NIS

Network Information Service. A system of programs and data files that Unix machines use to collect, collate, and share specific information about machines, users, filesystems, and network parameters throughout a network of computers.

NMS

Powerful workstation with one or more network management applications installed.

ns-slapd

Red Hat's LDAP Directory Server daemon or service that is responsible for all actions of the Directory Server.

object class

Defines an entry type in the directory by defining which attributes are contained in the entry.

object identifier

A string, usually of decimal numbers, that uniquely identifies a schema element, such as an object class or an attribute, in an object-oriented system. Object identifiers are assigned by ANSI, IETF or similar organizations.

operational attribute

Contains information used internally by the directory to keep track of modifications and subtree properties. Operational attributes are not returned in response to a search unless explicitly requested.

parent access

When granted, indicates that users have access to entries below their own in the directory tree if the bind DN is the parent of the targeted entry.

pass-through subtree

In pass-through authentication, the PTA directory server will pass through bind requests to the authenticating Directory Server from all clients whose DN is contained in this subtree.

password file

A file on Unix machines that stores Unix user login names, passwords, and user ID numbers. It is also known as **/etc/passwd** because of where it is kept.

password policy

A set of rules that governs how passwords are used in a given directory.

permission

In the context of access control, permission states whether access to the directory information is granted or denied and the level of access that is granted or denied.

PDU

Encoded messages which form the basis of data exchanges between SNMP devices.

pointer CoS

A pointer CoS identifies the template entry using the template DN only.

presence index

Allows searches for entries that contain a specific indexed attribute.

protocol

A set of rules that describes how devices on a network exchange information.

protocol data unitproxy authentication

A special form of authentication where the user requesting access to the directory does not bind with its own DN but with a proxy DN.

proxy DN

Used with proxied authorization. The proxy DN is the DN of an entry that has access permissions to the target on which the client-application is attempting to perform an operation.

PTA

Mechanism by which one Directory Server consults another to check bind credentials.

PTA directory server

In pass-through authentication (PTA), the PTA Directory Server is the server that sends (passes through) bind requests it receives to the authenticating Directory Server.

PTA LDAP URL

In pass-through authentication, the URL that defines the authenticating Directory Server, pass-through subtree(s), and optional parameters.

RAM

Random access memory. The physical semiconductor-based memory in a computer. Information stored in RAM is lost when the computer is shut down.

rc.local

A file on Unix machines that describes programs that are run when the machine starts. It is also called **/etc/rc.local** because of its location.

RDN

The name of the actual entry itself, before the entry's ancestors have been appended to the string to form the full distinguished name.

referential integrity

Mechanism that ensures that relationships between related entries are maintained within the directory.

read-only replica

A replica that refers all update operations to read-write replicas. A server can hold any number of read-only replicas.

read-write replica

A replica that contains a writable copy of directory information and can be updated. A server can hold any number of read-write replicas.

replica

A database that participates in replication.

replication

Act of copying directory trees or subtrees from supplier servers to replica servers.

replication agreement

Set of configuration parameters that are stored on the supplier server and identify the databases to replicate, the replica servers to which the data is pushed, the times during which replication can occur, the DN and credentials used by the supplier to bind to the consumer, and how the connection is secured.

RFC

Request for Comments. Procedures or standards documents submitted to the Internet community. People can send comments on the technologies before they become accepted standards.

role

An entry grouping mechanism. Each role has *members*, which are the entries that possess the role.

role-based attributes

Attributes that appear on an entry because it possesses a particular role within an associated CoS template.

root

The most privileged user available on Unix machines. The root user has complete access privileges to all files on the machine.

root suffix

The parent of one or more sub suffixes. A directory tree can contain more than one root suffix.

SASL

An authentication framework for clients as they attempt to bind to a directory.

schema

Definitions describing what types of information can be stored as entries in the directory. When information that does not match the schema is stored in the directory, clients attempting to access the directory may be unable to display the proper results.

schema checking

Ensures that entries added or modified in the directory conform to the defined schema. Schema checking is on by default, and users will receive an error if they try to save an entry that does not conform to the schema.

Secure Sockets Layerself access

When granted, indicates that users have access to their own entries if the bind DN matches the targeted entry.

server daemon

The server daemon is a process that, once running, listens for and accepts requests from clients.

server service

A process on Windows that, once running, listens for and accepts requests from clients. It is the SMB server on Windows NT.

Server Selector

Interface that allows you select and configure servers using a browser.

service

A background process on a Windows machine that is responsible for a particular system task. Service processes do not need human intervention to continue functioning.

SIE

Server Instance Entry. The ID assigned to an instance of Directory Server during installation.

single-supplier replication

The most basic replication scenario in which multiple servers, up to four, each hold a copy of the same read-write replicas to replica servers. In a single-supplier replication scenario, the supplier server maintains a changelog.

SIRslapd

LDAP Directory Server daemon or service that is responsible for most functions of a directory except replication.

SNMP

Used to monitor and manage application processes running on the servers by exchanging data about network activity.

SNMP master agent

Software that exchanges information between the various subagents and the NMS.

SNMP subagent

Software that gathers information about the managed device and passes the information to the master agent. Also called a subagent.

SSL

A software library establishing a secure connection between two parties (client and server) used to implement HTTPS, the secure version of HTTP. Also called Secure Sockets Layer.

standard index

index maintained by default.

sub suffix

A branch underneath a root suffix.

substring index

Allows for efficient searching against substrings within entries. Substring indexes are limited to a minimum of two characters for each entry.

suffix

The name of the entry at the top of the directory tree, below which data is stored. Multiple suffixes are possible within the same directory. Each database only has one suffix.

superuser

The most privileged user available on Unix machines. The superuser has complete access privileges to all files on the machine. Also called root.

supplier

Server containing the writable copy of directory trees or subtrees that are replicated to replica servers.

supplier server

In the context of replication, a server that holds a replica that is copied to a different server is called a supplier for that replica.

supplier-initiated replication

Replication configuration where supplier servers replicate directory data to any replica servers.

symmetric encryption

Encryption that uses the same key for both encrypting and decrypting. DES is an example of a symmetric encryption algorithm.

system index

Cannot be deleted or modified as it is essential to Directory Server operations.

target

In the context of access control, the target identifies the directory information to which a particular ACL applies.

target entry

The entries within the scope of a CoS.

TCP/IP

Transmission Control Protocol/Internet Protocol. The main network protocol for the Internet and for enterprise (company) networks.

time/date format

Indicates the customary formatting for times and dates in a specific region.

TLS

The new standard for secure socket layers; a public key based protocol. Also Transport Layer Security.

topology

The way a directory tree is divided among physical servers and how these servers link with one another.

uid

A unique number associated with each user on a Unix system.

URL

Uniform Resource Locator. The addressing system used by the server and the client to request documents. It is often called a location. The format of a URL is *protocol://machine:port/document*. The port number is necessary only on selected servers, and it is often assigned by the server, freeing the user of having to place it in the URL.

virtual list view index

Speeds up the display of entries in the Directory Server Console. Virtual list view indexes can be created on any branch point in the directory tree to improve display performance.

X.500 standard

The set of ISO/ITU-T documents outlining the recommended information model, object classes and attributes used by directory server implementation.

APPENDIX C. REVISION HISTORY

Note that revision numbers relate to the edition of this manual, not to version numbers of Red Hat Directory Server.

11.4-1

Tue Nov 09 2021, Marc Muehlfeld (mmuehlfeld@redhat.com)

- Red Hat Directory Server 11.4 release of this guide.

11.3-1

Tue May 11 2021, Marc Muehlfeld (mmuehlfeld@redhat.com)

- Red Hat Directory Server 11.3 release of this guide.

11.2-1

Tue Nov 03 2020, Marc Muehlfeld (mmuehlfeld@redhat.com)

- Red Hat Directory Server 11.2 release of this guide.

11.1-1

Tue Apr 28 2020, Marc Muehlfeld (mmuehlfeld@redhat.com)

- Red Hat Directory Server 11.1 release of this guide.

11.0-1

Tue Nov 05 2019, Marc Muehlfeld (mmuehlfeld@redhat.com)

- Red Hat Directory Server 11.0 release of this guide.