Sun Java System Directory Server Enterprise Edition 6.3 管理指南

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

文件號碼：820–4819
2008 年 4 月

080424@19860

目錄

前言 23

1 安裝之前 33
管理架構與安裝 33
單一系統與分散式安裝的比較 36
安裝目錄服務控制中心的位置 36
建立伺服器實例的位置 38
Directory Server Enterprise Edition 軟體發行檔 38
Java Enterprise System 發行檔 39
原生修補程式 39
Zip 發行檔 40
Java Enterprise System 發行檔與 Zip 發行檔的比較 40
安裝於 Solaris 區域中 41

第 1 部分 目錄伺服器管理 43

2 目錄伺服器工具 45
目錄伺服器管理簡介 45
決定 DSCC 與指令行的使用時機 45
決定某程序是否可以使用 DSCC 完成 46
較適合使用 DSCC 的情況 46
目錄服務控制中心介面 47
DSCC 的管理使用者 47
▼ 存取 DSCC 47
DSCC 標籤說明 50
DSCC 線上說明 51
目錄伺服器指令行工具 51

目錄伺服器指令的位置 52
設定 dsconf 的環境變數 52
dsadm 與 dsconf 的比較 52
使用 dsadm 與 dsconf 取得說明 53
使用 dsconf 修改配置特性 54
使用 dsconf 設定多重值特性 54
線上手冊 55
舊有工具 55

3 目錄伺服器實例與尾碼 57
建立伺服器實例與尾碼的快速程序 57
建立及刪除目錄伺服器實例 57
▼建立目錄伺服器實例 57
▼刪除目錄伺服器實例 60
啓動、停止與重新啓動目錄伺服器實例 61
▼啓動、停止與重新啓動目錄伺服器 61
建立尾碼 62
▼建立尾碼 62
停用或啓用尾碼 64
▼停用尾碼後再予以啓用 64
設定參照並使尾碼成爲唯讀模式 64
▼設定參照，使尾碼變成唯讀模式 65
刪除尾碼 66
▼刪除尾碼 66
壓縮尾碼 66
▼離線壓縮尾碼 66

4 目錄伺服器配置 67
顯示目錄伺服器實例的配置 67
使用 DSCC 修改配置 68
從指令行修改配置 68
修改 dse.ldif 檔案 68
配置管理使用者 69
▼建立具有超級使用者存取權的管理使用者 69
▼配置目錄管理員 70

保護配置資訊 71
配置 DSCC 71
▼變更共用代理程式容器的連接埠號碼 71
▼重設目錄服務管理員密碼 72
▼延伸 DSCC 階段作業自動逾時延遲 72
配置 DSCC 的容錯移轉 73
DSCC 的疑難排解 74
變更目錄伺服器連接埠號碼 74
▼修改連接埠號碼、啓用連接埠以及停用連接埠 74
配置 DSML 76
▼啓用 DSML-over-HTTP 服務 76
▼停用 DSML-over-HTTP 服務 77
▼配置 DSML 安全性 77
DSML 身份識別對映 78
▼爲 HTTP 標頭定義新的身份識別對映 79
將伺服器設爲唯讀 79
▼啓用或停用伺服器唯讀模式 80
配置記憶體 80
填充快取 80
▼修改資料庫快取 80
▼監視資料庫快取 81
▼監視項目快取 81
▼修改項目快取 82
▼配置堆疊記憶體臨界值 82
設定每個用戶端帳號的資源限制 83
▼配置堆疊記憶體臨界值 84

5 目錄伺服器項目 85
管理項目 85
使用 DSCC 管理項目 86
使用目錄編輯器管理項目 86
使用 `ldapmodify` 與 `ldapdelete` 管理項目 86
▼使用 `ldapmodify` 移動項目或爲其重新命名 93
使用修改 DN 作業的指示與限制 94
設定參照 95

設定預設參照 96
▼設定預設參照 96
設定智慧型參照 96
▼建立及修改智慧型參照 97
檢查有效屬性語法 98
▼開啓自動語法檢查 98
追蹤目錄項目的修改 98
▼關閉項目修改追蹤 99
爲屬性值加密 99
屬性加密與效能 100
屬性加密用法注意事項 101
▼配置屬性加密 102

6 目錄伺服器安全性 105
對目錄伺服器使用 SSL 106
管理憑證 106
▼檢視預設自行簽署的憑證 107
▼管理自行簽署的憑證 107
▼請求 CA 簽署的伺服器憑證 108
▼增加 CA 簽署的伺服器憑證與可信任的 CA 憑證 109
▼更新過期的 CA 簽署伺服器憑證 112
▼匯出及匯入 CA 簽署伺服器憑證 112
配置憑證資料庫密碼 113
▼配置伺服器，讓使用者必須提供憑證密碼 113
爲目錄伺服器備份及復原憑證資料庫 114
配置 SSL 通訊 114
停用非安全通訊 114
▼停用 LDAP 明文連接埠 114
選擇加密密碼 114
▼選擇加密密碼 115
配置憑證層級與認證方法 116
設定目錄伺服器中的 SASL 加密層級 117
▼需要 SASL 加密 118
▼不允許 SASL 加密 118
透過 DIGEST-MD5 的 SASL 認證 118

▼配置 DIGEST-MD5 機制 118
透過 GSSAPI 的 SASL 認證 (僅限 Solaris 作業系統) 121
▼配置 Kerberos 系統 121
▼配置 GSSAPI 機制 121
配置 LDAP 用戶端以使用安全性 123
在用戶端中使用 SASL DIGEST-MD5 123
在用戶端中使用 Kerberos SASL GSSAPI 125
▼在主機上配置 Kerberos V5 125
▼指定 Kerberos 認證的 SASL 選項 125
傳遞式認證 136

7 目錄伺服器存取控制 139
建立、檢視及修改 ACI 139
▼建立、修改及刪除 ACI 139
▼檢視 ACI 屬性值 140
▼檢視根層級的 ACI 140
存取控制用法範例 141
授予匿名存取權 142
授予個人項目的寫入存取權 143
授予特定層級的存取權 144
限制主要角色的存取 145
授予某角色對整個尾碼的完整存取權 145
授予某群組對尾碼的完整存取權 146
授予新增及刪除群組項目的權限 147
允許使用者在群組中加入或移除本身 148
授予對群組或角色的條件式存取權 148
拒絕存取 149
代理授權 150
使用篩選設定目標 151
爲含有逗號的 DN 定義權限 152
檢視有效權限 152
限制取得有效權限控制的存取 152
使用「取得有效權限」控制 153
進階存取控制：使用巨集 ACI 156
巨集 ACI 範例 156

巨集 ACI 語法 158
記錄存取控制資訊 161
▼設定 ACI 記錄 161
使用 TCP 包裝的用戶端主機存取控制 162
▼啓用 TCP 包裝 162
▼停用 TCP 包裝 162

8 目錄伺服器密碼策略 163
密碼策略與工作表 164
密碼策略設定 164
定義密碼策略的工作表 167
管理預設密碼策略 168
密碼策略屬性與 dsconf 伺服器特性之間的關聯 168
▼檢視預設密碼策略設定 169
▼變更預設密碼策略設定 170
管理專用密碼策略 171
套用的密碼策略 171
▼建立密碼策略 172
▼指定密碼策略給個別帳號 174
▼使用角色與 CoS 指定密碼策略 174
▼設定第一次登入密碼策略 176
在 pwdSafeModify 爲 TRUE 時從指令行修改密碼 179
重設過期的密碼 180
▼使用密碼修改延伸作業重設密碼 180
▼密碼過期時允許寬限認證 181
設定帳號特性 182
▼設定帳號的查詢限制 182
▼設定帳號的大小限制 182
▼設定帳號的時間限制 183
▼設定帳號的閒置逾時 183
手動鎖定帳號 184
▼檢查帳號狀態 184
▼停用帳號 184
▼重新啓用帳號 185

9 目錄伺服器備份與復原 187
二進位備份 187
僅備份目錄資料 187
▼備份目錄資料 188
▼備份 dse.ldif 檔案 188
備份檔案系統 189
▼備份檔案系統 189
備份至 LDIF 190
匯出至 LDIF 190
▼將尾碼匯出至 LDIF 190
二進位復原 191
▼復原伺服器 191
復原 dse.ldif 配置檔案 191
▼復原 dse.ldif配置檔案 192
從 LDIF 檔案匯入資料 192
初始化尾碼 193
▼初始化尾碼 193
大量增加、修改及刪除項目 194
▼大量增加、修改及刪除項目 194
復原已複寫的尾碼 195
復原單一主伺服器方案中的供應者 195
復原多重主伺服器方案中的供應者 196
復原集散中心 196
復原專屬用戶 197
復原多重主伺服器方案中的主伺服器 197
▼開始透過指令行接受更新 198
嚴重損壞回復 198
▼製作嚴重損壞回復所需的備份 198
▼復原嚴重損壞回復 199

10 目錄伺服器群組、角色與 CoS 201
關於群組、角色與服務類別 201
管理群組 202
▼建立新的靜態群組 202
▼建立新的動態群組 203

管理角色 203
安全地使用角色 204
從指令行管理角色 204
延伸角色的範圍 206
▼延伸角色的範圍 206
服務類別 207
安全地使用 CoS 207
從指令行管理 CoS 209
建立角色型屬性 215
監視 CoS 外掛程式 216
設定 CoS 記錄 216
維護參照完整性 217
參照完整性的運作方式 217
▼配置參照完整性外掛程式 218

11 目錄伺服器複寫 221
規劃複寫部署 222
配置與管理複寫的建議介面 222
配置複寫的步驟摘要 222
▼配置複寫的步驟摘要 222
啓用專屬用戶上的複寫 224
▼建立用戶複本的尾碼 224
▼啓用用戶複本 225
▼執行進階用戶配置 225
啓用集散中心上的複寫 226
▼建立集散複本的尾碼 226
▼啓用集散複本 226
▼修改集散複本上的變更記錄設定 227
啓用主伺服器複本上的複寫 227
▼建立主伺服器複本的尾碼 227
▼啓用主伺服器複本 227
▼修改主伺服器複本上的變更記錄設定 228
配置複寫管理員 228
使用非預設複寫管理員 228
▼設定非預設複寫管理員 229

▼變更預設複寫管理員密碼 230
建立與變更複寫協議 231
▼建立複寫協議 231
▼變更複寫協議的目標 232
部分複寫 232
部分複寫的注意事項 232
▼配置部分複寫 233
複寫優先權 233
▼配置複寫優先權 233
初始化複本 234
▼從遠端 (供應者) 伺服器初始化複寫的尾碼 234
從 LDIF 初始化複本 235
▼從 LDIF 初始化複寫的尾碼 235
▼匯出複寫的尾碼至 LDIF 236
使用二進位副本初始化複寫的尾碼 237
在串聯複寫中初始化複本 240
▼在串聯複寫中初始化複本 240
編製複寫的尾碼之索引 241
遞增多個項目到大型複寫的尾碼 241
▼將多個項目增加至大型複寫的尾碼 241
複寫與參照完整性 242
經由 SSL 的複寫 242
▼配置 SSL 的複寫作業 242
經由 WAN 的複寫 244
配置網路參數 244
排程複寫活動 245
▼排程複寫活動 246
配置複寫壓縮 246
▼配置複寫壓縮 246
修改複寫拓樸 247
變更複寫管理員 247
管理複寫協議 247
升級或降級複本 248
▼升級或降級複本 249
停用複寫的尾碼 249
▼停用複寫的尾碼 250

保持複寫的尾碼同步化 250
▼強制執行複寫更新 250
將主伺服器複本移至新的機器 251
▼從現有的複寫拓樸中移除主伺服器 251
▼將主伺服器增加至現有的複寫拓樸 251
Directory Server 6.3 之前的版本複寫 252
在 Directory Server 6.3 與 Directory Server 5.1 或 5.2 之間進行複寫 252
使用回溯變更記錄 252
▼啓用回溯變更記錄 253
▼配置回溯變更記錄以記錄指定尾碼的更新 253
▼配置回溯變更記錄以記錄刪除項目的屬性 253
▼修剪回溯變更記錄 254
存取控制與回溯變更記錄 255
取得複寫狀態 255
在 DSCC 中取得複寫狀態 255
透過使用指令行取得複寫狀態 256
解決常見複寫衝突 257
使用 DSCC 解決複寫衝突 257
使用指令行解決複寫衝突 257
解決命名衝突 257
▼重新命名具有多值命名屬性的衝突項目 258
▼重新命名具有單值命名屬性的衝突項目 258
解決孤立項目的衝突 259
解決可能的互通操作問題 260

12 目錄伺服器模式 261
管理模式檢查 261
▼修正模式規範遵循問題 262
關於自訂模式 262
預設目錄伺服器模式 263
物件識別碼 263
命名屬性與物件類別 264
定義新的物件類別時 264
定義新的屬性時 266
建立自訂模式檔案時 266

透過 LDAP 管理屬性類型 267
建立屬性類型 267
▼ 建立屬性類型 268
檢視屬性類型 269
▼ 檢視屬性類型 269
刪除屬性類型 269
▼ 刪除屬性類型 270
透過 LDAP 管理物件類別 270
建立物件類別 270
▼ 建立物件類別 271
檢視物件類別 272
▼ 檢視物件類別 272
刪除物件類別 272
▼ 刪除物件類別 273
延伸目錄伺服器模式 273
使用自訂模式檔案延伸模式 274
▼ 使用自訂模式檔案延伸模式 275
透過 LDAP 延伸模式 275
▼ 透過 LDAP 延伸模式 275
使用模式檔案與複寫延伸模式 276
▼ 使用模式檔案與複寫延伸模式 276
複寫目錄模式 276
限制模式複寫 278
▼ 限制模式複寫 278

13 目錄伺服器編製索引 279
管理索引 279
▼ 列出索引 279
▼ 建立索引 280
▼ 修改索引 280
▼ 產生索引 281
▼ 刪除索引 282
變更索引清單臨界值 282
▼ 變更索引清單臨界值 283
重新編製尾碼的索引 284

管理瀏覽索引 285
用戶端搜尋的瀏覽索引 285
▼ 建立瀏覽索引 285
▼ 增加或修改瀏覽索引項目 285
▼ 重新產生瀏覽索引 287

14 目錄伺服器屬性值唯一性 289
屬性值唯一性簡介 289
執行 uid 與其他屬性的唯一性 290
▼ 執行 uid 屬性的唯一性 290
▼ 執行其他屬性的唯一性 291
對複寫使用唯一性外掛程式 292
單一主伺服器複寫案例 292
多重主伺服器複寫案例 292

15 目錄伺服器記錄 293
記錄分析工具 293
檢視目錄伺服器記錄 293
▼ 追蹤目錄伺服器記錄 294
配置目錄伺服器的記錄 295
▼ 修改記錄配置 295
▼ 啓用稽核記錄 296
手動自動重建目錄伺服器記錄 296
▼ 手動自動重建記錄檔 296

16 目錄伺服器監視 299
設定目錄伺服器的 SNMP 299
▼ 設定 SNMP 299
啓用 Java ES MF 監視 300
▼ 啓用 Java ES MF 監視 300
Java ES MF 監視的疑難排解 301
使用 cn=monitor 監視伺服器 301

第 2 部分　目錄代理伺服器管理 303

17　目錄代理伺服器工具 305
使用目錄代理伺服器的 DSCC 305
▼存取目錄代理伺服器的 DSCC 305
目錄代理伺服器的指令行工具 306
目錄代理伺服器指令的位置 306
設定 dpconf 的環境變數 307
dpadm 與 dpconf 的比較 307
使用 dpconf 設定多重值特性 308
取得有關使用 dpadm 與 dpconf 的說明 309

18　目錄代理伺服器實例 311
使用目錄代理伺服器實例 311
▼建立目錄代理伺服器實例 311
▼尋找目錄代理伺服器實例的狀態 312
▼啓動與停止目錄代理伺服器 312
▼檢視是否需要重新啓動目錄代理伺服器實例 313
▼重新啓動目錄代理伺服器 313
▼刪除目錄代理伺服器實例 313
配置目錄代理伺服器實例 314
▼顯示目錄代理伺服器實例的配置 314
▼修改目錄代理伺服器的配置 315
配置代理伺服器管理員 317
▼配置代理伺服器管理員 318
要求重新啓動伺服器的配置變更 318
備份與復原目錄代理伺服器實例 319
▼備份目錄代理伺服器實例 319
▼復原目錄代理伺服器實例 320

19　LDAP 資料檢視 321
建立 LDAP 資料檢視 321
建立與配置 LDAP 資料來源 321
▼建立 LDAP 資料來源 321

▼配置 LDAP 資料來源 322
建立與配置 LDAP 資料來源池 324
▼建立 LDAP 資料來源池 324
▼配置 LDAP 資料來源池 324
將 LDAP 資料來源附加至資料來源池 325
▼將 LDAP 資料來源附加至資料來源池 325
使用 LDAP 資料檢視 326
▼建立 LDAP 資料檢視 327
▼配置 LDAP 資料檢視 327
使用目錄代理伺服器存取目錄伺服器的配置項目 329
▼透過使用目錄代理伺服器存取目錄伺服器的配置項目 329
重新命名屬性與 DN 330
▼配置屬性重新命名 330
▼配置 DN 重新命名 330
配置檢視排除基底與替代搜尋基底 332
▼手動配置 `excluded-subtrees` 與 `alternate-search-base-dn` 特性 332
建立與配置使用範例的資料檢視 333
預設資料檢視 333
不論請求的目標 DN 爲何，均路由所有請求的資料檢視 334
當子樹狀結構清單儲存在多個資料相同的資料來源中時，路由請求的資料檢視 . 335
▼配置當子樹狀結構清單儲存在多個資料相同的資料來源中時，路由請求的資料檢視 336
當不同的子樹狀結構儲存在不同的資料來源中時，提供單一存取點的資料檢視 . 337
▼配置當不同的子樹狀結構儲存在不同的資料來源中時，提供單一存取點的資料檢視 337
當上層與從屬子樹狀結構儲存在不同的資料來源中時，提供單一存取點的資料檢視 338
▼配置當上層與從屬子樹狀結構儲存在不同的資料來源中時，提供單一存取點的資料檢視 339

20 目錄代理伺服器憑證 341
預設的自行簽署憑證 341
▼檢視預設的自行簽署憑證 341
建立、請求與安裝目錄代理伺服器的憑證 342
▼建立目錄代理伺服器的非預設自行簽署憑證 342
▼請求目錄代理伺服器的 CA 簽署憑證 342

▼安裝目錄代理伺服器的 CA 簽署伺服器憑證 343
更新過期的目錄代理伺服器 CA 簽署憑證 344
▼更新過期的目錄代理伺服器 CA 簽署伺服器憑證 344
列出憑證 344
▼列出伺服器憑證 344
▼列出 CA 憑證 345
將憑證從後端 LDAP 伺服器增加至目錄代理伺服器憑證資料庫 345
▼將憑證從後端目錄伺服器增加至目錄代理伺服器憑證資料庫 346
匯出憑證至後端 LDAP 伺服器 347
▼配置目錄代理伺服器以匯出用戶端憑證至後端 LDAP 伺服器 347
備份與復原目錄代理伺服器憑證資料庫 347
提示輸入存取憑證資料庫的密碼 348
▼提示輸入存取憑證資料庫的密碼 348
▼停用存取憑證資料庫的密碼提示 348

21 目錄代理伺服器負載平衡與用戶端相似性 351
配置負載平衡 351
▼選取負載平衡演算法 351
▼配置負載平衡加權 352
負載平衡的配置範例 353
▼配置比例演算法以進行負載平衡 353
▼配置飽和演算法以進行負載平衡 354
▼配置操作相似性演算法以進行全域帳號封鎖 355
▼配置操作相似性演算法以進行快取最佳化 357
▼配置容錯移轉演算法以進行負載平衡 358
配置目錄代理伺服器執行負載平衡 359
配置用戶端相似性 360
▼配置用戶端相似性 360
用戶端相似性的配置範例 361
▼配置當資料來源池包含主機與用戶時，複寫延遲的用戶端相似性 361
▼將用戶端相似性配置爲利用讀取作業驗證每個寫入作業 362
▼配置以連線爲基礎之路由的用戶端相似性 362

22 目錄代理伺服器分佈 363
配置目錄代理伺服器分佈演算法 363

配置模式對應分佈演算法 363
配置數值分佈演算法 364
配置字母分佈演算法 365
配置複寫分佈演算法 365
配置自訂分佈演算法 365
▼配置自訂分佈演算法 366
配置目錄代理伺服器以分佈尾碼資料 366
建立與配置使用範例的資料檢視 370
當子樹狀結構的不同部分儲存在不同的資料來源中時，提供單一存取點的資料檢視 370
▼配置當子樹狀結構的不同部分儲存在不同的資料來源中時，提供單一存取點的資料檢視 370
具階層與分佈演算法的資料檢視 371
▼配置具階層與分佈演算法的資料檢視 372

23 目錄代理伺服器虛擬 375
建立與配置 LDIF 資料檢視 375
▼建立 LDIF 資料檢視 376
▼配置 LDIF 資料檢視 376
定義虛擬資料檢視的存取控制 377
▼定義新的 ACI 儲存庫 377
▼配置虛擬存取控制 378
定義虛擬資料檢視的模式檢查 379
▼定義模式檢查 379
建立與配置連結資料檢視 380
▼建立連結資料檢視 380
▼配置連結資料檢視 380
▼配置連結資料檢視，以啓用多個連結資料檢視對單一資料檢視的參照 382
▼配置連結檢視的輔助檢視 382
建立與配置 JDBC 資料檢視 383
▼建立 JDBC 資料檢視 384
▼配置 JDBC 資料檢視 385
▼配置 JDBC 表格、屬性與物件類別 386
定義 JDBC 表格之間的關係 387
虛擬配置範例 390
連結 LDAP 目錄與 MySQL 資料庫 390

連結多個不同的資料來源 396

24 虛擬資料轉換 407
配置虛擬資料轉換 407
▼增加虛擬轉換 407
▼移除虛擬轉換 408
虛擬轉換範例 408
從項目的現有屬性導出屬性 408
將虛擬屬性對映至實體屬性 409
顯示第二個虛擬屬性值，由其他實體屬性指定 410
將第二個值儲存至由其他實體屬性指定的屬性 411
從輸出移除屬性 412
儲存項目時遮罩屬性 413
顯示屬性的預設值 414
將預設值儲存至屬性 415

25 目錄代理伺服器與後端 LDAP 伺服器之間的連線 417
配置目錄代理伺服器與後端 LDAP 伺服器之間的連線 417
▼配置目錄代理伺服器與後端 LDAP 伺服器之間的連線數 418
▼配置連線逾時 418
▼配置連線池等待逾時 419
配置目錄代理伺服器與後端 LDAP 伺服器之間的 SSL 419
▼配置目錄代理伺服器與後端 LDAP 伺服器之間的 SSL 419
選擇目錄代理伺服器的 SSL 密碼與協定 420
▼選擇密碼與協定清單 420
將請求轉寄至後端 LDAP 伺服器 421
利用重新執行連結轉寄請求 421
▼利用重新執行連結轉寄請求 421
利用代理授權轉寄請求 422
▼透過使用代理授權轉寄請求 422
▼當請求包含代理授權控制時，透過使用代理授權轉寄請求 423
轉寄不含用戶端身份識別的請求 423
▼轉寄不含用戶端身份識別的請求 423
以替代使用者身份轉寄請求 423
▼配置遠端使用者對映 424

▼配置本機使用者對映 424
▼配置匿名用戶端的使用者對映 425

26 用戶端與目錄代理伺服器之間的連線 427
建立、配置與刪除連線處理程式 427
▼建立連線處理程式 427
▼配置連線處理程式 428
▼刪除連線處理程式 430
▼配置資料檢視的相似性 430
建立與配置請求篩選策略與搜尋資料隱藏規則 431
▼建立請求篩選策略 431
▼配置請求篩選策略 431
▼建立搜尋資料隱藏規則 432
請求篩選策略與搜尋資料隱藏規則範例 433
建立與配置資源限制策略 434
▼建立資源限制策略 434
▼配置資源限制策略 434
▼自訂搜尋限制 435
配置目錄代理伺服器爲以連線爲基礎的路由器 436
▼配置目錄代理伺服器爲以連線爲基礎的路由器 436

27 目錄代理伺服器用戶端認證 439
配置用戶端與目錄代理伺服器之間的偵聽程式 439
▼配置用戶端與目錄代理伺服器之間的偵聽程式 439
認證目錄代理伺服器的用戶端 440
▼配置憑證型認證 441
▼配置匿名存取 441
▼爲 SASL 外部連結配置目錄代理伺服器 441

28 目錄代理伺服器記錄 443
檢視目錄代理伺服器記錄 443
配置目錄代理伺服器記錄 444
▼配置目錄代理伺服器存取與錯誤記錄 444
配置目錄代理伺服器記錄自動重建 446

▼配置定期自動重建存取與錯誤記錄 446
▼手動自動重建存取與錯誤記錄檔 447
▼停用存取與錯誤記錄自動重建 447
記錄自動重建的配置範例 448
刪除目錄代理伺服器記錄 449
▼配置根據時間刪除存取與錯誤記錄 449
▼配置根據檔案大小刪除存取與錯誤記錄 450
▼配置根據可用磁碟空間刪除存取與錯誤記錄 450
將警示記錄至 syslogd 常駐程式 450
▼配置目錄代理伺服器將警示記錄至 syslogd 常駐程式 450
配置作業系統接受 syslog 警示 451
▼配置 Solaris 作業系統接受 syslog 警示 451
▼配置 Linux 接受 syslog 警示 452
▼配置 HP-UX 接受 syslog 警示 452
經由目錄代理伺服器與目錄伺服器存取記錄追蹤用戶端請求 453
▼追蹤由目錄伺服器經目錄代理伺服器至用戶端應用程式的作業 453

29 目錄代理伺服器監視與警示 457
擷取關於目錄代理伺服器的監視資料 457
擷取關於資料來源的監視資料 457
▼透過偵聽錯誤監視資料來源 458
▼透過定期建立專屬連線監視資料來源 458
▼透過測試建立的連線監視資料來源 459
配置目錄代理伺服器的管理警示 460
▼啓用管理警示 460
▼將管理警示配置爲傳送至 Syslog 461
▼將管理警示配置爲傳送至電子郵件 461
▼將管理警示配置爲執行程序檔 462
透過使用 JVM 擷取關於目錄代理伺服器的監視資料 462
▼檢視 JVM 堆疊大小 462
▼在目錄代理伺服器執行時監視 JVM 堆疊大小 463

索引 465

前言

本管理指南提供從指令行配置目錄伺服器與目錄代理伺服器功能的程序性資訊。使用網路型介面 (目錄服務控制中心) 配置這些功能的相關指示位於線上說明中。

本書的適用對象

本管理指南針對目錄伺服器與目錄代理伺服器軟體的管理員所撰寫。

閱讀本書前

本書不提供軟體安裝方面的資訊。如需安裝資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」。

若要從舊版目錄伺服器或目錄代理伺服器進行遷移，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Migration Guide」以取得有關遷移伺服器的指示。若不熟悉本版的新功能，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Evaluation Guide」以取得新功能的簡介。

本書架構

第 1 部分提供管理目錄伺服器的程序性資訊。

第 2 部分提供管理目錄代理伺服器的程序性資訊。

本指南所使用的範例

爲求一致性，本指南將使用相同的範例資料。請將這些值取代爲您系統所適用的值。

表 P–1 範例中所使用的預設值

變數	範例中所使用的值
尾碼 (SUFFIX_DN)	dc=example,dc=com
實例路徑 (INSTANCE_PATH)	目錄伺服器：/local/ds/ 目錄代理伺服器：/local/dps/
主機名稱 (HOST)	host1、host2、host3
連接埠 (PORT)	LDAP：超級使用者的預設值：389。非超級使用者的預設值：1389 SSL 預設值：超級使用者的預設值：636。非超級使用者的預設值：1636

Directory Server Enterprise Edition 文件集

此 Directory Server Enterprise Edition 文件集說明如何使用 Sun Java System Directory Server Enterprise Edition 評估、設計、部署及管理目錄服務。此外也會說明如何開發 Directory Server Enterprise Edition 的用戶端應用程式。Directory Server Enterprise Edition 文件集位於 http://docs.sun.com/coll/1224.4 和 http://docs.sun.com/coll/1632.3。

如需 Directory Server Enterprise Edition 的簡介，請依序參閱下列文件。

表 P–2 Directory Server Enterprise Edition 文件

文件標題	內容
「Sun Java System Directory Server Enterprise Edition 6.3 版本說明」	包含 Directory Server Enterprise Edition 的最新相關資訊，包括已知問題等。
「Sun Java System Directory Server Enterprise Edition 6.3 Documentation Center」	包含指向文件集之重要區域的連結。
「Sun Java System Directory Server Enterprise Edition 6.3 Evaluation Guide」	簡介此發行版本的主要功能。說明這些功能的運作方式，以及它們在虛擬部署環境中提供了哪些可在單一系統上實作的功能。
「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」	說明如何根據 Directory Server Enterprise Edition，規劃及設計高度可用且可延伸的目錄服務。呈現部署規劃與設計的基本概念與原則。討論解決方案的生命週期，並提供在您根據 Directory Server Enterprise Edition 規劃解決方案時所適用的高階範例與策略。

表 P–2 Directory Server Enterprise Edition 文件 (續)

文件標題	內容
「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」	說明如何安裝 Directory Server Enterprise Edition 軟體。說明如何選取要安裝的元件、在安裝後配置這些元件，以及驗證配置的元件能否正確運作。 如需安裝目錄編輯器的相關指示，請移至http://docs.sun.com/coll/DirEdit_05q1 安裝目錄編輯器之前，請務必先閱讀「Sun Java System Directory Server Enterprise Edition 6.3 版本說明」中有關目錄編輯器的資訊。
「Sun Java System Directory Server Enterprise Edition 6.3 Migration Guide」	提供從舊版目錄伺服器、目錄代理伺服器與 Identity Synchronization for Windows 遷移的相關指示。
「Sun Java System Directory Server Enterprise Edition 6.3 管理指南」	提供用以管理 Directory Server Enterprise Edition 的指令行指示。 如需使用目錄服務控制中心 (DSCC) 管理 Directory Server Enterprise Edition 的相關提示與指示，請參閱 DSCC 中所提供的線上說明。 如需管理目錄編輯器的相關指示，請移至http://docs.sun.com/coll/DirEdit_05q1 如需安裝與配置 Identity Synchronization for Windows 的相關指示，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」中的第 II部分「Installing Identity Synchronization for Windows」。
「Sun Java System Directory Server Enterprise Edition 6.3 Developer's Guide」	展示如何使用 Directory Server Enterprise Edition 所提供的工具和 API 來開發目錄用戶端應用程式。
「Sun Java System Directory Server Enterprise Edition 6.3 Reference」	介紹 Directory Server Enterprise Edition 的技術與概念基礎。說明其元件、架構、程序與功能。同時也提供開發人員 API 的參照。
「Sun Java System Directory Server Enterprise Edition 6.3 Man Page Reference」	說明可透過 Directory Server Enterprise Edition 取得的指令行工具、模式物件與其他公用介面。本文件內各小節可安裝爲線上手冊的一部分。
「Sun Java System Directory Server Enterprise Edition 6.3 Troubleshooting Guide」	提供有關使用各種工具定義問題範圍、收集資料以及對問題區域疑難排解的資訊。
「Sun Java System Identity Synchronization for Windows 6.0 Deployment Planning Guide」	提供用以規劃及部署 Identity Synchronization for Windows 的一般性指導與最佳做法。

相關文件

SLAMD Distributed Load Generation Engine 是設計來著重測試與分析網路型應用程式效能的 Java™ 應用程式。此應用程式最初由 Sun Microsystems, Inc. 開發問世，用以評估及分析 LDAP 目錄伺服器的效能。SLAMD 經由 Sun Public License 這項 OSI 核准的開放原始碼授權，以開放原始碼應用程式的方式提供。若要取得 SLAMD 的相關資訊，請移至 http://www.slamd.com/。SLAMD 也會以 java.net 專案的形式提供。請參閱 https://slamd.dev.java.net/。

Java Naming and Directory Interface (JNDI) 技術支援使用 Java 應用程式的 LDAP 與 DSML v2 存取目錄伺服器。如需 JNDI 的相關資訊，請參閱 http://java.sun.com/products/jndi/。JNDI 指導中含有如何使用 JNDI 的詳細說明與範例。此指導位於 http://java.sun.com/products/jndi/tutorial/。

Directory Server Enterprise Edition 能夠以獨立產品、Sun Java Enterprise System 的元件、Sun 產品的部分套裝軟體 (如 Sun Java Identity Management Suite) 或 Sun 其他軟體產品的附加套裝軟體之形式進行授權。Java Enterprise System 是一項支援跨網路或網際網路環境中各種企業應用程式的軟體基礎架構。若將 Directory Server Enterprise Edition 授權爲 Java Enterprise System 的元件之一，請詳讀 http://docs.sun.com/coll/1286.3 和 http://docs.sun.com/coll/1412.2 上的系統文件。

Identity Synchronization for Windows 透過限定授權使用 Message Queue。如需 Message Queue 文件，請移至 http://docs.sun.com/coll/1307.2 和 http://docs.sun.com/coll/1421.2。

Identity Synchronization for Windows 可使用 Microsoft Windows 密碼策略。

- 如需有關 Windows 2003 密碼策略的資訊，請參閱線上 Microsoft 文件。
- 如需 Microsoft Certificate Services 企業根憑證授權單位的相關資訊，請參閱線上 Microsoft 支援文件。
- 如需在 Microsoft 系統上透過 SSL 配置 LDAP 的相關資訊，請參閱線上 Microsoft 支援文件。

可再分發的檔案

Directory Server Enterprise Edition 不提供任何可以再分發的檔案。

預設路徑與指令位置

本節說明文件中所使用的預設路徑，並提供指令在不同作業系統與部署類型中的所在位置。

預設路徑

本節中的表格說明此文件中所使用的預設路徑。如需安裝檔案的完整描述，請參閱下列產品文件。

- 「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 14 章「Directory Server File Reference」
- 「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 25 章「Directory Proxy Server File Reference」
- 「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的附錄 A「Directory Server Resource Kit File Reference」

表 P–3 預設路徑

預留位置	說明	預設值
install-path	表示 Directory Server Enterprise Edition 軟體的基底安裝目錄。 軟體會安裝在此基底 *install-path* 下的目錄中。例如，目錄伺服器軟體安裝在 *install-path*`/ds6/` 中。	當您使用 `dsee_deploy`(1M) 從 zip 發行檔中進行安裝時，預設的 *install-path* 即爲目前的目錄。您可以使用 `dsee_deploy` 指令的 `-i` 選項設定 *install-path*。 當您從原生套裝軟體發行檔進行安裝時，可能會使用 Java Enterprise System 安裝程式，此時預設的 *install-path* 即爲下列其中一個位置： ■ Solaris 系統 - `/opt/SUNWdsee/` ■ Red Hat 系統 - `/opt/sun/` ■ Windows 系統 - `C:\Program Files\Sun\JavaES5\DSEE`
instance-path	表示目錄伺服器或目錄代理伺服器實例的完整路徑。 本文件對目錄代理伺服器使用 `/local/dps/`，對目錄伺服器使用 `/local/ds/`。	不存在預設路徑。但**本機**檔案系統中一律會有實例路徑。 建議您使用下列目錄： Solaris 系統：`/var` 若您使用 Sun Cluster：`/global`
serverroot	表示 Identity Synchronization for Windows 安裝位置的父系目錄	視您的安裝而定。請注意，目錄伺服器已不再有 *serverroot* 的概念。
`isw-`*hostname*	表示 Identity Synchronization for Windows 實例目錄	視您的安裝而定
/path/to/`cert8.db`	表示 Identity Synchronization for Windows 之用戶端憑證資料庫的預設路徑和檔案名稱	*current-working-dir*`/cert8.db`
serverroot`/isw-`*hostname*`/logs/`	表示系統管理員、每個連接器和用戶端記錄程式之 Identity Synchronization for Windows 本機記錄的預設路徑	視您的安裝而定

表 P–3 預設路徑 (續)

預留位置	說明	預設值
serverroot/isw-*hostname*/logs/central/	表示 Identity Synchronization for Windows 中央記錄的預設路徑	視您的安裝而定

指令位置

本節的表格提供 Directory Server Enterprise Edition 文件中所使用之指令的位置。若想進一步瞭解各項指令，請參閱相關的線上手冊。

表 P–4 指令位置

指令	Java ES，原生套裝軟體發行檔	Zip 發行檔
cacaoadm	Solaris - /usr/sbin/cacaoadm	Solaris - *install-path*/dsee6/cacao_2/usr/sbin/cacaoadm
	Red Hat - /opt/sun/cacao/bin/cacaoadm	Red Hat、HP-UX - *install-path*/dsee6/cacao_2/cacao/bin/cacaoadm
	Windows - *install-path*\share\cacao_2\bin\cacaoadm.bat	Windows - *install-path*\dsee6\cacao_2\bin\cacaoadm.bat
certutil	Solaris - /usr/sfw/bin/certutil	*install-path*/dsee6/bin/certutil
	Red Hat - /opt/sun/private/bin/certutil	
dpadm(1M)	*install-path*/dps6/bin/dpadm	*install-path*/dps6/bin/dpadm
dpconf(1M)	*install-path*/dps6/bin/dpconf	*install-path*/dps6/bin/dpconf
dsadm(1M)	*install-path*/ds6/bin/dsadm	*install-path*/ds6/bin/dsadm
dsccmon(1M)	*install-path*/dscc6/bin/dsccmon	*install-path*/dscc6/bin/dsccmon
dsccreg(1M)	*install-path*/dscc6/bin/dsccreg	*install-path*/dscc6/bin/dsccreg
dsccsetup(1M)	*install-path*/dscc6/bin/dsccsetup	*install-path*/dscc6/bin/dsccsetup
dsconf(1M)	*install-path*/ds6/bin/dsconf	*install-path*/ds6/bin/dsconf
dsee_deploy(1M)	未提供	*install-path*/dsee6/bin/dsee_deploy
dsmig(1M)	*install-path*/ds6/bin/dsmig	*install-path*/ds6/bin/dsmig
entrycmp(1)	*install-path*/ds6/bin/entrycmp	*install-path*/ds6/bin/entrycmp

表 P–4 指令位置 (續)

指令	Java ES，原生套裝軟體發行檔	Zip 發行檔
fildif(1)	*install-path*/ds6/bin/fildif	*install-path*/ds6/bin/fildif
idsktune(1M)	未提供	在解壓縮之 Zip 發行檔的根目錄
insync(1)	*install-path*/ds6/bin/insync	*install-path*/ds6/bin/insync
ns-accountstatus(1M)	*install-path*/ds6/bin/ns-accountstatus	*install-path*/ds6/bin/ns-accountstatus
ns-activate(1M)	*install-path*/ds6/bin/ns-activate	*install-path*/ds6/bin/ns-activate
ns-inactivate(1M)	*install-path*/ds6/bin/ns-inactivate	*install-path*/ds6/bin/ns-inactivate
repldisc(1)	*install-path*/ds6/bin/repldisc	*install-path*/ds6/bin/repldisc
schema_push(1M)	*install-path*/ds6/bin/schema_push	*install-path*/ds6/bin/schema_push
smcwebserver	Solaris、Linux - /usr/sbin/smcwebserver	此指令僅適用於使用本機套裝軟體發行檔安裝的 DSCC。
	Windows - *install-path*\share\webconsole\bin\smcwebserver	
wcadmin	Solaris、Linux - /usr/sbin/wcadmin	此指令僅適用於使用本機套裝軟體發行檔安裝的 DSCC。
	Windows - *install-path*\share\webconsole\bin\wcadmin	

印刷排版慣例

下表說明本書所使用的印刷排版變更。

表 P–5 印刷排版慣例

字體	意義	範例
AaBbCc123	指令、檔案及目錄的名稱；螢幕畫面輸出。	請編輯您的 .login 檔案。 請使用 ls -a 列出所有檔案。 machine_name% you have mail.
AaBbCc123	您所鍵入的內容 (與螢幕畫面輸出相區別)。	machine_name% **su** Password:
AaBbCc123	預留位置，使用實際名稱或值取代	用以移除檔案的指令爲 rm *filename*。

表 P–5 印刷排版慣例 *(續)*

字體	意義	範例
AaBbCc123	書籍標題、新術語與要強調的字詞 (請注意，某些強調的項目線上顯示爲粗體)	請閱讀「使用者指南」中的第 6 章。 **快取**是指本機儲存的副本。 請**勿**儲存此檔案。

指令範例中的 Shell 提示符號

下表列出預設的系統提示符號與超級使用者提示符號。

表 P–6 Shell 提示符號

Shell	提示符號
UNIX 與 Linux 系統上的 C shell	`machine_name%`
UNIX 與 Linux 系統上的 C shell 超級使用者	`machine_name#`
UNIX 與 Linux 系統上的 Bourne shell 與 Korn shell	`$`
UNIX 與 Linux 系統上的 Bourne shell 與 Korn shell 超級使用者	`#`
Microsoft Windows 指令行	`C:\`

符號慣例

下表說明本書可能使用的符號。

表 P–7 符號慣例

符號	說明	範例	意義
`[ ]`	包含選用引數與指令選項。	`ls [-l]`	`-l` 選項不是必要選項。
`{ | }`	包含必要指令選項的一組選擇。	`-d {y|n}`	`-d` 選項要求使用 `y` 引數或 `n` 引數。
`${ }`	表示變數參照。	`${com.sun.javaRoot}`	參照 `com.sun.javaRoot` 變數的值。
-	連結需要同時按下的多個按鍵。	Ctrl-A	按住 Ctrl 鍵，再按 A 鍵。
+	連結需要連續按下的多個按鍵。	Ctrl+A+N	按下 Ctrl 鍵，放開，然後再按下後面兩個鍵。

表 P–7 符號慣例 *(續)*

符號	說明	範例	意義
→	表示圖形化使用者介面中的功能表項目選取。	[檔案] → [新增] → [範本]	從 [檔案] 功能表中，選擇 [新增]；再從 [新增] 子功能表中，選擇 [範本]。

文件、支援與培訓

Sun 網站提供下列其他資源的相關資訊：

- 文件 (`http://www.sun.com/documentation/`)
- 支援 (`http://www.sun.com/support/`)
- 培訓 (`http://www.sun.com/training/`)

協力廠商網站參照

本文件中會參照協力廠商的 URL，以提供其他相關資訊。

備註 – Sun 對於本文件中所提及之協力廠商網站的可用性不承擔任何責任。Sun 對於此類網站或資源中的 (或透過它們所取得的) 任何內容、廣告、產品或其他材料不做背書，也不承擔任何責任。對於因使用或依靠此類網站或資源中的 (或透過它們所取得的) 任何內容、產品或服務而造成的、名義上造成的或連帶產生的任何實際或名義上之損壞或損失，Sun 概不負責，也不承擔任何責任。

搜尋 Sun 產品文件

除了從 docs.sun.com 網站搜尋 Sun 產品文件以外，也可以在搜尋欄位中鍵入下列語法，以所選擇的搜尋引擎進行搜尋：

search-term `site:docs.sun.com`

例如，若要搜尋**目錄伺服器**，請鍵入：

```
"Directory Server" site:docs.sun.com
```

若要在搜尋中納入其他 Sun 網站，如 java.sun.com、www.sun.com 與 developers.sun.com，請在搜尋欄位中以 `sun.com` 取代 `docs.sun.com`。

Sun 歡迎您提出寶貴意見

Sun 致力於提高文件品質，因此誠心歡迎您提出意見與建議。若要提出您的意見，請至 http://docs.sun.com 並按一下 [傳送意見] (Send Comments)。在線上表格中，請提供文件標題及文件號碼。文件號碼位於書本的標題頁或文件的 URL 中，通常是一組 7 位或 9 位數的數字。例如，本書的文件號碼爲 820-4819。

在您提出意見時，可能需要在表單中輸入英文版書名和文件號碼，本書的英文版文件號碼和書名爲：820-2763 和「Sun Java System Directory Server Enterprise Edition 6.3 Administration Guide」。

◆ ◆ ◆ 第 1 章

1 安裝之前

在生產環境中安裝 Directory Server Enterprise Edition 軟體之前，請取得透過「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」協助所建立之部署規劃。在備妥規劃下，閱讀本節以判斷如何著手開始進行部署安裝。

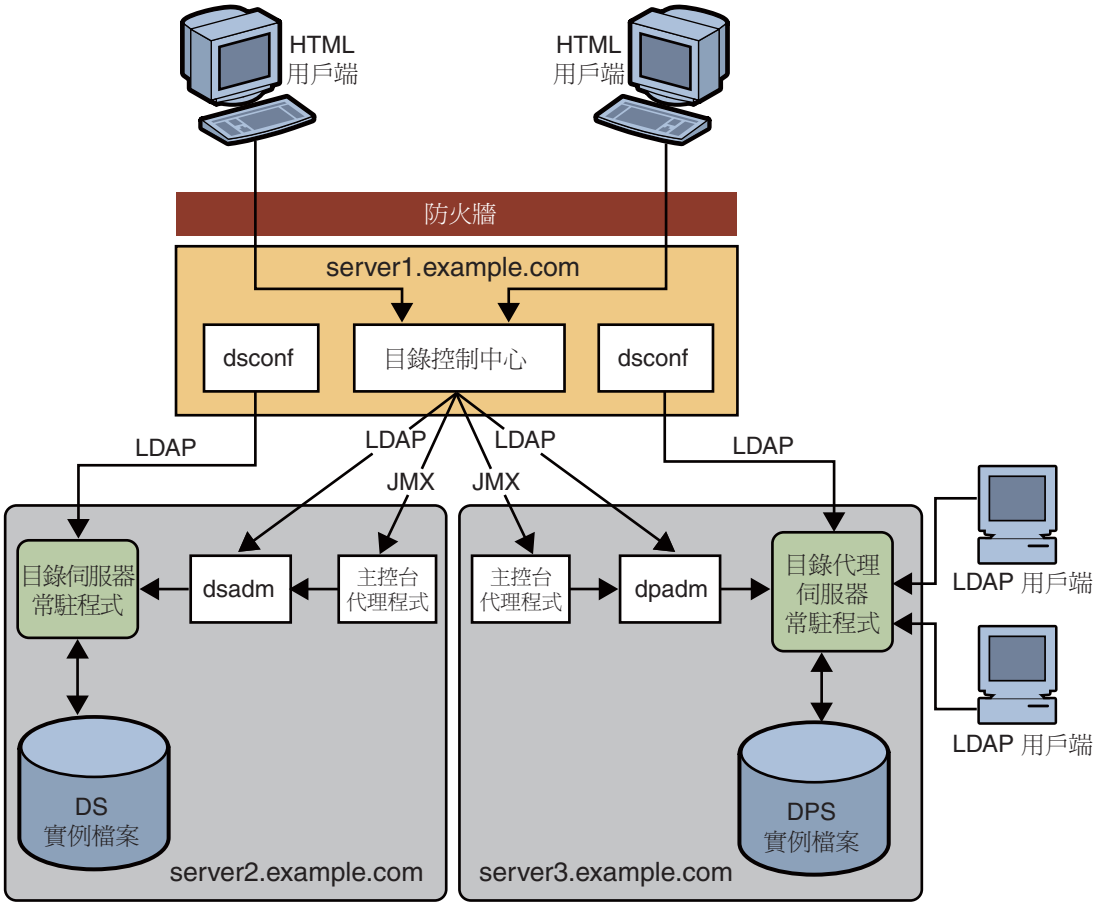
本章包含以下小節。

- 第 33 頁的「管理架構與安裝」簡短說明管理架構概念，此爲在生產環境中進行安裝作業的關鍵。
- 第 36 頁的「單一系統與分散式安裝的比較」將涉及單一主機系統的安裝與涉及多重系統的安裝進行比較與對照。
- 第 38 頁的「Directory Server Enterprise Edition 軟體發行檔」會比較已上市之不同的 Directory Server Enterprise Edition 軟體發行檔。
- 第 41 頁的「安裝於 Solaris 區域中」則在討論於 Solaris 區域中安裝 Directory Server Enterprise Edition 時的考量重點。

管理架構與安裝

本節重點在於管理架構的主要部份，而您必須瞭解這些主要部份才可在生產環境中安裝伺服器軟體。本節並不會討論 Directory Server Resource Kit 隨附的開發人員與效能調校工具。您可以在管理架構之外另行安裝這類工具。

閱讀本節之前，請先閱讀「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中的「Directory Server Enterprise Edition Administration Model」。下圖顯示網路流量的流動情形。圖中顯示配置管理工具、DSCC、`dsconf`(1M)、`dpconf`(1M)、本機管理代理程式，以及伺服器之間的網路流量。另外也會顯示本機代理程式、本機指令行工具、`dsadm`(1M) 與 `dpadm`(1M)，以及您所管理的伺服器之間的通訊。



請注意，指令行管理與監視工具 (dsconf(1M) 與 dpconf(1M)) 僅需您所管理的伺服器之 LDAP 存取權即可。LDAP 流量一般透過預設連接埠 389 (LDAP) 與 636 (使用 SSL 的安全 LDAP) 流動。當您以非**超級使用者**的使用者身份建立伺服器時，預設連接埠爲 1389 (LDAP) 與 1636 (使用 SSL 的安全 LDAP)。

依慣例，僅有**超級使用者**身份的使用者可使用小於 1024 的保留連接埠號碼安裝軟體。Solaris 系統可讓管理員允許非**超級使用者**身份的使用者，透過以角色爲基礎的存取控制 (RBAC) 使用專用連接埠。

DSCC 是以下列模式執行的 Web 應用程式：

- 使用本機套裝軟體發行檔進行安裝時，位於 Sun Java Web Console 架構內。
- 使用 Zip 發行檔安裝時，位於 Sun Java Web Console 外。

您一般僅會在部署的一部系統上安裝 DSCC。接著便可從該 DSCC 安裝，管理所有伺服器。您可以根據安裝 Directory Server Enterprise Edition 所使用的軟體發行檔 (如果使用

Zip 發行檔安裝，則根據應用程式伺服器的配置)，使用 `https://`*hostname*`:6789`、`http://`*hostname*`:8080` 或 `https://`*hostname*`:8181` 等 URL，透過瀏覽器存取 DSCC。

DSCC 需要伺服器的 LDAP 存取權，才可進行線上管理作業。DSCC 也需要隨伺服器安裝之代理程式的 Java Management Extension (JMX) 存取權。代理程式代理 DSCC 執行伺服器程序管理作業，此作業無法透過 LDAP 在執行中的伺服器上執行。您可以使用 DSCC 建立並啓動新的伺服器。

在一般安裝程序中，會同時安裝本機 DSCC 代理程式與伺服器軟體。DSCC 使用特定連接埠號碼透過網路聯繫代理程式。因此，您必須接受預設連接埠號碼 `11162`，或指定不同的連接埠號碼。

代理程式會執行於伺服器系統上的共用代理程式容器內。此共用代理程式容器提供其代理程式管理應用程式的單一外部連接埠。共用代理程式容器亦會彙總資源，以在多個本機代理程式共用容器的系統上儲存資源。共用代理程式容器是在預設連接埠 `11162` 上偵聽 DSCC 的代理程式，其並會將管理流量路由至其他代理程式。DSCC 透過共用代理程式容器與本機代理程式通訊。爲進行疑難排解，共用代理程式容器可使用 `cacaoadm` 指令獨立管理。

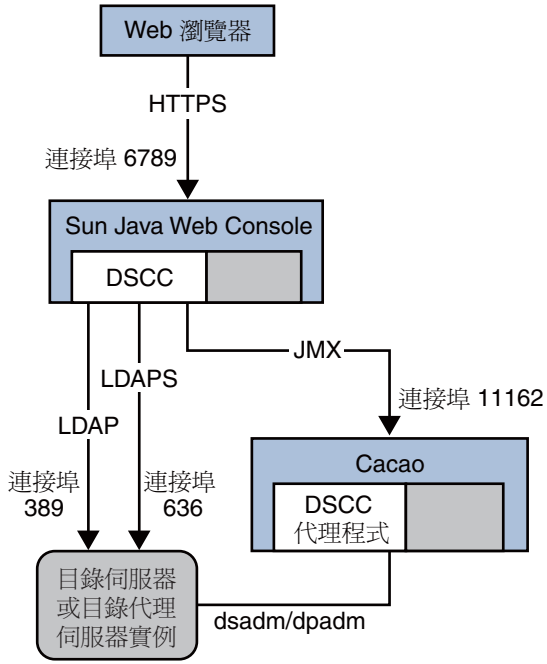


圖 1–1 安裝本機套裝軟體發行檔之後的連接埠與協定

每次從 Zip 發行檔安裝 Directory Server Enterprise Edition 軟體時，也會一併安裝共用代理程式容器實例。因此，當您在相同主機系統上平行安裝多個版本時，**僅會有一個版**

本可使用預設連接埠。您無法從共用代理程式容器實例已使用預設連接埠的 Zip 發行檔進行安裝。您必須接著爲其他共用代理程式容器實例指定不同的連接埠號碼。

伺服器軟體安裝是三階段的程序。

1. 安裝配置管理軟體。

   配置管理工具會隨即安裝並初始化 DSCC。

   當 DSCC 將其配置資料儲存於其本身私人的目錄伺服器實例中時，會在 DSCC 安裝期間同時從本機套裝軟體安裝目錄伺服器。
2. 在您規劃執行伺服器實例的系統上安裝伺服器軟體。

   伺服器軟體、必要的程式庫、本機管理工具與本機代理程式會隨即安裝。會安裝所有軟體以讓您設定目錄服務，但此時還不會執行任何伺服器。
3. 在系統上建立與配置伺服器實例。

   目錄伺服器與目錄代理伺服器實例會隨即建立。這些實例會使用 DSCC，或透過隨伺服器軟體安裝的本機管理工具來建立。接著會透過 DSCC 或配置管理指令行工具，配置伺服器實例。

當您在單一主機系統上安裝所有項目時，前兩個階段會合併。DSCC 會使用本機代理程式在伺服器上執行特定的作業。因此，您必須在本機共用代理程式容器中安裝本機代理程式。

在 Zip 發行檔中，會在第二階段期間將配置 DSCC 所使用的網頁歸檔 (WAR) 檔案複製到系統。在第一階段期間，不會完成任何安裝作業或任何 WAR 檔案初始化。WAR 檔案會進一步透過支援的應用程式伺服器進行部署以配置 DSCC。

單一系統與分散式安裝的比較

本節會將單一主機系統安裝與涉及多部系統的安裝進行比較與對照。

以下是安裝可進行的方式：

1. 在與您管理之伺服器**相同的主機**上，安裝 DSCC 並配置管理工具。或在與您從遠端管理之伺服器**不同的主機**上安裝工具。
2. 在**相同的主機**上建立多個伺服器實例，或在**不同的主機**上建立各個伺服器實例。

安裝目錄服務控制中心的位置

在與您管理的伺服器**相同的主機**上安裝DSCC，可爲評估與部署提供快速且簡單的解決方案。此解決方案不適用於生產安裝，生產安裝時仰賴備援系統及伺服器複本以提供高可用性。

當您安裝 DSCC 時，也會一併安裝目錄伺服器軟體。DSCC 使用其本身私人的目錄伺服器實例，儲存配置資訊。若同時安裝目錄伺服器的本機代理程式與 DSCC，則可使用 DSCC 在系統上建立目錄伺服器實例。您不需要知道其他主機名稱與連接埠號碼，便可執行此作業。

您可以在與您從遠端管理的伺服器**不同的主機**上安裝 DSCC。此解決方案適用生產安裝，生產安裝時仰賴備援系統及伺服器複本以提供高可用性。

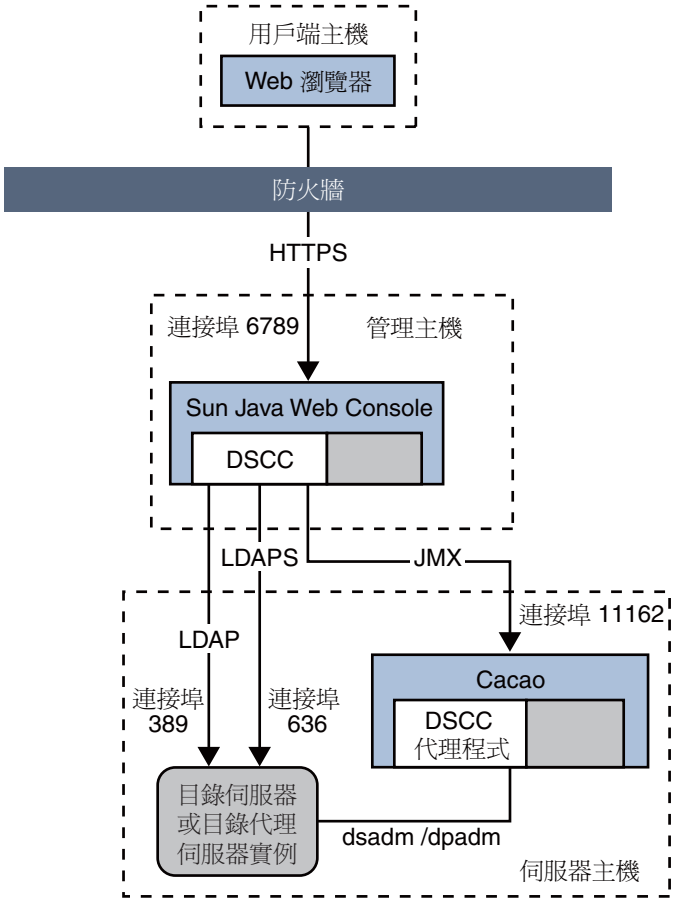


圖 1–2 安裝本機套裝軟體發行檔之後不同系統上的管理主機與伺服器主機

您必須是**超級使用者**，才可於管理主機上安裝 DSCC。但是，您可使用安裝於管理主機上之 DSCC，管理以非**超級使用者**身份所安裝的伺服器主機。

備註 – 使用由支援的應用程式伺服器部署之 WAR 檔案所配置的 DSCC，會在 Sun Java Web Console 外安裝 DSCC，且任何非**超級使用者**身份的使用者皆可執行此動作。

例如，您可以在資料中心外的伺服器甚至是適合的工作站上，安裝 DSCC。您也可以從 Zip 發行檔於資料中心內的伺服器主機上安裝伺服器軟體，並以非**超級使用者**身份執行這類安裝。透過安全 LDAP 與 JMX，您可以接著透過管理主機上的 DSCC 建立、配置及管理所有伺服器。

建立伺服器實例的位置

生產安裝時仰賴備援系統、負載平衡、容錯移轉功能與伺服器複本，提供高可用性。因此，您一般會在多部主機系統上建立伺服器。但是，功能強大的主機系統則可能各自可包含多個伺服器實例。

當您在單一主機系統上建立多個伺服器實例時，僅有其中一個伺服器實例可在預設連接埠上進行偵聽。若是僅安裝 Directory Server Enterprise Edition 軟體一次，多個伺服器實例即可共用相同的共用代理程式容器。

若是在系統上安裝多個 Directory Server Enterprise Edition 版本，則各個版本會隨附其各自的共用代理程式容器。僅有其中一個共用代理程式容器可在預設連接埠上偵聽 JMX 管理流量。

Directory Server Enterprise Edition 軟體發行檔

本節會比較已上市之不同 Directory Server Enterprise Edition 軟體發行檔。

- 第 39 頁的 「Java Enterprise System 發行檔」會介紹 Sun Java Enterprise System 隨附的本機套裝軟體發行檔。

  您可以使用 Java Enterprise System 安裝程式從 Java Enterprise System 發行檔安裝軟體。
- 第 39 頁的 「原生修補程式」可讓您升級 Directory Server Enterprise Edition 6.0 與 6.1 安裝。
- 第 40 頁的 「Zip 發行檔」會介紹支援以非**超級使用者**身份進行安裝的 Zip 發行檔。
- 第 40 頁的 「Java Enterprise System 發行檔與 Zip 發行檔的比較」則摘要各個發行檔所提供的軟體。

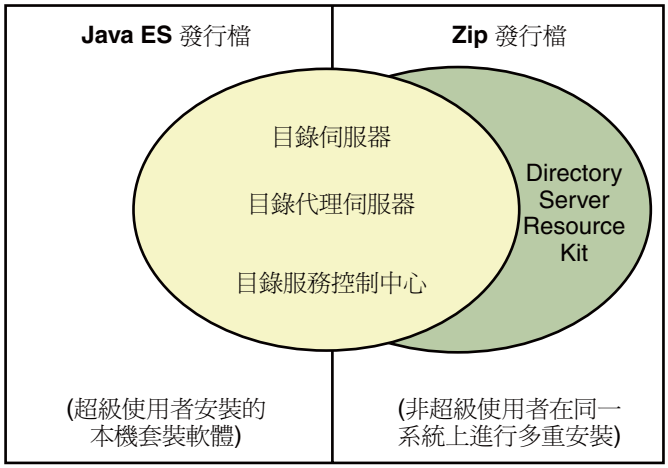


圖 1–3 兩種軟體發行檔

Java Enterprise System 發行檔

本節介紹 Java ES 安裝程式隨附的 *Java Enterprise System* **發行檔**。

Java ES 安裝程式提供圖形化精靈、指令行互動精靈，以及無訊息安裝功能，將本機套裝軟體增加至系統。由於此發行檔以本機套裝軟體爲基礎，因此您必須是**超級使用者**，才可使用 Java ES 安裝程式執行安裝。

Java ES 安裝程式提供在 Solaris 與 Linux 上的全新 Directory Server Enterprise Edition 6.3 安裝。若要在 Windows 上安裝 Directory Server Enterprise Edition 6.3，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」中的「Installation Procedure Quick Reference」。HP-UX 上不提供 Directory Server Enterprise Edition 6.3。

所有 Java ES 軟體會仰賴包含基本元件與程式庫的共用架構協同合作。因此，您可以在單一系統上安裝所有軟體產品。

Java ES 安裝軟體也可協助安裝共用元件。此軟體會與系統整合，因此您可以配置在作業系統重新啓動時自動重新啓動目錄服務。透過以本機套裝軟體所進行的安裝，您可以從屬於作業系統的套裝軟體版本控制與修補工具獲益。

本指南不會說明使用 Java ES 安裝程式之所有替代安裝方法。但是，本指南會討論與 Directory Server Enterprise Edition 6.2 軟體安裝相關的主要 Java ES 安裝程式精靈畫面。如需有關使用 Java ES 安裝程式所有功能的詳細指示，請參閱 `http://docs.sun.com/coll/1286.3` 的 Java Enterprise System 文件。

原生修補程式

本節介紹可讓您升級 Directory Server Enterprise Edition 6.0 與 6.1 安裝的原生修補程式。

您必須是**超級使用者**，才可使用原生修補程式進行安裝。這些修補程式會套用於現有的 Directory Server Enterprise Edition 6.0 或 6.1 安裝之上。原生修補程式包含與 Java Enterprise System 發行檔中相同的所有 Directory Server Enterprise Edition 元件，但僅會升級 Directory Server Enterprise Edition 6.0 或 6.1 安裝中已安裝的元件。您無法使用原生修補程式對 Directory Server Enterprise Edition 中的任何元件進行全新安裝。

但可以透過在 Directory Server Enterprise Edition 6.0 安裝之後安裝原生修補程式，於 Windows 上安裝 Directory Server Enterprise Edition 6.3。Java Enterprise System 發行檔不提供在 Windows 上進行 Directory Server Enterprise Edition 6.3 全新安裝。

Zip 發行檔

本節介紹提供 `dsee_deploy`(1M) 指令行安裝程式的 *Zip* **發行檔**。

此發行檔提供獨立軟體，您可以在本機磁碟上任何具有寫入權限的位置安裝軟體。您可以非**超級使用者**身份，安裝與管理 Zip 發行檔軟體。

由於 Zip 發行檔軟體爲獨立軟體，因此從 Zip 發行檔執行的每個軟體安裝彼此無關。因此，您可以在相同的系統上，從多個 Zip 發行檔版本安裝軟體。系統管理員必須手動配置您安裝的軟體，使其於作業系統重新啓動時重新啓動。

此外，使用 Zip 發行檔時，必須謹慎記錄安裝的項目以及已套用的修補程式。

Java Enterprise System 發行檔與 Zip 發行檔的比較

本節說明各個發行檔所支援的軟體。

Java ES 與 Zip 發行檔皆可讓您以非**超級使用者**身份，建立與配置目錄伺服器與目錄代理伺服器實例。

Directory Server Enterprise Edition 軟體元件	Java Enterprise System 發行檔	Zip 發行檔
目錄服務控制中心	提供	提供，可透過使用應用程式伺服器部署 WAR 檔案進行配置
目錄伺服器	提供	提供，可使用 `dsee_deploy` 進行安裝
目錄代理伺服器	提供	提供，可使用 `dsee_deploy` 進行安裝
目錄編輯器	此發行檔不提供	提供，但**無法**使用 `dsee_deploy` 安裝

Directory Server Enterprise Edition 軟體元件	Java Enterprise System 發行檔	Zip 發行檔
Identity Synchronization for Windows	此發行檔不提供	提供，但**無法**使用 `dsee_deploy` 安裝
Directory Server Resource Kit	此發行檔不提供	提供，可使用 `dsee_deploy` 安裝

備註 – 一個伺服器實例僅可由一個 DSCC 管理。

Identity Synchronization for Windows 與目錄編輯器軟體隨附於 Zip 發行檔中，但不使用 `dsee_deploy` 指令安裝。本指南包含 Identity Synchronization for Windows 安裝。請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」中的第 II部分「Installing Identity Synchronization for Windows」。

本指南不涵蓋目錄編輯器軟體的安裝。若計劃安裝目錄編輯器軟體，請閱讀「Sun Java System Directory Editor 1 2005Q1 Installation and Configuration Guide」中的安裝說明。

安裝於 Solaris 區域中

本節討論在 Solaris 區域中安裝 Directory Server Enterprise Edition 時的考量重點。

全域與完全本機 Solaris 區域提供 Directory Server Enterprise Edition 軟體完整的系統。Directory Server Enterprise Edition 軟體將這兩個區域視爲無關的實體系統。Directory Server Enterprise Edition 安裝類似於安裝在獨立的系統上。軟體不會與其他區域共用服務或檔案位置。

在稀疏區域中，您可以安裝某些服務以全系統的方式使用。因此，多個 Java ES 伺服器實例可使用單一 Java Enterprise System 通用元件服務實例。例如，稀疏區域中的 Directory Server Enterprise Edition 軟體可使用安裝於全域中的相同共用代理程式容器與 Java ES Monitoring Framework。但是，您必須安裝全系統服務，才可完成相依於全系統服務的稀疏區域軟體安裝。

當您在稀疏區域中進行安裝時，Directory Server Enterprise Edition 不需要您使用全系統服務。當您從 Zip 發行檔安裝獨立軟體時，也可以在稀疏區域中安裝通用元件服務。因此，在稀疏區域中安裝 Zip 發行檔類似於在獨立系統上安裝。

下表概述 Directory Server Enterprise Edition 安裝的限制，實際上就是在稀疏區域中安裝的限制。

Directory Server Enterprise Edition 軟體元件	軟體發行檔	在全域或完全本機區域中安裝的限制	稀疏區域安裝的限制
目錄服務控制中心	Java Enterprise System 發行檔	沒有限制	先在全域中安裝 Java Enterprise System 共用元件，然後在稀疏區域中安裝目錄服務控制中心。
	Zip 發行檔	沒有限制	沒有限制
目錄伺服器	Java Enterprise System 發行檔	沒有限制	先在全域中安裝 Java Enterprise System 共用元件，然後在稀疏區域中安裝目錄伺服器。
	Zip 發行檔	沒有限制	沒有限制
目錄代理伺服器	Java Enterprise System 發行檔	沒有限制	先在全域中安裝 Java Enterprise System 共用元件，然後在稀疏區域中安裝目錄代理伺服器。
	Zip 發行檔	沒有限制	沒有限制
目錄編輯器	Zip 發行檔	沒有限制	Web 應用程式容器必須允許在稀疏區域中進行安裝。
Identity Synchronization for Windows	Zip 發行檔	不支援	不支援
Directory Server Resource Kit	Zip 發行檔	沒有限制	沒有限制

如需有關從 Java Enterprise System 發行檔在稀疏區域中安裝的相關資訊，請參閱 `http://docs.sun.com/coll/1286.3` 的 Java Enterprise System 文件。

第 1 部分

目錄伺服器管理

◆ ◆ ◆ 第 2 章

目錄伺服器工具

Sun Java™ System Directory Server Enterprise Edition 提供可在複寫環境中管理多重伺服器、實例與尾碼的瀏覽器介面與指令行工具。本章提供有關目錄伺服器管理工具的簡介資訊。

本章包含下列主題：

- 第 45 頁的「目錄伺服器管理簡介」
- 第 45 頁的「決定 DSCC 與指令行的使用時機」
- 第 47 頁的「目錄服務控制中心介面」
- 第 51 頁的「目錄伺服器指令行工具」

目錄伺服器管理簡介

此文件集中的其他指南提供有關目錄伺服器管理架構的資訊。

- 如需目錄伺服器管理架構的簡介，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中的「Directory Server Enterprise Edition Administration Model」。
- 如需有關目錄伺服器管理架構之更詳細的參考資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 1 章「Directory Server Overview」。

決定 DSCC 與指令行的使用時機

Directory Server Enterprise Edition 提供管理目錄伺服器與目錄代理伺服器的兩個使用者介面：瀏覽器介面目錄服務控制中心 (DSCC) 與指令行介面。

決定某程序是否可以使用 DSCC 完成

本指南中大部分的程序皆可使用指令行或 DSCC 執行。本指南中的程序說明如何使用指令行完成程序。大部分的情況下皆可使用 DSCC 執行相同作業。若 DSCC 可用於執行特定程序，則程序的開頭處就會提供執行說明。

DSCC 線上說明針對如何使用 DSCC 執行本指南中的程序，提供了詳細的指示。

較適合使用 DSCC 的情況

比起使用指令行，DSCC 可讓您更容易執行某些作業與工作，以下幾節中將有所說明。一般而言，必須套用至數部伺服器的指令最好使用 DSCC 執行。

檢視伺服器與尾碼複寫狀態

DSCC 具有顯示表格，可呈現已在 DSCC 中註冊的所有伺服器實例、所有已配置的尾碼，以及這些項目的狀態。

伺服器表格位於 [目錄伺服器] 標籤上，可顯示伺服器的操作狀態。如需可能之伺服器狀態的完整清單，請參閱目錄伺服器線上說明。

尾碼表格位於 [尾碼] 標籤上，並會顯示複寫狀態資訊，例如項目數以及任何遺失變更的數量與存在時間。如需有關此表格所顯示之資訊的更多資訊，請參閱目錄伺服器線上說明。

管理伺服器群組

伺服器群組可協助您監控並配置伺服器。您可以建立群組，以及將伺服器指定至群組中。例如，您可以依地理位置或功能，將伺服器群組化。若具有大量的伺服器，則可以篩選 [目錄伺服器] 標籤上所顯示的伺服器，而僅顯示群組中的伺服器。您也可以將某一伺服器的配置 (例如，索引或快取設定) 複製到群組中的其他所有伺服器。如需有關如何設定及使用伺服器群組的指示，請參閱目錄伺服器線上說明。

複製配置設定

DSCC 可讓您將現有伺服器、尾碼或複寫協議的配置設定，複製到一或多個其他的伺服器、尾碼或複寫協議。如需有關如何執行前述各項作業的資訊，請參閱目錄伺服器線上說明。

配置複寫

使用 DSCC 可讓您快速而輕鬆地設定複寫拓樸。只需建立伺服器實例，再使用 DSCC 所提供的步驟指定各伺服器的角色即可。DSCC 可自動爲您建立複寫協議。如需有關如何使用 DSCC 配置複寫的更多資訊，請參閱目錄伺服器線上說明。

目錄服務控制中心介面

目錄服務控制中心 (DSCC) 是可讓您使用瀏覽器管理目錄伺服器與目錄代理伺服器的使用者介面。

若要配置 DSCC，請參閱第 71 頁的 「配置 DSCC」。如需有關使用 DSCC 的資訊，請參閱下列幾節。

DSCC 的管理使用者

DSCC 具有數項管理登入。

- **作業系統使用者**。可建立伺服器實例，是唯一有權使用 `dsadm` 指令，在伺服器實例上執行作業系統指令的使用者。DSCC 可能會在某些情況下要求提供作業系統使用者密碼。此使用者必須具有密碼，且必須能夠建立目錄伺服器實例。
- **目錄管理員**。伺服器的 LDAP 超級使用者。預設 DN 爲 `cn=Directory Manager`。
- **目錄伺服器管理員**。管理目錄伺服器。此使用者的權利與目錄管理員相同，但須受存取控制、密碼策略與認證需求的限制。您可以依需要建立目錄伺服器管理員，數量不限。
- **目錄服務管理員**。透過 DSCC 管理多部機器上的伺服器配置與資料。此使用者對已於 DSCC 中註冊的每部伺服器具有與目錄管理員相同的權利，並且是目錄伺服器管理員群組的成員之一。

▼ 存取 DSCC

若在存取 DSCC 時遇到任何困難，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」中的「To Troubleshoot Directory Service Control Center Access」。

1. **請確定已如「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」中的「Software Installation」所述，正確安裝 DSCC。**
2. **若已使用本機套裝軟體安裝檔案安裝了 DSCC，請遵循下列步驟：**

   a. **開啓瀏覽器，然後以下列格式鍵入 DSCC 主機 URL：**

      `https://`*hostname*`:6789`

      例如：

      ```
      https://host1:6789
      ```

      其中 hostname 是您安裝 DSCC 軟體所在的系統。

      Sun Java Web Console 預設連接埠爲 `6789`。

下圖顯示 Sun Java Web Console 登入視窗。

圖 2–1 Sun Java Web Console 登入視窗

b. 登入 Sun Java Web Console。

- 若這是您第一次登入 Sun Java Web Console，請以安裝 DSCC 軟體所在系統的**超級使用者**身份登入。
- 若這是後續登入，請鍵入作業系統使用者名稱與密碼。此使用者應具有啓動、停止及管理目錄伺服器實例的權限。

您登入時會看見應用程式的清單。

c. 選取目錄服務控制中心 (DSCC)。

DSCC 登入視窗會隨即顯示。

3 **若已使用 Zip 安裝檔案安裝了 DSCC，請遵循下列步驟：**

a. **鍵入 DSCC 主機 URL，以直接在您喜好的應用程式伺服器內存取 DSCC。根據應用程式伺服器配置之不同，DSCC 主機 URL 會是下列其中之一。**

```
https://hostname:8181/dscc
```

或

```
http://hostname:8080/dscc
```

b. **使用下列指令初始化 DSCC。**

```
$ install path/dscc6/bin/dsccsetup ads-create
```

4 **登入 DSCC。**

若這是第一次登入 DSCC，則必須設定目錄服務管理員密碼。當您日後登入時，請使用第一次登入時所設定的密碼。

現在即已登入 DSCC，並位於 [常用工作] 標籤上。

圖 2–2 DSCC [常用工作] 標籤

5 **使用標籤進行瀏覽。**

- [常用工作] 標籤中具有常用視窗與精靈的捷徑。
- [目錄伺服器] 標籤會顯示 DSCC 所管理的所有目錄伺服器。若要檢視用以管理及配置特定伺服器的其他選項，請按一下該伺服器的名稱。

- [代理伺服器] 標籤會顯示 DSCC 所管理的所有目錄代理伺服器。若要檢視用以管理及配置特定伺服器的其他選項，請按一下該伺服器的名稱。

備註 – 如需有關如何使用 DSCC 執行作業的指示，請參閱 DSCC 線上說明。

DSCC 標籤說明

圖 2–3 伺服器子標籤上的目錄伺服器清單

使用 DSCC 中的標籤瀏覽介面。

[常用工作] 標籤

[常用工作] 標籤 (請參閱圖 2–2) 是開啓 DSCC 時所看到的第一個介面。其中包含常用管理作業的連結，例如搜尋目錄資料、檢查記錄與管理伺服器。

[目錄伺服器] 標籤

[目錄伺服器] 標籤 (請參閱圖 2–3) 會列出 DSCC 中已註冊的所有目錄伺服器。您可以檢視每部伺服器的狀態與實例路徑，其會顯示實例的所在位置。

按一下伺服器名稱時，會看見另一個視窗，其中會顯示僅與該部伺服器相關的不同標籤組。

[代理伺服器] 標籤

[代理伺服器] 標籤會列出 DSCC 中已註冊的所有目錄代理伺服器。您可以檢視每部伺服器的狀態與實例路徑，其會顯示實例的所在位置。

按一下伺服器名稱時，會看見另一個視窗，其中會顯示僅與該部伺服器相關的不同標籤組。

[伺服器群組] 標籤

[伺服器群組] 標籤可讓您指定伺服器至群組中，以利於伺服器的管理。若您具有爲數眾多的伺服器，可以使用篩選器而僅顯示特定群組中的伺服器。您也可以將某一伺服器的配置 (例如，索引或快取設定) 複製到群組中的其他所有伺服器。

[設定] 標籤

此標籤會顯示 DSCC 連接埠號碼，並可讓您建立及刪除目錄服務管理員。

DSCC 線上說明

線上說明提供下列項目：

- 目前所使用之頁面的即時線上說明。
- 使用 DSCC 執行管理與配置程序時的一般說明。

在大部分的頁面中，只要按一下畫面右上角的 [說明] 按鈕，即可存取說明。在精靈中要存取說明時，請按一下 [說明] 標籤。也可以從 [常用工作] 標籤存取線上說明。

目錄伺服器指令行工具

您在 DSCC 上執行的大部分作業皆可使用指令行工具執行。這些工具可讓您直接從指令行管理目錄伺服器，以及使用程序檔管理您的伺服器。

主要的目錄伺服器指令爲 `dsadm` 與 `dsconf`。您可以使用這些指令執行備份、匯出至 LDIF 以及管理憑證等作業。如需有關這些指令的資訊，請參閱 `dsadm`(1M) 線上手冊與 `dsconf`(1M) 線上手冊。

`dpconf`、`dsconf`、`dsmig`、`dsccmon`、`dsccreg` 與 `dsccsetup` 是 LDAP 指令，因此您必須指定使用者連結 DN 與密碼，這些指令才可用以認證。而 `dpadm` 與 `dsadm` 指令則在實例檔案上運作。

本節包含下列有關目錄伺服器指令行工具的資訊：

- 第 52 頁的 「目錄伺服器指令的位置」
- 第 52 頁的 「設定 `dsconf` 的環境變數」
- 第 52 頁的 「`dsadm` 與 `dsconf` 的比較」
- 第 53 頁的 「使用 `dsadm` 與 `dsconf` 取得說明」
- 第 54 頁的 「使用 `dsconf` 修改配置特性」
- 第 55 頁的 「線上手冊」

目錄伺服器指令的位置

目錄伺服器指令行工具位於預設的安裝目錄中：

install-path/ds6/bin

安裝目錄視您的作業系統而定。第 26 頁的「預設路徑與指令位置」中列出了所有作業系統的安裝路徑。

設定 dsconf 的環境變數

dsconf 指令需要可使用環境變數預設的一些選項。在使用指令時若未指定選項，或未設定環境變數，則會使用預設值。您可以配置下列選項的環境變數：

-D *user DN*	使用者連結 DN。環境變數：LDAP_ADMIN_USER。預設值：cn=Directory Manager。
-w *password-file*	使用者連結 DN 的密碼檔案。環境變數： LDAP_ADMIN_PWF。預設值：密碼提示。
-h *host*	主機名稱。環境變數：DIRSERV_HOST。預設值：local host。
-p *LDAP-port*	LDAP 連接埠號碼。環境變數：DIRSERV_PORT。預設值：389。
-e, --unsecured	指定 dsconf 預設應開啓一個無障礙連線。環境變數：DIRSERV_UNSECURED。若未設定此變數，dsconf 預設會開啓安全連線。

如需詳細資訊，請參閱 dsconf(1M) 線上手冊。

dsadm 與 dsconf 的比較

下表顯示 dsadm 與 dsconf 指令的比較。

表 2–1 dsadm 與 dsconf 指令的比較

	dsadm 指令	dsconf 指令
說明	必須直接在本地主機上執行的管理指令。例如： ■ 啓動及停止伺服器 ■ 建立伺服器實例	可從遠端主機上執行的管理指令。例如： ■ 啓用複寫 ■ 設定快取大小

表 2–1 dsadm 與 dsconf 指令的比較 (續)

	dsadm 指令	dsconf 指令
注意	必須停止伺服器 (dsadm stop 與 dsadm info 指令除外)。 伺服器由伺服器實例路徑 (*instance-path*) 進行識別。 您必須具備伺服器實例路徑的作業系統存取權限。	伺服器必須正在執行中。 伺服器由主機名稱 (-h) 連接埠 (-p) 或 LDAPS 安全連接埠 (-P) 進行識別。 若未指定連接埠號碼，dsconf 會使用預設連接埠 (389 適用於 LDAP)。 您必須具備配置資料的 LDAP 存取權限，例如，具備使用者 cn=admin,cn=Administrators,cn=config 的身份。

使用 dsadm 與 dsconf 取得說明

如需有關如何使用 dsadm 和 dsconf 指令的完整資訊，請參閱 dsadm(1M) 線上手冊與 dsconf(1M) 線上手冊。

- 若要取得子指令的清單，請鍵入適當的指令：

```
$ dsadm --help

$ dsconf --help
```

- 若要取得如何使用子指令的相關資訊，請鍵入適當的指令：

```
$ dsadm subcommand --help

$ dsconf subcommand --help
```

使用 dsconf 修改配置特性

許多 dsconf 子指令可讓您檢視與修改配置特性。

- 若要列出目錄伺服器中所使用的配置特性，請鍵入：

  ```
  $ dsconf help-properties
  ```

- 若要尋找特定的特性，請搜尋說明特性的輸出。

  例如，若您使用 UNIX® 平台，同時想搜尋所有與參照相關的特性，請使用下列指令。

  ```
  $ dsconf help-properties | grep -i referral
  SER  referral-url                       rw  M  LDAP_URL | undefined
       Referrals returned to clients requesting a DN not stored in this
       Directory Server (Default: undefined)
  SUF  referral-mode                      rw     disabled|enabled|only-on-write
       Specifies how referrals are used for requests involving the suffix
       (Default: disabled)
  SUF  referral-url                       rw  M  LDAP_URL | undefined
       Server(s) to which updates are referred (Default: undefined)
  SUF  repl-rewrite-referrals-enabled     rw     on|off
       Specifies whether automatic referrals are overwritten (Default: off)
  ```

  請注意，特性會依目標物件群組化，例如尾碼 (SUF) 與伺服器 (SER)。rw 關鍵字表示特性爲可讀取與可寫入。M 關鍵字表示特性爲多重值。

- 若要檢視伺服器屬性，請使用詳細模式。以 UNIX 系統爲例，請鍵入：

  ```
  $ dsconf help-properties -v | grep -i referral-mode
  SUF  referral-mode    rw  disabled|enabled|only-on-write  nsslapd-state
    Specifies how referrals are used for requests involving the suffix
    (Default: disabled)
  ```

如需有關個別特性的更多資訊，請參閱該特性的線上手冊。這些線上手冊位於「Sun Java System Directory Server Enterprise Edition 6.3 Man Page Reference」中。

使用 dsconf 設定多重值特性

某些特定的目錄伺服器特性可容納多重值。指定這些值的語法如下：

```
$ dsconf set-container-prop -h host -p port container-name \
 property:value1 property:value2
```

例如，若要爲伺服器設定多個加密密碼，請使用下列指令：

```
$ dsconf set-server-prop -h host1 -p 1389 ssl-cipher-family:SSL_RSA_WITH_RC4_128_MD5 \
 ssl-cipher-family:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
```

若要將值增加至已包含值的多重值特性，請使用下列語法：

```
$ dsconf set-container-prop -h host -p port container-name property+:value
```

若要從已包含值的多重值特性移除值，請使用下列語法：

```
$ dsconf set-container-prop -h host -p port container-name property-:value
```

例如，在前述的方案中，若要將 SHA 加密密碼增加至密碼清單，請執行此指令：

```
$ dsconf set-server-prop -h host1 -p 1389 \
 ssl-cipher-family+:TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
```

若要從清單中移除 MD5 密碼，請執行此指令：

```
$ dsconf set-server-prop -h host1 -p 1389 ssl-cipher-family-:SSL_RSA_WITH_RC4_128_MD5
```

線上手冊

線上手冊可提供目錄伺服器中所使用之所有指令與屬性的說明。此外，線上手冊也會顯示如何在部署中使用指令的一些實用範例。

舊有工具

因爲向下相容性的原因，一般目錄伺服器工具隨附舊有工具。這些工具雖仍存在，但實際上已停用。

◆ ◆ ◆ 第 3 章

目錄伺服器實例與尾碼

本章說明如何建立及管理目錄伺服器實例與尾碼。另有多項目錄管理作業也是在尾碼層級上進行配置，但其相關內容另載於本書的其他章節。

本章包含下列主題：

- 第 57 頁的「建立伺服器實例與尾碼的快速程序」
- 第 57 頁的「建立及刪除目錄伺服器實例」
- 第 61 頁的「啓動、停止與重新啓動目錄伺服器實例」
- 第 62 頁的「建立尾碼」
- 第 64 頁的「停用或啓用尾碼」
- 第 64 頁的「設定參照並使尾碼成爲唯讀模式」
- 第 66 頁的「刪除尾碼」
- 第 66 頁的「壓縮尾碼」

建立伺服器實例與尾碼的快速程序

本章包含如何建立伺服器實例與尾碼的詳細資訊。如需快速建立目錄伺服器實例與尾碼以及匯入某些範例資料，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」中的「Server Instance Creation」。

建立及刪除目錄伺服器實例

本節說明如何建立及刪除目錄伺服器實例。

▼ 建立目錄伺服器實例

您必須使用指令行工具或瀏覽器介面目錄服務控制中心 (DSCC) 建立目錄伺服器實例，才能管理資料。在 DSCC 中，目錄伺服器實例通常簡稱爲「目錄伺服器」。

當您建立目錄伺服器實例時，您目錄伺服器所需的檔案與目錄將會建立在所指定的 *instance-path* 中。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

若使用 DSCC 建立新的伺服器實例，則可選擇從現有的伺服器複製部分或全部的伺服器配置設定。

1 **建立新的目錄伺服器實例，並設定實例路徑。**

```
$ dsadm create instance-path
```

系統會提示您爲此伺服器設定目錄管理員的密碼。

若要爲伺服器實例指定非預設的連接埠號碼或任何其他參數，請參閱 `dsadm`(1M) 線上手冊。

例如，若要在 `/local/ds` 目錄中建立新的實例，請使用此指令：

```
$ dsadm create /local/ds
Choose the Directory Manager password:
Confirm the Directory Manager password:
Use 'dsadm start /local/ds' to start the instance
```

2 **檢查伺服器實例是否已正確建立。**

```
$ dsadm info instance-path
```

例如：

```
$ dsadm info /local/ds1
Instance Path:          /local/ds1
Owner:                  user1(group1)
Non-secure port:        1389
Secure port:            1636
Bit format:             64-bit
State:                  Running
Server PID:             22555
DSCC url:               -
SMF application name:   -
Start at boot:          Disabled
Instance version:       D-A00
```

3 **(可選擇) 如果目錄伺服器已使用 Java Enterprise System 安裝程式或本機套裝軟體安裝進行了安裝，且作業系統提供有一套服務管理解決方案，則可將伺服器視為服務加以管理，如下表所示。**

作業系統	指令
Solaris 10	若您在 Sun 叢集環境中進行作業，請使用此指令： `dsadm enable-service --type CLUSTER` *instance-path resource-group* 其他情況： `dsadm enable-service --type SMF` *instance-path*
Solaris 9	若您在 Sun 叢集環境中進行作業，請使用此指令： `dsadm enable-service --type CLUSTER` *instance-path resource_group* 其他情況： `dsadm autostart` *instance-path*
Linux、HP-UX	`dsadm autostart` *instance-path*
Windows	`dsadm enable-service --type WIN_SERVICE` *instance-path*

4 啓動目錄伺服器。

```
$ dsadm start instance-path
```

備註 – 伺服器正在執行中，但不含任何資料或尾碼。使用 `dsconf` 建立尾碼。

5 (可選擇) 使用下列其中一個方法註冊伺服器實例：

- 存取 URL `https://host:6789`，並透過 DSCC 註冊伺服器。
- 使用指令 `dsccreg add-server`。
  如需詳細資訊，請參閱 `dsccreg`(1M) 線上手冊。

6 若要使用密碼策略，且目錄伺服器實例是獨立的，或屬於已遷移至 `DS6-only` 密碼策略模式的複寫拓樸，請將實例變更為該模式。

```
$ dsconf pwd-compat -h host -p port to-DS6-migration-mode

## Beginning password policy compatibility changes .
## Password policy compatibility changes finished.

Task completed (slapd exit code: 0).
$ dsconf pwd-compat -h host -p port to-DS6-mode

## Beginning password policy compatibility changes .
## Password policy compatibility changes finished.

Task completed (slapd exit code: 0).
```

▼ 刪除目錄伺服器實例

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 停止目錄伺服器。

```
$ dsadm stop instance-path
```

2 如果之前使用 DSCC 管理伺服器，請使用指令行取消註冊伺服器。

```
$ dsccreg remove-server /local/ds
Enter DSCC administrator's password:
/local/ds is an instance of DS
Enter password of "cn=Directory Manager" for /local/ds:
This operation will restart /local/ds.
Do you want to continue ? (y/n) y
Unregistering /local/ds from DSCC on localhost.
Connecting to /local/ds
Disabling DSCC access to /local/ds
Restarting /local/ds
```

如需詳細資訊，請參閱 `dsccreg`(1M) 線上手冊。

3 (可選擇) 如果之前是在服務管理解決方案中啓用伺服器實例，則需停止將伺服器視為服務進行管理。

作業系統	指令
Solaris 10	若您在 Sun 叢集環境中進行作業，請使用此指令： `dsadm disable-service --type CLUSTER` *instance-path* 其他情況： `dsadm disable-service --type SMF` *instance-path*
Solaris 9	若您在 Sun 叢集環境中進行作業，請使用此指令： `dsadm disable-service --type CLUSTER` *instance-path* 其他情況： `dsadm autostart --off` *instance-path*
Linux、HP-UX	`dsadm autostart --off` *instance-path*
Windows	`dsadm disable-service --type WIN_SERVICE` *instance-path*

4 刪除伺服器實例。

```
$ dsadm delete instance-path
```

注意 – 此指令會移除所有內容，包含資料庫和資料。

如果實例已啓用爲服務，或者若實例已在系統啓動時自動啓動，則 `dsadm delete` 需要超級使用者存取權。

啓動、停止與重新啓動目錄伺服器實例

若要從指令行啓動、停止或重新啓動伺服器，請分別使用 `dsadm start`、`dsadm stop` 與 `dsadm restart` 指令。

備註 – 若所停止及重新啓動的目錄伺服器實例，在依配置保存項目的記憶體中含有大量快取，這些快取將需要一段時間進行回存。在快取回存時，實例的回應速度將會減緩。

這些指令必須由建立目錄伺服器的相同 UID 與 GID 執行，或由超級使用者執行。例如，若目錄伺服器以 `user1` 的身份執行，您即應以 `user1` 的身份執行 `start`、`stop` 與 `restart` 等公用程式。

備註 – 在 Solaris 上，角色型存取控制可讓您以超級使用者以外的使用者身份執行目錄伺服器。

▼ 啓動、停止與重新啓動目錄伺服器

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。但是，這不會套用至啓用與停用服務管理的步驟。在啓動及停止目錄伺服器時，必須在指令行上執行服務管理的啓用與停用作業。

- **若要啓動、停止或重新啓動目錄伺服器，請執行下列其中一項：**
  - 若要啓動伺服器，請鍵入：

    ```
    $ dsadm start instance-path
    ```

    例如，若要以實例路徑 `/local/ds` 啓動伺服器，請使用此指令：

    ```
    $ dsadm start /local/ds
    ```
  - 若要停止伺服器，請鍵入：

    ```
    $ dsadm stop instance-path
    ```

例如：

```
$ dsadm stop /local/ds
```

- 若要重新啓動伺服器，請鍵入：

```
$ dsadm restart instance-path
```

例如：

```
$ dsadm restart /local/ds
```

建立尾碼

在建立目錄伺服器實例後，必須爲伺服器的目錄資訊樹狀結構 (DIT) 建立一或多個尾碼。DIT 包含伺服器中所有的項目，各個項目有其各自的辨別名稱 (DN)。DN 的階層式特性可建立分支與尾節點，以建構樹狀結構中的資料。管理員會以尾碼與子尾碼的形式定義及管理 DIT。DSCC 可對建立及管理這些元素的作業進行控制。此外，也可以使用指令行工具。

如需有關架構目錄資料的概念性資訊，以及有關尾碼的一般資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」。

如下列程序所述，您可以使用 `dsconf create-suffix` 指令在目錄中建立尾碼配置。由於根尾碼與子尾碼皆以相同的方式經由內部管理，因此從指令行建立這兩種尾碼的程序也幾乎是相同的。此程序說明僅適用於必要選項的 `dsconf create-suffix` 指令。如需有關此指令之其他選項的更多資訊，請參閱 `dsconf`(1M) 線上手冊或執行下列指令：

```
$ dsconf create-suffix --help
```

配置項目可由任何管理使用者加以建立。但若是尾碼的頂端項目，則**必須**由目錄管理員建立，或以 `cn=admin,cn=Administrators,cn=config` 等目錄伺服器管理員的身份所建立。

▼ 建立尾碼

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

如果使用 DSCC 建立新的尾碼，可選擇複製現有尾碼的部分或所有尾碼配置設定。

1 建立根尾碼。

確定伺服器正在執行中，然後鍵入此指令：

```
$ dsconf create-suffix -h host -p port suffix-DN
```

其中 *suffix-DN* 是新尾碼的完整 DN。就根尾碼而言，一般慣例是使用網域元件 (dc) 命名屬性。

例如，若要建立 DN `dc=example,dc=com` 的尾碼，請使用此指令：

```
$ dsconf create-suffix -h host1 -p 1389 dc=example,dc=com
```

此指令會以下列方式建立新尾碼：

- 建立根尾碼的頂端 (或基底) 項目。
- 建立尾碼與資料庫在 `cn=config` 中的配置項目。
- 預設資料庫名稱取決於尾碼 DN。

如需有關所有尾碼的資訊 (包括已建立的新尾碼)，請使用此指令：

```
$ dsconf list-suffixes -h host -p port -v
```

`-v` 選項會顯示詳細模式，可列出尾碼上有多少項目，以及所有的複寫資訊。

備註 – 若有多個目錄伺服器實例，請使用 `-h` *host name* 與 `-p` *port number* 選項，指定尾碼所屬的伺服器實例。

若要爲資料庫檔案指定非預設路徑，請使用 `-L` 選項。您可以在後續的階段中變更尾碼資料庫路徑。若要執行此項作業，請使用 `dsconf set-suffix-prop` *suffix-DN* `db-path:`*new-db-path* 指令，然後停止伺服器，以手動方式移動資料庫檔案，再重新啓動伺服器。

若要檢視在建立尾碼時所能使用的所有選項，請參閱 `dsconf`(1M) 線上手冊。

---

2 請視需要建立子尾碼：

```
$ dsconf create-suffix -h host -p port subSuffix-DN
```

接著將子尾碼附加到根尾碼中。

```
$ dsconf set-suffix-prop -h host -p port subSuffix-DN parent-suffix-dn:parentSuffix-DN
```

其中，*parentSuffix-DN* 的值必須與前一個步驟中的 *suffix-DN* 值相同。子尾碼的 *suffix-DN* 中包含了子尾碼的相關辨別名稱 (RDN) 及其父系尾碼的 DN。

例如，若要建立子尾碼 `ou=Contractors,dc=example,dc=com`，並將子尾碼附加至根尾碼，請鍵入：

```
$ dsconf create-suffix -h host1 -p 1389 ou=Contractors,dc=example,dc=com
$ dsconf set-suffix-prop -h host1 -p 1389 ou=Contractors,dc=example,dc=com \
 parent-suffix-dn:dc=example,dc=com
```

當此項目增加到目錄時，伺服器的資料庫模組會自動在下列目錄中建立資料庫檔案：

instance-path`/db/`*database-name*

其中，*database-name* 是從部分尾碼自動建立的名稱。以前述範例爲例，*database-name* 將是 `Contractors`

3 **(可選擇) 以資料初始化尾碼。請參閱****第 193 頁的 「初始化尾碼」**。

停用或啓用尾碼

您有時可能需要停用尾碼以進行維護，或基於安全性考量而停用其內容。若停用尾碼，將使伺服器無法讀取或寫入尾碼的內容，以回應任何用戶端作業。當您停用尾碼後，即不再具備該尾碼的存取權，而參照模式將自動設爲停用。

▼ 停用尾碼後再予以啓用

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **停用尾碼。**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN enabled:off
```

備註 – 您無法停用已啓用複寫的尾碼，因爲在複寫的尾碼中，大部分的特性皆由複寫機制所決定。

---

2 **啓用尾碼。**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN enabled:on
```

設定參照並使尾碼成為唯讀模式

若要限制尾碼的存取，但不完全停用尾碼，您可以修改存取權限以允許唯讀存取。在此情況下，必須定義其他伺服器的寫入作業參照。您也可以同時拒絕讀取與寫入存取，並定義所有作業對尾碼的參照。

參照也可用於將用戶端應用程式暫時指向不同的伺服器。例如，在備份尾碼的內容時，可以增加其他尾碼的參照。

若尾碼是複寫環境中的用戶，複寫機制將可決定參照設定的值。雖然可以手動修改參照設定，但參照仍將會在下次複寫更新時遭到覆寫。如需有關設定複寫參照的資訊，請參閱第 225 頁的 「執行進階用戶配置」。

參照屬於標籤式 URL，亦即可選擇性地在尾端加上空格字元與標籤的 LDAP URL。例如：

```
ldap://phonebook.example.com:389/
```

或是：

```
ldap://phonebook.example.com:389/ou=All%20People,dc=example,dc=com
```

由於空格字元有其意義，因此在參照的 URL 部分中，任何空格字元均必須使用 `%20` 換碼。

▼ 設定參照，使尾碼變成唯讀模式

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **設定參照 URL。**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN referral-url:LDAP-URL
   ```

   其中，*LDAP-URL* 是含有目標之主機名稱、連接埠號碼與 DN 的有效 URL。

   例如：

   ```
   $ dsconf set-suffix-prop -h host1 -p 1389 dc=example,dc=com \
    referral-url:ldap://phonebook.example.com:389/
   ```

   您可以指定任何數量的 LDAP URL。

2. **設定參照模式，使尾碼變成唯讀模式。**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN referral-mode:only-on-write
   ```

   若要讓尾碼無法供讀取與寫入作業使用，並且對所有請求傳回參照，請將 `referral-mode` 設為 `enabled`。

3. **此指令一旦成功執行，尾碼即會變成唯讀或無法存取，並可傳回參照的模式。**

4. **(可選擇) 當尾碼無法使用後，停用參照即可讓尾碼回復為可讀寫的狀態。**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN referral-mode:disabled
   ```

   參照停用時，尾碼即會自動變成可讀寫的狀態，除非您已將尾碼的 `enabled` 特性設為 `off`，而停用了尾碼本身。

刪除尾碼

刪除尾碼會從 DIT 移除其整個分支。

備註 – 當您刪除尾碼時，會將其所有資料項目從目錄中永久移除。如此也會移除所有尾碼配置資訊，包含其複寫配置。

您無法刪除了父系尾碼，但將其子尾碼保留在 DIT 中做爲新的根尾碼。若要刪除整個含有子尾碼的分支，必須同時刪除已刪除之父系的子尾碼，及其可能的子尾碼。

▼ 刪除尾碼

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **移除尾碼配置項目：**

  ```
  $ dsconf delete-suffix -h host -p port [subSuffix-DN] suffix-DN
  ```

  此指令會從 *suffix-DN* 上的基底項目開始，移除伺服器中的尾碼。尾碼在目錄中將不會再顯示出來或供存取。

壓縮尾碼

Directory Server 6.3 支援離線壓縮尾碼。此發行版本不支援線上壓縮。如果有可用的儲存空間，壓縮尾碼可重新組織資料庫金鑰，而減少資料庫的大小。

▼ 離線壓縮尾碼

停止伺服器並備份資料庫，再執行此作業。

- **壓縮必要的尾碼。**

  ```
  $ dsadm repack instance-path suffix-dn
  ```

  所有與指定尾碼相關的 `.db3` 檔案都會隨即壓縮。

  若搭配 `-b` 選項使用此指令，則可指定後端資料庫名稱，而非尾碼 DN。至少必須指定一個尾碼或一個後端伺服器。

◆ ◆ ◆ 第 4 章

4 目錄伺服器配置

本章說明如何配置目錄伺服器。您可以使用 dsconf 指令 (請參閱 dsconf(1M) 線上手冊)。

您也可以使用目錄服務控制中心 (DSCC)，此方法較多人使用。 DSCC 會在配置程序中做進一步的檢查，以儘可能降低錯誤。此外，DSCC 也可讓您將某個伺服器實例的配置複製到另一個伺服器實例上。如需有關使用 DSCC 的更多資訊，請參閱 DSCC 線上說明。

顯示目錄伺服器實例的配置

若要顯示目錄伺服器實例的配置，請執行 dsconf info。

```
$ dsconf info -h host -p port
Instance path   :  instance path
Global State    :  read-write
Host Name       :  host
Port            :  port
Secure port     :  secure port
Total entries   :  20844

Suffixes        :  suffix-DN

Dest. Servers   :  host:port

On-Going Tasks  :  import
Finished Tasks  :  backup
```

以上輸出假設您已建立尾碼以及與目標伺服器的複寫協議。它也會顯示進行中的匯入作業與已完成的備份作業。

使用 DSCC 修改配置

修改配置的建議方法爲使用 DSCC。此瀏覽器介面提供了作業型控制，可協助您快速而有效地設定配置。使用 DSCC 可讓您修改某部伺服器的配置設定，並將其複製到其他伺服器上。此外，DSCC 介面亦可爲您管理配置的複雜性與相互依賴性。如需以 DSCC 修改配置的詳細程序，請參閱 DSCC 線上說明。

從指令行修改配置

您可以撰寫採用指令行工具的程序檔，將配置作業自動化。

在指令行上使用 `dsconf` 指令修改配置。此指令會使用 LDAP 修改 `cn=config` 子樹狀結構。如需有關 `dsconf` 的更多資訊，請參閱第 51 頁的 「目錄伺服器指令行工具」。

對於無法以 `dsconf` 執行的作業，請改用 `ldapmodify` 指令。

備註 – 若要使用 `dsconf set-server-prop` 指令修改伺服器配置特性，您必須了解所能修改的特性及其預設值。使用下列指令可顯示所有特性的說明：

```
$ dsconf help-properties -v
```

搜尋所需項目的特性說明。以 UNIX 平台爲例，鍵入下列內容即可搜尋記憶體快取特性：

```
$ dsconf help-properties -v | grep cache
```

如需有關 `cn=config` 中配置項目的更多資訊，以及所有配置項目與屬性的完整描述 (包含所允許值的範圍)，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」。

修改 `dse.ldif` 檔案

目錄伺服器會將其所有配置資訊儲存於此檔案：

instance-path`/config/dse.ldif`

注意 – 直接編輯 `dse.ldif` 檔案內容以修改配置的做法很容易造成錯誤，不建議使用。但若您選擇手動編輯此檔案，請在編輯檔案前先停止伺服器，再於編輯完成後重新啓動。

`dse.ldif` 檔案使用 LDAP 資料互換格式 (LDIF)。LDIF 以文字表示項目、屬性及其值，並且是 RFC 2849 (`http://www.ietf.org/rfc/rfc2849`) 中所述的標準格式。

dse.ldif 檔案中的目錄伺服器配置由下列項目構成：

- cn=config 項目的屬性與值。
- 在子樹狀結構中位於 cn=config 下的所有項目及其屬性與值。
- 根項目 ("") 與 cn=monitor 項目的物件類別與存取控制指令。這些項目的其他屬性會由伺服器產生。

  只有具有目錄伺服器實例的系統使用者有權讀取及寫入檔案。

  目錄伺服器可透過 LDAP 讓所有配置設定成爲可讀寫的內容。依預設，具有授權的任何人皆可讀取目錄的 cn=config 分支，但只有「目錄管理員」(cn=Directory Manager) 與 cn=Administrators,cn=config 下的管理使用者可對其進行寫入。管理使用者可檢視及修改配置項目，與任何其他目錄項目並無二致。

  請勿於 cn=config 項目下建立非配置項目，因爲這些項目將儲存在 dse.ldif 檔案中，而此檔案並非像一般項目一樣屬於可延伸性高的資料庫。因此，如果 cn=config 下儲存了許多項目 (特別是可能經常更新的項目)，則效能可能降低。但若將「複寫管理員」(供應者連結 DN) 項目等特殊的使用者項目儲存在 cn=config 下，將可能有助於配置資訊的集中化。

配置管理使用者

目錄伺服器含有預設管理使用者、「目錄管理員」與 cn=admin,cn=Administrators,cn=config 使用者。這兩種使用者具有相同的存取權限，但 cn=admin,cn=Administrators,cn=config 須受 ACI 支配。

本節說明如何建立具有超級使用者存取權的管理使用者，以及如何配置「目錄管理員」。

▼ 建立具有超級使用者存取權的管理使用者

若要建立與 cn=admin,cn=Administrators,cn=config 具有相同權限的新管理使用者，請在 cn=Administrators,cn=config 群組中建立新的使用者。此群組中的所有使用者均須受全域 ACI 支配，該全域 ACI 可允許與「目錄管理員」相同之存取權限。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **建立新的管理使用者**。

  例如，若要建立新的使用者 cn=Admin24,cn=Administrators,cn=config，請鍵入：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=admin24,cn=Administrators,cn=config
changetype: add
objectclass: top
objectclass: person
```

```
userPassword: password
description: Administration user with the same access rights as Directory Manager.
```

`-D` 與 `-w` 選項分別可爲有權建立此項目的使用者指定連結 DN 與密碼。

▼ 配置目錄管理員

「目錄管理員」是具有特權的伺服器管理員，相當於 UNIX 系統上的 `root` 使用者。存取控制不適用於「目錄管理員」。

大部分的管理作業均不需用到「目錄管理員」。您可以改用使用者 `cn=admin,cn=Administrators,cn=config`，或任何您在 `cn=Administrators,cn=config` 下建立的其他使用者。唯一需要「目錄管理員」的作業，是變更根 ACI 以及複寫的疑難排解作業，例如修復複寫與搜尋標記。

您可以變更目錄管理員 DN 與密碼，以及建立可從中自動讀取密碼的檔案。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 尋找現有的目錄管理員 DN。

```
$ dsconf get-server-prop -h host -p port root-dn
root-dn:cn=Directory Manager
```

2 在必要時修改「目錄管理員」設定。

- 若要修改目錄管理員 DN，請鍵入：

  ```
  $ dsconf set-server-prop -h host -p port root-dn:new-root-dn
  ```

  若目錄管理員 DN 中有空格，請使用引號。例如：

  ```
  $ dsconf set-server-prop -h host1 -p 1389 root-dn:"cn=New Directory Manager"
  ```

- 若要變更目錄管理員密碼，請鍵入：

  ```
  $ dsconf set-server-prop -h host -p port root-pwd:new-root-dn-password
  ```

  若因安全性考量而不以指令行引數的形式傳送純文字密碼，請建立用以設定密碼的暫存檔。

  ```
  $ echo password > /tmp/pwd.txt
  ```

  此檔案只能讀取一次，您必須儲存密碼以供日後使用。設定伺服器根密碼檔案特性。

  ```
  $ dsconf set-server-prop -h host -p port root-pwd-file:/tmp/pwd.txt
  ```

  此指令會提示伺服器讀取密碼檔案。設定密碼檔案特性後，請移除暫存密碼檔案。

```
$ rm /tmp/pwd.txt
```

保護配置資訊

根目錄伺服器項目 (以零長度 DN "" 進行基底物件搜尋所傳回的項目) 與 cn=config、cn=monitor 及 cn=schema 下的子樹狀結構，含有「目錄伺服器」自動產生的存取控制指令 (ACI)。這些 ACI 可用以決定目錄項目的使用者權限。這些 ACI 足夠評估作業之用。但運用在生產環境部署時，就必須評估您的存取控制需求，並設計您自己的存取控制。

若您基於安全性考量而要隱藏一或多個其他子樹狀結構的存在，並保護您的配置資訊，您必須將其他 ACI 置於 DIT 上。

- 將 ACI 屬性放置於您所要隱藏之子樹狀結構基底上的項目中。
- 將 ACI 放置於 namingContexts 屬性的根 DSE 項目中。名爲 namingContexts 的根 DSE 項目屬性，含有各個目錄伺服器資料庫的基底 DN 清單。
- 將 ACI 置於 cn=config 與 cn=monitor 子樹狀結構中。子樹狀結構 DN 也會儲存在 cn=config 與 cn=monitor 下的對映樹狀結構項目中。

如需有關建立 ACI 的更多資訊，請參第 7 章。

配置 DSCC

本節提供下列有關配置 DSCC 的資訊：

- 第 71 頁的 「變更共用代理程式容器的連接埠號碼」
- 第 72 頁的 「重設目錄服務管理員密碼」
- 第 72 頁的 「延伸 DSCC 階段作業自動逾時延遲」
- 第 73 頁的 「配置 DSCC 的容錯移轉」
- 第 74 頁的 「DSCC 的疑難排解」

▼ 變更共用代理程式容器的連接埠號碼

預設的共用代理程式容器連接埠號碼爲 11162。共用代理程式容器會將 DSCC 代理程式連接埠定義爲 jmxmp-connector-port。若基於管理原因，DSCC 代理程式與共用代理程式容器必須使用不同的連接埠號碼，請使用下列程序。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 以超級使用者的身份，驗證 jmxmp-connector-port 現有的連接埠號碼。

```
$ su
Password:
# cacaoadm list-params
```

```
...
jmxmp-connector-port=11162
...
```

2 **變更 DSCC 代理程式連接埠號碼。**

變更 DSCC 代理程式連接埠號碼時，必須停止共用代理程式容器。

```
# cacaoadm stop
# cacaoadm set-param jmxmp-connector-port=new-port
# cacaoadm start
```

如需此指令的位置，請參閱第 28 頁的 「指令位置」。

3 **在 DSCC 中取消註冊您的伺服器，然後使用新的 DSCC 代理程式連接埠號碼予以重新註冊。**

此外，當您建立新的伺服器時，您必須指定非預設的 DSCC 代理程式連接埠號碼。

▼ 重設目錄服務管理員密碼

若要重設目錄服務管理員密碼，請依本程序中所述使用 DSCC。

1 **如第 47 頁的 「存取 DSCC」中所述存取 DSCC。**

2 **按一下 [設定] 標籤，然後選擇 [目錄服務管理員]。**

3 **按一下要變更密碼的「目錄服務管理員」名稱。**

4 **在特性畫面中，輸入新的密碼。**

在 [確認密碼] 欄位中再次鍵入新密碼，以進行確認。按一下 [確定] 以儲存變更。

▼ 延伸 DSCC 階段作業自動逾時延遲

經過一段時間後，您的 DSCC 階段作業將會逾時，並將您登出 DSCC。使用此程序可延伸逾時延遲。請注意，此程序可延伸 DSCC 的逾時以及 Sun Java Web Console 中所有其他應用程式的逾時。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 **以超級使用者身份延長逾時延遲。**

```
# wcadmin add -p -a ROOT session.timeout.value=mm
```

其中 *mm* 是逾時之前的時間，以分鐘數計算。

例如，若要將逾時設爲兩小時，請鍵入：

```
$ su
Password:
# wcadmin add -p -a ROOT session.timeout.value=120
Set 1 properties for the ROOT application.
# wcadmin list -p
Shared service properties (name, value):
    session.timeout.value  120
    ...
```

2 **重新啓動 Sun Java Web Console。**

```
# smcwebserver restart
Shutting down Sun Java(TM) Web Console Version 3.0.2 ...
Starting Sun Java(TM) Web Console Version 3.0.2 ...
The console is running.
```

如需這些指令的位置，請參閱第 28 頁的 「指令位置」。

配置 DSCC 的容錯移轉

DSCC 會顯示您已在 DSCC 中註冊的伺服器。

若先前安裝 DSCC 的機器故障，您可以改在其他機器上安裝 DSCC，然後重新註冊伺服器。但這麼做可能很花時間。若想要透過 DSCC 立即存取伺服器，您可以配置 DSCC 容錯移轉。

若要配置 DSCC 容錯移轉，請考量下列注意事項：

- 已註冊之伺服器的所有資訊，均儲存於 DSCC 登錄中。此登錄爲目錄伺服器實例。您可以使用管理指令 `dsadm` 與 `dsconf` 管理登錄。
- DSCC 登錄具有下列預設特質：

  伺服器實例　Solaris — `/var/opt/SUNWdsee/dscc6/dcc/ads`

  Linux 與 HP-UX — `/var/opt/sun/dscc6/dcc/ads`

  Windows — `C:\Program Files\Sun\DSEE\var\dscc6\dcc\ads`

  尾碼　`cn=dscc`

  連接埠　LDAP `3998`、LDAPS `3999`
- 在兩部或更多機器上安裝 DSCC 後，即可設定 DSCC 登錄尾碼間的複寫。請依第 11 章中所述，使用複寫指令行程序。此外，如需設定簡單複寫配置的範例，請參閱 `dsconf`(1M) 線上手冊。

設定複寫後，您即可從不同的機器存取在 DSCC 中註冊的相同伺服器。例如，若您設定了 `host1` 與 `host2` 上不同 DSCC 登錄尾碼間的複寫，您即可在 `https://host1:6789` 或 `https://host2:6789` 上使用 DSCC 管理相同的伺服器。而在主機故障時，將可從其他主機存取 DSCC。

DSCC 的疑難排解

如需有關 DSCC 的疑難排解資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」中的「To Troubleshoot Directory Service Control Center Access」。

變更目錄伺服器連接埠號碼

您可以使用 DSCC 或使用 `dsconf set-server-prop` 指令，修改使用者目錄伺服器的 LDAP 連接埠號碼或 LDAPS 安全連接埠號碼。

若變更了連接埠號碼，請注意下列事項：

- 若設定了非專用連接埠號碼，且目錄伺服器安裝在其他使用者可存取的機器上，該連接埠即有可能因其他應用程式曝露於被劫持的風險下。換言之，其他應用程式可連結至相同的位址/連接埠對。此惡意應用程式可能會接著處理原本以目錄伺服器爲目標的請求。也就是說，惡意應用程式可用以擷取認證程序中所使用的密碼，以變更用戶端請求或伺服器回應，或產生阻絕服務攻擊。若要避免此安全性風險，請使用 `listen-address` 或 `secure-listen-address` 特性來指定目錄伺服器偵聽所在的介面 (位址)。

若您使用指令行變更了連接埠號碼，請注意下列事項：

- 若定義於其他伺服器上的複寫協議中參照了目錄伺服器，複寫協議即必須進行更新以使用新的連接埠號碼。
- 若您先前使用 DSCC 管理伺服器，則在變更連接埠號碼後，將暫時無法檢視伺服器。若要再次檢視伺服器，必須取消註冊伺服器，然後使用新的連接埠號碼在 DSCC 中予以重新註冊。

▼ 修改連接埠號碼、啓用連接埠以及停用連接埠

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

備註 – 完成修改之後必須重新啓動伺服器，變更才會生效。

---

1 **驗證連接埠現有的設定**。

```
$ dsconf get-server-prop -h host -p port port-type
```

其中 *port-type* 爲下列其中一項：

`ldap-port`	LDAP 預設連接埠
`ldap-secure-port`	LDAPS 安全連接埠
`dsml-port`	DSML 預設連接埠
`dsml-secure-port`	DSML 安全連接埠

例如，若要顯示 LDAPS 安全連接埠，請鍵入：

```
$ dsconf get-server-prop -h host1 -p 2501 ldap-secure-port
Enter "cn=Directory Manager" password:
ldap-secure-port : 2511
```

若傳回的結果爲整數，表示連接埠已啓用。若傳回的結果爲 `disabled`，表示連接埠已停用。

備註 – 您也可以使用 `dsadm` 列出 LDAP 預設連接埠與 LDAPS 安全連接埠。

---

2 **必要時，請修改連接埠號碼或啓用連接埠**。

```
$ dsconf set-server-prop -h host -p port port-type:new-port
```

例如，若要將 LDAP 連接埠號碼從 1389 變更爲 1390，請使用此指令：

```
$ dsconf set-server-prop -h host1 -p 1389 ldap-port:1390
```

若要在連接埠號碼 2250 上啓用 DSML 安全連接埠，請使用此指令：

```
$ dsconf set-server-prop -h host1 -p 1389 dsml-secure-port:2250
```

3 **必要時，請停用連接埠**。

```
$ dsconf set-server-prop -h host -p port port-type:disabled
```

例如，若要停用 DSML 安全連接埠，請使用此指令：

```
$ dsconf set-server-prop -h host1 -p 1389 dsml-secure-port:disabled
```

配置 DSML

除了以「簡易目錄存取協定 (LDAP)」處理請求以外，目錄伺服器也會回應以目錄服務標記語言版本 2 (DSMLv2) 傳送的請求。DSML 是另一個可供用戶端編碼目錄作業的方法。伺服器處理 DSML 就像處理任何其他請求一般，所使用之存取控制與安全性功能均相同。DSML 處理允許其他多種用戶端類型存取您的目錄內容。

目錄伺服器支援透過超文字傳輸協定 (HTTP/1.1) 使用 DSMLv2，而在傳輸 DSML 內容時，會使用簡易物件存取協定 (SOAP) 1.1 版做爲程式設計協定。如需有關這些協定的更多資訊以及 DSML 請求的範例，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 10 章「Directory Server DSMLv2」。

本節包含下列主題：

- 第 76 頁的 「啓用 DSML-over-HTTP 服務」
- 第 77 頁的 「停用 DSML-over-HTTP 服務」
- 第 78 頁的 「DSML 身份識別對映」

▼ 啓用 DSML-over-HTTP 服務

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **將 DSML 模式設為** on。

   ```
   $ dsconf set-server-prop -h host -p port dsml-enabled:on
   ```

2. **設定安全 DSML 連接埠**。

   ```
   $ dsconf set-server-prop -h host -p port dsml-secure-port:port
   ```

3. **設定非安全 DSML 連接埠**。

   ```
   $ dsconf set-server-prop -h host -p port dsml-port:port
   ```

   此連接埠預設會設定爲 `disabled`

4. **重新啓動伺服器**。

   ```
   $ dsadm restart instance-path
   ```

接下來的步驟 根據您所定義的參數與屬性值，DSML 用戶端可使用下列 URL 傳送請求至此伺服器：

`http://`*host*`:`*DSML-port*`/` *relative-URL*

`https://`*host*`:`*secure-DSML-port* `/`*relative-URL*

備註 – 您可以使用 `dsml-relative-root-url` 特性讀取與設定 *relative-URL*。

▼ 停用 DSML-over-HTTP 服務

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **將 DSML 模式設為 `off`。**

   ```
   $ dsconf set-server-prop -h host -p port dsml-enabled:off
   ```

2. **將安全 DSML 連接埠設為 `disabled`。**

   ```
   $ dsconf set-server-prop -h host -p port dsml-secure-port:disabled
   ```

3. **重新啓動伺服器。**

   ```
   $ dsasm restart instance-path
   ```

▼ 配置 DSML 安全性

您可以配置接受 DSML 請求時所需的安全性層級。若要執行此作業，您必須配置 DSML 用戶端認證。

- **設定 DSML 用戶端認證模式。**

  ```
  $ dsconf set-server-prop -h host -p port dsml-client-auth-mode:dsml-mode
  ```

  `dsml-client-auth-mode` 特性預設會設定爲 `client-cert-first`。

  dsml-mode 可爲下列其中之一：

  - `http-basic-only` - 此爲預設值。伺服器會使用「HTTP 授權」標頭的內容，尋找可對映至目錄中之項目的使用者名稱。此程序及其配置會透過 SSL 進行加密，但不需要用戶端憑證。相關說明請參閱第 78 頁的 「DSML 身份識別對映」。
  - `client-cert-only` - 伺服器會使用來自用戶端憑證中的憑證，進行用戶端的身份識別。使用此值時，所有 DSML 用戶端均必須使用安全 HTTPS 連接埠傳送 DSML 請求及提供憑證。伺服器會檢查用戶端憑證是否符合目錄中的項目。如需更多資訊，請參閱第 6 章。
  - `client-cert-first` - 伺服器會嘗試先以用戶端憑證 (若有的話) 進行用戶端認證。否則，伺服器將會使用「授權」標頭的內容進行用戶端認證。

若 HTTP 請求未提供任何憑證與「授權」標頭，伺服器即會以匿名連結執行 DSML 請求。下列情況也會使用匿名連結：

- 在指定 `client-cert-only` 時，用戶端提供了有效的「授權」標頭，但不具憑證。
- 在指定 `http-basic-only` 時，用戶端提供了有效的憑證，但不具「授權」標頭。

無論用戶端認證方法爲何，若已提供憑證，但不符合項目，或已指定「HTTP 授權」標頭，但此標頭無法對映至使用者項目，DSML 請求即會遭拒絕，並產生錯誤訊息 403：「禁止」。

DSML 身份識別對映

執行不具憑證的基本認證時，目錄伺服器會使用名爲**身份識別對映**的機制，判斷在接受 DSML 請求時所應使用的連結 DN。此機制會從 HTTP 請求的「授權」標頭中擷取資訊，以判斷用以連結的身份識別。

DSML/HTTP 的預設身份識別對映，由您伺服器配置中的下列項目所指定。

```
dn: cn=default,cn=HTTP-BASIC,cn=identity mapping,cn=config
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
cn: default
dsSearchBaseDN: ou=people
dsSearchFilter: (uid=${Authorization})
```

此配置表示伺服器應使用 HTTP 使用者 ID，做爲目錄伺服器尾碼中所儲存之 DN 的 `uid` 值。例如，若 HTTP 使用者爲 `bjensen`，伺服器就會嘗試使用 DN `uid=bjensen,ou=people` 執行連結。

爲使對映正常運作，您必須完成 `dsSearchBaseDN` 的值。例如，您可以將 `dsSearchBaseDN` 的值變更爲 `ou=people,dc=example,dc=com`。接著，若 HTTP 使用者爲 `bjensen`，伺服器就會嘗試使用 DN `uid=bjensen,ou=people,dc=example,dc=com` 執行連結。

```
dn: cn=default,cn=HTTP-BASIC,cn=identity mapping,cn=config
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
cn: default
dsSearchBaseDN: ou=people,dc=example,dc=com
dsSearchFilter: (uid=${Authorization})
```

在對映項目屬性 `dsSearchFilter` 內，您可以使用 `${`*header*`}` 格式的預留位置，其中 *header* 是 HTTP 標頭的名稱。

以下是 DSML 對映中最常使用的標頭。

`${Authorization}`	此字串會由「HTTP 授權」標頭中所含的使用者名稱取代。授權標頭中含有使用者名稱及其密碼，但在此預留位置中只有使用者名稱會遭取代。
`${From}`	此字串會由「HTTP 寄件者」標頭中可能包含的電子郵件位址取代。
`${host}`	此字串會由 DSML 請求之 URL 的主機名稱與連接埠號碼取代，此亦即伺服器的主機名稱與連接埠號碼。

若要讓 DSML 請求執行不同類型的身份識別對映，請爲 HTTP 標頭定義新的身份識別對映。

▼ 為 HTTP 標頭定義新的身份識別對映

1. **為此協定編輯預設 DSML-over-HTTP 身份識別對映，或建立自訂對映。**

   對映項目必須位於 `cn=HTTP-BASIC,cn=identity mapping,cn=config` 項目下。

   在指令行上使用 `ldapmodify` 指令增加此項目，如第 86 頁的 「使用 ldapmodify 增加項目」中所述。

2. **重新啓動目錄伺服器，使您的新對映生效。**

   自訂對映會先進行評估。若沒有成功的自訂對映，則會接著評估預設對映。若所有對映皆無法判斷 DSML 請求的連結 DN，即會禁止並拒絕 DSML 請求 (錯誤 403)。

將伺服器設為唯讀

您目錄中的每個尾碼均可單獨設爲唯讀模式，並可傳回特定的參照 (若已定義)。目錄伺服器針對適用於所有尾碼同時可傳回全域參照的伺服器 (若已定義)，也提供了唯讀模式。

設計伺服器唯讀模式的目的，是要讓管理員在執行重新編製尾碼索引之類的作業時，防止目錄內容遭到修改。因此，伺服器唯讀模式並不適用於下列配置分支：

- `cn=config`
- `cn=monitor`
- `cn=schema`

無論唯讀設定爲何，這些分支應一律受存取控制指令 (ACI) 的保護，以免遭到管理員以外的使用者修改 (請參閱第 7 章)。全域唯讀模式可防止目錄中所有其他尾碼的更新作業，包括「目錄管理員」所初始化的更新作業在內。

唯讀模式啓用後，也會使尾碼的複寫作業中斷。主伺服器複本將不再有任何變更需複寫，即使它繼續複寫唯讀模式啓用前所做的任何變更，仍是如此。用戶複本必須等到唯讀模式停用後，才能接收更新。多重主伺服器複寫環境中的主伺服器不具任何需複寫的變更，且無法從其他主伺服器接收更新。

▼ 啓用或停用伺服器唯讀模式

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **啓用全域唯讀模式**。

```
$ dsconf set-server-prop -h host -p port read-write-mode:read-only
```

2. **當您準備好時，即可停用唯讀模式**。

```
$ dsconf set-server-prop -h host -p port read-write-mode:read-write
```

配置記憶體

本節提供管理不同記憶體類型的相關資訊。如需不同快取類型的描述以及有關快取調校的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 5 章「Directory Server Data Caching」。

填充快取

填充快取是指將資料填入快取中，使後續的目錄伺服器運作方式反映出正常的作業效能，而不是不穩定的效能。在進行標準檢查時若產生可重複顯示的結果，以及在測量與分析可能的最佳化程度時，填充快取都有其效用。

請儘可能避免主動填充快取。在測量效能之前，請讓快取透過用戶端與目錄伺服器之間正常或一般的互動方式進行填充。

如需填充資料庫快取的工具，請至 `http://www.slamd.com`。

▼ 修改資料庫快取

注意 – 修改快取可能會嚴重影響伺服器效能。修改快取時請多加留意。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **取得目前的資料庫快取層級。**

   ```
   $ dsconf get-server-prop -h host -p port db-cache-size
   ```

2. **變更資料庫快取層級。**

   ```
   $ dsconf set-server-prop -h host -p port db-cache-size:size
   ```

   其中 *size* 可以 GB (G)、MB (M)、KB (k) 或位元組 (b) 表示。所指定的大小必須是受機器支援的大小。

▼ 監視資料庫快取

安裝上的預設快取層級僅適用於測試環境，不適用於生產環境。進行微調時，您可以監視伺服器的資料庫快取。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **監視資料庫快取。**

  ```
  $ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
   -b "cn=monitor,cn=ldbm database,cn=plugins,cn=config" "(objectclass=*)"
  ```

  若資料庫快取大小足夠，並且已完成填充，則符合率 (`dbcachehitratio`) 應該會很高。此外，已讀入的頁面數 (`dbcachepagein`) 與已寫出的乾淨頁面數 (`dbcacheroevict`) 應該都很低。此處的「高」與「低」是相對於部署限制的表示。

▼ 監視項目快取

進行微調時，您可以檢查一或多個尾碼的項目快取。使用下列程序即可檢視項目快取層級。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **監視項目快取。**

  ```
  $ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
   -b "cn=monitor,cn=db-name,cn=ldbm database,cn=plugins,cn=config" "(objectclass=*)"
  ```

  若尾碼的項目快取大小足以保存尾碼中大部分的項目，且快取已完成填充，則符合率 (`entrycachehitratio`) 應該會很高。

若您已填充快取，則您將會在先前空白的項目快取填入時，看見項目快取大小 (currententrycachesize) 接近項目快取大小上限 (maxentrycachesize)。在理想狀況下，項目中的大小 (currententrycachecount) 應等於或非常接近尾碼中的總項目數 (ldapentrycachecount)。

▼ 修改項目快取

注意 – 修改快取可能會嚴重影響伺服器效能。修改快取時請多加留意。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **取得目前的項目快取層級。**

   ```
   $ dsconf get-suffix-prop -h host -p port suffix-DN entry-cache-count entry-cache-size
   ```

2. **變更項目快取數。**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN entry-cache-count:integer
   ```

   其中 *integer* 是要儲存在快取中的項目數。

3. **變更項目快取大小。**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN entry-cache-size:size
   ```

   其中 *size* 是以 GB (G)、MB (M)、KB (k) 或位元組 (b) 表示的快取大小。所指定的大小必須是受機器支援的大小。

▼ 配置堆疊記憶體臨界值

若要限制 nsslapd 程序所使用的堆疊記憶體數，您可以設定動態記憶體佔用空間的臨界值。當目錄伺服器執行於共用或稀疏資源的機器上時，您即可設定此臨界值。

備註 – 此臨界值只能設定於 Solaris 與 Linux 平台上。

如需有關記憶體大小的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中的「Directory Server and Memory」。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

備註 – heap-high-threshold-size 與 heap-low-threshold-size 特性的預設值爲 undefined。

1 **設定堆疊記憶體高臨界值上限。**

```
$ dsconf set-server-prop -h host -p port heap-high-threshold-size:value
```

其中 *value* 是 undefined，或是以 GB (G)、MB (M)、KB (k) 或位元組 (b) 表示的記憶體大小。所指定的大小必須是受機器支援的大小。

如需有關 heap-high-threshold-size 適用值的建議，請參閱 server(5dsconf) 線上手冊。

2 **您可以選擇是否設定堆疊記憶體低臨界值上限。**

```
$ dsconf set-server-prop -h host -p port heap-low-threshold-size:value
```

其中 *value* 是 undefined，或是以 GB (G)、MB (M)、KB (k) 或位元組 (b) 表示的記憶體大小。所指定的大小必須是受機器支援的大小。

如需有關 heap-low-threshold-size 適用值的建議，請參閱 server(5dsconf) 線上手冊。

設定每個用戶端帳號的資源限制

您可以控制每個用戶端帳戶在伺服器上的搜尋作業資源限制。您可以在帳號的作業屬性中設定此類限制，而讓目錄伺服器根據用戶端用以連結至目錄的帳號加以強制執行。

您可以設定下列限制：

- 查詢限制可指定一個搜尋作業所檢查的項目數上限。
- 大小限制可指定在回應搜尋作業時所傳回的項目數上限。
- 時間限制可指定用以處理搜尋作業的時間上限。
- 閒置逾時可指定用戶端連線遭捨棄之前處於閒置狀態的時間上限。

備註 – 依預設，「目錄管理員」所能使用的資源數不受限制。

您對特定使用者帳號所設定的資源限制，優先於伺服器整體配置中所設定的資源限制。本節將就每個帳號的資源限制設定提供相關資訊。

本節中的範例會直接在項目的屬性中設定資源限制。您也可以使用「服務類別 (CoS)」機制，設定帳號的資源限制。爲用戶端應用程式擷取項目時，CoS 機制即會產生運算屬性。如需有關定義 CoS 的更多資訊，請參閱第 207 頁的 「服務類別」。

▼ 配置堆疊記憶體臨界值

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **使用** `dsconf get-server-prop` **指令可讀取資源限制伺服器特性。**

```
$ dsconf get-server-prop -h host -p port look-through-limit search-size-limit \
 search-time-limit idle-timeout
look-through-limit  :  5000
search-size-limit   :  2000
search-time-limit   :  3600
idle-timeout        :  none
```

輸出中顯示，搜尋作業在處理搜尋時，最多查詢 5000 個項目、最多傳回 2000 個項目，以及最多使用伺服器時間一小時 (3600 秒)。

2 **變更查詢限制。**

```
$ dsconf set-server-prop -h host -p port look-through-limit:integer
```

其中 *integer* 是搜尋作業檢查的最大項目數。

3 **變更搜尋大小限制。**

```
$ dsconf set-server-prop -h host -p port search-size-limit:integer
```

其中 *integer* 是搜尋作業傳回的最大項目數。

4 **變更搜尋時間限制。**

```
$ dsconf set-server-prop -h host -p port serach-time-limit:integer
```

其中 *integer* 是處理搜尋作業所耗費的最長時間。

5 **變更閒置逾時。**

```
$ dsconf set-server-prop -h host -p port idle-timeout:integer
```

其中 *integer* 是用戶端連線中斷之前可維持閒置的最長時間。

◆ ◆ ◆ 第 5 章

目錄伺服器項目

本章將討論如何管理您目錄中的資料項目。其中也將說明如何設定參照以及進行屬性值的加密。

規劃目錄部署時，您必須分門別類目錄所將包含的資料類型。建立項目及修改預設模式之前，請先閱讀「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中的相關章節。

您必須先定義適當的存取控制指令 (ACI)，才能修改您的目錄。如需詳細資訊，請參閱第 7 章。

本章包含下列主題：

- 第 85 頁的「管理項目」
- 第 95 頁的「設定參照」
- 第 98 頁的「檢查有效屬性語法」
- 第 98 頁的「追蹤目錄項目的修改」
- 第 99 頁的「爲屬性值加密」

管理項目

管理項目的最佳方式需視環境而定：

- 若您在管理方面多半使用 DSCC，而只想搜尋或修改少許項目，請使用 DSCC。如需有關 DSCC 的更多資訊，請參閱第 47 頁的「目錄服務控制中心介面」。
- 若不對目錄伺服器執行任何管理作業，而只想搜尋或修改少許項目，請使用目錄編輯器。如需有關目錄編輯器的資訊，請參閱「Sun Java System Directory Editor 1 2005Q1 Installation and Configuration Guide」。
- 若要搜尋或修改大量項目，請使用指令行公用程式 `ldapmodify` 與 `ldapdelete`。

使用 DSCC 管理項目

DSCC可讓您檢視項目所有可讀取的未加密屬性，以及編輯其可寫入的屬性。它也可讓您增加及移除屬性、設定多值屬性，以及管理項目的物件類別。如需有關如何使用 DSCC 管理項目的更多資訊，請參閱 DSCC 線上說明。如需更多有關 DSCC 的一般資訊，請參閱第 47 頁的 「目錄服務控制中心介面」。

使用目錄編輯器管理項目

目錄編輯器是一項易於使用的目錄編輯工具，可供管理員與一般使用者搜尋、建立與編輯資料。這項資料採用使用者、群組與容器的形式。

使用 ldapmodify 與 ldapdelete 管理項目

`ldapmodify` 與 `ldapdelete` 指令行公用程式提供完整的功能，可增加、編輯與刪除目錄內容。使用這些公用程式，可讓您管理伺服器的配置項目與使用者項目中的資料。這些公用程式亦可用以寫入程序檔，以執行一或多個目錄的大量管理。

本書中的多項程序皆會用到 `ldapmodify` 與 `ldapdelete` 指令。以下幾節將說明執行程序時所需進行的基本作業。如需有關 `ldapmodify` 與 `ldapdelete` 指令的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」。

指令行公用程式的輸入一律用於 LDIF 中，可直接從指令行提供，也可以透過輸入檔提供。下節將提供 LDIF 輸入的相關資訊，而後續幾節將說明每個修改類型的 LDIF 輸入。

如需有關正確格式化 LDIF 輸入的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Guidelines for Providing LDIF Input」。

以下幾節將說明這些基本作業：

- 第 86 頁的 「使用 ldapmodify 增加項目」
- 第 88 頁的 「使用 `ldapmodify` 修改項目」
- 第 92 頁的 「使用 `ldapdelete` 刪除項目」
- 第 92 頁的 「使用 `ldapmodify` 刪除項目」
- 第 92 頁的 「使用 `ldapsearch` 搜尋項目」

使用 ldapmodify 增加項目

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

您可以使用 `ldapmodify` 的 `-a` 選項，將一或多個項目增加至目錄。下列範例將建立包含使用者的結構項目，再建立使用者項目：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: ou=People,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
ou: People
description: Container for user entries

dn: uid=bjensen,ou=People,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgPerson
uid: bjensen
givenName: Barbara
sn: Jensen
cn: Babs Jensen
telephoneNumber: (408) 555-3922
facsimileTelephoneNumber: (408) 555-4000
mail: bjensen@example.com
userPassword: secret
```

`-D` 與 `-w` 選項分別可爲有權建立這些項目的使用者指定連結 DN 與密碼。`-a` 選項可指定將增加 LDIF 內的所有項目。接著，每個項目會依其 DN 與屬性值列出，每個項目皆會以空行隔開。每個項目在輸入後即由 `ldapmodify` 公用程式予以建立，且公用程式會報告任何發生的錯誤。

依慣例，項目的 LDIF 會列出下列屬性：

1. 項目的 DN。
2. 物件類別清單。
3. 一或多個命名屬性。此爲 DN 中所使用的屬性，並不一定是必要屬性之一。
4. 所有物件類別的必要屬性清單。
5. 您所要納入的任何允許的屬性。

鍵入 `userPassword` 屬性的值時，請提供純文字形式的密碼。伺服器將爲此值加密，並只會儲存加密值。請確實限制讀取權限，以保護 LDIF 檔案中所出現的純文字密碼。

您也可以使用不需在指令行上加入 `-a` 選項的替代形式 LDIF。此形式的好處是，您可以合併項目增加陳述式與項目修改陳述式，如下範例所示。

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: ou=People,dc=example,dc=com
changetype: add
objectclass: top
objectclass: organizationalUnit
ou: People
```

```
description: Container for user entries

dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgPerson
uid: bjensen
givenName: Barbara
sn: Jensen
cn: Barbara Jensen
telephoneNumber: (408) 555-3922
facsimileTelephoneNumber: (408) 555-4000
mail: bjensen@example.com
userPassword: secret
```

`changetype: add` 關鍵字表示，具有指定 DN 的項目應以所有後續的屬性建立。其他所有的選項與 LDIF 慣例皆與本節稍早的說明相同。

在這兩個範例中，您都可以使用 `-f` *filename* 選項從檔案讀取 LDIF，而不需從終端機輸入讀取。LDIF 檔案所含的格式必須與用於終端機輸入的格式相同，視 `-a` 選項的使用情形而定。

使用 `ldapmodify` 修改項目

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

使用 `changetype: modify` 關鍵字增加、取代或移除屬性及其位於現有項目中的值。當您指定 `changetype: modify` 時，必須同時提供一或多個變更作業，以指定項目的修改方式。可用的三個 LDIF 變更作業如下範例所示：

```
dn: entryDN
changetype: modify
add: attribute
attribute: value...
-
replace: attribute
attribute: newValue...
-
delete: attribute
[attribute: value]
...
```

請使用連字符 (`-`) 隔開同一行中相同項目的作業，並使用空行隔開不同項目的作業群組。您也可以爲每個作業指定數個 *attribute*: *value* 對。

增加屬性值

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

下列範例將說明如何使用相同的 add LDIF 語法在現有的多值屬性中增加值，以及在尚不存在的屬性中增加值：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: cn
cn: Babs Jensen
-
add: mobile
mobile: (408) 555-7844
```

若有下列任一情況，此作業即可能失敗，而伺服器將傳回錯誤：

- 屬性中已有指定的值存在。
- 值未遵循針對屬性所定義的語法。
- 項目的物件類別非必要或不允許該屬性類型。
- 該屬性類型不是多值屬性，同時屬性中已有值存在。

使用二進位屬性子類型

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

attribute;binary 子類型表示不論屬性值的實際語法爲何，這些屬性值皆必須透過 LDAP 以二進位資料傳輸。此子類型適用於不具 LDAP 字串表示法的複雜語法，如 userCertificate。除此用途之外，均不應使用二進位子類型。

與 ldapmodify 指令搭配使用時，可在任何 LDIF 陳述式的屬性名稱中增加適當的子類型。

輸入二進位值時，可以直接以 LDIF 文字鍵入，或從其他檔案中加以讀取。下列範例說明從檔案中加以讀取的 LDIF 語法：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
version: 1
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: userCertificate;binary
userCertificate;binary:< file:///local/cert-file
```

若要以 :< 語法指定檔案名稱，必須以行 version:1 做爲 LDIF 陳述式的開頭。ldapmodify 在處理此陳述式時，會將屬性設爲從指定檔案的所有內容讀取出來的值。

以語言子類型增加屬性

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

屬性的語言與發音子類型會指定本土化的值。當您指定屬性的語言子類型時，該子類型會以下列形式增加到屬性名稱中：

attribute;lang-*CC*

其中，*attribute* 是現有的屬性類型，而 *cc* 是用以指定語言的雙字母國碼 (地區碼)。您可以選擇性地增加發音子類型到語言子類型中，以指定音譯的本土化值。此案例中的屬性名稱如下：

attribute;lang-*CC*;phonetic

若要對具有子類型的屬性執行作業，必須明確符合其子類型。例如，若要修改具有 lang-fr 語言子類型的屬性值，必須在修改作業中納入 lang-fr，如下所示：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: homePostalAddress;lang-fr
homePostalAddress;lang-fr: 34, rue de la Paix
```

備註 – 若屬性值含有非 ASCII 字元，則必須以 UTF-8 編碼這些字元。

修改屬性值

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

下列範例說明如何使用 LDIF 中的 replace 語法變更屬性的值：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
replace: sn
sn: Morris
-
replace: cn
cn: Barbara Morris
cn: Babs Morris
```

指定屬性目前所有的值均會遭移除，而加入所有指定的值。

變更屬性值後，可以使用 ldapsearch 指令驗證變更。

屬性值的空格結尾

修改屬性值時，請勿不慎在值的結尾留下空格。空格結尾可能會致使值以 base-64 編碼 (如 34xy57eg) 顯示。

屬性值的結尾若爲空格，則會將此空格編碼爲屬性值的一部分。當您使用 DSCC 或 ldapsearch 指令驗證變更時，所看見的值可能會是純文字，但也可能以 base-64 編碼文字呈現。這取決於您所使用的是哪個目錄伺服器用戶端。

刪除屬性值

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

下列範例將說明如何徹底刪除屬性，以及如何僅刪除多值屬性的某個值：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
delete: facsimileTelephoneNumber
-
delete: cn
cn: Babs Morris
```

僅使用 delete 語法而未指定 *attribute: value* 對時，屬性中所有的值皆會遭移除。若您指定了 *attribute: value* 對，則只會移除該值。

修改多值屬性的某個值

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

若要以 ldapmodify 指令修改多值屬性的某個值，必須執行兩項作業，如下範例所示：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
delete: mobile
mobile: (408) 555-7845
-
add: mobile
mobile: (408) 555-5487
```

使用 ldapdelete 刪除項目

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

請使用 ldapdelete 指令行公用程式從目錄中刪除項目。此公用程式可連結至目錄伺服器，並根據項目的 DN 刪除一或多個項目。您必須提供有權刪除指定項目的連結 DN。

您無法刪除具有子項的項目。LDAP 協定不允許子項目無父系的情形。例如，在您刪除所有屬於組織單位的項目之前，都無法刪除組織單位項目。

下列範例將說明組織單位中的一個項目。此項目及其父系項目可依序刪除。

```
$ ldapdelete -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
uid=bjensen,ou=People,dc=example,dc=com
ou=People,dc=example,dc=com
```

使用 ldapmodify 刪除項目

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

使用 ldapmodify 公用程式時，也可以使用 changetype: delete 關鍵字刪除項目。使用 ldapdelete 時的所有限制於此時同樣適用，如上一節所述。使用 LDIF 語法刪除項目的好處在於，您可以同一個 LDIF 檔案中執行多種不同的作業。

下列範例將執行與前一範例相同的刪除作業：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: delete

dn: ou=People,dc=example,dc=com
changetype: delete
```

使用 ldapsearch 搜尋項目

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

您可以使用 ldapsearch 指令行公用程式尋找並擷取目錄項目。請注意，ldapsearch 公用程式不是 Solaris 平台提供的公用程式，而是 Directory Server Resource Kit 的一部份。

如需有關使用 ldapsearch、常用 ldapsearch 選項、已接受的格式及範例之更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」。

▼ 使用 ldapmodify 移動項目或為其重新命名

此程序會使用修改 DN 作業。執行此作業前，請確定您已熟悉第 94 頁的 「使用修改 DN 作業的指示與限制」一節。

針對此程序的某些部分，您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。此程序的其他部分只能使用指令行完成。

備註 – 修改屬於群組 `uniquemember` 的項目 DN 時，必須啓用參照完整性外掛程式。參照完整性可確保群組成員會隨著項目移動而調整。如需有關如何啓用及配置參照完整性外掛程式的資訊，請參閱第 218 頁的 「配置參照完整性外掛程式」。

---

1 若要將項目從某父系移至另一個父系，請先擴充父系項目的 ACI 權限。

- 請在要移動之項目的目前父系項目上使用 `allow (export ...)` 語法，以確定 ACI 允許 `export` 作業
- 請在要移動之項目的未來父系項目上使用 `allow (import ...)` 語法，以確定 ACI 允許 `import` 作業

如需有關使用 ACI 的資訊，請參閱第 7 章。

2 請確定修改 DN 作業已全域啓用，或至少已針對將受移動作業影響的一或多個尾碼啓用。

爲確保與舊目錄伺服器發行版本的相容性，修改 DN 作業預設爲不啓用。

若您先前已啓用修改 DN 作業，請跳至下個步驟。

若要全域啓用伺服器的修改 DN 作業，請使用此指令：

```
$ dsconf set-server-prop -h host -p port moddn-enabled:on
```

3 執行 `ldapmodify` 指令。

此步驟會使用修改 DN 作業。請執行下列其中一項動作：

- 移動項目。

  例如，下列指令會將項目 `uid=bjensen` 從包商的子樹狀結構 `ou=Contractors,dc=example,dc=com`，移至員工的子樹狀結構 `ou=People,dc=example,dc=com`：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=Contractors,dc=example,dc=com
changetype: modrdn
newrdn: uid=bjensen
deleteoldrdn: 0
newsuperior: ou=People,dc=example,dc=com
```

- 爲項目重新命名。

  例如，下列指令會將項目 uid=bbjensen 重新命名爲 uid=bjensen：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bbjensen,ou=People,dc=example,dc=com
changetype: modrdn
newrdn: uid=bjensen
deleteoldrdn: 1
```

撰寫 LDIF 陳述式時，請留意下列屬性：

- dn - 指定要重新命名或移動的項目。
- changetype: modrdn - 指定要使用修改 DN 作業。
- newrdn - 指定新命名的屬性。
- deleteoldrdn - 指定先前的命名屬性是否應從項目中移除 (1 表示是，0 表示否)。

  請注意，命名屬性若爲項目定義中的必要屬性，即無法從項目中移除。
- newsuperior - 指定項目的新上層屬性。

如需有關 ldapmodify 指令及其選項的資訊，請參閱 ldapmodify(1) 線上手冊。

4 **若在移動或重新命名含有大量項目的子樹狀結構時發生資源限制錯誤，請增加資料庫所能使用的鎖定數。**

```
$ dsconf set-server-prop -h host -p port db-lock-count:value
```

修改此特性後必須重新啓動伺服器，變更方能生效。

使用修改 DN 作業的指示與限制

如上一節所述，在使用修改 DN 作業時，請遵循以下幾節所說明的指示。

使用修改 DN 作業的一般指示

- 請勿使用修改 DN 作業將項目從某個尾碼移至其他尾碼，或使用此作業移動根尾碼或將其重新命名。
- 請確定您所執行的是 Directory Server 5.2 2005Q1 或更高版本。修改 DN 作業無法使用於 Directory Server 5.2 2005Q1 之前的目錄伺服器版本。
- 請勿於應用程式中使用 entryid 作業屬性，因爲此屬性保留僅限內部使用。entryid 屬性在項目移動後可能會變更。
- 爲伺服器上的所有尾碼全域性地啓用修改 DN 作業，或針對要執行此作業的每個尾碼個別啓用。修改 DN 作業預設爲停用。
- 針對要執行修改 DN 作業的每個尾碼，擴充其 ACI 權限。Import 存取權限可讓項目匯入指定的 DN。Export 存取權限可讓項目從指定的 DN 匯出。

- 執行修改 DN 作業前，請確定此作業不會中斷用戶端認證。若您移動了參照用戶端憑證的項目，用戶端認證即會中斷。移動項目後，請驗證您的憑證。
- 執行修改 DN 作業前，請確定此作業不會中斷您的應用程式。項目經過重新命名或移動後，可能會影響到數個尾碼，或變更項目的下列特性：
  - 項目的已篩選角色範圍。
  - 項目的巢式角色，此處的巢式角色含有篩選的角色。
  - 項目的動態群組成員身份。

搭配使用修改 DN 作業與複寫的指示

注意 – 使用修改 DN 作業時若未符合下列需求，將可能導致複寫中斷，而使目錄服務失效。

- 請確定您的複寫拓樸中所執行的所有伺服器皆爲 Directory Server 5.2 或更高版本。您無法在 Directory Server 5.2 之前的目錄伺服器版本上使用修改 DN 作業。
- 啓用複寫拓樸中所有伺服器的修改 DN 作業。若主伺服器支援修改 DN 作業，但用戶伺服器不支援，複寫即會失敗。供應者伺服器的錯誤記錄中將寫入如下訊息：

  ```
  Unable to start a replication session with MODDN enabled
  ```

  若要重新啓動複寫，請重新配置複寫拓樸，以啓用所有伺服器的修改 DN 作業。接著須以下列方式之一啓動複寫階段作業：
  - 依照第 250 頁的 「強制執行複寫更新」中的指示進行作業。
  - 藉由變更供應者伺服器的項目。變更即會複寫到用戶伺服器上。
- 啓用並配置拓樸中所有主伺服器複本的參照完整性外掛程式。此動作可確保伺服器能夠保持群組與角色的參照完整性。如需有關如何啓用及配置參照完整性外掛程式的資訊，請參閱第 218 頁的 「配置參照完整性外掛程式」。

  執行修改 DN 作業後，請保留時間讓參照完整性外掛程式複寫其變更。

設定參照

若用戶端應用程式在本機無法得知所應連絡的伺服器，您可以使用參照予以告知。參照是目錄伺服器傳回用戶端以替代結果之遠端尾碼或項目的指標。用戶端必須接著在參照中所命名的遠端伺服器上再次執行此作業。

有三種情況下會執行重新導向：

- 用戶端應用程式所請求的項目不存在於本機伺服器上，同時該伺服器依配置會傳回預設參照。
- 整個尾碼已因維護或安全性考量而停用時。

  伺服器將會傳回該尾碼所定義的參照。尾碼層級參照的說明請見第 64 頁的 「設定參照並使尾碼成爲唯讀模式」。當用戶端請求寫入作業時，尾碼的唯讀複本也會傳回參照至主伺服器。
- 當用戶端特別存取智慧型參照時。

  智慧型參照是您所建立的項目。伺服器會傳回智慧型參照所定義的參照。

在任何情況下，參照皆是含有主機名稱、連接埠號碼與其他伺服器之 DN (選用) 的 LDAP URL。例如，`ldap://east.example.com:389`。

如需有關如何在目錄部署中使用參照的概念性資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」。

以下幾節將說明設定目錄預設參照以及建立與定義智慧型參照的程序。

設定預設參照

若是用戶端應用程式提交作業所在的 DN 不在目錄伺服器所維護的尾碼上，則會傳回預設參照給用戶端應用程式。伺服器會傳回所有定義的參照，但不會定義其傳回順序。

▼ 設定預設參照

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **使用 `dsconf` 指令行公用程式可設定一或多個預設參照。**

  ```
  $ dsconf set-server-prop -h host -p port suffix-DN referral-url:referral-URL
  ```

  例如：

  ```
  $ dsconf set-server-prop -h host1 -p 1389 dc=example,dc=com \
   referral-url:ldap://east.example.com:1389
  ```

設定智慧型參照

智慧型參照可讓您將目錄項目或目錄樹狀結構對映至特定的 LDAP URL。使用智慧型參照，可讓您將用戶端應用程式納入特定的伺服器或特定伺服器上特定項目的參照。

智慧型參照通常會指向另一部伺服器上具有相同 DN 的實際項目。但您仍可定義相同或不同伺服器上任意項目的智慧型參照。例如，您可以將具有下列 DN 的項目定義爲智慧型參照：

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

智慧型參照會指向伺服器 `east.example.com` 上的另一個項目：

```
cn=Babs Jensen,ou=Sales,o=east,dc=example,dc=com
```

目錄使用智慧型參照的方式會遵循 RFC 4511 之 4.1.10 小節中所指定的標準 (`http://www.ietf.org/rfc/rfc4511.txt`)。

▼ 建立及修改智慧型參照

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **若要建立智慧型參照，請使用 `referral` 與 `extensibleObject` 物件類別建立項目。**

`referral` 物件類別可讓預期的 `ref` 屬性包含 LDAP URL。`extensibleObject` 物件類別可讓您將任何模式屬性用爲命名屬性，以符合目標項目。

例如，若要定義下列可傳回智慧型參照的項目，而非項目 `uid=bjensen`，請使用此指令：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: referral
uid: bjensen
ref: ldap://east.example.com/cn=Babs%20Jensen,ou=Sales,o=east,dc=example,dc=com
```

備註 – LDAP URL 中任何位於空格之後的資訊均會遭伺服器忽略。因此，在您打算做爲參照的任何 LDAP URL 中，所有空格都必須以 `%20` 取代。其他特殊字元則必須加上引號。

在您定義智慧型參照後，`uid=bjensen` 項目的修改實際上將對其他伺服器的 `cn=Babs Jensen` 項目執行。`ldapmodify` 指令會自動追蹤參照，例如：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: replace
replace: telephoneNumber
telephoneNumber: (408) 555-1234
```

2 **(可選擇) 若要修改智慧型參照項目，請使用 `ldapmodify` 的 `-M` 選項：**

```
$ ldapmodify -M -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: replace
replace: ref
ref: ldap://east.example.com/cn=Babs%20Jensen,ou=Marketing,o=east,dc=example,dc=com
```

檢查有效屬性語法

目錄伺服器可讓您在執行下列作業時檢查屬性的完整性：

- 使用 `dsadm import` 或 `dsconf import` 匯入資料。
- 使用 LDAP 或 DSML 增加項目、修改項目或修改項目的 DN。

檢查可確保屬性值符合 IETF 的建議。所有不符的屬性皆會遭到拒絕，並記錄於錯誤記錄中。記錄訊息包含連線與作業 ID (若適用)。

伺服器預設不會自動檢查前述作業的語法。若要開啓語法檢查，請使用下列程序。

備註 – 語法檢查與模式檢查不同。如需有關模式檢查的資訊，請參閱第 261 頁的 「管理模式檢查」。

▼ 開啓自動語法檢查

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **若要開啓自動語法檢查，請使用此指令：**

```
$ dsconf set-server-prop -h host -p port check-syntax-enabled:on
```

追蹤目錄項目的修改

伺服器預設會保存新建或已修改之項目的特殊屬性，如 LDAP v3 規格中所指定。這些特殊屬性會儲存在尾碼的項目上，並且包含：

- `creatorsName` — 最初建立項目之使用者的 DN。
- `createTimestamp` — 項目建立時的時間戳記，使用 GMT 格式。
- `modifiersName` — 上次修改項目的使用者 DN。
- `modifyTimestamp` — 項目修改時的時間戳記，使用 GMT 格式。

▼ 關閉項目修改追蹤

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

注意 – 關閉項目修改追蹤會導致不相符的資料。由於有多項應用程式皆依賴這些屬性，且停用此功能後效能只會略爲提昇，因此建議您不要關閉項目修改追蹤。

- **關閉伺服器的項目修改追蹤。**

```
$ dsconf set-server-prop -h host -p port suffix-DN mod-tracking-enabled:off
```

為屬性值加密

屬性加密可保護目錄中所儲存之機密資料的安全性。屬性加密可讓您指定項目的某些屬性必須以加密格式儲存。如此可防止儲存在資料庫檔案、備份檔案與匯出的 LDIF 檔案中的資料遭讀取。

使用此功能時，屬性值在儲存至目錄伺服器資料庫之前會先進行加密，並在傳回至用戶端之前解密回原始值。您必須使用存取控制以防止用戶端在無權限的情況下存取此類屬性，並在屬性值傳輸於用戶端與目錄伺服器之間時，使用 SSL 爲屬性值加密。如需資料安全性的一般架構性簡介與屬性加密的特定架構性簡介，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」。

必須在伺服器上配置 SSL 並加以啓用，屬性加密方可運作。但依預設所有屬性皆不會加密。屬性加密配置於尾碼層級，這表示屬性會在尾碼中含有該屬性的每個項目中加密。若要爲整個目錄中的某一屬性加密，您必須在每個尾碼中啓用該屬性的加密。

注意 – 屬性加密會影響所有與尾碼相關的資料及索引檔案。若要修改現有尾碼的加密配置，您**必須**先匯出其內容，並在變更配置後重新匯入內容。DSCC 可協助您執行這些步驟。如需有關使用 DSCC 的更多資訊，請參閱第 47 頁的 「目錄服務控制中心介面」。

爲進一步確保安全性，只要您開啓屬性加密，即應手動刪除仍可能含有未加密值的資料庫快取檔案與資料庫記錄檔。如需刪除這些檔案的程序，請參閱第 102 頁的 「配置屬性加密」。

您應在載入或建立新尾碼中的資料之前啓用加密屬性。

若您選擇加密被某些項目做爲命名屬性的屬性，DN 中所出現的值將不會加密。儲存於項目中的值將會加密。

雖然您在配置加密時可以選取 userPassword 屬性，但除非遇到密碼必須以純文字格式儲存的情況，否則安全性並不會因此有任何提昇。DIGEST-MD5 SASL 認證也是如此。若密碼已在其密碼策略中定義了加密機制，則進一步的加密所增添的安全性將極其有限，且反而會影響每個連結作業的效能。

在儲存時，加密屬性的開頭會加上密碼標記，表示其使用了加密演算法。使用 DES 加密演算法的加密屬性將如下所示：

```
{CKM_DES_CBC}3hakc&jla+=snda%
```

當您從線上匯入資料而在檢視時爲資料加密後，您即已提供對伺服器進行認證的金鑰資料庫密碼，往後將不會再出現此提示。若您進行離線資料匯入，目錄伺服器則會先提示您提供密碼，始允許您爲匯入的資料加密。爲資料解密時 (此作業需要更高的安全性)，無論此匯出作業是線上或是離線，目錄伺服器都會自動提示您提供金鑰資料庫密碼。如此將可進一步確保安全性。

備註 – 只要憑證或私密金鑰未變更，伺服器就會繼續產生相同的金鑰。因此，只要兩個伺服器實例使用相同的憑證，資料即可從一個伺服器實例傳輸 (匯出並匯入) 至另一個實例。

屬性加密與效能

屬性加密雖然可提昇資料安全性，但也會影響系統效能。請仔細評估哪些屬性需要加密，而僅就您認爲特別機密的屬性進行加密。

由於機密資料可透過索引檔案直接存取，因此對應於加密屬性的索引鍵必須進行加密，使屬性受到完整的保護。在索引已對目錄伺服器效能造成影響 (尚未納入索引鍵加密所造成的影響) 的情況下，請在首次將資料匯入或增加到資料庫**之前**，先配置屬性加密。此程序可確保加密屬性的索引具有先佔性。

屬性加密用法注意事項

實作屬性加密功能時，請考量下列事項：

- 一般而言，在修改屬性加密配置時，最理想的做法是先匯出資料，再進行配置變更，然後匯入新配置的資料。

  如此可確保所有配置變更均能保有其完整性，而不會有功能上的缺失。否則，某些功能可能會有所缺失，而破壞資料的安全性。

- 在現有的資料庫上修改屬性加密配置，可能會對系統效能造成嚴重的影響。

  例如，假設您有一個含有現行資料的資料庫實例。此資料庫含有先前所儲存之具有 `mySensitiveAttribute` 屬性的項目。此屬性的值以純文字格式儲存於資料庫與索引檔案中。若您日後決定爲 `mySensitiveAttribute` 屬性加密，則資料庫實例中所有的資料皆必須先匯出再重新匯入資料庫，以確保伺服器能夠使用屬性加密配置，更新資料庫與索引檔案。若能一開始就爲屬性加密，即可避免此時所造成的效能影響。

- 以解密格式匯出資料時若使用了錯誤的密碼，匯出即會遭到拒絕。

  若使用者要以解密格式匯出資料，伺服器將依其安全機制提示使用者提供密碼。若使用者所提供的密碼不正確，伺服器即會拒絕解密匯出作業。您可以直接輸入密碼，或提供密碼所在檔案的路徑。請注意，此檔案具有與 SSL 密碼檔案相同的語法。請參閱第 113 頁的 「配置憑證資料庫密碼」。

  使用 `dsconf` 指令時若要搭配 `--decrypt-attr` 選項，`set password prompt` 必須設定爲 `on`，且您必須如第 113 頁的 「配置憑證資料庫密碼」中所述，選擇好了憑證資料庫密碼。

- 演算法可進行變更，但若變更過程有誤，結果中的索引功能即可能有所缺失。

  若要變更用以加密資料的演算法，請匯出資料、修改屬性加密配置，再匯入資料。若未遵循此程序，依據初始加密演算法所建立的索引將會無法運作。

  因爲加密屬性的開頭會加上密碼標記以表示其已使用加密演算法，所以匯入資料的動作將由內部伺服器作業負責。因此，目錄伺服器可讓您在變更演算法之前以加密格式匯出資料。

- 變更伺服器的 SSL 憑證會使您無法將加密的資料解密。

  屬性加密功能會使用伺服器的 SSL 憑證產生本身的金鑰，用以執行後續的加密與解密作業。因此，對加密的資料進行解密時，SSL 憑證是不可或缺的項目。若未先將資料解密即變更憑證，即無法進行資料解密。爲避免陷入此窘境，請先以解密格式匯出資料，然後在變更憑證後重新匯入資料。

- 若要以加密格式傳輸資料，亦即在兩個伺服器實例間匯出及匯入資料，則兩個伺服器實例皆必須使用相同的憑證。

  如需相關資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 管理指南」中的「Encrypting Attribute Values」。

▼ 配置屬性加密

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **如果要配置屬性加密的尾碼中含有任何項目，即須先將該尾碼的內容匯出至 LDIF 檔案。**

若尾碼中含有加密屬性，而您想要使用匯出的 LDIF 檔案重新初始化尾碼，可以讓屬性以匯出的 LDIF 保持加密形式。

2 **若要啓用屬性加密，請使用此指令：**

```
$ dsconf create-encrypted-attr -h host -p port suffix-DN attr-name cipher-name
```

其中 *cipher-name* 爲下列其中一項：

- `des` - DES 區塊密碼
- `des3` - Triple-DES 區塊密碼
- `rc2` - RC2 區塊密碼
- `rc4` - RC4 串流密碼

例如：

```
$ dsconf create-encrypted-attr -h host1 -p 1389 dc=example,dc=com uid rc4
```

3 **若要將加密屬性回復為其原始狀態，請使用此指令：**

```
$ dsconf delete-encrypted-attr -h host -p port suffix-DN attr-name
```

4 **若您變更了配置而為一或多個屬性加密，且這些屬性在匯入作業之前即有值，請清除資料庫快取並移除記錄。**

在資料庫快取與資料庫記錄中，不會顯示任何未加密的值。

備註 – 若是刪除這些檔案，則會遺失某些追蹤資訊。此外，當您刪除這些檔案後，伺服器將處於回復模式，而需要很長的時間才能重新啓動。

若要清除資料庫快取及移除記錄，請執行以下作業：

a. **停止目錄伺服器，如第 61 頁的 「啓動、停止與重新啓動目錄伺服器實例」中所述。**

b. **以超級使用者或具有管理員權限的使用者身份，從檔案系統中刪除資料庫快取檔案。**

```
# rm instance-path/db/__db.*
```

c. 從檔案系統中刪除資料庫記錄檔。

```
# rm instance-path/db/log.0000000001
```

d. 重新啓動目錄伺服器。

伺服器會自動建立新的資料庫快取檔案。在重新填入快取前，此尾碼中的作業效能可能會略受影響。

5 以 LDIF 檔案初始化尾碼，如第 193 頁的 「初始化尾碼」中所述。

載入檔案且建立對應的索引時，指定屬性的所有值均將加密。

◆ ◆ ◆ 第 6 章

目錄伺服器安全性

目錄伺服器支援數項可透過網路提供安全且可信任之通訊的機制。LDAPS 是於安全通訊端層 (SSL) 之上執行的標準 LDAP 協定。LDAPS 會爲資料加密，並選擇性地使用憑證進行認證。本章所提及之 SSL 一詞，係指支援的協定 SSL2、SSL3 與 TLS 1.0。

目錄伺服器也支援可在原本未加密的 LDAP 連線上啓用 TLS 的 Start 傳輸層安全性 (Start TLS) 延伸作業。

此外，目錄伺服器也支援透過簡單驗證與安全層 (SASL) 所執行的通用安全服務 API (GSSAPI)。GSSAPI 可讓您在 Solaris 作業系統 (Solaris OS) 上使用 Kerberos 第 5 版安全性協定。身份識別對映機制會接著爲 Kerberos 主體與目錄中的身份識別建立關聯。

如需其他安全性資訊，請參閱 NSS 網站，網址是 `http://www.mozilla.org/projects/security/pki/nss/`。

本章提供透過 SSL 配置安全性的程序。如需有關 ACI 的資訊，請參閱第 7 章。如需有關使用者存取與密碼的資訊，請參閱第 8 章。

本章包含下列主題：

- 第 106 頁的「對目錄伺服器使用 SSL」
- 第 106 頁的「管理憑證」
- 第 114 頁的「配置 SSL 通訊」
- 第 116 頁的「配置憑證層級與認證方法」
- 第 123 頁的「配置 LDAP 用戶端以使用安全性」
- 第 136 頁的「傳遞式認證」

對目錄伺服器使用 SSL

安全通訊端層 (SSL) 可在目錄伺服器及其用戶端之間提供加密通訊與可選擇的認證功能。SSL 可透過 LDAP 使用，或與 DSML-over-HTTP 搭配使用。SSL 預設會透過 LDAP 啓用，但若您使用 DSML-over-HTTP，亦可輕鬆啓用 SSL。此外，複寫經配置後亦可使用 SSL，讓伺服器之間能進行安全通訊。

若在簡單認證 (連結 DN 與密碼) 中使用 SSL，即會對伺服器所收送的所有資料進行加密。加密可確保機密度與資料完整性。用戶端可選擇性地透過簡單驗證與安全層 (SASL) 使用憑證，對目錄伺服器或第三方安全性機制進行認證。憑證型認證可使用公開金鑰加密法，防止用戶端或伺服器進行僞造或模擬。

目錄伺服器能夠在個別的連接埠上同時進行 SSL 與非 SSL 通訊。基於安全性考量，您也可以限定所有通訊皆需使用 LDAP 安全連接埠。用戶端認證也是可配置的。您可以將用戶端認證設爲必要或允許項目。此設定將決定您所執行的安全性層級。

SSL 可支援 Start TLS 延伸作業，而爲一般 LDAP 連線提供安全性。用戶端可連結至標準 LDAP 連接埠而使用傳輸層安全性協定，以維護連線的安全性。Start TLS 作業可讓用戶端享有更大的彈性，並有利於簡化連接埠配置。

SSL 所提供的加密機制亦可用於屬性加密。啓用 SSL 可讓您對尾碼配置屬性加密，以維護資料儲存在目錄時的安全性。如需更多資訊，請參閱第 99 頁的 「爲屬性值加密」。

如需額外的安全性，您可以透過存取控制指令 (ACI)，設定對目錄內容的存取控制。ACI 須使用特定的認證方法，同時確保資料僅可透過安全通道進行傳輸。設定 ACI 可彌補使用 SSL 與憑證的不足之處。如需更多資訊，請參閱第 7 章。

SSL 預設會透過 LDAP 啓用，在使用 DSML-over-HTTP 時，亦可輕鬆啓用 SSL。此外，您可能會如以下幾節所述，想修改某些方面的 SSL 配置。

管理憑證

本節說明如何管理目錄伺服器中的 SSL 憑證。

若要在目錄伺服器上執行 SSL，您必須使用自行簽署的憑證或公開金鑰基礎架構 (PKI) 解決方案。

PKI 解決方案與外部憑證授權機構 (CA) 相關。使用 PKI 解決方案時，必須具備 CA 簽署的伺服器憑證，其中包含公開金鑰與私密金鑰。此憑證隨各目錄伺服器而不同。您也需具備可信任的 CA 憑證，其中包含公開金鑰。可信任的 CA 憑證可確保來自您的 CA 之所有伺服器憑證皆可信任。此憑證有時亦稱爲 CA 根金鑰或根憑證。

備註 – 若使用憑證進行測試，可能會使用自行簽署的憑證。但在生產環境中使用自行簽署的憑證，並不是安全的做法。在生產環境中，請使用可信任的憑證授權機構 (CA) 憑證。

本節中的程序使用 `dsadm` 與 `dsconf` 指令。如需有關這些指令的資訊，請參閱 `dsadm`(1M) 線上手冊與 `dsconf`(1M) 線上手冊。

本節提供下列有關在目錄伺服器上配置憑證的資訊：

- 第 107 頁的「檢視預設自行簽署的憑證」
- 第 107 頁的「管理自行簽署的憑證」
- 第 108 頁的「請求 CA 簽署的伺服器憑證」
- 第 109 頁的「增加 CA 簽署的伺服器憑證與可信任的 CA 憑證」
- 第 112 頁的「更新過期的 CA 簽署伺服器憑證」
- 第 112 頁的「匯出及匯入 CA 簽署伺服器憑證」
- 第 113 頁的「配置憑證資料庫密碼」
- 第 114 頁的「爲目錄伺服器備份及復原憑證資料庫」

▼ 檢視預設自行簽署的憑證

目錄伺服器實例於首次建立時，會包含預設自行簽署的憑證。**自行簽署的憑證**是成對的公開與私密金鑰組，其中的公開金鑰由私密金鑰所簽署。自行簽署的憑證有效期限爲三個月。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

- **若要檢視預設自行簽署的憑證，請使用此指令：**

```
$ dsadm show-cert instance-path defaultCert
```

▼ 管理自行簽署的憑證

當您建立目錄伺服器實例時，系統自動提供預設的自行簽署憑證。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1. **若要以非預設設定建立自行簽署的憑證，請使用此指令：**

```
$ dsadm add-selfsign-cert instance-path cert-alias
```

其中 *cert-alias* 是您提供用以識別憑證的名稱。

若要檢視此指令的所有選項，請參閱 dsadm(1M) 線上手冊或指令行說明：

```
$ dsadm add-selfsign-cert --help
```

2 **當自行簽署的憑證過期時，停止伺服器實例並更新憑證。**

```
$ dsadm stop instance-path
$ dsadm renew-selfsign-cert instance-path cert-alias
```

3 **重新啓動伺服器實例。**

```
$ dsadm start instance-path
```

▼ 請求 CA 簽署的伺服器憑證

此程序說明如何請求及安裝用於目錄伺服器的 CA 簽署伺服器憑證。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **產生 CA 簽署的伺服器憑證請求。**

```
$ dsadm request-cert [-W cert-pwd-file] {-S DN | --name name [--org org] \
  [--org-unit org-unit] [--city city] [--state state] [--country country]} \
  [-o output-file] [-F format] instance-path
```

例如，若要爲 Example 公司請求 CA 簽署的伺服器憑證，請使用此指令：

```
$ dsadm request-cert --name host1 --org Example --org-unit Marketing \
 -o my_cert_request_file /local/ds
```

爲完整識別伺服器，憑證授權機構可能會要求此範例中所顯示的所有屬性。如需每個屬性的說明，請參閱 dsadm(1M) 線上手冊。

當您使用 dsadm request-cert 請求憑證時，除非指定 ASCII 做爲輸出格式，否則所產生的憑證請求將是二進位憑證請求。若您指定 ASCII，所產生的憑證請求則會是 PEM 格式的 PKCS #10 憑證請求。PEM 是 RFC 1421 至 1424 (http://www.ietf.org/rfc/rfc1421.txt) 所指定的安全電子郵件 (Privacy Enhanced Mail) 格式，可用以透過 US-ASCII 字元呈現 base64 編碼的憑證請求。請求的內容如下列範例所示：

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBrjCCARcCAQAwbjELMAkGA1UBhMCVXMxEzARBgNVBAgTCkNBElGT1JOSUExLD
AqBgVBAoTI25ldHNjYXBlIGNvb11bmljYXRpb25zIGNvcnBvcmF0aWuMRwwGgYDV
QQDExNtZWxsb24umV0c2NhcGUuY29tMIGfMA0GCSqGSIb3DQEBAUAA4GNADCBiQK
BgCwAbskGh6SKYOgHy+UCSLnm3ok3X3u83Us7u0EfgSLR0f+K41eNqqWRftGR83e
mqPLDOf0ZLTLjVGJaHJn4l1gG+JDf/n/zMyahxtV7+T8GOFFigFfuxJaxMjr2j7I
vELlxQ4IfZgwqCm4qQecv3G+N9YdbjveMVXW0v4XwIDAQABAAwDQYJKoZIhvcNAQ
EEBQADgYEAZyZAm8UmP9PQYwNy4Pmypk79t2nvzKbwKVb97G+MT/gw1pLRsuBoKi
```

```
nMfLgKp1Q38K5Py2VGW1E47/rhm3yVQrIiwV+Z8Lcc=
-----END NEW CERTIFICATE REQUEST-----
```

2 **根據適當程序，將憑證請求傳輸到您的憑證授權單位。**

取得憑證授權單位憑證的程序會因使用的憑證授權單位而異。有些商業 CA 會提供網站供您自動下載憑證。其他 CA 則會在您請求時，以電子郵件傳送。

傳送請求之後，必須等候 CA 回應以提供憑證。請求的回應時間各異。例如，若您的 CA 屬公司內部，則可能只需一或兩天即可回應您的請求。若所選取的 CA 屬公司外部，則可能需要數週才可回應您的請求。

3 **請儲存您從憑證授權機構接收的憑證。**

將憑證備份在安全之處。萬一憑證遺失，可以使用備份檔案重新加以安裝。可以將其儲存於文字檔中。PEM 格式的 PKCS #11 憑證內容如下列範例所示：

```
-----BEGIN CERTIFICATE-----
MIICjCCAZugAwIBAgICCEEwDQYJKoZIhKqvcNAQFBQAwfDELMAkGA1UEBhMCVVMx
IzAhBgNVBAoGlBhbG9a2FWaWxsZGwSBXaWRnZXRzLCBJbmMuMR0wGwYDVQQLExRX
aWRnZXQgTW3FrZXJzICdSJyBVczEpMCcGAx1UEAxgVGVzdCBUXN0IFRlc3QgVGVz
dCBUZXN0IFlc3QgQ0EswHhcNOTgwMzEyMDIzMzUWhcNOTgwMzI2MDIzMpzU3WjBP
MQswCYDDVQQGEwJVUzEoMCYGA1UEChMfTmV0c2NhcGUgRGlyZN0b3J5VIFB1Ymxp
Y2F0aW9uczEWMB4QGA1UEAxMNZHVgh49dq2tLNvbjTBaMA0GCSqGSIb3DQEBAQUA
A0kAMEYkCQCksMR/aLGdfp4m0OiGgijG5KgOsyRNvwGYW7kfW+8mmijDtZaRjYNj
jcgpF3VnlbxbclX9LVjjNLC5737XZdAgEDozYwpNDARBglghkgBhvhCEAQEEBAMC
APAwHkwYDVR0jBBgwFAU67URjwCaGqZHUpSpdLxlzwJKiMwDQYJKoZIhQvcNAQEF
BQADgYEAJ+BfVem3vBOPBveNdLGfjlb9hucgmaMcQa9FA/db8qimKT/ue9UGOJqL
bwbMKBBopsDn56p2yV3PLIsBgrcuSoBCuFFnxBnqSiTS7YiYgCWqWaUA0ExJFmD6
6hBLseqkSWulk+hXHN7L/NrViO+7zNtKcaZLlFPf7d7j2MgX4Bo=
-----END CERTIFICATE-----
```

▼ 增加 CA 簽署的伺服器憑證與可信任的 CA 憑證

此程序說明如何安裝用於目錄伺服器之 CA 簽署的伺服器憑證與可信任的 CA 憑證。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **增加 CA 簽署的伺服器憑證。**

```
$ dsadm add-cert --ca instance-path cert-alias cert-file
```

其中 *cert-alias* 是提供用以識別憑證的名稱，而 *cert-file* 是文字檔，內含 PEM 格式的 PKCS #11 憑證。

例如，若要安裝 CA 簽署的伺服器憑證，可以使用如下的指令：

```
$ dsadm add-cert /local/ds server-cert /local/safeplace/serv-cert-file
```

憑證此時已完成安裝，但尚未受信任。若要信任 CA 簽署的伺服器憑證，必須安裝憑證授權機構的憑證。

2 增加可信任的憑證授權機構憑證。

```
$ dsadm add-cert --ca instance-path cert-alias cert-file
```

--ca 選項會指出此憑證爲信任的憑證授權機構憑證。

例如，若要安裝憑證授權機構所提供之可信任的憑證，必須使用此指令：

```
$ dsadm add-cert --ca /local/ds CA-cert /local/safeplace/ca-cert-file
```

3 (可選擇) 驗證所安裝的憑證。

- 若要列出所有伺服器憑證並顯示其有效日期與別名，請鍵入：

  ```
  $ dsadm list-certs instance-path
  ```

  例如：

  ```
  $ dsadm list-certs /local/ds1
  Enter the certificate database password:
  Alias       Valid from Expires on Self-   Issued by          Issued to
                                    signed?
  ----------- ---------- ---------- ------- -----------------  -----------------
  serverCert  2000/11/10 2011/02/10 n       CN=CA-Signed Cert, CN=Test Cert,
              18:13      18:13              OU=CA,O=com        dc=example,dc=com
  defaultCert 2006/05/18 2006/08/18 y       CN=host1,CN=DS,    Same as issuer
              16:28      16:28              dc=example,dc=com
  2 certificates found
  ```

  目錄代理伺服器的實例預設會包含名爲 defaultCert 的預設伺服器憑證。Same as issuer 表示預設憑證爲自行簽署的伺服器憑證。

- 若要列出可信任的 CA 憑證，請鍵入：

  ```
  $ dsadm list-certs -C instance-path
  ```

  例如：

  ```
  $ dsadm list-certs -C /local/ds1
  Enter the certificate database password:
  Alias   Valid from Expires on Self-   Issued by           Issued to
                                signed?
  ------- ---------- ---------- ------- -----------------   --------------
  CA-cert 2000/11/10 2011/02/10 y       CN=Trusted CA Cert, Same as issuer
          18:12      18:12              OU=CA,O=com
  1 certificate found
  ```

- 若要檢視憑證的詳細資訊 (包括憑證過期日)，請鍵入：

```
$ dsadm show-cert instance-path cert-alias
```

例如，若要檢視伺服器憑證，請鍵入：

```
$ dsadm show-cert /local/ds1 "Server-Cert"
Enter the certificate database password:
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
        Signature Algorithm: PKCS #1 MD5 With RSA Encryption
        Issuer:
            "CN=Server-Cert,O=Sun,C=US"
        Validity:
            Not Before: Fri Nov 10 18:12:20 2000
            Not After : Thu Feb 10 18:12:20 2011
        Subject:
            "CN=CA Server Cert,OU=ICNC,O=Sun,C=FR"
        Subject Public Key Info:
            Public Key Algorithm: PKCS #1 RSA Encryption
            RSA Public Key:
                Modulus:
                    bd:76:fc:29:ca:06:45:df:cd:1b:f1:ce:bb:cc:3a:f7:
                    77:63:5a:82:69:56:5f:3d:3a:1c:02:98:72:44:36:e4:
                    68:8c:22:2b:f0:a2:cb:15:7a:c4:c6:44:0d:97:2d:13:
                    b7:e3:bf:4e:be:b5:6a:df:ce:c4:c3:a4:8a:1d:fa:cf:
                    99:dc:4a:17:61:e0:37:2b:7f:90:cb:31:02:97:e4:30:
                    93:5d:91:f7:ef:b0:5a:c7:d4:de:d8:0e:b8:06:06:23:
                    ed:5f:33:f3:f8:7e:09:c5:de:a5:32:2a:1b:6a:75:c5:
                    0b:e3:a5:f2:7a:df:3e:3d:93:bf:ca:1f:d9:8d:24:ed
                Exponent: 65537 (0x10001)
    Signature Algorithm: PKCS #1 MD5 With RSA Encryption
    Signature:
        85:92:42:1e:e3:04:4d:e5:a8:79:12:7d:72:c0:bf:45:
        ea:c8:f8:af:f5:95:f0:f5:83:23:15:0b:02:73:82:24:
        3d:de:1e:95:04:fb:b5:08:17:04:1c:9d:9c:9b:bd:c7:
        e6:57:6c:64:38:8b:df:a2:67:f0:39:f9:70:e9:07:1f:
        33:48:ea:2c:18:1d:f0:30:d8:ca:e1:29:ec:be:a3:43:
        6f:df:03:d5:43:94:8f:ec:ea:9a:02:82:99:5a:54:c9:
        e4:1f:8c:ae:e2:e8:3d:50:20:46:e2:c8:44:a6:32:4e:
        51:48:15:d6:44:8c:e6:d2:0d:5f:77:9b:62:80:1e:30
    Fingerprint (MD5):
        D9:FB:74:9F:C3:EC:5A:89:8F:2C:37:47:2F:1B:D8:8F
    Fingerprint (SHA1):
        2E:CA:B8:BE:B6:A0:8C:84:0D:62:57:85:C6:73:14:DE:67:4E:09:56

    Certificate Trust Flags:
        SSL Flags:
```

```
        Valid CA
        Trusted CA
        User
        Trusted Client CA
    Email Flags:
        User
    Object Signing Flags:
        User
```

▼ 更新過期的 CA 簽署伺服器憑證

當您的 CA 簽署伺服器憑證 (公開金鑰與私密金鑰) 過期時，請使用此程序加以更新。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **從憑證授權機構取得更新的 CA 簽署伺服器憑證。**
2. **當您收到更新的憑證時，請停止伺服器實例，再安裝該憑證。**

   ```
   $ dsadm stop instance-path
   $ dsadm renew-cert instance-path cert-alias cert-file
   ```

3. **重新啓動伺服器實例。**

   ```
   $ dsadm start instance-path
   ```

▼ 匯出及匯入 CA 簽署伺服器憑證

有時您可能會想匯出憑證的公開與私密金鑰，以便後續可匯入憑證。例如，可能會想讓其他伺服器使用您的憑證。

此程序中的指令可用於含有萬用字元的憑證，例如 "cn=*,o=example"。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **匯出憑證。**

   ```
   $ dsadm export-cert [-o output-file] instance-path cert-alias
   ```

   例如：

   ```
   $ dsadm export-cert -o /tmp/first-certificate /local/ds1 "First Certificate"
   $ dsadm export-cert -o /tmp/first-ca-server-certificate /local/ds1/ defaultCert
   Choose the PKCS#12 file password:
   Confirm the PKCS#12 file password:
   ```

```
$ ls /tmp
first-ca-server-certificate
```

2 **匯入憑證。**

```
$ dsadm import-cert instance-path cert-file
```

例如，若要將憑證匯入伺服器實例：

```
$ dsadm import-cert /local/ds2 /tmp/first-ca-server-certificate
Enter the PKCS#12 file password:
```

3 **(可選擇) 若已將憑證匯入伺服器，請配置伺服器以使用匯入的憑證。**

```
$ dsconf set-server-prop -e -h host -p port ssl-rsa-cert-name:server-cert
```

配置憑證資料庫密碼

目錄伺服器預設會透過儲存的密碼，在內部管理 SSL 憑證資料庫密碼。管理憑證時，使用者並不需鍵入憑證密碼或指定密碼檔案。此選項僅會隱藏密碼，而不會對密碼進行加密，因此並不安全。

但若您在使用憑證時想要有更多的控制，則可以配置伺服器，讓使用者在使用指令行時必須提供密碼。在此情況下，對於 `autostart`、`backup`、`disable-service`、`enable-service`、`info`、`reindex`、`restore` 與 `stop` 以外的所有 `dsadm` 子指令，使用者都必須鍵入憑證資料庫密碼。憑證資料庫位於 *instance-path*`/alias` 目錄中。

▼ 配置伺服器，讓使用者必須提供憑證密碼

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **停止伺服器。**

```
$ dsadm stop instance-path
```

2 **將密碼提示旗標設為 on。**

```
$ dsadm set-flags instance-path cert-pwd-prompt=on
```

系統會要求您選擇新的憑證密碼。

3 **啓動伺服器。**

```
$ dsadm start instance-path
```

為目錄伺服器備份及復原憑證資料庫

當您備份目錄伺服器的實例時，會備份目錄伺服器的配置與憑證。備份的憑證儲存在 *archive-path*/alias 目錄中。

如需有關如何備份及復原目錄伺服器的資訊，請參閱第 198 頁的 「製作嚴重損壞回復所需的備份」。

配置 SSL 通訊

本節包含停用及啓用 SSL 的相關程序。

停用非安全通訊

建立伺服器實例時，預設會建立 LDAP 明文連接埠與 LDAP 安全連接埠 (LDAPS)。但在某些情況下，可能會想停用非 SSL 通訊，而使伺服器通訊只能透過 SSL 進行。

SSL 連線啓用時會使用預設的自行簽署憑證。若您認爲有必要，也可安裝自己的憑證。如需在伺服器啓動後管理憑證及停用 SSL 的相關指示，請參閱第 6 章。如需憑證、憑證資料庫與取得 CA 簽署伺服器憑證的簡介，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」。

▼ 停用 LDAP 明文連接埠

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **停用 LDAP 明文連接埠。**

   若要停用非安全點，必須連結至 LDAP 安全點。此範例說明主機伺服器 `host1` 上的預設 LDAP 安全連接埠 `1636` 之連結。

   ```
   $ dsconf set-server-prop -h host1 -P 1636 ldap-port:disabled
   ```

2. **重新啓動伺服器，使變更生效。**

   ```
   $ dsadm restart /local/ds
   ```

   您現在再也不需要在非安全連接埠 1389 上進行連結。

選擇加密密碼

密碼是用於加密與解密資料的演算法。一般而言，密碼在加密期間所使用的位元數越多，加密就越**嚴密**或安全。SSL 的密碼也可經由使用的訊息認證類型來識別。訊息認證也是一種演算法，可計算能夠確保資料完整性的**總和檢查**。

當用戶端初始化與伺服器的 SSL 連線時，用戶端與伺服器必須協議出用以加密資訊的密碼。在任何雙向加密程序中，雙方都必須使用相同的密碼。所使用的密碼取決於伺服器所保存之密碼清單目前的順序。伺服器會選擇用戶端所呈現的密碼中，第一個符合其清單中之密碼的密碼。目錄伺服器的預設密碼值爲 `all`，代表基礎 SSL 程式庫所支援的所有已知安全密碼。但您也可以修改此值，而僅接受特定密碼。

如需有關目錄伺服器可用之密碼的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」。

▼ 選擇加密密碼

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **請確定您的伺服器已啓用 SSL。**

請參閱第 114 頁的 「配置 SSL 通訊」。

2 **檢視可用的 SSL 密碼。**

```
$ dsconf get-server-prop -h host -p port ssl-supported-ciphers
ssl-supported-ciphers  :  TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
ssl-supported-ciphers  :  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
ssl-supported-ciphers  :  TLS_DHE_RSA_WITH_AES_256_CBC_SHA
ssl-supported-ciphers  :  TLS_DHE_DSS_WITH_AES_256_CBC_SHA
...
```

3 **(可選擇) 若要保有非加密資料的副本，請在設定 SSL 密碼前先匯出資料。**

請參閱第 190 頁的 「匯出至 LDIF」。

4 **設定 SSL 密碼。**

```
$ dsconf set-server-prop -h host -p port ssl-cipher-family:cipher
```

例如，若要將密碼系列設爲 `SSL_RSA_WITH_RC4_128_MD5` 與 `SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA`，請鍵入：

```
$ dsconf set-server-prop -h host1 -P 1636 ssl-cipher-family:SSL_RSA_WITH_RC4_128_MD5 \
 ssl-cipher-family:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
Enter "cn=Directory Manager" password:
Before setting SSL configuration, export Directory Server data.
Do you want to continue [y/n] ? y
Directory Server must be restarted for changes to take effect.
```

5 **(可選擇) 將 SSL 密碼增加至現有清單。**

如果已指定密碼清單，且需要增加密碼，請使用此指令：

```
$ dsconf set-server-prop -h host -p port ssl-cipher-family+:cipher
```

例如，若要增加 SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA 密碼，請輸入：

```
$ dsconf set-server-prop -h host1 -P 1636 \
 ssl-cipher-family+:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
```

6 **重新啓動伺服器，使變更生效。**

```
$ dsadm restart /local/ds
```

配置憑證層級與認證方法

套用至用戶端的安全模式會透過憑證層級與認證方法的組合進行定義。

目錄伺服器支援下列憑證層級：

- **匿名**。允許匿名存取目錄的特定部分表示具有目錄存取權限的任何人皆有讀取權限。

  若使用匿名憑證層級，則必須授與所有 LDAP 命名項目與屬性的讀取權限。

注意 – 請勿允許目錄的匿名寫入權限，因爲任何人皆可變更其具有寫入權限的 DIT 資訊，包含其他使用者的密碼或其本身的身份識別。

- **代理**。用戶端使用代理帳號進行認證或連結至目錄。

  代理帳號可以是允許連結至目錄的任何項目。代理帳號需要足夠的存取權限，才能執行目錄上的命名服務功能。您必須使用代理憑證層級在每個用戶端上配置 proxyDN 與 proxyPassword。加密的 proxyPassword 儲存在本機用戶端上。
- **代理匿名**。其中定義了多個憑證層級的多重值項目。

  指定代理匿名層級的用戶端會先嘗試使用其代理身份識別進行認證。如果用戶端因爲任何原因 (例如使用者封鎖、密碼過期)，而無法以代理使用者的身份進行認證，則用戶端會使用匿名存取。根據目錄配置的方式，如此可能會導致不同的服務層級。

用戶端認證是伺服器驗證用戶端身份識別的機制。

用戶端認證可透過下列方式之一執行：

- 提供 DN 與密碼。
- 透過用戶端所呈現的憑證。

  憑證型認證會使用透過 SSL 協定取得的用戶端憑證，尋找使用者的身份識別項目。在憑證型認證中，用戶端會傳送可指定外部機制的 SASL 連結請求。此連結請求需依賴已建立的 SSL 認證機制。

- 透過 SASL 型機制。
  - 在所有作業系統上，SASL 均透過 DIGEST-MD5 運作。
  - 在 Solaris 作業系統上，SASL 會採用可讓用戶端認證透過 Kerberos V5 進行的 GSSAPI 機制。

  使用這兩個 SASL 機制之一時，必須同時配置伺服器，使其執行身份識別對映。SASL 憑證稱爲**主體**。這兩項機制都必須具有特定的對映，以決定「主體」內容的連結 DN。當「主體」對映至單一使用者項目，而 SASL 機制驗證該名使用者的身份識別時，使用者的 DN 即爲連線的連結 DN。
- 透過 SSL 用戶端認證模式。

  若想讓所有用戶端均在 SSL 層上進行授權，請使用 SSL 用戶端認證。用戶端應用程式會傳送其 SSL 憑證至伺服器，以進行認證。您可以使用 `SSL-client-auth-mode` 旗標指定伺服器允許、需要或不允許 SSL 用戶端認證。預設會允許用戶端進行認證。

本節提供下列有關在目錄伺服器上配置兩個 SASL 機制的資訊。

- 第 117 頁的 「設定目錄伺服器中的 SASL 加密層級」
- 第 118 頁的 「透過 DIGEST-MD5 的 SASL 認證」
- 第 121 頁的 「透過 GSSAPI 的 SASL 認證 (僅限 Solaris 作業系統)」

如需有關配置安全性的更多資訊，請參閱第 123 頁的 「配置 LDAP 用戶端以使用安全性」。

設定目錄伺服器中的 SASL 加密層級

配置 SASL 機制之前，必須先指定是否需要加密。SASL 加密的需求，由安全性強度係數 (SSF) 的最大值與最小值所設定。

`dsSaslMinSSF`(5dsat) 與 `dsSaslMaxSSF`(5dsat) 屬性代表加密金鑰長度，這些屬性儲存於 `cn=SASL, cn=security, cn=config` 之中。

伺服器允許任何層級的加密，包含不加密。這表示目錄伺服器接受大於 256 的 `dsSaslMinSSF` 與 `dsSaslMaxSSF` 值。但目前沒有任何 SASL 機制支援大於 128 的 SSF。目錄伺服器會在 SSF 可用的最大值 (128) 以內協調這些值。因此，視可用的基礎機制之不同，SSF 實際上的最大值可能會小於配置的最大值。

SASL 安全性係數認證取決於兩個主要項目：伺服器與用戶端應用程式所請求的最小與最大係數，以及基礎安全性元件所提供的可用加密機制。概略地說，伺服器與用戶端會嘗試使用小於或等於兩者之最大設定係數，但大於或等於兩者之最小係數的最高可用安全性係數。

目錄伺服器預設的最小 SASL 安全性係數 `dsSaslMinSSF` 爲 `0`，表示沒有任何保護。除非您變更目錄伺服器的最小值，否則實際上的最小值取決於用戶端設定。實務上應將最小值設爲您實際上要讓伺服器與用戶端使用的最低層級。若伺服器與用戶端無法協調出符合最小需求的機制，則不會建立連線。

▼ 需要 SASL 加密

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **若需要 SASL 加密，請將 `dsSaslMinSSF` 值設為所需的最小加密。**

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=SASL, cn=security, cn=config
changetype: modify
replace: dsSaslMinSSF
dsSaslMinSSF: 128
^D
```

▼ 不允許 SASL 加密

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **若不允許 SASL 加密，請將 `dsSaslMinSSF` 與 `dsSaslMaxSSF` 值都設為零。**

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=SASL, cn=security, cn=config
changetype: modify
replace: dsSaslMinSSF
dsSaslMinSSF: 0

replace: dsSaslMaxSSF
dsSaslMaxSSF: 0
```

透過 DIGEST-MD5 的 SASL 認證

DIGEST-MD5 機制會比較用戶端傳送的雜湊值與使用者密碼的雜湊值，以認證用戶端。但由於此機制必須讀取使用者密碼，因此所有想透過 DIGEST-MD5 進行認證的使用者，在目錄中都必須要有 `{CLEAR}` 密碼。將 `{CLEAR}` 密碼儲存到目錄時，您必須如第 7 章中所述，確實透過 ACI 適當地限制密碼值的存取權。此外，您還必須如第 99 頁的「爲屬性值加密」中所述，在尾碼中配置屬性加密。

▼ 配置 DIGEST-MD5 機制

下列程序說明如何配置目錄伺服器以使用 DIGEST-MD5。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **使用 `ldapsearch` 指令，驗證 `DIGEST-MD5` 是根項目上的 `supportedSASLMechanisms` 屬性值。**

   例如，下列指令可顯示哪些是啓用的 SASL 機制：

   ```
   $ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
    -s base -b "" "(objectclass=*)" supportedSASLMechanisms
   Enter bind password:
   dn:
   supportedSASLMechanisms: EXTERNAL
   supportedSASLMechanisms: DIGEST-MD5
   supportedSASLMechanisms: GSSAPI
   ```

2. **若 DIGEST-MD5 尚未啓用，請加以啓用。**

   ```
   $ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
   Enter bind password:
   dn: cn=SASL, cn=security, cn=config
   changetype: modify
   add: dsSaslPluginsEnable
   dsSaslPluginsEnable: DIGEST-MD5
   -
   replace: dsSaslPluginsPath
   dsSaslPluginsPath: SASL-library
   ```

   其中 *SASL-library* 爲下列其中一項：

   JES 安裝　　`/usr/lib/mps/sasl2`

   Zip 安裝　　*install-path*`/dsee6/private/lib`

3. **針對 DIGEST-MD5 使用預設身份識別對映，或建立新的對映。**

   如需相關資訊，請參閱第 119 頁的 「DIGEST-MD5 身份識別對映」。

4. **請確定所有將使用 DIGEST-MD5 而透過 SSL 存取伺服器的使用者，均已將密碼儲存在 `{CLEAR}` 中。**

   如需密碼儲存機制，請參閱第 8 章。

5. **若修改了 SASL 配置項目或任何 DIGEST-MD5 身份識別對映項目，請重新啓動目錄伺服器。**

DIGEST-MD5 身份識別對映

SASL 機制的身份識別對映會嘗試比對 SASL 身份識別的憑證與目錄中的使用者項目。如果在對映中找不到對應 SASL 身份識別的 DN，認證即會失敗。如需此機制的完整描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」。

SASL 身份識別是一個名爲 *Principal* 的字串，用以表示各機制之特定格式下的使用者。在 DIGEST-MD5 中，用戶端應建立含有 `dn:` 前綴與 LDAP DN，或含有 `u:` 前綴與任何用戶端指定文字的主體。對映期間，用戶端所傳送的主體可用於 `${Principal}` 預留位置中。

伺服器配置的下列項目，即爲 DIGEST-MD5 的預設身份識別對映：

```
dn: cn=default,cn=DIGEST-MD5,cn=identity mapping,cn=config
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
objectClass: dsPatternMatching
cn: default
dsMatching-pattern: \${Principal}
dsMatching-regexp: dn:(.*)
dsMappedDN: \$1
```

此身份識別對映會假設主體的 `dn` 欄位含有現有使用者在目錄中的實際 DN。

▼ 定義您自己的 DIGEST-MD5 身份識別對映

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **編輯預設對映項目，或在 `cn=DIGEST-MD5,cn=identity mapping,cn=config` 下建立新的對映項目。**

   下列指令將說明此對映的定義方式：

   ```
   $ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
   Enter bind password:
   dn: cn=unqualified-username,cn=DIGEST-MD5,cn=identity mapping
   cn=config
   objectclass: dsIdentityMapping
   objectclass: dsPatternMatching
   objectclass: nsContainer
   objectclass: top
   cn: unqualified-username
   dsMatching-pattern: \${Principal}
   dsMatching-regexp: u:(.*)@(.*)\\.com
   dsSearchBaseDN: dc=\$2
   dsSearchFilter: (uid=\$1)
   ```

2. **重新啓動目錄伺服器，使您的新對映生效。**

透過 GSSAPI 的 SASL 認證 (僅限 Solaris 作業系統)

透過 SASL 的通用安全服務 API (GSSAPI) 可讓您使用協力廠商的安全性系統 (例如 Kerberos V5) 來認證用戶端。GSSAPI 程式庫僅適用於 Solaris 作業系統 SPARC® 平台。Sun 建議您在 Sun Enterprise Authentication Mechanism™ 1.0.1 伺服器上安裝 Kerberos V5 實作。

此伺服器會使用 GSSAPI 驗證使用者的身份識別。接著，SASL 機制會套用 GSSAPI 對映規則，以取得此連線期間對所有作業而言皆爲連結 DN 的 DN。

▼ 配置 Kerberos 系統

根據製造商的指示配置 Kerberos 軟體。若您使用 Sun Enterprise Authentication Mechanism 1.0.1 伺服器，請執行此程序。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **配置 `/etc/krb5` 中的檔案。**
2. **建立用以儲存使用者與服務的 Kerberos 資料庫。**
3. **在資料庫中，建立 LDAP 服務的主體。**

   ```
   $ ldap/server-FQDN@realm
   ```

   其中 *server-FQDN* 是您目錄伺服器完全合格的網域名稱。
4. **啓動 Kerberos 常駐程式程序。**

備註 – DNS 必須配置於主機電腦上。

如需這些步驟的詳細指示，請參閱軟體文件。另請參閱第 126 頁的 「使用 GSSAPI 與 SASL 配置 Kerberos 認證的範例」。

▼ 配置 GSSAPI 機制

下列程序說明如何配置目錄伺服器以在 Solaris 作業系統上使用 GSSAPI：

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **如第 122 頁的 「GSSAPI 身份識別對映」中所述，建立 GSSAPI 的預設身份識別對映與所有自訂對映。**

2. **建立用以儲存服務金鑰的 keytab。**
   您的 LDAP 服務金鑰會儲存在 keytab 中。

   a. **請確定只有目錄伺服器使用者可讀取 keytab。**

   b. **將檔案名稱變更為預設值 `/etc/krb5/krb5.keytab` 以外的名稱。**

   c. **設定環境變數 `KRB5_KTNAME`，以確保所使用的是新的 keytab，而非預設 keytab。**

3. **若修改了 SASL 配置項目或任何 GSSAPI 身份識別對映項目，請重新啓動目錄伺服器。**
   請注意，DNS 必須配置於主機電腦上。

GSSAPI 身份識別對映

SASL 機制的身份識別對映會嘗試比對 SASL 身份識別的憑證與目錄中的使用者項目。如果在對映中找不到對應 SASL 身份識別的 DN，認證即會失敗。

SASL 身份識別是一個名爲 *Principal* 的字串，用以表示各機制之特定格式下的使用者。在 Kerberos 中使用 GSSAPI 時，主體將是 *uid* [/*instance*] [@*realm*] 格式的身份識別。*uid* 可包含一個可選擇的 *instance* 識別碼，再加上通常爲網域名稱的可選擇 *realm*。例如，下列字串皆爲有效的使用者主體：

```
bjensen
bjensen/Sales
bjensen@EXAMPLE.COM
bjensen/Sales@EXAMPLE.COM
```

目錄中最初並未定義 GSSAPI 對映。根據您的用戶端定義所使用之主體的方式，定義預設對映以及所需要的任何自訂對映。

▼ 定義 GSSAPI 的身份識別對映

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **在 `cn=GSSAPI,cn=identity mapping, cn=config` 下建立新的對映項目。**
   如需身份識別對映項目中的屬性定義，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」。*instance-path*`/ldif/identityMapping_Examples.ldif` 中含有 GSSAPI 的對映範例。

   此檔案中的預設 GSSAPI 對映，會假設主體中僅包含一個使用者 ID。此對映可判斷位在目錄之固定分支中的使用者：

```
dn: cn=default,cn=GSSAPI,cn=identity mapping,cn=config
objectclass: dsIdentityMapping
objectclass: nsContainer
```

```
objectclass: top
cn: default
dsMappedDN: uid=\${Principal},ou=people,dc=example,dc=com
```

此檔案中的另一個範例說明如何在使用者 ID 位於含有已知範圍的主體中時，判斷使用者 ID。

```
dn: cn=same_realm,cn=GSSAPI,cn=identity mapping,cn=config
objectclass: dsIdentityMapping
objectclass: dsPatternMatching
objectclass: nsContainer
objectclass: top
cn: same_realm
dsMatching-pattern: \${Principal}
dsMatching-regexp: (.*)@EXAMPLE.COM
dsMappedDN: uid=\$1,ou=people,dc=EXAMPLE,dc=COM
```

2 **重新啓動目錄伺服器，使您的新對映生效。**

配置 LDAP 用戶端以使用安全性

以下小節說明如何在需要建立與目錄伺服器的安全連線之 LDAP 用戶端中，配置與使用 SSL。在 SSL 連線中，伺服器會將其憑證傳送至用戶端。用戶端必須先信任此憑證，以進行伺服器認證。接著，用戶端可選擇針對兩個 SASL 機制之一傳送其本身的憑證或資訊，以初始化其中一個用戶端認證機制。SASL 機制爲 DIGEST-MD5 與使用 Kerberos V5 的 GSSAPI。

下列小節將以 `ldapsearch` 工具做爲啓用 SSL 之 LDAP 用戶端的範例。

若要在其他 LDAP 用戶端上配置 SSL 連線，請參閱應用程式隨附的文件。

備註 – 有一些用戶端應用程式可實作 SSL，但無法驗證伺服器是否具有可信任的憑證。這些用戶端應用程式可使用 SSL 協定進行資料加密，但無法確保機密度，也無法防止模擬。

下列幾節說明如何配置 LDAP 用戶端以使用安全性：

在用戶端中使用 SASL DIGEST-MD5

當您在用戶端中使用 DIGEST-MD5 機制時，不需要安裝使用者憑證。但若要使用加密的 SSL 連線，則仍須如第 106 頁的 「管理憑證」中所述，信任伺服器憑證。

指定範圍

範圍定義選取認證身份識別的來源名稱空間。在 DIGEST-MD5 認證中，必須對特定的範圍進行認證。

目錄伺服器會以機器完全合格的主機名稱做爲 DIGEST-MD5 的預設範圍。伺服器會使用位於 `nsslapd-localhost` 配置屬性中的主機名稱小寫值。

若未指定範圍，則會使用伺服器所提供的預設範圍。

指定環境變數

在 UNIX 環境中，您必須設定 `SASL-PATH` 環境變數，LDAP 工具才找得到 DIGEST-MD5 程式庫。DIGEST-MD5 程式庫是一種可由 SASL 外掛程式動態載入的共用程式庫。請以下列方式設定 `SASL_PATH` 環境變數：

```
export SASL_PATH=SASL-library
```

此路徑假設目錄伺服器安裝於呼叫 LDAP 工具的相同主機上。

`ldapsearch` 指令範例

您可以在未使用 SSL 的情況下，執行 DIGEST-MD5 用戶端認證。下列範例將使用預設的 DIGEST-MD5 身份識別對映，判斷連結 DN：

```
$ ldapsearch -h host1 -p 1389 \
 -o mech=DIGEST-MD5 [ \
 -o realm="example.com"] \
 -o authid="dn:uid=bjensen,dc=example,dc=com" \
 -w - \
 -o authzid="dn:uid=bjensen,dc=example,dc=com" \
 -o secProp="minssf=56,maxssf=256,noplain" \
 -b "dc=example,dc=com" "(givenname=Richard)"
```

上述範例說明如何使用 `-o` (小寫字母 o) 選項，指定 SASL 選項。`realm` 是可選擇的項目，但若要指定此項目，則必須使用伺服器主機電腦完全合格的網域名稱。`authid` 與 `authzid` 皆須存在同時完全相同，但並不會使用爲代理伺服器作業所設定的 `authzid`。`-w` 密碼選項會套用至 `authid`。

`authid` 的值爲身份識別對映中所使用的主體。`authid` 應包含 `dn:` 前綴與目錄中的有效使用者 DN，或包含 `u:` 前綴與用戶端所決定的任何字串。`authid` 的此項用法可讓您使用第 119 頁的 「DIGEST-MD5 身份識別對映」中所述的對映。

讓 SSL 連線透過 LDAPS 安全連接埠提供加密，以及讓 DIGEST-MD5 提供用戶端認證，是最常見的配置。下列範例將透過 SSL 執行相同的作業：

```
$ ldapsearch -h host1 -P 1636 \
 -Z -P .mozilla/bjensen/BJE6001.slt/cert8.db \
 -N "cert-example" -w - \
 -o mech=DIGEST-MD5 [-o realm="example.com"] \
 -o authid="dn:uid=bjensen,dc=example,dc=com" \
 -o authzid="dn:uid=bjensen,dc=example,dc=com" \
 -o secProp="minssf=0,maxssf=0,noplain" \
 -b "dc=example,dc=com" "(givenname=Richard)"
```

在此範例中，由於作業會透過 SSL 執行，因此 `ldapsearch` 指令必須使用 `-N` 與 `-w` 選項。但這些選項並不會使用於用戶端認證上。伺服器會在 `authid` 值中執行主體的其他 DIGEST-MD5 身份識別對映。

在用戶端中使用 Kerberos SASL GSSAPI

當您在用戶端中使用 GSSAPI 機制時，不需要安裝使用者憑證，但是必須配置 Kerberos V5 安全性系統。此外，若要使用加密的 SSL 連線，則必須如第 106 頁的 「管理憑證」中所述，信任伺服器憑證。

▼ 在主機上配置 Kerberos V5

您必須在將要執行 LDAP 用戶端的主機電腦上配置 Kerberos V5。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **請根據安裝指示進行 Kerberos V5 的安裝。**

   Sun 建議安裝 Sun Enterprise Authentication Mechanism 1.0.1 用戶端軟體。

2. **配置 Kerberos 軟體。**

   使用 Sun Enterprise Authentication Mechanism 軟體，配置 `/etc/krb5` 下的檔案。此配置可設定 `kdc` 伺服器，並定義預設範圍與您的 Kerberos 系統所需的所有其他配置。

3. **請視需要修改檔案 `/etc/gss/mech`，使 `kerberos_v5` 成為第一個列出的值。**

▼ 指定 Kerberos 認證的 SASL 選項

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **使用以 GSSAPI 機制啓用的用戶端應用程式前，請先以使用者主體初始化 Kerberos 安全性系統。**

   ```
   $ kinit user-principal
   ```

   其中 *user-principal* 是您的 SASL 身份識別，如 `bjensen@example.com`。

2 **指定使用 Kerberos 時所需的 SASL 選項。**

請注意，在 UNIX 環境中，必須將 SASL_PATH 環境變數設爲 SASL 程式庫的正確路徑。以 Korn shell 爲例：

```
$ export SASL_PATH=SASL-library
```

此路徑假設目錄伺服器安裝於呼叫 LDAP 工具的相同主機上。

ldapsearch 工具的下列範例說明如何使用 -o (小寫字母 o) 選項，指定使用 Kerberos 時所需的 SASL 選項：

```
$ ldapsearch -h www.host1.com -p 1389 -o mech=GSSAPI -o authid="bjensen@EXAMPLE.COM" \
 -o authzid="bjensen@EXAMPLE.COM" -b "dc=example,dc=com" "(givenname=Richard)"
```

authid 可以省略，因爲它會出現在 kinit 指令所初始化的 Kerberos 快取中。如果 authid 存在，authid 與 authzid 必須完全相同，但並不會使用爲代理伺服器作業所設定的 authzid。authid 的值爲身份識別對映中所使用的主體。主體必須是完整的，其中必須包含範圍。請參閱第 122 頁的 「GSSAPI 身份識別對映」。

使用 GSSAPI 與 SASL 配置 Kerberos 認證的範例

爲目錄伺服器配置 Kerberos 有時是很複雜的作業。您應以 Kerberos 文件做爲首要參考資料。

如需更多說明，請以下列範例程序評估所應執行的步驟。但請注意，此程序僅是範例。您必須根據本身的配置與環境，適當地修改此程序。

如需有關在 Solaris 作業系統中配置及使用 Kerberos 的更多資訊，請參閱「System Administration Guide: Security Services」。此指南是 Solaris 文件集的一部分。您也可以參考線上手冊。

此範例與前述步驟的相關資訊如下：

1. 第 127 頁的 「此範例的假設」
2. 第 127 頁的 「所有機器：編輯 Kerberos 用戶端配置檔」
3. 第 129 頁的 「所有機器：編輯管理伺服器 ACL 配置檔」
4. 第 129 頁的 「KDC 機器：編輯 KDC 伺服器配置檔」
5. 第 130 頁的 「KDC 機器：建立 KDC 資料庫」
6. 第 130 頁的 「KDC 機器：建立管理主體與 Keytab」
7. 第 130 頁的 「KDC 機器：啓動 Kerberos 常駐程式」
8. 第 131 頁的 「KDC 機器：爲 KDC 與目錄伺服器機器增加主機主體」
9. 第 131 頁的 「KDC 機器：增加目錄伺服器的 LDAP 主體」
10. 第 132 頁的 「KDC 機器：爲 KDC 增加測試使用者」
11. 第 132 頁的 「目錄伺服器機器：安裝目錄伺服器」
12. 第 132 頁的 「目錄伺服器機器：配置目錄伺服器以啓用 GSSAPI」
13. 第 133 頁的 「目錄伺服器機器：建立目錄伺服器 Keytab」
14. 第 134 頁的 「目錄伺服器機器：將測試使用者增加至目錄伺服器」

15. 第 135 頁的 「目錄伺服器機器：以測試使用者的身份取得 Kerberos 票證」
16. 第 135 頁的 「用戶端機器：透過 GSSAPI 對目錄伺服器進行認證」

此範例的假設

此範例中的程序說明如何將一部機器配置成爲金鑰分配中心 (KDC)，並將另一部機器配置成執行目錄伺服器的機器。完成此程序後，使用者即可透過 GSSAPI 執行 Kerberos 認證。

在同一部機器上可同時執行 KDC 與目錄伺服器。若選擇在同一部機器上同時執行兩者，請使用相同的程序，但可在目錄伺服器機器的步驟中省略已對 KDC 機器執行過的部分。

此程序對於所使用的環境有許多相關假設。使用範例程序時，請根據您的環境適當地修改其值。這些假設包含：

- 此系統安裝了全新的 Solaris 9 軟體，以及最新的建議修補程式叢集。若未安裝適當的 Solaris 修補程式，目錄伺服器的 Kerberos 認證即可能失敗。

  請注意，此處所列之程序雖然大致與 Solaris 10 的程序相同，但仍有某些不同之處。配置檔的格式略有不同，某些指令的輸出也可能不同。
- 執行 Kerberos 常駐程式的機器具有 `kdc.example.com` 完全合格的網域名稱。此機器必須配置成以 DNS 做爲命名服務。Kerberos 必須進行此配置。若改用其他如 `file` 的命名服務，特定作業將因此失敗。
- 執行目錄伺服器的機器具有 `directory.example.com` 完全合格的網域名稱。此機器也必須配置成以 DNS 做爲命名服務。
- 目錄伺服器機器會做爲透過 Kerberos 對目錄伺服器進行認證時的用戶端系統。此認證可從任何能夠同時與目錄伺服器和 Kerberos 常駐程式進行通訊的系統執行。然而，此範例的所有必要元件皆隨附於目錄伺服器，而認證會從該系統執行。
- 目錄伺服器中的使用者具有格式爲 `uid=` *username*`,ou=People,dc=example,dc=com` 的 DN。對應的 Kerberos 主體爲 *username*`@EXAMPLE.COM`。若使用不同的命名機制，則必須使用不同的 GSSAPI 身份識別對映。

所有機器：編輯 Kerberos 用戶端配置檔

`/etc/krb5/krb5.conf` 配置檔中含有可讓 Kerberos 用戶端與 KDC 通訊的必要資訊。

在 KDC 機器、目錄伺服器機器，以及任何將使用 Kerberos 對目錄伺服器進行認證的用戶端機器上，編輯 `/etc/krb5/krb5.conf` 配置檔。

- 將您所看到的每個 `"___default_realm___"` 取代爲 `"EXAMPLE.COM"`。
- 將您所看到的每個 `"___master_kdc___"` 取代爲 `"kdc.example.com"`。
- 由於只會有一部 Kerberos 伺服器，因此請移除含有 `"___slave_kdcs___"` 的文字行。

- 將 "___domain_mapping___" 取代爲 ".example.com = EXAMPLE.COM" (請注意 .example.com 開頭處的句點)。

更新的 /etc/krb5/krb5.conf 配置檔應如下列範例的內容所示。

範例 6–1 編輯後的 Kerberos 用戶端配置檔 /etc/krb5/krb5.conf

```
#pragma ident   "@(#)krb5.conf  1.2     99/07/20 SMI"
# Copyright (c) 1999, by Sun Microsystems, Inc.
# All rights reserved.
#
# krb5.conf template
# In order to complete this configuration file
# you will need to replace the __<name\>__ placeholders
# with appropriate values for your network.
#

[libdefaults]
        default_realm = EXAMPLE.COM
[realms]
        EXAMPLE.COM = {
                kdc = kdc.example.com
                admin_server = kdc.example.com
        }
[domain_realm]
        .example.com = EXAMPLE.COM
[logging]
        default = FILE:/var/krb5/kdc.log
        kdc = FILE:/var/krb5/kdc.log
        kdc_rotate = {

# How often to rotate kdc.log. Logs will get rotated no more
# often than the period, and less often if the KDC is not used
# frequently.
                period = 1d

# how many versions of kdc.log to keep around (kdc.log.0, kdc.log.1, ...)
                versions = 10
        }

[appdefaults]
        kinit = {
                renewable = true
                forwardable= true
        }
        gkadmin = {
                help_url =
```

範例 6–1 編輯後的 Kerberos 用戶端配置檔 /etc/krb5/krb5.conf （續）

```
 http://docs.sun.com:80/ab2/coll.384.1/SEAM/@AB2PageView/1195
        }
```

所有機器：編輯管理伺服器 ACL 配置檔

在 /etc/krb5/kadm5.acl 配置檔中，將 "___default_realm___" 取代爲 "EXAMPLE.COM"。更新的檔案應如下列範例所示。

範例 6–2 編輯後的管理伺服器 ACL 配置檔

```
#
# Copyright (c) 1998-2000 by Sun Microsystems, Inc.
# All rights reserved.
#
# pragma ident   "@(#)kadm5.acl  1.1     01/03/19 SMI"
*/admin@EXAMPLE.COM *
```

KDC 機器：編輯 KDC 伺服器配置檔

編輯 /etc/krb5/kdc.conf 檔案，將 "___default_realm___" 取代爲 "EXAMPLE.COM"。更新的檔案應如下列範例所示。

範例 6–3 編輯後的 KDC 伺服器配置檔 /etc/krb5/kdc.conf

```
# Copyright 1998-2002 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident  "@(#)kdc.conf   1.2     02/02/14 SMI"

[kdcdefaults]
        kdc_ports = 88,750

[realms]
        EXAMPLE.COM = {
                profile = /etc/krb5/krb5.conf
                database_name = /var/krb5/principal
                admin_keytab = /etc/krb5/kadm5.keytab
                acl_file = /etc/krb5/kadm5.acl
                kadmind_port = 749
                max_life = 8h 0m 0s
                max_renewable_life = 7d 0h 0m 0s
                default_principal_flags = +preauth
        }
```

KDC 機器：建立 KDC 資料庫

```
$ /usr/sbin/kdb5_util create -r EXAMPLE.COM -s
Initializing database '/var/krb5/principal' for realm 'EXAMPLE.COM',
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key: password
Re-enter KDC database master key to verify: password
$
```

KDC 機器：建立管理主體與 Keytab

使用下列指令，建立具有 kws/admin@EXAMPLE.COM 主體與管理常駐程式所將使用之服務金鑰的管理使用者。

```
$ /usr/sbin/kadmin.local
kadmin.local:  add_principal kws/admin
Enter password for principal "kws/admin@EXAMPLE.COM": secret
Re-enter password for principal "kws/admin@EXAMPLE.COM": secret
Principal "kws/admin@EXAMPLE.COM" created.
kadmin.local:  ktadd -k /etc/krb5/kadm5.keytab kadmin/kdc.example.com
Entry for principal kadmin/kdc.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local:  ktadd -k /etc/krb5/kadm5.keytab changepw/kdc.example.com

Entry for principal changepw/kdc.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local:  ktadd -k /etc/krb5/kadm5.keytab kadmin/changepw
Entry for principal kadmin/changepw with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local:  quit$
```

KDC 機器：啓動 Kerberos 常駐程式

執行下列指令，以啓動 KDC 與管理常駐程式：

```
$ /etc/init.d/kdc start
$ /etc/init.d/kdc.master start
$
```

KDC 程序會以 /usr/lib/krb5/krb5kdc 的形式顯示在程序清單中。管理常駐程式會顯示爲 /usr/lib/krb5/kadmind。

請注意，Solaris 10 作業系統中的常駐程式係由「服務管理功能 (SMF)」架構所管理。在 Solaris 10 作業系統上啓動常駐程式：

```
$ svcadm disable network/security/krb5kdc
$ svcadm enable network/security/krb5kdc
$ svcadm disable network/security/kadmin
$ svcadm enable network/security/kadmin
$
```

KDC 機器：為 KDC 與目錄伺服器機器增加主機主體

使用以下一系列的指令，爲 KDC 與目錄伺服器機器的 Kerberos 資料庫增加主機主體。`klist` 等特定 Kerberos 公用程式會使用主機主體。

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  add_principal -randkey host/kdc.example.com
Principal "host/kdc.example.com@EXAMPLE.COM" created.
kadmin:  ktadd host/kdc.example.com
Entry for principal host/kdc.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:  add_principal -randkey host/directory.example.com
Principal "host/directory.example.com@EXAMPLE.COM" created.
kadmin:  ktadd host/directory.example.com
Entry for principal host/directory.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:  quit
$
```

KDC 機器：增加目錄伺服器的 LDAP 主體

目錄伺服器必須具有本身的主體，才能對進行認證的使用者驗證其所持有之 Kerberos 票證。目錄伺服器目前程序內定爲需要 `ldap/`*fqdn*`@`*realm* 的主體，其中 *fqdn* 是目錄伺服器的完全合格網域名稱，而 *realm* 是 Kerberos 範圍。*fqdn* 必須符合安裝目錄伺服器時所提供的完全合格名稱。在本例中，目錄伺服器的主體是 `ldap/directory.example.com@EXAMPLE.COM`。

使用以下一系列的指令，建立目錄伺服器的 LDAP 主體：

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin: add_principal -randkey ldap/directory.example.com
Principal "ldap/directory.example.com@EXAMPLE.COM" created.
kadmin: quit
$
```

KDC 機器：為 KDC 增加測試使用者

Kerberos 資料庫中必須有進行認證的使用者，才能執行 Kerberos 認證。在此範例中，使用者的使用者名稱爲 `kerberos-test`，這表示 Kerberos 主體爲 `kerberos-test@EXAMPLE.COM`。

使用此範例中的指令序列建立使用者：

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  add_principal kerberos-test
Enter password for principal "kerberos-test@EXAMPLE.COM": secret

Re-enter password for principal "kerberos-test@EXAMPLE.COM": secret

Principal "kerberos-test@EXAMPLE.COM" created.
kadmin:  quit
$
```

目錄伺服器機器：安裝目錄伺服器

安裝 Directory Server 6.0 與最新的修補程式。以下是範例設定。

變數類型	範例值
完全合格的電腦名稱	`directory.example.com`
安裝目錄	`/opt/SUNWdsee`
實例路徑	`/local/ds`
伺服器使用者	`unixuser`
伺服器群組	`unixgroup`
伺服器連接埠	`389`
尾碼	`dc=example,dc=com`

目錄伺服器機器：配置目錄伺服器以啓用 GSSAPI

首先建立 `/data/ds/shared/bin/gssapi.ldif` 檔案以定義目錄伺服器所應使用的對映，再根據主體識別進行認證的 Kerberos 使用者。建立下列範例中所顯示的相同檔案內容。

範例 6–4 gssapi.ldif 檔案內容

```
dn: cn=GSSAPI,cn=identity mapping,cn=config
changetype: add
objectClass: top
objectClass: nsContainer
cn: GSSAPI
dn: cn=default,cn=GSSAPI,cn=identity mapping,cn=config
changetype: add
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
objectClass: dsPatternMatching
cn: default
dsMatching-pattern: \${Principal}
dsMatching-regexp: (.*)@EXAMPLE.COM
dsMappedDN: uid=\$1,ou=People,dc=example,dc=com

dn: cn=SASL,cn=security,cn=config
changetype: modify
replace: dsSaslPluginsPath
dsSaslPluginsPath: /usr/lib/mps/sasl2/libsasl.so
```

接著使用 `ldapmodify` 指令更新目錄伺服器，以啓用具有適當對映的 GSSAPI，如下列範例所示：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -a -f /data/ds/shared/bin/gssapi.ldif
adding new entry cn=GSSAPI,cn=identity mapping,cn=config
adding new entry cn=default,cn=GSSAPI,cn=identity mapping,cn=config
modifying entry cn=SASL,cn=security,cn=config
$
```

目錄伺服器機器：建立目錄伺服器 Keytab

如前所述，目錄伺服器必須在 KDC 中具有本身的主體，才能透過 GSSAPI 認證 Kerberos 使用者。爲使認證正確運作，主體資訊必須位於目錄伺服器機器的 Kerberos keytab 中。此項資訊必須位於執行目錄伺服器的使用者帳號可讀取的檔案中。

使用下列指令序列，建立具有正確特性的 keytab 檔案：

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  ktadd -k /local/ds/config/ldap.keytab ldap/directory.example.com
Entry for principal ldap/directory.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab
 WRFILE:/local/ds/config/ldap.keytab.
kadmin:  quit
$
```

變更此自訂 keytab 中的權限與所有權。讓用以執行目錄伺服器的使用者帳號擁有此 keytab，且只有該名使用者可加以讀取：

```
$ chown unixuser:unixgroup /local/ds/config /ldap.keytab
$ chmod 600 /local/ds/config/ldap.keytab
$
```

目錄伺服器預設會嘗試使用 `/etc/kerb5/krb5.keytab` 檔案中的標準 Kerberos keytab。但若讓目錄伺服器使用者可讀取此檔案，將會構成安全性風險，爲目錄伺服器建立自訂 keytab 的用意即在於此。

配置目錄伺服器以使用新的自訂 keytab。請設定 `KRB5_KTNAME` 環境變數，以執行此作業。

最後，請重新啓動目錄伺服器，變更方可生效：

```
$ KRB5_KTNAME=/etc/krb5/ldap.keytab dsadm restart /local/ds
```

目錄伺服器機器：將測試使用者增加至目錄伺服器

若要對目錄伺服器認證 Kerberos 使用者，使用者必須有目錄項目可對應於其 Kerberos 主體。

在上一個步驟中，測試使用者以 `kerberos-test@EXAMPLE.COM` 的主體增加到 Kerberos 資料庫中。由於身份識別對映配置已增加到目錄中，因此該名使用者的對應目錄項目必須具有 DN `uid=kerberos-test,ou=People,dc=example,dc=com`。

在目錄中增加使用者之前，必須以下列內容建立檔案 `testuser.ldif`。

範例 6–5 新的 `testuser.ldif` 檔案

```
dn: uid=kerberos-test,ou=People,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: kerberos-test
givenName: Kerberos
sn: Test
cn: Kerberos Test
description: An account for testing Kerberos authentication through GSSAPI
```

接著，請使用 `ldapmodify` 將此項目增加到伺服器中：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -f testuser.ldif
adding new entry uid=kerberos-test,ou=People,dc=example,dc=com
$
```

目錄伺服器機器：以測試使用者的身份取得 Kerberos 票證

測試使用者存在於 Kerberos 資料庫、目錄伺服器與 KDC 中。因此，現在可以透過 GSSAPI 使用 Kerberos，利用測試使用者的身份，對目錄伺服器進行認證。

首先，請使用 `kinit` 指令取得使用者的 Kerberos 票證，如下列範例所示：

```
$ kinit kerberos-test
Password for kerberos-test@EXAMPLE.COM: secret
$
```

接著，請使用 `klist` 指令檢視此票證的相關資訊：

```
$ klist
Ticket cache: /tmp/krb5cc_0
Default principal: kerberos-test@EXAMPLE.COM

Valid starting            Expires                   Service principal
Sat Jul 24 00:24:15 2004  Sat Jul 24 08:24:15 2004  krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until Sat Jul 31 00:24:15 2004
$
```

用戶端機器：透過 GSSAPI 對目錄伺服器進行認證

最後一個步驟是使用 GSSAPI，對目錄伺服器進行認證。目錄伺服器隨附的 `ldapsearch` 公用程式可支援 SASL 認證，包含 GSSAPI、DIGEST-MD5 與 EXTERNAL 等機制。但若要使用 GSSAPI 進行連結，則必須爲用戶端提供 SASL 程式庫的路徑。藉由設定 `SASL_PATH` 環境變數提供 `lib/sasl` 目錄的路徑：

```
$ SASL_PATH=SASL-library
$ export SASL_PATH
$
```

若要使用 `ldapsearch` 實際執行目錄伺服器的 Kerberos 型認證，您必須包含 `-o mech=GSSAPI` 與 `-o authzid=`*principal* 引數。

此外，還必須指定完全合格的主機名稱，在此案例中爲 `-h directory.example.com`，此名稱必須與伺服器之 `cn=config` 的 `nsslapd-localhost` 屬性值相符。在此必須使用 `-h` 選項，因爲 GSSAPI 認證程序中必須要有用戶端所提供的主機名稱，用以比對伺服器所提供的主機名稱。

下列範例將在 `dc=example,dc=com` 項目認證爲先前所建立的 Kerberos 測試使用者帳號時，擷取此項目：

```
$ ldapsearch -h directory.example.com -p 389 -o mech=GSSAPI \
 -o authzid="kerberos-test@EXAMPLE.COM" -b "dc=example,dc=com" -s base "(objectClass=*)"
version: 1
```

```
dn: dc=example,dc=com
dc: example
objectClass: top
objectClass: domain
$
```

查看目錄伺服器存取記錄，以確定認證是否已如預期般進行處理：

```
$ tail -12 /local/ds/logs/access

[24/Jul/2004:00:30:47 -0500] conn=0 op=-1 msgId=-1 - fd=23 slot=23 LDAP
        connection from 1.1.1.8 to 1.1.1.8
[24/Jul/2004:00:30:47 -0500] conn=0 op=0 msgId=1 - BIND dn="" method=sasl
     version=3 mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=0 msgId=1 - RESULT err=14 tag=97
     nentries=0 etime=0, SASL bind in progress
[24/Jul/2004:00:30:47 -0500] conn=0 op=1 msgId=2 - BIND dn="" method=sasl
     version=3 mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=1 msgId=2 - RESULT err=14 tag=97
     nentries=0 etime=0, SASL bind in progress
[24/Jul/2004:00:30:47 -0500] conn=0 op=2 msgId=3 - BIND dn="" method=sasl
     version=3 mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=2 msgId=3 - RESULT err=0 tag=97
     nentries=0 etime=0 dn="uid=kerberos-test,ou=people,dc=example,dc=com"
[24/Jul/2004:00:30:47 -0500] conn=0 op=3 msgId=4 - SRCH base="dc=example,dc=com"
     scope=0 filter="(objectClass=*)" attrs=ALL
[24/Jul/2004:00:30:47 -0500] conn=0 op=3 msgId=4 - RESULT err=0 tag=101 nentries=1
     etime=0
[24/Jul/2004:00:30:47 -0500] conn=0 op=4 msgId=5 - UNBIND
[24/Jul/2004:00:30:47 -0500] conn=0 op=4 msgId=-1 - closing - U1
[24/Jul/2004:00:30:48 -0500] conn=0 op=-1 msgId=-1 - closed.
$
```

此範例說明連結爲三步驟的程序。前兩個步驟會傳回 LDAP 結果 14 (SASL 連結進行中)，而第三個步驟說明連結已順利完成。`method=sasl` 與 `mech=GSSAPI` 標記說明連結採用 GSSAPI SASL 機制。成功連結回應結尾處的 `dn="uid=kerberos-test,ou=people,dc=example,dc=com"`，說明連結已由適當的使用者身份執行。

傳遞式認證

傳遞式認證 (PTA) 是一項可讓連結請求據以依照連結 DN 進行篩選的機制。一部目錄伺服器 (委託方) 在收到連結請求後，即可根據篩選器洽詢其他目錄伺服器 (委託)，以認證連結請求。此外，此功能中的 PTA 外掛程式還可讓委託方目錄伺服器，針對未必儲存在其本機資料庫中的項目，接受簡單密碼型連結作業。

DSCC 亦可使用 PTA 外掛程式，與伺服器進行私密通訊。在 DSCC 中註冊伺服器實例時，即會啓用 PTA 外掛程式，並將 DSCC URL 增加爲引數。

```
$ dsconf get-plugin-prop -h host -p port "Pass Through Authentication" enabled argument
argument   :  ldap://DSCC_URL:DSCC_PORT/cn=dscc
enabled    :  on
```

備註 – 請儘可能避免在您自己的使用中修改 PTA 外掛程式。修改 PTA 外掛程式可能會導致 DSCC 的存取問題。

若必須修改 PTA 外掛程式，請務必執行下列動作：

- 將 `enabled` 特性保持爲 `on`。
- 即使您可在 `argument` 特性中增加其他值，但仍請於引數中保留 DSCC URL。

若已停用 PTA 外掛程式，或 DSCC URL 已從引數中移除，則伺服器實例在 DSCC 中將顯示爲 `inaccessible`。若發生此情況，DSCC 會自動讓您選擇是否要重設 PTA 外掛程式。

◆ ◆ ◆ 第 7 章

7 目錄伺服器存取控制

控制對目錄的存取是建立安全目錄不可或缺的一部分。本章將說明存取控制指令 (ACI)，這些指令可決定要授予存取目錄之使用者的權限。

請在目錄部署的規劃階段過程，即定義適用整體安全性策略的存取控制策略。如需有關規劃存取控制策略的提示，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」。

如需有關 ACI 的其他資訊，包含 ACI 語法與連結規則，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」。

本章包含下列主題：

- 第 139 頁的「建立、檢視及修改 ACI」
- 第 141 頁的「存取控制用法範例」
- 第 152 頁的「檢視有效權限」
- 第 156 頁的「進階存取控制：使用巨集 ACI」
- 第 161 頁的「記錄存取控制資訊」
- 第 162 頁的「使用 TCP 包裝的用戶端主機存取控制」

建立、檢視及修改 ACI

您可以使用目錄服務控制中心 (DSCC) 或指令行建立 ACI。無論選擇何種方式，檢視及複製現有的 ACI 值，通常會比重新建立新的 ACI 來得容易。

您可以在 DSCC 中檢視及修改 `aci` 屬性值。如需有關如何透過 DSCC 修改 ACI 的資訊，請參閱 DSCC 線上說明。

▼ 建立、修改及刪除 ACI

若要透過使用指令行建立 ACI，必須先使用 LDIP 陳述式在檔案中建立 ACI。接著，必須透過使用 `ldapmodify` 指令將 ACI 增加到您的目錄樹狀結構中。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **在 LDIF 檔案中建立 ACI。**

```
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (target)(version 3.0; acl "name";permission bindrules;)
```

此範例說明增加 ACI 的方式。若要修改或刪除 ACI，請以 `replace` 或 `delete` 取代 `add`。

如需更多常用 ACI 的範例，請參閱第 141 頁的 「存取控制用法範例」。

2 **使用 LDIF 檔案進行變更。**

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w - -f ldif-file
```

▼ 檢視 ACI 屬性值

ACI 會儲存爲項目的一或多個 `aci` 屬性值。`aci` 屬性是一種可由目錄使用者讀取及修改的多值作業屬性。因此，ACI 屬性本身應由 ACI 予以保護。通常會授予管理員使用者完整的 `aci` 屬性存取權。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **透過執行下列 `ldapsearch` 指令，以檢視項目的 ACI 屬性值：**

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b entryDN -s base "(objectclass=*)" aci
```

其結果爲可複製到新的 LDIF ACI 定義以進行編輯的 LDIF 文字。因爲 ACI 的值爲長字串，因此 `ldapsearch` 作業的輸出可能會以數行顯示。此外，第一個空格是連續記號。若不讓 LDIF 輸出包含連續記號，請使用 `-T` 選項。複製並貼上 LDIF 輸出時，請將輸出格式納入考量。

備註 – 若要檢視 `aci` 值所授予及拒絕的權限，請參閱第 152 頁的 「檢視有效權限」。

▼ 檢視根層級的 ACI

當您建立尾碼時，有些預設 ACI 會建立於頂層或根層級上。這些 ACI 可讓預設管理使用者 `cn=admin,cn=Administrators,cn=config` 擁有與「目錄管理員」相同的目錄資料存取權。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **檢視預設根層級 ACI。**

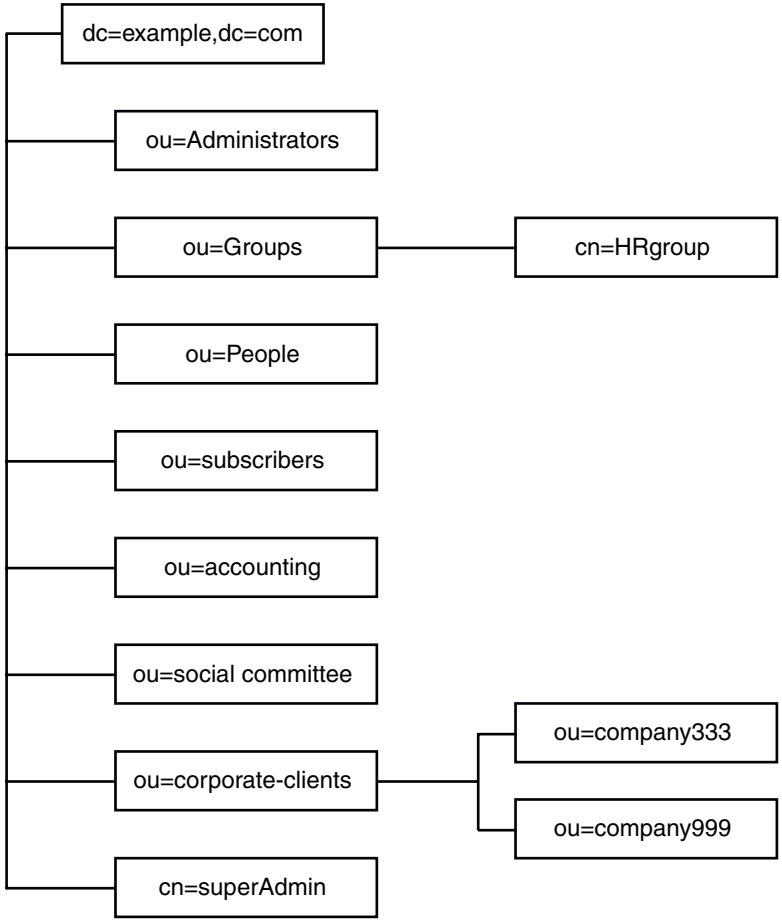
```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b "" -s base "(objectclass=*)" aci
```

存取控制用法範例

本節中的範例說明虛構的 ISP 公司 Example.com 將如何實作其存取控制策略。

此外，您也可以在安裝程式所附的範例 LDIF 檔案中找到 ACI 範例 *install_path*`/ds6/ldif/Example.ldif`。

這些範例均可說明如何透過使用 LDIF 檔案執行指定的作業。下圖將透過圖形說明 example.com 的目錄資訊樹狀結構。



Example.com 提供網站託管服務與網際網路存取。Example.com 其中一部分的網站託管服務，是託管客戶公司的目錄。Example.com 實際上託管並局部管理兩家中型公司 Company333 與 Company999 的目錄。Example.com 也爲許多個別訂閱者提供網際網路存取。

Example.com 想制定下列存取規則：

- 授予 Example.com 員工對整個 Example.com 樹狀結構的匿名讀取、搜尋與比較存取權。請參閱第 142 頁的 「授予匿名存取權」。
- 授予 Example.com 員工對個人資訊 (如 `homeTelephoneNumber` 與 `homeAddress`) 的存取權。請參閱第 143 頁的 「授予個人項目的寫入存取權」。
- 授予 Example.com 訂閱者對公司連絡資訊的 `dc=example,dc=com` 項目的讀取權限，但無法讀取該項目下的任何項目。請參閱第 144 頁的 「授予特定層級的存取權」。
- 授予 Example.com 員工在其項目中增加任何角色的權限，但某些重要角色除外。請參閱第 145 頁的 「限制主要角色的存取」。
- 授予特定管理員對尾碼擁有與「目錄管理員」相同的權限。請參閱第 145 頁的 「授予某角色對整個尾碼的完整存取權」。
- 授予 Example.com「人力資源」群組對「人事」分支中多個項目的所有權限。請參閱第 146 頁的 「授予某群組對尾碼的完整存取權」。
- 授予所有 Example.com 員工在目錄的「社委會」分支下建立群組項目，以及刪除某員工擁有之群組項目的權限。請參閱第 147 頁的 「授予新增及刪除群組項目的權限」。
- 授予所有 Example.com 員工在目錄的「社委會」分支下，將本身增加到群組項目中的權限。請參閱第 148 頁的 「允許使用者在群組中加入或移除本身」。
- 授予 Company333 與 Company999 的目錄伺服器管理員 (角色) 在特定條件下，對其個別目錄樹狀結構分支的存取權。這些條件包括 SSL 認證、時間與日期限制，以及指定位置。請參閱第 148 頁的 「授予對群組或角色的條件式存取權」。
- 授予個別訂閱者對其本身項目的存取權。請參閱第 143 頁的 「授予個人項目的寫入存取權」。
- 拒絕個別訂閱者對其本身項目內之帳單資訊的存取權。請參閱第 149 頁的 「拒絕存取」。
- 全面授予對個別訂閱者子樹狀結構的匿名存取權，但特別要求不列出的訂閱者除外。若有必要，此部分的目錄可爲防火牆外的唯讀伺服器，並每天更新一次。請參閱第 142 頁的 「授予匿名存取權」與第 151 頁的 「使用篩選設定目標」。

授予匿名存取權

大部分的目錄依配置均可讓您匿名存取至少一個尾碼以進行讀取、搜尋或比較。若您執行公司人員目錄，例如要供員工搜尋的電話簿，也可以設定這些權限。Example.com 內部即爲此情況，如 第 143 頁的 「ACI “Anonymous Example.com”」 中所說明。

做爲 ISP，Example.com 也想要建立可供全世界存取的公用電話簿，以通告其所有訂閱者的連絡資訊。相關說明請參閱第 143 頁的「ACI “Anonymous World”」。

ACI “Anonymous Example.com”

在 LDIF 中，若要將整個 Example.com 樹狀結構的讀取、搜尋與比較權限授予 Example.com 員工，請撰寫下列陳述式：

```
aci: (targetattr !="userPassword")(version 3.0; acl "Anonymous
 example"; allow (read, search, compare)
 userdn= "ldap:///anyone") ;)
```

此範例假設 `aci` 已增加至 `dc=example,dc=com entry`。請注意，`userPassword` 屬性排除於 ACI 的範圍之外。

備註 – 使用與保護密碼屬性的範例中所用相同的語法 `(targetattr !="`*attribute-name*`")`，來保護機密屬性及不應顯示的屬性。

ACI “Anonymous World”

在 LDIF 中，若要全面授予個別訂閱者子樹狀結構的讀取與搜尋存取權，但拒絕任何人存取特別要求不列出的訂閱者資訊，您可以撰寫下列陳述式：

```
aci: (targetfilter= "(!(unlistedSubscriber=yes))")
 (targetattr="homePostalAddress || homePhone || mail")
 (version 3.0; acl "Anonymous World"; allow (read, search)
 userdn="ldap:///anyone";)
```

此範例假設 `ou=subscribers,dc=example, dc=com` 項目中已加入 ACI。此範例亦假設每個訂閱者項目均具有設爲 `yes` 或 `no` 的 `unlistedSubscriber` 屬性。目標定義會根據此屬性的值篩選不列出的訂閱者。如需篩選定義的詳細資訊，請參閱第 151 頁的「使用篩選設定目標」。

授予個人項目的寫入存取權

有許多目錄伺服器管理員雖允許內部使用者變更其本身項目中的某些屬性，但不是所有屬性皆可變更。Example.com 的目錄伺服器管理員允許使用者變更其本身的密碼、家用電話號碼與住家地址，但其他項目則一概不准變更。相關說明請參閱第 144 頁的「ACI “Write Example.com”」。

若訂閱者對目錄建立了 SSL 連線，則 Example.com 的策略亦將允許其訂閱者在 Example.com 樹狀結構中更新本身的個人資訊。相關說明請參閱第 144 頁的「ACI “Write Subscribers”」。

ACI “Write Example.com”

備註 – 設定此權限後，即會授予使用者刪除屬性值的權限。

在 LDIF 中，若要授予 Example.com 員工更新其家用電話號碼與住家地址的權限，請撰寫下列陳述式：

```
aci: (targetattr="homePhone ||
 homePostalAddress")(version 3.0; acl "Write Example.com";
 allow (write) userdn="ldap:///self" ;)
```

此範例假設 `ou=People,dc=example,dc=com` 項目中已加入 ACI。

ACI “Write Subscribers”

備註 – 設定此權限後，即會授予使用者刪除屬性值的權限。

在 LDIF 中，若要授予 Example.com 訂閱者更新其家用電話號碼的權限，請撰寫下列陳述式：

```
aci: (targetattr="homePhone")
 (version 3.0; acl "Write Subscribers"; allow (write)
 userdn= "ldap://self" and authmethod="ssl";)
```

此範例假設 `ou=subscribers,dc=example, dc=com` 項目中已加入 `aci`，且使用者必須使用 SSL 進行連結。

請注意，Example.com 訂閱者並無其住家地址的寫入存取權，因爲他們可能會刪除該屬性。住家地址是 Example.com 寄發帳單時的重要業務資訊。

授予特定層級的存取權

您可以設定 ACI 範圍來影響目錄樹狀結構內不同的層級，以微調所要允許的存取層級。目標 ACI 範圍可設爲下列其中一項：

`base` 項目本身

`onelevel` 項目本身與低一層的所有項目

`subtree` 項目本身及該項目下不限深度的所有項目

ACI “Read Example.com only”

在 LDIF 中，若要授予 Example.com 訂閱者對公司連絡資訊項目 `dc=example,dc=com` 的讀取權限，但不允許存取該項目下的任何項目，請撰寫下列陳述式：

```
aci: (targetscope="base") (targetattr="*")(version 3.0;
 acl "Read Example.com only";  allow (read,search,compare)
 userdn="ldap:///cn=*,ou=subscribers,dc=example,dc=com";)
```

此範例假設 dc=example, dc=com 項目中已加入 ACI。

限制主要角色的存取

您可以使用目錄中的角色定義來識別您的企業所不可或缺的功能，例如網路與目錄的管理。

例如，您可以在全球各地的公司網站上，找出於每週特定工作日之特定時間提供服務之系統管理員的子集，以建立 superAdmin 角色。或者，您也可以在特定網站上建立 First Aid 角色，內含所有受過急救訓練的工作成員。如需有關建立角色定義的資訊，請參閱第 203 頁的 「管理角色」。

當某個角色在重要的公司或企業功能方面提供了任何種類的專用使用者權限時，建議您限制對於該角色的存取。以 Example.com 爲例，員工可在其本身的項目中加入任何角色，但 superAdmin 角色除外，如下列範例所說明。

ACI “Roles”

在 LDIF 中，若要授予 Example.com 員工在其本身的項目中加入 superAdmin 以外之任何角色的權限，請撰寫下列陳述式：

```
aci: (targetattr="*") (targattrfilters="add=nsRoleDN:
 (nsRoleDN !="cn=superAdmin, dc=example, dc=com")")
 (version 3.0; acl "Roles"; allow (write)
 userdn= "ldap:///self" ;)
```

此範例假設 ou=People,dc=example, dc=com 項目中已加入 ACI。

授予某角色對整個尾碼的完整存取權

有時爲特定使用者針對尾碼授予與「目錄管理員」相同的權限，確實有其效用。在 Example.com 中，Kirsten Vaughan 是目錄伺服器的管理員。她具有 superAdmin 的角色。此角色具有下列優勢：

- 因爲他們這一類的管理員連結會強制使用強度較高的認證 (如 SSL)，而較爲安全
- 因爲「目錄管理員」密碼較少人知道，而較爲安全
- 可透過記錄提高追溯性

備註 – 將 Kirsten Vaughan 加入 `cn=Administrators,cn=config` 群組，亦會授予其與「目錄管理員」相同的權限。

若要讓使用者對整部伺服器獲得與「目錄管理員」相同的權限，請遵循第 69 頁的 「建立具有超級使用者存取權的管理使用者」中的程序。

ACI “Full Access”

在 LDIF 中，若要授予管理員 Kirsten Vaughan 與「目錄管理員」相同的權限，請使用下列陳述式：

```
aci: (targetattr="*") (version 3.0; acl "Full Access";
 allow (all) groupdn= "ldap:///cn=SuperAdmin,dc=example,dc=com"
 and authmethod="ssl" ;)
```

此範例假設根項目 "" (無文字) 中已加入 ACI。

授予某群組對尾碼的完整存取權

大部分的目錄都具有用以識別特定公司功能的群組。可賦予群組所有或部分目錄的存取權。對整個群組套用存取權限，即不需個別設定每個成員的存取權限。而是只要將使用者加入群組中，即可授予其存取權限。

例如，當您建立目錄伺服器實例時，依預設會建立對目錄具有完整存取權的「管理員」群組 `cn=Administrators,cn=config`。

在 Example.com 中，「人力資源」群組對目錄的 `ou=People` 分支具有完整存取權，因此可更新員工目錄，如第 146 頁的 「ACI “HR”」中所說明。

ACI “HR”

在 LDIF 中，若要授予 HR 群組對目錄之員工分支的所有權限，請使用下列陳述式：

```
aci: (targetattr="*") (version 3.0; acl "HR"; allow (all)
  groupdn= "ldap:///cn=HRgroup,ou=Groups,dc=example,dc=com";)
```

此範例假設下列項目中已加入 ACI：

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

授予新增及刪除群組項目的權限

有些組織會允許員工在樹狀結構中建立項目，以提升其工作效率，並鼓勵他們爲公司的活力注入一己之力。以 Example.com 爲例，社委會即是由各種不同性質的社團所組成，如網球、游泳、滑雪與角色扮演等。

Example.com 的任何員工皆可建立代表新社團的群組項目，如 第 147 頁的 「ACI “Create Group”」 中所說明。

凡屬 Example.com 員工，即可成爲這些群組之一的成員，如第 148 頁的 「允許使用者在群組中加入或移除本身」中所說明。

只有群組所有者可修改或刪除群組項目，如 第 147 頁的 「ACI “Delete Group”」 中所說明。

ACI “Create Group”

在 LDIF 中，若要授予 Example.com 員工在 `ou=Social Committee` 分支下建立群組項目的權限，請撰寫下列陳述式：

```
aci: (targetattr="*") (targattrfilters="add=objectClass:
(|(objectClass=groupOfNames)(objectClass=top))")
(version 3.0; acl "Create Group"; allow (read,search,add)
userdn= "ldap:///uid=*,ou=People,dc=example,dc=com")
and dns="*.Example.com";)
```

此範例假設 `ou=Social Committee,dc=example,dc=com` 項目中已加入 ACI。

備註 –

- 此 ACI 並未授予寫入權限，這表示該項目的建立者無法修改項目。
- 由於伺服器會以隱藏方式加入 `top` 值，因此您必須在 `targattrfilters` 關鍵字中指定 `objectClass=top`。
- 此 ACI 會將用戶端機器限定在 `example.com` 網域中。

ACI “Delete Group”

在 LDIF 中，若要授予 Example.com 員工對他們在 `ou=Social Committee` 分支下所屬群組的群組項目進行修改或刪除的權限，請撰寫下列陳述式：

```
aci: (targetattr = "*") (targattrfilters="del=objectClass:
(objectClass=groupOfNames)")
 (version 3.0; acl "Delete Group"; allow (write,delete)
 userattr="owner#GROUPDN";)
```

此範例假設 ou=Social Committee,dc=example,dc=com 項目中已加入 aci。

請注意，使用 DSCC 建立此 ACI 是缺乏效率的方式，因爲您必須用手動編輯模式以建立目標篩選器，並檢查群組所有權。

允許使用者在群組中加入或移除本身

許多目錄都會設定 ACI 允許使用者在群組 (如郵件收信人清單) 中加入或移除本身。

在 Example.com 中，員工可將其本身加入 ou=Social Committee 子樹狀結構下的任何群組項目中，如第 148 頁的「ACI “Group Members”」中所說明。

ACI “Group Members”

在 LDIF 中，若要授予 Example.com 員工將其本身加入群組中的權限，請撰寫下列陳述式：

```
aci: (targettattr="member")(version 3.0; acl "Group Members";
 allow (selfwrite)
 (userdn= "ldap:///uid=*,ou=People,dc=example,dc=com") ;)
```

此範例假設 ou=Social Committee, dc=example,dc=com 項目中已加入 ACI。

授予對群組或角色的條件式存取權

當您授予目錄的群組或角色專用存取權時，經常必須確定入侵者無法模擬您的專用使用者使用這些權限。因此，授予群組或角色的重要存取權時所依據的存取控制規則，通常涉及許多條件。

以 Example.com 爲例，它爲託管的兩家公司 Company333 與 Company999 皆建立了「目錄伺服器管理員」角色。Example.com 希望這兩家公司能夠自行管理資料並實作存取控制規則，同時又能保護資料不受入侵。

因此，Company333 與 Company999 將在符合下列條件的情況下，對其各自位於目錄樹狀結構下的分支具有完整權限：

- 連線將使用透過 SSL 的憑證進行認證。
- 請求存取的時間爲星期一至星期四的 8:00 至 18:00。
- 兩家公司會透過各自的指定 IP 位址請求存取。

上述條件會列在兩家公司的 ACI 中，即 ACI “Company333” 與 ACI “Company999”。由於這兩個 ACI 的內容完全相同，下列範例將僅使用 “Company333” ACI 進行說明。

ACI “Company333”

在 LDIF 中，若要以前述條件爲限，授予 Company333 對自己目錄分支的完整存取權，請撰寫下列陳述式：

```
aci: (targetattr = "*") (version 3.0; acl "Company333"; allow (all)
 (roledn="ldap:///cn=DirectoryAdmin,ou=Company333,
 ou=corporate clients,dc=example,dc=com") and (authmethod="ssl")
 and (dayofweek="Mon,Tues,Wed,Thu") and (timeofday >= "0800" and
 timeofday <= "1800") and (ip="255.255.123.234"); )
```

此範例假設 `ou=Company333,ou=corporate clients,dc=example,dc=com` 項目中已加入 ACI。

拒絕存取

若您已允許尾碼大部分的存取，您可以在現有的 ACI 下拒絕少部分尾碼的存取。

備註 – 您應儘可能避免拒絕存取，因爲這樣可能會導致意外或複雜的存取控制運作方式。若要限制存取，請搭配使用範圍設定、屬性清單與目標篩選器等方法。

此外，刪除拒絕存取 ACI 並不會移除權限，而會擴充其他 ACI 所設定的權限。

目錄伺服器在評估存取權限時，會先讀取 `deny` 權限，再讀取 `allow` 權限。

在後續的範例中，Example.com 希望讓所有訂閱者都能夠讀取其本身項目下的帳單資訊，如連線時間或帳戶餘額等。Example.com 亦明確希望拒絕該資訊的寫入存取。read 存取的說明位於 第 149 頁的 「ACI “Billing Info Read”」 中。`deny` 存取的說明位於 第 150 頁的 「ACI “Billing Info Deny”」 中。

ACI “Billing Info Read”

在 LDIF 中，若要授予訂閱者讀取其本身項目中之帳單資訊的權限，請撰寫下列陳述式：

```
aci: (targetattr="connectionTime || accountBalance")
 (version 3.0; acl "Billing Info Read"; allow (search,read)
  userdn="ldap:///self";)
```

此範例假設模式中已建立相關屬性，且 `ou=subscribers,dc=example,dc=com` 項目中已加入 ACI。

ACI “Billing Info Deny”

在 LDIF 中，若要拒絕訂閱者修改其本身項目中之帳單資訊的權限，請撰寫下列陳述式：

```
aci: (targetattr="connectionTime || accountBalance")
 (version 3.0; acl "Billing Info Deny";
 deny (write) userdn="ldap:///self";)
```

此範例假設模式中已建立相關屬性，且 `ou=subscribers,dc=example,dc=com` 項目中已加入 ACI。

代理授權

代理授權方法是特殊的認證方式。使用本身的身份識別連結至目錄的使用者，會透過代理授權而授予其他使用者的權限。

若要配置目錄伺服器以允許代理伺服器請求，您必須執行下列動作：

- 授予管理員與其他使用者相同的代理權限。
- 授予一般使用者如存取控制策略所定義的一般存取權限。

備註 – 您可以將代理權限授予目錄的任何使用者，但「目錄管理員」除外。此外，您無法以「目錄管理員」的 DN 做爲代理 DN。在授予代理權限時，您必須極爲謹慎，因爲您會授予可將任何 DN (「目錄管理員」DN 除外) 指定爲代理 DN 的權限。若目錄伺服器在相同的作業中收到多個代理認證控制，即會有錯誤傳回用戶端應用程式，而作業嘗試將會失敗。

代理授權範例

Example.com 想讓連結爲 `MoneyWizAcctSoftware` 的用戶端應用程式對 LDAP 資料具有與「會計管理員」相同的存取權限。

所套用的參數如下：

- 用戶端應用程式的連結 DN 爲 `uid=MoneyWizAcctSoftware, ou=Applications,dc=example,dc=com`。
- 用戶端應用程式請求存取的目標子樹狀結構爲 `ou=Accounting,dc=example,dc=com`。
- 目錄中含有對 `ou=Accounting,dc=example,dc=com` 子樹狀結構具有存取權限的「會計管理員」。

若要使用與「會計管理員」相同的存取權限，讓用戶端應用程式取得「會計」子樹狀結構的存取權，下列必須爲眞：

- 「會計管理員」必須具有 ou=Accounting,dc=example,dc=com 子樹狀結構的存取權限。例如，下列 ACI 可授予「會計管理員」項目的所有權限：

```
aci: (targetattr="*") (version 3.0; acl "allowAll-AcctAdmin"; allow
 (all) userdn="ldap:///uid=AcctAdministrator,ou=Administrators,
 dc=example,dc=com";)
```

- 目錄中必須含有下列將代理權限授予用戶端應用程式的 ACI：

```
aci: (targetattr="*") (version 3.0; acl "allowproxy- accountingsoftware";
 allow (proxy) userdn= "ldap:///uid=MoneyWizAcctSoftware,ou=Applications,
dc=example,dc=com";)
```

正確使用此 ACI 後，MoneyWizAcctSoftware 用戶端應用程式即可連結至目錄，並傳送 ldapsearch 或 ldapmodify 之類的 LDAP 指令，而要求代理 DN 的存取權限。

在此範例中，若用戶端要執行 ldapsearch 指令，指令中將會納入下列控制：

```
$ ldapsearch -D "uid=MoneyWizAcctSoftware,ou=Applications,dc=example,dc=com" -w - \
 -Y "dn: uid=AcctAdministrator,ou=Administrators,dc=example,dc=com" ...
```

若用戶端要執行 ldapmodify 指令，指令中將會納入下列控制：

```
$ ldapmodify -h hostname -p port \
-D "uid=MoneyWizAcctSoftware,ou=Applications,dc=example,dc=com" -w - \
-Y"dn: uid=AcctAdministrator,ou=Administrators,dc=example,dc=com"
dn: uid=AcctAdministrator,ou=Administrators,dc=example,dc=com
changetype: modify
delete: userpassword
-
add: userpassword
userpassword: admin1
```

請注意，用戶端會以本身的形式連結，但仍會被授予代理伺服器項目的權限。用戶端不需具備代理伺服器項目密碼。

使用篩選設定目標

若要設定允許存取目錄中眾多項目的存取控制，您可以使用篩選器設定目標。

在 LDIF 中，若要使用篩選器讓 HR 中所有的使用者均可存取員工項目，請撰寫下列陳述式：

```
aci: (targetattr="*") (targetfilter=(objectClass=employee))
 (version 3.0; acl "HR access to employees";
 allow (all) groupdn= "ldap:///cn=HRgroup,ou=People,dc=example,dc=com";)
```

此範例假設 ou=People,dc=example,dc=com 項目中已加入 ACI。

備註 – 由於搜尋篩選不會直接爲您管理存取的物件命名，因此在允許或拒絕物件的存取權時，請勿搞錯對象。若不愼允許或拒絕了錯誤物件的存取權，您的目錄將變得更爲複雜，因而提高風險。此外，使用篩選也可能使您難以對目錄內的存取控制問題進行疑難排解。

為含有逗號的 DN 定義權限

包含逗號的 DN 需要在 LDIF ACI 陳述式中進行特殊處理。在 ACI 陳述式的目標與連結規則部分中，逗號之前必須加上一個反斜線 (\)。此語法如下列範例所說明：

```
dn: o=Example.com Bolivia\, S.A.
objectClass: top
objectClass: organization
aci: (target="ldap:///o=Example.com Bolivia\,S.A.") (targetattr="*")
(version 3.0; acl "aci 2"; allow (all) groupdn =
"ldap:///cn=Directory Administrators, o=Example.com Bolivia\, S.A.";)
```

檢視有效權限

維護目錄中某些項目的存取策略時，您必須瞭解所定義的 ACI 在安全性方面會受到哪些影響。目錄伺服器可讓您檢視 ACI 授予指定項目之指定使用者的有效權限，以評估現有的 ACI。

目錄伺服器會對「取得有效權限」控制加以回應，此控制可納入搜尋作業中。此控制的回應會傳回搜尋結果中有關項目與屬性的有效權限資訊。這項額外資訊含有每一個項目以及各項目中每一個屬性的讀取與寫入權限。您可以針對用於搜尋的連結 DN 或任意 DN 請求這些權限。此選擇可讓管理員測試目錄使用者的權限。

有效權限功能需依賴 LDAP 控制。您必須確定用以連結至遠端伺服器的代理伺服器身份識別亦可存取有效權限屬性。

限制取得有效權限控制的存取

檢視有效權限的作業是一項必須受到保護與適當限制的目錄作業。

若要限制有效權限資訊的存取，請修改 getEffectiveRights 屬性的預設 ACI。接著請建立 getEffectiveRightsInfo 屬性的新 ACI。

例如，下列 ACI 將僅允許「目錄伺服器管理員群組」的成員取得有效權限：

```
aci: (targetattr != "aci")(version 3.0; acl
 "getEffectiveRights"; allow(all) groupdn =
 "ldap:///cn=Directory Administrators,ou=Groups,dc=example,dc=com";)
```

若要取得有效權限資訊，您必須具備使用「有效權限」控制的存取控制權限，**以及** aclRights 屬性的讀取存取權。這種雙層的存取控制，可提供能夠在必要時進一步微調的基本安全性。與代理伺服器相似，若您對某項目中的 aclRights 屬性具有讀取存取權，即可請求任何人對該項目及其屬性之權限的相關資訊。這表示，管理資源的使用者能夠決定擁有資源權限的人員，即使這名使用者並未實際管理具有這些權限的人員亦然。

若請求權限資訊的使用者無權使用「有效權限」控制，作業即會失敗並傳回錯誤訊息。但若請求權限資訊的使用者有權使用控制，但缺少讀取 aclRights 屬性的權限，aclRights 屬性即不會出現在傳回的項目中。此運作方式會反映目錄伺服器的一般搜尋運作方式。

使用「取得有效權限」控制

使用 ldapsearch 指令與 -J "1.3.6.1.4.1.42.2.27.9.5.2" 選項，指定「取得有效權限」控制。依預設，控制會在搜尋結果中傳回項目與屬性的連結 DN 項目有效權限。

請使用下列選項變更預設運作方式：

- -c "dn: *bind DN* " — 搜尋結果會顯示以指定 *DN* 連結之使用者的有效權限。此選項可讓管理員檢查其他使用者的有效權限。選項 -c "dn:" 可顯示匿名認證的有效權限。
- -X "*attributeName ...* " — 搜尋結果會同時包含已命名屬性的有效權限。使用此選項可指定未出現在搜尋結果中的屬性。例如，您可以使用此選項，指定使用者是否有權增加目前不在項目中的屬性。
- 使用 -c 和/或 -X 選項時，-J 選項與「取得有效權限」控制的 OID 即已包含在內，不需另行指定。若您指定了空值的「有效權限」控制，則會擷取目前使用者的權限。此外，也會擷取目前的 ldapsearch 作業所傳回之屬性與項目的權限。

  接著，您必須選取所要檢視的資訊類型。請選擇簡單權限，或選擇可說明這些權限之授予或拒絕情形的詳細記錄資訊。在搜尋結果中增加 aclRights 或 aclRightsInfo 屬性，可決定傳回的資訊類型。您可以同時請求兩種屬性，以接收所有的有效權限資訊，但實際上簡單權限會在詳細的記錄資訊中重複這些資訊。

備註 – `aclRights` 與 `aclRightsInfo` 屬性的運作方式與虛擬作業屬性相仿。這些屬性並未儲存在目錄中，且非經明確請求不會傳回。它們會由目錄伺服器在回應「取得有效權限」控制時產生。

因此，這些屬性無法用於任何類型的篩選或搜尋作業中。

有效權限功能會繼承可影響存取控制的其他參數。這些參數包括一天中的時間、認證方法、機器位址與名稱。

下列範例將說明使用者 Carla Fuente 能夠以何種方式檢視她在目錄中的權限。在結果之中，`1` 表示已授予權限，`0` 表示已拒絕權限。

```
$ ldapsearch -J "1.3.6.1.4.1.42.2.27.9.5.2 -h host1.Example.com -p 389 \
 -D "uid=cfuente,ou=People,dc=example,dc=com" -w - -b "dc=example,dc=com" \
 "(objectclass=*)" aclRights
Enter bind password:
dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

此結果爲 Carla Fuente 顯示了她在目錄中至少具有讀取權限的項目，並顯示她可修改本身的項目。「有效權限」控制不會略過一般存取權限，因此使用者不會看見他不具讀取權限的項目。在下列範例中，「目錄管理員」將可看見 Carla Fuente 不具讀取權限的項目：

```
$ ldapsearch -h host1.Example.com -p 389 -D cn=admin,cn=Administrators,cn=config -w - \
 -c "dn: uid=cfuente,ou=People,dc=example,dc=com" -b "dc=example,dc=com" \
 "(objectclass=*)" aclRights
Enter bind password:
dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=Directory Administrators, dc=example,dc=com
```

```
aclRights;entryLevel: add:0,delete:0,read:0,write:0,proxy:0
dn: ou=Special Users,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:0,write:0,proxy:0
dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

在上方的輸出中，「目錄管理員」可看出 Carla Fuente 甚至無法檢視目錄樹狀結構中的「特殊使用者」或「目錄伺服器管理員」分支。在下列範例中，「目錄管理員」將可看出 Carla Fuente 無法修改其本身項目中的 `mail` 與 `manager` 屬性：

```
$ ldapsearch -h host1.Example.com -p 389 -D cn=admin,cn=Administrators,cn=config -w - \
 -c "dn: uid=cfuente,ou=People,dc=example,dc=com" -b "dc=example,dc=com" \
 "(uid=cfuente)" aclRights "*"
Enter bind password:
version: 1
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;attributeLevel;mail: search:1,read:1,compare:1,
 write:0,selfwrite_add:0,selfwrite_delete:0,proxy:0
mail: cfuente@Example.com
aclRights;attributeLevel;uid: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
uid: cfuente
aclRights;attributeLevel;givenName: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
givenName: Carla
aclRights;attributeLevel;sn: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
sn: Fuente
aclRights;attributeLevel;cn: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
cn: Carla Fuente
aclRights;attributeLevel;userPassword: search:0,read:0,
 compare:0,write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
userPassword: {SSHA}wnbWHIq2HPiY/5ECwe6MWBGx2KMiZ8JmjF80Ow==
aclRights;attributeLevel;manager: search:1,read:1,compare:1,
 write:0,selfwrite_add:0,selfwrite_delete:0,proxy:0
manager: uid=bjensen,ou=People,dc=example,dc=com
aclRights;attributeLevel;telephoneNumber: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
```

```
telephoneNumber: (234) 555-7898
aclRights;attributeLevel;objectClass: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

進階存取控制：使用巨集 ACI

使用重複目錄樹狀結構的組織可藉由使用巨集，以最佳化目錄中所使用的 ACI 數目。當您降低目錄樹狀結構中的 ACI 數目時，存取控制策略將更易於管理。此外，ACI 記憶體的使用效率也會獲得提昇。

巨集是指在 ACI 中用以表示 DN 或部分 DN 的預留位置。您可以使用巨集代表 ACI 之目標部分中的 DN，和/或連結規則部分中的 DN。就實務上來說，當目錄伺服器收到內送 LDAP 作業時，即會根據 LDAP 作業預定爲目標的資源比對 ACI 巨集。執行比對的目的在於找出相符的子字串 (如果有的話)。若有相符項目，連結規則方的巨集即會以相符的子字串展開，並在該展開的連結規則完成評估後判定資源的存取權。

本節含有巨集 ACI 的範例與巨集 ACI 語法的相關資訊。

巨集 ACI 範例

下列是巨集 ACI 的優點及其運作方式的最佳範例。圖 7–1 說明使用巨集 ACI 的目錄樹狀結構可有效降低整體的 ACI 數目。

在此圖例中，請注意相同的樹狀結構 (`ou=groups,ou=people`) 具有重複的子網域模式。此模式也同樣在樹狀結構中的各處重複，因爲 Example.com 目錄樹狀結構中存有兩個尾碼 `dc=hostedCompany2,dc=example,dc=com` 與 `dc=hostedCompany3,dc=example,dc=com`，並未顯示在此圖中。

目錄樹狀結構中的 ACI 也具有重複的模式。例如，下列 ACI 位於 `dc=hostedCompany1,dc=example,dc=com` 節點上：

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))(version 3.0;
 acl "Domain access"; allow (read,search) groupdn=
 "ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,
 dc=example,dc=com";)
```

此 ACI 會授予 domainAdmins 群組對 dc=hostedCompany1,dc=example,dc=com 樹狀結構中任何項目的讀取與搜尋權限。

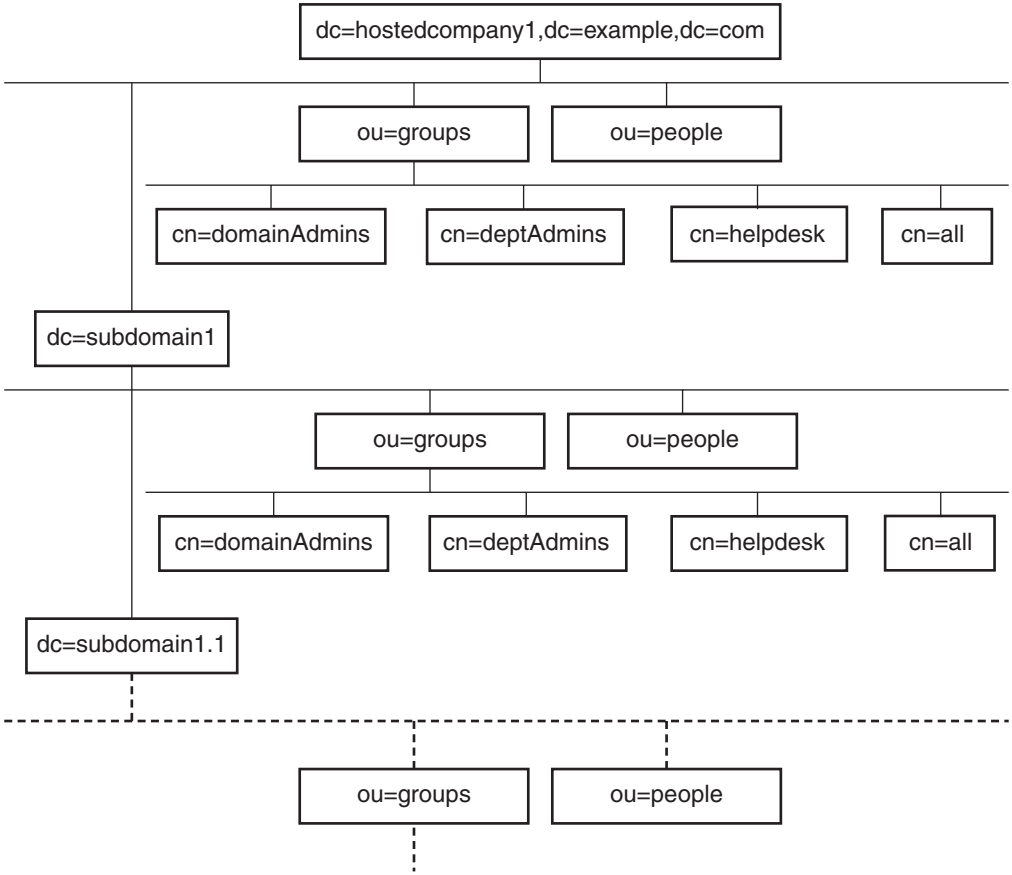


圖 7–1 巨集 ACI 的範例目錄樹狀結構

下列 ACI 位於 dc=hostedCompany1,dc=example,dc=com 節點上：

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,dc=example,dc=com";)
```

下列 ACI 位於 dc=subdomain1,dc=hostedCompany1, dc=example,dc=com 節點上：

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
```

```
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,dc=hostedCompany1,
  dc=example,dc=com";)
```

下列 ACI 位於 dc=hostedCompany2,dc=example,dc=com 節點上：

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany2, dc=example,dc=com";)
```

下列 ACI 位於 dc=subdomain1,dc=hostedCompany2, dc=example,dc=com 節點上：

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,dc=hostedCompany2,
 dc=example,dc=com";)
```

在上方的四個 ACI 中，唯一的不同之處是 groupdn 關鍵字中所指定的 DN。只要使用 DN 的巨集，即可在樹狀結構的根，亦即 dc=example,dc=com 節點上，以一個 ACI 取代這些 ACI。此巨集 ACI 如下所示：

```
aci: (target="ldap:///ou=Groups,($dn),dc=example,dc=com")
 (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

請注意，此時必須加入先前未使用的 target 關鍵字。

在前述範例中，ACI 的數目已從四個縮減爲一個。但其眞正的好處在於您在目錄樹狀結構中所減少的重複模式。

巨集 ACI 語法

爲了簡化本節中的討論，用於提供連結憑證的 ACI 關鍵字 (例如 userdn、roledn、groupdn 和 userattr) 總稱爲 ACI 的**主體**。主體可決定套用 ACI 的對象。

下表說明可取代特定 ACI 關鍵字的巨集。

表 7–1 巨集 ACI 關鍵字

巨集	說明	ACI 關鍵字
($dn)	用以在目標中進行比對，以及在主體中直接取代。	target, targetfilter, userdn, roledn, groupdn, userattr

表 7–1 巨集 ACI 關鍵字 *(續)*

巨集	說明	ACI 關鍵字
`[$dn]`	用以取代在主體的子樹狀結構中使用的多個 RDN。	`targetfilter, userdn, roledn, groupdn, userattr`
`($attr.`*attrName*`)`	用以將目標項目中的 *attributeName* 屬性值取代至主體中。	`userdn, roledn, groupdn, userattr`

巨集 ACI 關鍵字具有下列限制：

- 在主體中使用 `($dn)` 與 `[$dn]` 巨集時，您**必須**定義含有 `($dn)` 巨集的目標。
- 您可以在主體中合併 `($dn)` 巨集 (但不可合併 `[$dn]` 巨集) 與 `($attr.`*attrName*`)` 巨集。

比對目標中的 ($dn)

ACI 目標中的 `($dn)` 巨集可將本身與 LDAP 請求預定爲目標的項目進行比較，以決定取代值。例如，您的 LDAP 請求將下列項目預定爲目標：

```
cn=all,ou=groups,dc=subdomain1, dc=hostedCompany1,dc=example,dc=com
```

此外，您具有將目標定義如下的 ACI：

```
(target="ldap:///ou=Groups,($dn),dc=example,dc=com")
```

`($dn)` 巨集與 “`dc=subdomain1, dc=hostedCompany1`” 相符。因此，此子字串將做爲 ACI 主體中的取代值。

取代主體中的 ($dn)

在 ACI 的主體中，`($dn)` 巨集會由目標中相符的整個子字串所取代。例如：

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,($dn),dc=example,dc=com"
```

主體會變成：

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,
 dc=subdomain1,dc=hostedCompany1,dc=example,dc=com"
```

在巨集展開後，目錄伺服器會在完成一般程序後評估 ACI，以判定是否已授予存取權。

備註 – 使用巨集取代的 ACI 與標準 ACI 不同，它不一定會將存取權授予目標項目的子項。這是因爲，當目標爲子 DN 時，取代可能不會在主體字串中建立有效的 DN。

取代主體中的 [$dn]

[$dn] 的取代機制與 ($dn) 略有不同。目標資源的 DN 會進行數次檢查，並逐次捨棄最左邊的 RDN 元件，直到找到相符項目爲止。

例如，假設您的 LDAP 請求將 cn=all,ou=groups, dc=subdomain1,dc=hostedCompany1,dc=example,dc=com 子樹狀結構預定爲目標，並且具有下列 ACI：

```
aci: (targetattr="*")
 (target="ldap:///ou=Groups,($dn),dc=example,dc=com")
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],
 dc=example,dc=com";)
```

伺服器會以下列方式繼續作業，以展開此 ACI：

1. 伺服器驗證目標中的 ($dn) 確實符合 dc=subdomain1,dc=hostedCompany1。
2. 伺服器將主體中的 [$dn] 取代爲 dc=subdomain1,dc=hostedCompany1。

   產生的主體爲 groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=subdomain1,dc=hostedCompany1,dc=example,dc=com"。若因連結 DN 爲該群組的成員而授予存取權，即會停止巨集展開，並評估 ACI。若連結 DN 不是成員，則會繼續進行程序。
3. 伺服器將主體中的 [$dn] 取代爲 dc=hostedCompany1。

   產生的主體爲 groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=example,dc=com"。連結 DN 會再次被視爲此群組的成員進行測試，若爲其成員，則會完整評估 ACI。若連結 DN 不是成員，巨集展開則會停在最後一個具有相符值的 RDN 上，而此 ACI 的 ACI 評估即告完成。

[$dn] 巨集的好處是，它可透過彈性的方式，授予目錄樹狀結構中**所有**子網域的網域層級管理員存取權。因此，在表示不同網域間的階層式關係時，[$dn] 巨集即可派上用場。

例如，請考量下列 ACI：

```
aci: (target="ldap:///ou=*,($dn),dc=example,dc=com") (targetattr="*")
(targetfilter=(objectClass=nsManagedDomain))
(version 3.0; acl "Domain access"; allow (read,search) groupdn=
"ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

ACI 可授予 cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=example,dc=com 的成員對 dc=hostedCompany1 下所有子網域的存取權。因此，凡屬於該群組的管理員，即可存取如子樹狀結構 ou=people,dc=subdomain1.1,dc=subdomain1 的項目。

但在此同時，cn=DomainAdmins,ou=Groups, dc=subdomain1.1 的成員將遭拒絕存取 ou=people,dc=subdomain1, dc=hostedCompany1 與 ou=people,dc=hostedCompany1 節點。

($attr.*attrName*) 的巨集比對

($attr.*attrname*) 巨集一律用於 DN 的主體部分中。例如，您可以定義下列 roledn：

```
roledn = "ldap:///cn=DomainAdmins,($attr.ou),dc=HostedCompany1,dc=example,dc=com"
```

現在，假設伺服器收到將下列項目預設爲目標的 LDAP 作業：

```
dn: cn=Babs Jensen,ou=People,dc=HostedCompany1,dc=example,dc=com
cn: Babs Jensen
sn: Jensen
ou: Sales
...
```

爲評估 ACI 的 roledn 部分，伺服器會讀取目標項目中所儲存的 ou 屬性值。接著，伺服器會在主體中取代此值，以展開巨集。在此範例中，roledn 會展開如下：

```
roledn = "ldap:///cn=DomainAdmins,ou=Sales,dc=HostedCompany1,dc=example,dc=com"
```

然後，目錄伺服器會根據一般 ACI 評估演算法進行 ACI 的評估。

若巨集中的已命名屬性爲多值屬性，則會依序使用每個值展開巨集。會採用第一個產生成功比對的值。

記錄存取控制資訊

若要在錯誤記錄中取得存取控制的相關資訊，您必須設定適當的記錄層級。

▼ 設定 ACI 記錄

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **設定記錄層級，以將 ACI 處理納入考量。**

```
$ dsconf set-log-prop -h host -p port error level:err-acl
```

使用 TCP 包裝的用戶端主機存取控制

您可以使用 TCP 包裝程式，控制連線在 TCP 上所接受或遭拒絕的主機或 IP 位址。您可以透過 TCP 包裝，限制用戶端主機的存取。如此一來，您即可對目錄伺服器的初始 TCP 連線使用非主機式保護。

雖然您可以爲目錄伺服器設定 TCP 包裝，但如此可能會使效能受到嚴重影響，特別是在阻絕服務攻擊期間。使用在目錄伺服器以外維護的主機式防火牆，或使用 IP 連接埠篩選，將可達到最佳效能。

▼ 啓用 TCP 包裝

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 **在實例路徑內，建立 `hosts.allow` 檔案或 `hosts.deny` 檔案。**

例如，您可以在 *instance-path*`/config` 中建立檔案。請確定您所建立的檔案格式符合 `hosts_access`(4)。

2 **設定存取檔案的路徑。**

```
$ dsconf set-server-prop -h host -p port host-access-dir-path:path-to-file
```

例如：

```
$ dsconf set-server-prop -h host -p port host-access-dir-path:/local/ds1/config
"host-access-dir-path" property has been set to "/local/ds1/config".
The "/local/ds1/config" directory on host1 must contain valid hosts.allow
and/or hosts.deny files.
Directory Server must be restarted for changes to take effect.
```

▼ 停用 TCP 包裝

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **將主機存取路徑設為 ""。**

```
$ dsconf set-server-prop -h host -p port host-access-dir-path:""
```

◆ ◆ ◆ 第 8 章

8 目錄伺服器密碼策略

當使用者連線至目錄伺服器時，會認證該使用者。目錄會根據認證期間建立的身份識別，授予使用者存取權限與資源限制。本章中的**帳號**大致表示使用者項目。帳號也反映該使用者可在目錄上執行作業的權限。在此密碼策略的說明中，每個帳號都會與使用者項目以及密碼相關。

本章也會說明密碼策略的帳號啓用。目錄伺服器管理員無須遵守密碼策略，可直接對帳號進行鎖定與解除鎖定。

本章不包含認證方法。部分認證方法 (例如 SASL GSSAPI 與用戶端 SSL 憑證型認證) 不需要使用密碼。本章中關於密碼策略的資訊不適用於這類認證方法。如需配置認證機制的相關指示，請參閱第 6 章。

本章不涵蓋 Directory Server 6.3 與目錄伺服器先前版本之間的密碼策略相容性問題。當您建立 Directory Server 6.3 實例時，密碼策略實作預設爲 Directory Server 5 相容模式，以從早期版本進行升級。若要完整使用本章所述的密碼策略功能，您需要變更密碼策略相容性模式。如需有關設定密碼相容性模式的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Migration Guide」中的「Password Policy Compatibility」。

本章包含下列主題：

- 第 164 頁的「密碼策略與工作表」
- 第 168 頁的「管理預設密碼策略」
- 第 171 頁的「管理專用密碼策略」
- 第 179 頁的「在 pwdSafeModify 爲 TRUE 時從指令行修改密碼」
- 第 180 頁的「重設過期的密碼」
- 第 182 頁的「設定帳號特性」
- 第 184 頁的「手動鎖定帳號」

密碼策略與工作表

本節說明密碼策略設定，並提供工作表協助您定義符合您需求的密碼策略。

備註 – 若要使用預設密碼策略，請參閱第 168 頁的 「管理預設密碼策略」。

密碼策略設定

在目錄伺服器中指定密碼策略時，會修改或建立包含物件類別 pwdPolicy(5dsoc) 的項目。

定義特定使用者類型的密碼策略時，需要考量下列事項：

- 如何在出現入侵者嘗試破解密碼時封鎖帳號。
  如需詳細資訊，請參閱第 164 頁的 「帳號封鎖策略」。
- 如何變更密碼。
  如需詳細資訊，請參閱第 165 頁的 「密碼變更的策略」。
- 允許的密碼值。
  如需詳細資訊，請參閱第 165 頁的 「密碼內容的策略」。
- 如何處理密碼過期。
  如需詳細資訊，請參閱第 166 頁的 「密碼過期的策略」。
- 伺服器是否記錄上次成功認證的時間。
  請參閱第 167 頁的 「追蹤上次認證時間的策略」。

本章後續幾節說明如何處理密碼策略的這幾點。使用第 167 頁的 「定義密碼策略的工作表」釐清規劃要執行的各項密碼策略。

帳號封鎖策略

本節說明決定帳號封鎖的策略屬性。

目錄伺服器帳號大致表示使用者項目，以及該使用者在目錄上執行作業所具有的權限。每個帳號會與連結 DN 及使用者密碼相關。當出現入侵者嘗試破解密碼時，希望目錄伺服器會鎖定帳號。鎖定功能可避免入侵者使用帳號進行連結，也能避免入侵者繼續攻擊。

身爲管理員，您也可以手動停用共用一個角色的所有使用者之帳號。如需相關指示，請參閱第 184 頁的 「手動鎖定帳號」。此外，密碼策略的主要部分必須在**無人**介入且目錄伺服器鎖定帳號的情況下指定。

首先，您必須指定目錄伺服器可以使用 pwdLockout(5dsat) 在發生太多失敗連結時，自動鎖定帳號。目錄伺服器會記錄嘗試連結帳號的連續失敗次數。您可以使用 pwdMaxFailure(5dsat) 指定在目錄伺服器鎖定帳號之前可允許的連續失敗次數。

目錄伺服器會嚴守密碼策略鎖定帳號。此純粹爲機械化作業。帳號鎖定的原因可能不是因爲入侵者對帳號發動攻擊，而是因爲使用者鍵入的密碼不正確。因此，您可以使用 pwdFailureCountInterval(5dsat) 指定目錄伺服器清除失敗嘗試的記錄之前，等待下一次嘗試的時間。您可以使用 pwdLockoutDuration(5dsat) 指定目錄伺服器自動解除鎖定帳號之前，封鎖應持續的時間。如果使用者的錯誤有正當理由而不是出於惡意，管理員不須介入解除鎖定使用者帳號。

如果您的使用者資料已在複寫拓樸之間複寫，封鎖屬性會隨其他項目資料一併複寫。pwdIsLockoutPrioritized(5dsat) 屬性的預設設定爲 TRUE，因此會以較高的優先權複寫封鎖屬性的更新。因而會限制使用者在連續嘗試連結至任何單一複本 pwdMaxFailure 次之後，便遭到封鎖，其他多個複本在遭到封鎖之前的嘗試次數可能更少。如需有關如何確定使用者在整個複寫拓樸中遭到封鎖前，剛好有 pwdMaxFailure 次嘗試次數的詳細資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中的「Preventing Authentication by Using Global Account Lockout」。

密碼變更的策略

本節說明決定密碼變更的策略屬性。

目錄伺服器在許多部署中是身份識別資料的儲存庫。使用者應該能夠如 pwdAllowUserChange(5dsat) 所指定變更密碼，因此您無須變更密碼。

允許使用者變更密碼之後，可能也要控制使用者可以變更密碼的情況。您可以使用 pwdSafeModify(5dsat) 指定變更密碼的使用者必須先提供正確的現有密碼，才能取代密碼。如需如何修改密碼的範例，請參閱第 179 頁的 「在 pwdSafeModify 爲 TRUE 時從指令行修改密碼」。您可以使用 pwdInHistory(5dsat) 指定目錄伺服器會記住的密碼數目，避免使用者重複使用相同的密碼。您也可以設定 pwdMinAge(5dsat) 以避免使用者變更密碼過於頻繁。

在許多情況下，可利用管理員身份或是由某些您所管理的應用程式，在目錄中建立使用者項目。您可以指定使用者密碼值在使用者第一次連結至新帳號時進行變更。您也可能需要重設使用者密碼，重設之後，使用者應在下次使用帳號時變更密碼。目錄伺服器具有特定屬性 pwdMustChange(5dsat)，您可用以表示當另一個使用者重設密碼值之後，使用者是否必須變更密碼。

您也可以指定目錄伺服器管理員在設定 passwordRootdnMayBypassModsChecks(5dsat) 以變更密碼時，不須遵守策略。

密碼內容的策略

本節說明決定密碼內容的策略屬性。

雖然密碼值一般不會在目錄搜尋中傳回，但是攻擊者可能會取得目錄資料庫的存取權。因此，密碼值一般會以使用 passwordStorageScheme(5dsat) 指定的支援雜湊格式之一儲存。

您也可以設定 pwdCheckQuality(5dsat)，強制檢查密碼是否符合最基本的密碼品質定義。伺服器接著會檢查密碼是否不符合 cn、givenName、mail、ou、sn 或 uid 屬性的任何值。密碼與任何值的比較皆區分大小寫。

其他檢查可透過設定 pwdCheckQuality(5dsat) 提供。您可以設定 pwdMinLength(5dsat) 強制密碼至少有指定的字元數。此外，啓用 Strong Password Check (密碼強度檢查) 外掛程式時，目錄伺服器會檢查密碼是否不含外掛程式所用之字典檔案的字串。伺服器也會檢查密碼是否包含不同字元類型的適當混合。

您可以使用 dsconf set-server-prop 指令啓用密碼強度檢查。使用 pwd-strong-check-enabled 特性可啓動外掛程式，並重新啓動伺服器以使變更生效。使用 pwd-strong-check-require-charset 特性可指定密碼所需的字元集。pwd-strong-check-require-charset 特性包含下列值：

lower	新密碼必須包含小寫字元。
upper	新密碼必須包含大寫字元。
digit	新密碼必須包含數字。
special	新密碼必須包含特殊字元。
any-two	新密碼至少必須從上述兩個字元集中，各包含至少一個字元。
any-three	新密碼至少必須從上述三個字元集中，各包含至少一個字元。

pwd-strong-check-require-charset 特性的預設設定爲 lower && upper && digit && special。

密碼過期的策略

本節說明決定密碼過期的策略屬性。

若要確保使用者定期變更密碼，請設定 pwdMaxAge(5dsat)，以配置目錄伺服器在密碼到達特定存在期限之後，設定密碼爲過期。

使用者必須收到密碼即將過期的通知。您可以配置目錄伺服器傳回用以連結的密碼即將過期之警告。使用 pwdExpireWarning(5dsat) 定義過期之前多久，才應在用戶端連結時傳回警告。**請注意，用戶端應用程式會收到警告。使用者不會直接收到警告。**當用戶端應用程式收到密碼即將過期的警告時，必須通知一般使用者。

您可以設定 pwdGraceAuthNLimit(5dsat)，允許使用者以過期的密碼嘗試連結一或多次。如此一來，無法及時變更密碼的使用者仍得以連結以變更密碼。請注意，當使用者使用寬限登入進行連結時，使用者可執行任何作業。寬限登入執行起來就好像密碼尚未過期一般。

目錄伺服器會在每次項目上的密碼有所修改時，更新作業屬性 pwdChangedTime(5dsat)。因此，若要啓用密碼過期，已超過期限的使用者密碼將在您啓用密碼過期時立即過期。若您覺得此運作方式不合用，請使用警告與寬限登入。

追蹤上次認證時間的策略

本節包含密碼策略屬性 pwdKeepLastAuthTime(5dsat) 的用法。

一旦設定 pwdKeepLastAuthTime，目錄伺服器就會在每次使用者認證時，追蹤上次成功連結的時間。時間會記錄在使用者項目的 pwdLastAuthTime(5dsat) 作業屬性中。

由於此運作方式會爲每個成功的連結作業增加一個更新，因此預設不會啓用 pwdKeepLastAuthTime 功能。您必須明確地啓用此功能，才能於部署中使用。

定義密碼策略的工作表

此工作表設計用以協助您定義密碼策略，以透過指令行介面或使用目錄服務控制中心 (DSCC) 執行。每個密碼策略各使用一個工作表。

記錄密碼策略項目的 DN 之後，請記錄每個策略區中關於屬性設定的決策。另請記錄使用這些設定的理由。

密碼策略工作表

密碼策略項目辨別名稱
dn: cn=

策略區	屬性	於此處寫入設定	於此處寫入使用這些設定的理由
帳號封鎖	pwdFailureCountInterval(5dsat)		
	pwdIsLockoutPrioritized(5dsat)		
	pwdLockout(5dsat)		
	pwdLockoutDuration(5dsat)		
	pwdMaxFailure(5dsat)		
密碼變更	passwordRootdnMayBypassModsChecks(5dsat)		
	pwdAllowUserChange(5dsat)		
	pwdInHistory(5dsat)		
	pwdMinAge(5dsat)		
	pwdMustChange(5dsat)		
	pwdSafeModify(5dsat)		

策略區	屬性	於此處寫入設定	於此處寫入使用這些設定的理由
密碼內容	passwordStorageScheme(5dsat)		
	pwdCheckQuality(5dsat)		
	pwdMinLength(5dsat)		
密碼過期	pwdExpireWarning(5dsat)		
	pwdGraceAuthNLimit(5dsat)		
	pwdMaxAge(5dsat)		
追蹤上次認證時間	pwdKeepLastAuthTime(5dsat)		

備註 – 當 pwdCheckQuality 屬性設爲 2 時，伺服器可以執行其他檢查。Password Check (密碼檢查) 外掛程式會同時啓用，外掛程式的設定會影響要在新密碼的值上執行之檢查作業。

管理預設密碼策略

預設密碼策略會套用到目錄實例中未定義專用策略的所有使用者。但是，預設密碼策略不會套用到目錄管理員。如需策略範圍的詳細資訊，請參閱第 171 頁的「套用的密碼策略」。

預設密碼策略是可以使用 dsconf 指令配置的策略。您也可以讀取 cn=Password Policy,cn=config 以檢視預設密碼策略。

本節顯示每個策略區的策略屬性以及相關的 dsconf 伺服器特性，亦說明如何檢視與變更預設密碼策略設定。

密碼策略屬性與 dsconf 伺服器特性之間的關聯

下表顯示每個密碼策略區的密碼策略屬性與相關的 dsconf 伺服器特性。

策略區	策略屬性	dsconf 伺服器特性
帳號封鎖	pwdFailureCountInterval	pwd-failure-count-interval
	pwdLockout	pwd-lockout-enabled
	pwdLockoutDuration	pwd-lockout-duration
	pwdMaxFailure	pwd-max-failure-count
密碼變更	passwordRootdnMayBypassModsChecks	pwd-root-dn-bypass-enabled
	pwdAllowUserChange	pwd-user-change-enabled
	pwdInHistory	pwd-max-history-count
	pwdMinAge	pwd-min-age
	pwdMustChange	pwd-must-change-enabled
	pwdSafeModify	pwd-safe-modify-enabled
密碼內容	pwdCheckQuality	pwd-check-enabled、 pwd-accept-hashed-password-enabled、 pwd-strong-check-dictionary-path、 pwd-strong-check-enabled、 pwd-strong-check-require-charset
	pwdMinLength	pwd-min-length
	passwordStorageScheme	pwd-storage-scheme
密碼過期	pwdExpireWarning	pwd-expire-warning-delay
	pwdGraceAuthNLimit	pwd-grace-login-limit
	pwdMaxAge	pwd-max-age
追蹤上次認證時間	pwdKeepLastAuthTime	pwd-keep-last-auth-time-enabled

備註 – pwdCheckQuality 的相關特性可配置 Password Check (密碼檢查) 外掛程式。因此，這五個特性適用於整個伺服器實例。這五個特性也適用於 pwdCheckQuality: 2 的其他密碼策略。

▼ 檢視預設密碼策略設定

您可以使用 dsconf 指令檢視預設密碼策略設定。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **讀取預設密碼策略配置。**

  ```
  $ dsconf get-server-prop -h host -p port -v -i \
  -w password-file | grep ^pwd-
  ```

  password-file 包含目錄管理員的密碼。

```
pwd-accept-hashed-pwd-enabled      :  N/A
pwd-check-enabled                  :  off
pwd-compat-mode                    :  DS5-compatible-mode
pwd-expire-no-warning-enabled      :  on
pwd-expire-warning-delay           :  1d
pwd-failure-count-interval         :  10m
pwd-grace-login-limit              :  disabled
pwd-keep-last-auth-time-enabled    :  off
pwd-lockout-duration               :  1h
pwd-lockout-enabled                :  off
pwd-lockout-repl-priority-enabled  :  on
pwd-max-age                        :  disabled
pwd-max-failure-count              :  3
pwd-max-history-count              :  disabled
pwd-min-age                        :  disabled
pwd-min-length                     :  6
pwd-mod-gen-length                 :  6
pwd-must-change-enabled            :  off
pwd-root-dn-bypass-enabled         :  off
pwd-safe-modify-enabled            :  off
pwd-storage-scheme                 :  SSHA
pwd-strong-check-dictionary-path   :  /local/ds6/plugins/words-english-big.txt
pwd-strong-check-enabled           :  off
pwd-strong-check-require-charset   :  lower
pwd-strong-check-require-charset   :  upper
pwd-strong-check-require-charset   :  digit
pwd-strong-check-require-charset   :  special
pwd-supported-storage-scheme       :  CRYPT
pwd-supported-storage-scheme       :  SHA
pwd-supported-storage-scheme       :  SSHA
pwd-supported-storage-scheme       :  NS-MTA-MD5
pwd-supported-storage-scheme       :  CLEAR
pwd-user-change-enabled            :  on
```

▼ 變更預設密碼策略設定

您可以使用 `dsconf` 指令設定伺服器特性，以變更預設密碼策略。

備註 – 完成此程序之前，請閱讀並完成第 167 頁的 「定義密碼策略的工作表」。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **從工作表轉譯設定至 `dsconf` 指令特性設定中。**
2. **使用 `dsconf set-server-prop` 指令適當地變更預設密碼策略特性。**
   例如，下列指令可讓目錄管理員在修改密碼時違反預設策略：

```
$ dsconf set-server-prop -h host -p port pwd-root-dn-bypass-enabled:on
```

管理專用密碼策略

專用密碼策略會定義於 `pwdPolicy`(5dsoc) 項目中。您可以在目錄樹狀結構的任何位置定義策略，通常會在以策略決定的帳號複寫之子樹狀結構中。該策略具有格式爲 `cn=`*policy name*`,`*subtree* 的 DN。

定義密碼策略之後，可以在想要的使用者項目中設定 `pwdPolicySubentry`(5dsat) 屬性以指定密碼策略。

本節包含下列主題：

- 第 171 頁的 「套用的密碼策略」
- 第 172 頁的 「建立密碼策略」
- 第 174 頁的 「指定密碼策略給個別帳號」
- 第 174 頁的 「使用角色與 CoS 指定密碼策略」
- 第 176 頁的 「設定第一次登入密碼策略」

套用的密碼策略

目錄伺服器可讓您配置多個密碼策略。本節說明預設密碼策略與專用密碼策略。本節亦會說明當多個密碼策略可套用到指定的帳號時，將強制執行的策略。

第一次建立目錄伺服器實例時，該實例會有預設密碼策略。預設密碼策略會顯示在配置項目 `cn=PasswordPolicy,cn=config` 中。預設密碼策略會套用到目錄中除目錄管理員以外的所有帳號。

在所有的目錄伺服器密碼策略中，`cn=PasswordPolicy,cn=config` 有物件類別 `pwdPolicy`(5dsoc) 與物件類別 `sunPwdPolicy`(5dsoc)。

備註 – 建立目錄伺服器實例時，密碼策略屬性會保留在 Directory Server 5 相容模式中，以從舊版本升級。在 Directory Server 5 相容模式中，目錄伺服器也會處理具有物件類別 `passwordPolicy`(5dsoc) 的密碼策略項目。

如「Sun Java System Directory Server Enterprise Edition 6.3 Migration Guide」中所述，在升級完成之後，便可以在完整功能模式中使用新的密碼策略。目錄應用程式不會感覺到管理動作的進行。

本章包含使用新密碼策略功能的密碼策略配置。

您可以變更預設密碼策略以覆寫預設設定。您可以使用 `dsconf`(1M) 指令設定預設密碼策略的伺服器特性。這類伺服器特性名稱一般會以 `pwd-` 前綴爲開頭。變更這類特性的設定時，會覆寫該實例的預設密碼策略。但是，複寫不會複製對複本所做的變更。對預設密碼策略所做的變更是實例配置的一部分，而不是目錄資料的一部分。

除了配置預設密碼策略之外，您還可以配置**專用密碼策略**。專用密碼策略會由目錄樹狀結構中的項目所定義。專用密碼策略項目會有與預設密碼策略相同的物件類別 `pwdPolicy`(5dsoc)，因此會有相同的策略屬性。由於專用密碼策略爲標準目錄項目，策略項目會以標準目錄項目相同的方式進行複寫。

使用者項目透過作業屬性 `pwdPolicySubentry`(5dsat) 的值參照專用密碼策略。使用者項目參照專用密碼策略時，該策略會覆寫實例的預設密碼策略。在許多部署中，您需要指定使用者角色。您可以設定 `pwdPolicySubentry` 值配置角色與服務類別 (CoS) 搭配使用，以決定套用到使用者帳號的密碼策略。若要覆寫某個角色所設定的密碼策略，請直接變更使用者項目上的 `pwdPolicySubentry` 值。

總結本節，一開始會套用預設密碼策略。您可以變更預設密碼策略以覆寫預設值。接著，您可以建立專用密碼策略項目以覆寫預設密碼策略。以角色與 CoS 指定密碼策略時，可以指定個別項目的密碼策略以覆寫 CoS 指定的策略。

▼ 建立密碼策略

建立與修改專用密碼策略的方式與建立及修改任何其他目錄項目的方式相同。下列程序示範如何使用文字編輯器在 LDIF 中寫入密碼策略項目。接著，您可以使用 `ldapmodify` 指令與 `-a` 選項，將密碼策略項目增加至目錄中。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

開始之前 除非另外指定，否則此處顯示的資料範例來自於 `Example.ldif`。

1 為要建立的密碼策略完成密碼策略工作表。

如需範例，請參閱第 167 頁的「定義密碼策略的工作表」。

2 在以工作表為基礎的 LDIF 中寫入密碼策略項目。

例如，下列策略項目會指定 Example.com 臨時員工的密碼策略，該員工的子樹狀結構根目錄爲 dc=example,dc=com：

```
dn: cn=TempPolicy,dc=example,dc=com
objectClass: top
objectClass: pwdPolicy
objectClass: sunPwdPolicy
objectClass: LDAPsubentry
cn: TempPolicy
pwdAttribute: userPassword
pwdCheckQuality: 2
pwdLockout: TRUE
pwdLockoutDuration: 300
pwdMaxFailure: 3
pwdMustChange: TRUE
```

除了預設密碼策略設定之外，此處顯示的策略還會指定其他運作方式，同時會強制執行密碼品質檢查。帳號會在三次連續連結失敗之後鎖定五分鐘 (300 秒)。重設密碼之後必須變更密碼。指定策略給使用者帳號之後，此處**明確**指定的設定會覆寫預設密碼策略。

3 將密碼策略項目增加至目錄中。

例如，下列指令會增加 dc=example,dc=com 下 Example.com 臨時員工的密碼策略。密碼策略已儲存在名爲 pwp.ldif 的檔案中。

```
$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f pwp.ldif
Enter bind password:
adding new entry cn=TempPolicy,dc=example,dc=com

$ ldapsearch -D uid=kvaughan,ou=people,dc=example,dc=com -w --b dc=example,dc=com \
"(&(objectclass=ldapsubentry)(cn=temppolicy))"
Enter bind password:
version: 1
dn: cn=TempPolicy,dc=example,dc=com
objectClass: top
objectClass: pwdPolicy
objectClass: LDAPsubentry
cn: TempPolicy
pwdCheckQuality: 2
pwdLockout: TRUE
pwdLockoutDuration: 300
pwdMaxFailure: 3
pwdMustChange: TRUE
$
```

如 Example.ldif 中所示，kvaughan 是人力資源部門的經理，具有修改 dc=example,dc=com 項目的存取權。Vaughan 的連結密碼如 Example.ldif 中所示，爲 bribery。

另請參閱 若要定義您所定義的策略決定之使用者帳號，請參閱第 174 頁的「指定密碼策略給個別帳號」或第 174 頁的「使用角色與 CoS 指定密碼策略」。

▼ 指定密碼策略給個別帳號

本程序會指定現有的密碼策略給單一使用者帳號。

備註 – 若要完成此程序，必須有可指定的專用密碼策略。請參閱第 172 頁的「建立密碼策略」。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

除非另外指定，否則此處顯示的資料範例來自於 `Example.ldif`。

- **將密碼策略 DN 增加至使用者項目之 `pwdPolicySubentry` 屬性的值。**

  例如，下列指令會指定在第 172 頁的「建立密碼策略」中定義的密碼策略給 DN 爲 `uid=dmiller,ou=people,dc=example,dc=com` 的 David Miller 項目：

```
$ cat pwp.ldif
dn: uid=dmiller,ou=people,dc=example,dc=com
changetype: modify
add: pwdPolicySubentry
pwdPolicySubentry: cn=TempPolicy,dc=example,dc=com

$ ldapmodify -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f pwp.ldif
Enter bind password:
modifying entry uid=dmiller,ou=people,dc=example,dc=com

$ ldapsearch -D uid=kvaughan,ou=people,dc=example,dc=com -w - -b dc=example,dc=com \
"(uid=dmiller)" pwdPolicySubentry
Enter bind password:
version: 1
dn: uid=dmiller, ou=People, dc=example,dc=com
pwdPolicySubentry: cn=TempPolicy,dc=example,dc=com
$
```

  如 `Example.ldif` 中所示，`kvaughan` 是人力資源部門的經理，具有修改 `dc=example,dc=com` 項目的存取權。Vaughan 的連結密碼如 `Example.ldif` 中所示，爲 `bribery`。

▼ 使用角色與 CoS 指定密碼策略

本程序經由套用角色與服務類別 (CoS)，指定現有的專用密碼策略給一組使用者。如需角色與 CoS 的詳細資訊，請參閱第 10 章。

備註 – 若要完成此程序，必須有可指定的專用密碼策略。請參閱第 172 頁的「建立密碼策略」。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

除非另外指定，否則此處顯示的資料範例來自於 `Example.ldif`。

1 **建立要由密碼策略所決定的項目角色。**

例如，下列指令會爲 Example.com 的臨時員工建立篩選的角色：

```
$ cat tmp.ldif
dn: cn=TempFilter,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: TempFilter
nsRoleFilter: (&(objectclass=person)(status=contractor))
description: filtered role for temporary employees

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f tmp.ldif
Enter bind password:
modifying entry cn=TempFilter,ou=people,dc=example,dc=com

$
```

如 `Example.ldif` 中所示，`kvaughan` 是人力資源部門的經理，具有修改 `dc=example,dc=com` 項目的存取權。Vaughan 的連結密碼如 `Example.ldif` 中所示，爲 `bribery`。

2 **建立服務類別以產生密碼策略項目的 DN。**

DN 爲具有您所建立的角色之使用者的 `pwdPolicySubentry` 屬性值。

例如，下列指令會爲 Example.com 的臨時員工建立篩選的角色。這些指令會爲擁有該角色的使用者指定 `cn=TempPolicy,dc=example,dc=com`。

```
$ cat cos.ldif
dn: cn=PolTempl,dc=example,dc=com
objectclass: top
objectclass: nsContainer

dn: cn="cn=TempFilter,ou=people,dc=example,dc=com",
 cn=PolTempl,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
```

```
objectclass: LDAPsubentry
objectclass: costemplate
cosPriority: 1
pwdPolicySubentry: cn=TempPolicy,dc=example,dc=com

dn: cn=PolCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDN: cn=PolTempl,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: pwdPolicySubentry operational

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f cos.ldif
Enter bind password:
modifying entry cn=TempFilter,ou=people,dc=example,dc=com

$
```

狀態爲 `contractor` 的使用者現在變成遵循密碼策略 `cn=TempPolicy,dc=example,dc=com`。

▼ 設定第一次登入密碼策略

在許多部署中，新帳號所套用的密碼策略會與已建立的帳號所套用的密碼策略不同。本節示範第一次登入密碼策略。該策略提供使用者三天試用新建立的帳號並設定新密碼，之後才會鎖定帳號。該策略執行方式對重設密碼的使用者而言也相同。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **建立新建立帳號的專用密碼策略。**

   例如，增加設定過期時間爲三天 (259,200 秒) 的密碼策略項目。此密碼策略的 `pwdMustChange`(5dsat) 也會設爲 `TRUE`，表示使用者第一次連結時必須變更密碼。

   ```
   $ cat firstLogin.ldif
   dn: cn=First Login,dc=example,dc=com
   objectClass: top
   objectClass: LDAPsubentry
   objectClass: pwdPolicy
   objectClass: sunPwdPolicy
   cn: First Login
   passwordStorageScheme: SSHA
   pwdAttribute: userPassword
   pwdInHistory: 0
   pwdExpireWarning: 86400
   pwdLockout: TRUE
   ```

```
pwdMinLength: 6
pwdMaxFailure: 3
pwdMaxAge: 259200
pwdFailureCountInterval: 600
pwdAllowUserChange: TRUE
pwdLockoutDuration: 3600
pwdMinAge: 0
pwdCheckQuality: 2
pwdMustChange: TRUE

$ ldapmodify -a -D cn=admin,cn=Administrators,cn=config -w - -f firstLogin.ldif
Enter bind password:
adding new entry cn=First Login,dc=example,dc=com

$
```

2 建立包含所有新建立帳號的角色。

建立此角色會設定一些區分新建立的帳號與已建立的帳號之方式。

a. 將新帳號定義為 pwdReset(5dsat) 屬性設為 TRUE 的帳號。

若是密碼管理員等其他使用者變更使用者密碼時，pwdReset 會設爲 TRUE。

b. 建立識別新帳號的角色。

例如，下列指令會建立已重設密碼的帳號角色。

```
$ cat newRole.ldif
dn: cn=First Login Role,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: First Login Role
nsRoleFilter: (pwdReset=TRUE)
description: Role to assign password policy for new and reset accounts

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f newRole.ldif
Enter bind password:
adding new entry cn=First Login Role,ou=people,dc=example,dc=com

$
```

3 使用服務類別指定新建立帳號的密碼策略。

```
$ cat newCoS.ldif
dn: cn=First Login Template,dc=example,dc=com
objectClass: top
objectClass: nsContainer
```

```
dn: cn="cn=First Login Role,ou=people,dc=example,dc=com",
 cn=First Login Template,dc=example,dc=com
objectClass: top
objectClass: extensibleObject
objectClass: LDAPSubEntry
objectClass: CoSTemplate
cosPriority: 1
pwdPolicySubentry: cn=First Login,dc=example,dc=com

dn: cn=First Login CoS,dc=example,dc=com
objectClass: top
objectClass: LDAPSubEntry
objectClass: CoSSuperDefinition
objectClass: CoSClassicDefinition
cosTemplateDN: cn=First Login Template,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: pwdPolicySubentry operational

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -f newCoS.ldif
Enter bind password:
adding new entry cn=First Login Template,dc=example,dc=com

adding new entry cn="cn=First Login Role,ou=people,dc=example,dc=com",
 cn=First Login Template,dc=example,dc=com

adding new entry cn=First Login CoS,dc=example,dc=com

$
```

範例 8–1 檢查密碼策略指定

增加符合已增加的角色之新使用者。增加使用者以驗證新使用者遵循新密碼策略，而現有使用者則不用。

```
$ cat quentin.ldif
dn: uid=qcubbins,ou=People,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
uid: qcubbins
givenName: Quentin
sn: Cubbins
cn: Quentin Cubbins
mail: quentin.cubbins@example.com
userPassword: ch4ngeM3!
description: New account
```

```
$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f quentin.ldif
Enter bind password:
adding new entry uid=qcubbins,ou=People,dc=example,dc=com

$ ldapsearch -D uid=kvaughan,ou=people,dc=example,dc=com -w - \
-b dc=example,dc=com uid=qcubbins nsrole pwdPolicySubentry
Enter bind password:
version: 1
dn: uid=qcubbins,ou=People,dc=example,dc=com
nsrole: cn=first login role,ou=people,dc=example,dc=com
pwdPolicySubentry: cn=First Login,dc=example,dc=com
$ ldapsearch -b dc=example,dc=com uid=bjensen nsrole pwdPolicySubentry
version: 1
dn: uid=bjensen, ou=People, dc=example,dc=com
$
```

請注意，Barbara Jensen 的現有帳號由預設密碼策略所決定。但是，Quentin Cubbins 的新帳號則是由您定義的密碼策略所決定。

在 pwdSafeModify 為 TRUE 時從指令行修改密碼

當使用者密碼策略的 pwdSafeModify 設爲 TRUE 時，必須提供舊密碼，才能以新密碼變更密碼。預設密碼策略的指令 dsconf set-server-prop pwd-safe-modify-enabled:on 有相同效果。

您可以使用 ldappasswd(1) 指令變更密碼。此指令可支援安全密碼修改。此指令執行 RFC 3062「LDAP Password Modify Extended Operation (http://www.ietf.org/rfc/rfc3062.txt)」

您可以使用 ldapmodify(1) 指令變更密碼。傳遞給案例中 ldapmodify 指令的 LDIF 應如下所示：

```
dn: DN of user whose password you are changing
changetype: modify
delete: userPassword
userPassword: old password
-
add: userPassword
userPassword: new password
```

您也可以使用 LDAP 密碼修改延伸的作業。設定延伸作業的支援說明於第 180 頁的 「使用密碼修改延伸作業重設密碼」中。

重設過期的密碼

當密碼策略強制執行密碼過期時，部分使用者將無法及時變更密碼。本節顯示如何變更過期的密碼。

備註 – 目錄伺服器會在每次項目上的密碼有所修改時，更新作業屬性 `pwdChangedTime(5dsat)`。因此，若要啓用密碼過期，已超過期限的使用者密碼將在您啓用密碼過期時立即過期。若您覺得此運作方式不合用，請使用警告與寬限登入。

本節包含使用密碼修改延伸作業以重設密碼，以及在密碼過期時允許寬限認證的程序。

本節所述的機制旨在供管理員使用，或供處理實際使用者與目錄互動的應用程式使用。您通常會信任應用程式會確認一般使用者使用機制的方式實際上與您所要的方式相同。

▼ 使用密碼修改延伸作業重設密碼

使用者帳號會在密碼過期時鎖定。重設密碼時會解除鎖定帳號。管理員等其他使用者可以重設密碼。重設密碼之後，目錄伺服器會解除鎖定使用者帳號。目錄伺服器支援 RFC 3062「LDAP Password Modify Extended Operation (`http://www.ietf.org/rfc/rfc3062.txt`)」。延伸作業可讓您允許目錄伺服器管理員或目錄應用程式，透過密碼重設以解除鎖定帳號。

如本程序中所示，允許使用密碼修改延伸作業時請小心。請將存取權僅授予所信任的管理員與應用程式。請勿以純文字格式在網路上傳遞密碼。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **授予使用者密碼管理員或密碼管理應用程式的存取權。**

2. **允許密碼管理員存取並使用密碼修改延伸作業。**

   下列指令設定 ACI 允許 `Password Managers` 角色成員在 SSL 連線上使用密碼修改延伸作業：

```
$ cat exop.ldif
dn: oid=1.3.6.1.4.1.4203.1.11.1,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid: 1.3.6.1.4.1.4203.1.11.1
cn: Password Modify Extended Operation
aci: (targetattr != "aci")(version 3.0;
 acl "Password Modify Extended Operation
 "; allow( read, search, compare, proxy ) (roledn = "
```

```
 ldap:///cn=Password Managers,dc=example,dc=com" and authmethod = "SSL");)

$ ldapmodify -a -D cn=admin,cn=Administrators,cn=config -w - -f exop.ldif
Enter bind password:
adding new entry oid=1.3.6.1.4.1.4203.1.11.1,cn=features,cn=config

$
```

`cn=features,cn=config` 下的項目可讓您管理對於使用密碼修改延伸作業的存取權。

3 **讓密碼管理員重設使用者密碼**。

此步驟會解除鎖定使用者帳號，且可以使用 `ldappasswd(1)` 指令完成。

4 **(可選擇) 如果使用者必須變更密碼，請讓密碼管理員通知使用者**。

如果決定使用者項目的密碼策略包含 `pwdMustChange: TRUE`，使用者必須在重設密碼之後變更密碼。

▼ 密碼過期時允許寬限認證

本程序說明如何提供使用者寬限認證，讓使用者可以變更過期的密碼。

寬限認證會由處理密碼策略請求與回應控制的應用程式所管理。此程序的簡單範例顯示如何在應用程式中使用控制。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 **請確定使用者具有使用密碼策略請求與回應控制的應用程式之存取權**。

該應用程式應確保使用者可適當地處理寬限認證。

2 **允許應用程式使用密碼策略控制**。

下列指令設定 ACI 可允許 `Password Managers` 角色成員使用密碼策略控制：

```
$ cat ctrl.ldif
dn: oid=1.3.6.1.4.1.42.2.27.8.5.1,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid: 1.3.6.1.4.1.42.2.27.8.5.1
cn: Password Policy Controls
aci: (targetattr != "aci")(version 3.0; acl "Password Policy Controls
 "; allow( read, search, compare, proxy ) roledn = "
 ldap:///cn=Password Managers,dc=example,dc=com";)

$ ldapmodify -a -D cn=admin,cn=Administrators,cn=config -w - -f ctrl.ldif
Enter bind password:
adding new entry oid=1.3.6.1.4.1.42.2.27.8.5.1,cn=features,cn=config
```

```
$
```

cn=features,cn=config 下的項目之唯一目的，是讓您管理使用密碼策略請求與回應控制的作業之存取權。

3. **將密碼策略中的 pwdGraceAuthNLimit 設為密碼過期後可允許的認證數目。**

4. **請確定應用程式會引導一般使用者在寬限認證無效之前，及時變更過期的密碼。**

設定帳號特性

以下小節說明如何設定帳號的查詢限制、大小限制、時間限制與閒置逾時。

▼ 設定帳號的查詢限制

- **使用 ldapmodify 指令可設定 nsLookThroughLimit 的值。**
  下列指令可移除 Barbara Jensen 的查詢限制：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsLookThroughLimit
nsLookThroughLimit: -1
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

▼ 設定帳號的大小限制

- **使用 ldapmodify 指令可設定 nsSizeLimit 的值。**
  下列指令可移除 Barbara Jensen 的大小限制：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsSizeLimit
nsSizeLimit: -1
^D
```

```
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

▼ 設定帳號的時間限制

- **使用 `ldapmodify` 指令可設定 `nsTimeLimit` 的值。**
  下列指令可移除 Barbara Jensen 的時間限制：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsTimeLimit
nsTimeLimit: -1
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

▼ 設定帳號的閒置逾時

- **使用 `ldapmodify` 指令可設定 `nsIdleTimeout` 的值。**
  下列指令將 Barbara Jensen 的閒置逾時設爲五分鐘 (300 秒)：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsIdleTimeout
nsIdleTimeout: 300
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

手動鎖定帳號

目錄伺服器可讓您配置密碼策略，在達到指定的失敗連結嘗試次數時，強制封鎖帳號。如需詳細資訊，請參閱第 164 頁的 「帳號封鎖策略」。本節包含目錄管理員可以使用的手動帳號鎖定與啓用工具。

目錄管理員無須使用封鎖持續時間計時器，便可管理帳號封鎖。鎖定的帳號會在密碼手動重設之前維持在鎖定狀態。目錄管理員也可以無限地停用部分帳號。

本節顯示如何檢查帳號狀態、停用帳號以及重新啓用帳號。

▼ 檢查帳號狀態

如此處所示檢查帳號狀態。

備註 – 您必須以目錄管理員身份連結。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **使用 `ns-accountstatus` 指令檢查帳號或角色的狀態。**

  下列指令會檢查 Barbara Jensen 的帳號狀態：

```
$ ns-accountstatus -D "cn=Directory Manager" -j pwd.txt \
 -I uid=bjensen,ou=people,dc=example,dc=com
uid=bjensen,ou=people,dc=example,dc=com activated.
$
```

  如需詳細資訊，請參閱 `ns-accountstatus`(1M) 線上手冊。

▼ 停用帳號

如此處所示停用帳號或角色。

備註 – 您必須以目錄管理員身份連結。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **使用 `ns-inactivate` 指令停用帳號或角色。**

  下列指令會停用 Barbara Jensen 的帳號：

```
$ ns-inactivate -D "cn=Directory Manager" -j pwd.txt \
-I uid=bjensen,ou=people,dc=example,dc=com
uid=bjensen,ou=people,dc=example,dc=com inactivated.
$
```

如需詳細資訊，請參閱 ns-inactivate(1M) 線上手冊。

▼ 重新啓用帳號

如此處所示解除鎖定帳號或角色。

備註 – 您必須以目錄管理員身份連結。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **使用 ns-activate 指令重新啓用帳號或角色。**
  下列指令會重新啓用 Barbara Jensen 的帳號：

```
$ ns-activate -D "cn=Directory Manager" -j pwd.txt \
-I uid=bjensen,ou=people,dc=example,dc=com
uid=bjensen,ou=people,dc=example,dc=com activated.
$
```

  如需詳細資訊，請參閱 ns-activate(1M) 線上手冊。

◆ ◆ ◆ 第 9 章

9 目錄伺服器備份與復原

您的目錄伺服器所管理的資料經常需大量匯入。Directory Server Enterprise Edition 提供了可匯入及匯出整個尾碼的工具。它也提供可同時備份所有尾碼以及從備份復原所有資料的工具。

開始進行任何備份或復原作業前，請確實設計適合本身情況所需的備份與復原策略。如需有關不同備份選項的更多資訊、應考量的事項，以及規劃備份與復原策略的指導方針，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中的「Designing Backup and Restore Policies」。

本章包含下列主題：

- 第 187 頁的「二進位備份」
- 第 190 頁的「備份至 LDIF」
- 第 191 頁的「二進位復原」
- 第 192 頁的「從 LDIF 檔案匯入資料」
- 第 195 頁的「復原已複寫的尾碼」
- 第 198 頁的「嚴重損壞回復」

二進位備份

本節說明如何執行目錄資料的二進位備份。除了本節所提到的二進位備份程序以外，您也可以製作二進位副本，用以初始化複寫拓樸中的尾碼。請參閱第 237 頁的「使用二進位副本初始化複寫的尾碼」。

僅備份目錄資料

二進位資料備份會儲存目錄資料的副本，以便在以後資料庫檔案損毀或遭刪除時使用。此作業不會備份配置資料。若要備份整個目錄伺服器以供嚴重損壞回復之用，請參閱第 198 頁的「嚴重損壞回復」。

注意 – 切勿在備份作業期間停止伺服器。

執行備份的頻率必須高於**清除延遲**。由屬性 nsDS5ReplicaPurgeDelay 所指定的清除延遲，係指對變更記錄執行內部清除作業之前所歷經的一段時間 (以秒爲單位)。預設清除延遲爲 604800 秒 (1 週)。變更記錄會保存更新記錄，可能已複寫、也可能未複寫。

若您執行備份的頻率低於清除延遲，變更記錄可能會在備份之前即已清除。因此，當您使用備份復原資料時，即會遺失變更。

本節所述之一切備份程序，預設會將伺服器檔案的副本儲存在相同的主機上。接著，您應將備份複製並儲存到不同的機器或檔案系統上，以確保更高的安全性。

▼ 備份目錄資料

您必須停止目錄伺服器，方可執行 dsadm backup 指令。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **備份目錄資料**。

  ```
  $ dsadm backup instance-path archive-dir
  ```

  例如：

  ```
  $ dsadm backup /local/ds /local/tmp/20051205
  ```

備註 – 您可以在伺服器執行時，使用 dsconf backup 指令備份目錄資料。但目錄資料若在備份執行時有所變更，即難以正確回復。若要避免在使用 dsconf backup 時出現此問題，請設定複寫參照，或將伺服器設爲唯讀。

如需有關 dsadm 與 dsconf 指令的更多資訊，請參閱 dsadm(1M) 線上手冊與 dsconf(1M) 線上手冊。

▼ 備份 dse.ldif 檔案

復原伺服器時，dse.ldif 配置檔案必須包含與伺服器備份時相同的配置資訊。

- **備份您的 dse.ldif 配置檔案**。

  ```
  $ cp instance-path/config/dse.ldif archive-dir
  ```

當您執行下列動作時，目錄伺服器會自動備份目錄 *instance-path*/config 中的 dse.ldif 配置檔案。

- 當您啓動目錄伺服器時，即會在名爲 dse.ldif.startOK 的檔案中建立 dse.ldif 檔案的備份。
- 當您修改 cn=config 分支時，檔案會先備份到 config 目錄中的 dse.ldif.bak 檔案，接著伺服器才會將修改寫入 dse.ldif 檔案中。

備份檔案系統

此程序會使用**凍結模式**功能。凍結模式可讓您停止磁碟的資料庫更新，因此執行檔案系統快照作業時將更加安全。您可以額外採用凍結模式，使備份更形可靠。

進行檔案系統備份時，絕不能讓伺服器將使用者資料寫入磁碟中。若您確定在某段時間內不會有更新，請在這段期間進行備份。若您無法確定未來是否會有更新，請先將伺服器設爲凍結模式，再進行備份。

使用凍結模式的伺服器會繼續寫入存取與錯誤記錄。在單一伺服器拓樸中，於凍結模式開啓時接收的作業，會導致傳回 LDAP 錯誤。所記錄的錯誤訊息即爲離線資料庫的標準錯誤。在複寫拓樸中，會傳回參照。若要讓凍結模式正確運作，即不應在資料庫上執行任何其他作業。

請注意，使用凍結模式的伺服器資料庫比使用唯讀模式的伺服器資料庫更爲穩定。唯讀模式與凍結模式不同，它允許建立作業及修改配置項目。開啓凍結模式後，所有配置的資料庫均會離線。任何進行中的內部作業，都會收到資料庫即將離線的通知。進行中的 LDAP 作業會先行完成，同時資料庫環境會進行清除。包括搜尋使用者資料等後續的內送作業，在凍結模式設爲關閉之前都將遭拒絕。但在凍結模式開啓期間，您仍可搜尋配置參數。

▼ 備份檔案系統

針對此程序的某些部分，您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。此程序的其他部分只能使用指令行完成。

1. **(可選擇) 將伺服器設為凍結模式。**

   ```
   $ dsconf set-server-prop -h host -p port read-write-mode:frozen
   ```

2. **使用適合您檔案系統類型的工具，進行檔案系統的備份。**

3. **若您的伺服器已在凍結模式中，請重新將伺服器設為讀寫模式。**

   ```
   $ dsconf set-server-prop -h host -p port read-write-mode:read-write
   ```

   若您的伺服器收到來自其他伺服器的複寫更新，這些更新將在凍結模式關閉後立即啓動。

備份至 LDIF

備份至 LDIF 可讓您將目錄資料備份至格式化的 LDIF 檔案中。

匯出至 LDIF

您可以利用 LDIF 所匯出的尾碼內容，備份目錄資料。匯出資料對執行下列作業大有助益：

- 備份您伺服器中的資料
- 將您的資料複製到其他目錄伺服器
- 將您的資料匯出至其他應用程式
- 在變更您的目錄拓樸後重新寫入尾碼

匯出作業並不會匯出配置資訊 (`cn=config`)。

注意 – 請勿於匯出作業進行時停止伺服器。

▼ 將尾碼匯出至 LDIF

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **使用下列其中一項指令，將尾碼匯出至 LDIF 檔案：**
  - 若您的伺服器是本機伺服器，並且已停止，請鍵入：

    ```
    $ dsadm export instance-path suffix-DN LDIF-file
    ```
  - 若您的伺服器位於遠端且正在執行中，請鍵入：

    ```
    $ dsconf export -h host -p port suffix-DN LDIF-file
    ```

  下列範例將使用 `dsconf export`，將兩個尾碼匯出至同一個 LDIF 檔案：

  ```
  $ dsconf export -h host1 -p 1389 ou=people,dc=example,dc=com \
   ou=contractors,dc=example,dc=com /local/ds/ldif/export123.ldif
  ```

  `dsadm export` 與 `dsconf export` 指令亦可與 `--no-repl` 選項搭配使用，以指定將不匯出任何複寫資訊。依預設，複寫的尾碼會與複寫資訊一併匯出至 LDIF 檔案。產生的 LDIF 檔案會包含複寫機制所使用的屬性子類型。此 LDIF 檔案可接著匯入用戶伺服器以初始化用戶複本，如第 234 頁的 「初始化複本」中所述。

  如需有關這些指令的更多資訊，請參閱 `dsadm`(1M) 與 `dsconf`(1M) 線上手冊。

二進位復原

下列程序將說明如何復原您目錄中的尾碼。您的伺服器必須使用第 187 頁的 「僅備份目錄資料」中所述之程序完成備份。復原複寫協議中所載的尾碼之前，請先參閱第 195 頁的 「復原已複寫的尾碼」。

注意 – 請勿於復原作業期間停止伺服器。由於復原伺服器會覆寫所有現有的資料庫檔案，因此備份之後的一切資料修改都會遺失。

▼ 復原伺服器

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **請使用下列其中一項指令復原您的伺服器：**
  - 若您的伺服器是本機伺服器，並且已停止，請鍵入：

    ```
    $ dsadm restore instance-path archive-dir
    ```

    例如，若要從備份目錄中復原備份，請鍵入：

    ```
    $ dsadm restore /local/ds/ local/ds/bak/2006_07_01_11_34_00
    ```
  - 若您的伺服器位於遠端且正在執行中，請鍵入：

    ```
    $ dsconf restore -h host -p port archive-dir
    ```

    例如，若要從備份目錄中復原備份：

    ```
    $ dsconf restore -h host1 -p 1389 /local/ds/bak/2006_07_01_11_34_00
    ```

如需有關這些指令的更多資訊，請參閱 `dsadm`(1M) 與 `dsconf`(1M) 線上手冊。

復原 `dse.ldif` 配置檔案

目錄伺服器會在以下目錄中建立兩個 `dse.ldif` 檔案的備份副本：

instance-path`/config`

`dse.ldif.startOK` 檔案會在伺服器啓動時，記錄 `dse.ldif` 檔案的副本。`dse.ldif.bak` 檔案含有 `dse.ldif` 檔案最新變更的備份。將含有最新變更的檔案複製到您的目錄中。

▼ 復原 dse.ldif 配置檔案

針對此程序的某些部分，您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。此程序的其他部分只能使用指令行完成。

1. **停止伺服器。**

   ```
   $ dsadm stop instance-path
   ```

2. **切換至含有配置檔案的目錄。**

   ```
   $ cd instance-path/config
   ```

3. **以已知有效的備份配置檔案覆寫 `dse.ldif` 檔案，例如：**

   ```
   $ cp dse.ldif.startOK dse.ldif
   ```

4. **以下列指令啓動伺服器：**

   ```
   $ dsadm start instance-path
   ```

從 LDIF 檔案匯入資料

您可以使用下列方式將資料匯入目錄伺服器尾碼：

- 從 LDIF 檔案初始化尾碼。此作業會刪除尾碼中目前的資料，並將其取代爲 LDIF 檔案的內容。
- 使用 LDIF 檔案執行大量 `ldapadd`、`ldapmodify` 或 `ldapdelete` 作業。如此可讓您大量增加、修改及刪除目錄中任何尾碼的項目。

下表說明初始化尾碼與大量增加、修改及刪除項目等作業間的差異。

表 9–1 初始化尾碼與大量匯入資料的比較

比較網域	初始化尾碼	大量增加、修改及刪除項目
覆寫內容	覆寫 內容	不覆寫內容
LDAP 作業	僅增加	增加、修改、刪除
效能	快速	較緩慢
回應伺服器失敗	不可分割 (失敗後會遺失所有變更)	最大效率 (保留失敗前所做的所有變更)
LDIF 檔案位置	用戶端或伺服器本機	於用戶端機器

表 9–1 初始化尾碼與大量匯入資料的比較 *(續)*

比較網域	初始化尾碼	大量增加、修改及刪除項目
匯入配置資訊 (cn=config)	匯入配置資訊	不匯入配置資訊

指令	若伺服器爲本機伺服器，並且已停止：
	`dsadm import`
	若伺服器位於遠端，且正在執行中：
	`dsconf import`
		`ldapmodify -B`

初始化尾碼

初始化尾碼後，會將尾碼中的現有資料覆寫爲僅含有增加項目之 LDIF 檔案的內容。

您必須認證爲「目錄管理員」或「管理員」，方可初始化尾碼。

伺服器執行時，只有「目錄管理員」與「管理員」可匯入含有根項目的 LDIF 檔案。基於安全性考量，只有這些使用者可存取尾碼的根項目，例如 `dc=example,dc=com`。

復原複寫協議中所載的尾碼之前，請先參閱第 195 頁的 「復原已複寫的尾碼」。

▼ 初始化尾碼

備註 – 您所匯入的所有 LDIF 檔案均須使用 UTF-8 字元集編碼。

初始化尾碼時，LDIF 檔案必須含有對應尾碼的根項目與所有目錄樹狀結構節點。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **使用下列其中一項指令，可從 LDIF 檔案初始化尾碼，亦即將資料庫內容匯入 LDIF 檔案中。**

注意 – 這些指令可覆寫您尾碼中的資料。

  - 若您的伺服器是本機伺服器，並且已停止，請鍵入：

    ```
    $ dsadm import instance-path LDIF-file suffix-DN
    ```

    下列範例將使用 `dsadm import` 指令，將兩個 LDIF 檔案匯入單一尾碼中：

```
$ dsadm import /local/ds /local/file/example/demo1.ldif \
 /local/file/example/demo2.ldif dc=example,dc=com
```

- 若您的伺服器位於遠端且正在執行中，請鍵入：

```
$ dsconf import -h host -p port LDIF-file suffix-DN
```

下列範例使用 `dsconf import` 匯入 LDIF 檔案。執行此指令時不需具備超級使用者權限，但必須將您認證爲具有超級使用者權限的使用者，如「目錄管理員」。

```
$ dsconf import -h host1 -p 1389 /local/file/example/demo1.ldif \
 ou=People,dc=example,dc=com
```

備註 – 如果在多個尾碼上平行執行 `dsconf import`、`dsconf reindex` 或同時執行兩個指令，作業事件記錄會變大而可能對效能造成不良影響。

如需有關這些指令的更多資訊，請參閱 `dsadm`(1M) 與 `dsconf`(1M) 線上手冊。

大量增加、修改及刪除項目

執行 `ldapmodify` 作業時，您可以大量增加、修改或刪除項目。這些項目指定於含有修改或刪除現有項目之更新陳述式的 LDIF 檔案中。此作業不會清除已存在的項目。

您的目錄伺服器所管理的任何尾碼均可爲變更項目的目標。與任何其他增加項目的作業相同，伺服器也會在所有的新項目匯入時爲其編製索引。

`ldapmodify` 指令會透過 LDAP 匯入 LDIF 檔案，並執行該檔案包含的所有作業。使用此指令可讓您同時修改所有目錄尾碼中的資料。

復原複寫協議中所載的尾碼之前，請先參閱第 195 頁的 「復原已複寫的尾碼」。

▼ 大量增加、修改及刪除項目

備註 – 您所匯入的所有 LDIF 檔案均須使用 UTF-8 字元集編碼。

匯入 LDIF 檔案時，父系項目必須位於目錄中，或先從檔案中增加。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **從 LDIF 檔案大量增加、修改或刪除。**

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -B baseDN -f LDIF-file
```

下列範例將使用 ldapmodify 指令執行匯入。執行此指令不需要超級使用者權限，但是您必須以具有超級使用者權限的使用者身份進行認證，例如 cn=Directory Manager 或 cn=admin,cn=Administrators,cn=config。最後一個參數指定所匯入的 LDIF 檔案名稱。

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - \
 -B dc=example,dc=com -f /local/ds/ldif/demo.ldif
```

復原已複寫的尾碼

進行復原前，需要特別注意供應者伺服器與用戶伺服器之間複寫的尾碼。請儘可能透過複寫機制更新尾碼，而不要從備份進行復原。

復原供應者或集散中心實例時，伺服器配置的內容必須與製作備份時相同。爲確實做到這一點，請在復原目錄伺服器資料前，先復原 dse.ldif 檔案。請參閱第 191 頁的「復原 dse.ldif 配置檔案」。

本節說明復原複本的方法與時機，以及如何確定此複本與其他複本在作業後仍保持同步。若要初始化複本，請參閱第 234 頁的「初始化複本」。

若您具有大型複寫尾碼，而想要增加許多項目並確定複寫更新皆正確增加，請參閱第 241 頁的「遞增多個項目到大型複寫的尾碼」。

本節包含下列項目的相關資訊：

- 第 195 頁的「復原單一主伺服器方案中的供應者」
- 第 196 頁的「復原多重主伺服器方案中的供應者」
- 第 196 頁的「復原集散中心」
- 第 197 頁的「復原專屬用戶」
- 第 197 頁的「復原多重主伺服器方案中的主伺服器」

復原單一主伺服器方案中的供應者

屬於單一主伺服器供應者的尾碼，含有整個複寫拓樸的授權資料。因此，復原此尾碼即等於重新初始化整個拓樸中的所有資料。只有在需要從所要復原的備份內容重新初始化所有資料時，始應復原單一主伺服器。

若單一主伺服器資料因錯誤而無法回復，請考慮使用其中一個用戶的資料，因爲其中可能包含比備份還新的更新。在此情況下，您必須將用戶複本的資料匯出至 LDIF 檔案，再從 LDIF 檔案重新初始化主伺服器。

無論您選擇復原備份還是在主伺服器複本上匯入 LDIF 檔案，接下來都必須重新初始化由此複本接收更新的所有集散中心與用戶複本。供應者伺服器的記錄檔會記錄一則訊息，提醒您必須重新初始化用戶。

復原多重主伺服器方案中的供應者

在多重主伺服器複寫中，其他主伺服器均各含有複寫資料的授權副本。您無法復原舊的備份，因爲它目前的複本內容可能已過期。請儘可能以複寫機制將主伺服器更新爲其他主伺服器的內容。

若無法這麼做，請以下列其中一種方式復原多重主伺服器複本：

- 最簡單的方法是不要復原備份，而從其他主伺服器之一重新初始化預定主伺服器。如此可確保最新的資料將可傳送至預定主伺服器，並可供複寫之用。請參閱第 235 頁的「從 LDIF 初始化複本」。
- 至於有數百萬個項目的複本，製作二進位副本以復原其他主伺服器之一的較新備份，可能是比較快的做法。請參閱第 237 頁的「使用二進位副本初始化複寫的尾碼」。
- 若您的主伺服器備份不比**任何**其他主伺服器的變更記錄內容最長存在時間舊，即可使用備份進行此主伺服器的復原。如需變更記錄存在時間的說明，請參閱第 228 頁的「修改主伺服器複本上的變更記錄設定」。復原舊的備份時，其他主伺服器將使用其變更記錄中所有在備份儲存後處理的修改，對此主伺服器進行更新。

無論您以何種方式復原或重新初始化，主伺服器複本在初始化後仍會處於唯讀模式。此運作方式可讓複本與其他主伺服器進行同步化，以便您後續允許寫入作業，如第 197 頁的「復原多重主伺服器方案中的主伺服器」中所述。

在復原或重新初始化的主伺服器上允許寫入作業前先彙整所有複本的好處在於，所有集散中心或用戶伺服器都將不需再進行重新初始化。

復原集散中心

本節僅適用於複寫機制無法自動更新集散中心複本的情況。資料庫檔案受損或複寫作業中斷過久，都屬於這種情況。在此類情況下，您必須以下列其中一種方式復原或重新初始化集散中心複本：

- 最簡單的方法是不要復原備份，而從其中一個主伺服器複本重新初始化集散中心。如此可確保最新的資料將可傳送至集散中心，並可供複寫之用。請參閱第 193 頁的「初始化尾碼」。
- 至於有數百萬個項目的複本，製作二進位副本以復原來自其他集散中心之複寫尾碼的較新備份，可能是比較快的做法。請參閱第 237 頁的「使用二進位副本初始化複寫的尾碼」。若沒有其他集散中心複本可複製，請以前述方式重新初始化集散中心，或以後述方式加以復原，視情況而定。
- 若您集散中心的備份不比其**任何**供應者的變更記錄內容最長存在時間舊，即可使用備份 (集散中心或主伺服器複本) 進行此集散中心的復原。復原集散中心時，其供應者將使用其變更記錄中所有在備份儲存後處理的修改，對此集散中心進行更新。

備註 – 無論您以何種方式復原或重新初始化集散中心複本，您接下來都**必須**重新初始化此集散中心的所有用戶，包括任何其他層級的集散中心在內。

復原專屬用戶

本節僅適用於複寫機制無法自動更新專屬用戶複本的情況。資料庫檔案受損或複寫作業中斷過久，都屬於這種情況。在此類情況下，您必須以下列其中一種方式復原或重新初始化用戶：

- 最簡單的方法是不要復原備份，而由其供應者之一重新初始化用戶 (主伺服器或集散中心複本)。如此可確保最新的資料將可傳送至用戶，並可供複寫之用。請參閱第 235 頁的 「從 LDIF 初始化複本」。
- 至於有數百萬個項目的複本，製作二進位複本以復原來自其他用戶之複寫尾碼的較新備份，可能是比較快的做法。請參閱第 237 頁的 「使用二進位副本初始化複寫的尾碼」。若沒有其他用戶可複製，請以前述方式重新初始化此複本，或以後述方式加以復原，視情況而定。
- 若您用戶的備份不比其**任何**供應者的變更記錄內容最長存在時間舊，即可使用備份 (集散中心或主伺服器複本) 進行此用戶的復原。復原用戶時，其供應者將使用其變更記錄中所有在備份儲存後處理的修改，對此用戶進行更新。

復原多重主伺服器方案中的主伺服器

使用多重主伺服器複寫時，其他主伺服器可能會在指定主伺服器復原期間處理變更作業。因此，當復原完成時，新的主伺服器必須再行接收復原資料中未納入的最新更新。由於復原主伺服器可能需要很長的時間，在此期間擱置的更新也可能因此爲數眾多。

爲彙整這些擱置更新，剛復原的主伺服器將自動設爲唯讀模式，以利復原後的用戶端作業執行。但只有在透過指令行從 LDIF 檔案匯入資料或使用備份執行二進位複本，以進行主伺服器的復原時，才會設定爲此。

因此，在復原之後，多重主伺服器配置中的主伺服器會處理複寫更新並允許讀取作業，但也會傳回所有來自用戶端之寫入作業的參照。

若要在允許更新前先驗證新的主伺服器是否已完全與其他主伺服器同步化，請手動啓用初始化之主伺服器的更新。

備註 – 由於主伺服器複本可能因其新運作方式而會傳送參照之故，需執行寫入作業的用戶端可能會達到其配置的躍點限制。您可以增加用戶端的躍點限制配置，使其能夠存取可用的主伺服器。若所有主伺服器複本均已初始化或重新初始化，則所有寫入作業都將失敗，因爲沒有複本會接受用戶端更新。

無論如何，您都應該密切監視已初始化的主伺服器，並適當設定參照屬性，以儘可能提升伺服器的回應能力。

▼ 開始透過指令行接受更新

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

您可以在程序檔中使用下列指令，將初始化多重主伺服器複本的程序自動化。

1. **使用 `insync` 工具，可確保複本已與所有其他的主伺服器彙整。**

   若所有伺服器上各修改間的延遲爲零，或複本從未有變更需複寫 (延遲爲 `-1`)，表示複本已同步化。如需更多資訊，請參閱 `insync`(1) 線上手冊。

2. **開始接受更新。**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN repl-accept-client-update-enabled:on
   ```

   此指令會自動將伺服器設爲讀寫模式。

嚴重損壞回復

若要備份或復原您的目錄伺服器以供嚴重損壞回復之用，請使用下列程序。

▼ 製作嚴重損壞回復所需的備份

針對此程序的某些部分，您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。此程序的其他部分只能使用指令行完成。

1. **使用 `dsadn backup` 或 `dsconf backup` 指令備份您的資料庫檔案。**

   使用第 187 頁的 「二進位備份」中的程序，並將備份檔案儲存於安全的位置。

2. **將配置目錄 *instance-path*`/config` 複製到安全的位置。**

3. **將模式目錄 *instance-path*`/config/schema` 複製到安全的位置。**

4. **將別名目錄 *instance-path*`/alias` 複製到安全的位置。**

▼ 復原嚴重損壞回復

針對此程序的某些部分，您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。此程序的其他部分只能使用指令行完成。

1. **安裝與先前主機上相同版本的目錄伺服器。**
2. **使用 `dsadm create` 指令建立伺服器實例。**
   使用在備份時所使用的相同實例。請參閱第 62 頁的「建立尾碼」。
3. **復原配置目錄** *instance-path*`/config`。
4. **復原模式目錄** *instance-path*`/config/schema`。
5. **復原別名目錄** *instance-path*`/alias`。
6. **確定復原的伺服器所使用的配置正確無誤。**
   例如，目錄結構與外掛程式配置均須與備份伺服器上的相同。
7. **使用 `dsconf restore` 指令復原您的資料庫檔案。**
   使用第 191 頁的「二進位復原」中的程序。

◆ ◆ ◆ 第 10 章

10 目錄伺服器群組、角色與 CoS

管理代表使用者的項目除了其資料在目錄中採階層式資料結構之外，通常還需要建立共用屬性值的群組。目錄伺服器透過群組、角色與服務類別 (CoS) 提供進階的項目管理功能。

本章包含下列主題：

- 第 201 頁的 「關於群組、角色與服務類別」
- 第 202 頁的 「管理群組」
- 第 203 頁的 「管理角色」
- 第 207 頁的 「服務類別」
- 第 217 頁的 「維護參照完整性」

關於群組、角色與服務類別

群組、角色與 CoS 定義如下：

- 群組是命名成員清單或成員篩選器之其他項目的項目。若是群組由成員清單所組成，目錄伺服器會爲每個使用者項目的 `isMemberOf` 屬性產生值。因此，使用者項目的 `isMemberOf` 屬性會顯示該項目所屬的所有群組。
- 角色提供與群組相同的功能，且會更進一步透過機制產生每個角色成員的 `nsrole` 屬性。
- CoS 會產生運算屬性，讓項目共用屬性值，而不用在各個項目中儲存該屬性。

  您無法使用 `isMemberOf` 屬性讓靜態群組的所有成員自動繼承共用運算屬性值。

目錄伺服器提供根據角色、群組與 CoS 運算屬性值執行搜尋的能力。任何作業中使用的篩選字串可以包含 `nsRole` 屬性或任何 CoS 定義所產生的屬性。篩選字串也可用以在此屬性值上執行任何比較作業。但是，CoS 運算屬性無法編製索引。因此，與 CoS 產生的屬性相關之任何搜尋，皆可能會消耗大量的時間與記憶體資源。

若要完全發揮角色、群組與服務類別所提供的功能，請在目錄部署的規劃階段中決定群組策略。如需這些功能及其如何能簡化拓樸的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中的「Grouping Directory Data and Managing Attributes」。

若要深入瞭解角色與群組的運作方式，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 8 章「Directory Server Groups and Roles」。如需 CoS 的詳細說明，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 9 章「Directory Server Class of Service」。

管理群組

群組可讓您建立項目的關聯，以方便管理。例如，使用群組可讓定義存取控制指令 (ACI) 更加容易。群組定義是特殊的項目，可以在靜態清單中命名這些項目的成員，或提供定義一組動態項目的篩選。

不論群組定義項目的位置爲何，群組的可能成員範圍爲整個目錄。爲了簡化管理，通常會在單一位置儲存所有群組定義項目，該位置通常位於根尾碼下的 `ou=Groups`。

群組可分爲靜態群組與動態群組兩種類型。

- **靜態群組**。定義靜態群組繼承 `groupOfNames` 或 `groupOfUniqueNames` 物件類別的項目。群組成員會依其 DN 列出，做爲 `member` 或 `uniqueMember` 屬性的多個值。

  您也可以改用靜態群組的 `isMemberOf` 屬性。`isMemberOf` 屬性會在開始搜尋時進行計算並增加至使用者項目，並在完成搜尋之後再次移除。此功能提供群組的簡易管理及快速讀取。
- **動態群組**。定義動態群組的項目繼承 `groupOfURLs` 物件類別。群組成員身份由一或多個在多值 `memberURL` 屬性中指定之篩選所定義。動態群組中的成員即爲每次評估篩選時，符合任一篩選的項目。

▼ 建立新的靜態群組

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 **使用 `ldapmodify` 指令建立新的靜態群組。**

例如，若要建立稱爲 System Administrators 的新靜態群組並增加一些成員，請使用此指令：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=System Administrators, ou=Groups, dc=example,dc=com
changetype: add
cn: System Administrators
objectclass: top
```

```
objectclass: groupOfNames
ou: Groups
member: uid=kvaughan, ou=People, dc=example,dc=com
member: uid=rdaugherty, ou=People, dc=example,dc=com
member: uid=hmiller, ou=People, dc=example,dc=com
```

2 **檢查是否已建立新群組以及是否已增加成員。**

例如，若要檢查 Kirsten Vaughan 是否在新的 System Administrators 群組中，請鍵入：

```
$ ldapsearch -b "dc=example,dc=com" uid=kvaughan isMemberOf
uid=kvaughan,ou=People,dc=example,dc=com
isMemberOf: cn=System Administrators, ou=Groups, dc=example,dc=com
isMemberOf: cn=HR Managers,ou=groups,dc=example,dc=com
```

▼ 建立新的動態群組

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **使用 `ldapmodify` 指令建立新的動態群組。**

例如，若要建立名爲「3rd Floor」的新動態群組，以包含其辦公室號碼開頭爲 3 的所有員工，您可以使用此指令：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=3rd Floor, ou=Groups, dc=example,dc=com
changetype: add
cn: 3rd Floor
objectclass: top
objectclass: groupOfUrls
ou: Groups
memberURL: ldap:///dc=example,dc=com??sub?(roomnumber=3*)
```

管理角色

角色是設計用來讓應用程式更有效且更容易使用的替代分組機制。雖然角色以類似群組的方式定義與管理，但是每個成員項目所產生的角色屬性會自動表示項目的角色。例如，應用程式可以讀取項目的角色，而不是選取群組並瀏覽成員清單。

角色的範圍預設會限制在所定義範圍的子樹狀結構內。但是，您可以延伸巢式角色的範圍。您可以允許範圍延伸到位於其他子樹狀結構的巢式角色以及包含目錄中任意位置的成員。如需詳細資訊，請參閱第 206 頁的 「延伸角色的範圍」與第 206 頁的 「巢式角色定義的範例」。

本節說明如何安全地使用角色，以及如何從指令行管理角色。

安全地使用角色

若要安全地使用角色，必須將存取控制指令 (ACI) 設爲保護適當的屬性。例如，使用者 A 擁有管理角色 MR。管理角色相當於靜態群組，經由將 nsRoleDN 屬性增加至項目，明確地指定角色給每個成員項目。已透過指令行使用帳號停用鎖定 MR 角色。亦即，由於該使用者的 nsAccountLock 屬性經運算爲 true，因此使用者 A 無法連結至伺服器。但是，假設使用者已連結，並知悉其已因 MR 角色而處於鎖定狀態。如果不存在 ACI 以防止使用者具備寫入存取 nsRoleDN 屬性，使用者可以從其本身的項目移除 nsRoleDN 屬性，並解除鎖定。

若要避免使用者移除 nsRoleDN 屬性，必須套用 ACI。針對篩選的角色，您必須保護能避免使用者修改屬性以放棄篩選的角色之篩選部分。應禁止使用者增加、刪除或修改篩選的角色所用之屬性。同理，如果已運算篩選的屬性值，可以修改篩選的屬性值之所有屬性皆須受到保護。由於巢式角色包含篩選與管理的角色，應針對巢式角色中所含的各角色考量以上幾點。

如需設定安全性 ACI 的詳細指示，請參閱第 7 章。

從指令行管理角色

角色會在 Directory Administrator 可以透過指令行公用程式存取的項目中定義。建立角色之後，可以依下列方式指定成員給角色：

- 管理角色的成員在其項目中有 nsRoleDN 屬性。
- 篩選角色的成員爲符合 nsRoleFilter 屬性中指定之篩選的項目。
- 巢式角色的成員爲巢式角色定義項目的 nsRoleDN 屬性中所指定之角色成員。

所有角色定義均繼承 LDAPsubentry 與 nsRoleDefinition 物件類別。下列範例顯示各種角色類型特定的其他物件類別與相關屬性。

管理角色定義的範例

若要爲所有行銷員工建立一個角色，請使用下列 ldapmodify 指令：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsSimpleRoleDefinition
objectclass: nsManagedRoleDefinition
cn: Marketing
description: managed role for marketing staff
```

請注意，nsManagedRoleDefinition 物件類別繼承 LDAPsubentry、nsRoleDefinition 與 nsSimpleRoleDefinition 物件類別。

依下列方式更新名爲 Bob 的行銷人員項目，以指定角色：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Bob Arnold,ou=marketing,ou=People,dc=example,dc=com
changetype: modify
add: nsRoleDN
nsRoleDN: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
```

nsRoleDN 屬性類型表示項目是受管理角色的成員之一。管理的角色由角色定義的 DN 所識別。若要讓使用者修改自己的 nsRoleDN 屬性，但不想讓使用者增加或移除 nsManagedDisabledRole，請增加下列 ACI：

```
aci: (targetattr="nsRoleDN")(targattrfilters="add=nsRoleDN:
(!(nsRoleDN=cn=AdministratorRole,dc=example,dc=com)),
del=nsRoleDN:(!(nsRoleDN=cn=nsManagedDisabledRole,dc=example, dc=com)")
(version3.0;aci "allow mod of nsRoleDN by self except for critical values";
allow(write) userdn="ldap:///self";)
```

篩選角色定義的範例

若要爲銷售經理設定篩選角色 (假設所有經理皆有 isManager 屬性)，請使用下列 ldapmodify 指令：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=ManagerFilter,ou=sales,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: ManagerFilter
nsRoleFilter: (isManager=True)
Description: filtered role for sales managers
```

請注意，nsFilteredRoleDefinition 物件類別繼承 LDAPsubentry、nsRoleDefinition 與 nsComplexRoleDefinition 物件類別。nsRoleFilter 屬性類型可指定篩選，尋找 ou=sales 組織中有部屬的所有員工，例如：

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Fuentes)"
dn: cn=Carla Fuentes,ou=sales,ou=People,dc=example,dc=comcn: Carla Fuentes
isManager: TRUE...
nsRole: cn=ManagerFilter,ou=sales,ou=People,
dc=example,dc=com
```

備註 – 篩選角色的篩選字串可以根據任何屬性，CoS 機制產生的運算屬性除外。

當篩選的角色成員爲使用者項目時，您可以選擇限制成員增加或移除角色成員的能力。使用 ACI 保護篩選的屬性。

巢式角色定義的範例

巢式角色進行巢狀的角色會使用 nsRoleDN 屬性指定。使用下列指令可建立包含上一個範例中所建立的行銷人員與行銷經理角色成員之角色：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=MarketingSales,ou=marketing,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsNestedRoleDefinition
cn: MarketingSales
nsRoleDN: cn=ManagerFilter,ou=sales,ou=People,dc=example,dc=com
nsRoleDN: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
nsRoleScopeDN: ou=sales,ou=People,dc=example,dc=com
```

請注意，nsNestedRoleDefinition 物件類別繼承 LDAPsubentry、nsRoleDefinition 與 nsComplexRoleDefinition 物件類別。nsRoleDN 屬性包含行銷管理角色與行銷經理篩選角色的 DN。上一個範例中的兩個使用者 Bob 與 Carla 會是此新巢式角色的成員。

此篩選範圍包含預設範圍 (亦即篩選所在的子樹狀結構) 以及任何 nsRoleScopeDN 屬性值下的子樹狀結構。此時，ManagerFilter 位於 ou=sales,ou=People,dc=example,dc=com 子樹狀結構中。此子樹狀結構必須增加至範圍。

延伸角色的範圍

目錄伺服器提供允許角色的範圍延伸到角色定義項目以外的子樹狀結構之屬性。此單值屬性 nsRoleScopeDN 包含要增加至現有角色的範圍 DN。nsRoleScopeDN 屬性僅能增加至巢式角色。請參閱第 206 頁的「巢式角色定義的範例」。

▼ 延伸角色的範圍

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

nsRoleScopeDN 屬性可讓您延伸某個子樹狀結構中的角色範圍，以包含其他子樹狀結構的項目。例如，請想像 example.com 目錄樹狀結構中有兩個主要的子樹狀結構：o=eng,dc=example,dc=com (工程部門子樹狀結構) 與 o=sales,dc=example,dc=com

(行銷部門子樹狀結構)。在工程部門子樹狀結構中，有某個使用者需要存取行銷部門子樹狀結構中的角色 (SalesAppManagedRole) 所管理之行銷部門應用程式。若要延伸角色範圍，請執行下列作業：

1. **在工程部門子樹狀結構中建立使用者的角色。**
   例如，建立角色 EngineerManagedRole。此範例使用管理的角色，不過也有可能是篩選的角色或巢式角色。

2. **例如，在行銷部門子樹狀結構中建立巢式角色 SalesAppPlusEngNestedRole，以裝載新建立的 EngineerManagedRole 與初始 SalesAppManagedRole。**

3. **使用要增加的工程部門子樹狀結構範圍 DN，將 nsRoleScopeDN 屬性增加至 SalesAppPlusEngNestedRole， 在此例中為 o=eng,dc=example,dc=com。**
   工程部門的使用者必須具備必要的權限，才能存取 SalesAppPlusEngNestedRole 角色，進而使用行銷部門的應用程式。此外還必須複寫整個角色範圍。

備註 – 對巢式角色延伸範圍的限制，表示之前是某一個網域中管理角色的管理員，僅有權使用已在另一個網域存在的角色。管理員無法在另一個網域中建立任意的角色。

服務類別

服務類別 (CoS) 機制產生運算屬性的方式，如同爲用戶端應用程式擷取項目一般，因而簡化了項目管理並可降低儲存需求。CoS 機制允許在項目之間共用屬性，且如同群組與角色，CoS 依賴輔助程式項目。

如需如何在部署中使用 CoS 的說明，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中的「Managing Attributes With Class of Service」。

如需如何在目錄伺服器中執行 CoS 的說明，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 9 章「Directory Server Class of Service」。

備註 – 任何搜尋作業皆可測試是否存在 CoS 產生的屬性或比較屬性的值。運算屬性的名稱可能會用在來自用戶端搜尋作業的任何篩選字串中，篩選角色中所用的內部篩選除外。

安全地使用 CoS

下列幾節說明每個 CoS 項目資料的讀取與寫入保護之一般原則。定義個別存取控制指令 (ACI) 的詳細程序如第 7 章中所述。

保護 CoS 定義項目

雖然 CoS 定義項目不包含所產生屬性的值，但會提供尋找該值的資訊。讀取 CoS 定義項目可指出如何尋找包含該值的範本項目。寫入此項目會修改運算屬性的產生方式。

因此您應爲 CoS 定義項目定義讀取與寫入 ACI。

保護 CoS 範本項目

CoS 範本項目包含產生的 CoS 屬性值。因此，範本中的 CoS 屬性至少必須受到 ACI 的讀取與更新保護。

- 在**指標** *CoS* 的案例中，應禁止重新命名單一範本項目。在大多數情況下，最簡單的做法是保護整個範本項目。
- 在**類別** *CoS* 案例中，所有範本項目在定義項目中皆指定有共用父系。如果父系項目中僅儲存了範本，則父系項目的存取控制能保護範本。但是，如果父系下的其他項目需要存取，則必須個別保護範本項目。
- 在**間接** *CoS* 的案例中，範本可以是目錄中的任何項目，包含仍需要進行存取的使用者項目。視需要的不同，您可以控制對整個目錄中 CoS 屬性的存取，或確保 CoS 屬性在每個用爲範本的項目中皆安全。

保護 CoS 的目標項目

在 CoS 定義的範圍中，所有會爲其產生 CoS 運算屬性的項目，皆會參與計算屬性值。

若是在目標項目中已存在 CoS 屬性，CoS 機制預設不會覆寫此值。若不想要此運作方式，請定義 CoS 會覆寫目標項目，或保護所有潛在目標項目中的 CoS 屬性。

間接與類別 CoS 也依賴於目標項目中的限定符號屬性。此屬性會指定要使用的範本項目 DN 或 RDN。您應使用 ACI 在 CoS 的整個範圍內保護此屬性，或在需要保護的個別目標項目上保護此屬性。

保護其他相依性

運算的 CoS 屬性可以根據其他產生的 CoS 屬性與角色進行定義。您必須瞭解並保護這些相依性，以確保運算的 CoS 屬性受到保護。

例如，目標項目中的 CoS 限定符號屬性可能是 `nsRole`。因此，角色定義必須也受到 ACI 保護。

計算運算屬性值的過程中所包含的任何屬性或項目，一般應會有 ACI 控制讀取與寫入存取。因此，應完善規劃或簡化複合相依性，以降低後續存取控制實作的複雜度。將其他運算屬性上的相依性降至最小，能改善目錄效能並減少維護作業。

從指令行管理 CoS

由於所有配置資訊與範本資料皆會儲存爲目錄中的項目，因此您可以使用 LDAP 指令行工具配置並管理 CoS 定義。本節顯示如何從指令行建立 CoS 定義項目與 CoS 範本項目。

從指令行建立 CoS 定義項目

所有 CoS 定義項目皆有 `LDAPsubentry` 物件類別，且繼承 `cosSuperDefinition` 物件類別。此外，每種 CoS 類型皆繼承特定物件類別並包含對應的屬性。下表列出與各種 CoS 定義項目類型相關的物件類別與屬性。

表 10–1 CoS 定義項目中的物件類別與屬性

CoS 類型	CoS 定義項目

指標 CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosPointerDefinition`
	`cosTemplateDN:` *DN*
	`cosAttribute:` *attributeName override merge*

間接 CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosIndirectDefinition`
	`cosIndirectSpecifier:` *attributeName*
	`cosAttribute:` *attributeName override merge*

類別 CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosClassicDefinition`
	`cosTemplateDN:` *DN*
	`cosSpecifier:` *attributeName*
	`cosAttribute:` *attributeName override merge*

`cosAttribute` 一律爲多值。各值會定義 CoS 機制所產生的一個屬性。

您可以使用 CoS 定義項目中的下列屬性。如需有關各屬性的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Man Page Reference」中個別屬性的說明。

表 10–2 CoS 定義項目屬性

屬性	CoS 定義項目內的用途
`cosAttribute` *attributeName override merge*	定義要產生值的運算屬性名稱。此屬性爲多值，每個值代表其值會從範本產生的屬性名稱。*override* 與 *merge* 限定元指定在此表之後所述的特殊案例中，CoS 屬性值的運算方式。 *attributeName* 無法包含任何子類型。含子類型的屬性名稱會遭忽略，不過仍會處理 `cosAttribute` 的其他值。
`cosIndirectSpecifier` *attributeName*	定義目標項目中的屬性名稱，間接 CoS 會使用其值識別範本項目。已命名的屬性稱爲限定符號，必須包含每個目標項目中完整的 DN 字串。此屬性爲單值，但是 *attributeName* 可以是多值，以定義多個範本。
`cosSpecifier` *attributeName*	定義目標項目中的屬性名稱，類別 CoS 會使用其值識別範本項目。已命名的屬性稱爲限定符號，且必須包含範本項目的 RDN 中可以找到的字串。此屬性爲單值，但是 *attributeName* 可以是多值，以定義多個範本。
`cosTemplateDN` *DN*	提供範本項目的完整 DN 給指標 CoS 定義，或提供範本項目的基底 DN 給類別 CoS。此屬性爲單值。

備註 – 您無法使用 `isMemberOf` 屬性做爲 `CosSpecifier`，讓靜態群組的所有成員自動繼承共用運算屬性值。

`cosAttribute` 屬性允許 CoS 屬性名稱之後跟隨兩個限定元：override 限定元與 merge 限定元。

override 限定元說明 CoS 動態產生的屬性已實際存在於項目中時的運作方式。*override* 限定元可以是下列其中之一：

- `default` (**預設**) (或無限定元) - 表示當屬性類型與運算屬性類型相同時，伺服器不會覆寫儲存在項目中的實際屬性值。
- `override` - 表示即使項目已儲存了值，伺服器也一律會傳回 CoS 產生的值。
- `operational` - 表示只有在搜尋中明確請求時，才會傳回該屬性。操作屬性不需要通過模式檢查便會傳回。`operational` 限定元的運作方式與 `override` 限定元相同。

  如果屬性在模式中也定義爲操作，您僅能有一個操作屬性。例如，如果 CoS 產生 `description` 屬性的值，由於 `description` 屬性在模式中未標示爲操作屬性，因此您無法使用 `operational` 限定元。

merge 限定元不存在或爲 merge-schemes。此限定元允許運算的 CoS 屬性可爲來自多個範本或多個 CoS 定義的多重值。如需更多資訊，請參閱第 211 頁的「多值 CoS 屬性」。

覆寫實際屬性值

您可能依下列方式建立包含 override 限定元的指標 CoS 定義項目：

```
dn: cn=pointerCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=exampleUS,cn=data
cosAttribute: postalCode override
```

此指標 CoS 定義項目表示項目與產生 postalCode 屬性值的範本項目 cn=exampleUS,cn=data 相關。override 限定元表示如果目標項目內存在 postalCode 屬性，則此值會取代屬性值。

備註 – 如果使用 operational 或 override 限定元定義 CoS 屬性，則無法在 CoS 範圍內任何項目中的屬性「實際」值上執行寫入作業。

多值 CoS 屬性

當您指定 merge-schemes 限定元時，產生的 CoS 屬性可以兩種方式成爲多值屬性：

- 若是間接或類別 CoS，目標項目中的限定符號屬性可以是多值。在此例中，每個值各會決定一個範本，且來自各個範本的該值是產生的值之一部分。
- 任何類型的多重 CoS 定義項目，可以在其 cosAttribute 中包含相同的屬性名稱。在此例中，如果所有定義包含 merge-schemes 限定元，產生的屬性會包含各個定義所運算的所有值。

這兩種情況可同時發生，並定義更多值。但是在所有情況下，重複的值僅會在產生的屬性中一次傳回。

如果沒有 merge-schemes 限定元，會使用範本項目的 cosPriority 屬性，爲產生的屬性決定所有範本之間的單一值。此範例會在下一節中說明。

merge-schemes 限定元永遠不會合併在目標中以範本產生的值所定義之「實際」值。*merge* 限定元與 *override* 限定元無關。所有配對皆有可能，且可進行各配對的運作方式。此外，限定元可以在屬性名稱之後以任何順序指定。

備註 – 同一屬性有多個 CoS 定義時，所有定義皆必須有相同的 *override* 與 *merge* 限定元。當 CoS 定義中有不同的成對之限定元時，會任意從所有定義選取其中一個組合。

CoS 屬性優先權

若有多個 CoS 定義或多值限定符號，但沒有 `merge-schemes` 限定元，則目錄伺服器會使用優先權屬性選取單一範本，以定義運算屬性的單一值。

`cosPriority` 屬性表示特定範本在所考慮的所有範本之間的全域優先權。優先權零是最高優先權。沒有 `cosPriority` 屬性的範本會視爲最低的優先權。當兩個以上的範本提供一個屬性值，但有相同的優先權或沒有優先權時，會隨意選擇值。

使用 `merge-schemes` 限定元時不會考量範本優先權。合併時，所有考量的範本會定義一個值，而不管範本定義的優先權。如下節中所述，會在 CoS 範本項目上定義 `cosPriority` 屬性。

備註 – `cosPriority` 屬性不得有負值。此外，間接 CoS 產生的屬性不支援優先權。請勿在間接 CoS 定義的範本項目中使用 `cosPriority`。

從指令行建立 CoS 範本項目

使用指標 CoS 或類別 CoS 時，範本項目會包含 `LDAPsubentry` 與 `cosTemplate` 物件類別。必須特別針對 CoS 定義建立此項目。使 CoS 範本項目成爲 `LDAPsubentry` 物件類別的實例，即可允許執行一般搜尋，而不受配置項目的阻礙。

間接 CoS 機制的範本爲目錄中任意的現有項目。目標不需要提前識別或提供 `LDAPsubentry` 物件類別，但是目標必須有輔助 `cosTemplate` 物件類別。只有在 CoS 評估爲產生運算屬性與其值時，才需存取間接 CoS 範本。

在所有情況下，CoS 範本項目必須包含目標項目上 CoS 所產生的屬性與值。屬性名稱指定於 CoS 定義項目的 `cosAttribute` 屬性中。

下列範例顯示產生 `postalCode` 屬性的指標 CoS 最高優先權之範本項目：

```
dn: cn=ZipTemplate,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalCode: 95054
cosPriority: 0
```

下節提供範本項目範例，以及各種 CoS 定義項目類型的範例。

指標 CoS 的範例

下列指令會建立具有 cosPointerDefinition 物件類別的指標 CoS 定義項目。此定義項目使用上節範例中陳述的 CoS 範本項目，以在 ou=People,dc=example,dc=com 樹狀結構的所有項目之間共用郵遞區號。

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=pointerCoS,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=ZipTemplate,ou=People,dc=example,dc=com
cosAttribute: postalCode
```

CoS 範本項目 (cn=ZipTemplate,ou=People,dc=example,dc=com) 會提供儲存在其 postalCode 屬性中的值，給所有位於 ou=People,dc=example,dc=com 尾碼下的項目。如果搜尋相同子樹狀結構中沒有郵遞區號的任何項目，則會看到產生的屬性值：

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
postalCode: 95054
```

間接 CoS 的範例

間接 CoS 可命名 cosIndirectSpecifier 屬性中的屬性，可找出每個目標特有的範本。間接 CoS 的範本項目可以是目錄中的任何項目，包含其他使用者項目。此間接 CoS 範例使用目標項目的 manager 屬性識別 CoS 範本項目。範本項目是經理的使用者項目。經理的使用者項目包含要產生的屬性值。在此案例中爲 departmentNumber 的值。

下列指令建立包含 cosIndirectDefinition 物件類別的間接 CoS 定義項目：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=generateDeptNum,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosIndirectDefinition
cosIndirectSpecifier: manager
cosAttribute: departmentNumber
```

接著，將 cosTemplate 物件類別增加至範本項目中，並確定這些項目會定義要產生的屬性。在此範例中，所有經理項目皆爲範本：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Carla Fuentes,ou=People,dc=example,dc=com
changetype: modify
add: objectclass
objectclass: cosTemplate
-
add: departmentNumber
departmentNumber: 318842
```

在此 CoS 中，包含 manager 屬性的目標項目 (ou=People,dc=example,dc=com 下的項目) 自動會有其經理的部門號碼。由於伺服器沒有 departmentNumber 屬性，因此會在目標項目上運算該屬性。但是，departmentNumber 屬性會以目標項目的一部分傳回。例如，如果 Babs Jensen 的經理定義爲 Carla Fuentes，其部門號碼如下所示：

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
manager: cn=Carla Fuentes,ou=People,dc=example,dc=com
departmentNumber: 318842
```

類別 CoS 的範例

此範例說明如何使用類別 CoS 產生郵寄地址。產生的值會在範本項目中指定，此範本項目由 CoS 定義中的 cosTemplateDN 與目標項目中的 cosSpecifier 屬性值之組合所找到。下列指令使用 cosClassicDefinition 物件類別建立定義項目：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=classicCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDn: ou=People,dc=example,dc=com
cosSpecifier: building
cosAttribute: postalAddress
```

使用相同的指令，建立提供每棟大樓的郵寄地址之範本項目：

```
dn: cn=B07,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalAddres: 7 Old Oak Street, Anytown, CA 95054
```

在此 CoS 中，包含 building 屬性的目標項目 (ou=People,dc=example,dc=com 下的項目) 自動會有對應的郵寄地址。CoS 機制會搜尋其 RDN 中有限定符號屬性值的範本項目。在此範例中，如果 Babs Jensen 指定給大樓 B07，會產生如下的郵寄地址：

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
building: B07
postalAddress: 7 Old Oak Street, Anytown, CA 95054
```

建立角色型屬性

您可以建立類別 CoS 機制，根據項目處理的角色來產生該項目的屬性值。例如，您可以使用角色型屬性，設定伺服器查詢限制爲逐項。

若要建立角色型屬性，請使用 nsRole 屬性做爲類別 CoS 的 CoS 定義項目內之 cosSpecifier。由於 nsRole 屬性可以爲多值，因此您可以定義有多個可能範本項目的 CoS 模式。若要解決要使用哪個範本項目的模擬兩可之情況，請於 CoS 範本項目中包含 cosPriority 屬性。

例如，您可以建立 CoS，允許經理角色的成員超過標準電子信箱配額。經理角色如下：

```
dn: cn=ManagerRole,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: ManagerRole
nsRoleFilter: (isManager=True)
Description: filtered role for managers
```

依下列方式建立類別 CoS 定義項目：

```
dn: cn=generateManagerQuota,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDn: cn=managerCOS,ou=People,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: mailboxquota override
```

CoS 範本名稱必須是 cosTemplateDn 與 nsRole 值 (角色的 DN) 的組合。例如：

```
dn:cn="cn=ManagerRole,ou=People,dc=example,dc=com",\
 cn=managerCOS,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
mailboxquota: 1000000
```

CoS 範本項目提供 `mailboxquota` 屬性值。`override` 的其他限定元會通知 CoS 覆寫目標項目中任何現有的 `mailboxquota` 屬性值。屬於角色成員的目標項目，將具有該角色與 CoS 所產生的運算屬性，例如：

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -\
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Fuentes)"
dn: cn=Carla Fuentes,ou=People,dc=example,dc=comcn: Carla Fuentes
isManager: TRUE...nsRole: cn=ManagerRole,ou=People,dc=example,dc=com
mailboxquota: 1000000
```

備註 – 角色項目與 CoS 定義項目應位於目錄樹狀結構中的相同位置，才會在其範圍中具有相同的目標項目。CoS 目標項目也應該位於相同位置，以方便尋找與維護。

監視 CoS 外掛程式

目錄伺服器可讓您監視 CoS 外掛程式的部分功能。CoS 監視屬性儲存在 `cn=monitor,cn=Class of Service,cn=plugins,cn=config` 項目中。如需此項目下每個屬性及這些屬性所提供之資訊的詳細說明，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Man Page Reference」。

設定 CoS 記錄

當目錄伺服器強制在多個適用的定義項目間做出任意的區分時，伺服器會記錄警告訊息。這類警告訊息採取此格式：

```
Definition /defDN1/ and definition /defDN2/ compete to provide attribute
 '/type/' at priority /level/
```

當目錄伺服器強制在多個可能適用的定義項目間做出任意的區分時，您也可以配置伺服器記錄參考訊息。若要執行此項作業，請將錯誤記錄設爲包含來自外掛程式的訊息。

備註 – 由於設定其他記錄層級可能造成記錄負載過大，建議不要在生產伺服器上設定記錄。

這類參考訊息的內容採取下列格式：

```
Definition /defDN1/ and definition /defDN2/ potentially compete
to provide attribute '/type/' at priority /level/
```

您可以接著選擇是否要設定在定義項目上適用的 CoS 優先權，以解決這類 CoS 模擬兩可的情形。

維護參照完整性

參照完整性是外掛程式機制，可確保維持項目之間的關係。群組成員身份類型等多種屬性類型包含其他項目的 DN。參照完整性可用以確定移除項目時，也會移除所有包含其 DN 的屬性。

例如，如果從目錄移除使用者項目並啓用參照完整性，伺服器也會從使用者爲其成員的任何群組中移除該使用者。如果未啓用參照完整性，管理員必須手動從群組移除使用者。如果您要整合目錄伺服器與其他相依於使用者與群組管理目錄的 Sun Java System 產品，這會是很重要的功能。

參照完整性的運作方式

啓用參照完整性外掛程式時，會在刪除、重新命名或移動作業之後，立即在特定屬性上執行完整性更新。預設會停用參照完整性外掛程式。

每當您在目錄中刪除、重新命名或移動使用者或群組項目時，會將此作業記錄至參照完整性記錄檔：

instance-path`/logs/referint`

在稱爲**更新間隔**的指定時間過後，伺服器會在啓用參照完整性的所有屬性上執行搜尋，並將該搜尋所產生的項目與記錄檔中已刪除或修改的項目 DN 進行比較。如果記錄檔顯示項目已經刪除，也會刪除對應的屬性。如果記錄檔顯示項目已經變更，也會據以修改對應的屬性值。

啓用參照完整性外掛程式的預設配置後，它會在發生刪除、重新命名或移動作業之後，立即對 member、uniquemember、owner、seeAlso 與 nsroledn 屬性執行完整性更新。不過，您可以配置參照完整性外掛程式的運作方式，以符合您的需求。可以配置下列運作方式：

- 在不同檔案中記錄參照完整性更新。
- 修改更新間隔。

  若要降低參照完整性更新對系統的影響，可以增加更新之間的時間間隔。
- 選取要套用參照完整性的屬性。

  如果使用或定義包含 DN 值的屬性，可能會想要參照完整性外掛程式監視這些屬性。

▼ 配置參照完整性外掛程式

備註 – 參照完整性外掛程式使用的所有資料庫內的所有屬性皆必須編製索引。這些索引必須建立於所有資料庫的配置中。啓用回溯變更記錄時，必須編製 cn=changelog 尾碼的索引。如需相關資訊，請參閱第 13 章。

部分限制與在複寫的環境中使用參照完整性外掛程式相關。如需這些限制的清單，請參閱第 242 頁的 「複寫與參照完整性」。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **請確定已配置所有複本，同時已定義所有複寫協議。**
2. **決定您要維護其參照完整性的屬性集與要在主伺服器上使用的更新間隔。**
3. **在所有主伺服器上，使用相同的屬性集與更新間隔來啓用參照完整性外掛程式。**
   - 若要定義參照完整性的屬性，請使用此指令：

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-attr:attribute-name \
      ref-integrity-attr:attribute-name
     ```
   - 若要將參照完整性屬性增加至現有的屬性清單中，請使用此指令：

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-attr+:attribute-name
     ```
   - 若要定義參照完整性的更新間隔，請使用此指令：

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-check-delay:duration
     ```
   - 若要啓用參照完整性，請使用此指令：

```
$ dsconf set-server-prop -h host -p port ref-integrity-enabled:on
```

4 確認已在所有用戶伺服器上停用參照完整性外掛程式。

◆ ◆ ◆ 第 11 章

11 目錄伺服器複寫

複寫是一套機制，讓目錄內容可利用此機制從目錄伺服器自動複製到一或多個其他的目錄伺服器。所有寫入作業都會自動鏡像到其他目錄伺服器。可在支援的不同平台之間進行複寫。如需支援平台的清單，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 版本說明」中的「平台支援」。如需複寫概念、複寫方案及如何規劃目錄部署中的複寫之完整說明，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」。

一般會在複寫拓樸中，將伺服器上的某個尾碼複寫至伺服器上的其他尾碼，或以伺服器上的其他尾碼複寫伺服器上的某個尾碼。因此，複本、複寫的尾碼與複寫的伺服器等字詞會交換使用。

本章說明使用指令行設定各種複寫方案所執行的作業，其中包含下列主題：

- 第 222 頁的「規劃複寫部署」
- 第 222 頁的「配置與管理複寫的建議介面」
- 第 222 頁的「配置複寫的步驟摘要」
- 第 224 頁的「啓用專屬用戶上的複寫」
- 第 226 頁的「啓用集散中心上的複寫」
- 第 227 頁的「啓用主伺服器複本上的複寫」
- 第 228 頁的「配置複寫管理員」
- 第 231 頁的「建立與變更複寫協議」
- 第 232 頁的「部分複寫」
- 第 233 頁的「複寫優先權」
- 第 234 頁的「初始化複本」
- 第 241 頁的「編製複寫的尾碼之索引」
- 第 241 頁的「遞增多個項目到大型複寫的尾碼」
- 第 217 頁的「維護參照完整性」
- 第 242 頁的「經由 SSL 的複寫」
- 第 244 頁的「經由 WAN 的複寫」
- 第 247 頁的「修改複寫拓樸」
- 第 252 頁的「Directory Server 6.3 之前的版本複寫」
- 第 252 頁的「使用回溯變更記錄」

- 第 255 頁的 「取得複寫狀態」
- 第 257 頁的 「解決常見複寫衝突」

規劃複寫部署

您可以使用無限部主伺服器進行複寫部署的配置。部署中不需要包含集散中心或用戶。本章包含配置集散中心與用戶的複寫程序，不過其爲可選擇的項目。

開始配置複寫之前，必須清楚瞭解貴機構中要部署複寫的方式。您必須瞭解「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中所述的複寫概念。同時還須使用「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中所提供的設計準則，謹愼規劃未來的複寫配置。

配置與管理複寫的建議介面

配置與管理複寫最簡單的方式是使用目錄服務控制中心 (DSCC)。您可以使用 DSCC 自動配置複寫。您可以選擇設定複寫拓樸所需的自動化層級，例如，是否要在複寫配置期間初始化尾碼。DSCC 也會提供檢查以避免錯誤。此外，DSCC 提供複寫拓樸的圖形化檢視。

DSCC 線上說明提供使用 DSCC 設定複寫的程序。

備註 – 只有在無法使用 DSCC 配置複寫時，才請使用本章所提供的指令行程序。

配置複寫的步驟摘要

第 222 頁的 「配置複寫的步驟摘要」假設複寫的是單一尾碼。若要複寫多個尾碼，可以平行配置每部伺服器上的尾碼。亦即可以在多個尾碼上重複配置複寫的每個步驟。

本章其餘部分包含如何配置複寫的詳細指示。

▼ 配置複寫的步驟摘要

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

若要配置複寫拓樸，請按照本程序中概要說明的一般步驟進行。

1. **在包含專屬用戶複本的所有伺服器上，執行下列作業：**

   a. **為用戶複寫的尾碼建立空的尾碼。**
   請參閱第 224 頁的 「建立用戶複本的尾碼」。

b. **啓用用戶複寫的尾碼。**
   請參閱第 225 頁的「啓用用戶複本」。

c. **(可選擇) 配置進階用戶設定。**
   請參閱第 225 頁的「執行進階用戶配置」。

2 **請視需要在包含集散複寫的尾碼之所有伺服器上，執行下列作業：**

a. **為集散複寫的尾碼建立空的尾碼。**
   請參閱第 226 頁的「建立集散複本的尾碼」。

b. **啓用集散複寫的尾碼。**
   請參閱第 226 頁的「啓用集散複本」。

c. **(可選擇) 配置進階集散中心設定。**
   請參閱第 227 頁的「修改集散複本上的變更記錄設定」。

3 **在包含主伺服器複寫的尾碼之所有伺服器上，執行下列作業：**

a. **為主伺服器複寫的尾碼建立尾碼。**
   請參閱第 227 頁的「建立主伺服器複本的尾碼」。

b. **啓用主伺服器複寫的尾碼。**
   請參閱第 227 頁的「啓用主伺服器複本」。

c. **(可選擇) 配置進階主伺服器設定。**
   請參閱第 228 頁的「修改主伺服器複本上的變更記錄設定」。

備註 – 請確定建立複寫協議之前已啓用所有複本，如此一來，建立複寫協議之後便能立即初始化用戶複本。用戶初始化一律會是設定複寫的最後一個階段。

---

4 **請確定複寫管理員配置已完成。**

- 如果計劃使用預設管理員，請在所有伺服器上設定預設複寫管理員密碼。請參閱第 230 頁的「變更預設複寫管理員密碼」。
- 如果計劃使用非預設複寫管理員，請在所有伺服器上定義替代複寫管理員項目。請參閱第 228 頁的「使用非預設複寫管理員」。

5 **依下列方式在所有主伺服器複本上建立複寫協議：**

a. **在多重主伺服器拓樸的主伺服器之間**

b. **在主伺服器與其專屬用戶之間**

c. **在主伺服器與集散複本之間**

請參閱第 231 頁的「建立與變更複寫協議」。

6 **(可選擇) 若要使用部分複寫，請立即進行配置。**
請參閱第 232 頁的「部分複寫」。

7 **(可選擇) 若要使用複寫優先權，請立即進行配置。**
請參閱第 233 頁的「複寫優先權」。

8 **配置集散複本與其用戶之間的複寫協議。**
請參閱第 231 頁的「建立與變更複寫協議」。

9 **若是多重主伺服器複寫，請從包含原始資料副本的相同主伺服器複本，初始化所有主伺服器。**
請參閱第 234 頁的「初始化複本」。

10 **初始化集散複本與用戶複本。**
請參閱第 234 頁的「初始化複本」。

啓用專屬用戶上的複寫

專屬用戶是複寫的尾碼之唯讀副本。專屬用戶會從連結做爲複寫管理員的伺服器接收更新，以進行變更。用戶伺服器的配置作業包含準備空的尾碼以保留複寫的尾碼，以及啓用該尾碼上的複寫。選擇性進階配置可包含設定參照、變更清除延遲與修改特性。

下列幾節說明如何在伺服器上配置一個專屬用戶複寫的尾碼。在將包含專屬用戶複寫的尾碼之每部伺服器上，重複所有程序。

▼ 建立用戶複本的尾碼

- **如果不存在空的尾碼，請在具有相同 DN 的用戶上建立尾碼做為預定主伺服器複本。**
如需相關指示，請參閱第 62 頁的「建立尾碼」。

注意 – 如果尾碼存在且不是空值，其內容會在從主伺服器初始化複寫的尾碼時遺失。

▼ 啓用用戶複本

建立空的尾碼之後，需要啓用用戶複寫的尾碼。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **啓用用戶複寫的尾碼。**

  ```
  $ dsconf enable-repl -h host -p port consumer suffix-DN
  ```

  例如：

  ```
  $ dsconf enable-repl -h host1 -p 1389 consumer dc=example,dc=com
  ```

▼ 執行進階用戶配置

若要爲進階功能配置用戶複寫的尾碼，請立即執行。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **若要使用 SSL 進行參照，請設定安全參照。**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN referral-url:ldaps://servername:port
   ```

   例如：

   ```
   $ dsconf set-suffix-prop -h host1 -p 1389 dc=example,dc=com \
    referral-url:ldaps://server2:2389
   ```

   複寫機制會自動配置用戶傳回複寫拓樸中所有已知主伺服器的參照。這些預設參照假設用戶端將在標準連線上使用簡單認證。若要讓用戶端可使用 SSL 安全連線連結至主伺服器，請增加使用安全**連接埠**號碼，同時格式爲 `ldaps://servername:port` 的參照。請注意，如果主伺服器配置爲僅能使用安全連線，URL 預設會指向安全連接埠。

   如果增加多個 LDAP URL 做爲參照，可以強制用戶僅傳送這些 LDAP URL 的參照，而不會傳送主伺服器複本的參照。例如，假設您要用戶端一律參照主伺服器上的安全連接埠，而不是預設連接埠。請建立這些安全連接埠的 LDAP URL 清單，並設定特性以使用這些參照。若要定義特定主伺服器或目錄伺服器代理處理所有更新，也可以使用專用參照。

2. **若要變更用戶的複寫清除延遲，請使用此指令：**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN repl-purge-delay:time
   ```

   例如，若要將清除延遲設爲 2 天，請鍵入：

   ```
   $ dsconf set-suffix-prop -h host1 -p 1389 edc=example,dc=com repl-purge-delay:2d
   ```

用戶伺服器會儲存關於複寫的尾碼內容更新之內部資訊，而清除延遲參數則會指定必須保留此資訊的時間。清除延遲會部分決定用戶與其主伺服器之間的複寫可於中斷多久後仍能正常回復。這與供應者伺服器上變更記錄的 MaxAge 參數相關。這兩個參數之間較短者會決定兩部伺服器之間的複寫可以停用或中斷但仍能正常回復之最長時間。在大多數情況下，預設值 7 天便已足夠。

啓用集散中心上的複寫

集散複本會做爲用戶與主伺服器，進一步分佈複寫的資料給更多的用戶。集散複本從其供應者接收複寫更新，再傳送複寫更新給其用戶。這些複本不接受修改，而會傳回參照至主伺服器。

集散伺服器的配置作業包含準備空的尾碼以保留複寫的尾碼，以及啓用該尾碼上的複寫。選擇性進階配置可包含選擇不同的複寫管理員、設定參照、設定清除延遲，以及修改變更記錄參數。

下列幾節說明如何配置一部集散伺服器。在將包含集散複寫的尾碼之每部伺服器上，重複所有程序。

▼ 建立集散複本的尾碼

- **如果不存在空的尾碼，請在具有相同 DN 的集散伺服器上建立尾碼做為預定主伺服器複本。**

  如需相關指示，請參閱第 62 頁的 「建立尾碼」。

  如果尾碼存在且不是空值，其內容會在從主伺服器初始化複寫的尾碼時遺失。

▼ 啓用集散複本

如果有集散複本，請立即啓用。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **啓用集散複寫的尾碼。**

  ```
  $ dsconf enable-repl -h host -p port hub suffix-DN
  ```

  例如：

  ```
  $ dsconf enable-repl -h host1 -p 1389 hub dc=example,dc=com
  ```

▼ 修改集散複本上的變更記錄設定

您可能會想爲進階集散配置修改的唯一參數，會與該變更記錄相關。集散伺服器需要變更記錄才能做爲供應者。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **若要修改集散中心上的變更記錄設定，請使用下列指令之一：**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN repl-cl-max-age:value

$ dsconf set-suffix-prop -h host -p port suffix-DN repl-cl-max-entry-count:value
```

啓用主伺服器複本上的複寫

主伺服器複本包含主伺服器的資料副本，並會集中所有修改，再傳播更新給所有其他複本。主伺服器會記錄所有變更、檢查其用戶的狀態，並視需要傳送更新給用戶。在多重主伺服器複寫中，主伺服器複本也會從其他主伺服器接收更新。

主伺服器的配置作業包含定義內含主伺服器複本的尾碼、啓用主伺服器複本，以及視需要配置複本以供進階複寫使用。

下列幾節說明如何配置一部主伺服器。在將包含主伺服器複寫的尾碼之每部伺服器上，重複所有程序。

▼ 建立主伺服器複本的尾碼

- **在將包含要複寫的項目之主伺服器上，選擇或建立尾碼。**

  如需相關指示，請參閱第 62 頁的 「建立尾碼」。

  若要確定多重主伺服器的配置與初始化正確，請僅載入一部含資料的主伺服器。其他複寫的尾碼之任何資料將會遭覆寫。

▼ 啓用主伺服器複本

啓用主伺服器上的複寫時，必須指定複寫 ID。複寫 ID 會用以辨別更新陳述式的所有者，以及解決多重主伺服器複寫可能發生的衝突。因此，複寫 ID 對此尾碼的所有主伺服器複本而言必須是唯一的。複寫 ID 一旦設定便不得變更。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **啓用主伺服器複寫的尾碼。**

  ```
  $ dsconf enable-repl -h host -p port -d ReplicaID master suffix-DN
  ```

  其中 *ReplicaID* 是介於 1 至 65534 之間的整數。

  例如，若要建立複本 ID 爲 1 的主伺服器複寫的尾碼，請使用此指令：

  ```
  $ dsconf enable-repl -h host1 -p 1389 -d 1 master dc=example,dc=com
  ```

▼ 修改主伺服器複本上的變更記錄設定

若是進階主伺服器配置，您可能會想要修改變更記錄設定。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **若要修改主伺服器上的變更記錄設定，請使用下列指令之一：**

  ```
  $ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-age:value
  ```

  ```
  $ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-entry-count:value
  ```

配置複寫管理員

本節說明如何配置非預設複寫管理員，以及如何設定預設複寫管理員密碼。

使用非預設複寫管理員

複寫管理員是供應者傳送複寫更新時，將用來連結至用戶伺服器的使用者。包含接收更新之尾碼的所有伺服器，至少必須有一個複寫管理員項目。

目錄伺服器有一個預設複寫管理員項目可用於每部伺服器上使用，特別是針對簡單複寫方案：`cn=replication manager,cn=replication,cn=config`。複寫機制會自動以此使用者配置用戶複本，以簡化複本的部署。

如果有更複雜的複寫方案，針對每個複寫的尾碼可能需要數個具有不同密碼的複寫管理員。您可利用一或多個新的複寫管理員取代現有的預設複寫管理員。

注意 – 請勿在使用複寫管理員的 DN 與密碼之伺服器上連結或執行作業。複寫管理員僅供複寫機制使用。其他任何用途均可能需要重新初始化複本。

請勿使用目錄管理員做爲複寫管理員。由於 cn=admin,cn=Administrators,cn=config 項目會供其他管理作業使用，您也不得使用此使用者或管理群組中的任何其他使用者做爲複寫管理員。

爲各個用戶選擇複寫管理員之後，請確實記下所選擇或建立的複寫管理員 DN。稍後以此供應者的用戶建立複寫協議時，會需要此 DN 與其密碼。

▼ 設定非預設複寫管理員

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 在所有用戶 (目標) 複寫的尾碼上，建立新的複寫管理員與密碼。

```
$ ldapmodify -a -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn:"cn=new-replication-manager,cn=replication,cn=config"
objectclass: top
objectclass: person
userpassword:password
sn:new-replication-manager
```

例如：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn:"cn=ReplicationManager3,cn=replication,cn=config"
objectclass: top
objectclass: person
userpassword:secret
sn:ReplicationManager3
```

2 在所有用戶 (目標) 複寫的尾碼上，設定複寫管理員連結 DN。

```
$ dsconf set-suffix-prop -h host -p port suffix-DN \
 repl-manager-bind-dn:"cn=new-replication-manager,cn=replication,cn=config"
```

例如：

```
$ dsconf set-suffix-prop -h host1 -p 1389 dc=example,dc=com \
 repl-manager-bind-dn:"cn=ReplicationManager3,cn=replication,cn=config"
```

3 **針對已在所有供應者 (來源) 複寫的尾碼上建立的所有複寫協議，設定複寫管理員連結 DN。**

a. 建立暫存檔以設定新的複寫管理員密碼。

此檔案只能讀取一次，您必須儲存密碼以供日後使用。

```
$ echo password > password-file
```

b. 設定複寫管理員連結 DN 與密碼，供複寫機制執行更新時使用。

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN host:port \
 auth-bind-dn:"cn=new-replication-manager,cn=replication,cn=config" \
 auth-pwd-file:password-file
```

例如：

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 auth-bind-dn:"cn=ReplicationManager3,cn=replication,cn=config" \
 auth-pwd-file:pwd.txt
```

c. 移除密碼暫存檔。

```
$ rm password-file
```

▼ 變更預設複寫管理員密碼

1 **建立暫存檔以設定複寫管理員密碼。**

此檔案只能讀取一次，您必須儲存密碼以供日後使用。

```
$ echo password > password-file
```

2 **在複寫拓樸中的所有用戶 (目標) 伺服器上，設定複寫管理員連結密碼。**

```
$ dsconf set-server-prop -h host -p port def-repl-manager-pwd-file:password-file
```

例如：

```
$ dsconf set-server-prop -h host1 -p 1389 def-repl-manager-pwd-file:pwd.txt
```

3 **移除密碼暫存檔。**

```
$ rm password-file
```

建立與變更複寫協議

複寫協議是供應者上的一組參數，其可配置並控制傳送更新給指定用戶的方式。複寫協議必須建立於傳送更新給其用戶之供應者複寫的尾碼上。您必須爲每個要更新的用戶，在供應者上建立複寫協議。

▼ 建立複寫協議

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

如果使用 DSCC 建立新的複寫協議，可以選擇從現有的複寫協議複製部分或所有的複寫協議配置設定。

1 **為要進行複寫的各個用戶，從主伺服器建立複寫協議。**

```
$ dsconf create-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port [consumer-host:consumer-port]
```

例如：

```
$ dsconf create-repl-agmt -h host1 -p 1389 dc=example,dc=com host2:1389
```

若要使用指令行列出現有的複寫協議，請使用 `dsconf list-repl-agmts` 指令。

備註 – 如果在複寫執行時變更主伺服器上的連接埠號碼，則無須重新初始化伺服器。但是，指向舊位址 (*host*: *old-port*) 的舊複寫協議將再也無法使用。若要複寫在變更連接埠號碼之前如同往常繼續執行，則必須以新位址 (*host*:*new-port*) 建立新協議。

---

2 **檢查複寫協議是否已正確建立。**

```
$ dsconf show-repl-agmt-status -h host -p port suffix-DN consumer-host:consumer-port
```

3 **如果認證狀態不正確，請執行** `dsconf accord-repl-agmt` **指令。**

備註 – 請只有在使用預設複寫管理員時才使用指令 `dsconf accord-repl-agmt`。如果已建立新的複寫管理員，請勿使用此指令，這麼做會覆寫部分必要的設定。

`dsconf accord-repl-agmt` 指令可確保供應者與目標伺服器皆共用相同的複寫認證設定。

```
$ dsconf accord-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

例如：

```
$ dsconf accord-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

▼ 變更複寫協議的目標

本程序會變更現有複寫協議所指向的遠端複本。現有協議的尾碼 DN 與配置維持不變。

- **在複寫協議中變更遠端複本的主機名稱與連接埠號碼。**

  ```
  $ dsconf change-repl-dest -h host -p port suffix-DN host:port new-host:new-port
  ```

  如果此指令透過 `-A` *protocol* 選項執行，可以變更複寫所使用的認證協定。

部分複寫

複寫作業預設會將複寫的尾碼中整個項目複製到用戶複本。您可以使用部分複寫功能，選取要使用的尾碼以及要包含或排除的屬性。部分複寫會配置於複寫協議中，讓您可爲主伺服器各用戶複寫的尾碼定義屬性集。您可以控制要分佈的資料，並更有效地使用複寫頻寬與用戶資源。

例如，若要減少複寫頻寬，可以選擇不要複寫含有 `photo`、`jpegPhoto` 與 `audio` 等一般說來過大的值之屬性。因此，用戶上將無法使用這些屬性。在另外一個情況下，您可能會選擇在專門執行認證的用戶伺服器上，僅複寫 `uid` 與 `userpassword` 屬性。

部分複寫的注意事項

備註 – 部分複寫無法在 Directory Server 5.2 之前的產品版本中使用。配置部分複寫協議時，主伺服器與用戶複本至少必須使用 Directory Server 5.2。

啓用或修改部分屬性集需要重新初始化用戶複本。因此，您必須在部署之前決定部分複寫需求，並在第一次初始化複寫的尾碼之前定義屬性集。

複寫小型的屬性集時，由於考慮到特定屬性上 ACI、角色與 CoS 等複雜功能的相依性，因此您必須謹慎進行。此外，不複寫 ACI、角色或 CoS 機制的限定符號或篩選中所提及的其他屬性，可能會危及資料的安全性。不複寫可能也會導致搜尋中傳回不同的屬性集。管理要排除的屬性清單比管理要包含的屬性清單要來得安全，且不容易有人爲疏失。

如果複寫的屬性集不允許所有複寫的項目遵守模式，則必須關閉用戶伺服器上的模式檢查。由於複寫機制會略過用戶上的模式檢查，因此複寫不相符的項目不會導致錯誤。但是，用戶將包含這些不相符的項目，並會關閉模式檢查以向其用戶端顯示一致的狀態。

部分複寫會配置於包含集散中心與專屬用戶的主伺服器複本之複寫協議中。不支援在多重主伺服器複寫環境中配置兩個主伺服器複本之間的部分複寫。此外，如果多個主伺服器有含相同複本的複寫協議，則所有協議皆須複寫相同的屬性集。

▼ 配置部分複寫

若要配置部分複寫，必須指定尾碼、決定包含或排除該尾碼上的屬性，並接著選擇要包含或排除的屬性。如果選擇在尾碼上排除屬性，將會自動包含所有其他屬性。同理，如果選擇在尾碼上包含特定屬性，將會自動排除所有其他屬性。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **在位於來源伺服器的複寫協議上配置部分複寫。**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port property:value
```

其中 *property* 是 `repl-fractional-exclude-attr` 或 `repl-fractional-include-attr`。

例如，若要配置部分協議排除 JPEG 與 TIFF 圖片在尾碼 `dc=example,dc=com` 上進行複寫，請使用此指令：

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389
 repl-fractional-exclude-attr:jpegPhoto repl-fractional-exclude-attr:tiffPhoto
```

若要在應排除的現有屬性清單中增加一個屬性，請使用此指令：

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port repl-fractional-exclude-attr+:
attribute
```

複寫優先權

指定複寫優先權爲可選擇的項目。您可以建立複寫規則以指定某些變更 (例如更新使用者密碼) 以高優先權進行複寫。複寫規則中指定的任何變更皆會以高優先權進行複寫，而所有其他變更則會以一般優先權進行複寫。

備註 – 複寫優先權規則僅需要建立於主伺服器上。不需要爲集散中心與用戶進行配置。

▼ 配置複寫優先權

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **若要在主伺服器上建立新的複寫優先權規則，請使用此指令：**

```
$ dsconf create-repl-priority -h host -p port suffix-DN priority-name property:value
```

您可以使用下列一或多個特性設定複寫優先權：

- 作業類型，`op-type`
- 連結 DN，`bind-dn`
- 基底 DN，`base-dn`
- 屬性類型，`attr`

priority-name 由使用者定義。

例如，若要建立複寫規則，指定以高優先權複寫使用者密碼變更，請使用此指令：

```
$ dsconf create-repl-priority -h host2 -p 1389 dc=example,dc=com pw-rule \
 attr:userPassword
```

若要顯示目前的複寫規則，請使用 `dsconf list-repl-priorities -v` 指令。此指令搭配 `-v` 選項使用時，會顯示其他與複寫優先權規則相關的資訊。

```
$ dsconf list-repl-priorities -h host2 -p 1389 -v
```

如需更多資訊，請參閱 `dsconf`(1M) 線上手冊。

初始化複本

建立複寫協議並配置兩個複本之後，必須初始化用戶複寫的尾碼，才會開始複寫。您可以在初始化期間，實際將資料從供應者複寫的尾碼複製到用戶複寫的尾碼。

此外，部分錯誤情況或配置變更會需要重新初始化複本。例如，如果因爲任何理由從備份復原單一主伺服器複寫的尾碼中之資料，則必須重新初始化其所更新的所有複本。

重新初始化時，會刪除用戶上複寫的尾碼之內容，並以主伺服器上的尾碼內容取代。如此做可確保複本會進行同步化，且複寫更新可以繼續進行。本節中所述的所有初始化方法會自動重建用戶複本的索引，使得用戶能以最佳方式回應用戶端的讀取請求。

使用多重主伺服器複寫時，如果拓樸中有其他主伺服器已更新用戶，則用戶可能無須重新初始化。

▼ 從遠端 (供應者) 伺服器初始化複寫的尾碼

您可以使用現有的複寫協議，從遠端伺服器初始化尾碼。此初始化方法比其他方法簡單，因此請儘可能使用此方法。而僅在有大量資料使得匯入耗費太多時間時使用其他方法。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

使用 DSCC 以線上方式初始化複寫的尾碼，是初始化或重新初始化用戶的簡單方式。但是，如果初始化大量的項目，此程序可能很耗時。此時，使用指令行以離線方式初始化用戶可能比較有效率。

1 **初始化複本。**

```
$ dsconf init-repl-dest -h host -p port suffix-DN destination-host:destination-port [destination-host:destination-port]
```

其中 *destination-host*:*destination-port* 是您從遠端伺服器初始化目標伺服器的主機與連接埠。

2 **(可選擇) 請為各個協議檢查尾碼是否已初始化。**

```
$ dsconf show-repl-agmt-status -h host -p port suffix-DN destination-host:destination-port
```

從 LDIF 初始化複本

▼ 從 LDIF 初始化複寫的尾碼

本程序概要說明從 LDIF 檔案初始化複寫的尾碼所用之一般步驟。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

使用 DSCC 以線上方式初始化複寫的尾碼，是初始化或重新初始化用戶的簡單方式。但是，如果初始化大量的項目，此程序可能很耗時。此時，使用指令行以離線方式初始化用戶可能比較有效率。

1 **請確定已設定複寫協議。**

您必須在初始化複本**之前**執行此項作業。

2 **從主伺服器複寫的尾碼匯出尾碼資料的原始副本至 LDIF 檔案。**

請參閱第 236 頁的 「匯出複寫的尾碼至 LDIF」。

您可以在多重主伺服器複寫環境中，使用從原始主伺服器匯出的 LDIF 檔案同時初始化其他主伺服器與任何用戶。您可以在串聯複寫環境中，使用相同的檔案同時初始化集散複本及其用戶。

在所有的情況下，皆須以從配置的主伺服器複本匯出之 LDIF 檔案開始。您無法使用任意的 LDIF 檔案初始化所有複本，因爲該檔案可能不包含複寫中介資料。

3 **如果初始化部分複本，請篩選檔案而僅保留複寫的屬性，再將該檔案傳輸到所有用戶伺服器。**

請參閱第 236 頁的 「爲部分複寫篩選 LDIF 檔案」。

4 **初始化複本。**

請執行下列其中一項動作：

- 若要在離線的 (停止的) 伺服器上快速進行初使化，請使用 `dsadm import` 指令。

  ```
  $ dsadm import instance-path LDIF_file suffix-DN
  ```

- 若要從 LDIF 檔案以線上方式初始化複本，請使用 `dsconf import` 指令。

  ```
  $ dsconf import -h host -p port LDIF_file suffix-DN
  ```

  使用 `dsconf import` 會比使用 `dsadm import` 還要慢，但是您無須在執行匯入作業期間停止伺服器。

如需初始化尾碼的詳細資訊與範例，請參閱第 193 頁的「初始化尾碼」。如需詳細的指令用法，請參閱 `dsadm`(1M) 線上手冊與 `dsconf`(1M) 線上手冊。

5 **(可選擇) 請為各個協議檢查尾碼是否已初始化。**

```
$ dsconf show-repl-agmt-status -h host -p port suffix-DN destination-host:destination-port
```

▼ 匯出複寫的尾碼至 LDIF

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

- **使用下列指令之一匯出 LDIF 檔案中的複寫尾碼內容：**

  - 若是以離線方式匯出，請鍵入：

    ```
    $ dsadm export instance-path suffix-DN LDIF_file
    ```

  - 若是以線上方式匯出，請鍵入：

    ```
    $ dsconf export -h host -p port suffix-DN LDIF_file
    ```

  下列範例將匯出整個 `dc=example,dc=com` 複寫的尾碼與複寫資訊至檔案 `example_replica_export.ldif`：

  ```
  $ dsconf export -h host2 -p 1389 dc=example,dc=com  \
   /local/ds/ldif/example_export_replica.ldif
  ```

  如需更多資訊，請參閱第 190 頁的「備份至 LDIF」、`dsadm`(1M) 線上手冊與 `dsconf`(1M) 線上手冊。

為部分複寫篩選 LDIF 檔案

使用 DSCC 時，不會感覺到正在初始化配置有部分複寫的複本。初始化期間僅會將選取的屬性傳送至用戶。

如果已配置部分複寫，應剔除所有未使用的屬性，再將匯出的 LDIF 檔案複製到用戶伺服器。目錄伺服器爲此用途提供有 `fildif` 工具。此工具會篩選指定的 LDIF 檔案，僅保留複寫協議中定義之屬性集所允許的屬性。

此工具會讀取伺服器的配置，以決定屬性集定義。若要讀取配置檔案，必須以超級使用者身份或以擁有程序與檔案 (由 `nsslapd-localuser` 屬性所指定) 的使用者身份執行 `fildif` 工具。例如，下列指令會篩選上一範例內 `dc=example,dc=com` 尾碼所匯出的檔案：

```
$ fildif -i /local/ds1/ldif/example_master.ldif \
 -o /local/ds1/ldif/filtered.ldif -b "cn=host2.example.com:1389, \
 cn=replica,cn=\\"dc=example,dc=com\\",cn=mapping tree,cn=config" -p /local/ds1
```

如需瞭解 `fildif` 指令的位置，請參閱第 28 頁的「指令位置」。

`-i` 與 `-o` 選項分別是輸入與輸出檔案。`-b` 選項是定義部分複寫的複寫協議之 DN。您可以使用此指令尋找此 DN：

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b "cn=config" "(&(objectclass=nsds5replicationagreement) (nsDS5ReplicaPort=replica-port) \
 (nsDS5ReplicaHost=replica-host))" dn
```

例如：

```
$ ldapsearch -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "cn=config" "(&(objectclass=nsds5replicationagreement) \
 (nsDS5ReplicaPort=2090)(nsDS5ReplicaHost=host2))" dn
Enter bind password:
version: 1
dn: cn=host2:1389,cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
```

如需 `fildif` 工具完整的指令行語法，請參閱 `fildif`(1) 線上手冊。

您接著可以使用 `fildif` 產生的 `filtered.ldif` 檔案，初始化此複寫協議中的用戶。如第 192 頁的「從 LDIF 檔案匯入資料」中所述，傳輸檔案至用戶伺服器並匯入檔案。

使用二進位副本初始化複寫的尾碼

二進位副本可讓您使用某部伺服器的二進位備份檔案，復原相同的目錄內容到另一部伺服器上，以複製整部伺服器。您可以使用二進位副本初始化，或從主伺服器或集散伺服器的二進位副本重新初始化任何伺服器，或從其他用戶伺服器的二進位副本重新初始化用戶。

備註 – 此進階程序會與目錄伺服器的資料庫檔案互動，且應僅由有經驗的管理員使用。

如果複本有大型資料庫檔案，例如包含百萬條項目的複本，在此功能上設定某些限制有助執行與節省時間。

使用二進位副本的複寫限制

由於二進位副本會將資料庫檔案從一部機器移動到另一部，該機制會遵守下列嚴格限制：

- 兩部機器必須執行相同的作業系統，包含所有服務軟體或修補程式。
- 兩部機器必須共用相同的處理器架構。例如，您可以在兩部 UltraSPARC® T1 處理器之間執行二進位副本，但不得在 UltraSPARC T1 與 AMD Opteron 處理器之間執行二進位副本。
- 兩部機器必須是大尾數法或小尾數法。
- 兩部機器必須以相同方式對映記憶體。例如，您可以在兩部 64 位元系統的伺服器實例之間執行二進位副本，但不得在 32 位元系統的伺服器實例與 64 位元系統的伺服器實例之間執行二進位副本。
- 兩部機器必須安裝相同版本的目錄伺服器，包含二進位格式 (32 位元或 64 位元)、服務軟體與修補程式層級。
- 兩部伺服器必須有分到相同尾碼的相同目錄樹狀結構。**所有**尾碼的資料庫檔案**必須**同時複製。無法複製個別尾碼。
- 兩部伺服器上必須爲各尾碼配置相同的索引，包含 VLV (虛擬清單檢視) 索引。這些尾碼的資料庫必須有相同的名稱。
- 各伺服器必須有配置爲複本的相同尾碼。
- 如果配置部分複寫，則必須在所有伺服器上進行相同的配置。
- 兩部伺服器皆不得使用屬性加密。
- 如果啓用屬性值唯一性外掛程式，必須在兩部伺服器上有相同的外掛程式配置，且必須如下列程序所述在新的副本上重新配置外掛程式。

  下列程序說明執行二進位副本的替代方式：不需要停止伺服器的二進位副本，以及使用最低磁碟空間量的二進位副本。

建立初始化伺服器的二進位副本

本節說明如何建立初始化伺服器的二進位副本，以及如何建立使用最低磁碟空間的二進位副本。

▼ 建立初始化伺服器的二進位副本

本程序可用以執行二進位副本，以初始化複寫的伺服器，因爲其使用標準備份功能建立伺服器資料庫檔案的副本。執行標準備份可確保所有資料庫檔案皆處於一致的狀態，而不需要停止伺服器。

本程序有幾點限制。備份與復原作業會在相同機器上建立資料庫檔案的副本，因此會加倍各機器上這些檔案所需的磁碟空間量。此外，如果目錄包含十億位元組的資料，這些檔案上的實際複製作業可能需要相當長的時間。

針對此程序的某些部分，您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。此程序的其他部分只能使用指令行完成。

1. **為新複寫的尾碼在目標機器上安裝目錄伺服器，視需要建立新的伺服器實例，並根據第 238 頁的「使用二進位副本的複寫限制」配置伺服器。**

2. **在包含此複寫的尾碼之複寫拓樸中建立所有複寫協議。**

   在此複本中包含來自供應者的協議。如果此複本不是專屬用戶，請在其用戶中包含來自此複本的協議。請參閱第 231 頁的「建立與變更複寫協議」。

3. **選取完整配置與想要初始化的相同類型 (主伺服器、集散中心或用戶) 之初始化複本，並根據第 187 頁的「二進位備份」在複本上執行標準備份。**

4. **例如，使用 `ftp` 指令從備份目錄複製或傳輸檔案到目標機器上的目錄。**

5. **如果已在多重主伺服器複寫方案中初始化新主伺服器，請遵循第 197 頁的「復原多重主伺服器方案中的主伺服器」中的程序。**

▼ 使用二進位副本初始化使用最低磁碟空間的伺服器

本程序不會建立資料庫檔案的備份副本，因此會使用較少的磁碟空間與較少的時間。但是，您需要停止正在複製到排序中的伺服器，以確保資料庫檔案的狀態一致。

注意 – 本程序**不得**用以重新初始化已在多重主伺服器複寫方案中使用的主伺服器，僅能用以重新初始化用戶伺服器或初始化新的主伺服器。若要重新初始化現有的主伺服器複本，請使用線上初始化，匯入 LDIF 檔案，或遵循第 238 頁的「建立初始化伺服器的二進位副本」中的程序。

針對此程序的某些部分，您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。此程序的其他部分只能使用指令行完成。

1. **為新複寫的尾碼在目標機器上安裝目錄伺服器，視需要建立新的伺服器實例，並根據第 238 頁的「使用二進位副本的複寫限制」配置伺服器。**

2. **在包含此複本的複寫拓樸中建立所有複寫協議。**

   在此複本中包含來自供應者的協議。如果此複本不是專屬用戶，請在其用戶中包含來自此複本的協議。請參閱第 231 頁的「建立與變更複寫協議」。

3. **如第 61 頁的「啓動、停止與重新啓動目錄伺服器實例」中所述，停止要初始化或重新初始化的目標伺服器。**

4. **選取完整配置與想要初始化的相同類型 (主伺服器、集散中心或用戶) 初始化的複本，同時停止此伺服器。**

   如果複製的是多重主伺服器配置中的主伺服器複本，請確保已完整更新所有其他主伺服器最近的變更，再停止伺服器。

5. **從目標伺服器移除所有資料庫檔案，包含作業事件記錄、變更記錄與區域檔案 (`__db.`*xxx* 檔案)。**

   除非已遷移檔案，否則資料庫檔案與作業事件記錄會位於 *instance-path*`/db` 目錄中。

6. **例如，使用 `ftp` 指令從來源複本機器複製或傳輸所有資料庫檔案 (包含作業事件記錄與變更記錄) 到目標機器。**

   除非已遷移檔案，否則資料庫檔案與作業事件記錄會位於 *instance-path*`/db` 目錄中。

   如果初始化的是主伺服器或集散複本，也須複製變更記錄中的所有檔案，變更記錄預設會位於 *instance-path*`/changelog` 中。

7. **重新啓動來源與目標伺服器。**

在串聯複寫中初始化複本

串聯複寫時，請一律依下列程序中所顯示的順序初始化複本。

▼ 在串聯複寫中初始化複本

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1. **如果也有多重主伺服器複寫，請確保其中一部主伺服器有要複寫的完整資料集，再使用此主伺服器初始化其他主伺服器的複本。**

2. **從第一層集散複本的主伺服器複本初始化複本。**

3. **如果有多層集散中心，請從之前初始化的集散層級初始化每個層級。**

4. **從最後一層集散複本初始化專屬用戶上的複本。**

編製複寫的尾碼之索引

索引不會從一部伺服器實例自動複寫到另一部伺服器實例。若要爲所有保留複寫的尾碼之伺服器實例編製其屬性的索引，請執行下列動作之一。

- 以 DSCC 中伺服器群組的方式，管理保留複寫的尾碼之所有伺服器實例。將索引增加至群組中一部伺服器，再使用「複製伺服器配置」動作將索引設定複製到群組中的其他伺服器。

  如需有關 DSCC 的更多資訊，請參閱第 47 頁的 「目錄服務控制中心介面」。
- 如第 13 章中所述，使用 `dsconf` 指令管理每部伺服器實例上的索引。
- 如第 237 頁的 「使用二進位副本初始化複寫的尾碼」中所述，使用二進位副本初始化尾碼。

遞增多個項目到大型複寫的尾碼

如果您的目錄已有大量項目，而還想再增加大量項目，請勿使用 `ldapmodify -a`，這麼做會很耗時。請改搭配在複寫的拓樸中增加項目之選項，使用 `dsconf import` 指令遞增新項目。當您匯入項目時，會產生包含增加項目與複寫中介資料的 LDIF 檔案。您可以接著匯入此產生的 LDIF 檔案至其他複本。產生的 LDIF 檔案可確保在您增加資料的複本之間持續同步化複寫。

▼ 將多個項目增加至大型複寫的尾碼

開始之前 本程序會產生大型 LDIF 檔案。在執行第一個 `dsconf import` 指令之前，請確定有足夠的磁碟空間可供產生的 LDIF 檔案使用。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

注意 – 您可以使用本程序以分次傳遞方式初始化有大量項目的伺服器。但是，如果其中一個匯入失敗，便會遺失整個資料庫。請務必於每次匯入之前備份資料。

1. **匯入項目到任何主伺服器複本。**

   ```
   $ dsconf import -h host -p port -K generated-LDIF-file suffix-DN
   ```

   `-K` 選項可確保不會移除現有的資料，也會產生包含複寫程序所需的新項目與資訊之檔案 *generated-LDIF-file*。

2. **在所有其他複本中，匯入上一個步驟所產生的檔案。**

   ```
   $ dsconf import -h host -p port \
   -K -f incremental-output=no generated-LDIF-file suffix-DN
   ```

   選項 `-f incremental-output=no` 會指定不會產生其他 LDIF 檔案。本程序僅需要一個產生的 LDIF 檔案。

複寫與參照完整性

如果您搭配複寫使用參照完整性外掛程式，您必須在所有主伺服器上啓用外掛程式。您不需要在集散伺服器或用戶伺服器上啓用外掛程式。

下列限制與使用複寫環境中的參照完整性外掛程式相關：

- 必須在所有包含主伺服器複本的伺服器上啓用外掛程式。
- 必須在每部主伺服器上以相同配置啓用外掛程式。
- 不需要在僅包含集散複本或用戶複本的伺服器上啓用外掛程式。

如需有關配置參照完整性外掛程式的資訊，請參閱第 218 頁的「配置參照完整性外掛程式」。

經由 SSL 的複寫

您可以配置複寫中所包含的目錄伺服器，以經由 SSL 連線執行所有複寫作業。

▼ 配置 SSL 的複寫作業

本程序顯示在複寫拓樸中設定兩部主伺服器的複寫之指令範例。

備註 – 此範例顯示使用自行簽署的憑證之簡單複寫配置。在生產環境中設定經由 SSL 的複寫時，如果改用憑證授權機構信任的憑證會更安全。

如果供應者伺服器憑證爲僅用於 SSL 伺服器的憑證，而無法在 SSL 訊號交換期間做爲用戶端，則經由 SSL 的複寫會失敗。

經由 SSL 保護複寫時，複寫管理員的認證仍會使用簡單連結與密碼完成。您可以使用以用戶端爲基礎的認證來完整保護複寫，但是這麼做需要更複雜的設定。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1. **建立並啓動新的伺服器。**

```
$ dsadm create -p 1389 -P 1636 /local/ds1
$ dsadm create -p 2389 -P 2636 /local/ds2

$ dsadm start /local/ds1
$ dsadm start /local/ds2
```

2. **在所有伺服器上建立空的尾碼。**

```
$ dsconf create-suffix -e -i -w password-file -p 1389 dc=example,dc=com
$ dsconf create-suffix -e -i -w password-file -p 2389 dc=example,dc=com
```

3 在所有伺服器上設定多重主伺服器密碼檔案。

```
$ dsconf set-server-prop -e -i -w password-file -h example1.server -p 1389 \
 def-repl-manager-pwd-file:/local/ds1/replmanrpwd1.txt
$ dsconf set-server-prop -e -i -w password-file -h example2.server -p 2389 \
 def-repl-manager-pwd-file:/local/ds1/replmanrpwd2.txt
```

4 在所有伺服器上啓用複寫。

```
$ dsconf enable-repl -h example1.server -p 1389 -e -i -w password-file -d 1 master dc=example,dc=com
$ dsconf enable-repl -h example2.server -p 2389 -e -i -w password-file -d 2 master dc=example,dc=com
```

5 在所有伺服器上檢視現有的預設憑證。

```
$ dsadm show-cert -F der -o certfile1 /local/ds1 defaultCert
$ dsadm show-cert -F der -o certfile2 /local/ds2 defaultCert
```

6 從所有其他伺服器將 CA 信任的憑證增加至所有伺服器上。

```
$ dsadm add-cert --ca /local/ds1 "ds2 Repl Manager Cert" certfile2
$ dsadm add-cert --ca /local/ds2 "ds1 Repl Manager Cert" certfile1
```

7 在所有主伺服器與集散 (來源) 伺服器上，建立所有用戶 (目標) 伺服器的複寫協議。

請注意，複寫協議使用 LDAP 安全連接埠。

```
$ dsconf create-repl-agmt -h example1.server -p 1389 -e -i -w password-file\
 --auth-protocol "ssl-simple" dc=example,dc=com example2.server:2636
$ dsconf create-repl-agmt -h example2.server -p 2389 -e -i -w password-file\
 --auth-protocol "ssl-simple" dc=example,dc=com example1.server:1636
```

8 請針對所有複寫協議，在複寫協議中將認證密碼檔案配置為用戶 (目標) 伺服器的複寫管理員密碼檔案。

```
$ dsconf set-repl-agmt-prop -h example1.server -p 1389 -e -i -w password-file\
 dc=example,dc=com example2.server:2636 auth-pwd-file:/local/ds1/replmanrpwd2.txt
$ dsconf set-repl-agmt-prop -h example2.server -p 2389 -e -i -w password-file\
 dc=example,dc=com example1.server:1636 auth-pwd-file:/local/ds1/replmanrpwd1.txt
```

如果選擇該選項，在初始化尾碼之後，供應者會將所有複寫更新訊息經由 SSL 傳送給用戶，並將使用憑證。用戶初始化如果使用爲 SSL 配置的協議透過 DSCC 執行，也將使用安全連線。

9 請在所有伺服器上重新啓動伺服器，以使配置變更生效。

```
$ dsadm restart /local/ds1
$ dsadm restart /local/ds2
```

10 在其中一部主伺服器上初始化尾碼。

```
$ dsconf import -h example1.server -p 1389 -e -i \
-w password-file /tmp/Example.ldif dc=example,dc=com
```

11 **在尚未初始化的所有伺服器上，使用複寫協議初始化伺服器。**

```
$ dsconf init-repl-dest -e -i -w password-file \
-h example1.server -p 1389 dc=example,dc=com example1.server:2636
```

經由 WAN 的複寫

目錄伺服器可讓您執行所有複寫格式，包含透過廣域網路 (WAN) 連線的機器之間的多重主節點複寫。此複寫可讓供應者伺服器初始化與更新用戶時，使用較高延遲與較低頻寬的最佳網路頻寬。

備註 – 部署或疑難排解經由 WAN 複寫的複寫拓樸時，必須檢查網路速度、延遲與資料封包遺失。這幾方面任何一個網路問題都可能會導致複寫延遲。

此外，複寫資料傳輸率在頻寬方面一律會比可用實際媒體允許的速率低。如果複本之間的更新磁碟區無法實際符合可用的頻寬，調校將無法避免複本在沉重的更新負載下有所差異。複寫延遲與更新效能受到許多因素的影響，包含但不限於：修改率、項目大小、伺服器硬體、錯誤率、平均延遲與平均頻寬。

如果您對貴環境中的複寫有任何疑問，請連絡 Sun 服務提供者。

複寫機制的內部參數預設會針對 WAN 進行最佳化。但是，如果複寫因爲上述因素過慢，您可能需要憑經驗調整視窗大小與群組大小參數。您也能排程複寫避開網路尖峰時段，因而改善整體網路使用率。最後，目錄伺服器支援壓縮複寫資料以最佳情況使用頻寬。

配置網路參數

視窗與群組網路參數可決定複寫機制如何將項目群組化，以用更有效率的方式將其在網路上進行傳送。這些參數會影響供應者與用戶如何交換複寫更新訊息與回應。這些參數可配置於每個複寫協議中，讓您可以根據各用戶的特定網路條件自訂複寫效能。

監視您所做的任何修改效果，並據以調整參數。如需相關指示，請參閱第 255 頁的 「取得複寫狀態」。您無須中斷複寫以修改視窗大小與群組大小參數。

配置視窗大小

視窗大小 (預設值 `10`) 表示不需要用戶立即回應即可傳送的最大更新訊息數目。

快速連續傳送多則訊息，會比在每則訊息之後等待回應來得有效率。您可以使用適當的視窗大小，降低複本等待複寫更新或回應抵達的時間。

如果用戶複本比供應者慢，請將視窗大小增加到比預設值還要高的值，例如 100，並在進一步調整之前再次檢查複寫效能。當複寫更新率很高而使得更新之間的時間縮短時，即使是區域網路 (LAN) 連線的複本也會從較高的視窗大小獲益。

▼ 配置視窗大小

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **修改視窗大小。**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port transport-window-size:value
```

例如：

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 transport-window-size:20
```

配置群組大小

群組大小 (預設值 1) 表示可以隨附在單一更新訊息中的最大資料修改數目。如果網路連線阻礙了複寫的進行，請將群組大小增加爲比預設值還要高的值，例如 10，再重新檢查複寫效能。

增加群組大小時，請確定以下爲眞：

- 視窗大小會設爲比群組大小還要大。
- 視窗大小除以群組大小會遠大於用戶上 cn=config 下的 nsslapd-maxThreadsPerConn 值 (一般爲兩倍)。

  群組大小設爲大於 1 時，供應者不會等待填滿一個群組後，再傳送更新給用戶。

▼ 配置群組大小

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **修改群組大小。**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
 consumer-host:consumer-port transport-group-size:value
```

例如：

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 transport-group-size:10
```

排程複寫活動

如果複本之間立即同步化不是很重要，您可以在網路使用率低的期間排程複寫。資料複寫的完成速度應於網路使用率高時快得多。

您可以排程複寫在一天當中的某個時間開始與結束，以每天或每週爲基準。您可以透過用戶各自的複寫協議，獨立爲每個用戶執行此項作業。新排程會立即生效，而導致對應用戶的下一個資料複寫延遲到排程允許的第一個複寫完成。

▼ 排程複寫活動

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **修改複寫排程**。

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
 host:port repl-schedule:value
```

例如，若要設定複寫在每晚 2:00 與 4:00 之間執行，請鍵入：

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 repl-schedule:"0200-0400 0123456"
```

其中 `0123456` 表示一週內的各天，`0` 表示星期日，`1` 表示星期一，依此類推。

配置複寫壓縮

若要降低複寫使用的頻寬，可以配置複寫在更新用戶時壓縮要傳送的資料。複寫機制使用 Zlib 壓縮程式庫。供應者與用戶必須執行於 Solaris 或 Linux 平台上，才能啓用壓縮。

您應該憑經驗測試與選取壓縮層級，以在 WAN 環境中使用預期複寫時達到最佳結果。由於壓縮與解壓縮運算會使複寫變慢，因此請勿在具有廣域網路頻寬的地方設定 LAN 中的此參數。

▼ 配置複寫壓縮

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **在主伺服器的複寫協議項目上配置複寫壓縮**。

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
 consumer-host:consumer-port transport-compression:level
```

其中 *level* 可以是 `high`、`medium`、`low` 或 `none`。

例如，若要在傳送複寫更新給 `host1:1389` 上的用戶時使用最快速的壓縮，請鍵入：

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 transport-compression:high
```

如需有關設定壓縮層級的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」。

修改複寫拓樸

本節說明管理現有複寫拓樸的以下幾方面：

- 第 247 頁的「變更複寫管理員」
- 第 247 頁的「管理複寫協議」
- 第 248 頁的「升級或降級複本」
- 第 249 頁的「停用複寫的尾碼」
- 第 250 頁的「保持複寫的尾碼同步化」

變更複寫管理員

您可以編輯複寫協議，以變更用以連結至用戶伺服器的複寫管理員身份識別。若要避免複寫中斷，請在用戶上定義新的複寫管理員項目或憑證項目，再修改複寫協議。但是，如果複寫因爲連結失敗而中斷，複寫機制將在複寫回復設定的限制內，於更正錯誤時自動傳送所有必要的更新。如需相關程序，請參閱第 228 頁的「使用非預設複寫管理員」。

管理複寫協議

您可以停用、啓用或刪除複寫協議。

停用複寫協議。

停用複寫協議時，主伺服器會停止傳送更新給指定的用戶。複寫到該伺服器的動作會停止，但是會保留協議中所有的設定。您可能會在稍後重新啓用協議而繼續複寫。如需有關中斷之後繼續複寫機制的資訊，請參閱第 248 頁的「啓用複寫協議」。

▼ **停用複寫協議。**

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

● **停用複寫協議。**

```
$ dsconf disable-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

例如：

```
$ dsconf disable-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

啓用複寫協議

啓用複寫協議會繼續指定用戶的複寫。但是，如果複寫中斷的時間比複寫回復設定所允許的時間長，且其他供應者尚未更新用戶，就必須重新初始化用戶。複寫回復設定的大小爲此供應者變更記錄的大小上限以及用戶清除延遲之最長存在期限 (請參閱第 225 頁的 「執行進階用戶配置」)。

當中斷時間很短且複寫可以回復時，主伺服器將會在重新啓用協議時自動更新用戶。

▼ 啓用複寫協議

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **啓用複寫協議。**

```
$ dsconf enable-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

例如：

```
$ dsconf enable-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

刪除複寫協議。

刪除複寫協議時會停止對應用戶的複寫，並會移除關於協議的所有配置資訊。若要在稍後繼續複寫，請改以第 247 頁的 「停用複寫協議。」中所述停用協議。

▼ 刪除複寫協議

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **刪除複寫協議。**

```
$ dsconf delete-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

例如：

```
$ dsconf delete-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

升級或降級複本

升級或降級複本會變更其在複寫拓樸中的角色。專屬用戶可升級爲集散中心，而集散中心可升級爲主伺服器。主伺服器可降級爲集散中心，而集散中心也可降級爲專屬用戶。但是，主伺服器無法直接降級爲用戶，正如用戶無法直接升級爲主伺服器。

多重主伺服器複寫機制內所允許的升級與降級使得拓樸非常具有彈性。之前由用戶複本提供服務的站點可能會增大，而需要集散中心與多個複本以處理負載。如果負載包含許多複本內容修改，集散中心會變成主伺服器以允許更快速的本機變更，而能接著複寫至其他站點的其他主伺服器。

升級或降級複本時，請注意下列事項：

- 如果您將用戶升級，它會變成集散中心。如果您將集散中心升級，它會變成主伺服器。您無法將伺服器直接從用戶升級爲主伺服器。您必須先將用戶升級爲集散中心，然後再將集散中心升級爲主伺服器。反之亦然，當您將主伺服器降級爲用戶時，必須先將主伺服器降級爲集散中心，然後才能從集散中心降級爲用戶。
- 將主伺服器降級爲集散中心時，複本會變成唯讀且會配置成傳送參照至其他主伺服器。新的集散中心會保留所有用戶，不論是集散中心或專屬用戶。
- 將單一主伺服器降級爲集散中心會建立不含主伺服器複本的拓樸。假設您將定義新的主伺服器，目錄伺服器可讓您執行此項作業。但是，最好增加新主伺服器做爲多重主伺服器並初始化該主伺服器，再降級其他主伺服器。
- 將集散中心降級爲用戶之前，必須停用或刪除出入集散中心的所有複寫協議。若未這麼做，降級作業將失敗並出現錯誤：`LDAP_OPERATIONS_ERROR` “`Unable to demote a hub to a read-only replica if some agreements are enabled (`**如果啓用某些協議，便無法將集散中心降級為唯讀的複本**`)`”。

  如果其他集散中心或主伺服器尚未更新集散中心的用戶，之後將無法更新。您應該在其他集散中心或主伺服器上建立新的協議，以更新這些用戶。
- 將用戶升級爲集散中心時，會啓用其變更記錄，且可利用用戶定義新的協議。
- 將集散中心升級爲主伺服器時，複本會接受修改請求，且您可利用其他主伺服器、集散中心或專屬用戶定義新的協議。

▼ 升級或降級複本

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **使用下列其中一個指令升級或降級複本：**

```
$ dsconf promote-repl -h host -p port role suffix-DN

$ dsconf demote-repl -h host -p port role suffix-DN
```

其中 *role* 是 `master`、`hub` 或 `consumer`。

停用複寫的尾碼

停用複寫的尾碼會從複寫拓樸中移除該尾碼。若尾碼角色爲主伺服器、集散中心或用戶，尾碼將再也無法更新或傳送更新。停用供應者伺服器上的尾碼會刪除所有複寫協議，且如果再次啓用複本，將必須重新建立複寫協議。

▼ 停用複寫的尾碼

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **停用複寫的尾碼**。

  ```
  $ dsconf disable-repl -h host -p port suffix-DN
  ```

  例如：

  ```
  $ dsconf disable-repl -h host2 -p 1389 dc=example,dc=com
  ```

保持複寫的尾碼同步化

在停止複寫中所包含的目錄伺服器以進行定期維護之後，必須確保當伺服器恢復連線時，能立即透過複寫取得更新。若是多重主伺服器環境中的主伺服器，必須有多重主伺服器集中的其他主伺服器更新目錄資訊。在其他情況下，集散伺服器或專屬用戶伺服器離線進行維護之後，當伺服器再次連線時，必須由主伺服器加以更新。

本節說明複寫重試演算法，並說明如何不等待下次重試便強制執行複寫更新。

備註 – 本節所述的程序僅能在已設定複寫**且**已初始化用戶時使用。

複寫重試演算法

當來源複本複寫至目標失敗時，會定期依遞增時間間隔重試。重試間隔會視錯誤類型而定。

請注意，如果所配置的複寫協議一律會使來源複本與目標複本保持同步化，光是立即更新離線超過五分鐘的複本都會不足。

▼ 強制執行複寫更新

如果停止複寫，您可以強制執行目標尾碼的複寫更新。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **在來源伺服器上重新啓動目標伺服器的複寫更新**。

  ```
  $ dsconf update-repl-dest-now -h host -p port suffix-DN destination-host:destination-port
  ```

  例如：

  ```
  $ dsconf update-repl-dest-now -h host2 -p 1389 dc=example,dc=com host1:1389
  ```

將主伺服器複本移至新的機器

在某些情況中，可能需要將主伺服器複本移至不同的機器。若不需要使用相同的主機名稱與連接埠號碼，請使用 `dsconf change-repl-dest` 變更遠端複本的主機名稱與連接埠號碼。如需更多資訊，請參閱第 232 頁的「變更複寫協議的目標」。

若需要保留相同的主機名稱與連接埠號碼，您必須從現有的拓樸中移除主伺服器，再將主伺服器重新加入拓樸。

由於 DSCC 會處理所有受影響的複寫協議，因此使用 DSCC 比較容易執行這些作業。但是，如果使用 DSCC，您將無法指定主伺服器在拓樸中原本就有的相同複本 ID。若要使用相同的複本 ID，您必須使用如下的指令行來執行這些作業。

▼ 從現有的複寫拓樸中移除主伺服器

開始之前 請確定已複寫所有來自主伺服器的變更。

1. **如果可以的話，請使用二進位副本備份主伺服器，以避免遺失任何變更。**
2. **將主伺服器複本降級為集線器複本。**
   請參閱第 248 頁的「升級或降級複本」。
3. **等候集線器啓動至其他伺服器的複寫。**
   當集線器開啓至拓樸中其他伺服器的複寫階段作業時，RUV 中會保留該集線器，但參照已不再使用。
4. **停止集線器。**
   請參閱第 61 頁的「啓動、停止與重新啓動目錄伺服器實例」。
5. **從拓樸中移除集線器。**
   請參閱第 249 頁的「停用複寫的尾碼」。

▼ 將主伺服器增加至現有的複寫拓樸

1. **使用相同複本 ID 增加主伺服器複本。**
   請參閱第 227 頁的「啓用主伺服器複本上的複寫」。
2. **從該主伺服器重建至拓樸中其他複本的複寫協議。**
3. **初始化新的主伺服器。**
   a. **如果您之前可以備份主伺服器，請從此備份初始化主伺服器。**

b. 如果您之前無法備份主伺服器 (例如因為當機)，請從拓樸中的其他主伺服器初始化該主伺服器。

Directory Server 6.3 之前的版本複寫

本節提供有關如何使用 6.3 之前的目錄伺服器版本配置複寫之資訊。

在 Directory Server 6.3 與 Directory Server 5.1 或 5.2 之間進行複寫

Directory Server 5.1、5.2 與 6.3 在複寫配置方面相容，但下列幾點例外：

- Directory Server 6.3 之前的版本不支援複寫優先權。如果在 6.3 主伺服器複本上配置複寫優先權，複寫優先權會傳輸到執行 Directory Server 6.3 的用戶，而不會傳輸到執行目錄伺服器之前版本的任何用戶。
- 包含 Directory Server 5.1 或 5.2 主伺服器的複寫拓樸上不支援具有無限部主伺服器。雖然 Directory Server 6.3 支援複寫拓樸中有無限部主伺服器，但如果複寫拓樸包含任何 Directory Server 5.2 主伺服器，此數目會限制爲四。Directory Server 5.1 不支援多重主伺服器複寫。

使用回溯變更記錄

LDAP 用戶端使用回溯變更記錄確定對目錄伺服器資料所做的變更歷程記錄。回溯變更記錄儲存在與目錄伺服器變更記錄不同的資料庫中，位於尾碼 `cn=changelog` 下。

回溯變更記錄可以在複寫拓樸中的獨立伺服器或每部伺服器上啓用。在一部伺服器上啓用回溯變更記錄時，預設會記錄該伺服器上所有尾碼的更新。回溯變更記錄可以配置爲僅記錄指定尾碼的更新。

如需有關在複寫的拓樸中使用回溯變更記錄的資訊，以及有關使用回溯變更記錄的限制，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Replication and the Retro Change Log Plug-In」。

如需有關回溯變更記錄中某項目的屬性之資訊，請參閱 `changeLogEntry`(5dsoc) 線上手冊。

如需修改回溯變更記錄的更多資訊，請參閱 `dsconf`(1M) 線上手冊。

本節說明可以使用回溯變更記錄的各種方式。

▼ 啓用回溯變更記錄

若要使用回溯變更記錄，必須啓用記錄。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 **修改回溯變更記錄配置項目：**

```
$ dsconf set-server-prop -h host -p port retro-cl-enabled:on
```

2 **重新啓動伺服器。**

如需相關資訊，請參閱第 61 頁的 「啓動、停止與重新啓動目錄伺服器實例」。

▼ 配置回溯變更記錄以記錄指定尾碼的更新

在一部伺服器上啓用回溯變更記錄時，預設會記錄該伺服器上所有尾碼的更新。本程序說明如何配置回溯變更記錄僅記錄指定尾碼的更新。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 **修改回溯變更記錄配置項目：**

```
$ dsconf set-server-prop -h host -p port retro-cl-suffix-dn:suffix-DN
```

例如，若要僅記錄在 `cn=Contractors,dc=example,dc=com` 尾碼與 `ou=People,dc=example,dc=com` 尾碼上的變更，請使用此指令：

```
$ dsconf set-server-prop -h host2 -p 1389 \
 retro-cl-suffix-dn:"cn=Contractors,dc=example,dc=com" \
 retro-cl-suffix-dn:"ou=People,dc=example,dc=com"
```

若要將尾碼增加至現有的指定尾碼清單中，請使用此指令：

```
$ dsconf set-server-prop -h host -p port retro-cl-suffix-dn+:suffix-DN
```

2 **重新啓動伺服器。**

如需相關資訊，請參閱第 61 頁的 「啓動、停止與重新啓動目錄伺服器實例」。

▼ 配置回溯變更記錄以記錄刪除項目的屬性

本程序說明如何配置回溯變更記錄，以在刪除項目時記錄該項目的指定屬性。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **指定必須記錄的屬性：**

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr: \
    attribute1  attribute2
   ```

   例如，若要將回溯變更記錄設爲記錄刪除項目的 UID 屬性，請使用此指令：

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr:uid
   ```

   若要將屬性增加至現有的指定屬性清單中，請使用此指令：

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr+:attribute
   ```

2. **重新啓動伺服器。**

   如需相關資訊，請參閱第 61 頁的「啓動、停止與重新啓動目錄伺服器實例」。

▼ 修剪回溯變更記錄

回溯變更記錄中的項目在指定的一段時間過後，會自動移除。若要配置項目經過多久的時間後會自動刪除，請務必啓用回溯變更記錄，再設定 `cn=Retro Changelog Plugin`、`cn=plugins` 與 `cn=config` 項目中的 `nsslapd-changelogmaxage` 配置屬性。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **檢查是否已啓用回溯變更記錄。**

   ```
   $ dsconf get-server-prop -h host -p port retro-cl-enabled
   ```

2. **若未啓用回溯變更記錄，請啓用記錄。**

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-enabled:on
   ```

3. **為記錄的變更設定最長存在期限。**

   ```
   $ dsconf set-server-prop -h host -p port retro-cl-max-age:duration
   ```

   其中 *duration* 可以是`undefined` (無存在期限) 或下列其中之一：

   - `s` 表示秒
   - `m` 表示分鐘
   - `h` 表示小時
   - `d` 表示天
   - `w` 表示週

   例如，若要將回溯變更記錄最長存在期限設爲兩天，請鍵入：

   ```
   $ dsconf set-server-prop -h host 2 -p 1389 retro-cl-max-age:2d
   ```

   每 5 分鐘會從變更記錄移除超過此存在期限的項目。

存取控制與回溯變更記錄

回溯變更記錄支援搜尋作業。該記錄適用於包含此格式的篩選器之搜尋：

```
(&(changeNumber>=X)(changeNumber<=Y))
```

根據一般規則，請勿於回溯變更記錄項目上執行增加或修改作業。您可以刪除項目以修剪記錄大小。修改預設存取控制策略是唯一需要在回溯變更記錄上執行的修改作業。

建立回溯變更記錄時，預設會套用下列存取控制規則：

- 所有認證的使用者 (`userdn=anyone`，以與 `userdn=all` 的匿名存取有所區分) 皆會被授予回溯變更記錄最上層項目 `cn=changelog` 之讀取、搜尋與比較權限。
- 除了以隱含方式授予目錄管理員之外，不會授予寫入與刪除存取權。

  請勿授予匿名使用者讀取權，因爲回溯變更記錄項目可能包含密碼等機密資訊的修改。如果不希望認證的使用者能檢視回溯變更記錄內容，可能會想要進一步限制該內容的存取權。

若要修改套用到回溯變更記錄的預設存取控制策略，請修改 `cn=changelog` 項目的 `aci` 屬性。請參閱第 7 章。

取得複寫狀態

您可以使用 DSCC 或使用指令行工具取得複寫狀態。

在 DSCC 中取得複寫狀態

您可以使用 [尾碼] 標籤以圖形化方式檢視複寫，包含複寫協議與複寫延遲。如需更多資訊，請參閱 DSCC 線上說明。

此外，您可以使用 DSCC 檢視複寫拓樸，如下圖所示。

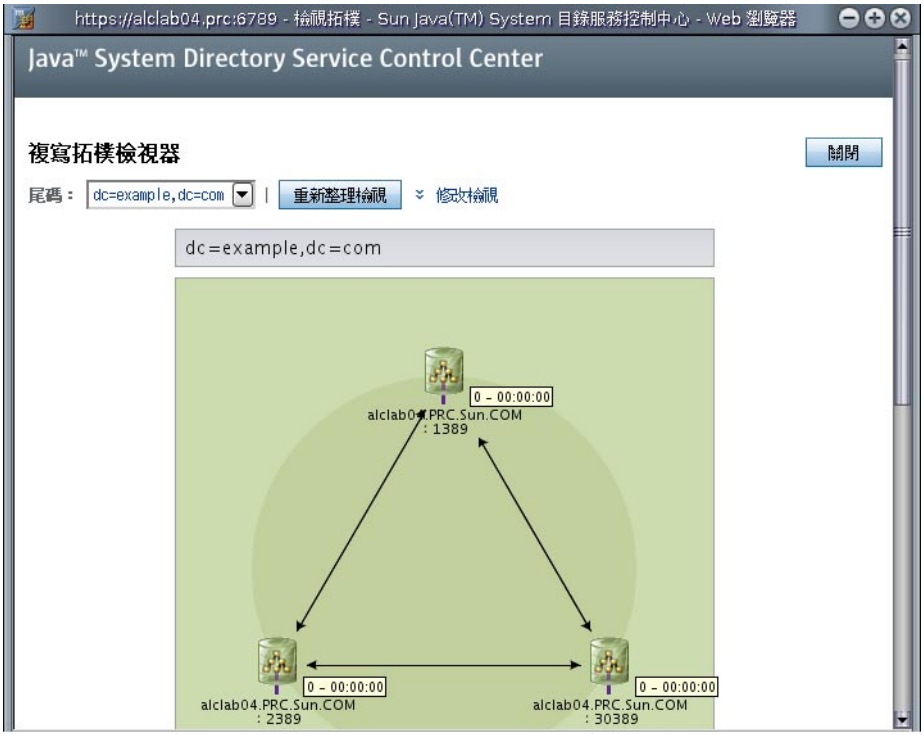


圖 11–1 複寫拓樸範例

透過使用指令行取得複寫狀態

如果無法使用 DSCC，請使用指令行工具取得複寫部署的相關資訊。

線上手冊提供完整的指令行語法與這些工具的使用範例。

- `repldisc` - 「探索」與建構包含複寫部署中所有已知伺服器的表格。請參閱 `repldisc`(1) 線上手冊。
- `insync` - 表示供應者與一或多個用戶複本之間的同步化狀態。請參閱 `insync`(1) 線上手冊。
- `entrycmp` - 比較兩個以上複本中的相同項目。請參閱 `entrycmp`(1) 線上手冊。

若要尋找這些指令所在的目錄，請參閱第 28 頁的 「指令位置」。

解決常見複寫衝突

多重主伺服器複寫使用約略一致的複寫模式。這表示相同的項目可能會在不同的伺服器上同時修改。當更新在兩部伺服器之間互傳時，必須解決所有衝突的變更。大多數的衝突會自動解決。例如，與各伺服器上的變更相關之時間戳記會經由使用最近的變更而獲得解決。但是，某些變更衝突需要手動介入才能解決。

本節包含下列主題：

- 第 257 頁的「使用 DSCC 解決複寫衝突」
- 第 257 頁的「使用指令行解決複寫衝突」
- 第 257 頁的「解決命名衝突」
- 第 259 頁的「解決孤立項目的衝突」
- 第 260 頁的「解決可能的互通操作問題」

使用 DSCC 解決複寫衝突

解決複寫衝突最簡單的方式是使用 DSCC。如需相關資訊，請參閱 DSCC 線上說明。

使用指令行解決複寫衝突

您可以使用指令行解決複寫衝突。無法經由複寫程序自動解決之變更衝突的項目，會包含操作屬性 `nsds5ReplConflict` 做爲衝突記號。

若要尋找出現衝突的項目，請定期搜尋包含此屬性的項目。例如，您可以使用下列 `ldapsearch` 指令尋找衝突的項目：

```
$ ldapsearch -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config \
 -w - -b "dc=example,dc=com" "(nsds5ReplConflict=*)"
```

請注意，預設會編製 `nsds5ReplConflict` 屬性的索引。

解決命名衝突

具有相同 DN 的項目如果是在伺服器彼此複寫變更之前所建立的，則可能是建立在不同的主伺服器上。經過複寫之後，衝突解決機制會自動重新命名第二個建立的項目。

具有 DN 命名衝突的項目會在其 DN 中包含操作屬性 `nsuniqueid` 提供的唯一識別碼，進行重新命名。

例如，如果項目 uid=bjensen,ou=People,dc=example,dc=com 在兩部主伺服器上同時建立，則這兩部主伺服器在複寫之後會有以下兩個項目：

- uid=bjensen,ou=People,dc=example,dc=com
- nsuniqueid=66446001-1dd211b2-66225011-2ee211db+uid=bjensen,dc=example,dc=com

必須爲第二個項目提供有用的 DN。您可以刪除衝突的項目，並再次以非衝突的名稱增加。但是，重新命名項目可確保其內容不會變更。重新命名程序會視命名屬性是單值或多值而有所不同。請參閱下列程序。

▼ 重新命名具有多值命名屬性的衝突項目

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 重新命名項目，同時保留舊 RDN 值，例如：

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: nsuniqueid=66446001-1dd211b2-66225011-2ee211db+uid=bjensen,dc=example,dc=com
changetype: modrdn
newrdn: uid=bj66446001
deleteoldrdn: 0
^D
```

您無法刪除此步驟中的舊 RDN 值，因爲該值還同時包含無法刪除的 nsuniqueid 操作屬性。

2 移除命名屬性的舊 RDN 值與衝突記號屬性，例如：

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bj66446001,dc=example,dc=com
changetype: modify
delete: uid
uid: bjensen
-
delete: nsds5ReplConflict
^D
```

▼ 重新命名具有單值命名屬性的衝突項目

複寫項目中的命名屬性爲單值時，例如 dc (網域元件)，您無法僅重新命名項目爲相同屬性的其他值。相反的，您必須提供項目暫存名稱。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **使用不同的命名屬性重新命名項目，並保留舊 RDN，例如：**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: nsuniqueid=66446001-1dd211b2-66225011-2ee211db+dc=HR,dc=example,dc=com
changetype: modrdn
newrdn: o=TempHREntry
deleteoldrdn: 0
^D
```

   您無法刪除此步驟中的舊 RDN 值，因爲該值還同時包含無法刪除的 `nsuniqueid` 操作屬性。

2. **將想要的命名屬性變更為唯一值，並移除衝突記號屬性，例如：**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: o=TempHREntry,dc=example,dc=com
changetype: modify
replace: dc
dc: NewHR
delete: nsds5ReplConflict
^D
```

3. **將項目重新命名回預定命名屬性，例如：**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: dc=NewHR,dc=example,dc=com
changetype: modrdn
newrdn: dc=HR
deleteoldrdn: 1
^D
```

   您可以將 `deleteoldrdn` 屬性值設爲 `1`，刪除成對的暫存屬性值組 `o=TempHREntry`。若要保留此屬性，請將 `deleteoldrdn` 屬性值設爲 `0`。

解決孤立項目的衝突

複寫刪除作業時，若是用戶伺服器發現要刪除的項目有子項目，衝突解決程序會建立**接點**項目，以避免目錄中出現孤立項目。

同理，複寫增加作業時，若是用戶伺服器找不到父項目，衝突解決程序就會建立表示父系的接點項目，使得新項目不會成爲孤立項目。

接點項目是包含物件類別 glue 與 extensibleObject 的暫存項目。接點項目會以下列各種方式建立：

- 如果衝突解決程序發現有刪除的項目符合唯一識別碼，接點項目會是該項目的復原內容。接點項目也包含 glue 物件類別與 nsds5ReplConflict 屬性。

  此情況下，您可以修改接點項目以移除 glue 物件類別與 nsds5ReplConflict 屬性，使該項目維持爲一般項目，或刪除接點項目與其子項目。
- 伺服器會以 glue 與 extensibleObject 物件類別建立基本項目。

  此情況下，您必須修改項目使其變成有意義的項目，或刪除項目與所有子項目。

解決可能的互通操作問題

如需依賴屬性唯一性的應用程式互通操作，例如郵件伺服器，您可能必須限制包含 nsds5ReplConflict 屬性的項目之存取權。如果沒有限制這些項目的存取權，僅需要一個屬性的應用程式會同時選擇原始項目與包含 nsds5ReplConflict 的衝突解決項目，而作業將因此失敗。

若要限制存取，您必須使用下列指令修改授予匿名讀取權的預設 ACI：

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: dc=example,dc=com
changetype: modify
delete: aci
aci: (target ="ldap:///dc=example,dc=com")
 (targetattr !="userPassword"
 (version 3.0;acl "Anonymous read-search  access";
 allow (read, search, compare)(userdn = "ldap:///anyone");)
-
add: aci
aci: (target="ldap:///dc=example,dc=com")
 (targetattr!="userPassword")
 (targetfilter="(!(nsds5ReplConflict=*))")(version 3.0;acl
 "Anonymous read-search access";allow (read, search, compare)
 (userdn="ldap:///anyone");)
^D
```

新的 ACI 會避免搜尋結果中傳回包含 nsds5ReplConflict 屬性的項目。

◆ ◆ ◆ 第 12 章

12 目錄伺服器模式

目錄伺服器附有標準模式，內含數以百計的物件類別與屬性。雖然標準物件類別與屬性應足以因應您大部分的需求，但仍有可能需要建立新的物件類別與屬性，以延伸模式。如需標準模式的簡介，以及有關設計適合您部署之模式的指示，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」。

本章說明如何管理您的模式，其中包含下列主題：

- 第 261 頁的「管理模式檢查」
- 第 262 頁的「關於自訂模式」
- 第 267 頁的「透過 LDAP 管理屬性類型」
- 第 270 頁的「透過 LDAP 管理物件類別」
- 第 273 頁的「延伸目錄伺服器模式」
- 第 276 頁的「複寫目錄模式」

管理模式檢查

開啓模式檢查後，目錄伺服器會確認所有的匯入、增加以及修改作業確實符合目前所定義的目錄模式。

- 每個項目的物件類別與屬性均符合模式。
- 項目中包含其所有已定義之物件類別的所有必要屬性。
- 項目僅包含其物件類別所允許的屬性。

備註 – 修改項目時，目錄伺服器會對整個項目執行模式檢查，而不是僅檢查進行修改的屬性。因此，若項目中有任何物件類別或屬性不符合模式，作業即可能失敗。

但模式檢查並不會驗證屬性值在語法方面的有效性。

模式檢查預設爲開啓。一般情況下，在執行目錄伺服器時請開啓模式檢查。許多用戶端應用程式均假設，開啓模式檢查即表示所有項目皆符合模式。但在開啓模式檢查

後，目錄伺服器並不會因此而驗證目錄中現有的內容。唯一能夠確保所有目錄內容均符合模式的方法，是在增加任何項目或重新初始化所有項目之前開啓模式檢查。

在某些情況下您會想關閉模式檢查，例如爲使已知符合模式的 LDIF 檔案加快匯入作業速度而關閉，是其中之一。但如此做會有匯入項目不符合模式的風險。若關閉模式檢查，則無法偵測不符合模式的匯入項目。

如需在複寫環境中使用模式檢查的詳細資訊，請參閱第 276 頁的 「複寫目錄模式」。

▼ 修正模式規範遵循問題

當項目不符合模式時，即可能無法搜尋此項目，而使項目的修改作業失敗。若要更正此問題，請遵循此程序中的步驟。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

開始之前 爲避免日後需要修正模式規範遵循問題，請在進行部署前事先規劃您的模式，以儘可能減少模式變更。如需更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」。

1. **若想找出項目不符合的原因，請擷取該項目，並手動將其與目前定義的模式進行比較。**
   如需詳細資訊，請參閱第 269 頁的 「檢視屬性類型」與第 272 頁的 「檢視物件類別」。
2. **修改項目以符合模式，或修改模式以符合項目。**

關於自訂模式

若標準模式不足以因應您的目錄需求，可以加以延伸。自訂模式時，請遵循下列指示：

- 儘可能重複使用現有的模式元素。
- 儘可能減少您爲每個物件類別所定義的必要屬性數目。
- 請不要定義多個相同用途的物件類別或屬性。
- 儘可能保持模式的單純性。

自訂模式時，請勿在標準模式中修改、刪除或取代屬性或物件類別的任何現有定義。這些動作可能導致無法與其他目錄和 LDAP 用戶端應用程式相容的問題。

請勿修改任何目錄伺服器內部操作屬性。但您可以爲外部應用程式建立您自己的操作變數。

一律定義物件類別，而不要使用 `objectClass: extensibleObject`。目錄伺服器不會對具有物件類別 `extensibleObject` 的項目執行模式檢查，因此無法限制或檢查項目中所含的屬性。目錄伺服器無法檢查出應用程式中的拼字錯誤，例如將 `givenName` 屬性類型誤植爲 `giveName`。再者，目錄伺服器也必須假設 `extensibleObject` 項目所有其他未定義的屬性皆爲多值屬性，且使用區分大小寫的字串語法。此外，有些應用程式須依賴具有特定物件類別的項目。一般而言，若您的應用程式必須使用物件類別的延伸，則不應放棄模式管理。此時應建立內含應用程式所需屬性的輔助物件類別。

本節包含有關預設目錄模式與建立自訂屬性與物件類別的資訊。

預設目錄伺服器模式

目錄伺服器隨附之模式的相關說明，位於儲存在 *instance-path*`/config/schema/` 目錄中的檔案集內。

此目錄包含目錄伺服器與相關產品的所有共用模式。LDAP v3 標準使用者與組織模式位於 `00core.ldif` 檔案中。舊版目錄所使用的配置模式則位於 `50ns-directory.ldif` 檔案中。

備註 – 請勿在伺服器執行時修改此目錄中的檔案。

物件識別碼

每個 LDAP 物件類別或屬性，都必須指定唯一的名稱與物件識別碼 (OID)。定義模式時，OID 在您的組織中必須是唯一的。一個 OID 即足以因應您所有的模式需求。接著，您可以爲屬性與物件類別的此 OID 新增分支。

取得及指定您模式中的 OID 時，須執行下列動作：

- 從網際網路位址指派機構 (IANA) 或國家機構爲您的組織取得 OID。

  有些國家/地區已爲公司指定 OID。若您的組織尚無 OID，您可以從 IANA 加以取得。

- 建立 OID 登錄，以便追蹤 OID 指定。

  OID 登錄是一份由您自己維護的清單，其中列出您的目錄模式中所使用的 OID 與 OID 說明。OLD 登錄可確保任何 OID 皆不會用於多項用途上。

- 建立 OID 樹狀結構中的分支以容納模式元素。

  以 *OID*`.1` 代表屬性，並以 *OID*`.2` 代表物件類別，在 OID 分支或您的目錄模式下建立至少兩個分支。若要定義您自己的對應規則或控制，您可以視需要新增分支，如 *OID*`.3`。

命名屬性與物件類別

建立新屬性與物件類別的名稱時，請讓名稱具有意義，以便使用模式。

您可以在自訂元素中加上唯一前綴，以避免自訂模式元素與現有模式元素之間產生命名衝突。以 Example.com Corporation 爲例，它可在其各個自訂的模式元素前加上前綴 `Example`。它也可增加名爲 `ExamplePerson` 的特殊物件類別，以識別其目錄中的 Example.com 員工。

請注意，在 LDAP 中，屬性類型名稱與物件類別名稱均會**區分大小寫**。應用程式應將其視爲區分大小寫的字串。

定義新的物件類別時

當現有的物件類別不支援您必須儲存在目錄項目中的所有資訊時，您可以新增物件類別。

有兩種方法可新建物件類別：

- 建立許多新的物件類別，一一用於需要增加屬性的各個物件類別結構上。
- 建立單一物件類別，並使其支援您爲目錄所建立的所有屬性。您可以將此類型的物件類別定義爲 AUXILIARY 物件類別，而加以建立。

  假設您的站點必須建立 ExampleDepartmentNumber 與 ExampleEmergencyPhoneNumber 屬性。您可以建立數個允許這些屬性有一些子集的物件類別。您可以建立名爲 ExamplePerson 的物件類別，並使其允許 ExampleDepartmentNumber 與 ExampleEmergencyPhoneNumber 屬性。ExamplePerson 的父系將是 inetOrgPerson。接著，您可以建立名爲 ExampleOrganization 的物件類別，並使其也允許 ExampleDepartmentNumber 與 ExampleEmergencyPhoneNumber 屬性。ExampleOrganization 的父系將是 organization 物件類別。

  新的物件類別會以 LDAP v3 模式顯式，如下顯示：

```
objectclasses: (1.3.6.1.4.1.42.2.27.999.1.2.3 NAME 'ExamplePerson'
 DESC 'Example Person Object Class' SUP inetorgPerson STRUCTURAL MAY
 (ExampleDepartmentNumber $ ExampleEmergencyPhoneNumber) )
objectclasses: (1.3.6.1.4.1.42.2.27.999.1.2.4 NAME
 'ExampleOrganization' DESC 'Example Organization Object Class' SUP
 organization STRUCTURAL MAY (ExampleDepartmentNumber
 $ ExampleEmergencyPhoneNumber) )
```

  此外，您也可以建立允許上述所有屬性的單一物件類別。接著，您即可對要使用屬性的任何項目使用此物件類別。單一物件類別將如下所示：

```
objectclasses: (1.3.6.1.4.1.42.2.27.999.1.2.5 NAME 'ExampleEntry'
 DESC  'Example Auxiliary Object Class' SUP top  AUXILIARY MAY
 (ExampleDepartmentNumber $ ExampleEmergencyPhoneNumber) )
```

  新的 ExampleEntry 物件類別會以 AUXILIARY 標示，表示無論其結構物件類別爲何，皆可用於任何項目。

在決定如何實作新的物件類別時，請考量下列事項。

- 多重 STRUCTURAL 物件類別將會導致需建立及維護更多模式元素。

  一般而言，會將元素的數量保持在較小的情況，而不需太多的維護。但若您打算在模式中增加二或三個物件類別，使用單一物件類別可能會較爲容易些。

- 多重 STRUCTURAL 物件類別在資料設計上的要求較爲愼重與嚴苛。

  嚴苛的資料設計將迫使您考量在物件類別結構中納入資料各個部分。這項限制可能確實有幫助，但也可能綁手綁腳。

- 單一 AUXILIARY 物件類別可在您需要將資料放置到多種類型的物件類別結構上時，簡化所需的資料設計。

  例如，假設您在人員與群組項目上皆需要 preferredOS。您可以僅建立單一物件類別以允許此屬性。

- 請設計與實際物件及群組元素相關，且可形成合理群組的物件類別。
- 請避免爲新的物件類別設定必要屬性。

  必要屬性會使您的模式缺乏彈性。當您建立新的物件類別時，請設定允許的屬性，而不要設定必要屬性。

  定義新的物件類別後，您必須決定物件類別所允許及要求的屬性，以及此物件類別繼承哪些或哪個物件類別。

定義新的屬性時

當現有的屬性不支援您必須儲存在目錄項目中的所有資訊時，您可以新增屬性。請儘可能使用標準屬性。請搜尋原本即已存在於預設目錄模式中的屬性，並使用與新的物件類別相關的屬性。

例如，您可能會發現，您想儲存到人員項目上的資訊，超出 `person`、`organizationalPerson` 或 `inetOrgPerson` 物件類別所支援的範圍。您想在目錄中儲存生日時，標準目錄伺服器模式中卻沒有屬性存在。您可以建立名爲 `dateOfBirth` 的新屬性。請定義允許此屬性的新輔助類別，以允許此屬性用於代表人員的項目上。

建立自訂模式檔案時

建立自訂模式檔案時，請留意下列事項，尤其是使用複寫時更需特別注意：

- 新增模式元素時，所有屬性皆須先完成定義，方可用於物件類別中。您可以在相同的模式檔案中定義屬性與物件類別。
- 您所建立的每個自訂屬性或物件類別，應定義於單一的模式檔案中。此做法可防止伺服器在載入最新建立的模式時，覆寫了任何先前的定義。目錄伺服器在載入模式檔案時，會先依據數字順序，再依字母順序。
- 手動定義新的模式定義時，一般理想的做法是將這些定義增加到 `99user.ldif` 檔案中。

  當您使用 LDAP 更新模式元素時，新的元素會自動寫入 `99user.ldif` 檔案中。因此，您在自訂模式檔案中所做的任何其他模式定義變更，均可能遭到覆寫。僅使用 `99user.ldif` 檔案，可防止模式元素重複的情況，以及模式變更遭到覆寫的風險。
- 由於目錄伺服器會依據字母數字順序載入模式檔案，且會先載入數字，因此您應以下列格式爲自訂模式檔案命名：

  `[00-99]` *`filename`*`.ldif`

  此數字高於任何已定義的目錄標準模式。

  若您以低於標準模式檔案的數字爲模式檔案命名，伺服器即可能在載入模式時發生錯誤。此外，所有標準屬性與物件類別皆會在您的自訂模式元素完成載入後，才會載入。

- 請確定自訂模式檔案的名稱在數字與字母上皆未高於 99user.ldif，因爲目錄伺服器會使用最高順序的檔案進行其內部模式管理。

  例如，若您建立模式檔案，並將其命名爲 99zzz.ldif，則在您下次更新模式時，所有 X-ORIGIN 值爲 'user defined' 的屬性，都將寫入 99zzz.ldif 中。結果將造成兩個 LDIF 檔案含有重複的資訊，而 99zzz.ldif 檔案中的部分資訊可能會遭清除。
- 根據一般通則，應以下列兩個項目識別您所增加的自訂模式元素：
  - 自訂模式檔案之 X-ORIGIN 欄位中的 'user defined'；
  - 更具說明性的標籤，如 X-ORIGIN 欄位中的 'Example.com Corporation defined'，以便讓其他管理員更容易瞭解自訂模式元素。例如，X-ORIGIN ('user defined' 'Example.com Corporation defined')。

    若您以手動方式增加模式元素，且未使用 X-ORIGIN 欄位中的 'user defined'，模式元素在 DSCC 中即會處於唯讀狀態。

    當您使用 LDAP 或 DSCC 增加自訂模式定義時，伺服器即會自動增加 'user defined' 值。但若您未在 X-ORIGIN 欄位中加入更多說明性的值，日後就可能就難以瞭解此模式的相關用途。

  請手動將自訂模式檔案傳播到所有的伺服器上，因爲這些變更不會自動複寫。

  當您變更目錄模式時，伺服器會留存時間戳記，以記錄變更模式的時間。在每個複寫階段作業開始時，伺服器會將其時間戳記與其用戶的時間戳記進行比較，然後在必要時發送模式變更。對於自訂模式檔案，伺服器只會保存一個時間戳記，而此戳記與 99user.ldif 檔案相關聯。這表示，您對 99user.ldif 以外的檔案所做的任何自訂模式檔案變更或增加，都不會進行複寫。因此，您必須將自訂模式檔案傳播到所有其他的伺服器上，以確保所有模式資訊均存在於拓樸各處。

透過 LDAP 管理屬性類型

本節說明如何透過 LDAP 建立、檢視及刪除屬性類型。

建立屬性類型

cn=schema 項目具有多值屬性 attributeTypes，此屬性含有目錄模式中各種屬性類型的定義。您可以使用 ldapmodify(1) 指令增加至這些定義。

新的屬性類型定義，以及您對使用者定義的屬性類型所做的變更，都會儲存在 99user.ldif 檔案中。

對於每個屬性類型定義，您必須至少提供一個 OID 以定義您新的屬性類型。請考慮對新的屬性類型至少使用下列元素：

- **屬性 OID**。對應至您屬性的物件識別碼。OID 是一個可唯一識別模式物件的字串，通常爲小數點十進位數字。

  如需嚴格的 LDAP v3 規範遵循，則必須提供有效數值 OID。若想進一步瞭解 OID，或要爲您的企業申請前綴，請將電子郵件寄至 IANA (網際網路位址指派機構) `iana@iana.org`，或造訪 IANA 網站 (`http://www.iana.org`)。
- **屬性名稱**。對應至屬性的唯一名稱。也稱爲其屬性類型。屬性名稱必須以字母開頭，且只能包含 ASCII 字母、數字與連字符。

  屬性名稱可包含大寫字母，但 LDAP 用戶端不會以大小寫分辨屬性。以區分大小寫的方式處理屬性名稱時，應遵循 RFC 4512 (`http://www.ietf.org/rfc/rfc4512.txt`) 第 2.5 款的規定。

  您可以選擇性地爲屬性類型加入替代屬性名稱，亦即別名。
- **屬性說明**。以簡短的說明性文字說明屬性的用途。
- **語法**。供 OID 參照，用以說明屬性將包含的資料。

  屬性語法及其 OID 均列載於 RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`) 中。
- **允許的值數目**。屬性預設爲多值屬性，但您也可以將屬性限定爲單一值。

▼ 建立屬性類型

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **根據 RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`) 中所指定的語法，備妥您的屬性類型定義**。

2. **使用 `ldapmodify`(1) 指令，增加您的屬性類型定義**。

   請注意，目錄伺服器會將 `X-ORIGIN 'user defined'` 增加到您所提供的定義中。

範例 12–1 建立屬性類型

下列範例使用 `ldapmodify` 指令，增加使用「目錄字串」語法的新屬性類型。

```
$ cat blogURL.ldif
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: ( 1.2.3.4.5.6.7
 NAME ( 'blog' 'blogURL' )
 DESC 'URL to a personal weblog'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE )
```

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -f blogURL.ldif
Enter bind password:
modifying entry cn=schema

$
```

在生產環境中，您必須提供有效且唯一的 OID，而非 1.2.3.4.5.6.7。

檢視屬性類型

cn=schema 項目具有多值屬性 attributeTypes，此屬性含有目錄模式中各種屬性類型的定義。您可以使用 ldapsearch(1) 指令讀取這些定義。

▼ 檢視屬性類型

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **使用 ldapsearch 指令，檢視目前位於您目錄模式中的所有屬性類型定義。**

範例 12–2 檢視屬性類型

下列指令可顯示所有屬性類型的定義：

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" attributeTypes
```

-T 選項可使 ldapsearch 指令不會進行 LDIF 換行，而使您能夠更容易地使用 grep 或 sed 之類的指令處理輸出。若您接著透過 grep 指令對此指令的輸出使用管道符號，就只能檢視目錄模式內使用者定義的延伸。例如：

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" attributeTypes | grep "user defined"
 attributeTypes: ( 1.2.3.4.5.6.7 NAME ( 'blog' 'blogURL' )
 DESC 'URL to a personal weblog'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
 X-ORIGIN 'user defined' )
```

刪除屬性類型

cn=schema 項目具有多值屬性 attributeTypes，此屬性含有目錄模式中各種屬性類型的定義。您可以使用 ldapmodify(1) 指令刪除具有 X-ORIGIN 'user defined' 的定義。

模式係由 LDAP 檢視定義於 cn=schema 中，因此您可以使用 ldapsearch 與 ldapmodify 公用程式，以線上方式檢視及修改模式。但您只能刪除在 X-ORIGIN 欄位中具有 'user defined' 值的模式元素。伺服器將不會刪除其他定義。

您對使用者定義的屬性所做之變更，會儲存在 99user.ldif 檔案中。

▼ 刪除屬性類型

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **檢視所要刪除之屬性類型的定義。**
   如需詳細資訊，請參閱第 269 頁的 「檢視屬性類型」。

2. **使用 ldapmodify(1) 指令，刪除模式中所出現的屬性類型定義。**

範例 12–3 刪除屬性類型

下列指令將刪除範例 12–1 中所建立的屬性類型：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=schema
changetype: delete
delete: attributeTypes
attributeTypes: ( 1.2.3.4.5.6.7 NAME ( 'blog' 'blogURL' )
 DESC 'URL to a personal weblog'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
 X-ORIGIN 'user defined' )
^D
```

請注意，您必須納入 X-ORIGIN 'user defined' 這項由目錄伺服器增加，而用以將此模式定義歸類爲延伸的屬性。

透過 LDAP 管理物件類別

本節說明如何透過 LDAP 建立、檢視及刪除物件類別。

建立物件類別

cn=schema 項目具有多值屬性 objectClasses，此屬性含有目錄模式中各種物件類別的定義。您可以使用 ldapmodify(1) 指令增加至這些定義。

新的物件類別定義，以及您對使用者定義的物件類別所做的變更，都會儲存在 99user.ldif 檔案中。

若要建立數個繼承其他物件類別的物件類別，您必須先建立父系物件類別。若您的新物件類別使用自訂屬性，則也必須先定義這些屬性。

對於每個物件類別定義，必須至少提供一個 OID。請考慮對新的物件類別至少使用下列元素：

- **物件類別 OID**。對應至您物件類別的物件識別碼。OID 是一個可唯一識別模式物件的字串，通常爲小數點十進位數字。

  如需嚴格的 LDAP v3 規範遵循，則必須提供有效數值 OID。若想進一步瞭解 OID，或要爲您的企業申請前綴，請將電子郵件寄至 IANA (網際網路位址指派機構) `iana@iana.org`，或造訪 IANA 網站 (`http://www.iana.org`)。
- **物件類別名稱**。對應至物件類別的唯一名稱。
- **父系物件類別**。此物件類別從中繼承屬性的現有物件類別。

  若不讓此物件類別繼承其他特定物件類別，請使用 `top`。

  一般說來，若要爲使用者項目新增屬性，父系會是 `inetOrgPerson` 物件類別。若要爲公司項目新增屬性，父系通常是 `organization` 或 `organizationalUnit`。若要爲群組項目新增屬性，父系通常是 `groupOfNames` 或 `groupOfUniqueNames`。
- **必要屬性**。列出及定義此物件類別**必須**具備的屬性。
- **允許的屬性**。列出及定義此物件類別可具備的其他屬性。

▼ 建立物件類別

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **根據 RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`) 中所指定的語法，備妥您的物件類別定義。**

2. **使用 `ldapmodify`(1) 指令，增加您的物件類別定義。**

   請注意，目錄伺服器會將 `X-ORIGIN 'user defined'` 增加到您所提供的定義中。

範例 12–4 建立物件類別

下列範例使用 `ldapmodify` 指令新增物件類別。

```
$ cat blogger.ldif
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 1.2.3.4.5.6.8
 NAME 'blogger'
 DESC 'Someone who has a blog'
 SUP inetOrgPerson
 STRUCTURAL
 MAY blog )
```

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -f blogger.ldif
Enter bind password:
modifying entry cn=schema

$
```

在生產環境中，您必須提供有效且唯一的 OID，而非 1.2.3.4.5.6.8。

檢視物件類別

cn=schema 項目具有多值屬性 objectClasses，此屬性含有目錄模式中各種物件類別的定義。您可以使用 ldapsearch(1) 指令讀取這些定義。

▼ 檢視物件類別

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **使用 ldapsearch 指令，檢視目前位於您目錄模式中的所有物件類別定義。**

範例 12–5 檢視物件類別

下列指令可顯示所有物件類別的定義：

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" objectClasses
```

-T 選項可使 ldapsearch 指令不會進行 LDIF 換行，而使您能夠更容易地使用 grep 或 sed 之類的指令處理輸出。若您接著透過 grep 指令對此指令的輸出使用管道符號，就只能檢視目錄模式內使用者定義的延伸。例如：

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" objectClasses | grep "user defined"
objectClasses: ( 1.2.3.4.5.6.8 NAME 'blogger'
 DESC 'Someone who has a blog' STRUCTURAL MAY blog
 X-ORIGIN 'user defined' )
$
```

刪除物件類別

cn=schema 項目具有多值屬性 objectClasses，此屬性含有目錄模式中各種物件類別的定義。您可以使用 ldapmodify(1) 指令刪除具有 X-ORIGIN 'user defined' 的定義。

模式係由 LDAP 檢視定義於 cn=schema 中，因此您可以使用 ldapsearch 與 ldapmodify 公用程式，以線上方式檢視及修改模式。但您只能刪除在 X-ORIGIN 欄位中具有 'user defined' 值的模式元素。伺服器將不會刪除其他定義。

您對使用者定義的元素所做的變更，會儲存在 99user.ldif 檔案中。

▼ 刪除物件類別

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **檢視要刪除之物件類別的定義。**
   如需詳細資訊，請參閱第 272 頁的 「檢視物件類別」。
2. **使用 ldapmodify(1) 指令，刪除模式中所出現的物件類別定義。**

範例 12–6 刪除物件類別

下列指令將刪除範例 12–4中所建立的物件類別：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=schema
changetype: delete
delete: objectClasses
objectClasses: ( 1.2.3.4.5.6.8 NAME 'blogger' DESC 'Someone who has a blog'
 STRUCTURAL MAY blog X-ORIGIN 'user defined' )
^D
```

請注意，您必須納入 X-ORIGIN 'user defined' 這項由目錄伺服器增加，而用以將此模式定義歸類爲延伸的屬性。

延伸目錄伺服器模式

當您新增屬性至模式時，必須建立含有新屬性的新物件類別。雖然直接將屬性增加到已含有您所需之大部分屬性的現有物件類別中，似乎是很方便的做法，但此舉卻會危及與 LDAP 用戶端之間的互通操作。

目錄伺服器與現有 LDAP 用戶端之間的互通操作，依賴於標準 LDAP 模式。若您變更標準模式，也將在升級伺服器時遇到困難。同理，您亦不可刪除標準模式元素。

目錄伺服器模式儲存在 cn=schema 項目的屬性中。與配置項目相同，這也是可在伺服器啓動期間從檔案讀取的 LDAP 模式檢視。

您用以延伸目錄伺服器模式的方法，取決於您是否要控制模式延伸儲存時所使用的檔案名稱。此外也取決於您是否要透過複寫發送變更給用戶。請參閱下表，以根據您特定的情況決定所要執行的程序。

表 12–1 延伸模式的方式

作業	指示
您不使用複寫。想增加自訂模式檔案，藉以延伸模式。	第 275 頁的 「使用自訂模式檔案延伸模式」
您希望透過 LDAP 延伸模式。	第 275 頁的 「透過 LDAP 延伸模式」
您要使用複寫。想在所有伺服器上保留自訂模式檔案的檔案名稱。	第 275 頁的 「使用自訂模式檔案延伸模式」
您要使用複寫。想在主伺服器複本上增加自訂模式檔案，藉以延伸模式。接著，要讓複寫機制將模式延伸複製到用戶伺服器上。	第 276 頁的 「使用模式檔案與複寫延伸模式」

如需有關物件類別、屬性與目錄模式的更多資訊，以及延伸模式的指導方針，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中的「Designing a Directory Schema」。如需有關標準屬性與物件類別的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Man Page Reference」。

本節針對可延伸目錄模式的多種方法提供相關資訊。

使用自訂模式檔案延伸模式

模式檔案是位於 *instance-path*/config/schema/ 中的 LDIF 檔案。*instance-path* 對應於目錄伺服器實例所在的檔案系統目錄。例如，實例可能位於 /local/ds/ 中。這些檔案可定義標準模式，供目錄伺服器以及所有依賴目錄伺服器的伺服器使用。這些檔案與標準模式說明於「Sun Java System Directory Server Enterprise Edition 6.3 Reference」與「Sun Java System Directory Server Enterprise Edition 6.3 Man Page Reference」中。

模式檔案只會在伺服器啓動時讀取一次。檔案的 LDIF 內容會增加至 cn=schema 中模式的常駐記憶體 LDAP 檢視。模式定義的順序是很重要的，因此模式檔案名稱的開頭處會加上號碼，並依據字母數字順序載入。只有安裝期間所定義的系統使用者，才能寫入此目錄中的模式檔案。

直接在 LDIF 檔案中定義模式時，請勿於 X-ORIGIN 欄位中使用 'user defined' 值。此值必須保留給透過 cn=schema 的 LDAP 檢視所定義，同時出現在檔案 99user.ldif 中的模式元素使用。

99user.ldif 檔案含有其他 ACI，供 cn=schema 項目與所有使用指令行或 DSCC 增加的模式定義使用。新增模式定義時，即會覆寫 99user.ldif 檔案。若要修改此檔案，您必須立即重新啓動伺服器，以確保更新目前的變更。

請勿修改其他模式檔案中所定義的標準模式。但您可以新增檔案以定義新的屬性與物件類別。例如，若要在多部伺服器上定義新的模式元素，您可以先將元素定義於名爲 98mySchema.ldif 的檔案中，再將此檔案複製到所有伺服器的模式目錄中。接著您必須重新啓動所有伺服器，以載入新的模式檔案。

▼ 使用自訂模式檔案延伸模式

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **建立您自己的模式定義檔案，如 `98mySchema.ldif`。**
   模式檔案中的定義所使用之語法，說明於 RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`) 中。

2. **(可選擇) 若此伺服器是可傳送更新至其他伺服器的主伺服器複本，請將您的模式定義檔案複製到複寫拓樸中的每個伺服器實例上。**
   複寫機制無法偵測您對含有模式的 LDIF 檔案所做的任何直接變更。因此，即使在重新啓動主伺服器後，您的變更仍不會複寫至用戶。

3. **請重新啓動已複製了模式定義檔案之每個目錄伺服器實例。**
   您的變更將在伺服器重新啓動後生效，並重新載入模式定義。

透過 LDAP 延伸模式

模式係由 LDAP 檢視定義於 `cn=schema` 中，因此您可以使用 `ldapsearch` 與 `ldapmodify` 公用程式，以線上方式檢視及修改模式。但您只能修改在 `X-ORIGIN` 欄位中具有 `'user defined'` 值的模式元素。伺服器會拒絕對其他定義的任何修改。

新的元素定義，以及您對使用者定義的元素所做的變更，都會儲存在 `99user.ldif` 檔案中。

▼ 透過 LDAP 延伸模式

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

開始之前 從指令行修改模式定義很可能會發生錯誤，因爲您必須一字不漏地鍵入冗長的值。但您可以在更新目錄模式時所需的程序檔中使用此功能。

1. **使用 `ldapmodify(1)` 指令，增加或刪除個別的 `attributeTypes` 屬性值。**
   如需詳細資訊，請參閱第 268 頁的 「建立屬性類型」或第 270 頁的 「刪除屬性類型」。

2. **使用 `ldapmodify(1)` 指令，增加或刪除個別的 `objectClasses` 屬性值。**
   如需詳細資訊，請參閱第 271 頁的 「建立物件類別」或第 273 頁的 「刪除物件類別」。

另請參閱 若要修改其中一個值，必須先刪除該特定值，再將其增加爲新值。之所以必須這麼做，是因爲這些屬性是多值屬性。如需詳細資訊，請參閱第 91 頁的 「修改多值屬性的某個值」。

使用模式檔案與複寫延伸模式

如需有關自訂模式檔案的資訊，請參閱第 274 頁的 「使用自訂模式檔案延伸模式」。下列程序說明如何使用複寫機制，將模式延伸傳播至拓樸中的所有伺服器。

▼ 使用模式檔案與複寫延伸模式

針對此程序的某些部分，您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。此程序的其他部分只能使用指令行完成。

1. **以下列其中一種方式備妥您的模式延伸：**

   - **建立您自己的模式定義檔案，如** `98mySchema.ldif`。
   - **將模式延伸增加到** `99user.ldif` **中。**

   模式檔案中的定義所使用之語法，說明於 RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`)。

2. **在存放模式定義檔案的主伺服器上，執行** `schema_push` **指令。**

   此程序檔不會實際將模式發送至複本。此程序檔會將特殊的屬性寫入模式檔案中，使模式檔案在載入時隨即進行複寫。如需更多資訊，請參閱 `schema_push`(1M) 線上手冊。

3. **重新啓動存放模式定義檔案的主伺服器。**

   複寫機制無法偵測您對含有模式的 LDIF 檔案所做的任何直接變更。當您在執行 `schema_push` 後重新啓動伺服器時，伺服器會載入所有模式檔案，接著複寫機制會將新的模式複寫至用戶。

複寫目錄模式

每當您在兩部伺服器之間配置一或多個尾碼的複寫時，模式定義也會自動複寫。模式定義的自動複寫可確保所有複本均具有完整且相同的模式，並以此模式定義所有可複寫至用戶的物件類別與屬性。主伺服器因此也含有主伺服器模式。

但模式複寫並不是即時的，即使您透過 LDAP 修改模式也是如此。目錄資料更新時，或在模式修改後第一次啓動複寫階段作業時，皆會觸發模式複寫。

若要對所有複本執行模式，至少必須對所有主伺服器啓用模式檢查。在執行 LDAP 作業的主伺服器上檢查模式時，模式並不需在用戶更新後進行檢查。爲提昇效能，複寫機制會略過對用戶複本的模式檢查。

備註 – 請勿關閉集散中心與專屬用戶的模式檢查。模式檢查並不會影響用戶的效能。請持續開啓模式檢查，以指出複本內容是否符合其模式。

主伺服器會在用戶初始化期間自動將模式複寫至用戶。主伺服器也會在透過 DSCC 或透過指令行工具修改模式時，自動複寫模式。依預設會複寫整個模式。任何在用戶上尚不存在的其他模式元素，皆會在用戶上建立，並儲存於 `99user.ldif` 檔案中。

例如，假設主伺服器在啓動時於 `98mySchema.ldif` 檔案中含有模式定義。此外同時假設，您接下來將定義與其他伺服器 (主伺服器、集散中心或專屬用戶) 的複寫協議。當您後續由此主伺服器初始化複本時，複寫的模式將含有來自 `98mySchema.ldif` 的定義，但定義會儲存在複本伺服器上的 `99user.ldif` 中。

於用戶初始化期間複寫模式後，若在主伺服器上修改 `cn=schema` 中的模式，也將使整個模式複寫至用戶。因此，透過指令行公用程式或 DSCC 對主伺服器模式所做的任何修改，均會複寫至用戶。這些修改會儲存在主伺服器的 `99user.ldif` 中，而透過前述的相同機制，這些修改亦可儲存在用戶的 `99user.ldif` 中。

請考量下列在複寫環境中維護模式一致性的指示：

- 請勿修改用戶伺服器上的模式。

  修改用戶伺服器上的模式可能會導致複寫錯誤。這是因爲用戶上的模式差異，可能會導致來自供應者的更新不符合用戶上的模式。
- 在多重主伺服器複寫環境中，請修改單一主伺服器上的模式。

  當您修改兩部主伺服器的模式時，最新更新的主伺服器會將其模式版本傳播至用戶。用戶上的模式因此可能會變得與其他主伺服器上的模式不一致。

配置**部分複寫**時，需同時考量下列事項：

- 當供應者在部分複寫配置中發送模式時，部分用戶複本上的模式將是主伺服器複本之模式的副本。因此，模式將可能無法對應於所套用的部分複寫配置。
- 一般而言，目錄伺服器會依照模式中的定義複寫每個項目的所有必要屬性，以避免違反模式。當您配置部分複寫而排除必要屬性時，必須停用模式檢查。
- 若對部分複寫啓用模式檢查，可能無法以離線方式初始化複本。若排除必要屬性，目錄伺服器將不允許您從 LDIF 載入資料。
- 若您對部分用戶複本停用了模式檢查，部分用戶複本所在的整個伺服器實例，都將不會執行模式檢查。因此，請避免將相同伺服器實例上的供應者複本配置爲部分用戶。

限制模式複寫

複寫機制每次複寫模式時，預設會將整個模式傳送至用戶。以下兩種情況不適合將整個模式傳送至用戶：

- 使用 DSCC 或從指令行對 `cn=schema` 所進行的修改，僅限於使用者定義的模式元素，所有標準模式皆不會變更。若您經常修改模式，在每次傳送大量的未變更模式元素時，都會使得效能受到影響。您可以僅就使用者定義的模式元素進行複寫，而提昇複寫與伺服器的效能。
- 當目錄伺服器上的主伺服器複寫至 Directory Server 5.1 上的用戶時，這些版本的配置屬性模式將不會相同，而會產生衝突。在此情況下，您**必須**僅就使用者定義的模式元素進行複寫。

備註 – 目錄伺服器使用 `11rfc2307.ldif` 模式檔案。此模式檔案符合 RFC 2307 (`http://www.ietf.org/rfc/rfc2307.txt`)。

Directory Server 5.2 之前的目錄伺服器版本使用 `10rfc2307.ldif` 模式檔案。

▼ 限制模式複寫

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **將模式複寫限定為僅複寫使用者定義的模式。**

```
$ dsconf set-server-prop -h host -p port repl-user-schema-enabled:on
```

預設值 `off` 會在必要時進行整個模式的複寫。

◆ ◆ ◆ 第 13 章

13 目錄伺服器編製索引

目錄伺服器的索引如同書中的索引一般，經由建立搜尋字串與目錄內容參照的關聯可加速搜尋。

如需有關索引類型與索引調校的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 6 章「Directory Server Indexing」。

本章包含下列主題：

- 第 279 頁的「管理索引」
- 第 285 頁的「管理瀏覽索引」

管理索引

本節說明如何管理特定屬性的索引，包含建立、修改與刪除索引的相關資訊。如需虛擬清單檢視 (VLV) 作業特有的程序，請參閱第 285 頁的「管理瀏覽索引」。

▼ 列出索引

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

- **若要列出現有的索引與其特性，請使用此指令：**

```
$ dsconf list-indexes -h host -p port -v suffix-DN
```

▼ 建立索引

備註 – 您無法建立新的系統索引。僅會保留目錄伺服器內部定義的現有系統索引。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **建立新的索引配置。**

   使用 `dsconf create-index` 指令行公用程式，指定要編製索引的屬性，以配置新的索引資訊。

   例如，若要建立 `preferredLanguage` 屬性的索引項目，請使用此指令：

   ```
   $ dsconf create-index -h host -p port dc=example,dc=com preferredLanguage
   ```

   備註 – 指令 `dsconf create-index` 會設定索引配置，但是不會實際建立搜尋所需的索引檔案。產生索引檔案會影響效能。在建立新的索引配置之後手動產生索引檔案，可讓您在編製索引程序期間有更多的控制權。

   建立索引時，請一律使用屬性的主要名稱。請勿使用屬性的別名。屬性的主要名稱爲模式中所列的第一個屬性名稱，例如 `userid` 屬性的主要名稱爲 `uid`。

2. **(可選擇) 使用 `dsconf set-index-prop` 指令設定索引特性。**

   `dsconf create-index` 指令會以預設特性建立索引。若要修改這些特性，請使用 `dsconf set-index-prop` 指令。如需修改索引特性的詳細資訊，請參閱第 280 頁的 「修改索引」。

3. **產生索引檔案。**

   請參閱第 281 頁的 「產生索引」。

4. **針對所有要編製索引的伺服器重複上述步驟。**

▼ 修改索引

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **修改索引特性。**

   ```
   $ dsconf set-index-prop -h host -p port suffix-DN attr-name property:value
   ```

例如，若要啓用 preferredLanguage 索引的近似索引 approx-enabled，請使用此指令：

```
$ dsconf set-index-prop -h host -p port dc=example,dc=com preferredLanguage approx-enabled:on
```

您可以修改各個索引的下列特性：

- eq-enabled 相同
- pres-enabled 存在
- sub-enabled 子字串

可能需要修改的屬性之一爲 選用的 nsMatchingRule 屬性。此屬性包含伺服器已知的所有相符規則之 OID。它會啓用國際化索引的語言比較順序之 OID，以及 CaseExactMatch 等其他的相符規則。如需支援的語言環境清單及其相關比較順序的 OID，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」。

如需有關索引配置屬性的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」。

2 **重新產生新的索引。**
請參閱第 281 頁的 「產生索引」。

3 **針對所有包含已修改屬性的索引之伺服器重複上述步驟。**

▼ 產生索引

本程序會產生索引檔案，以讓新的或修改過的索引可進行搜尋。如果修改某個屬性的索引配置，所有以此屬性作爲篩選的搜尋都將視爲未編製索引。若要確保包含該屬性的搜尋能成功，請使用此程序指令在每次建立或修改屬性的索引配置時，重新產生現有的索引。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **依下列其中一種方式產生索引檔案：**
  - 以線上方式產生新的索引檔案。

    ```
    $ dsconf reindex -h host -p port [-t attr] suffix-DN
    ```

    其中 -t 會指定僅能重新編製特定單一屬性或特定多重屬性的索引，而不是所有屬性。

    例如，若要重新產生 preferredLanguage 索引，請鍵入：

    ```
    $ dsconf reindex -h host -p port -t preferredLanguage dc=example,dc=com
    ```

`dsconf reindex` 指令執行期間，可透過伺服器使用尾碼內容。但是在指令完成之前，無法編製搜尋的索引。重新編製索引是相當耗費資源之作業，可能會影響到伺服器上其他作業的效能。

- 以離線方式產生新的索引檔案。

  ```
  $ dsadm reindex -t attr instance-path suffix-DN
  ```

  例如，若要重新產生 `preferredLanguage` 索引，請鍵入：

  ```
  $ dsadm reindex -t preferredLanguage /local/ds dc=example,dc=com
  ```

- 重新初始化尾碼可在離線狀態下快速重新產生所有的索引。

  重新初始化尾碼時，會自動重新產生所有索引檔案。根據目錄大小的不同，重新初始化尾碼一般會比重新編製兩個或兩個以上的索引還要快。但是，初始化期間無法使用尾碼。如需更多資訊，請參閱第 284 頁的 「重新初始化以重新編製尾碼的索引」。

備註 – 如果在多個尾碼上平行執行 `dsconf import`、`dsconf reindex` 或同時執行兩個指令，作業事件記錄會變大而可能對效能造成不良影響。

▼ 刪除索引

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **移除為屬性配置的所有索引。**

  ```
  $ dsconf delete-index -h host -p port suffix-DN attr-name
  ```

  例如，下列指令會刪除 `preferredLanguage` 屬性的所有索引：

  ```
  $ dsconf delete-index -h host -p port dc=example,dc=com preferredLanguage
  ```

  刪除預設索引時請特別小心，此舉會影響目錄伺服器運作。

變更索引清單臨界值

搜尋速度緩慢可能肇因於系統索引清單大小超過了索引清單臨界值。索引清單臨界值是每個索引鍵值的上限數。若要判定是否已超過索引清單臨界值，請檢查存取記錄。存取記錄 `RESULT` 訊息結尾的 `notes=U` 旗標表示執行了未編製索引的搜尋。相同連線與作業的前一則 `SRCH` 訊息會指定使用的搜尋篩選。下列兩行範例會追蹤未編製索引的搜尋 `cn=Smith`，並傳回 10,000 個項目。已從訊息移除時間戳記。

```
conn=2 op=1 SRCH base="o=example.com" scope=0 filter="(cn=Smith)"
conn=2 op=1 RESULT err=0 tag=101 nentries=10000 notes=U
```

如果系統經常會超過索引清單臨界值，請考慮提高臨界值以改善效能。下列程序使用 `dsconf set-server-prop` 指令修改 `all-ids-threshold` 特性。如需有關調校索引與 `all-ids-threshold` 特性的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Tuning Indexes for Performance」。

▼ 變更索引清單臨界值

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 調整索引清單臨界值。

您可以在下列任何一個層級調整索引清單臨界值：

- 實例層級：

  ```
  dsconf set-server-prop -h host -p port all-ids-threshold:value
  ```

- 尾碼層級：

  ```
  dsconf set-suffix-prop -h host -p port suffix-DN all-ids-threshold:value
  ```

- 項目層級：

  ```
  dsconf set-index-prop -h host -p port suffix-DN all-ids-threshold:value
  ```

- 依搜尋類型的索引層級：

  ```
  dsconf set-index-prop -h host -p port suffix-DN all-ids-threshold search-type:value
  ```

  其中 *search-type* 爲下列其中之一：

  - `eq-enabled` 相同
  - `pres-enabled` 存在
  - `sub-enabled` 子字串

  近似索引無法配置 `all-ids-threshold` 特性。

您可以在依搜尋類型的索引層級使用 DSCC，設定臨界值。如需更多資訊，請參閱目錄伺服器線上說明。

2 重新產生尾碼索引。

請參閱第 281 頁的 「產生索引」。

3 如果已為舊的所有 `ID` 臨界值調校資料庫的快取大小，且伺服器有足夠的實體記憶體，請考慮加資料庫的快取大小。

根據**所有 `ID` 臨界值**增量範圍的 25%，增加資料庫的快取大小。

換言之，如果將**所有 `ID` 臨界值**從 4000 增加到 6000，便可將資料庫快取大小增加約 12.5%，以便容納增加的索引清單大小。

資料庫快取大小使用屬性 dbcachesize 進行設定。憑經驗尋找最適大小，再將變更套用到生產伺服器。

重新編製尾碼的索引

若索引檔案損毀或變更了某項屬性的索引，即必須重新編製尾碼的索引，以在對應的資料庫目錄中重建索引檔案。您可以在目錄伺服器執行時，或透過重新初始化尾碼來重新編製尾碼的索引。

在目錄伺服器執行期間重新編製尾碼的索引

重新編製尾碼的索引時，伺服器會檢查尾碼包含與重建索引檔案的所有項目。尾碼的內容在重新編製索引期間爲唯讀。由於伺服器必須掃描整個尾碼以尋找要重新編製索引的每個屬性，因此若尾碼有百萬個項目，本程序可能需要數小時。時間長度也會隨配置的索引而定。此外，在重新編製尾碼的索引期間無法使用索引，且伺服器的效能會受影響。

▼ 重新編製尾碼上的所有索引

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **重新編製尾碼上的所有索引。**

  ```
  $ dsconf reindex -h host -p port suffix-DN
  ```

  例如，若要初始化 dc=example,dc=com 尾碼上的所有索引，請使用此指令：

  ```
  $ dsconf reindex -h host -p port dc=example,dc=com
  ```

重新初始化以重新編製尾碼的索引

當您重新初始化尾碼時，會匯入新的內容，這表示會取代尾碼內容並建立新的索引檔案。由於所有屬性會在項目載入時平行編製索引，因此重新初始化尾碼會比重新編製多個屬性的索引還要快。但是請注意，重新初始化尾碼期間將無法使用尾碼。

▼ 透過重新初始化重新編製尾碼的索引

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **如第 64 頁的 「設定參照並使尾碼成為唯讀模式」中所述，將尾碼設為唯讀。**
2. **如第 190 頁的 「備份至 LDIF」中所述，將整個尾碼匯出至 LDIF 檔案。**

3. **如第 192 頁的 「從 LDIF 檔案匯入資料」中所述，匯入相同的 LDIF 檔案以重新初始化尾碼。**
   初始化期間無法使用尾碼。初始化完成之後，便可以使用所有配置的索引。

4. **如第 64 頁的 「設定參照並使尾碼成為唯讀模式」中所述，使尾碼再次可以寫入。**

管理瀏覽索引

瀏覽索引是特殊的索引，僅用於請求伺服器端排序結果的搜尋作業。「Sun Java System Directory Server Enterprise Edition 6.3 Reference」說明瀏覽索引在目錄伺服器中的運作方式。

用戶端搜尋的瀏覽索引

必須手動定義用於排序用戶端搜尋結果的自訂瀏覽索引。若要建立瀏覽索引或虛擬清單檢視 (VLV) 索引，請使用下列程序。本節亦包含增加或修改瀏覽索引項目與重新產生瀏覽索引的程序。

▼ 建立瀏覽索引

針對此程序的某些部分，您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。此程序的其他部分只能使用指令行完成。

1. **使用 `ldapmodify` 指令新增瀏覽索引項目或編輯現有的瀏覽索引項目。**
   如需相關指示，請參閱第 285 頁的 「增加或修改瀏覽索引項目」。

2. **執行 `dsconf reindex` 指令以產生一組新的瀏覽索引保留在伺服器上。**
   如需相關指示，請參閱第 287 頁的 「重新產生瀏覽索引」。

▼ 增加或修改瀏覽索引項目

瀏覽索引專用於指定基底項目與其子樹狀結構上指定的搜尋。瀏覽索引配置會在包含項目的尾碼資料庫配置中定義。

1. **配置目錄伺服器上每個瀏覽索引的 `vlvBase`、`vlvScope` 與 `vlvFilter` 屬性。**
   這些屬性會配置搜尋的基底、範圍與篩選器。這些屬性使用 `vlvSearch` 物件類別。

2. **配置每個瀏覽索引的 `vlvSort` 屬性。**
   此屬性會指定排序索引的一或多個屬性名稱。此項目是第一個項目的子項，並使用 `vlvIndex` 物件類別指定要排序的屬性與排序的順序。

下列範例使用 ldapmodify 指令建立瀏覽索引配置項目：

```
$ ldapmodify -a -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=people_browsing_index, cn=database-name,
cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvSearch
cn: Browsing ou=People
vlvBase: ou=People,dc=example,dc=com
vlvScope: 1
vlvFilter: (objectclass=inetOrgPerson)

dn: cn=Sort rev employeenumber, cn=people_browsing_index,
 cn=database-name,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvIndex
cn: Sort rev employeenumber
vlvSort: -employeenumber
^D
```

vlvScope 爲下列其中之一：

- 0 僅針對基底項目
- 1 可針對該基底的下一層子項
- 2 可針對根目錄在該基底的整個子樹狀結構

vlvFilter 是用戶端搜尋作業中所用相同的 LDAP 篩選。由於所有瀏覽索引項目皆位於相同位置，因此您應使用描述性 cn 值命名瀏覽索引。

每個 vlvSearch 項目至少須有一個 vlvIndex 項目。vlvSort 屬性是定義要排序的屬性與排序順序之屬性名稱的清單。屬性名稱之前的破折號 (-) 表示反向順序。您可以定義數個 vlvIndex 項目，以爲搜尋定義多個索引。您可以爲之前的範例增加下列項目：

```
$ ldapmodify -a -h host -p port
 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Sort sn givenname uid, cn=people_browsing_index,
 cn=database-name,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvIndex
cn: Sort sn givenname uid
vlvSort: sn givenname uid
^D
```

3 **若要修改瀏覽索引配置，請編輯對應的 vlvSearch 項目或對應的 vlvIndex 項目。**

4 **若要移除瀏覽索引而使得伺服器不再保留該瀏覽索引，請移除個別 vlvIndex 項目。**
或者，如果僅存在一個 vlvIndex 項目，則同時移除 vlvSearch 項目與 vlvIndex 項目。

▼ 重新產生瀏覽索引

- **建立瀏覽索引項目之後，請產生指定屬性的新瀏覽索引。**

  ```
  $ dsadm reindex -l -t attr-index instance-path suffix-DN
  ```

  該指令會掃描目錄內容，並建立瀏覽索引的資料庫檔案。

  下列範例會產生上一節定義的瀏覽索引：

  ```
  $ dsadm reindex -l -b database-name -t Browsing /local/ds \
   ou=People,dc=example,dc=com
  ```

  如需 `dsadm reindex` 指令的更多資訊，請參閱 `dsadm`(1M) 線上手冊。

◆ ◆ ◆ 第 14 章

14 目錄伺服器屬性值唯一性

UID 唯一性外掛程式可確保指定屬性的值在目錄或子樹狀結構的所有項目之間皆是唯一的。若有任何作業嘗試增加含有現有指定屬性值的項目，此外掛程式均會加以阻止。此外掛程式也會阻止任何增加或修改目錄中已有之屬性值的作業。

UID 唯一性外掛程式預設爲停用。此外掛程式啓用時，預設會確保 `uid` 屬性的唯一性。您可以建立新的外掛程式實例，以執行其他屬性值的唯一性。UID 唯一性外掛程式可確保單一伺服器上的屬性值唯一性。

本章包含下列主題：

- 第 289 頁的 「屬性值唯一性簡介」
- 第 290 頁的 「執行 `uid` 與其他屬性的唯一性」
- 第 292 頁的 「對複寫使用唯一性外掛程式」

屬性值唯一性簡介

UID 唯一性外掛程式是一種作業前外掛程式。它可在伺服器執行目錄的更新前，檢查 LDAP 增加、修改與修改 DN 作業。外掛程式可判斷作業是否會造成兩個項目具有相同屬性值。若是，伺服器即會終止作業，並傳回錯誤 19 `LDAP_CONSTRAINT_VIOLATION` 至用戶端。

您可以配置外掛程式，以在目錄的一或多個子樹狀結構中，或在特定物件類別的項目間執行唯一性。此配置可決定要執行唯一屬性值的項目集。

若要執行其他屬性的唯一性，您可以定義數個 UID 唯一性外掛程式實例。每個需要具備唯一值的屬性，各須定義一個外掛程式實例。您也可以爲同一個屬性設定數個外掛程式實例，而在數個項目集內執行「個別的」唯一性。一個指定的屬性值在每個子樹狀結構集內只能允許一次。

當您在現有的目錄上啓用屬性唯一性時，伺服器並不會檢查現有項目間的唯一性。只有在增加項目，或於增加或修改屬性時，才會執行唯一性。

UID 唯一性外掛程式預設爲停用，因爲此外掛程式會影響多重主伺服器複寫。您可以在使用複寫時啓用 UID 唯一性外掛程式，但應留意第 292 頁的 「對複寫使用唯一性外掛程式」中所述的運作方式。

執行 uid 與其他屬性的唯一性

本節說明如何啓用與配置 uid 屬性的預設唯一性外掛程式，以及如何強制執行任何其他屬性的唯一性。

▼ 執行 uid 屬性的唯一性

此程序說明如何使用 dsconf 指令，啓用及配置 UID 唯一性外掛程式。外掛程式配置項目的 DN 爲 cn=uid uniqueness,cn=plugins,cn=config。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

使用 DSCC 期間，不可於執行其他屬性的唯一性時，修改預設 UID 唯一性外掛程式。若不需要 UID 唯一性外掛程式，請如第 291 頁的 「執行其他屬性的唯一性」中所述，將該外掛程式保留爲停用，再爲其他屬性建立新的外掛程式實例。

1. **啓用外掛程式。**

```
$ dsconf enable-plugin -h host -p port "uid uniqueness"
```

2. **根據您對要執行唯一性的子樹狀結構所將進行的指定，修改外掛程式引數。**

   - 若要指定單一子樹狀結構的基底 DN，請鍵入：

```
$ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument:uid argument:subtreeBaseDN
```

     例如：

```
$ dsconf set-plugin-prop -h host1 -p 1389 "uid uniqueness" argument:uid \
 argument:dc=People,dc=example,dc=com
```

   - 若要指定多個子樹狀結構，請以子樹狀結構的完整基底 DN，將更多引數增加爲各個引數的值。

```
$ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument:uid \
 argument:subtreeBaseDN argument:subtreeBaseDN
```

   - 若子樹狀結構要根據其基底項目的物件類別進行指定，請將引數設爲下列值。uid 屬性的唯一性會在每個具有 *baseObjectClass* 之項目的子樹狀結構中執行。您可以選擇在第三個引數中指定 *entryObjectClass*，使外掛程式僅在以具有此物件類別的項目爲目標的作業中，執行唯一性。

```
$ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument:attribute=uid \
 argument:markerObjectClass=baseObjectClass argument:entryObjectClass=baseObjectClass
```

- 若要將引數增加至現有的引數清單中，請使用下列指令：

```
$ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument+:argument-value
```

3 重新啓動伺服器，使變更生效。

▼ 執行其他屬性的唯一性

UID 唯一性外掛程式可用以執行任何屬性的唯一性。您必須在目錄中的 `cn=plugins,cn=config` 下建立新項目，以建立新的外掛程式實例。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 建立新的外掛程式。

```
$ dsconf create-plugin -h host -p port -H lib-path -F init-func \
 -Y type plugin-name
```

plugin-name 應爲簡短同時包含屬性名稱的說明性名稱。例如，若要建立郵件 ID 屬性的唯一性外掛程式，請使用此指令：

```
$ dsconf create-plugin -h host1 -p 1389 -H /opt/SUNWdsee/ds6/lib/uid-plugin.so \
 -F NSUniqueAttr_Init -Y preoperation "mail uniqueness"
```

2 設定外掛程式特性：

```
$ dsconf set-plugin-prop -h host -p port plugin-name property:value
```

例如，若要設定郵件唯一性外掛程式的特性：

```
$ dsconf set-plugin-prop -h host1 -p 1389 "mail uniqueness" \
 desc:"Enforce unique attribute values..." version:6.0 \
 vendor:"Sun Microsystems, Inc." depends-on-type:database
```

3 啓用外掛程式。

```
$ dsconf enable-plugin -h host -p port plugin-name
```

4 指定外掛程式引數。

這些引數取決於您對要執行唯一性的子樹狀結構將要進行的指定。

- 若一或多個子樹狀結構要根據其基底 DN 進行定義，則第一個引數必須是應具有唯一值之屬性的名稱。後續引數爲子樹狀結構的基底項目之完整 DN。

```
$ dsconf set-plugin-prop -h host -p port plugin-name argument:attribute-name \
 argument:subtreeBaseDN argument:subtreeBaseDN...
```

- 若要將引數增加至現有的引數清單中，請使用下列指令：

  ```
  $ dsconf set-plugin-prop -h host -p port plugin-name argument+:argument-value
  ```

- 若子樹狀結構要根據其基底項目的物件類別進行定義，則第一個引數必須包含 `attribute=` *attribute-name*，用以指定應具有唯一值之屬性的名稱。第二個引數必須是 *baseObjectClass*，用以決定要執行唯一性之子樹狀結構的基底項目。您可以選擇在第三個引數中指定 *entryObjectClass*，使外掛程式僅在以具有此物件類別的項目爲目標的作業中，執行唯一性。

```
$ dsconf set-plugin-prop -h host -p port plugin-name argument:attribute=attribute-name \
 argument:markerObjectClass=baseObjectClass argument:requiredObjectClass=entryObjectClass
```

在所有外掛程式引數中，= 符號之前或之後皆不可有空格。

5 重新啓動伺服器，使變更生效。

對複寫使用唯一性外掛程式

在複寫作業中執行更新時，UID 唯一性外掛程式不會對屬性值執行任何檢查。這並不會影響單一主伺服器複寫，但外掛程式無法爲多重主伺服器複寫自動執行屬性的唯一性。

單一主伺服器複寫案例

由於用戶端應用程式所做的所有修改都會在主伺服器複本上執行，因此 UID 唯一性外掛程式應在主伺服器上啓用。外掛程式應配置成在複寫的尾碼中執行唯一性。由於主伺服器可確保所需屬性的值皆是唯一的，因此您無需在用戶伺服器上啓用外掛程式。

在單一主伺服器的用戶上啓用 UID 唯一性外掛程式，並不會干擾複寫作業或正常的伺服器作業。但它可能會使效能稍微降低。

多重主伺服器複寫案例

根據設計，UID 唯一性外掛程式並不適用於多重主伺服器複寫作業。由於多重主伺服器複寫所使用的複寫模式不具嚴謹的一致性，即使在兩部伺服器上同時增加相同的屬性值，甚至雙方都啓用外掛程式，也無法偵測出此相同值。

然而，若要執行唯一性檢查的屬性是命名屬性，且已在所有主伺服器上爲相同子樹狀結構中的同一屬性啓用唯一性外掛程式，則可使用 UID 唯一性外掛程式。

當這些條件都符合時，在複寫時所發生的唯一性衝突，將會報告爲命名衝突。命名衝突必須以手動方式解決。如需更多資訊，請參閱第 257 頁的「解決常見複寫衝突」。

◆ ◆ ◆ 第 15 章

15 目錄伺服器記錄

本章說明管理目錄伺服器記錄的方式。

如需協助您定義記錄策略的資訊，請使用「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中的「Designing a Logging Strategy」之記錄策略資訊。

如需記錄檔及其內容的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 7 章「Directory Server Logging」。

本章包含下列主題：

- 第 293 頁的「記錄分析工具」
- 第 293 頁的「檢視目錄伺服器記錄」
- 第 295 頁的「配置目錄伺服器的記錄」
- 第 296 頁的「手動自動重建目錄伺服器記錄」

記錄分析工具

Directory Server Resource Kit 提供記錄分析工具 `logconv`，可讓您分析目錄伺服器存取記錄。記錄分析工具會擷取使用率統計，也會計算重大事件的出現次數。如需此工具的更多資訊，請參閱 `logconv`(1) 線上手冊。

檢視目錄伺服器記錄

您可以在預設的 *instance-path*`/logs` 檔案中直接檢視伺服器上的記錄。如果您已修改預設路徑，則可以如下使用 `dsconf` 指令尋找記錄檔位置：

```
$ dsconf get-log-prop -h host -p port log-type path
```

或者透過目錄服務控制中心 (DSCC) 檢視記錄檔。DSCC 可讓您檢視與排序記錄項目。

下圖顯示 DSCC 中的目錄伺服器存取記錄範例。

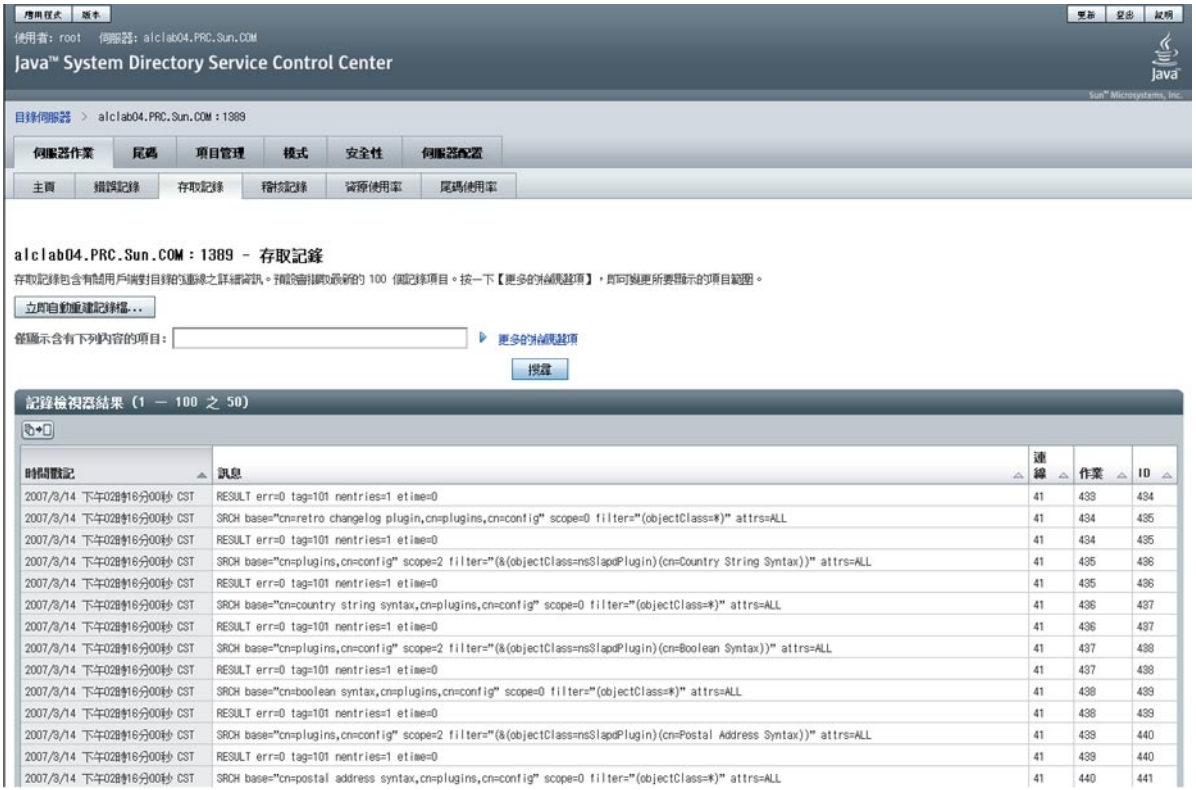


圖 15–1 DSCC 存取記錄

▼ 追蹤目錄伺服器記錄

您可以使用 `dsadm` 指令顯示目錄伺服器記錄的指定行數，或顯示指定時間以內的記錄項目。此範例追蹤錯誤記錄。若要追蹤存取記錄，請使用 `show-access-log` 取代 `show-error-log`。

1. **顯示特定時間以內的錯誤記錄項目。**

   ```
   $ dsadm show-error-log -A duration instance-path
   ```

   您必須指定持續的時間單位。例如，若要顯示 24 小時以內的錯誤記錄項目，請鍵入：

   ```
   $ dsadm show-error-log -A 24h /local/ds
   ```

2. **顯示錯誤記錄的指定行數 (從結尾處起算)。**

   ```
   $ dsadm show-error-log -L last-lines instance-path
   ```

行數以整數表示。例如，若要顯示最後 100 行，請鍵入：

```
$ dsadm show-error-log -L 100 /local/ds
```

若未指定值，預設顯示的行數爲 20 行。

配置目錄伺服器的記錄

記錄檔有許多方面可以修改。以下包含其中一些範例：

- 啓用稽核記錄

  稽核記錄與存取記錄及錯誤記錄不同，預設不會啓用。如需相關資訊，請參閱第 296 頁的「啓用稽核記錄」。
- 一般設定
  - 啓用或停用記錄
  - 啓用或停用記錄緩衝
  - 記錄檔位置
  - 詳細記錄
  - 記錄層級
- 記錄自動重建設定。
  - 定期建立新記錄
  - 建立新記錄檔前的記錄檔大小上限
- 記錄刪除設定
  - 刪除前的檔案最長存在期限
  - 刪除前的檔案大小上限
  - 刪除前的最低可用磁碟空間

下列程序說明如何修改記錄配置，以及如何啓用稽核記錄。

▼ 修改記錄配置

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1 **檢視要修改的記錄設定。**

```
$ dsconf get-log-prop -h host -p port log-type
```

例如，若要列出現有的錯誤記錄設定，請鍵入：

```
$ dsconf get-log-prop -h host1 -p 1389 error
Enter "cn=Directory Manager" password:
buffering-enabled         :  off
```

```
enabled                   :  on
level                     :  default
max-age                   :  1M
max-disk-space-size       :  100M
max-file-count            :  2
max-size                  :  100M
min-free-disk-space-size  :  5M
path                      :  /tmp/ds1/logs/errors
perm                      :  600
rotation-interval         :  1w
rotation-min-file-size    :  unlimited
rotation-time             :  undefined
verbose-enabled           :  off
```

2 **設定新值**。

設定特性要使用的值。

```
$ dsconf set-log-prop -h host -p port log-type property:value
```

例如，若要將錯誤記錄的自動重建間隔設爲兩天，請使用此指令：

```
$ dsconf set-log-prop -h host1 -p 1389 error rotation-interval:2d
```

▼ 啓用稽核記錄

稽核記錄與存取記錄及錯誤記錄不同，預設不會啓用。您必須啓用稽核記錄後，才能加以檢視。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **啓用稽核記錄**。

```
$ dsconf set-log-prop -h host -p port audit enabled:on
```

手動自動重建目錄伺服器記錄

記錄若是變得太大，則可以隨時手動自動重建記錄。自動重建會備份現有的記錄檔，然後建立一個全新的記錄檔。

▼ 手動自動重建記錄檔

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **自動重建記錄檔。**

  ```
  $ dsconf rotate-log-now -h host -p port log-type
  ```

  例如，若要自動重建存取記錄：

  ```
  $ dsconf rotate-log-now -h host1 -p 1389 access
  ```

◆ ◆ ◆ 第 16 章

16 目錄伺服器監視

目錄伺服器可利用多種方法加以監視。這些方法說明於「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 3 章「Directory Server Monitoring」。

本章說明如何設定及管理目錄伺服器中的監視。

本章包含下列主題：

- 第 299 頁的「設定目錄伺服器的 SNMP」
- 第 300 頁的「啓用 Java ES MF 監視」
- 第 301 頁的「Java ES MF 監視的疑難排解」
- 第 301 頁的「使用 cn=monitor 監視伺服器」

設定目錄伺服器的 SNMP

本節說明如何將伺服器設爲透過 SNMP 接受監視。

如需目錄伺服器中 SNMP 實作的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Directory Server and SNMP」。

▼ 設定 SNMP

針對此程序的某些部分，您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。此程序的其他部分只能使用指令行完成。

1 **啓用 Java ES Monitoring Framework 外掛程式。**

請使用程序第 300 頁的「啓用 Java ES MF 監視」。此程序亦會啓用屬於 Java ES MF 一部分的共用代理程式容器。

2 **存取由 MIB 所定義，同時透過代理程式公開的 SNMP 管理物件。**

此步驟所需的作業，完全依存於您的 SNMP 管理系統。如需相關指示，請參閱 SNMP 管理系統文件。

公開 MIB 時，您可以爲此 MIB 使用 RFC 文字檔。這些檔案可由 `http://www.ietf.org/rfc/rfc2605.txt` 與 `http://www.ietf.org/rfc/rfc2788.txt` 取得。

啓用 Java ES MF 監視

若要使用 Sun Java ES Monitoring Framework (Java ES MF) 進行監視，則必須啓用 Java ES MF 外掛程式。

如需有關管理 Java ES MF 的更多資訊，請參閱「Sun Java Enterprise System 5 Monitoring Guide」。

▼ 啓用 Java ES MF 監視

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **初始化及註冊 Java ES Monitoring Framework。**

```
$ dsccsetup mfwk-reg
```

如需此指令的位置，請參閱第 28 頁的 「指令位置」。

2 **啓用 Java ES Monitoring Framework 外掛程式。**

```
$ dsconf enable-plugin -h host -p port "Monitoring Plugin"
Enter "cn=Directory Manager" password:
Directory Server must be restarted for changes to take effect.
```

3 **重新啓動目錄伺服器實例。**

```
$ dsadm restart instance-path
```

4 **驗證是否已啓用 Java ES Monitoring Framework 外掛程式。**

```
$ dsconf get-plugin-prop -h host -p port -v "Monitoring Plugin"
Enter "cn=Directory Manager" password:
Reading property values of the plugin "Monitoring Plugin"...
argument          :
depends-on-named  :
depends-on-type   :  database
desc              :  Monitoring plugin
enabled           :  on
```

```
feature          :  Monitoring
init-func        :  mf_init
lib-path         :  /opt/SUNWdsee/ds6/lib/mf-plugin.so
type             :  object
vendor           :  Sun Microsystems, Inc.
version          :  6.0
```

Java ES MF 監視的疑難排解

如果 Java ES MF 監視無法運作，請確定已如「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」中的第 1 章「Installing Directory Server Enterprise Edition 6.3」所述，正確安裝共用代理程式容器。

若仍無法解決問題，請參閱「Sun Java Enterprise System 5 Monitoring Guide」。

使用 cn=monitor 監視伺服器

伺服器狀態、複寫狀態、資源使用率與其他監視資訊，皆可透過 DSCC 取得。

此外，您可以對下列項目執行搜尋作業，以從任何 LDAP 用戶端監視目錄伺服器目前的活動：

- `cn=monitor`
- `cn=monitor, cn=ldbm database, cn=plugins, cn=config`
- `cn=monitor,cn=`*dbName* `,cn=ldbm database,cn=plugins,cn=config`

  dbName 是所要監視之尾碼的資料庫名稱。請注意，除了各連線的相關資訊以外，`cn=monitor` 項目預設可由任何人讀取，包含匿名連結的用戶端。

  下列範例說明如何檢視一般的伺服器統計資料：

  ```
  $ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
   -s base -b "cn=monitor" "(objectclass=*)"
  ```

  如需這些項目中可用之所有監視屬性的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Directory ServerMonitoring Attributes」。

  許多可監視的參數皆會反映目錄伺服器的效能，且會受配置與調校的影響。如需有關每個配置屬性的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Man Page Reference」中該屬性的線上手冊。

第 2 部分

目錄代理伺服器管理

◆ ◆ ◆ 第 17 章

目錄代理伺服器工具

Sun Java™ System Directory Proxy Server 提供註冊與管理目錄代理伺服器實例的瀏覽器介面與指令行工具。此瀏覽器介面稱爲目錄服務控制中心 (DSCC)。本章說明透過使用 DSCC 或指令行管理目錄代理伺服器所需的基本作業。

若要決定是使用 DSCC 或是指令行來執行特定作業，請參閱第 45 頁的「決定 DSCC 與指令行的使用時機」。

如需有關管理架構的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中的「Directory Server Enterprise Edition Administration Model」。

本章包含下列主題：

- 第 305 頁的「使用目錄代理伺服器的 DSCC」
- 第 306 頁的「目錄代理伺服器的指令行工具」

使用目錄代理伺服器的 DSCC

本節說明如何存取目錄代理伺服器的 DSCC。

▼ 存取目錄代理伺服器的 DSCC

1. **請以與存取目錄伺服器相同的方式存取 DSCC。**
   請參閱第 47 頁的「存取 DSCC」。

2. **在 [代理伺服器] 標籤上按一下可檢視與管理目錄代理伺服器。**
   下圖顯示目錄代理伺服器的初始視窗。

圖 17–1 目錄代理伺服器的初始 DSCC 視窗

3 **按一下目錄代理伺服器實例可檢視或管理該伺服器。**

備註 – 如需有關使用 DSCC 的更多資訊，請參閱線上說明。

目錄代理伺服器的指令行工具

用於搭配目錄代理伺服器使用的指令行工具稱爲 `dpadm` 與 `dpconf`。如需有關如何使用這些指令的相關資訊，請參閱 `dpadm`(1M) 線上手冊與 `dpconf`(1M) 線上手冊。

`dpconf`、`dsconf`、`dsmig`、`dsccmon`、`dsccreg` 與 `dsccsetup` 是 LDAP 指令，因此您必須指定使用者連結 DN 與密碼，這些指令才可用以認證。而 `dpadm` 與 `dsadm` 指令則在實例檔案上運作。

本節說明 `dpadm` 與 `dpconf` 指令的位置。亦提供有關環境變數、指令間比較及於何處獲取使用這些指令之說明的資訊。

目錄代理伺服器指令的位置

依預設，目錄代理伺服器指令行工具位於下列目錄中：

install-path`/dps6/bin`

安裝路徑會隨作業系統而異。第 26 頁的 「預設路徑與指令位置」中列出了所有作業系統的安裝路徑。

設定 dpconf 的環境變數

dpconf 指令必須具有可使用環境變數預先設定的一些選項。如果使用指令時未指定選項，或未設定環境變數，則會使用預設設定。您可以配置下列選項的環境變數：

-D *userDN*	使用者連結 DN。環境變數：LDAP_ADMIN_USER。預設值：cn=Proxy Manager。
-w *password-file*	使用者連結 DN 的密碼檔案。環境變數： LDAP_ADMIN_PWF。預設值：密碼提示。
-h *host*	主機名稱或 IP 位址。環境變數：DIR_PROXY_HOST。預設值：localhost。
-p *LDAP-port*	LDAP 連接埠號碼。環境變數：DIR_PROXY_PORT。預設值：如果伺服器實例以**超級使用者**身份執行，則爲 389；如果伺服器實例以一般使用者身份執行，則爲 1389。
-e, --unsecured	指定 dpconf 預設應開啓一個無障礙連線。環境變數：DIR_PROXY_UNSECURED。若未設定此變數，dpconf 預設會開啓安全連線。

如需更多詳細資訊，請參閱 dpconf(1M) 線上手冊。

dpadm 與 dpconf 的比較

下表顯示 dpadm 與 dpconf 指令的比較。

表 17–1 dpadm 與 dpconf 指令的比較

	dpadm 指令	dpconf 指令
用途	管理目錄代理伺服器本機實例上的程序或檔案	配置目錄代理伺服器的本機或遠端實例
使用者	作業系統使用者	LDAP 使用者
本機或遠端	指令**必須**位於本機實例上，亦即必須在伺服器執行的主機上執行指令。	指令**可以**位於本機實例上，也可以從網路上的任何位置執行。
使用指令的範例	建立目錄代理伺服器實例。 啓動與停止目錄代理伺服器實例。 管理憑證資料庫。	修改目錄代理伺服器實例的配置。 建立資料檢視。 配置資料來源池中的負載平衡。

表 17–1 dpadm 與 dpconf 指令的比較 *(續)*

	dpadm 指令	dpconf 指令
伺服器狀態	伺服器可以在執行中，也可以已停止。	伺服器**必須**正在執行。
指令識別伺服器實例的方式	經由指定實例路徑。實例路徑可以是相對路徑或絕對路徑。	經由指定主機名稱或 IP 位址與連接埠號碼。 指令使用 LDAP 連接埠 (-p) 或 LDAPS 安全連接埠 (-P)。如果指令行中未指定連接埠號碼，就會使用環境變數 PROXY_PORT。若未設定環境變數，則使用預設連接埠。

使用 dpconf 設定多重值特性

某些目錄代理伺服器特性可具有多重值。使用以下語法指定下列值：

```
$ dpconf set-container-prop -h host -p port \
 property:value [property:value]
```

例如，若要爲 LDAP 資料檢視 my-view 設定多個可寫入的屬性，請鍵入下列指令：

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 view-name\
 writable-  attr:uid writable-attr:cn writable-attr:userPassword
```

若要將值增加至已包含值的多重值特性，請鍵入下列指令：

```
$ dpconf set-container-prop -h host -p port \
 property+:value
```

若要從已包含值的多重值特性移除值，請鍵入下列指令：

```
$ dpconf set-container-prop -h host -p port\
 property-:value
```

例如，在上述方案中，若要將 sn 增加至可寫入屬性的清單，請鍵入下列指令：

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 view-name\
writable-attr+:sn
```

若要從可寫入屬性的清單中移除 cn，請鍵入下列指令：

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 view-name\
writable-attr-:cn
```

取得有關使用 dpadm 與 dpconf 的說明

如需有關如何使用 dpadm 與 dpconf 指令的資訊，請參閱 dpadm(1M) 線上手冊與 dpconf(1M) 線上手冊。

- 若要取得子指令的清單，請鍵入適當的指令：

  ```
  $ dpadm --help

  $ dpconf --help
  ```

- 若要取得如何使用子指令的相關資訊，請鍵入適當的指令：

  ```
  $ dpadm subcommand --help

  $ dpconf subcommand --help
  ```

- 若要取得有關 dpconf 指令中所使用之配置特性的資訊，請鍵入：

  ```
  $ dpconf help-properties
  ```

- 若要取得有關子指令配置特性的資訊，請使用此指令：

  ```
  $ dpconf help-properties subcommand-entity
  ```

  例如，若要尋找有關存取記錄特性的資訊，請鍵入：

  ```
  $ dpconf help-properties access-log
  ```

- 若要取得有關子指令中所使用之特性的資訊，請使用此指令：

  ```
  $ dpconf help-properties subcommand-entity property
  ```

  例如，若要尋找有關 set-access-log-prop 子指令之 log-search-filters 特性的資訊，請鍵入：

  ```
  $ dpconf help-properties access-log log-search-filters
  ```

- 若要列出一組實體的主要特性 (例如資料檢視或連線處理程式)，請搭配 list 子指令使用詳細選項 -v。

  例如，若要檢視所有連線處理程式的主要特性與相關特性，請使用此指令：

  ```
  $ dpconf list-connection-handlers -h host -p port -v
  Name                        is-enabled  priority  description
  --------------------------  ----------  --------  --------------------------
  anonymous                   false       99        unauthenticated connections
  default connection handler  true        100       default connection handler
  dscc administrators        true          1         Administrators connection
                                                     handler
  ```

如需有關個別特性的更多資訊，請參閱該特性的線上手冊。

◆ ◆ ◆ 第 18 章

18 目錄代理伺服器實例

本章說明如何管理目錄代理伺服器實例。本章包含下列主題：

- 第 311 頁的「使用目錄代理伺服器實例」
- 第 314 頁的「配置目錄代理伺服器實例」
- 第 319 頁的「備份與復原目錄代理伺服器實例」

使用目錄代理伺服器實例

建立目錄代理伺服器實例時，會在指定的路徑中建立實例必要的檔案與目錄。

▼ 建立目錄代理伺服器實例

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

如果使用 DSCC 建立新的伺服器實例，可選擇複製現有伺服器的部分或所有伺服器配置設定。

1 **建立目錄代理伺服器實例。**

```
$ dpadm create -p port instance-path
```

例如，若要在目錄 `/local/dps` 中建立新的實例，請使用此指令：

```
$ dpadm create -p 2389 /local/dps
```

若要指定該實例的任何其他參數，請參閱 `dpadm`(1M) 線上手冊。

2 **若需要則請鍵入密碼。**

3 **透過驗證實例狀態確認是否已建立實例。**

```
$ dpadm info instance-path
```

4 **(可選擇) 如果使用 Sun Java™ Enterprise System 安裝程式或原生套裝軟體安裝了目錄代理伺服器，且作業系統提供服務管理解決方案，則您可將伺服器視為服務加以管理，如下表所示。**

作業系統	指令
Solaris 10	`dpadm enable-service --type SMF` *instance-path*
Solaris 9	`dpadm autostart` *instance-path*
Linux、HP-UX	`dpadm autostart` *instance-path*
Windows	`dpadm enable-service --type WIN_SERVICE` *instance-path*

5 **(可選擇) 透過使用下列方法之一註冊伺服器實例：**

- 經由 URL `https://localhost:6789` 存取 DSCC，並登入瀏覽器介面。
- 使用指令 `dsccreg add-server`。

  如需詳細資訊，請參閱 `dsccreg`(1M) 線上手冊。

▼ 尋找目錄代理伺服器實例的狀態

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **尋找目錄代理伺服器實例的狀態。**

```
$ dpadm info instance-path
```

▼ 啓動與停止目錄代理伺服器

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **若要啓動或停止目錄代理伺服器，請執行下列作業其中之一。**
  - 若要啓動目錄代理伺服器，請鍵入：

    ```
    $ dpadm start instance-path
    ```

    例如，若要啓動 `/local/dps` 中的實例，請使用此指令：

    ```
    $ dpadm start /local/dps
    ```
  - 若要停止目錄代理伺服器，請鍵入：

    ```
    $ dpadm stop instance-path
    ```

例如：

```
$ dpadm stop /local/dps
```

▼ 檢視是否需要重新啓動目錄代理伺服器實例

有時，需要重新啓動伺服器才能使配置變更生效。使用此程序檢查是否需要在變更某項配置後重新啓動目錄代理伺服器實例。

- **檢視是否需要重新啓動伺服器。**

  ```
  $ dpconf get-server-prop -h host -p port is-restart-required
  ```

  - 如果指令傳回 `true`，即必須重新啓動目錄代理伺服器實例。
  - 如果指令傳回 `false`，則無需重新啓動目錄代理伺服器實例。

▼ 重新啓動目錄代理伺服器

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **重新啓動目錄代理伺服器。**

  ```
  $ dpadm restart instance-path
  ```

  例如，若要重新啓動 `/local/dps` 中的實例，請使用此指令：

  ```
  $ dpadm restart /local/dps
  ```

▼ 刪除目錄代理伺服器實例

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **(可選擇) 停止目錄代理伺服器實例。**

   ```
   $ dpadm stop instance-path
   ```

   如不停止實例，刪除指令會自動停止實例。但是，如果已在服務管理解決方案中啓用實例，則必須手動停止實例。

2. **(可選擇) 如果之前使用 DSCC 管理伺服器，請使用指令行取消註冊伺服器。**

   ```
   $ dsccreg remove-server /local/dps
   Enter DSCC administrator's password:
   /local/dps is an instance of DPS
   ```

```
Enter password of "cn=Proxy Manager" for /local/dps:
Unregistering /local/dps from DSCC on localhost.
Connecting to /local/dps
Disabling DSCC access to /local/dps
```

如需詳細資訊，請參閱 dsccreg(1M) 線上手冊。

3 **(可選擇) 如果之前是在服務管理解決方案中啓用伺服器實例，則需停止將伺服器視為服務進行管理。**

作業系統	指令
Solaris 10	`dpadm disable-service --type SMF` *instance-path*
Solaris 9	`dpadm autostart --off` *instance-path*
Linux、HP-UX	`dpadm autostart --off` *instance-path*
Windows	`dpadm disable-service --type WIN_SERVICE` *instance-path*

4 **刪除實例。**

```
$ dpadm delete instance-path
```

配置目錄代理伺服器實例

本節說明如何配置目錄代理伺服器實例。本節中的程序使用 dpadm 與 dpconf 指令。如需有關這些指令的資訊，請參閱 dpadm(1M) 與 dpconf(1M) 線上手冊。

▼ 顯示目錄代理伺服器實例的配置

- **輸入** `dpconf info`

```
$ dpconf info
Instance Path           :  instance path
Host Name               :  host
Secure listen address   :  IP address
Port                    :  port
Secure port             :  secure port
SSL server certificate  :  defaultServerCert

Directory Proxy Server needs to be restarted.
```

只有 Secure listen address 與 Non-secure listen address 設定爲非預設值時，dpconf info 才會顯示這些特性。上述輸出不會顯示 Non-secure listen address，因爲此特性並未設定爲非預設值。

如果需要重新啓動實例，dpconf info 也會提醒使用者重新啓動。

您也可以使用 `dpadm info` 來顯示目錄代理伺服器實例配置資訊。

▼ 修改目錄代理伺服器的配置

本節說明如何修改目錄代理伺服器的配置。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 尋找目錄代理伺服器目前的配置。

```
$ dpconf get-server-prop -h host -p port
```

```
allow-cert-based-auth                : allow
allow-ldapv2-clients                             : true
allow-persistent-searches                        : false
allow-sasl-external-authentication               : true
allow-unauthenticated-operations                 : true
allowed-ldap-controls                            : -
cert-data-view-routing-custom-list               : none
cert-data-view-routing-policy                    : all-routable
cert-search-attr-mappings                        : none
cert-search-base-dn                              : none
cert-search-bind-dn                              : none
cert-search-bind-pwd                             : none
cert-search-user-attr                            : userCertificate
configuration-manager-bind-dn                    : cn=proxy manager
configuration-manager-bind-pwd                   : {3DES}RPdIFbvoWdvhLR8lU43zCMZyKFGPxfFg
connection-pool-wait-timeout                     : 3s
data-source-read-timeout                         : 20s
data-view-automatic-routing-mode                 : automatic
email-alerts-enabled                             : false
email-alerts-message-from-address                : local
email-alerts-message-subject                     : Proxy Server Administrative Alert
email-alerts-message-subject-includes-alert-code : false
email-alerts-message-to-address                  : root@localhost
email-alerts-smtp-host                           : localhost
email-alerts-smtp-port                           : smtp
enable-remote-user-mapping                       : false
enable-user-mapping                              : false
enabled-admin-alerts                             : none
enabled-ssl-cipher-suites                        : JRE
enabled-ssl-protocols                            : SSLv3
enabled-ssl-protocols                            : TLSv1
encrypt-configuration                            : true
extension-jar-file-url                           : none
is-restart-required                              : false
```

```
number-of-search-threads                    : 20
number-of-worker-threads                    : 50
proxied-auth-check-timeout                  : 30m
remote-user-mapping-bind-dn-attr            : none
scriptable-alerts-command                   : echo
scriptable-alerts-enabled                   : false
search-mode                                 : parallel
search-wait-timeout                         : 10s
ssl-client-cert-alias                       : none
ssl-server-cert-alias                       : defaultServerCert
supported-ssl-cipher-suites                 : SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
supported-ssl-cipher-suites                 : SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
supported-ssl-cipher-suites                 : SSL_DHE_DSS_WITH_DES_CBC_SHA
supported-ssl-cipher-suites                 : SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
supported-ssl-cipher-suites                 : SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
supported-ssl-cipher-suites                 : SSL_DHE_RSA_WITH_DES_CBC_SHA
supported-ssl-cipher-suites                 : SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
supported-ssl-cipher-suites                 : SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
supported-ssl-cipher-suites                 : SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
supported-ssl-cipher-suites                 : SSL_DH_anon_WITH_DES_CBC_SHA
supported-ssl-cipher-suites                 : SSL_DH_anon_WITH_RC4_128_MD5
supported-ssl-cipher-suites                 : SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
supported-ssl-cipher-suites                 : SSL_RSA_EXPORT_WITH_RC4_40_MD5
supported-ssl-cipher-suites                 : SSL_RSA_WITH_3DES_EDE_CBC_SHA
supported-ssl-cipher-suites                 : SSL_RSA_WITH_DES_CBC_SHA
supported-ssl-cipher-suites                 : SSL_RSA_WITH_NULL_MD5
supported-ssl-cipher-suites                 : SSL_RSA_WITH_NULL_SHA
supported-ssl-cipher-suites                 : SSL_RSA_WITH_RC4_128_MD5
supported-ssl-cipher-suites                 : SSL_RSA_WITH_RC4_128_SHA
supported-ssl-cipher-suites                 : TLS_DHE_DSS_WITH_AES_128_CBC_SHA
supported-ssl-cipher-suites                 : TLS_DHE_RSA_WITH_AES_128_CBC_SHA
supported-ssl-cipher-suites                 : TLS_DH_anon_WITH_AES_128_CBC_SHA
supported-ssl-cipher-suites                 : TLS_KRB5_EXPORT_WITH_DES_CBC_40_MD5
supported-ssl-cipher-suites                 : TLS_KRB5_EXPORT_WITH_DES_CBC_40_SHA
supported-ssl-cipher-suites                 : TLS_KRB5_EXPORT_WITH_RC4_40_MD5
supported-ssl-cipher-suites                 : TLS_KRB5_EXPORT_WITH_RC4_40_SHA
supported-ssl-cipher-suites                 : TLS_KRB5_WITH_3DES_EDE_CBC_MD5
supported-ssl-cipher-suites                 : TLS_KRB5_WITH_3DES_EDE_CBC_SHA
supported-ssl-cipher-suites                 : TLS_KRB5_WITH_DES_CBC_MD5
supported-ssl-cipher-suites                 : TLS_KRB5_WITH_DES_CBC_SHA
supported-ssl-cipher-suites                 : TLS_KRB5_WITH_RC4_128_MD5
supported-ssl-cipher-suites                 : TLS_KRB5_WITH_RC4_128_SHA
supported-ssl-cipher-suites                 : TLS_RSA_WITH_AES_128_CBC_SHA
supported-ssl-protocols                     : SSLv2Hello
supported-ssl-protocols                     : SSLv3
supported-ssl-protocols                     : TLSv1
syslog-alerts-enabled                       : false
syslog-alerts-facility                      : USER
```

```
syslog-alerts-host                              : localhost
use-cert-subject-as-bind-dn                     : true
use-external-schema                             : false
user-mapping-anonymous-bind-dn                  : none
user-mapping-anonymous-bind-pwd                 : none
user-mapping-default-bind-dn                    : none
user-mapping-default-bind-pwd                   : none
verify-certs                                    : false
```

或者，檢視一或多個配置特性目前的設定。

```
$ dpconf get-server-prop -h host -p port property-name ...
```

例如，透過執行此指令找出是否允許未認證的作業：

```
$ dpconf get-server-prop -h host -p port allow-unauthenticated-operations
allow-unauthenticated-operations  :  true
```

2 **變更一或多個配置參數**。

```
$ dpconf set-server-prop -h host -p port property:value ...
```

例如，透過執行此指令禁止未認證的作業：

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:false
```

如果嘗試執行非法的變更，便無法完成變更。例如，如果將 `allow-unauthenticated-operations` 參數設爲 `f` 而非 `false`，會產生下列錯誤：

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:f
The value "f" is not a valid value for the property "allow-unauthenticated-operations".
Allowed property values: BOOLEAN
The "set-server-prop" operation failed.
```

3 **請視需要重新啓動目錄代理伺服器實例以使變更生效**。

如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的「重新啓動目錄代理伺服器」。

配置代理伺服器管理員

代理伺服器管理員是具有特權的管理員，相當於 UNIX® 系統上的超級使用者。在建立目錄代理伺服器實例時定義代理伺服器管理員項目。代理伺服器管理員的預設 DN 爲 `cn=Proxy Manager`。

您可以檢視並變更代理伺服器管理員 DN 與密碼，如下列程序所示。

▼ 配置代理伺服器管理員

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1 尋找代理伺服器管理員的配置。

```
$ dpconf get-server-prop -h host -p port configuration-manager-bind-dn configuration-manager-bind-pwd
configuration-manager-bind-dn   :  cn=proxy manager
configuration-manager-bind-pwd  :  {3DES}U77v39WX8MDpcWVrueetB0lfJlBc6/5n
```

代理伺服器管理員的預設值爲 `cn=proxy manager`。傳回配置管理員密碼的雜湊值。

2 變更代理伺服器管理員的 DN。

```
$ dpconf set-server-prop -h host -p port configuration-manager-bind-dn:bindDN
```

3 建立包含代理伺服器管理員密碼的檔案，並設定指向該檔案的特性。

```
$ dpconf set-server-prop -h host -p port configuration-manager-bind-pwd-file:filename
```

要求重新啓動伺服器的配置變更

目錄代理伺服器及其實體的大部分配置變更可在線上完成。某些變更需要重新啓動伺服器才會生效。如果變更下列清單中任何特性的配置，則必須重新啓動伺服器：

```
aci-data-view
bind-dn
client-cred-mode
custom-distribution-algorithm
db-name
db-pwd
db-url
db-user
distribution-algorithm
ldap-address
ldap-port
ldaps-port
listen-address
listen-port
load-balancing-algorithm
num-bind-init
num-read-init
num-write-init
number-of-search-threads
number-of-threads
number-of-worker-threads
ssl-policy
use-external-schema
```

特性的 rws 與 rwd 關鍵字表示對該特性所做的變更是否需要重新啓動伺服器。

- 如果特性具有 rws (讀取、寫入、靜態) 關鍵字，則變更特性後必須重新啓動伺服器。
- 如果特性具有 rwd (讀取、寫入、動態) 關鍵字，則動態執行對特性的修改作業 (不重新啓動伺服器)。

若要確定對特性的變更是否需要重新啓動伺服器，請執行下列指令：

```
$ dpconf help-properties | grep property-name
```

例如，若要確定變更 LDAP 資料來源的連結 DN 是否需要重新啓動伺服器，請執行下列指令：

```
$ dpconf help-properties | grep bind-dn
connection-handler       bind-dn-filters        rwd  STRING | any
This property specifies a set of regular expressions. The bind DN
of a client must match at least one regular expression in order for
the connection to be accepted by the connection handler. (Default: any)
ldap-data-source      bind-dn               rws  DN | ""
This property specifies the DN to use when binding to the LDAP data
source. (Default: undefined)
```

若要確定在某項配置變更之後是否必須重新啓動伺服器，請執行下列指令：

```
$ dpconf get-server-prop -h host -p port is-restart-required
```

備份與復原目錄代理伺服器實例

當您使用 dpadm 備份目錄代理伺服器時，會備份配置檔案與伺服器憑證。如果已執行目錄代理伺服器虛擬 ACI，也會備份 ACI。

目錄代理伺服器在每次伺服器成功啓動時，自動備份 conf.ldif 檔案。

▼ 備份目錄代理伺服器實例

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **停止目錄代理伺服器實例。**

   ```
   $ dpadm stop instance-path
   ```

2. **備份目錄代理伺服器實例。**

   ```
   $ dpadm backup instance-path archive-dir
   ```

archive-dir 目錄由 backup 指令建立，並且不可以在執行指令前存在。此目錄包含每個配置檔案與憑證的備份。

▼ 復原目錄代理伺服器實例

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

您必須建立目錄代理伺服器實例，才能開始復原作業。

1. **停止目錄代理伺服器實例。**

   ```
   $ dpadm stop instance-path
   ```

2. **復原目錄代理伺服器實例。**

   ```
   $ dpadm restore instance-path archive-dir
   ```

   - 如果存在實例路徑，則以無訊息方式執行復原作業。*archive-dir* 目錄中的配置檔案與憑證會取代 *instance-path* 目錄中的配置檔案與憑證。
   - 如果不存在實例路徑，復原作業會失敗。

◆ ◆ ◆ 第 19 章

19 LDAP 資料檢視

LDAP 資料檢視根據用戶端的請求來顯示 LDAP 伺服器中的資料，並指定回應請求的資料來源池。您可以透過定義 LDAP 資料檢視，執行下列作業：

- 以單一檢視顯示整個資料庫
- 爲資料庫中不同子樹狀結構提供不同檢視
- 提供不同資料庫的統一檢視

建立 LDAP 資料檢視

建立 LDAP 資料檢視包含下列步驟：

1. 第 321 頁的「建立 LDAP 資料來源」。
2. 第 324 頁的「建立 LDAP 資料來源池」。
3. 第 325 頁的「將 LDAP 資料來源附加至資料來源池」。
4. 第 327 頁的「建立 LDAP 資料檢視」。

建立與配置 LDAP 資料來源

本節描述如何使用 `dpconf` 指令建立與配置 LDAP 資料來源。如需有關這些主題的參考資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「LDAP Data Sources」。

如需有關如何建立與配置 LDAP 資料來源的資訊，請參閱下列程序。

▼ 建立 LDAP 資料來源

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1 建立資料來源。

```
$ dpconf create-ldap-data-source -h host -p port source-name host:port
```

在此指令中，*source-name* 是指定給新資料來源的名稱。*host* 與 *port* 是指 LDAP 伺服器在其上執行的主機與連接埠。請注意，資料來源依預設不使用 SSL。

如果由 IP V6 位址指定主機，建立資料來源時必須使用 IP V6 參照。例如，如果目錄代理伺服器連結至連接埠 2389 上 IP V6 位址爲 fe80::209:3dff:fe00:8c93 的主機，請使用下列指令建立資料來源：

```
$ dpconf create-ldap-data-source -h host1 -p 1389 ipv6-host \
 [fe80::209:3dff:fe00:8c93]:2389
```

如果使用主控台建立資料來源，則必須指定實際 IP V6 位址 (不包含方括號)。

如需有關如何修改 LDAP 資料來源特性的資訊，請參閱第 322 頁的「配置 LDAP 資料來源」。

2 (選用) 檢視資料來源清單。

```
$ dpconf list-ldap-data-sources -h host -p port
```

▼ 配置 LDAP 資料來源

下列程序說明如何顯示 LDAP 資料來源的特性，以及如何設定需要變更的特性。程序中說明可變更使用任何一個 LDAP 資料來源之特性的指令。同時亦說明如何取得特性的詳細資訊，以協助您設定該特性。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1 透過使用此指令語法檢視資料來源的特性：

```
$ dpconf get-ldap-data-source-prop -h host -p port \
[-M unit] [-Z unit] source-name [property...]
```

在此指令中，-M 與 -Z 是指用以顯示資料的單位。M 選項指定時間單位。-M 值可以是 M、w、d、h、m、s 或 ms，分別表示月、週、天、小時、分、秒或毫秒。-Z 選項指定資料大小單位。-Z 值可以是 T、G、M、k 或 b，分別表示兆位元組 (TB)、十億位元組 (GB)、百萬位元組 (MB)、千位元組 (KB) 或位元組。

如不指定某個特性，則顯示所有特性。LDAP 資料來源的預設特性如下：

```
bind-dn                       :  -
bind-pwd                      :  -
client-cred-mode              :  use-client-identity
connect-timeout               :  10s
description                   :  -
is-enabled                    :  false
is-read-only                  :  true
ldap-address                  :  host
ldap-port                     :  port
ldaps-port                    :  ldaps
```

```
monitoring-bind-timeout        :  5s
monitoring-entry-dn            :  ""
monitoring-entry-timeout       :  5s
monitoring-inactivity-timeout  :  2m
monitoring-interval            :  30s
monitoring-mode                :  proactive
monitoring-search-filter       :  (|(objectClass=*)(objectClass=ldapSubEntry))
num-bind-incr                  :  10
num-bind-init                  :  10
num-bind-limit                 :  1024
num-read-incr                  :  10
num-read-init                  :  10
num-read-limit                 :  1024
num-write-incr                 :  10
num-write-init                 :  10
num-write-limit                :  1024
proxied-auth-check-timeout     :  1.8s
proxied-auth-use-v1            :  false
ssl-policy                     :  never
use-tcp-no-delay               :  true
```

2 啓用資料來源。

```
$ dpconf set-ldap-data-source-prop -h host -p port source-name is-enabled:true
```

3 若要變更預設設定，請配置步驟 1 中列出的所有特性。

```
$ dpconf set-ldap-data-source-prop -h host -p port source-name property:value
```

例如，若要修改資料來源的項目，請將資料來源配置爲允許寫入作業。

```
$ dpconf set-ldap-data-source-prop -h host -p port source-name is-read-only:false
```

若要尋找子指令中所用特性的相關資訊，請執行此指令：

```
$ dpconf help-properties ldap-data-source property
```

例如，若要尋找有關 `is-read-only` 特性的資訊，請執行此指令：

```
dpconf help-properties ldap-data-source is-read-only
```

若要列出資料來源的主要特性，請搭配 `list-ldap-data-source` 子指令使用詳細選項 `-v`。

```
$ dpconf list-ldap-data-sources -v
Name         is-enabled  ldap-address  ldap-port  ldaps-port  description
-----------  ----------  ------------  ---------  ----------  -----------
datasource0  true        myHost        myPort     ldaps       -
datasource1  true        myHost        myPort     ldaps       -
```

4 **請視需要重新啓動目錄代理伺服器實例以使變更生效。**

如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的「重新啓動目錄代理伺服器」。如需要求重新啓動伺服器之配置變更的清單，請參閱第 318 頁的「要求重新啓動伺服器的配置變更」。

建立與配置 LDAP 資料來源池

本節描述如何使用 `dpconf` 指令建立與配置 LDAP 資料來源池。如需有關這些主題的參考資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「LDAP Data Sources」。

如需有關如何建立與配置資料來源池的資訊，請參閱下列程序：

▼ 建立 LDAP 資料來源池

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1 **建立一或多個資料來源池。**

```
$ dpconf create-ldap-data-source-pool -h host -p port pool-name
```

您可以在第一個 *pool-name* 之後指定其他資料來源池。如需有關如何修改資料來源池特性的資訊，請參閱第 324 頁的「配置 LDAP 資料來源池」。

2 **(選用) 檢視資料來源池的清單。**

```
$ dpconf list-ldap-data-source-pools -h host -p port
```

▼ 配置 LDAP 資料來源池

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1 **使用此指令語法檢視資料來源池的特性：**

```
$ dpconf get-ldap-data-source-pool-prop -h host -p port \
[-M unit] [-Z unit] pool-name [property...]
```

在此指令中，`-M` 與 `-Z` 是指用以顯示資料的單位。`M` 選項指定時間單位。`-M` 值可以是 `M`、`w`、`d`、`h`、`m`、`s` 或 `ms`，分別表示月、週、天、小時、分、秒或毫秒。`-Z` 選項指定資料大小單位。`-Z` 值可以是 `T`、`G`、`M`、`k` 或 `b`，分別表示兆位元組 (TB)、十億位元組 (GB)、百萬位元組 (MB)、千位元組 (KB) 或位元組。

如不指定某個特性，則顯示所有特性。LDAP 資料來源池的預設特性如下：

```
client-affinity-policy    :  write-affinity-after-write
client-affinity-timeout   :  20s
```

```
description                :  -
enable-client-affinity     :  false
load-balancing-algorithm   :  proportional
```

2 **配置步驟 1 中所列的特性。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 property:value
```

如需有關如何針對負載平衡與用戶端相似性配置資料來源池特性的資訊，請參閱第 21 章。

將 LDAP 資料來源附加至資料來源池

附加至資料來源池的資料來源稱爲**附加資料來源**。附加資料來源的特性決定資料來源池的負載平衡配置。當您配置附加資料來源的加權時，請考量資料來源池中所有附加資料來源的加權。請確保所有加權可依需求來搭配運作。如需有關如何配置負載平衡加權的資訊，請參閱第 352 頁的 「配置負載平衡加權」。

▼ 將 LDAP 資料來源附加至資料來源池

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **將一或多個資料來源附加至資料來源池。**

```
$ dpconf attach-ldap-data-source -h host -p port pool-name \
 source-name [source-name ...]
```

2 **(選用) 檢視指定資料來源池的附加資料來源清單。**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -E pool-name
```

在此指令中，`-E` 爲可選擇的，用以將顯示輸出修改爲每行顯示一個特性值。

3 **(選用) 檢視指定資料來源池的附加資料來源之主要特性。**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
```

在此指令中，`-v` 指定詳細輸出。例如，檢視範例資料來源池的特性。

```
$ dpconf list-attached-ldap-data-sources -h host1 -p 1389 -v My-pool
SRC_NAME     add-weight  bind-weight  compare-weight
-----------  ----------  -----------  --------------
datasource0  disabled    disabled     disabled
datasource1  disabled    disabled     disabled

delete-weight  modify-dn-weight  modify-weight  search-weight
-------------  ----------------  -------------  -------------
```

```
disabled       disabled          disabled       disabled
disabled       disabled          disabled       disabled
```

4 **(選用) 透過使用下列指令語法檢視附加資料來源的特性：**

```
$ dpconf get-attached-ldap-data-source-prop -h host -p port [-M unit] [-Z unit] \
 pool-name source-name [property...]
```

在此指令中，-M 與 -Z 是指用以顯示資料的單位。M 選項指定時間單位。-M 值可以是 M、w、d、h、m、s 或 ms，分別表示月、週、天、小時、分、秒或毫秒。-Z 選項指定資料大小單位。-Z 值可以是 T、G、M、k 或 b，分別表示兆位元組 (TB)、十億位元組 (GB)、百萬位元組 (MB)、千位元組 (KB) 或位元組。

如不指定某個特性，則顯示所有特性。

附加資料來源的特性定義負載平衡中各種作業類型的加權。附加資料來源的預設加權如下：

```
add-weight        :  disabled
bind-weight       :  disabled
compare-weight    :  disabled
delete-weight     :  disabled
modify-dn-weight  :  disabled
modify-weight     :  disabled
search-weight     :  disabled
```

您必須設定目錄代理伺服器附加資料來源的特性，才能如預期運作。在下列範例中，所有特性會設定爲 1。您可以根據需求變更這些特性的值。如需有關如何配置負載平衡的附加資料來源加權的資訊，請參閱第 352 頁的「配置負載平衡加權」。

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name add-weight:1
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name bind-weight:1
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name compare-weight:1
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name delete-weight:1
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name modify-dn-weight:1
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name modify-weight:1
$ dpconf set-attached-ldap-data-source-prop -h host -p port \
pool-name source-name search-weight:1
```

使用 LDAP 資料檢視

如需有關如何建立與配置 LDAP 資料檢視的資訊，請參閱下列程序：

▼ 建立 LDAP 資料檢視

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **建立 LDAP 資料檢視。**

```
$ dpconf create-ldap-data-view -h host -p port view-name pool-name suffix-DN
```

如需有關如何修改 LDAP 資料檢視特性的資訊，請參閱第 327 頁的 「配置 LDAP 資料檢視」。

2 **檢視 LDAP 資料檢視清單。**

```
$ dpconf list-ldap-data-views -h host -p port
```

▼ 配置 LDAP 資料檢視

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **檢視 LDAP 資料檢視的特性。**

```
$ dpconf get-ldap-data-view-prop -h host -p port view-name
```

如果建立資料檢視而未配置任何特性，資料檢視具有下列配置：

```
alternate-search-base-dn                    :  ""
attr-name-mappings                          :  none
base-dn                                     :  suffix-DN
contains-shared-entries                     :  false
custom-distribution-algorithm-class         :  none
description                                 :  -
distribution-algorithm                      :  none
dn-join-rule                                :  none
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  -
filter-join-rule                            :  none
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
ldap-data-source-pool                       :  pool-name
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  none
non-writable-attr                           :  none
numeric-attrs                               :  all
numeric-default-data-view                   :  false
```

```
numeric-lower-bound                          :  none
numeric-upper-bound                          :  none
pattern-matching-base-object-search-filter   :  all
pattern-matching-dn-regular-expression       :  all
pattern-matching-one-level-search-filter     :  all
pattern-matching-subtree-search-filter       :  all
process-bind                                 :  -
replication-role                             :  master
viewable-attr                                :  all except non-viewable-attr
writable-attr                                :  all except non-writable-attr
```

備註 – 代理伺服器管理員以外的所有使用者，皆可從後端伺服器看到 `cn=config` 與 `cn=monitor` 尾碼。依預設，後端伺服器的資料對代理伺服器管理員不可用。代理伺服器管理員可以使用的 `cn=config` 與 `cn=monitor` 子樹狀結構，是代理伺服器本身的子樹狀結構。

建立目錄代理伺服器實例時，系統使用空的資料檢視策略建立代理伺服器管理員的連線處理程式。如果代理伺服器管理員需要存取後端資料，則必須將資料檢視增加至代理伺服器管理員連線處理程式的資料檢視策略中。依預設，這類資料檢視不包含 `cn=config` 與 `cn=monitor` 子樹狀結構。

2 變更步驟 1 中所列的一或多個特性。

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 property:value [property:value ... ]
```

例如，若要存取資料來源中的 `dc=example,dc=com` 子樹狀結構，請在資料檢視中指定 `dn-mapping-source-base-dn`。

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView \
dn-mapping-source-base-dn:dc=example,dc=com
```

若要將值增加至多值特性，請使用此指令：

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name property+:value
```

若要從多值特性中移除值，請使用此指令：

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name property-:value
```

3 請視需要重新啓動目錄代理伺服器實例以使變更生效。

如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的 「重新啓動目錄代理伺服器」。

使用目錄代理伺服器存取目錄伺服器的配置項目

目錄代理伺服器的配置項目位於 cn=config 中。依預設，使用目錄代理伺服器存取配置項目時，會存取目錄代理伺服器的配置項目。

若要存取目錄伺服器的配置項目，最好直接連線至目錄伺服器，而非目錄代理伺服器。如需有關如何配置目錄伺服器的資訊，請參閱第 4 章。

注意 – 如果重新配置目錄代理伺服器存取目錄伺服器的配置項目，則很可能破壞目錄代理伺服器的管理架構。

如果您真的需要透過目錄代理伺服器存取目錄伺服器的配置項目，請採取特殊步驟以確保不會破壞目錄代理伺服器的管理架構。本節說明如何透過使用目錄代理伺服器存取目錄伺服器的配置項目。

▼ 透過使用目錄代理伺服器存取目錄伺服器的配置項目

1. **如第 321 頁的 「建立與配置 LDAP 資料來源」中所述，建立一或多個資料來源。**
2. **如第 324 頁的 「建立與配置 LDAP 資料來源池」中所述，建立 LDAP 資料來源池。**
3. **如第 325 頁的 「將 LDAP 資料來源附加至資料來源池」中所述，將一或多個資料來源附加至資料來源池。**

   - **若要顯示某個特定資料來源的配置項目，請僅附加一個 LDAP 資料來源至 LDAP 資料來源池。**

     ```
     $ dpconf attach-ldap-data-source -h host -p port pool-name data-source-name
     ```

     執行此步驟之後，用戶端可以存取連線至目錄代理伺服器之資料來源的配置項目。

   - **若要顯示任何資料來源的配置項目，請附加多個 LDAP 資料來源至 LDAP 資料來源池。**

     ```
     $ dpconf attach-ldap-data-source -h host -p port pool-name data-source-name \
       data-source-name ...
     ```

     執行此步驟之後，用戶端可以存取連線至目錄代理伺服器的某一資料來源的配置項目。但是，用戶端無法得知該配置項目所屬的資料來源。

4. **建立 LDAP 資料檢視以顯示 cn=config。**

   ```
   $ dpconf create-ldap-data-view -h host -p port view-name pool-name cn=config
   ```

重新命名屬性與 DN

目錄中的每個項目會由一個 DN 與一組屬性及其值所識別。通常，在用戶端定義的 DN 與屬性不會對映至在伺服器端定義的 DN 與屬性。您可以定義資料檢視以重新命名 DN 與屬性。當用戶端提出請求時，DN 和屬性會重新命名爲符合伺服器端的名稱。當結果傳回用戶端時，DN 與屬性會重新命名回符合用戶端的名稱。

如需有關屬性重新命名與 DN 重新命名的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Attribute Renaming and DN Renaming」。如需有關如何重新命名屬性與 DN 的資訊，請參閱下列程序：

▼ 配置屬性重新命名

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **在您要配置屬性對映的資料檢視上設定一或多個 `attr-name-mappings` 特性。**

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
  attr-name-mappings:client-side-attribute-name#server-side-attribute-name\
  [attr-name-mappings:client-side-attribute-name#server-side-attribute-name ...]
```

例如，將用戶端的 `surname` 重新命名爲伺服器端的 `sn`。

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView \
 attr-name-mappings:surname#sn
```

若要將屬性對映增加至現有的對映清單中，請使用此指令：

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 attr-name-mappings+:client-side-attribute-name#server-side-attribute-name
```

若要從現有的對映清單中移除屬性對映，請使用此指令：

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 attr-name-mappings-:client-side-attribute-name#server-side-attribute-name
```

▼ 配置 DN 重新命名

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **檢視您要重新命名 DN 的資料檢視之 `base-dn` 特性與 DN 對映特性。**

```
$ dpconf get-ldap-data-view-prop -h host -p port view-name base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
```

這些特性具有下列意義：

- `base-dn` 是用戶端子樹狀結構的 DN，相當於資料檢視的基底 DN。
- `dn-mapping-source-base-dn` 是伺服器端子樹狀結構的 DN。
- `dn-mapping-attrs` 可定義包含項目 DN 的屬性清單。

例如，未定義 DN 重新命名時，用戶端的 `dc=example,dc=com` 資料庫資料檢視具有下列值：

```
$ dpconf get-ldap-data-view-prop myDataView base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
 base-dn                   :  dc=example,dc=com
 dn-mapping-attrs          :  none
 dn-mapping-source-base-dn  :  none
```

2 將用戶端的 DN 對映至伺服器端的 DN。

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
  dn-mapping-source-base-dn:server-side-dn
```

例如，將用戶端的 `dc=example,dc=com` 資料庫對映至伺服器端的 `dc=example,dc=org`。

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView \
 dn-mapping-source-base-dn:dc=example,dc=org
```

3 DIT 部分中受步驟 2 影響的屬性若包含 DN，也必須將其重新命名。

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
  dn-mapping-attrs:attribute-name [dn-mapping-attrs:attribute-name ...]
```

例如，如果 `group` 屬性包含 DN 且所在的名稱空間受步驟 2 的重新命名作業影響，請依下列方式重新命名屬性：

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView dn-mapping-attrs:group
```

若要將 DN 對映增加至現有的對映清單中，請使用此指令：

```
$ dpconf set-ldap-data-view-prop -h host -p port \
view-name dn-mapping-attrs+:attribute-name
```

若要從現有的對映清單中移除 DN 對映，請使用此指令：

```
$ dpconf set-ldap-data-view-prop -h host -p port \
view-name dn-mapping-attrs-:attribute-name
```

4 檢視您已為其重新命名 DN 的資料檢視之 `base-dn` 特性與 DN 對映特性。

```
$ dpconf get-ldap-data-view-prop -h host -p port view-name base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
```

例如，DN 重新命名之後，用戶端的 dc=example,dc=com 資料庫資料檢視具有下列值：

```
$ dpconf get-ldap-data-view-prop -h host1 -p 1389 myDataView base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
 base-dn                    :  dc=example,dc=com
 dn-mapping-attrs           :  group
 dn-mapping-source-base-dn  :  dc=example,dc=org
```

配置檢視排除基底與替代搜尋基底

建立從屬資料檢視時，目錄代理伺服器自動將從屬資料檢視從上層資料檢視排除。當請求的目標爲從屬資料檢視時，該請求將傳送至從屬資料檢視，而不是上層資料檢視。

在從屬資料檢視中指定替代搜尋基底時，目標在上層資料檢視的搜尋作業會同時在從屬資料檢視中執行。

依預設，目錄代理伺服器自動配置 excluded-subtrees 與 alternate-search-base-dn 特性。下列程序說明如何手動配置這些特性。

▼ 手動配置 excluded-subtrees 與 alternate-search-base-dn 特性

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **將目錄代理伺服器配置為手動路由請求。**

```
$ dpconf set-server-prop -h host -p port data-view-automatic-routing-mode:manual
```

當 data-view-automatic-routing-mode 爲 manual 時，目錄代理伺服器不產生 excluded-subtrees 與 alternate-search-base-dn 特性。您必須手動設定這些特性的值。此處設定的值未經目錄代理伺服器檢查。請注意，錯誤地設定這些值會破壞管理路徑。

您可以將目錄代理伺服器配置爲手動路由部分請求。

```
$ dpconf set-server-prop -h host -p port data-view-automatic-routing-mode:limited
```

當 data-view-automatic-routing-mode 爲 limited 時，目錄代理伺服器不產生 excluded-subtrees 與 alternate-search-base-dn 特性。但是，目錄代理伺服器一定會檢查此處設定的值，確保其與管理路徑不衝突。

2 **配置檢視排除基底。**

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name excluded-subtrees:suffix-DN
```

檢視排除基底決定資料檢視不顯示其項目之 DIT 的分支。

3 **配置替代搜尋基底。**

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 alternate-search-base-dn:search-base-DN
```

替代搜尋基底決定 DIT 的其他分支，在其中可能可以找到屬於此資料檢視的項目。依預設，基底 DN 在所有資料檢視中定義爲替代搜尋基底。

建立與配置使用範例的資料檢視

本節包含下列有關資料檢視以及如何建立與配置這些檢視的資訊：

- 第 333 頁的 「預設資料檢視」
- 第 334 頁的 「不論請求的目標 DN 爲何，均路由所有請求的資料檢視」
- 第 335 頁的 「當子樹狀結構清單儲存在多個資料相同的資料來源中時，路由請求的資料檢視」
- 第 337 頁的 「當不同的子樹狀結構儲存在不同的資料來源中時，提供單一存取點的資料檢視」
- 第 338 頁的 「當上層與從屬子樹狀結構儲存在不同的資料來源中時，提供單一存取點的資料檢視」

本節中的範例假設連線處理程式允許目錄代理伺服器處理所有用戶端連線。

預設資料檢視

若建立了資料檢視但未配置任何特性，則資料檢視會具備下列配置：

```
alternate-search-base-dn                 :  ""
alternate-search-base-dn                 :  base-DN
attr-name-mappings                       :  none
base-dn                                  :  suffix-DN
contains-shared-entries                  :  -
description                              :  -
distribution-algorithm                   :  -
dn-join-rule                             :  -
dn-mapping-attrs                         :  none
dn-mapping-source-base-dn                :  none
excluded-subtrees                        :  -
filter-join-rule                         :  -
is-enabled                               :  true
is-read-only                             :  false
is-routable                              :  true
ldap-data-source-pool                    :  pool-name
```

```
lexicographic-attrs                          :  all
lexicographic-lower-bound                    :  none
lexicographic-upper-bound                    :  none
non-viewable-attr                            :  -
non-writable-attr                            :  -
numeric-attrs                                :  all
numeric-default-data-view                    :  false
numeric-lower-bound                          :  none
numeric-upper-bound                          :  none
pattern-matching-base-object-search-filter   :  all
pattern-matching-dn-regular-expression       :  all
pattern-matching-one-level-search-filter     :  all
pattern-matching-subtree-search-filter       :  all
process-bind                                 :  -
replication-role                             :  master
viewable-attr                                :  all except non-viewable-attr
writable-attr                                :  all except non-writable-attr
```

不論請求的目標 DN 為何，均路由所有請求的資料檢視

本節說明路由所有請求至資料來源池的資料檢視配置 (不論請求的目標 DN 爲何)。此資料檢視稱爲**根資料檢視**。依預設，建立目錄代理伺服器實例時建立根資料檢視。如需有關根資料檢視的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Data Views to Route All Requests, Irrespective of the Target DN of the Request」。

根資料檢視具有下列配置：

```
alternate-search-base-dn                  :  -
attr-name-mappings                        :  none
base-dn                                   :  ""
contains-shared-entries                   :  -
description                               :  Automatically-generated data view
                                             able to route client operations
                                             independently of the operation base dn
distribution-algorithm                    :  -
dn-join-rule                              :  -
dn-mapping-attrs                          :  none
dn-mapping-source-base-dn                 :  none
excluded-subtrees                         :  ""
excluded-subtrees                         :  cn=config
excluded-subtrees                         :  cn=monitor
excluded-subtrees                         :  cn=proxy manager
excluded-subtrees                         :  cn=virtual access controls
excluded-subtrees                         :  dc=example,dc=com
```

```
filter-join-rule                          :  -
is-enabled                                :  true
is-read-only                              :  false
is-routable                               :  true
ldap-data-source-pool                     :  defaultDataSourcePool
lexicographic-attrs                       :  all
lexicographic-lower-bound                 :  none
lexicographic-upper-bound                 :  none
non-viewable-attr                         :  -
non-writable-attr                         :  -
numeric-attrs                             :  all
numeric-default-data-view                 :  false
numeric-lower-bound                       :  none
numeric-upper-bound                       :  none
pattern-matching-base-object-search-filter :  all
pattern-matching-dn-regular-expression    :  all
pattern-matching-one-level-search-filter  :  all
pattern-matching-subtree-search-filter    :  all
process-bind                              :  -
replication-role                          :  master
viewable-attr                             :  all except non-viewable-attr
writable-attr                             :  all except non-writable-attr
```

當子樹狀結構清單儲存在多個資料相同的資料來源中時，路由請求的資料檢視

本節說明如何配置資料檢視，以路由目標爲子樹狀結構清單的請求至一組資料等效的資料來源中。如需有關此部署類型的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Data Views to Route Requests When a List of Subtrees Are Stored on Multiple, Data-Equivalent Data Sources」。

本節中的範例具有多個包含同一組子樹狀結構的資料來源。這類資料來源具有相同的資料，而且集中在一個資料來源池中以進行負載平衡。每個子樹狀結構皆配置有資料檢視，於用戶端請求時顯示該子樹狀結構。下圖顯示部署範例。

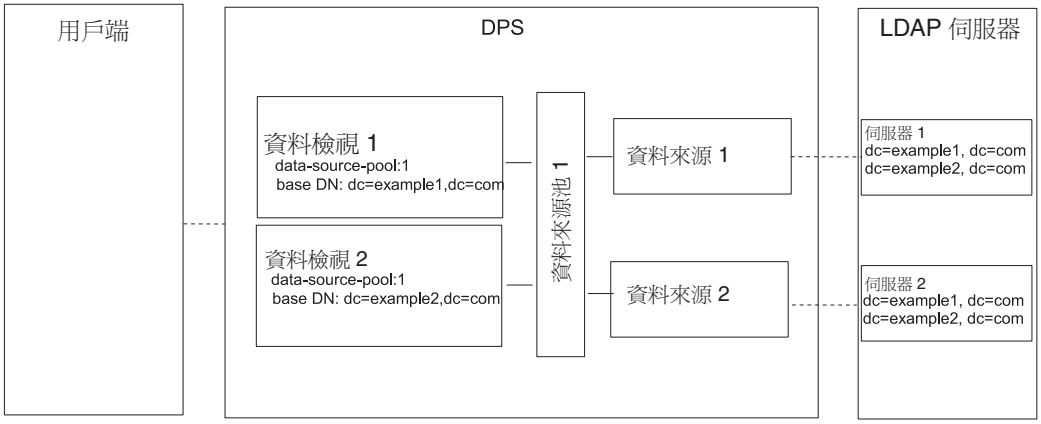


圖 19–1 當子樹狀結構清單儲存在多個資料相同的資料來源中時，路由請求的部署範例

▼ 配置當子樹狀結構清單儲存在多個資料相同的資料來源中時，路由請求的資料檢視

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **如第 321 頁的 「建立與配置 LDAP 資料來源」中所述，建立各個 LDAP 伺服器的資料來源。**

2. **如第 324 頁的 「建立與配置 LDAP 資料來源池」中所述，建立資料來源池。**

3. **如第 325 頁的 「將 LDAP 資料來源附加至資料來源池」中所述，將資料來源附加至資料來源池。**

4. **(可選擇) 配置負載平衡。**

   如需有關資訊，請參閱第 351 頁的 「配置負載平衡」。

5. **建立基底 DN 為 `dc=example1,dc=com`、參考資料來源池的資料檢視。**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-1 \
   data-source-pool-1 dc=example1,dc=com
   ```

6. **建立基底 DN 為 `dc=example2,dc=com`、參考資料來源池的另一個資料檢視。**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-2 \
   data-source-pool-1 dc=example2,dc=com
   ```

   資料檢視的其他特性和第 333 頁的 「預設資料檢視」中的預設資料檢視相同。

7 **請視需要重新啓動目錄代理伺服器實例以使變更生效。**

如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的 「重新啓動目錄代理伺服器」。

當不同的子樹狀結構儲存在不同的資料來源中時，提供單一存取點的資料檢視

本節說明如何配置資料檢視，以提供儲存在多個資料來源之不同子樹狀結構的單一存取點。如需有關此部署類型的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Data Views to Provide a Single Point of Access When Different Subtrees Are Stored on Different Data Sources」。

本節中的範例包含各個子樹狀結構的資料檢視。資料來源池會配置給每組資料相同的資料來源。下圖顯示部署範例。

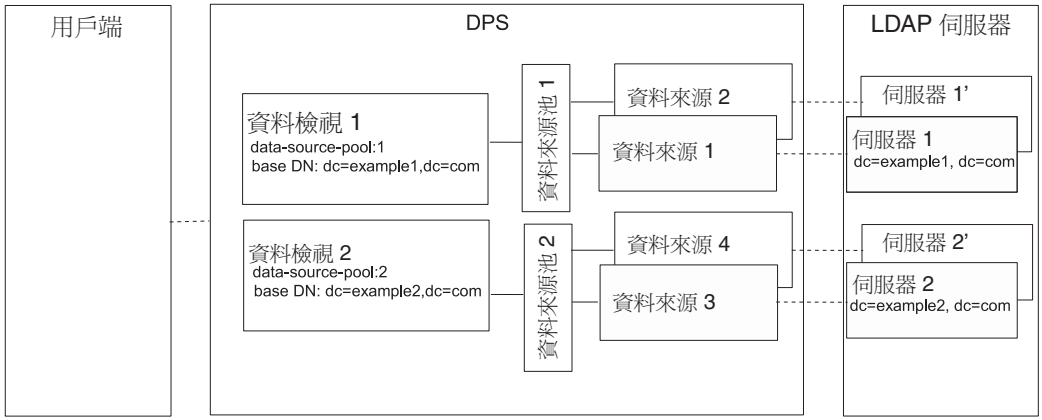


圖 19–2 當不同的子樹狀結構儲存在不同的資料來源時，提供單一存取點的部署範例

▼ 配置當不同的子樹狀結構儲存在不同的資料來源中時，提供單一存取點的資料檢視

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **如第 321 頁的 「建立與配置 LDAP 資料來源」中所述，建立各個 LDAP 伺服器的資料來源。**

2 **如第 324 頁的 「建立與配置 LDAP 資料來源池」中所述，建立兩個資料來源池。**

3. **如第 325 頁的「將 LDAP 資料來源附加至資料來源池」中所述，將包含 `dc=example1,dc=com` 的資料來源附加至 `data-source-pool-1`，並將包含 `dc=example2,dc=com` 的資料來源附加至 `data-source-pool-2`。**

4. **(可選擇) 配置負載平衡。**

   如需有關資訊，請參閱第 351 頁的「配置負載平衡」。

5. **建立基底 DN 為 `dc=example1,dc=com`、參考 `data-source-pool-1` 的資料檢視。**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-1 \
   data-source-pool-1 dc=example1,dc=com
   ```

6. **建立基底 DN 為 `dc=example2,dc=com`、參考 `data-source-pool-2` 的另一個資料檢視。**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-2 \
   data-source-pool-1 dc=example2,dc=com
   ```

   資料檢視的其他特性和第 333 頁的「預設資料檢視」中的預設資料檢視相同。

7. **請視需要重新啓動目錄代理伺服器實例以使變更生效。**

   如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的「重新啓動目錄代理伺服器」。

當上層與從屬子樹狀結構儲存在不同的資料來源中時，提供單一存取點的資料檢視

本節說明如何配置資料檢視，以在子樹狀結構的上層分支與從屬分支儲存在不同的資料來源中時，提供單一存取點。如需有關此部署類型的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Data Views to Route Requests When Superior and Subordinate Subtrees Are Stored in Different Data Sources」。

本節中的範例包含三個資料檢視。資料檢視 1 的基底 DN 是資料檢視 2 與 3 的上層基底 DN，亦即資料來源 2 與 3 包含從屬於資料來源 1 中子樹狀結構的子樹狀結構。下圖顯示部署範例。

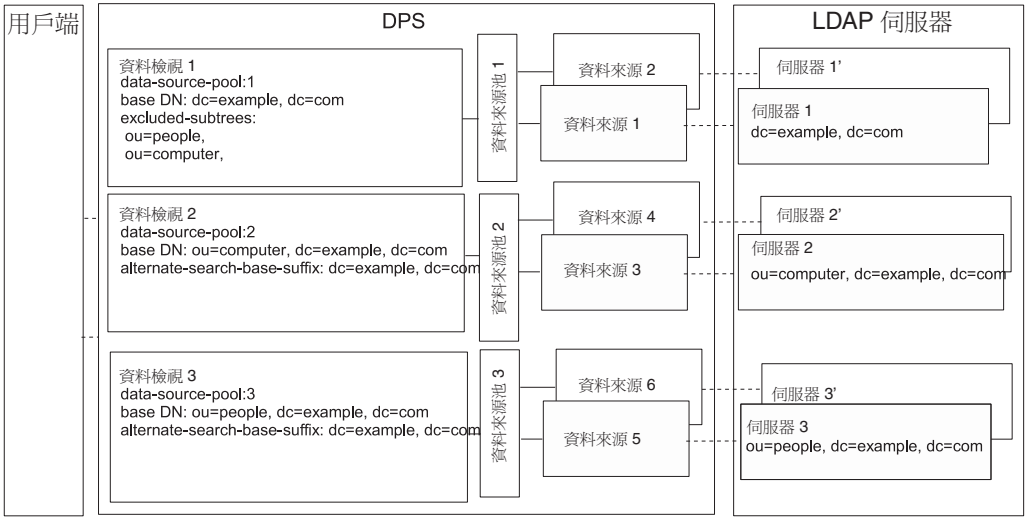


圖 19–3 當上層與從屬子樹狀結構儲存在不同的資料來源中時，路由請求的部署範例

目錄代理伺服器在子樹狀結構的從屬分支配置爲其他資料檢視的基底 DN 時，自動從資料檢視排除從屬分支。

▼ 配置當上層與從屬子樹狀結構儲存在不同的資料來源中時，提供單一存取點的資料檢視

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **如第 321 頁的 「建立與配置 LDAP 資料來源」中所述，建立各個 LDAP 伺服器的資料來源。**

2. **如第 324 頁的 「建立與配置 LDAP 資料來源池」中所述，建立三個資料來源池。**

3. **根據第 325 頁的 「將 LDAP 資料來源附加至資料來源池」中的指示，將資料來源附加至資料來源池。**

   - 將包含 `dc=example,dc=com` 的資料來源附加至 `data-source-pool-1`。
   - 將包含 `ou=computer,dc=example,dc=com` 的資料來源附加至 `data-source-pool-2`。
   - 將包含 `ou=people,dc=example,dc=com` 的資料來源附加至 `data-source-pool-3`。

4. **(可選擇) 配置負載平衡。**

   如需有關資訊，請參閱第 351 頁的 「配置負載平衡」。

5 **建立基底 DN 為 dc=example,dc=com、資料來源池為 data-source-pool-1 的資料檢視。**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-1 \
 data-source-pool-1 dc=example,dc=com
```

6 **建立基底 DN 為 ou=computer,dc=example,dc=com、資料來源池為 data-source-pool-2 的資料檢視。**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-2 \
 data-source-pool-2 ou=computer,dc=example,dc=com
```

7 **建立基底 DN 為 ou=people,dc=example,dc=com、資料來源池為 data-source-pool-3 的資料檢視。**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-3 \
 data-source-pool-3 ou=people,dc=example,dc=com
```

8 **檢視 excluded-subtrees 參數，以驗證子樹狀結構 ou=computer,dc=example, dc=com 與 ou=people,dc=example, dc=com 已從 dataview-1 排除。**

```
$ dpconf get-ldap-data-view-prop -h host1 -p 1389 dataview-1 excluded-subtrees
```

傳回排除的子樹狀結構清單。

9 **請視需要重新啓動目錄代理伺服器實例以使變更生效。**

如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的 「重新啓動目錄代理伺服器」。

◆ ◆ ◆ 第 20 章

20 目錄代理伺服器憑證

本章說明如何配置目錄代理伺服器憑證。如需有關配置**目錄伺服器**憑證的資訊，請參閱第 106 頁的 「管理憑證」。

本章中的程序使用 `dpadm` 與 `dpconf` 指令。如需有關這些指令的資訊，請參閱 `dpadm`(1M) 與 `dpconf`(1M) 線上手冊。

本章包含下列主題：

- 第 341 頁的 「預設的自行簽署憑證」
- 第 342 頁的 「建立、請求與安裝目錄代理伺服器的憑證」
- 第 344 頁的 「更新過期的目錄代理伺服器 CA 簽署憑證」
- 第 344 頁的 「列出憑證」
- 第 345 頁的 「將憑證從後端 LDAP 伺服器增加至目錄代理伺服器憑證資料庫」
- 第 347 頁的 「匯出憑證至後端 LDAP 伺服器」
- 第 347 頁的 「備份與復原目錄代理伺服器憑證資料庫」
- 第 348 頁的 「提示輸入存取憑證資料庫的密碼」

預設的自行簽署憑證

建立目錄代理伺服器實例時，其具有預設的自行簽署憑證。自行簽署的憑證是成對的公開與私密金鑰組，其中公開金鑰由目錄代理伺服器自行簽署。

▼ 檢視預設的自行簽署憑證

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **檢視預設的自行簽署憑證。**

```
$ dpadm show-cert instance-path defaultservercert
```

建立、請求與安裝目錄代理伺服器的憑證

若要在目錄代理伺服器上執行安全通訊端層 (SSL)，必須使用自行簽署的憑證或公開金鑰基礎架構 (PKI) 解決方案。

PKI 解決方案與外部憑證授權機構 (CA) 相關。若使用 PKI 解決方案，您需要包含公開金鑰與私密金鑰的 CA 簽署伺服器憑證。此憑證專屬於單一目錄代理伺服器實例。您還需要包含公開金鑰之**可信任的** *CA* **憑證**。可信任的 CA 憑證可確保來自您的 CA 之所有伺服器憑證皆可信任。此憑證有時亦稱爲 CA 根金鑰或根憑證。

如需有關如何建立非預設自行簽署憑證以及如何請求與安裝 CA 簽署憑證的資訊，請參閱下列程序。

▼ 建立目錄代理伺服器的非預設自行簽署憑證

當您建立目錄代理伺服器實例時，會自動提供預設的自行簽署憑證。若要以非預設設定值建立自行簽署的憑證，請使用此程序。

該程序會建立伺服器憑證的成對公開與私密金鑰組，其中公開金鑰由目錄代理伺服器簽署。自行簽署的憑證有效期限爲三個月。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **若要建立目錄代理伺服器的非預設自行簽署憑證，請鍵入：**

  ```
  $ dpadm add-selfsign-cert instance-path cert-alias
  ```

  其中 *cert-alias* 是自行簽署憑證的名稱。

  例如，您可以如下所示建立名爲 `my-self-signed-cert` 的憑證：

  ```
  $ dpadm add-selfsign-cert /local/dps my-self-signed-cert
  ```

  如需所有指令選項的說明，請參閱 `dpadm`(1M) 線上手冊，或在指令行中鍵入 `dpadm add-selfsign-cert --help`。

▼ 請求目錄代理伺服器的 CA 簽署憑證

自行簽署的憑證適用於測試目的。但是，在生產環境中，使用受信任的憑證授權單位 (CA) 憑證更爲安全。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **請求 CA 簽署的伺服器憑證。**

   ```
   $ dpadm request-cert instance-path cert-alias
   ```

其中 *cert-alias* 是請求的憑證名稱。憑證授權單位可能需要指令的所有選項以識別伺服器。如需所有指令選項的說明，請參閱 dpadm(1M) 線上手冊。

取得 CA 憑證的程序取決於所使用的 CA。有些商業 CA 提供網站供您下載憑證。其他 CA 則以電子郵件傳送憑證。

例如，您可依下列方式請求名爲 my-CA-signed-cert 的憑證：

```
$ dpadm request-cert -S cn=my-request,o=test /local/dps my-CA-signed-cert
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBYDCBygIBADAhMQ0wCwYDVQQDEwRnZXJpMRAwDgYDVQQDEwdteWNlcnQ0MIGfMA0GCSqGSIb3
DQEBAQUAA4GNADCBiQKBgQC3v9ubG468wnjBDAMbRrEkmFDTQzT+LO30D/ALLXOiElVsHrtRyWhJ
PG9cURI9uwqs15crxCpJvho1kt3SB9+yMB8Ql+CKnCQDHlNAfnn30MjFHShv/sAuEygFsN+Ekci5
W1jySYE2rzE0qKVxWLSILFo1UFRVRsUnORTX/Nas7QIDAQABoAAwDQYJKoZIhvcNAQEEBQADgYEA
fcQMnZNLpPobiX1xy1ROefPOhksVz8didY8Q2fjjaHG5lajMsqOROzubsuQ9Xh4ohT8kIA6xcBNZ
g8FRNIRAHCtDXKOdOm3CpJ8da+YGI/ttSawIeNAKU1DApF9zMb7c2lS4yEfWmreoQdXIC9YeKtF6
zwbn2EmIpjHzETtS5Nk=
-----END NEW CERTIFICATE REQUEST-----
```

透過使用 dpadm request-cert 指令請求憑證時，憑證請求是安全電子郵件 (PEM) 格式的 PKCS #10 憑證請求。PEM 是 RFC 1421 到 1424 所指定的格式。如需詳細資訊，請參閱 http://www.ietf.org/rfc/rfc1421.txt。PEM 格式表示 ASCII 格式的 Base64 編碼憑證請求。

請求 CA 簽署的憑證時，系統會建立臨時的自行簽署憑證。當您從 CA 接收並安裝 CA 簽署的憑證之後，新的憑證會取代臨時的自行簽署憑證。

2 **根據 CA 程序，將憑證請求傳送到 CA。**

傳送請求之後，必須等候 CA 回應以提供憑證。請求的回應時間各異。例如，如果是公司內部的 CA，回應時間可能很短。但如果是公司外部的 CA，可能需要數週才會回應您的請求。

3 **儲存從 CA 收到的憑證。**

將憑證儲存爲文字檔，並在安全的位置備份憑證。

▼ 安裝目錄代理伺服器的 CA 簽署伺服器憑證

若要信任 CA 簽署的伺服器憑證，必須在目錄代理伺服器實例上安裝該憑證。此程序會將 CA 憑證的公開金鑰安裝至目錄代理伺服器的憑證資料庫中。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **請檢查是否已經安裝此 CA 可信任的 CA 憑證。**

若要執行此作業，請如第 345 頁的 「列出 CA 憑證」中所述列出所有已安裝的 CA 憑證。

2. **如果尚未安裝可信任的 CA 憑證，請將該憑證增加至目錄代理伺服器實例的憑證資料庫中。**

   ```
   $ dpadm add-cert instance-path cert-alias cert-file
   ```

   其中 *cert-alias* 是可信任的 CA 憑證名稱，而 *cert-file* 是包含可信任的 CA 憑證之檔案名稱。

3. **將 CA 簽署的伺服器憑證安裝至憑證資料庫中。**

   ```
   $ dpadm add-cert instance-path cert-alias cert-file
   ```

   其中 *cert-alias* 是 CA 簽署的伺服器憑證名稱，而 *cert-file* 是包含 CA 簽署的伺服器憑證之檔案的名稱。請注意，此 *cert-alias* 必須與憑證請求中所用的 *cert-alias* 相同。

   例如，您可以使用下列方式，將名爲 `CA-cert` 的 CA 簽署的伺服器憑證增加至 `/local/dps` 上的憑證資料庫：

   ```
   $ dpadm add-cert /local/dps CA-cert /local/safeplace/ca-cert-file.ascii
   ```

更新過期的目錄代理伺服器 CA 簽署憑證

本節說明如何更新過期的 CA 簽署伺服器憑證。

▼ 更新過期的目錄代理伺服器 CA 簽署伺服器憑證

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **從 CA 取得已更新的憑證。**

2. **在目錄代理伺服器實例上安裝憑證。**

   ```
   $ dpadm renew-cert instance-path cert-alias cert-file
   ```

   其中 *cert-alias* 是新憑證的名稱，而 *cert-file* 是包含該憑證之檔案的名稱。如需所有指令選項的說明，請參閱 `dpadm`(1M) 線上手冊。

列出憑證

如需有關如何列出伺服器與 CA 憑證的資訊，請參閱下列程序。

▼ 列出伺服器憑證

此程序列出目錄代理伺服器實例上已安裝的所有憑證。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **列出目錄代理伺服器實例憑證資料庫中的伺服器憑證。**

  ```
  $ dpadm list-certs instance-path
  ```

  依預設，目錄代理伺服器實例包含名爲 `defaultservercert` 的伺服器憑證。`Same as issuer` 表示預設憑證爲自行簽署的伺服器憑證。

  例如：

```
$ dpadm list-certs /local/dps
Alias             Valid from       Expires on       Self-signed? Issued by          Issued to
----------------- ---------------- ---------------- ------------ ------------------ --------------
defaultservercert 2006/06/01 04:15 2008/05/31 04:15 y            CN=myserver:myport Same as issuer
1 certificate found.
```

▼ 列出 CA 憑證

此程序列出目錄代理伺服器實例上已安裝的 CA 憑證。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **列出目錄代理伺服器實例憑證資料庫中的 CA 憑證。**

  ```
  $ dpadm list-certs -C instance-path
  ```

  例如：

```
$ dpadm list-certs -C /local/dps
Alias   Valid from       Expires on        Built-in  Issued by    Issued to
------  ----------       ----------------  --------- ---------    ---------
CAcert1 1999/06/21 06:00 2020/06/21 06:00  y         CN=company1, O=company2
...
```

將憑證從後端 LDAP 伺服器增加至目錄代理伺服器憑證資料庫

本節說明如何將憑證從後端 LDAP 伺服器增加至目錄代理伺服器憑證資料庫。

▼ 將憑證從後端目錄伺服器增加至目錄代理伺服器憑證資料庫

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **使用以下指令語法以 PEM 格式顯示後端目錄伺服器的憑證：**

   `dsadm show-cert -F ascii` *instance-path* [*cert-alias*]

   若未指定 *cert-alias*，則會顯示預設的伺服器憑證。如需所有指令選項的說明，請參閱 `dsadm`(1M) 線上手冊。

   例如，預設的自行簽署伺服器憑證顯示爲：

```
$ dsadm show-cert -F ascii /local/ds defaultCert
-----BEGIN CERTIFICATE-----
MIICJjCCAY+gAwIBAgIFAIKL36kwDQYJKoZIhvcNAQEEBQAwVzEZMBcGA1UEChMQ
U3VuIE1pY3Jvc3lzdGVtczEZMBcGA1UEAxMQRGlyZWN0b3J5IFNlcnZlcjENMAsG
A1UEAxMEMjAxMTEQMA4GA1UEAxMHY29uZHlsZTAeFw0wNjA1MjIxMTQxNTVaFw0w
NjA4MjIxMTQxNTVaMFcxGTAXBgNVBAoTEFN1biBNaWNyb3N5c3RlbXMxGTAXBgNV
BAMTEERpcmVjdG9yeSBTZXJ2ZXIxDTALBgNVBAMTBDIwMTExEDAOBgNVBAMTB2Nv
bmR5bGUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAK9U3ry3sJmEzwQY8CGd
7S2MTZuBedo03Vea1lfDtD08WIsdDMzhHplTdeHAkWWNc8g2PDcEFXeWp9UXFMuD
Pcia7t8HtFkm73VmlriWhMd8nn3l2vkxhsPK2LHFEeOIUDR9LBBiMiEeLkjdoEhE
VLMSoYKqKI+Aa5grINdmtFzBAgMBAAEwDQYJKoZIhvcNAQEEBQADgYEAF4eDbSd7
qy2l10dIogT+rnXZ362gLTlQFCblhbGpmmptbegUdL1ITGv/62q1isPV2rW7CkjM
Cqb0fo3k5UkKKvW+JbMowpQeAPnlgpX612HuDr1tldnKV4eyU7gpG31t/cpACALQ
7OPi1A7oVb2Z8OJKfEJHkp3txBSsiI2gTkk=
-----END CERTIFICATE-----
```

2. **儲存憑證。**

   將憑證儲存爲文字檔，並在安全的位置備份憑證。

3. **將憑證從後端 LDAP 伺服器增加至目錄代理伺服器實例的憑證資料庫。**

   `$ dpadm add-cert` *instance-path cert-alias cert-file*

   其中 *cert-alias* 是憑證的名稱，而 *cert-file* 是包含該憑證之檔案的名稱。

   例如，您可以如下所示增加憑證 `defaultCert`：

```
$ dpadm add-cert /local/dps defaultCert /local/safeplace/defaultCert.ascii
```

匯出憑證至後端 LDAP 伺服器

後端 LDAP 伺服器可能需要目錄代理伺服器的憑證。本節說明如何配置目錄代理伺服器以匯出憑證至後端 LDAP 伺服器。

▼ 配置目錄代理伺服器以匯出用戶端憑證至後端 LDAP 伺服器

1 **指定要傳送至後端 LDAP 伺服器的憑證。**

```
$ dpconf set-server-prop -h host -p port ssl-client-cert-alias:cert-alias
```

其中 *cert-alias* 是憑證的名稱。如需所有指令選項的說明，請參閱 `dpconf`(1M) 線上手冊。

2 **將憑證內容複製到檔案中。**

```
$ dpadm show-cert -F ascii -o filename instance-path cert-alias
```

3 **如第 109 頁的 「增加 CA 簽署的伺服器憑證與可信任的 CA 憑證」中所述，將憑證增加至後端 LDAP 伺服器的憑證資料庫中。**

接下來的步驟 配置後端 LDAP 伺服器以執行用戶端認證。如需有關如何針對目錄伺服器執行此作業的資訊，請參閱第 116 頁的 「配置憑證層級與認證方法」。

另請參閱 如需有關配置用戶端與目錄代理伺服器之間基於憑證之認證的資訊，請參閱第 441 頁的 「配置憑證型認證」。

備份與復原目錄代理伺服器憑證資料庫

使用 `dpadm` 備份目錄代理伺服器時，即會備份伺服器憑證。備份的憑證儲存在 *archive-path*`/alias` 目錄中。

如需有關如何備份與復原目錄代理伺服器的資訊，請參閱第 319 頁的 「備份與復原目錄代理伺服器實例」。

提示輸入存取憑證資料庫的密碼

依預設，憑證資料庫的密碼由內部管理。因此，您無須鍵入憑證密碼或指定密碼檔案。當憑證資料庫透過儲存的密碼由內部管理時，密碼會儲存在安全的環境中。

若要讓憑證更安全且能夠更好地控制憑證，請配置目錄代理伺服器以提示在指令行中輸入密碼。之後，系統就會提示您輸入除 `autostart`、`backup`、`disable-service`、`enable-service`、`info`、 `restore` 與 `stop` 之外的所有 `dpadm` 子指令密碼。

如需有關配置目錄代理伺服器是否提示輸入密碼的資訊，請參閱下列程序。

▼ 提示輸入存取憑證資料庫的密碼

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **停止伺服器。**

```
$ dpadm stop instance-path
Directory Proxy Server instance 'instance-path' stopped
```

2. **將密碼提示旗標設為 `on`，然後鍵入並確認憑證資料庫密碼。**

```
$ dpadm set-flags instance-path cert-pwd-prompt=on
Choose the certificate database password:
Confirm the certificate database password:
```

3. **啓動伺服器，然後鍵入憑證資料庫密碼。**

```
$ dpadm start instance-path
Enter the certificate database password:
```

▼ 停用存取憑證資料庫的密碼提示

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **停止伺服器。**

```
$ dpadm stop instance-path
Directory Proxy Server instance 'instance-path' stopped
```

2. **將密碼提示旗標設為 `off`，然後鍵入現有的密碼。**

```
$ dpadm set-flags instance-path cert-pwd-prompt=off
Enter the old password:
```

3 啓動伺服器。

```
$ dpadm start instance-path
```

◆ ◆ ◆ 第 21 章

21 目錄代理伺服器負載平衡與用戶端相似性

如需有關負載平衡與用戶端相似性的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 16 章「Directory Proxy Server Load Balancing and Client Affinity」。本章包含下列主題：

- 第 351 頁的 「配置負載平衡」
- 第 360 頁的 「配置用戶端相似性」

配置負載平衡

如需有關負載平衡的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference 」中的「Load Balancing」。本節說明如何配置負載平衡並提供配置範例。

▼ 選取負載平衡演算法

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 透過檢視 LDAP 資料來源池的特性取得目前的負載平衡演算法。

```
$ dpconf get-ldap-data-source-pool-prop -h host -p port pool-name
```

LDAP 資料來源池的預設特性如下：

```
client-affinity-policy    :  write-affinity-after-write
client-affinity-timeout   :  20s
description               :  -
enable-client-affinity    :  false
load-balancing-algorithm  :  proportional
```

依預設，負載平衡演算法爲**比例**。

2 **配置 LDAP 資料來源池使用演算法**。

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
  load-balancing-algorithm:selected-algorithm
```

其中 *selected-algorithm* 是下列其中之一：

- **容錯移轉**
- **操作相似性**
- **比例**
- **飽和**

如需有關演算法的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Introduction to Load Balancing」。

3 **重新啓動目錄代理伺服器實例**。

```
$ dpadm restart instance-path
```

▼ 配置負載平衡加權

您需要根據資料來源池中任何其他附加資料來源的加權，來配置某個附加資料來源的加權。請考量所有附加資料來源的加權。如果資料來源針對某種作業類型的加權爲**已停用**，則該類型的請求將永遠不會傳送至該資料來源。如果資料來源的加權值爲 **0** (零)，則不會將請求分佈到該資料來源。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **檢視附加至資料來源池的資料來源清單**。

```
$ dpconf list-attached-ldap-data-sources -h host -p port pool-name
```

2 **檢視其中一個附加資料來源的特性**。

```
$ dpconf get-attached-ldap-data-source-prop pool-name \
 attached-data-source-name
```

附加資料來源的特性定義了各種作業類型的加權。附加資料來源的預設加權如下：

```
add-weight        :  disabled
bind-weight       :  disabled
compare-weight    :  disabled
delete-weight     :  disabled
modify-dn-weight  :  disabled
modify-weight     :  disabled
search-weight     :  disabled
```

3. **配置其中一個附加資料來源的加權。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name \
 attached-data-source-name add-weight:value \
 bind-weight:value compare-weight:value delete-weight:value \
 modify-dn-weight:value modify-weight:value search-weight:value
```

4. **針對其他附加資料來源重複步驟 2 與步驟 3。**

5. **比較附加資料來源的主要參數。**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
```

例如，資料來源池可以包含加權如下的資料來源：

```
$ dpconf list-attached-ldap-data-sources -h host1 -p 1389 -v myPool
SRC_NAME add-weight bind-weight compare-weight delete-weight
-------- ---------- ----------- -------------- -------------
DS-1     disabled   3               disabled       disabled
DS-2     2          2           2              2
DS-3     1          1           1              1

modify-dn-weight modify-weight search-weight
---------------- ------------- -------------
disabled         disabled      disabled
2                2             2
1                1             1
```

負載平衡的配置範例

本節包含配置每個負載平衡演算法的範例程序。

▼ 配置比例演算法以進行負載平衡

如需有關比例演算法的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Proportional Algorithm for Load Balancing」。

在此範例中，資料來源 *ds–1* 配置爲其他兩個資料來源加權的兩倍。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

開始之前 確保資料來源池至少有三個附加資料來源。如需有關如何建立資料來源及資料來源池的資訊，請參閱第 321 頁的 「建立 LDAP 資料檢視」。

1. **將資料來源池配置為使用比例演算法進行負載平衡。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
  load-balancing-algorithm:proportional
```

2 **配置第一個資料來源的特性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:2 bind-weight:2 compare-weight:2 delete-weight:2 modify-dn-weight:2 \
 modify-weight:2 search-weight:2
```

3 **配置第二個資料來源的特性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

4 **配置第三個資料來源的特性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 **比較附加資料來源的主要參數。**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
SRC_NAME add-weight bind-weight compare-weight delete-weight
-------- ---------- ----------- -------------- -------------
ds-1     2          2           2              2
ds-2     1          1           1              1
ds-3     1          1           1              1

modify-dn-weight modify-weight search-weight
---------------- ------------- -------------
2                2             2
1                1             1
1                1             1
```

6 **重新啓動目錄代理伺服器實例。**

```
$ dpadm restart instance-path
```

▼ 配置飽和演算法以進行負載平衡

如需有關飽和演算法的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Saturation Algorithm for Load Balancing」。

在此範例中，資料來源 `ds-1` 執行大部分連結作業，但不執行任何其他類型的作業。以下列加權配置三個資料來源：

- `ds-1` 以加權 3 配置連結作業，停用所有其他類型的作業。
- `ds-2` 以加權 2 配置所有作業。
- `ds-3` 以加權 1 配置所有作業。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

開始之前 確保資料來源池至少有三個附加資料來源。如需有關如何建立資料來源及資料來源池的資訊，請參閱第 321 頁的 「建立 LDAP 資料檢視」。

1. **將資料來源池配置為使用飽和演算法進行負載平衡。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:saturation
```

2. **配置第一個資料來源的特性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:disabled bind-weight:3 compare-weight:disabled delete-weight:disabled \
 modify-dn-weight:disabled modify-weight:disabled search-weight:disabled
```

3. **配置第二個資料來源的特性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:2 bind-weight:2 compare-weight:2 delete-weight:2 modify-dn-weight:2 \
 modify-weight:2 search-weight:2
```

4. **配置第三個資料來源的特性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5. **比較附加資料來源的主要參數。**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
SRC_NAME add-weight bind-weight compare-weight delete-weight
-------- ---------- ----------- -------------- -------------
ds-1     disabled   3               disabled       disabled
ds-2     2          2           2              2
ds-3     1          1           1              1

modify-dn-weight modify-weight search-weight
---------------- ------------- -------------
disabled         disabled      disabled
2                2             2
1                1             1
```

6. **重新啓動目錄代理伺服器實例。**

```
$ dpadm restart instance-path
```

▼ 配置操作相似性演算法以進行全域帳號封鎖

如需有關此演算法的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Operational Affinity Algorithm for Global Account Lockout」。

此範例有三個資料來源。將資料來源 `ds-1` 配置爲接收所有連結請求。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

開始之前 確保資料來源池至少有三個附加資料來源。如需有關如何建立資料來源及資料來源池的資訊，請參閱第 321 頁的 「建立 LDAP 資料檢視」。

1 將資料來源池配置為使用操作相似性演算法。

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:operational-affinity
```

2 配置第一個資料來源的特性。

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:1 bind-weight:100 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

3 配置第二個資料來源的特性。

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

4 配置第三個資料來源的特性。

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 比較附加資料來源的主要參數。

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
SRC_NAME add-weight bind-weight compare-weight delete-weight
-------- ---------- ----------- -------------- -------------
ds-1     1          100         1              1
ds-2     1          1           1              1
ds-3     1          1           1              1

modify-dn-weight modify-weight search-weight
---------------- ------------- -------------
1                1             1
1                1             1
1                1             1
```

6 重新啓動目錄代理伺服器實例。

```
$ dpadm restart instance-path
```

▼ 配置操作相似性演算法以進行快取最佳化

如需有關此演算法的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Operational Affinity Algorithm for Cache Optimization」。

此範例有三個資料來源。資料來源 `ds-1` 處理所有搜尋和比較作業。當 `ds-1` 回應請求時，目標項目會儲存在快取中。如果 `ds-1` 重複回應相同的請求，資料來源會使用快取的資料。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

開始之前 確保資料來源池至少有三個附加資料來源。如需有關如何建立資料來源及資料來源池的資訊，請參閱第 321 頁的 「建立 LDAP 資料檢視」。

1 將資料來源池配置為使用操作相似性演算法。

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:operational-affinity
```

2 配置第一個資料來源的特性。

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:1 bind-weight:1 compare-weight:100 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:100
```

3 配置第二個資料來源的特性。

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

4 配置第三個資料來源的特性。

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 比較附加資料來源的主要參數。

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
SRC_NAME add-weight bind-weight compare-weight delete-weight
-------- ---------- ----------- -------------- -------------
ds-1     1          1           100            1
ds-2     1          1           1              1
ds-3     1          1           1              1

modify-dn-weight modify-weight search-weight
---------------- ------------- -------------
1                1             100
```

```
1               1            1
1               1            1
```

6 **重新啓動目錄代理伺服器實例。**

```
$ dpadm restart instance-path
```

▼ 配置容錯移轉演算法以進行負載平衡

如需有關容錯移轉演算法的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Failover Algorithm for Load Balancing」。

此範例有三個資料來源。資料來源 `ds-1` 接收所有請求。如果 `ds-1` 失敗，`ds-2` 會接收所有請求直到 `ds-1` 回復爲止。如果 `ds-2` 在 `ds-1` 回復之前失敗，`ds-3` 會接收所有請求。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

開始之前 確保資料來源池至少有三個附加資料來源。如需有關如何建立資料來源及資料來源池的資訊，請參閱第 321 頁的 「建立 LDAP 資料檢視」。

1 **將資料來源池配置為使用容錯移轉演算法進行負載平衡。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:failover
```

2 **配置第一個資料來源的特性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:3 bind-weight:3 compare-weight:3 delete-weight:3 modify-dn-weight:3 \
 modify-weight:3 search-weight:3
```

3 **配置第二個資料來源的特性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:2 bind-weight:2 compare-weight:2 delete-weight:2 modify-dn-weight:2 \
 modify-weight:2 search-weight:2
```

4 **配置第三個資料來源的特性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 **比較附加資料來源的主要參數。**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
SRC_NAME add-weight bind-weight compare-weight delete-weight
-------- ---------- ----------- -------------- -------------
ds-1     3          3           3              3
ds-2     2          2           2              2
```

```
ds-3     1          1           1              1

modify-dn-weight modify-weight search-weight
---------------- ------------- -------------
3                3             3
2                2             2
1                1             1
```

6 重新啓動目錄代理伺服器實例。

```
$ dpadm restart instance-path
```

配置目錄代理伺服器執行負載平衡

簡單舉例來說，負載平衡就是將搜尋和比較作業傳送至一組目錄，並將其他作業傳送至另一組目錄。目錄代理伺服器會接收所有用戶端作業。伺服器必須決定哪一組取得讀取，而哪一組取得其他作業。

以下是配置目錄代理伺服器處理此負載平衡方案中的重要階段。

1. 增加目錄做爲目錄代理伺服器的資料來源。
2. 將資料來源增加至資料來源池。
3. 配置某些資料來源接受搜尋和比較作業，而其他資料來源則接受增加、連結、刪除、修改及修改 DN 作業。
4. 將資料來源池增加至資料檢視。

下列範例會用到偵聽連接埠 9389 的目錄代理伺服器。在此會如下所述配置代理伺服器以平衡負載。一個目錄伺服器實例 `ds1:1389`處理搜尋和比較作業，而另一個目錄伺服器實例 `ds2:2389` 則處理其他作業。

第一步是建立並啓用資料來源。此步驟需要重新啓動代理伺服器。

```
$ dpconf create-ldap-data-source -p 9389 ds1 localhost:1389
$ dpconf create-ldap-data-source -p 9389 ds2 localhost:2389
$ dpconf set-ldap-data-source-prop -p 9389 ds1 is-enabled:true
$ dpconf set-ldap-data-source-prop -p 9389 ds2 is-enabled:true
$ dpadm restart /local/dps
```

第二步是將資料來源增加至資料來源池。

```
$ dpconf create-ldap-data-source-pool -p 9389 "Directory Pool"
$ dpconf attach-ldap-data-source -p 9389 "Directory Pool" ds1 ds2
```

第三步是配置 `ds1` 接受搜尋和比較作業，而 `ds2` 接受其他作業。

```
$ dpconf set-attached-ldap-data-source-prop -p 9389 "Directory Pool" ds1 \
add-weight:disabled bind-weight:disabled compare-weight:1 delete-weight:disabled \
modify-dn-weight:disabled modify-weight:disabled search-weight:1
$ dpconf set-attached-ldap-data-source-prop -p 9389 "Directory Pool" ds2 \
add-weight:1 bind-weight:1 compare-weight:disabled delete-weight:1 \
modify-dn-weight:1 modify-weight:1 search-weight:disabled
```

第四步是將資料來源池增加至資料檢視，如此一來，用戶端應用程式的請求便會路由至該池。

```
$ dpconf create-ldap-data-view -p 9389 "Balanced View" "Directory Pool" \
dc=example,dc=com
```

配置用戶端相似性

用戶端相似性可降低負載平衡部署中的傳播延遲風險。如需有關用戶端相似性的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Client Affinity」。本節說明如何配置用戶端連線與資料來源之間的相似性，並提供配置範例。

▼ 配置用戶端相似性

本程序說明如何配置用戶端連線與資料來源之間的相似性。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 透過檢視資料來源池的特性檢視目前的負載平衡演算法。

```
$ dpconf get-ldap-data-source-pool-prop -h host -p port pool-name
```

資料來源池的預設特性如下：

```
client-affinity-policy    :  write-affinity-after-write
client-affinity-timeout   :  20s
description               :  -
enable-client-affinity    :  false
load-balancing-algorithm  :  proportional
```

下列參數可配置用戶端相似性：`client-affinity-policy`、`client-affinity-timeout` 與 `enable-client-affinity`。如需特性的說明及其有效值的清單，請鍵入：

```
dpconf help-properties ldap-data-source-pool client-affinity-policy \
 client-affinity-timeout enable-client-affinity
```

如需特性的更多資訊，請參閱下列線上手冊：`client-affinity-policy`(5dpconf)、`client-affinity-timeout`(5dpconf) 與 `enable-client-affinity`(5dpconf)。

2. **啓用用戶端相似性。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 enable-client-affinity:true
```

3. **選取用戶端相似性的策略。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:selected-policy
```

其中 *selected-policy* 是下列其中之一：

`write-affinity-after-write`
　第一個寫入請求之後的寫入請求相似性

`read-write-affinity-after-write`
　第一個寫入請求之後的所有請求相似性

`read-write-affinity-after-any`
　第一個讀取請求或寫入請求之後的所有請求相似性

`read-affinity-after-write`
　寫入請求之後的第一個讀取請求相似性

4. **配置用戶端相似性的持續時間。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-timeout:time-out[unit]
```

逾時的預設**單位**爲毫秒。

用戶端相似性的配置範例

本節包含與用戶端相似性相關的配置範例，並包含複寫延遲、驗證寫入作業以及以連線爲基礎的路由之範例。

▼ 配置當資料來源池包含主機與用戶時，複寫延遲的用戶端相似性

此程序爲首次寫入作業之後三秒內發生的所有讀取與寫入作業配置用戶端相似性。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **配置資料來源池的相似性參數。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:read-write-affinity-after-write client-affinity-timeout:3000 \
 enable-client-affinity:true
```

▼ 將用戶端相似性配置為利用讀取作業驗證每個寫入作業

此程序爲每個寫入作業之後的首次讀取作業配置用戶端相似性。此範例可用於由其中指定的連結 DN 會透過執行讀取作業驗證每個寫入作業之應用程式。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **配置資料來源池的相似性參數**。

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:read-affinity-after-write enable-client-affinity:true
```

▼ 配置以連線為基礎之路由的用戶端相似性

在 Directory Proxy Server 6.0 之前的版本中，用戶端與 LDAP 伺服器之間已開啓一個連線。來自用戶端的所有請求會使用相同的連線，直到連線關閉爲止。此路由類型稱爲**以連線為基礎的路由**。本程序說明如何配置以連線爲基礎的路由之用戶端相似性。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

開始之前 確保所有資料來源皆附加至資料來源池，且 `client-cred-mode` 設定爲 `use-client-identity`。

- **配置資料來源池的相似性參數**。

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:read-write-affinity-after-any enable-client-affinity:true
```

◆ ◆ ◆ 第 22 章

22 目錄代理伺服器分佈

利用目錄代理伺服器可透過資料檢視的定義進行分佈。資料檢視使用檢視基底定義，檢視基底則會決定該資料檢視中的項目之基底 DN。根據目錄代理伺服器中提供的分佈演算法，可以指定項目如何分佈至不同的資料檢視。

如需目錄代理伺服器分佈的簡介與使用案例範例的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 17 章「Directory Proxy Server Distribution」。

本章包含下列主題：

- 第 363 頁的 「配置目錄代理伺服器分佈演算法」
- 第 366 頁的 「配置目錄代理伺服器以分佈尾碼資料」
- 第 333 頁的 「建立與配置使用範例的資料檢視」

配置目錄代理伺服器分佈演算法

目錄代理伺服器提供下列分佈演算法：

- 模式對應
- 數值
- 字母
- 複寫
- 自訂

配置模式對應分佈演算法

目錄代理伺服器會根據請求的參數與一或多個模式之間的對應，將請求分佈至資料檢視。設定下列參數可配置模式對應分佈演算法：

- `pattern-matching-base-object-search-filter`
  `pattern-matching-base-object-search-filter`(5dpconf)

- `pattern-matching-dn-regular-expression` `pattern-matching-dn-regular-expression`(5dpconf)
- `pattern-matching-one-level-search-filter` `pattern-matching-one-level-search-filter`(5dpconf)
- `pattern-matching-subtree-search-filter` `pattern-matching-subtree-search-filter`(5dpconf)

結尾是 `filter` 的配置屬性爲 LDAP 篩選，而非常規表示式。這些 LDAP 篩選會對內送搜尋請求中所包含的 LDAP 篩選進行評估。

例如，使用下列設定可配置模式對應分佈演算法，將使用者 `uid` 爲偶數的請求傳送至 `even` 資料檢視，並將使用者 `uid` 爲奇數的請求傳送至 odd 資料檢視。

```
$ dpconf set-ldap-data-view-prop even
pattern-matching-base-object-search-filter:'|(uid=\2a)(uid=*0)(uid=*2)\
(uid=*4)(uid=*6)(uid=*8))'\
pattern-matching-one-level-search-filter:'|(uid=\2a)(uid=*0)(uid=*2)\
(uid=*4)(uid=*6)(uid=*8))'\
pattern-matching-subtree-search-filter:'|(uid=\2a)(uid=*0)(uid=*2)\
(uid=*4)(uid=*6)(uid=*8))'\
pattern-matching-dn-regular-expression:'uid=[0-9]+[02468]'
distribution-algorithm: pattern-matching

$ dpconf set-ldap-data-view-prop odd
pattern-matching-base-object-search-filter:'|(uid=\2a)(uid=*1)(uid=*3)\
(uid=*5)(uid=*7)(uid=*9))'\
pattern-matching-one-level-search-filter:'|(uid=\2a)(uid=*1)(uid=*3)\
(uid=*5)(uid=*7)(uid=*9))'\
pattern-matching-subtree-search-filter:'|(uid=\2a)(uid=*1)(uid=*3)\
(uid=*5)(uid=*7)(uid=*9))'\
pattern-matching-dn-regular-expression:'uid=[0-9]+[13579]'
distribution-algorithm: pattern-matching
```

在 `(uid=\2a)` 表示式中，`\2a` 是 * 的 ASCII 表示法，其中 `2` 與 `a` 是兩個十六進制數字。`(uid=\2a)` 表示式會確定資料檢視接受所有 `uid` 的請求。

模式對應演算法支援的語法由 Java Pattern 類別所指定 (請參閱 http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html (`http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html`))。此語法與常用之 `regex` 語法不同。

配置數值分佈演算法

目錄代理伺服器會根據請求中 RDN 的數值，將請求分佈至資料檢視。此數值擷取自資料檢視的基底 DN 下之第一個 RDN 的值。設定下列參數可定義數值界限：

- `numeric-attrs` `numeric-attrs`(5dpconf)

- `numeric-default-data-view` `numeric-default-data-view`(5dpconf)
- `numeric-lower-bound` `numeric-lower-bound`(5dpconf)
- `numeric-upper-bound` `numeric-upper-bound`(5dpconf)

例如，配置數值分佈演算法，將 `uid` 介於 0 與 99 之間的請求傳送至特定的資料檢視。爲其餘的使用者使用相同語法，但利用不同的資料檢視。

```
$ dpconf set-ldap-data-view-prop dataview distribution-algorithm:numeric \
 numeric-attrs:uid numeric-lower-bound:0 numeric-upper-bound:99
```

配置字母分佈演算法

目錄代理伺服器會根據請求中 RDN 的字母值，將請求分佈至資料檢視。字母界限擷取自資料檢視的基底 DN 下之第一個 RDN 的值。設定下列參數可定義字母界限：

- `lexicographic-attrs` `lexicographic-attrs`(5dpconf)
- `lexicographic-lower-bound` `lexicographic-lower-bound`(5dpconf)
- `lexicographic-upper-bound` `lexicographic-upper-bound`(5dpconf)

例如，配置字母分佈演算法，將名稱開頭介於 `A` 至 `M` 之間的使用者請求，傳送至一個資料檢視，並將其餘使用者的請求傳送至另一個資料檢視。

```
$ dpconf set-ldap-data-view-prop dataview distribution-algorithm:lexicographic \
 lexicographic-attrs:cn lexicographic-lower-bound:A lexicographic-upper-bound:M
```

配置複寫分佈演算法

目錄代理伺服器會根據複寫中資料檢視的角色，將請求分佈至資料檢視。演算法會將寫入作業分佈至資料來源池中的所有資料來源，並將讀取作業分佈至單一資料來源。複寫角色由 `replication-role` 參數所定義。資料檢視可有主機角色或用戶角色。

```
$ dpconf set-ldap-data-view-prop dataview distribution-algorithm:replication
```

配置自訂分佈演算法

您可以爲所有的資料檢視類型配置自訂分佈演算法，包含 `ldap-data-view`、`jdbc-data-view`、`ldif-data-view` 與 `join-data-view` 等資料檢視類型。在下列程序中，僅會設定 `ldap-data-view` 的演算法。

▼ 配置自訂分佈演算法

1. **將** `extension-jar-file-url` **特性設定為包含內有分佈演算法類別之 Java 歸檔 (JAR) 檔案的路徑。**

   ```
   $ dpconf set-server-prop -h host -p port extension-jar-file-url:jar file path
   ```

   jar file path 會被有效的 JAR 檔案路徑取代，例如 `file:/expt/dps/custom_plugin/myjar.jar`。

2. **在配置** `custom-distribution-algorithm` **之前，請將** `distribution-algorithm` **設定為** `none`。

   ```
   $ dpconf set-ldap-data-view-prop view name distribution-algorithm:none
   ```

3. **將** `custom-distribution-algorithm` **特性設定為您的自訂分佈演算法類別。**

```
$ dpconf set-ldap-data-view-prop view name custom-distribution-algorithm:PackageName.AlgoClassName
```

配置目錄代理伺服器以分佈尾碼資料

簡單舉例來說，資料分佈就是將 UID 開頭爲 A 到 M 的項目儲存至一組目錄，並將 UID 開頭爲 N 到 Z 的項目儲存至另一組目錄。目錄代理伺服器會接收所有用戶端作業。伺服器必須決定哪一組目錄處理 A 到 M，而哪一組目錄處理 N 到 Z。

以下是配置目錄代理伺服器以處理此資料分佈方案中的重要階段。

1. 增加目錄做爲目錄代理伺服器的資料來源。
2. 將資料來源增加至資料來源池，以處理不同的資料分佈。
3. 建立資料檢視，其設計目的是將用戶端請求分佈至適當的資料池。
4. 分割要載入至適當資料來源的 LDIF。
5. 將分割的 LDIF 匯入適當的資料來源。
6. 爲附加至適當資料池的資料來源，調整作業型加權。

下列範例會用到偵聽連接埠 9389 的目錄代理伺服器。爲了簡化範例，代理伺服器在此會如下所述，配置爲僅分佈在三個目錄伺服器實例之間。如需可用性與讀取延展性，請使用複寫的目錄拓樸來儲存 LDAP 資料。其中一個目錄伺服器實例 `dsA-M:1389` 會處理 UID 開頭爲 A 到 M 的使用者項目。另一個目錄伺服器實例 `dsN-Z:2389` 會處理 UID 開頭爲 N 到 Z 的使用者項目。最後一個目錄實例 `dsBase:3389`，則會處理尾碼的基底項目。

第一步是建立並啓用資料來源。基底資料來源內含不具有 UID 之尾碼根目錄附近的項目。在一般部署中，這些項目數會比分佈的項目數少很多。

```
$ dpconf create-ldap-data-source -p 9389 dsA-M localhost:1389
$ dpconf set-ldap-data-source-prop -p 9389 dsA-M is-enabled:true
```

```
$ dpconf create-ldap-data-source -p 9389 dsN-Z localhost:2389
$ dpconf set-ldap-data-source-prop -p 9389 dsN-Z is-enabled:true

$ dpconf create-ldap-data-source -p 9389 dsBase localhost:3389
$ dpconf set-ldap-data-source-prop -p 9389 dsBase is-enabled:true
```

第二步是將資料來源增加至資料來源池。

```
$ dpconf create-ldap-data-source-pool -p 9389 "Base Pool"
$ dpconf attach-ldap-data-source -p 9389 "Base Pool" dsBase

$ dpconf create-ldap-data-source-pool -p 9389 "A-M Pool"
$ dpconf attach-ldap-data-source -p 9389 "A-M Pool" dsA-M

$ dpconf create-ldap-data-source-pool -p 9389 "N-Z Pool"
$ dpconf attach-ldap-data-source -p 9389 "N-Z Pool" dsN-Z
```

第三步是建立資料檢視，其設計目的是將用戶端請求分佈至適當的資料池。請注意基底池處理 `dc=example,dc=com` 的方式，以及內含根據 UID 值分佈之資料的池處理 `ou=people,dc=example,dc=com` 的方式。此步驟需要重新啓動伺服器。

```
$ dpconf create-ldap-data-view -p 9389 "Base View" "Base Pool" \
dc=example,dc=com

$ dpconf create-ldap-data-view -p 9389 "A-M View" "A-M Pool" \
ou=people,dc=example,dc=com
$ dpconf set-ldap-data-view-prop -p 9389 "A-M View" \
distribution-algorithm:lexicographic lexicographic-attrs:uid \
lexicographic-lower-bound:a lexicographic-upper-bound:m
The proxy server will need to be restarted in order for the changes to take effect

$ dpconf create-ldap-data-view -p 9389 "N-Z View" "N-Z Pool" \
ou=people,dc=example,dc=com
$ dpconf set-ldap-data-view-prop -p 9389 "N-Z View" \
distribution-algorithm:lexicographic lexicographic-attrs:uid \
lexicographic-lower-bound:n lexicographic-upper-bound:z
The proxy server will need to be restarted in order for the changes to take effect
$ dpadm restart /local/dps
```

第四步是分割要載入至適當資料來源的 LDIF。此範例使用 `dsadm split-ldif` 指令執行初始分割及部分檔案編輯，以保留所有資料來源的頂端項目。如此即可保留指定存取控制指令的頂端項目，又可以爲每個資料來源使用單一匯入指令。

```
$ dpadm split-ldif /local/dps /local/ds6/ldif/Example.ldif /tmp/
[14/May/2007:21:14:13 +0200] - STARTUP    - INFO  - Java Version: 1.5.0_09
 (Java Home: /local/jre)
[14/May/2007:21:14:13 +0200] - STARTUP    - INFO  - Java Heap Space: Total Memory
 (-Xms) = 3MB,
```

```
 Max Memory (-Xmx) = 63MB
[14/May/2007:21:14:13 +0200] - STARTUP    - INFO  - Operating System: SunOS/sparc 5.10
[14/May/2007:21:14:15 +0200] - INTERNAL   - ERROR - Entry starting at line 0 does not
 start with a DN
[14/May/2007:21:14:15 +0200] - INTERNAL   - ERROR - Unable to parse line "# Kirsten is
 a Directory Administrator and therefore should not" of entry "uid=kvaughan, ou=People,
 dc=example,dc=com" starting at line 112 as an attribute/value pair -- no colon found.
[14/May/2007:21:14:15 +0200] - INTERNAL   - ERROR - Unable to parse line "# Robert is
 a Directory Administrator and therefore should not" of entry "uid=rdaugherty,
 ou=People, dc=example,dc=com" starting at line 298 as an attribute/value pair --
 no colon found.
[14/May/2007:21:14:16 +0200] - INTERNAL   - ERROR - Unable to parse line "# Harry is
 a Directory Administrator and therefore should not" of entry "uid=hmiller, ou=People,
 dc=example,dc=com" starting at line 556 as an attribute/value pair -- no colon found.
[14/May/2007:21:14:16 +0200] - INTERNAL   - INFO  - SplitLDIF processing complete.
  Processed 156 entries.
$ ls /tmp/*ldif
/tmp/a-m view.ldif    /tmp/base view.ldif   /tmp/n-z view.ldif
```

在進行匯入作業之前，此步驟還需要加至 LDIF 的頂端項目。

```
$ cp /local/ds6/ldif/Example.ldif /tmp/top.ldif
$ vi /tmp/top.ldif
$ cat /tmp/top.ldif
dn: dc=example,dc=com
objectclass: top
objectclass: domain
dc: example
aci: (target ="ldap:///dc=example,dc=com")(targetattr !=
 "userPassword")(version 3.0;acl "Anonymous read-search access";
 allow (read, search, compare)(userdn = "ldap:///anyone");)
aci: (target="ldap:///dc=example,dc=com") (targetattr =
  "*")(version 3.0; acl "allow all Admin group"; allow(all) groupdn =
  "ldap:///cn=Directory Administrators,ou=Groups,dc=example,dc=com";)

$ cat /tmp/top.ldif /tmp/base\ view.ldif > /tmp/top\ and\ base\ view.ldif
$ cat /tmp/top.ldif /tmp/a-m\ view.ldif > /tmp/top\ and\ a-m\ view.ldif
$ cat /tmp/top.ldif /tmp/n-z\ view.ldif > /tmp/top\ and\ n-z\ view.ldif
```

第五步是將分割的 LDIF 匯入適當資料來源。在此，處理基底項目的目錄位在連接埠 3389 上。處理 A-M 的目錄會偵聽連接埠 1389。處理 N-Z 的目錄會偵聽連接埠 2389。

```
$ dsconf import -p 1389 /tmp/top\ and\ a-m\ view.ldif dc=example,dc=com
...
Task completed (slapd exit code: 0).

$ dsconf import -p 2389 /tmp/top\ and\ n-z\ view.ldif dc=example,dc=com
...
```

```
Task completed (slapd exit code: 0).
$ dsconf import -p 3389 /tmp/top\ and\ base\ view.ldif dc=example,dc=com
...
Task completed (slapd exit code: 0).
```

第六步是爲附加至適當資料池的資料來源，調整作業型加權。如果用戶端應用程式執行搜尋以外的作業，也必須爲這些作業設定加權。

```
$ dpconf set-attached-ldap-data-source-prop -p 9389 "Base Pool" dsBase search-weight:1
$ dpconf set-attached-ldap-data-source-prop -p 9389 "A-M Pool" dsA-M search-weight:1
$ dpconf set-attached-ldap-data-source-prop -p 9389 "N-Z Pool" dsN-Z search-weight:1
```

設定作業型加權之後，用戶端應用程式便可透過目錄代理伺服器進行搜尋，而不會感到資料實際已進行了分佈。

下列搜尋會尋找 UID 開頭爲 R 的使用者。

```
$ ldapsearch -p 9389 -b dc=example,dc=com uid=rfisher
version: 1
dn: uid=rfisher, ou=People, dc=example,dc=com
cn: Randy Fisher
sn: Fisher
givenName: Randy
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
ou: Human Resources
ou: People
l: Cupertino
uid: rfisher
mail: rfisher@example.com
telephoneNumber: +1 408 555 1506
facsimileTelephoneNumber: +1 408 555 1992
roomNumber: 1579
```

下一個搜尋會尋找基底項目之一。

```
$ ldapsearch -p 9389 -b ou=groups,dc=example,dc=com cn=hr\ managers
version: 1
dn: cn=HR Managers,ou=groups,dc=example,dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: HR Managers
ou: groups
uniqueMember: uid=kvaughan, ou=People, dc=example,dc=com
uniqueMember: uid=cschmith, ou=People, dc=example,dc=com
description: People who can manage HR entries
```

建立與配置使用範例的資料檢視

本節包含下列有關資料檢視以及如何建立與配置這些檢視的資訊：

- 第 370 頁的「當子樹狀結構的不同部分儲存在不同的資料來源中時，提供單一存取點的資料檢視」
- 第 371 頁的「具階層與分佈演算法的資料檢視」

本節中的範例假設連線處理程式允許目錄代理伺服器處理所有用戶端連線。

當子樹狀結構的不同部分儲存在不同的資料來源中時，提供單一存取點的資料檢視

本節說明如何配置資料檢視，以提供子樹狀結構之不同部分的單一存取點。此範例包含具有相同基底 DN 的兩個資料檢視。數值分佈演算法用於將項目分隔至不同的資料檢視。資料來源池會配置給每組資料相同的資料來源。下圖顯示部署範例。

如需有關此部署類型的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Data Views to Route Requests When Different Parts of a Subtree Are Stored in Different Data Sources」。

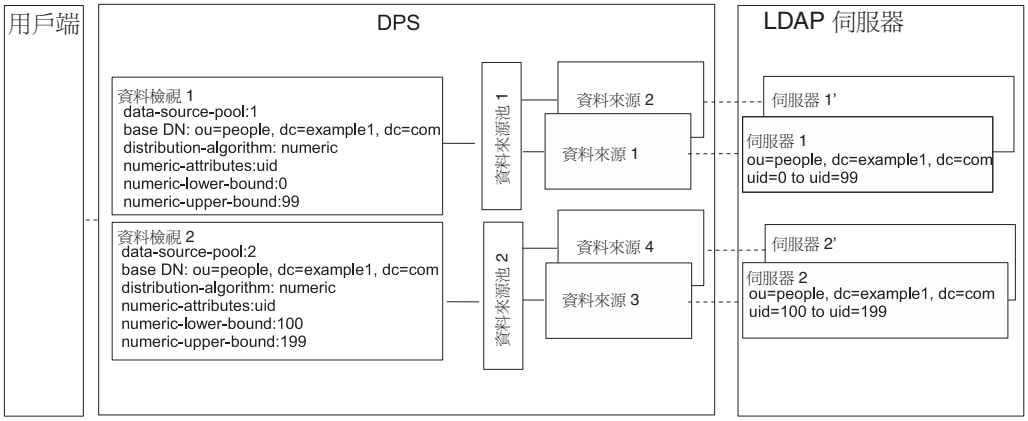


圖 22–1 當子樹狀結構的不同部分儲存在不同的資料來源中時，提供單一存取點的部署範例

▼ 配置當子樹狀結構的不同部分儲存在不同的資料來源中時，提供單一存取點的資料檢視

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1. **如第 321 頁的「建立與配置 LDAP 資料來源」中所述，建立各個 LDAP 伺服器的資料來源。**

2 **如第 324 頁的 「建立與配置 LDAP 資料來源池」中所述，建立兩個資料來源池。**

3 **如第 325 頁的 「將 LDAP 資料來源附加至資料來源池」中所述，將包含子樹狀結構一部分的資料來源附加至 `data-source-pool-1`，並將包含子樹狀結構另一部分的資料來源附加至 `data-source-pool-2`。**

4 **(可選擇) 配置負載平衡。**

如需有關資訊，請參閱第 351 頁的 「配置負載平衡」。

5 **建立具分佈演算法的資料檢視，以選取 `ou=people,dc=example,dc=com` 中 `uid` 介於 `0` 與 `99` 之間的項目，並為 `data-source-pool-1` 的直接請求配置資料檢視。**

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-1 \
 ldap-data-source-pool:data-source-pool-1 base-dn:ou=people,dc=example,dc=com \
 distribution-algorithm    :numeric numeric-attrs:uid numeric-lower-bound    :0 \
 numeric-upper-bound    :99
```

6 **建立另一個具分佈演算法的資料檢視，以選取 `ou=people,dc=example,dc=com` 中 `uid` 介於 `100` 與 `199` 之間的項目，並為 `data-source-pool-2` 的直接請求配置資料檢視。**

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-2 \
 ldap-data-source-pool:data-source-pool-2 base-dn:ou=people,dc=example,dc=com \
 distribution-algorithm:numeric numeric-attrs:uid numeric-lower-bound:100
 numeric-upper-bound    :199
```

資料檢視的其他特性和第 333 頁的 「預設資料檢視」中的預設資料檢視相同。

7 **請視需要重新啓動目錄代理伺服器實例以使變更生效。**

如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的 「重新啓動目錄代理伺服器」。

具階層與分佈演算法的資料檢視

本節說明如何配置資料檢視，以合併階層與分佈演算法。如需有關此部署類型的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Data Views With Hierarchy and a Distribution Algorithm」。

本節中的範例包含四個資料檢視。資料檢視 1 的基底 DN 是其他資料檢視的上層基底 DN。資料檢視 3 與 4 有相同的基底 DN，但是數值分佈演算法將項目分隔至不同的資料檢視。

目錄代理伺服器在子樹狀結構的從屬分支配置爲其他資料檢視的基底 DN 時，自動從資料檢視排除從屬分支。數值分佈演算法將相同子樹狀結構的項目分隔至不同的資料檢視。資料來源池會配置給每組資料相同的資料來源。

下圖顯示部署範例。

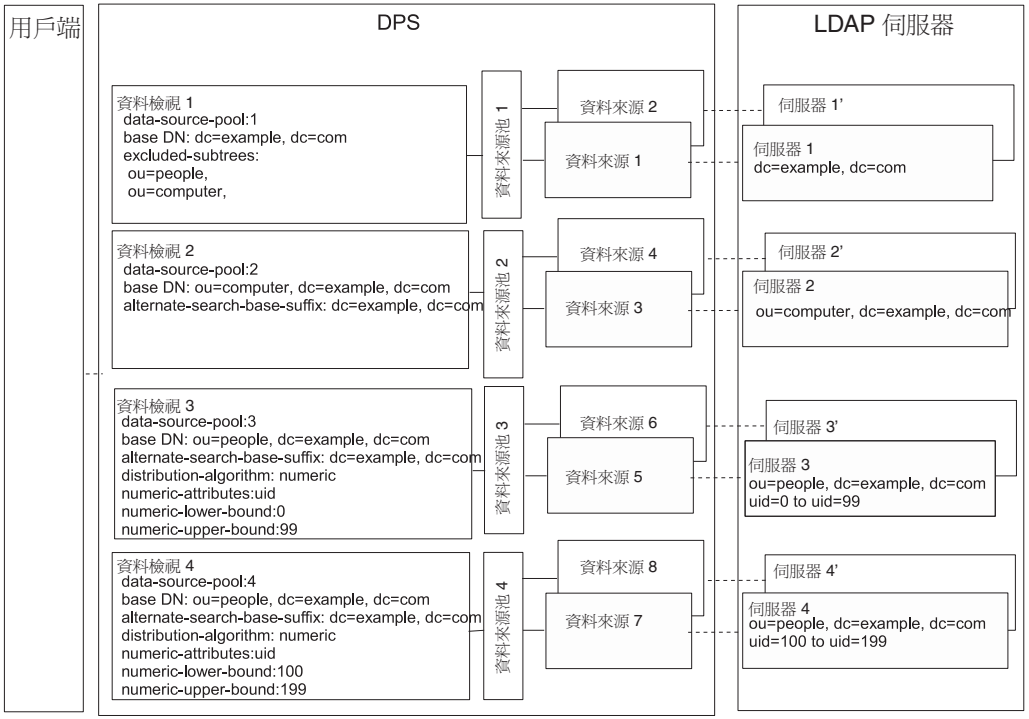


圖 22–2 具階層與分佈演算法的資料檢視範例

▼ 配置具階層與分佈演算法的資料檢視

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **如第 321 頁的 「建立與配置 LDAP 資料來源」中所述，建立各個 LDAP 伺服器的資料來源。**

2. **如第 324 頁的 「建立與配置 LDAP 資料來源池」中所述，建立四個資料來源池。**

3. **根據第 325 頁的 「將 LDAP 資料來源附加至資料來源池」中的指示，將資料來源附加至資料來源池。**

   - 將包含 `dc=example,dc=com` 的資料來源附加至 `data-source-pool-1`。
   - 將包含 `ou=computer,dc=example,dc=com` 的資料來源附加至 `data-source-pool-2`。
   - 將包含 `ou=people,dc=example,dc=com` 中 `uid` 介於 `0` 與 `99` 之項目的資料來源附加至 `data-source-pool-3`。
   - 將包含 `ou=people,dc=example,dc=com` 中 `uid` 介於 `100` 與 `199` 之項目的資料來源附加至 `data-source-pool-4`。

4. **(可選擇)配置負載平衡。**

   如需有關資訊，請參閱第 351 頁的 「配置負載平衡」。

5. **建立基底 DN 為 `dc=example,dc=com`、參考 `data-source-pool-1` 的資料檢視。**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-1 \
    data-source-pool-1 dc=example,dc=com
   ```

6. **建立基底 DN 為 `ou=computer,dc=example,dc=com` 、參考 `data-source-pool-2` 的資料檢視。**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-2 \
    data-source-pool-2 ou=computer,dc=example,dc=com
   ```

7. **建立基底 DN 為 `ou=people,dc=example,dc=com` 、參考 `data-source-pool-3` 的資料檢視。在資料檢視上配置分佈演算法以選取 `uid` 介於 `0` 與 `99` 的項目。**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-3 \
    data-source-pool-3 ou=people,dc=example,dc=com
   $ dpconf set-ldap-data-view-prop dataview-3 distribution-algorithm:numeric \
    numeric-attrs:uid numeric-lower-bound:0 numeric-upper-bound:99
   ```

8. **建立基底 DN 為 `ou=people,dc=example,dc=com`、參考 `data-source-pool-4` 的資料檢視，並在該資料檢視上配置分佈演算法以選取 `uid` 介於 `100` 與 `199` 的項目。**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-4 \
    data-source-pool-4 ou=people,dc=example,dc=com
   $ dpconf set-ldap-data-view-prop dataview-4 distribution-algorithm:numeric \
    numeric-attrs:uid numeric-lower-bound:100 numeric-upper-bound:199
   ```

9. **檢視 `excluded-subtrees` 參數，以驗證子樹狀結構 `ou=computer,dc=example, dc=com` 與 `ou=people,dc=example, dc=com` 已從 `dataview-1` 排除。**

   ```
   $ dpconf get-ldap-data-view-prop -h host1 -p 1389 dataview-1 excluded-subtrees
   ```

   傳回排除的子樹狀結構清單。

10. **重新啓動目錄代理伺服器實例，變更方可生效。**

    如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的 「重新啓動目錄代理伺服器」。

◆ ◆ ◆ 第 23 章

23 目錄代理伺服器虛擬

本章說明如何建立虛擬資料檢視。**虛擬資料檢視**會轉換資料來源，再對用戶端應用程式呈現不同的資料檢視。虛擬資料檢視包含轉換的 LDAP 資料檢視、LDIF 資料檢視、連結資料檢視及 JDBC™ 資料檢視。如需虛擬資料檢視的功能簡介與使用案例範例的說明，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 18 章「Directory Proxy Server Virtualization」。

您無法使用目錄服務控制中心 (DSCC) 執行本章中的程序。必須使用指令行。

本章包含下列主題：

- 第 375 頁的 「建立與配置 LDIF 資料檢視」
- 第 377 頁的 「定義虛擬資料檢視的存取控制」
- 第 379 頁的 「定義虛擬資料檢視的模式檢查」
- 第 380 頁的 「建立與配置連結資料檢視」
- 第 383 頁的 「建立與配置 JDBC 資料檢視」
- 第 390 頁的 「虛擬配置範例」

建立與配置 LDIF 資料檢視

LDIF 資料檢視是一種簡易式的虛擬資料檢視，其中的 LDIF 檔案會類似於 LDAP 資料來源。與 LDAP 資料檢視不相同，當設定 LDIF 資料檢視時，不建立資料來源或資料來源池，而是在建立資料檢視時指定 LDIF 檔案。依預設，無法寫入 LDIF 資料檢視。如需詳細資訊，請參閱第 377 頁的 「定義虛擬資料檢視的存取控制」。

如需有關建立與配置 LDIF 資料檢視的資訊，請參閱下列程序。

▼ 建立 LDIF 資料檢視

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 **建立 LDIF 資料檢視。**

```
$ dpconf create-ldif-data-view -h host -p port view-name path-to-ldif-file suffix-dn
```

2 **(可選擇) 檢視 LDIF 資料檢視清單。**

```
$ dpconf list-ldif-data-views -h host -p port
```

虛擬存取控制資料檢視是唯一的預設 LDIF 資料檢視。該資料檢視由伺服器產生，可將請求路由至虛擬存取控制指令 (ACI)。

▼ 配置 LDIF 資料檢視

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 **檢視 LDIF 資料檢視的特性。**

```
$ dpconf get-ldif-data-view-prop -h host -p port view-name
```

LDIF 資料檢視具有下列預設特性：

```
alternate-search-base-dn                  :  ""
alternate-search-base-dn                  :  dc=com
attr-name-mappings                        :  none
base-dn                                   :  suffixDN
bind-pwd-attr                             :  userPassword
contains-shared-entries                   :  -
db-pwd-encryption                         :  clear-text
description                               :  -
distribution-algorithm                    :  -
dn-join-rule                              :  -
dn-mapping-attrs                          :  none
dn-mapping-source-base-dn                 :  none
excluded-subtrees                         :  -
filter-join-rule                          :  -
is-enabled                                :  true
is-read-only                              :  false
is-routable                               :  true
ldif-data-source                          :  /path/to/filename.ldif
lexicographic-attrs                       :  all
lexicographic-lower-bound                 :  none
lexicographic-upper-bound                 :  none
non-viewable-attr                         :  -
non-writable-attr                         :  -
numeric-attrs                             :  all
```

```
numeric-default-data-view                    :  false
numeric-lower-bound                          :  none
numeric-upper-bound                          :  none
pattern-matching-base-object-search-filter   :  all
pattern-matching-dn-regular-expression       :  all
pattern-matching-one-level-search-filter     :  all
pattern-matching-subtree-search-filter       :  all
process-bind                                 :  -
replication-role                             :  master
viewable-attr                                :  all except non-viewable-attr
writable-attr                                :  all except non-writable-attr
```

2 變更步驟 1 中所列的一或多個特性。

```
$ dpconf set-ldif-data-view-prop -h host -p port view-name property:value \
 [property:value ... ]
```

例如，若要變更資料檢視的來源 LDIF 檔案，請設定 `ldif-data-source` 特性。

```
$ dpconf set-ldif-data-view-prop -h host1 -p 1389 -D cn="Proxy Manager" \
myLDIFDataView ldif-data-source:/local/files/example.ldif
```

定義虛擬資料檢視的存取控制

虛擬資料檢視上的 ACI 可以儲存在 LDAP 目錄或 LDIF 檔案中。如需有關虛擬 ACI 運作方式的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Access Control On Virtual Data Views」。

建立目錄代理伺服器實例時，定義虛擬存取控制的下列預設配置：

- 依預設在其中儲存 ACI 的 LDIF 檔案 (*instance-path*`/config/access_controls.ldif`)
- 名爲 `virtual access controls` 的 LDIF 資料檢視

  此資料檢視可讓目錄代理伺服器存取儲存在 LDIF 檔案中的 ACI。

▼ 定義新的 ACI 儲存庫

如果您不想使用之前所述的預設 ACI 配置，您可以定義不同的儲存庫。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 建立要在其中儲存虛擬 ACI 的儲存庫之資料檢視。

- 如果將 ACI 儲存在 LDAP 目錄中，請如第 19 章中所述，建立 LDAP 資料來源與資料檢視。
- 如果將 ACI 儲存在 LDIF 檔案中，請如第 375 頁的 「建立與配置 LDIF 資料檢視」中所述，建立 LDIF 資料檢視。

2 **將上一步驟中建立的資料檢視名稱指定為 ACI 資料檢視。**

```
$ dpconf set-virtual-aci-prop -h host -p port aci-data-view:data-view-name
```

3 **如果 ACI 儲存庫為 LDAP 目錄，請定義存取 ACI 資料檢視所需的憑證。**

```
$ dpconf set-virtual-aci-prop -h host -p port aci-manager-bind-dn:bind-dn
$ dpconf set-virtual-aci-prop -h host -p port aci-manager-bind-pwd-file:filename
```

▼ 配置虛擬存取控制

不論您使用何種 ACI 儲存庫，都必須配置虛擬存取控制。

備註 – 僅代理伺服器管理員可以建立 ACI 池，並經由 ACI 資料檢視直接管理 ACI。如果 ACI 儲存庫爲 LDAP 目錄，則必須修改該目錄的模式以包含 `aciSource` 物件類別與 `dpsaci` 屬性。如需有關自訂模式的詳細資訊，請參閱第 273 頁的 「延伸目錄伺服器模式」。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 **在 ACI 儲存庫中建立 ACI 池，並設定全域 ACI。**

如需有關全域 ACI 的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Global ACIs」。若要設定全域 ACI，請在 ACI 資料檢視的檢視基底下增加 `aciSource` 項目。例如：

```
% ldapmodify -p port -D "cn=proxy manager" -w -
dn: cn=aci-source-name,cn=virtual access controls
changetype: add
objectclass: aciSource
dpsaci: (targetattr="*") (target="ldap:///ou=people,o=virtual") (version 3.0;
 acl "perm1"; allow(all) groupdn="ldap:///cn=virtualGroup1,o=groups,o=virtual";)
cn: data-source-name
```

2 **配置一或多個連線處理程式使用此 ACI 池。**

```
% dpconf set-connection-handler-prop -h host -p port connection-handler \
aci-source:aci-source-name
```

3 **將所需的 ACI 增加至資料。**

若要執行此項作業，請建立包含 ACI 的虛擬項目。例如：

```
% ldapmodify -p port -D "cn=virtual application,ou=application users,dc=com" -w -
dn: ou=people,o=virtual
changetype: modify
add: dpsaci
dpsaci: (targetattr="*")(version 3.0; acl "perm1"; allow(all) userdn="ldap:///self";)
```

```
dpsaci: (targetattr="*")(version 3.0; acl "perm1"; allow(search, read, compare)
 userdn ="ldap:///anyone";)
```

備註 – 具有適當存取權限的任何使用者皆可透過資料檢視增加和擷取虛擬 ACI。

定義虛擬資料檢視的模式檢查

一般說來，LDAP 資料檢視的模式檢查會由後端目錄使用後端目錄的模式來執行。若要由目錄代理伺服器執行模式檢查，請使用下列程序。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

若要正規化請求，特別是 DN，請依下列方式設定伺服器的 `use-external-schema` 特性：

▼ 定義模式檢查

1 **表示伺服器實例應使用外部模式。**

```
$ dpconf set-server-prop -h host -p port use-external-schema:true
```

2 **針對連線處理程式啓用模式檢查。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler \
 schema-check-enabled:true
```

3 **建立顯示 `cn=schema` 的資料檢視。**

如果在 LDAP 目錄中定義外部模式，請如第 19 章中所述，建立檢視基底爲 `cn=schema` 的 LDAP 資料檢視。

如果在 LDIF 檔案中定義外部模式，請如第 375 頁的 「建立與配置 LDIF 資料檢視」中所述，建立檢視基底爲 `cn=schema` 的 LDIF 資料檢視。

4 **將此資料檢視增加至連線處理程式顯示的資料檢視清單中。**

依預設，連線處理程式顯示所有資料檢視。如果已定義連線處理程式顯示的自訂資料檢視清單，請將此資料檢視增加至清單中。

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler \
 data-view-routing-custom-list+:data-view-name
```

建立與配置連結資料檢視

連結資料檢視是多個資料檢視的彙總。如需有關連結資料檢視運作方式的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Join Data Views」。

如需有關如何建立與配置連結資料檢視的資訊，請參閱下列程序。

▼ 建立連結資料檢視

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **識別要彙總以形成連結檢視的主要與輔助資料檢視。**

   必須存在主要與輔助資料檢視，才能建立連結檢視。主要與輔助檢視可以是任何類型的資料檢視，包含 LDAP 資料檢視、LDIF 資料檢視、JDBC 資料檢視或其他連結資料檢視。必須配置輔助檢視上的特定特性，以允許其做爲連結檢視的來源。如需詳細資訊，請參閱第 382 頁的「配置連結檢視的輔助檢視」。

2. **建立連結資料檢視。**

   ```
   $ dpconf create-join-data-view -h host -p port view-name primary-view secondary-view \
    suffix-dn
   ```

3. **(可選擇) 檢視連結檢視清單以檢查是否已成功建立資料檢視。**

   ```
   $ dpconf list-join-data-views -h host -p port
   ```

▼ 配置連結資料檢視

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1. **檢視連結資料檢視的特性。**

   ```
   $ dpconf get-join-data-view-prop -h host -p port view-name
   ```

   連結資料檢視的預設特性如下：

   ```
   alternate-search-base-dn                    :  ""
   alternate-search-base-dn                    :  dc=com
   attr-name-mappings                          :  none
   base-dn                                     :  suffixDN
   contains-shared-entries                     :  -
   description                                 :  -
   distribution-algorithm                      :  -
   dn-join-rule                                :  -
   dn-mapping-attrs                            :  none
   ```

```
dn-mapping-source-base-dn                    :  none
excluded-subtrees                            :  -
filter-join-rule                             :  -
is-enabled                                   :  true
is-read-only                                 :  false
is-routable                                  :  true
join-rule-control-enabled                    :  false
lexicographic-attrs                          :  all
lexicographic-lower-bound                    :  none
lexicographic-upper-bound                    :  none
non-viewable-attr                            :  -
non-writable-attr                            :  -
numeric-attrs                                :  all
numeric-default-data-view                    :  false
numeric-lower-bound                          :  none
numeric-upper-bound                          :  none
pattern-matching-base-object-search-filter   :  all
pattern-matching-dn-regular-expression       :  all
pattern-matching-one-level-search-filter     :  all
pattern-matching-subtree-search-filter       :  all
primary-view                                 :  primary-view
process-bind                                 :  -
replication-role                             :  master
secondary-view                               :  secondary-view
viewable-attr                                :  all except non-viewable-attr
writable-attr                                :  all except non-writable-attr
```

2 **變更步驟 1 中所列的一或多個特性。**

```
$ dpconf set-join-data-view-prop -h host -p port view-name property:value \
 [property:value ... ]
```

例如，若要將資料來源的主要資料檢視變更爲 `myLDAPDataView`，請使用下列指令：

```
$ dpconf set-join-data-view-prop -h host1 -p 1389 -D cn="Proxy Manager" \
 myJoinDataView primary-view:myLDAPDataView
```

3 **配置連結資料檢視時，請在主要資料檢視和輔助資料檢視上設定 `viewable-attr` 和 `writable-attr` 特性。**

設定這些特性有助於在主要資料檢視與輔助資料檢視上適當地分割搜尋篩選。否則，當搜尋篩選包含輔助資料檢視的屬性時，搜尋結果可能會不一致。

4 **請視需要重新啓動目錄代理伺服器實例以使變更生效。**

如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的「重新啓動目錄代理伺服器」。

▼ 配置連結資料檢視，以啓用多個連結資料檢視對單一資料檢視的參照

在連結資料檢視中設定連結規則的配置資訊，使多個連結資料檢視可參照該資料檢視。若要達成目的，請執行下列作業：

1 **將連結資料檢視的** `join-rule-control-enabled` **設為** `true`。

```
$ dpconf set-join-data-view-prop view-name join-rule-control-enabled:true
```

將 `join-rule-control-enabled` 設爲 `true` 之後，伺服器會使用儲存在連結資料檢視中的連結規則配置資訊。如果在輔助資料檢視中儲存了連結規則配置資訊的連結資料檢視，則伺服器不會使用此資訊。若要讓伺服器使用此資訊，必須在連結資料檢視層級手動增加此配置資訊。

2 **定義確定輔助檢視如何關聯主要檢視的連結規則**。

連結規則可以是下列其中之一：

- DN 連結規則

```
$ dpconf set-join-data-view-prop view-name \
dn-join-rule:uid=\${primary-view-name.uid},ou=People,dc=example
```

- 篩選連結規則

```
$ dpconf set-join-data-view-prop view-name \
filter-join-rule:uid=\${primary-view-name.uid}
```

在上述指令中，當屬性名稱視爲變數處理時，會以 `${}` 括住。如果不使用以 `${}` 括住的屬性名稱，則會視爲常數處理。

如果在 Unix 中使用 bash 或 ksh，`\${`*primary-view-name* `.uid}` 中的 `$` 字元應如構建般由 `\` 字元退出，但 Windows 上不需要退出。

▼ 配置連結檢視的輔助檢視

必須配置輔助資料檢視上的特定特性，以允許其做爲連結檢視的來源。由於輔助檢視可以是任何類型的資料檢視，您使用的指令依賴於資料檢視類型。下列指令範例假設輔助檢視爲 LDAP 資料檢視。如需有關此處所述之特性的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Additional Secondary Data View Properties」。

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 **定義確定輔助檢視如何關聯主要檢視的連結規則**。

一律不可在連結檢視的主要資料檢視上設定 `filter-join-rule` 與 `dn-join-rule`。

連結規則可以是下列其中之一：

- DN 連結規則

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name \
dn-join-rule:uid=\${primary-view-name.uid},ou=People,dc=example
```

- 篩選連結規則

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name \
filter-join-rule:uid=\${primary-view-name.uid}
```

只有在連結資料檢視上的 `join-rule-control-enabled` 特性設爲 `false` 時，伺服器才使用 `dn-join-rule` 和 `filter-join-rule` 特性的配置。否則，如果連結資料檢視上的 `join-rule-control-enabled` 特性設爲 `true`，則會忽略輔助檢視上所設定的資訊。

2 如果在連結資料檢視上設定篩選連結規則，則必須在輔助資料檢視上設定虛擬轉換規則，才能在連結資料檢視上增加項目。

```
dpconf add-virtual-transformation secondary-view-name \
write add-attr-value dn uid=\${uid}
```

備註 – 若未設定此規則，則不可能在連結資料檢視上增加項目。

3 (可選擇) 指定輔助檢視上是否允許連結。

依預設，所有資料檢視上皆允許連結。若要禁止連結至輔助資料檢視，請執行下列指令：

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name process-bind:false
```

如需有關此特性的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Handling of Binds」。

4 (可選擇) 指定輔助檢視是否包含共用項目。

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name \
contains-shared-entries:true
```

如需有關此特性的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Handling of Shared Entries」。

建立與配置 JDBC 資料檢視

JDBC 資料檢視可讓 LDAP 用戶端應用程式得以存取關聯式資料庫。如需有關 JDBC 資料檢視運作方式的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「JDBC Data Views」。

如需有關如何建立與配置 JDBC 資料檢視的資訊，請參閱下列程序。

▼ 建立 JDBC 資料檢視

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 為關聯式資料庫建立 JDBC 資料來源。

```
$ dpconf create-jdbc-data-source -h host -p port -b db-name -B db-url -J driver-url \
[-J driver-url]... -S driver-class source-name
```

目前，每個 JDBC 資料檢視只支援一個 JDBC 資料來源。亦即，您無法在 JDBC 資料來源之間進行負載平衡。若要存取多個 JDBC 資料來源，可以建立每個資料來源的資料檢視，再使用連結資料檢視連結所有檢視。

建立 JDBC 資料來源時必須設定下列特性：

`db-name` 關聯式資料庫的名稱，例如 `payrolldb`。

`db-url` 資料庫的 URL，格式爲 `jdbc:` *vendor*`:`*driver*`://`*dbhost*`:` *dbport*。

`db-url` **不是**完整的 JDBC 資料庫 URL，因爲它不包含資料庫名稱。(資料庫名稱由 `db-name` 特性指定。)

若是 MySQL、DB2 和 Derby 資料庫，`db-url` 必須尾隨 `/`；若是 Oracle 資料庫，則必須尾隨 `:`。

`driver-class` JDBC 驅動程式類別，例如 `org.hsqldb.jdbcDriver`。

`driver-url` JDBC 驅動程式的路徑，例如 `file:///` *path/to/hsqldb*`/lib/hsqldb.jar`。

`driver-url` 特性爲多值特性。因此，`driver-url` 對 JDBC 驅動程式支援多重 JAR 檔案，以確保可對不同平台上之 JDBC 來源進行連線。

2 建立 JDBC 資料來源池。

```
$ dpconf create-jdbc-data-source-pool -h host -p port pool-name
```

3 將 JDBC 資料來源附加至 JDBC 資料來源池。

```
$ dpconf attach-jdbc-data-source -h host -p port pool-name source-name
```

4 建立 JDBC 資料檢視。

```
$ dpconf create-jdbc-data-view -h host -p port view-name pool-name suffix-DN
```

5 (可選擇) 檢視 JDBC 資料檢視清單以檢查是否已成功建立資料檢視。

```
$ dpconf list-jdbc-data-views -h host -p port
```

▼ 配置 JDBC 資料檢視

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 檢視 JDBC 資料檢視的特性。

```
$ dpconf get-jdbc-data-view-prop -h host -p port view-name
```

JDBC 資料檢視的預設特性如下：

```
alternate-search-base-dn                    :  -
attr-name-mappings                          :  none
base-dn                                     :  o=sql1
contains-shared-entries                     :  -
description                                 :  -
distribution-algorithm                      :  -
dn-join-rule                                :  -
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  -
filter-join-rule                            :  -
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
jdbc-data-source-pool                       :  pool-name
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  -
non-writable-attr                           :  -
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
process-bind                                :  -
replication-role                            :  master
viewable-attr                               :  all except non-viewable-attr
writable-attr                               :  all except non-writable-attr
```

2 變更步驟 1 中所列的一或多個特性。

```
$ dpconf set-jdbc-data-view-prop -h host -p port view-name property:value \
 [property:value ... ]
```

▼ 配置 JDBC 表格、屬性與物件類別

當您配置 JDBC 資料檢視時，還必須配置下列物件：

- **JDBC 物件類別**。將一或多個 JDBC 表格對映至 LDAP 物件類別。
- **JDBC 表格**。已針對每個關聯式資料庫表格進行定義。
- **JDBC 屬性**。定義 JDBC 表格中指定欄的 LDAP 屬性。

1 為關聯式資料庫中的每個表格建立 JDBC 表格。

```
% dpconf create-jdbc-table jdbc-table-name db-table
```

db-table 的名稱區分大小寫。請確保使用與關聯式資料庫中相同的大小寫，否則以該表格爲目標的作業可能會失敗。

2 為各關聯式資料庫表格中的每一欄建立 JDBC 屬性。

```
% dpconf add-jdbc-attr table-name attr-name sql-column
```

建立 JDBC 屬性會將表格欄對映至 LDAP 屬性。

3 (可選擇) 如果關聯式資料庫中的欄區分大小寫，請變更 JDBC 屬性的 LDAP 語法。

```
% dpconf set-jdbc-attr-prop table-name attr-name ldap-syntax:ces
```

依預設，`ldap-syntax` 的值爲 `cis`。這表示 `jdbc-attr` 不區分大小寫。如果關聯式資料庫區分大小寫，請將該值變更爲 `ces`。

某些關聯式資料庫 (例如 Oracle 與 DB2) 預設爲區分大小寫。依預設，LDAP 不區分大小寫。當目錄代理伺服器偵測到關聯式資料庫表格的某欄區分大小寫時，會將篩選內具有對應屬性的 `ldapsearch` 查詢轉譯爲使用 `UPPER` 函數的 SQL 查詢。

例如，將查詢 `ldapsearch -b "dc=mysuffix" "(attr=abc)"` 轉譯爲下列 SQL 查詢：

```
SELECT * FROM mytable WHERE (UPPER(attr)='ABC')
```

依預設，不編製此類查詢的索引。因此，會嚴重影響此類查詢的效能。

有兩種方法可以減輕對效能的影響：

- 透過將 jdbc-attr 的 `ldap-syntax` 特性設爲 `ces`。
- 透過使用 `UPPER` 函數爲每個 `jdbc-attr` 編製 LDAP 篩選可能會使用的索引。

備註 – 若關聯式資料庫不會區分大小寫，請使用 `ldap-syntax` 的預設值 `cis`。不區分大小寫的資料庫不支援 `ldap-syntax:ces`。

---

4 為 LDAP 關聯式資料庫表格建立 JDBC 物件類別。

```
% dpconf create-jdbc-object-class view-name objectclass primary-table \
  [secondary-table... ] DN-pattern
```

建立 JDBC 物件類別本質上是指定將與這些表格相關聯的 LDAP 物件類別。JDBC 物件類別還指定主要表格與輔助表格 (如果存在)。

在建立 JDBC 物件類別時，指定 DN 模式。DN 模式說明建構項目的 DN 時所使用的屬性。例如，當您將 DN 模式指定爲 `uid` 時，即會使用屬性 `uid` 與資料檢視的檢視基底建構項目的 DN。例如，`uid=bjensen,ou=people,dc=example,dc=com`。DN 模式可建構多個屬性。在此情況下，屬性應以 , (逗號) 分隔。例如，如果將 DN 模式指定爲 `uid`、`country`，資料檢視傳回的項目 DN 爲 `uid=bjensen,country=America,ou=people,dc=example,dc=com`。

在 JDBC 物件類別的 DN 模式中定義的所有子樹狀結構元件，均應有爲其定義的 JDBC 物件類別。例如，如果 JDBC 物件類別中有 DN 模式 `uid,ou`，應有 DN 模式爲 `ou` 的 JDBC 物件類別定義。這是目錄代理伺服器建構正確結構化之 DIT 的必要條件。否則，在搜尋結果中不會傳回含有 `ou=xxx,`*base-DN* 等值的子樹狀結構。

5 **如果存在輔助表格，請定義主要表格與輔助表格之間的連結規則。**

```
% dpconf set-jdbc-table-prop secondary-table-name filter-join-rule:join-rule
```

連結規則定義於輔助表格上，並確定輔助表格的資料如何連結至主要表格的資料。如何定義物件類別的主要與輔助表格之間的關係頗爲重要。如需詳細資訊，請參閱第 387 頁的 「定義 JDBC 表格之間的關係」。

6 **指定 JDBC 物件類別的超級類別。**

```
% dpconf set-jdbc-object-class-prop view-name objectclass super-class:value
```

超級類別表示 JDBC 物件類別從其繼承的 LDAP 物件類別。

定義 JDBC 表格之間的關係

在最簡單的情況下，JDBC 物件類別僅包含單一 (主要) 表格。由於沒有輔助表格，因此無須定義表格之間的關係。

如果物件類別包含多個表格，必須清楚地定義這些表格之間的關係。表格之間的關係一律定義於輔助表格上。輔助表格的下列特性可讓您定義這些關係：

- `is-single-row-table` 指定 LDAP 項目在表格中僅有一個相符列。
- `contains-shared-entries` 指定主要表格中的多個列會使用輔助表格中的某列。
- `filter-join-rule` 指出如何根據主要表格中的項目，從輔助表格中擷取項目。

以下範例說明如何根據前兩個特性的值定義篩選連結規則。這些範例假設物件類別具有一個主要表格與一個輔助表格。

範例 23–1 `is-single-row-table:true` 與 `contains-shared-entries:true`

這是這些特性的預設值。在本例中，主要與輔助表格之間的關係爲 *n->1*，亦即主要表格中有 *n* 列參照輔助表格某一共用列。

關聯式資料庫中，在主要表格中定義外來鍵 (FK)，並指向輔助表格的某欄。

例如，在某個機構中，有數名員工與同一位經理共事。將定義兩個關聯式資料庫表格，具有如下結構：

```
primary table : EMPLOYEE [ID, NAME, FK_MANAGER_ID]
secondary table : MANAGER  [ID, NAME]
```

定義下列物件類別與屬性：

```
object-class : employee
attr : name (from primary EMPLOYEE.NAME)
attr : manager (from secondary MANAGER.NAME)
```

在輔助表格中定義下列篩選連結規則：

```
ID=\${EMPLOYEE.FK_MANAGER_ID}"
```

若有多個輔助表格，則必須在每個輔助表格上配置 `filter-join-rule`。如需有關如何爲多個輔助表格配置 `filter-join-rule` 的更多資訊，請參閱步驟 11。

在此配置下，LDAP 作業的運作方式如下：

- **增加員工項目**。如果表格中的員工項目沒有經理，將新建一列。如果有經理，則使用現有的列。
- **取代項目中「manager」屬性的值**。`MANAGER.NAME` 列的值遭變更。
- **刪除員工項目**。由於共用經理項目，因此未刪除輔助表格中的該列。
- **刪除項目的「manager」屬性**。刪除輔助表格中的該列，且將外來鍵 (`EMPLOYEE.FK_MANAGER_ID`) 設爲 `NULL`。

範例 23–2 `is-single-row-table:true` 與 `contains-shared-entries:false`

在本例中，主要與輔助表格之間的關係爲 *1->1* 或 *1<-1*，亦即主要表格中有一列參照輔助表格中某一列。

在關聯式資料庫中，可能會在主要表格或輔助表格中定義外來鍵 (FK) 。

例如，在某個機構中，員工的 UID 儲存在一個表格中，而員工的姓氏則儲存在輔助表格中。將定義兩個關聯式資料庫表格，具有如下結構：

範例 23–2 is-single-row-table:true 與 contains-shared-entries:false *(續)*

```
primary table : UID [ID, VALUE, FK_SN_ID]
secondary table : SN [ID, VALUE]
```

定義下列物件類別與屬性：

```
object-class : employee
attr : uid (from primary UID.VALUE)
attr : sn (from secondary ID.VALUE)
```

在輔助表格中定義下列篩選連結規則：

```
ID=\${UID.FK_SN_ID}
```

此配置可能相反，外來鍵 FK_UID_ID 儲存在輔助表格中，並指向 UID.ID。

範例 23–3 is-single-row-table:false 與 contains-shared-entries:false

在本例中，主要與輔助表格之間的關係爲 *1->n*，亦即輔助表格有 *n* 列參照主要表格某一列。此範例說明多值屬性案例。多值屬性在輔助表格中以列集合表示，每個屬性值一列。

關聯式資料庫中，在輔助表格中定義外來鍵，並指向主要表格的某欄。

以一位員工可以有數個電話號碼的組織爲例。將定義兩個關聯式資料庫表格，具有如下結構：

```
primary table : EMPLOYEE [ID, NAME]
secondary table : PHONE [ID, VALUE, USER_ID]
```

定義下列物件類別與屬性：

```
object-class : employee
attr : cn (from primary EMPLOYEE.NAME)
attr : telephoneNumber (from secondary PHONE.VALUE)
```

在輔助表格中定義下列篩選連結規則：

```
USER_ID=\${EMPLOYEE.ID}
```

範例 23–4 is-single-row-table:false 與 contains-shared-entries:true

目錄代理伺服器目前不支援此案例。

虛擬配置範例

下一節提供兩個配置範例。這些配置說明虛擬目錄的主要功能，並指出這些功能的配置方式。

連結 LDAP 目錄與 MySQL 資料庫

本節程序說明的虛擬配置範例，會連結 LDAP 目錄與 MySQL 資料庫。LDAP 目錄是主要的資料來源，包含大部分的使用者資訊。mySQL 資料庫包含關於使用者的其他資訊。下圖說明產生的配置。

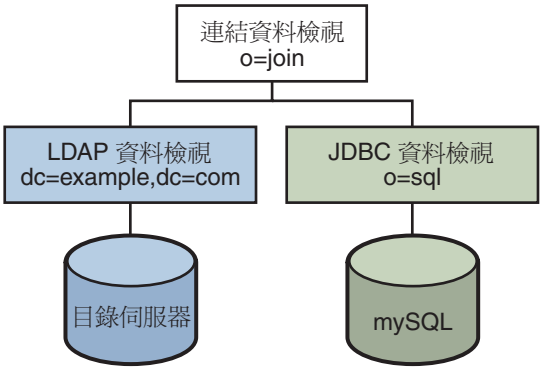


圖 23–1 虛擬配置範例

您可以使用 *install-path* `/ds6/ldif/Example.ldif` 中提供的資料範例複寫本範例，或者以自己的資料替代資料範例。

此配置可分爲三部分：

- 配置與測試 LDAP 資料檢視
- 配置與測試 JDBC 資料檢視
- 配置與測試連結資料檢視

爲簡便起見，本節中的所有指令假設目錄代理伺服器在本機的 `/local/dps` 中執行。這些指令還假設已設定下列環境變數：

`DIR_PROXY_PORT` `1389`

`LDAP_ADMIN_PWF` `pwd.txt`，包含管理員密碼的檔案。

`DIRSERV_PORT` `4389`

`LDAP_ADMIN_USER` `cn=Directory Manager`

配置與測試 LDAP 資料檢視

▼ 配置 LDAP 資料檢視

開始之前 本節中的作業假設下列資訊：

- 目錄伺服器實例在連接埠 4389 的 host1 執行。
- 目錄伺服器中的資料儲存在尾碼 dc=example,dc=com 下。若要複寫本範例，請建立目錄伺服器實例與尾碼 dc=example,dc=com，並匯入 *install-path*/ds6/ldif/Example.ldif 中的範例資料。

1 為目錄伺服器實例建立名為 myds1 的 LDAP 資料來源。

```
% dpconf create-ldap-data-source myds1 host1:4389
```

2 啓用資料來源，並允許該資料來源的寫入作業。

```
% dpconf set-ldap-data-source-prop myds1 is-enabled:true is-read-only:false
```

3 建立名為 myds1-pool 的 LDAP 資料來源池。

```
% dpconf create-ldap-data-source-pool myds1-pool
```

4 將 LDAP 資料來源附加至 LDAP 資料來源池。

```
% dpconf attach-ldap-data-source myds1-pool myds1
```

5 指定資料來源應從資料來源池接收 100% 的連結、增加、搜尋與修改作業。

```
% dpconf set-attached-ldap-data-source-prop myds1-pool myds1 add-weight:100 \
 bind-weight:100 modify-weight:100 search-weight:100
```

6 建立資料來源池的 LDAP 資料檢視，名稱為 myds1–view，基底 DN 為 dc=example,dc=com。

```
% dpconf create-ldap-data-view myds1-view myds1-pool dc=example,dc=com
```

▼ 測試 LDAP 資料檢視

1 以 dc=example,dc=com 下使用者的身份搜尋 LDAP 資料來源中的所有項目，來驗證是否可以讀取資料檢視。

```
% ldapsearch -p 1389 -D "uid=kvaughan,ou=people,dc=example,dc=com" -w bribery \
 -b dc=example,dc=com "objectclass=*"
```

備註 – 您必須使用 dc=example,dc=com 下使用者的憑證。若要使用 cn=Directory Manager，則必須定義資料檢視以處理該 DN。

---

2 以 dc=example,dc=com 下使用者的身份修改 userPassword 屬性，來驗證是否可以寫入資料檢視。

```
% ldapmodify -p 1389 -D "uid=kvaughan,ou=people,dc=example,dc=com" -w bribery
dn: uid=kvaughan,ou=people,dc=example,dc=com
changetype: modify
replace: userPassword
userPassword: myNewPassword
```

備註 – 目錄伺服器中的預設 ACI 可讓使用者修改自己的密碼。

配置與測試 JDBC 資料檢視

下列作業假設已安裝 mySQL 資料庫，此資料庫正在執行中並已填入資料，而且 mySQL 資料庫具有下列特性：

- 資料庫名稱：sample_sql
- 資料庫 URL：host2.example.com:3306/
- JDBC 驅動程式 URL：file:/net/host2.example/local/mysql/lib/jdbc.jar
- 驅動程式類別：com.mysql.jdbc.Driver
- 資料庫使用者：root
- 資料庫密碼檔案：mysqlpwd.txt

下表說明資料庫中的表格及其複合欄位。您需要此資訊才能設定 JDBC 資料檢視。

mySQL 表格	欄位
EMPLOYEE	ID、SURNAME、PASSWORD、ROOM、COUNTRY_ID
COUNTRY	ID、NAME
PHONE	USER_ID、NUMBER

▼ 配置 JDBC 資料檢視

1 為 SQL 資料庫建立名為 mysql1 的 JDBC 資料來源。

```
% dpconf create-jdbc-data-source -b sample_sql \
 -B jdbc:mysql://host2.example.com:3306/ \
 -J file:/net/host2.example/local/mysql/lib/jdbc.jar \
 -S com.mysql.jdbc.Driver mysql1
```

2 指定 SQL 資料庫的使用者名稱與密碼檔案。

```
% dpconf set-jdbc-data-source-prop mysql1 db-pwd-file:sqlpwd.txt db-user:root
```

3 **重新啓動代理伺服器。**

```
% dpadm restart /local/dps
```

4 **啓用資料來源，並允許該資料來源的寫入作業。**

```
% dpconf set-jdbc-data-source-prop mysql1 is-enabled:true is-read-only:false
```

5 **建立名為 `mysql1-pool` 的 JDBC 資料來源池。**

```
% dpconf create-jdbc-data-source-pool mysql1-pool
```

6 **將 JDBC 資料來源附加至資料來源池。**

```
% dpconf attach-jdbc-data-source mysql1-pool mysql1
```

7 **為資料來源池建立名為 `myjdbc1-view` 且基底 DN 為 `o=sql` 的 JDBC 資料檢視。**

```
% dpconf create-jdbc-data-view mysql1-view mysql1-pool o=sql
```

8 **為 MySQL 資料庫中的每個表格建立 JDBC 表格。**

```
% dpconf create-jdbc-table employee1 EMPLOYEE
% dpconf create-jdbc-table country1 COUNTRY
% dpconf create-jdbc-table phone1 PHONE
```

SQL 資料庫中表格的名稱區分大小寫。請確保使用與 SQL 資料庫中相同的大小寫。

9 **為各表格中的每一欄建立 JDBC 屬性。**

建立 JDBC 屬性會將 MySQL 欄對映至 LDAP 屬性。

```
% dpconf add-jdbc-attr employee1 uid ID
% dpconf add-jdbc-attr employee1 sn SURNAME
% dpconf add-jdbc-attr employee1 userPassword PASSWORD
% dpconf add-jdbc-attr employee1 roomNumber ROOM
% dpconf add-jdbc-attr phone1 telephoneNumber NUMBER
% dpconf add-jdbc-attr country1 countryName NAME
```

您不一定要建立 `phone1 user_id` 與 `country1 id` 欄的 JDBC 屬性，因爲這兩欄僅包含 `EMPLOYEE.ID` 中的值，且已建立其 LDAP 屬性 `uid`。

10 **為 LDAP `person` 物件類別建立 JDBC 物件類別。**

在此步驟中，將 `employee1` 表格識別爲主要表格，而將 `country1` 與 `phone1` 表格識別爲輔助表格。JDBC 物件類別的建立作業也需要 DN。在此範例中，DN 從資料檢視的 `uid` 屬性與基底 DN 中建構。

```
% dpconf create-jdbc-object-class mysql1-view person employee1 country1 phone1 uid
```

11 **定義主要表格與輔助表格之間的連結規則。**

連結規則定義於輔助表格上，並確定輔助表格的資料如何連結至主要表格的資料。

```
% dpconf set-jdbc-table-prop country1 filter-join-rule:ID=\${EMPLOYEE.COUNTRY_ID}
% dpconf set-jdbc-table-prop phone1 filter-join-rule:USER_ID=\${EMPLOYEE.ID}
```

12 指定 JDBC 物件類別的超級類別。

超級類別表示 JDBC 物件類別從其繼承屬性的 LDAP 物件類別。

```
% dpconf set-jdbc-object-class-prop mysql1-view person super-class:top
```

▼ 建立所需的 ACI

您必須透過配置 ACI 以啓用資料檢視的寫入存取，才能測試 JDBC 資料檢視。依預設拒絕對非 LDAP 資料檢視的寫入存取。就本例目的而言，增加一個允許使用者修改其密碼的全域 ACI 即已足夠。

1 以代理伺服器管理員的身份將 ACI 池增加至 JDBC 資料來源，並增加一個全域 ACI 以允許使用者修改其項目。

```
% ldapmodify -p 1389 -D "cn=proxy manager" -w password
dn: cn=mysql1,cn=virtual access controls
changetype: add
objectclass: acisource
dpsaci: (targetattr="*") (target = "ldap:///o=sql")
(version 3.0; acl "enable all access for all users "; allow(all)
userdn="ldap:///uid=kvaughan,o=sql";)
cn: mysql1
```

2 建立連線處理程式處理與 `o=sql` 網域的連線。

```
% dpconf create-connection-handler mysql1-handler
```

3 啓用連線處理程式，並將其配置為處理來自 `o=sql` 網域中使用者的所有連結。

```
% dpconf set-connection-handler-prop mysql1-handler is-enabled:true \
 bind-dn-filters:"uid=.*,o=sql"
```

4 將連線處理程式配置為使用先前增加的 ACI 池。

```
% dpconf set-connection-handler-prop mysql1-handler aci-source:mysql1
```

▼ 測試 JDBC 資料檢視

1 以 `o=sql` 下使用者的身份搜尋 JDBC 資料來源，來驗證是否可以讀取資料檢視。

```
% ldapsearch -p 1389 -D "uid=kvaughan,o=sql" -w mypwd -b o=sql "objectclass=*"
```

備註 – 您必須使用 `o=sql` 下的使用者憑證。

---

2 以 `o=sql` 下使用者的身份修改 `userPassword` 屬性，來驗證是否可以寫入資料檢視。

```
% ldapmodify -p 1389 -D "uid=kvaughan,o=sql" -w mypwd
dn: uid=kvaughan,o=sql
changetype: modify
```

```
replace: userPassword
userPassword: myNewpwd
```

建立與測試連結資料檢視

▼ 建立連結資料檢視

1 建立名為 myjoin1–view 的連結資料檢視。

將 LDAP 資料檢視指定爲主要資料檢視，而 JDBC 資料檢視爲輔助資料檢視。

```
% dpconf create-join-data-view myjoin1-view myds1-view mysql1-view o=join
```

2 在輔助資料檢視上定義連結規則。

下列連結規則指定輔助資料檢視項目的 uid 屬性應與主要資料檢視項目的 uid 屬性相符。

```
% dpconf set-jdbc-data-view-prop mysql1-view filter-join-rule:uid=\${myds1-view.uid}
```

3 如果在連結資料檢視上設定篩選連結規則，則必須在輔助資料檢視上設定虛擬轉換規則，才能在連結資料檢視上增加項目。

```
dpconf add-virtual-transformation secondary-view-name \
write add-attr-value dn uid=\${uid}
```

備註 – 若未設定此規則，則不可能在連結資料檢視上增加項目。

---

4 定義可以透過連結資料檢視從主要資料檢視讀取及寫入主要資料檢視的一組屬性。

```
% dpconf set-ldap-data-view-prop myds1-view viewable-attr:dn \
viewable-attr:cn viewable-attr:sn viewable-attr:givenName \
viewable-attr:objectClass viewable-attr:ou viewable-attr:l \
viewable-attr:uid viewable-attr:mail viewable-attr:telephoneNumber \
viewable-attr:facsimileTelephoneNumber viewable-attr:roomNumber \
viewable-attr:userPassword
% dpconf set-ldap-data-view-prop myds1-view writable-attr:dn \
writable-attr:cn writable-attr:sn writable-attr:givenName \
writable-attr:objectClass writable-attr:ou writable-attr:l \
writable-attr:uid writable-attr:mail writable-attr:telephoneNumber \
 writable-attr:facsimileTelephoneNumber writable-attr:roomNumber \
writable-attr:userPassword
```

這些定義僅套用至**連結**檢視的環境中。依預設，如果直接存取 LDAP 資料檢視，則可以讀取與寫入所有屬性。

5 定義可以透過連結資料檢視從輔助資料檢視讀取及寫入輔助資料檢視的一組屬性。

```
% dpconf set-jdbc-data-view-prop mysql1-view viewable-attr:dn \
 viewable-attr:objectclass viewable-attr:sn viewable-attr:roomNumber \
```

```
 viewable-attr:userpassword viewable-attr:jobtitle viewable-attr:countryName \
 viewable-attr:telephoneNumber
% dpconf set-jdbc-data-view-prop mysql1-view writable-attr:dn \
writable-attr:objectclass writable-attr:sn writable-attr:roomNumber \
writable-attr:userpassword writable-attr:jobtitle \
writable-attr:countryName writable-attr:telephoneNumber
```

這些定義僅套用至**連結**檢視的環境中。依預設，如果直接存取 JDBC 資料檢視，則可以讀取與寫入所有屬性。

▼ 建立所需的 ACI

1 以代理伺服器管理員的身份增加全域 ACI，來允許對連結資料檢視的匿名存取。

```
% ldapmodify -p 1389 -D "cn=proxy manager" -w password
dn: cn=myjoin1,cn=virtual access controls
changetype: add
objectclass: acisource
dpsaci: (targetattr="*") (target = "ldap:///o=join")
(version 3.0; acl "anonymous_access"; allow(all) userdn="ldap:///anyone";)
cn: myjoin1
```

2 將連線處理程式配置為使用先前增加的 ACI 池。

```
% dpconf set-connection-handler-prop default-connection-handler aci-source:myjoin1
```

▼ 測試連結資料檢視

1 以匿名使用者身份搜尋連結資料檢視。

此步驟將搜尋 Kirsten Vaughan 的項目，以查看是否同時從兩個連結檢視擷取資料。

```
% ldapsearch -p 1389 -b o=join "uid=kvaughan"
```

請注意，傳回的項目包含 LDAP 資料檢視與 JDBC 資料檢視的屬性。

2 以 `o=join` 下使用者的身份修改 `userPassword` 屬性，來驗證是否可以寫入連結資料檢視。

```
% ldapmodify -p 1389
dn: uid=kvaughan,ou=people,o=join
changetype: modify
replace: userPassword
userPassword: myPassword
```

連結多個不同的資料來源

此配置以 Example.com 機構爲例，說明虛擬目錄的部分功能可以滿足其特定的目錄服務需求。

資料儲存方案

Example.com 將組織的資料儲存在多個不同的資料來源中。爲了支援舊版，使用者資料分佈在 LDAP 目錄、平面 LDIF 檔案與 SQL 資料庫中。人力資源部門將使用者資料儲存在基底 DN 爲 `o=example.com` 的 LDAP 目錄中。薪資部門將資料儲存在 SQL 資料庫中。管理部門將部門與大樓編號等管理資料儲存在基底 DN 爲 `dc=example,dc=com` 的 LDIF 檔案中。

此外，Example.com 已買入名爲 Company22 的公司。Company22 也將其使用者資料儲存在基底 DN 爲 `dc=company22,dc=com` 的 LDAP 目錄中。

下圖提供如何儲存 Example.com 使用者資料的高階檢視。

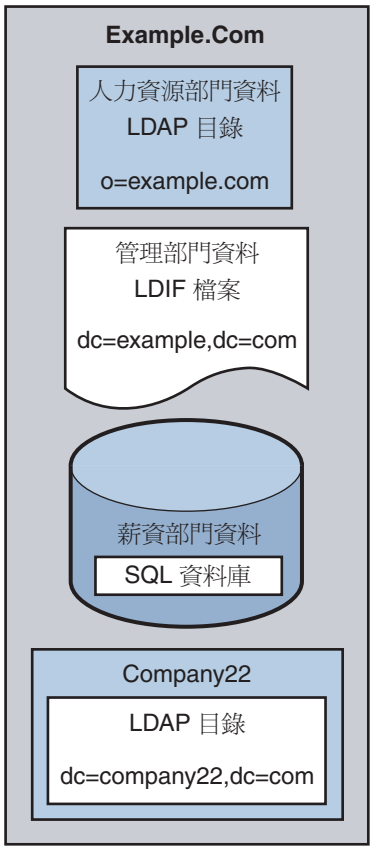


圖 23–2 不同來源中的資料儲存

用戶端應用程式需求

Example.com 具有數個必須能夠存取儲存在不同資料來源中的資料的 LDAP 用戶端應用程式。這些用戶端應用程式的需求不盡相同。因此需要不同的資料檢視。在某些情況

下，用戶端必須彙總資料。此外，某些用戶端應用程式必須能夠存取 Company22 的使用者資料，以便能夠同時管理 Example.com 的新舊員工。

下圖提供 Example.com 用戶端應用程式需求的高階檢視。

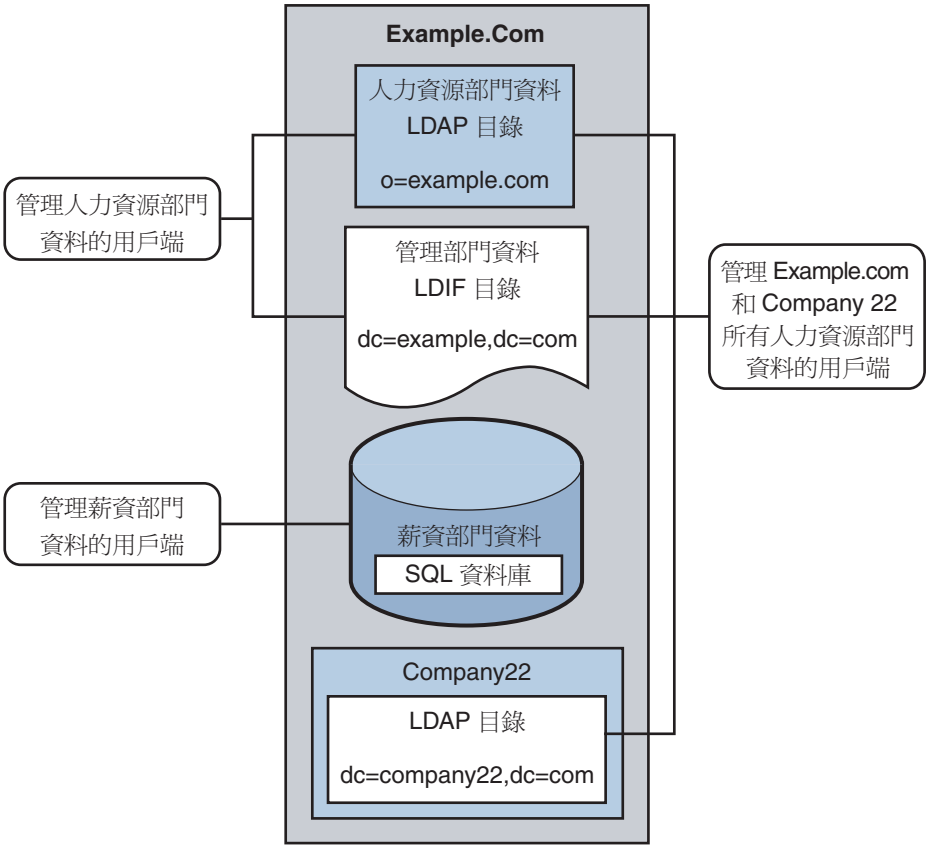


圖 23–3 用戶端應用程式需求

以下幾節引導您充分配置目錄代理伺服器資料檢視，以滿足本範例方案中所述之用戶端應用程式需求。如需有關資料檢視運作方式的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 17 章「Directory Proxy Server Distribution」與「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 18 章「Directory Proxy Server Virtualization」。

範例方案的配置分爲下列幾節：

- 第 399 頁的 「彙總來自人力資源部門 LDAP 目錄與管理部門 LDIF 檔案的資料」
- 第 401 頁的 「透過重新命名 DN 將資料從 Company22 增加至 Example.Com 的 DIT」
- 第 402 頁的 「將 Company 22 的資料增加至人力資源部門的資料」
- 第 403 頁的 「讓 LDAP 用戶端存取 SQL 資料庫中的薪資部門資料」

- 第 406 頁的「增加虛擬存取控制」

彙總來自人力資源部門 LDAP 目錄與管理部門 LDIF 檔案的資料

人力資源部門儲存員工姓名、開始工作日期與職等等資訊。管理部門儲存其他資料，例如大樓代碼與辦公室編號。處理人力資源部門資料的用戶端應用程式必須能夠存取合併自這兩個來源的資料。這兩個資料來源具有存在於每個項目中的共用屬性 `employeeNumber`。

下圖說明用戶端應用程式的需求。

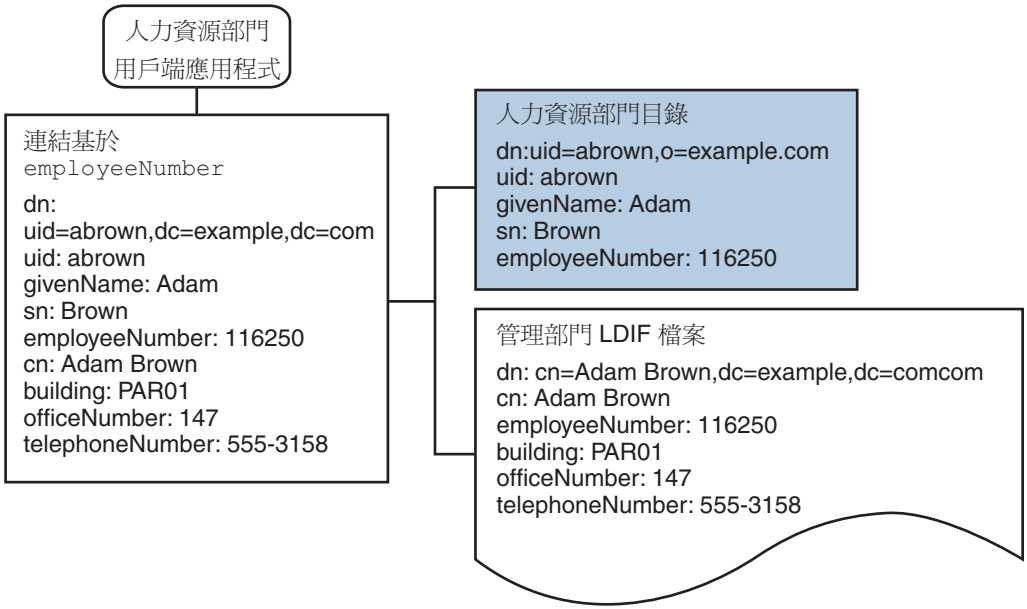


圖 23–4 來自 LDAP 目錄與 LDIF 檔案的資料彙總

若要滿足此應用程式需求，必須爲薪資部門的目錄與管理部門的 LDIF 檔案建立資料檢視。然後，連結這兩個資料檢視以提供對彙總資料的存取。此共用屬性可讓目錄代理伺服器彙總每個使用者的資料。

爲簡便起見，本節中所用的指令假設下列資訊：

- 目錄代理伺服器實例在本機上執行，並具有預設 LDAP 連接埠 (389)。
- 目錄代理伺服器實例位於 `/local/myDPS`。
- 已將包含代理伺服器管理員密碼之檔案的路徑設定爲變數 `LDAP_ADMIN_PWF`。如需有關設定目錄代理伺服器環境變數的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」中的「Environment Variables」。
- 薪資部門的 LDAP 目錄在連接埠 `2389` 上名爲 `payrollHost` 的主機上執行。

- 用以儲存管理部門資料的 LDIF 檔案名爲 `example.ldif`。

若要取得每個指令的完整語法，請執行不含任何選項的指令。例如：

```
$ dpconf create-ldap-data-view
Operands are missing
Usage: dpconf create-ldap-data-view VIEW_NAME POOL_NAME SUFFIX_DN
```

▼ 建立與啓用薪資部門目錄的 LDAP 資料檢視

1. **為薪資部門目錄建立 LDAP 資料來源。**

   ```
   $ dpconf create-ldap-data-source payroll-directory payrollHost:2389
   ```

2. **為薪資部門目錄建立 LDAP 資料來源池。**

   ```
   $ dpconf create-ldap-data-source-pool payroll-pool
   ```

3. **將薪資部門資料來源附加至資料來源池。**

   ```
   $ dpconf attach-ldap-data-source payroll-pool payroll-directory
   ```

4. **配置附加資料來源的加權。**

   ```
   $ dpconf set-attached-ldap-data-source-prop -h payrollHost -p 2389 \
   payroll-pool payroll-directory add-weight:2 \
   bind-weight:2 compare-weight:2 delete-weight:2 \
   modify-dn-weight:2 modify-weight:2 search-weight:2
   ```

5. **建立薪資部門目錄的 LDAP 資料檢視。**

   ```
   $ dpconf create-ldap-data-view payroll-view payroll-pool o=example.com
   ```

6. **啓用 LDAP 資料檢視以將用戶端請求路由至此資料檢視。**

   ```
   $ dpconf set-ldap-data-view-prop payroll-view is-enabled:true
   ```

7. **重新啓動目錄代理伺服器以使變更生效。**

   ```
   $ dpadm restart /local/myDPS
   ```

▼ 建立與啓用管理部門資料的 LDIF 資料檢視

1. **建立管理部門資料的 LDIF 資料檢視。**

   ```
   $ dpconf create-ldif-data-view admin-view example.ldif dc=example,dc=com
   ```

2. **啓用管理部門資料的 LDIF 資料檢視。**

   ```
   $ dpconf set-ldif-data-view-prop admin-view is-enabled:true
   ```

3 **指定管理部門檢視包含薪資部門檢視中多個項目所使用的項目。**

```
$ dpconf set-ldif-data-view-prop admin-view contains-shared-entries:true
```

當此特性設爲 TRUE 時，刪除薪資部門資料檢視中的某項目，不會導致刪除管理部門資料檢視中的共用項目。如果某項目不存在，將該項目增加至薪資部門資料檢視僅會將其增加至輔助資料檢視。

4 **重新啓動目錄代理伺服器以使變更生效。**

```
$ dpadm restart /local/myDPS
```

▼ 連結薪資部門資料檢視與管理部門資料檢視

1 **建立管理部門資料檢視的篩選連結規則，以指定彙總資料的方式。**

下列連結規則指定應根據使用者項目的 `employeeNumber` 屬性連結資料。

```
$ dpconf set-ldif-data-view-prop admin-view \
filter-join-rule:employeeNumber=\${payroll-view.employeeNumber}
```

2 **建立彙總這兩個資料檢視的連結資料檢視。**

對於此連結資料檢視，該機構使用尾碼 DN `dc=example,dc=com`。

```
$ dpconf create-join-data-view example-join-view payroll-view admin-view \
dc=example,dc=com
```

透過重新命名 DN 將資料從 Company22 增加至 Example.Com 的 DIT

Company22 的使用者資料儲存在 DN `dc=company22,dc=com` 下。雖然 Example.com 希望在大多數情況下能夠單獨儲存此使用者資料，但卻只有一個用戶端應用程式同時管理 Company 22 員工與其他 Example.com 員工。此用戶端應用程式要求 Company22 的使用者資料必須**類似**於 Example.com 的資料。

下圖說明用戶端應用程式的需求。

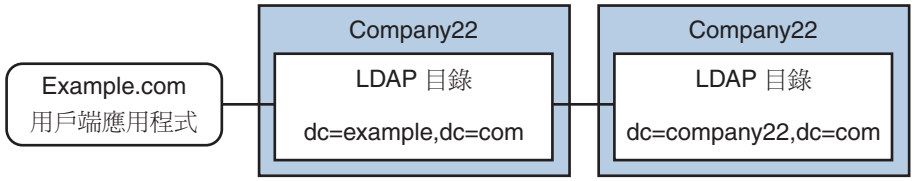


圖 23–5 DN 重新命名

若要滿足此應用程式需求，必須爲 Company22 的目錄建立虛擬 DN 爲 `dc=example,dc=com` 的資料檢視。

爲簡便起見，本節中所用的指令假設下列資訊：

- 目錄代理伺服器實例在本機上執行，並具有預設 LDAP 連接埠 (389)。
- 目錄代理伺服器實例位於 `/local/myDPS`。
- 已將包含代理伺服器管理員密碼之檔案的路徑設定爲變數 `LDAP_ADMIN_PWF`。如需有關設定目錄代理伺服器環境變數的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」中的「Environment Variables」。
- Company 22 的 LDAP 目錄在連接埠 `2389` 上名爲 `company22Host` 的主機上執行。

▼ 為 Company 22 的目錄建立含虛擬 DN 的資料檢視

1. **為 Company 22 的目錄建立 LDAP 資料來源。**

```
$ dpconf create-ldap-data-source company22-directory company22Host:2389
```

2. **為 Company 22 的目錄建立 LDAP 資料來源池。**

```
$ dpconf create-ldap-data-source-pool company22-pool
```

3. **將 Company 22 的資料來源附加至資料來源池。**

```
$ dpconf attach-ldap-data-source company22-pool company22-directory
```

4. **配置附加資料來源的加權。**

```
$ dpconf set-attached-ldap-data-source-prop -h company22Host -p 2389 \
company22-pool company22-directory add-weight:2 \
bind-weight:2 compare-weight:2 delete-weight:2 \
modify-dn-weight:2 modify-weight:2 search-weight:2
```

5. **為 Company 22 的目錄建立虛擬 DN 為 `dc=example,dc=com` 的 LDAP 資料檢視。**

```
$ dpconf create-ldap-data-view company22-view company22-pool dc=example,dc=com
```

6. **指示目錄代理伺服器將此虛擬 DN 對映至 Company 22 目錄中的實際 DN。**

```
$ dpconf set-ldap-data-view-prop company22-view \
dn-mapping-source-base-dn:dc=company22,dc=com
```

7. **啓用 Company 22 目錄的 LDAP 資料檢視，以將用戶端請求路由至此資料檢視。**

```
$ dpconf set-ldap-data-view-prop company22-view is-enabled:true
```

8. **重新啓動目錄代理伺服器以使變更生效。**

```
$ dpadm restart /local/myDPS
```

將 Company 22 的資料增加至人力資源部門的資料

人力資源部門需要 Example.com 與新買入的 Company 22 人力資源部門資料的彙總檢視。下圖說明人力資源部門全域應用程式的需求。

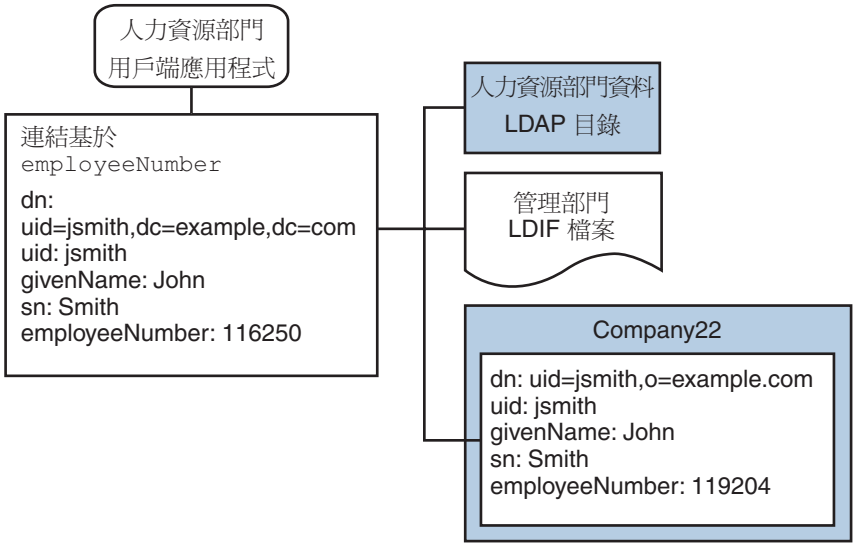


圖 23–6 來自連結資料檢視與 LDAP 資料檢視的資料彙總

▼ 連結範例連接資料檢視與 Company 22 資料檢視

1 **建立 Company 22 資料檢視的篩選連結規則，以指定彙總資料的方式。**

下列連結規則指定應根據使用者項目的 `employeeNumber` 屬性連結資料。

```
$ dpconf set-ldif-data-view-prop company22-view \
filter-join-rule:employeeNumber=\${example-join-view.employeeNumber}
```

2 **建立彙總 Company 22 的資料檢視與 Example.com 連結資料檢視的連結資料檢視。**

```
$ dpconf create-join-data-view global-join-view example-join-view \
company22-view dc=example,dc=com
```

讓 LDAP 用戶端存取 SQL 資料庫中的薪資部門資料

Example.com 的薪資部門將薪水資料儲存在 SQL 資料庫中。該資料庫有兩個表格，分別爲 `employee` 表格與 `salary` 表格。Example.com 具有要求能夠存取那些資料的 LDAP 用戶端應用程式。用戶端應用程式要求 SQL 資料必須**類似**於 LDAP 資料。

下圖說明用戶端應用程式的需求。

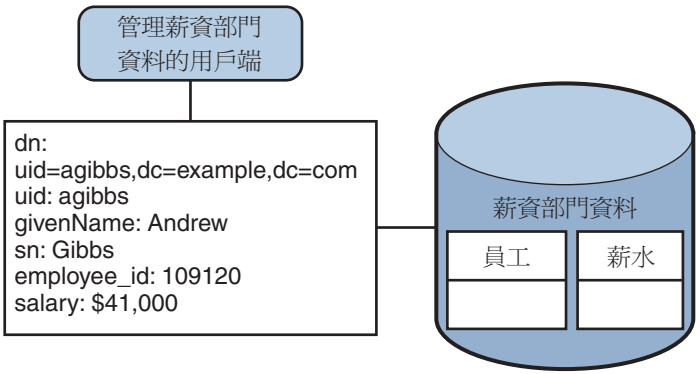


圖 23–7 提供 SQL 資料庫存取的 JDBC 資料檢視

若要滿足此應用程式需求，必須建立 JDBC 資料檢視將 SQL 表格中的欄對映至 LDAP 屬性。

爲簡便起見，本節中所用的指令假設下列資訊：

- 目錄代理伺服器實例在本機上執行，並具有預設 LDAP 連接埠 (389)。
- 目錄代理伺服器實例位於 `/local/myDPS`。
- 已將包含代理伺服器管理員密碼之檔案的路徑設定爲變數 `LDAP_ADMIN_PWF`。如需有關設定目錄代理伺服器環境變數的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Installation Guide」中的「Environment Variables」。
- SQL 資料庫已啓動且正在執行。
- 已將 `JAVA_HOME` 變數設定爲正確的 Java 路徑。
- SQL 資料庫的密碼儲存在 `myPasswordFile` 檔案中的 `myPassword`。

▼ 為 Example.com 的薪資部門資料庫建立 JDBC 資料檢視

1 為薪資部門資料庫建立 JDBC 資料來源。

```
$ dpconf create-jdbc-data-source -b payrollsqldb \
  -B jdbc:payrollsqldb:payrollsql://localhost/ \
  -J file://payrollsqldb.jar \
  -S org.payrollsqldb.jdbcDriver payroll-src
```

2 以 SQL 資料庫的特性配置 JDBC 資料來源。

```
$ dpconf set-jdbc-data-source-prop payroll-src \
  db-user:proxy db-pwd-file:password-file-location/myPasswordFile
```

3 啓用 JDBC 資料來源。

```
$ dpconf set-jdbc-data-source-prop payroll-src is-enabled:true
```

4 **為薪資部門資料庫建立 JDBC 資料來源池。**

```
$ dpconf create-jdbc-data-source-pool payroll-pool
```

5 **將薪資部門資料來源附加至資料來源池。**

```
$ dpconf attach-jdbc-data-source payroll-pool payroll-src
```

6 **為薪資部門資料庫建立虛擬 DN 為 o=payroll 的 JDBC 資料檢視。**

```
$ dpconf create-jdbc-data-view payroll-view payroll-pool o=payroll
```

7 **為 SQL 資料庫中的每個表格建立 JDBC 表格。**

```
$ dpconf create-jdbc-table jdbc-employee employee
$ dpconf create-jdbc-table jdbc-salary salary
```

8 **為 SQL 表格中的每一欄增加 JDBC 屬性。**

```
$ dpconf add-jdbc-attr jdbc-employee eid employee_id
$ dpconf add-jdbc-attr jdbc-employee first firstname
$ dpconf add-jdbc-attr jdbc-employee last lastname
$ dpconf add-jdbc-attr jdbc-employee description description
$ dpconf add-jdbc-attr jdbc-employee spouse spousename
$ dpconf add-jdbc-attr jdbc-salary salary salary
$ dpconf add-jdbc-attr jdbc-salary social ssn
```

9 **指定可以透過 JDBC 資料檢視進行檢視與寫入的屬性。**

```
$ dpconf set-jdbc-data-view-prop payroll-view \
 viewable-attr:eid \
 viewable-attr:first \
 viewable-attr:last \
 viewable-attr:desc \
 viewable-attr:spouse \
 viewable-attr:salary \
 viewable-attr:social
 $ dpconf set-jdbc-data-view-prop payroll-view \
 writable-attr:eid \
 writable-attr:first \
 writable-attr:last \
 writable-attr:description \
 writable-attr:spouse \
 writable-attr:salary \
writable-attr:social
```

10 **建立對映至 LDAP 物件類別的 JDBC 物件類別。**

下列指令建立對映至 LDAP person 物件類別的物件類別。該物件類別指定員工表格應作爲主要表格使用，而薪資表格應用做輔助表格。eid 屬性應用來建構 DN。

```
$ dpconf create-jdbc-object-class payroll-view \
 person jdbc-employee jdbc-salary eid
```

11 **在輔助表格上建立篩選連結規則，以指定輔助表格的資料如何連結至主要表格的資料。**

下列連結規則指定應根據 `employee_id` 屬性連結資料。

```
$ dpconf set-jdbc-table-prop jdbc-salary \
filter-join-rule:employee_id=\${employee.employee_id}
```

12 **建立 JDBC 物件類別的超級類別。**

```
$ dpconf set-jdbc-object-class-prop payroll-view person super-class:extensibleObject
```

增加虛擬存取控制

透過在 LDAP 目錄中定義 ACI 可處理該目錄的存取控制。透過虛擬資料檢視存取資料來源時，必須定義 ACI 僅會套用至透過這些資料檢視進行檢視的資料。

連線處理程式控制透過目錄代理伺服器的所有存取。如需有關連線處理程式的資訊，請參閱第 26 章。

▼ 增加允許匿名存取的 ACI

1 **增加 ACI。**

```
$ ldapadd -v -D "cn=proxy manager" -w password -p 389
dn: cn=ldifonly-acis,cn=virtual access controls
objectclass: top
objectclass: aciSource
cn: ldifonly-acis
dpsaci: (targetattr="*")(version 3.0; acl "anonymous_access"; allow(all) \
(userdn="ldap:///anyone");)
```

2 **將連線處理程式指向虛擬 ACI。**

```
$ dpconf set-connection-handler-prop anonymous aci-source:ldifonly-acis
```

3 **啓用連線處理程式。**

```
$ dpconf set-connection-handler-prop anonymous is-enabled:true
```

◆ ◆ ◆ 第 24 章

24 虛擬資料轉換

虛擬資料轉換定義於現有的資料檢視上，並可讓您建立使用虛擬資料檢視的虛擬資料。如需有關其運作方式的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Virtual Data Transformations」。

本章包含下列主題：

- 第 407 頁的「配置虛擬資料轉換」
- 第 408 頁的「虛擬轉換範例」

配置虛擬資料轉換

您可以將虛擬資料轉換增加至任何類型的資料檢視：LDAP 資料檢視、LDIF 資料檢視、連結資料檢視或 JDBC 資料檢視。

▼ 增加虛擬轉換

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 將轉換增加至資料檢視。

```
$ dpconf add-virtual-transformation -h host -p port view-name \
 transformation-model transformation-action attribute-name [parameters...]
```

transformation-model 可以是 `mapping`、`write` 或 `read` 轉換之一。

transformation-action 可以是 `add-attr`、`remove-attr`、`add-attr-value`、`remove-attr-value`、`def-value` 或 `attr-value-mapping` 動作之一。

請注意，根據 *transformation-model* 與 *transformation-action* 之不同，有可能一定要 *parameters*。

如需有關轉換模型、轉換動作及轉換參數的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Virtual Data Transformations」。

2. **(可選擇) 檢視在資料檢視上定義的虛擬轉換的清單。**

   ```
   $ dpconf list-virtual-transformations -h host -p port view-name
   ```

▼ 移除虛擬轉換

- **使用下列指令移除虛擬轉換。**

  ```
  dpconf remove-virtual-transformation view_name transformation_name
  ```

虛擬轉換範例

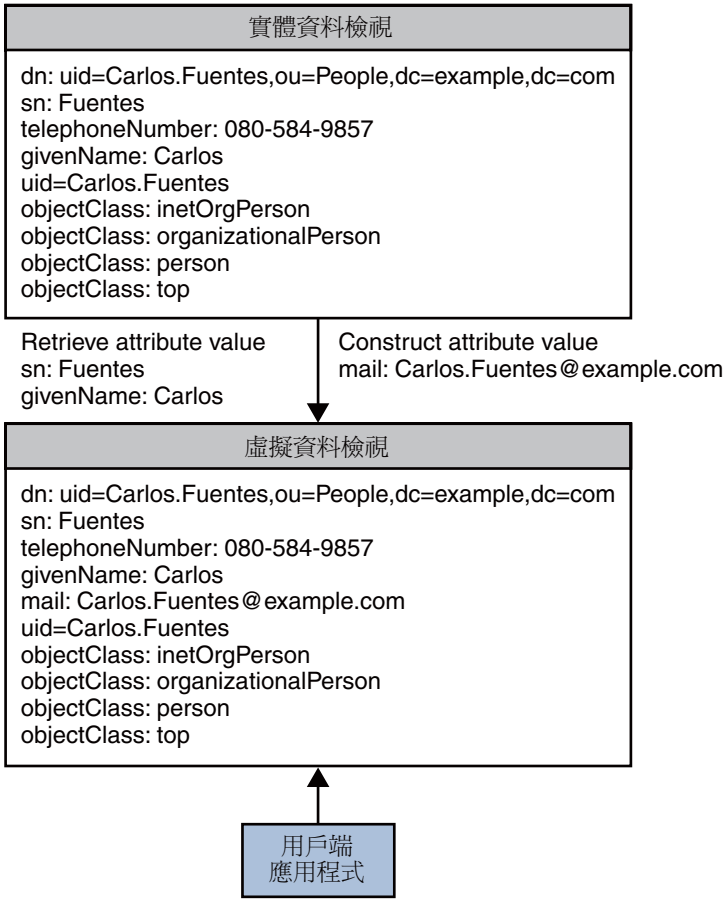
以下小節提供需要虛擬資料檢視的使用案例，以及執行使用案例時所需的轉換模型與動作之組合。

從項目的現有屬性導出屬性

使用下列轉換規則從項目的現有屬性導出屬性。例如，套用下列轉換規則時，會顯示源自 `givenName` 與 `sn` 屬性的 `mail` 屬性。

```
$ dpconf add-virtual-transformation dataview1 read add-attr \
mail \${givenName}.\${sn}@example.com
```

下圖表示當搜尋傳回使用者項目時，使用者項目所發生的轉換。



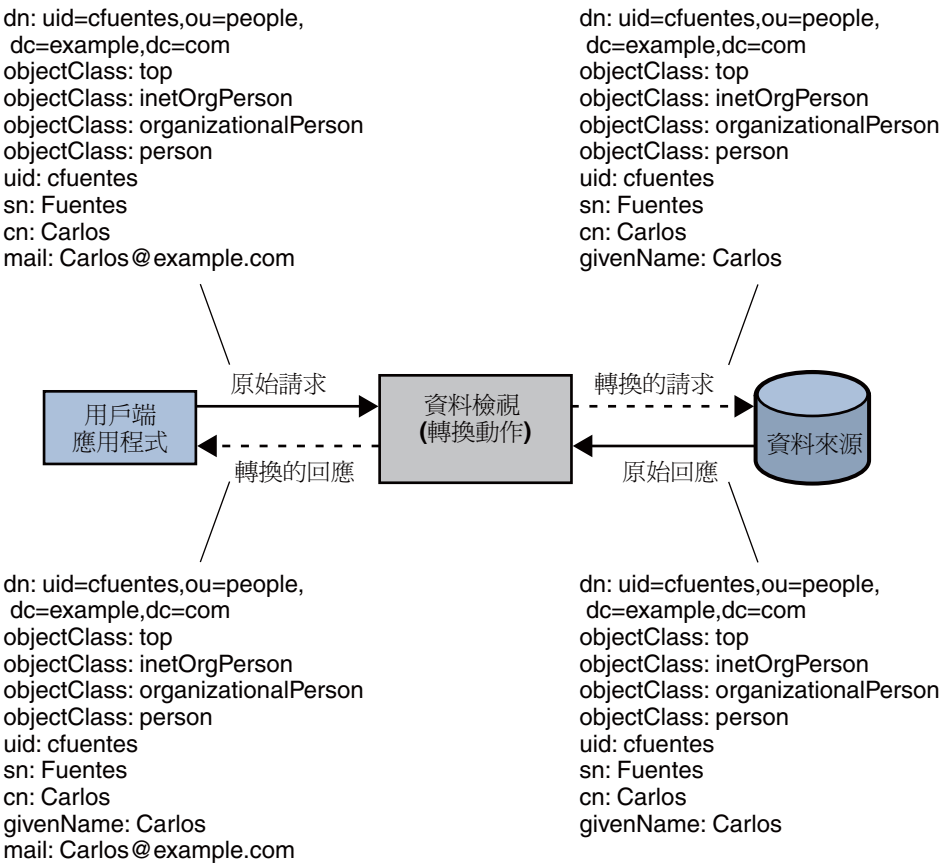
將虛擬屬性對映至實體屬性

使用下列對映轉換規則，增加當作純虛擬屬性一部分所提供的屬性。例如，套用下列轉換規則時，即使未在項目中指定，也會在伺服器中儲存 `givenName`。該值擷取自定義爲 `mail \${givenName}@example.com` 的純虛擬屬性。

```
$ dpconf add-virtual-transformation dataview1 mapping add-attr \
mail \${givenName}@example.com
```

首先增加包含虛擬屬性 `mail` (而非 `givenName` 屬性) 的項目。虛擬轉換會產生 `givenName` 屬性的值，並連同 `givenName` 一起儲存項目，而不儲存 `mail` 屬性。然後使用 `uid` 屬性執行搜尋，擷取 `givenName` 的值，而相同的虛擬轉換會產生虛擬屬性 `mail` 的值。

下圖表示使用者項目發生的轉換。

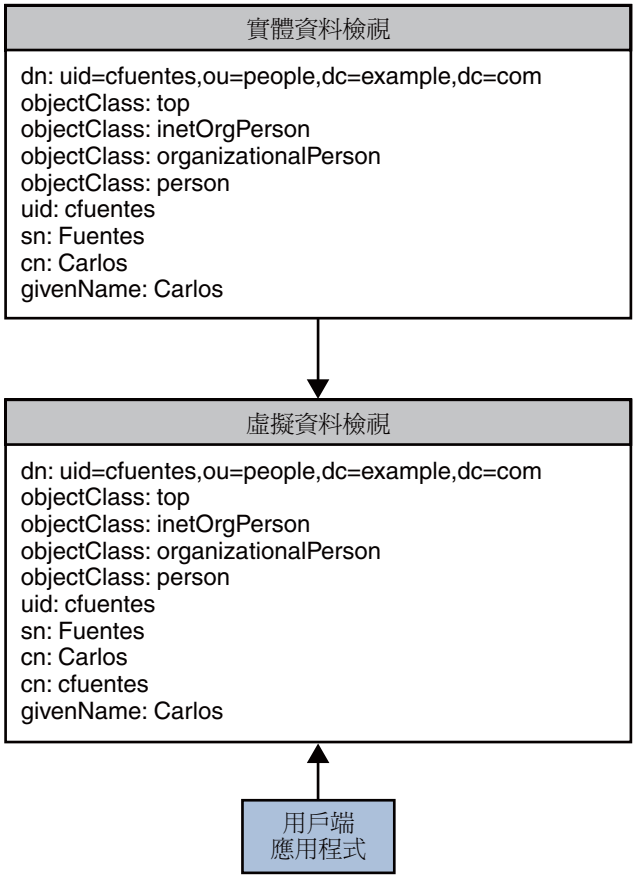


顯示第二個虛擬屬性值，由其他實體屬性指定

使用下列轉換顯示由其他屬性指定的屬性值。例如，顯示 `uid` 作爲 `cn`，以及顯示已儲存在項目中的 `cn` 值。下列指令行不會將其他值儲存至 `cn`，但會在結果傳回用戶端之前套用轉換。

```
$ dpconf add-virtual-transformation dataview1 read add-attr-value cn \${uid}
```

下圖表示當搜尋傳回使用者項目時，使用者項目所發生的轉換。

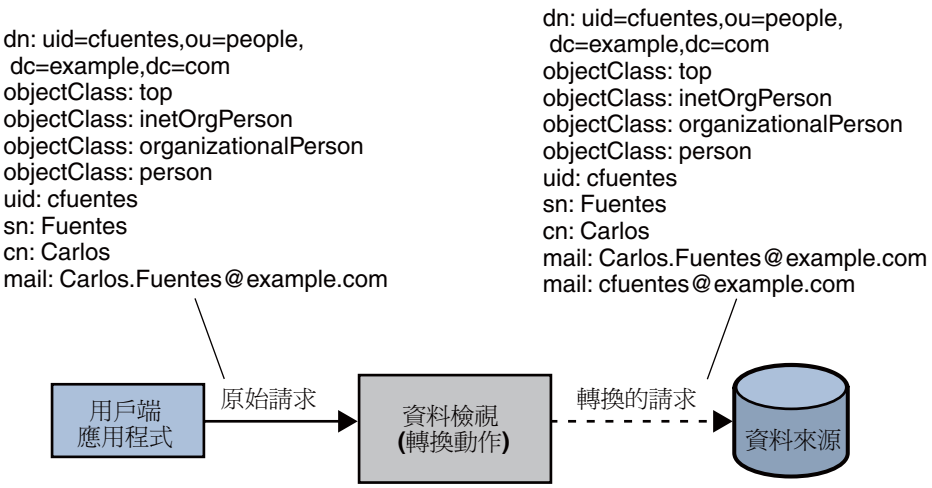


將第二個值儲存至由其他實體屬性指定的屬性

使用下列轉換規則儲存屬性值，以及增加新項目時所提供的值。在此情況下，當您增加項目時，即會儲存 `mail` 屬性的其他值。此轉換僅會在建立新項目時套用。

```
$ dpconf add-virtual-transformation dataview1 write add-attr-value \
mail \${uid}@example.com
```

下圖顯示增加請求時所發生的轉換。

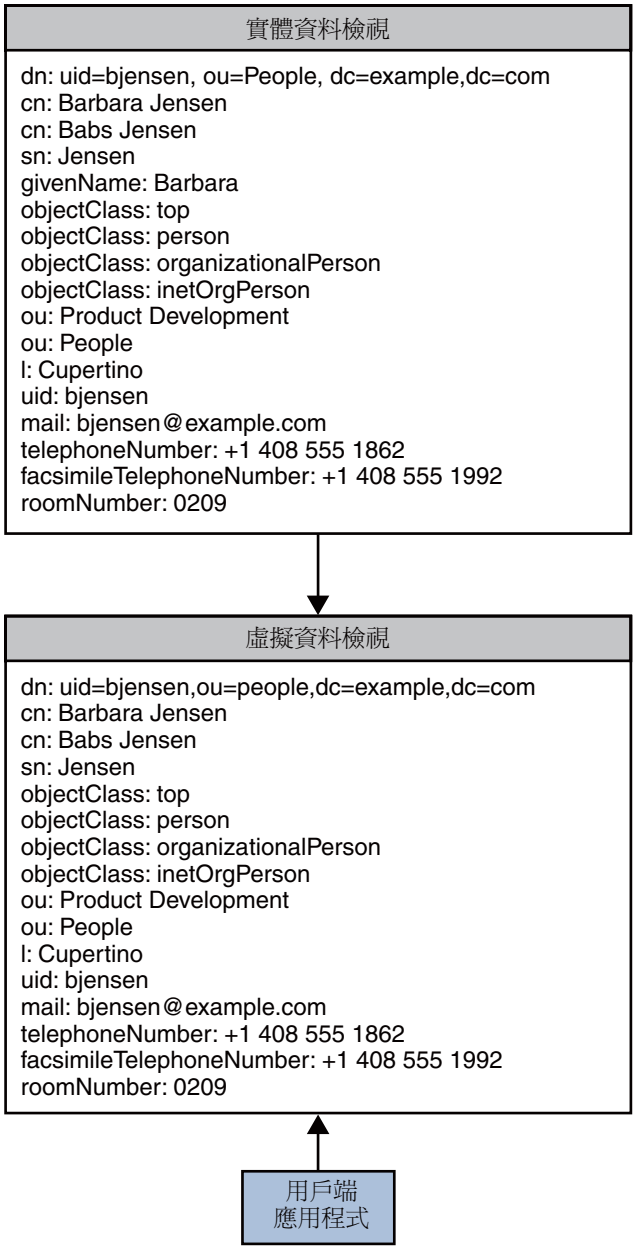


從輸出移除屬性

若不想在輸出中顯示屬性，請使用下列轉換規則。例如，套用下列轉換規則時，輸出中不會傳回 `givenName`。

```
dpconf add-virtual-transformation dataview1 read remove-attr givenName
```

下圖表示當搜尋傳回使用者項目時，使用者項目所發生的轉換。

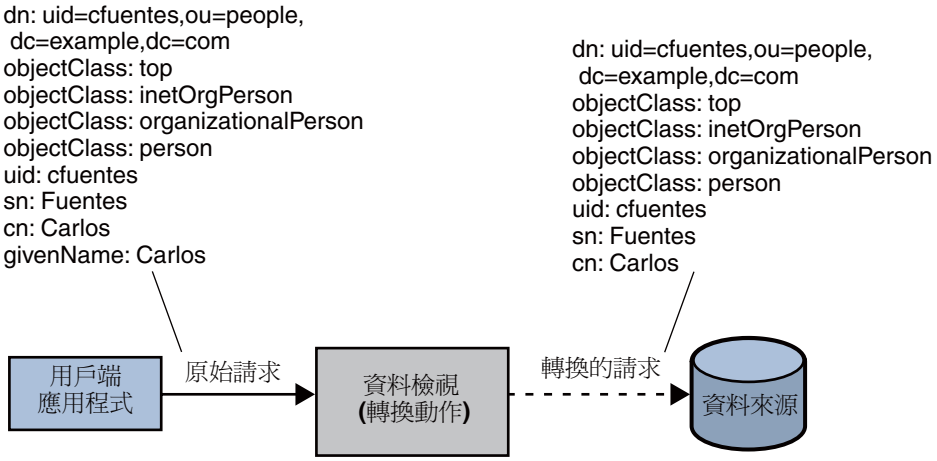


儲存項目時遮罩屬性

若不想儲存特定屬性，請使用下列轉換規則。例如，套用下列轉換規則時，`givenName` 屬性不會儲存於實體資料庫中。此轉換僅會在建立新項目時套用。

```
$ dpconf add-virtual-transformation dataview1 write remove-attr givenName
```

下圖顯示增加請求時所發生的轉換。

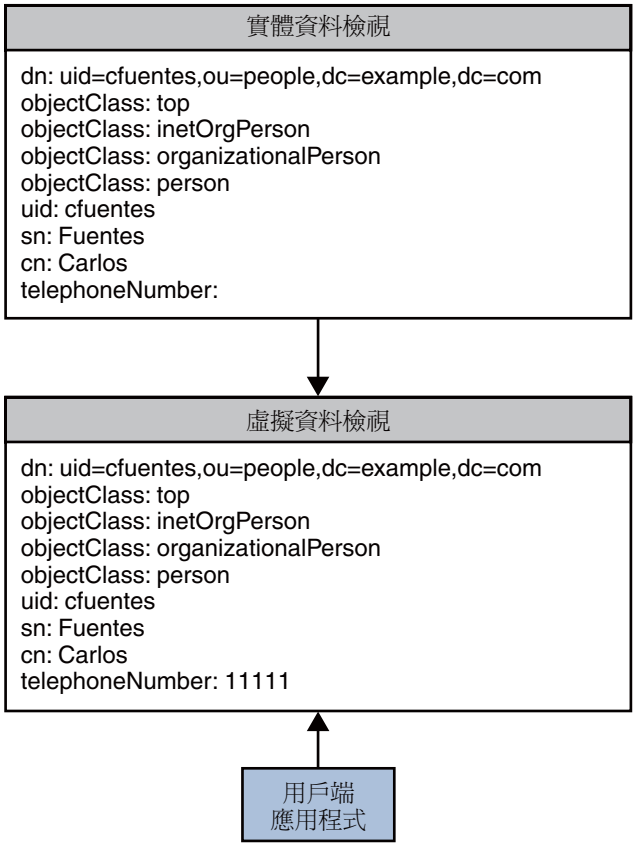


顯示屬性的預設值

若不想顯示指定給屬性的預設值，請使用下列轉換。例如，套用下列轉換時，本身不包含電話號碼的項目中會顯示預設電話號碼。

```
$ dpconf add-virtual-transformation data-view read 11111 telephoneNumber default-number
```

下圖表示當搜尋傳回使用者項目時，使用者項目所發生的轉換。

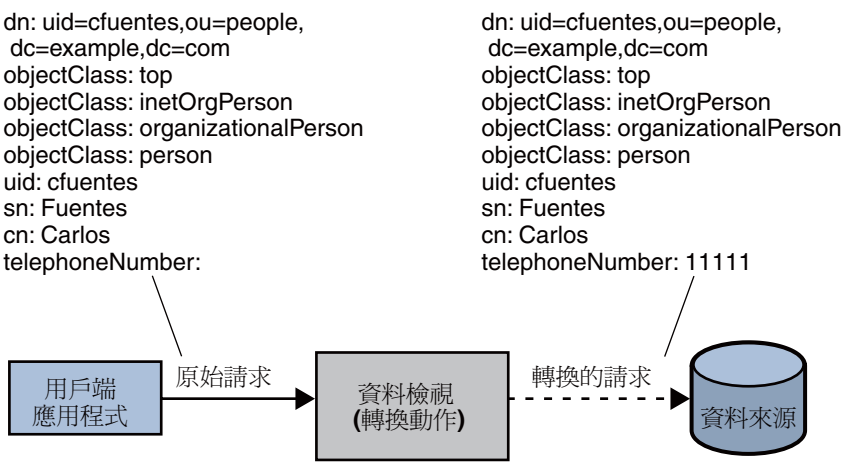


將預設值儲存至屬性

僅在建立項目期間未指定屬性值時，才會儲存預設值。若不想以預設值儲存屬性，請使用下列轉換規則。例如，套用下列轉換時，會隨您建立的每個項目各增加一個預設電話號碼。此轉換僅會在增加項目時套用。

```
$ dpconf add-virtual-transformation dataview1 write 11111 \
telephoneNumber telephone-number
```

下圖顯示增加請求時發生的轉換。


dn: uid=cfuentes,ou=people,
dc=example,dc=com
objectClass: top
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
uid: cfuentes
sn: Fuentes
cn: Carlos
telephoneNumber:
dn: uid=cfuentes,ou=people,
dc=example,dc=com
objectClass: top
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
uid: cfuentes
sn: Fuentes
cn: Carlos
telephoneNumber: 11111
用戶端
應用程式
原始請求
資料檢視
(轉換動作)
轉換的請求
資料來源

◆ ◆ ◆ 第 25 章

25 目錄代理伺服器與後端 LDAP 伺服器之間的連線

本章說明如何配置目錄代理伺服器與後端 LDAP 伺服器之間的連線。本章包含下列主題：

- 第 417 頁的 「配置目錄代理伺服器與後端 LDAP 伺服器之間的連線」
- 第 419 頁的 「配置目錄代理伺服器與後端 LDAP 伺服器之間的 SSL」
- 第 420 頁的 「選擇目錄代理伺服器的 SSL 密碼與協定」
- 第 421 頁的 「將請求轉寄至後端 LDAP 伺服器」

配置目錄代理伺服器與後端 LDAP 伺服器之間的連線

建立 LDAP 資料來源時，預設爲 LDAP 資料來源開啓的連線數爲六個，亦即每個讀取、連結及寫入作業各有兩個連線。若要驗證預設連線數，請輸入下列指令：

```
dpconf get-ldap-data-source-prop src-name num-read-init num-write-init num-bind-init
num-bind-init   :  2
num-read-init   :  2
num-write-init  :  2
```

當流量增加時，連線數會自動增加。

如需有關如何配置目錄代理伺服器與後端 LDAP 伺服器之間連線的資訊，請參閱下列程序：

▼ 配置目錄代理伺服器與後端 LDAP 伺服器之間的連線數

備註 – 本程序配置連結作業的連線數。若要配置讀取或寫入作業的連線數，請執行相同的程序，但以 `read` 或 `write` 取代 `bind`。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **配置目錄代理伺服器與後端 LDAP 伺服器之間連結作業的初始連線數。**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    num-bind-init:new-value
   ```

2. **配置連結作業的連線增量。**

   增量是每次請求超過目前連線數時所增加的連線數。

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    num-bind-incr:new-value
   ```

3. **配置連結作業的最大連線數。**

   到達最大連線數之後，即無法增加更多連線。

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    num-bind-limit:new-value
   ```

▼ 配置連線逾時

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **配置目錄代理伺服器嘗試連線至資料來源的最長時間。**

  ```
  $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
   connect-timeout:new-value
  ```

  例如，將連線逾時配置爲 10 毫秒。

  ```
  $ dpconf set-ldap-data-source-prop -h host1 -p 1389 data-source-name connect-timeout:10
  ```

▼ 配置連線池等待逾時

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **配置目錄代理伺服器等待連線池中建立的連線變為可用的最長時間。**

```
$ dpconf set-server-prop -h host -p port data-source-name \
 connection-pool-wait-timeout:value
```

例如，將逾時配置爲 20 秒。

```
$ dpconf set-ldap-data-source-prop -h host1 -p 1389 data-source-name \
 connection-pool-wait-timeout:20000
```

配置目錄代理伺服器與後端 LDAP 伺服器之間的 SSL

下列程序說明如何配置目錄代理伺服器與後端 LDAP 伺服器之間的 SSL。

▼ 配置目錄代理伺服器與後端 LDAP 伺服器之間的 SSL

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **配置目錄代理伺服器與後端 LDAP 伺服器之間的安全連接埠。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 ldaps-port:port-number
```

2. **配置目錄代理伺服器與後端 LDAP 伺服器之間的連線使用 SSL 的時機。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name ssl-policy:value
```

   - 如果 *value* 爲 `always`，則 SSL 始終可供連線使用。
   - 如果 *value* 爲 `client`，則在用戶端使用 SSL 時使用 SSL。

   如果連線未使用 SSL，可以透過使用 `startTLS` 指令將連線升級至 SSL。

   傳輸層安全性 (TLS) 是 SSL 的標準版本。透過 LDAP 的 TLS 是 IETF 已核准之保護 LDAP 安全標準方式。LDAPS 雖然實際上是標準，卻導致一些複雜度，例如服務使用兩個連接埠，而非只有一個。

3. **如第 420 頁的 「選擇目錄代理伺服器的 SSL 密碼與協定」中所述，選擇 SSL 的協定與密碼。**

4. **將目錄代理伺服器配置為驗證來自後端 LDAP 伺服器的 SSL 伺服器憑證。**
   如需有關資訊，請參閱第 346 頁的「將憑證從後端目錄伺服器增加至目錄代理伺服器憑證資料庫」。

5. **如果後端 LDAP 伺服器請求來自目錄代理伺服器的憑證，請配置目錄代理伺服器以傳送 SSL 用戶端憑證。**
   如需有關資訊，請參閱第 347 頁的「匯出憑證至後端 LDAP 伺服器」。

6. **重新啓動目錄代理伺服器實例，變更方可生效。**
   如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的「重新啓動目錄代理伺服器」。

選擇目錄代理伺服器的 SSL 密碼與協定

目錄代理伺服器可使用的密碼與協定取決於所使用的 Java™ 虛擬機器 (JVM™)。依預設，目錄代理伺服器使用爲 JVM 機器啓用的預設密碼與協定。

▼ 選擇密碼與協定清單

使用此程序擷取支援的密碼與協定，以及啓用的密碼與協定。如果是支援的密碼或協定，您可加以啓用或停用。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1. **檢視支援的密碼與協定清單。**

```
$ dpconf get-server-prop -h host -p port supported-ssl-cipher-suites \
 supported-ssl-protocols
```

2. **檢視啓用的密碼與協定清單。**

```
$ dpconf get-server-prop -h host -p port enabled-ssl-cipher-suites \
 enabled-ssl-protocols
```

3. **啓用一或多個支援的密碼或協定。**

   a. **啓用一或多個支援的密碼。**

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-suites:supported-ssl-cipher-suite \
 [enabled-ssl-cipher-suites:supported-ssl-cipher-suite ...]
```

若要將密碼增加至現有支援的密碼清單，請使用此指令：

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-suites+:supported-ssl-cipher-suite
```

b. 啓用一或多個支援的協定。

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-protocols:supported-ssl-cipher-protocol \
 [enabled-ssl-cipher-protocols:supported-ssl-cipher-protocol ...]
```

若要將協定增加至現有支援的協定清單，請使用此指令：

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-protocols+:supported-ssl-cipher-protocol
```

4 (可選擇) 停用支援的密碼或協定。

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-protocols-:supported-ssl-cipher-protocol
```

將請求轉寄至後端 LDAP 伺服器

本節包含各種方法的相關資訊，您可用來將目錄代理伺服器的請求轉寄至後端 LDAP 伺服器。

利用重新執行連結轉寄請求

如需有關目錄代理伺服器中用戶端憑證之連結轉寄的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Directory Proxy Server Configured for BIND Replay」。下列程序說明如何透過使用連結重新執行將請求從目錄代理伺服器轉寄至後端 LDAP 伺服器。

▼ 利用重新執行連結轉寄請求

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **配置資料來源用戶端憑證，以透過使用由用戶端提供的憑證認證後端 LDAP 伺服器。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-client-identity
```

利用代理授權轉寄請求

如需有關目錄代理伺服器中代理授權的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Directory Proxy Server Configured for Proxy Authorization」。

本節包含透過使用代理授權與代理授權控制轉寄請求的程序。

▼ 透過使用代理授權轉寄請求

1 **將資料來源配置為預期代理授權控制版本為 1 或 2。**

例如，將資料來源配置爲預期代理授權控制版本 1。

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 proxied-auth-use-v1:true
```

或者，將資料來源配置爲預期代理授權控制版本 2。

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 proxied-auth-use-v1:false
```

2 **將資料來源配置為透過使用代理授權認證後端 LDAP 伺服器。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-proxy-auth
```

若要將資料來源配置爲透過僅使用寫入作業的代理授權認證後端 LDAP 伺服器，請執行此指令：

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-proxy-auth-for-write
```

僅使用代理授權控制執行寫入作業時，用戶端身份識別不會轉寄至 LDAP 伺服器進行讀取請求。如需有關轉寄不含用戶端身份識別的請求的詳細資訊，請參閱第 423 頁的「轉寄不含用戶端身份識別的請求」。

3 **利用目錄代理伺服器的連結憑證配置資料來源。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 bind-dn:DPS-bind-dn bind-pwd-file:filename
```

4 **使用逾時配置資料來源。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 proxied-auth-check-timeout:value
```

目錄代理伺服器透過使用 `getEffectiveRights` 指令驗證用戶端 DN 是否具有代理授權的相關 ACI。在目錄代理伺服器中快取結果，並於 `proxied-auth-check-timeout` 過期時進行更新。

5. **請視需要重新啓動目錄代理伺服器實例以使變更生效。**

   如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的 「重新啓動目錄代理伺服器」。

▼ 當請求包含代理授權控制時，透過使用代理授權轉寄請求

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **將目錄代理伺服器配置為接受代理授權控制版本 1 和/或版本 2。**

```
$ dpconf set-server-prop -h host -p port allowed-ldap-controls:proxy-auth-v1 \
 allowed-ldap-controls:proxy-auth-v2
```

轉寄不含用戶端身份識別的請求

下列程序說明如何將請求從目錄代理伺服器轉寄至後端 LDAP 伺服器，而不轉寄用戶端身份識別。

▼ 轉寄不含用戶端身份識別的請求

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **透過使用目錄代理伺服器的憑證，將資料來源配置為驗證後端 LDAP 伺服器。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-specific-identity
```

2. **利用目錄代理伺服器的連結憑證配置資料來源。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 bind-dn:bind-dn-of-DPS bind-pwd-file:filename
```

3. **請視需要重新啓動目錄代理伺服器實例以使變更生效。**

   如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的 「重新啓動目錄代理伺服器」。

以替代使用者身份轉寄請求

本節包含有關如何以替代使用者身份轉寄請求的資訊。

▼ 配置遠端使用者對映

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **啓用要以替代使用者身份轉寄的作業。**

```
$ dpconf set-server-prop -h host -p port enable-user-mapping:true
```

2 **指定包含遠端對映 ID 的屬性的名稱。**

```
$ dpconf set-server-prop -h host -p port \
 remote-user-mapping-bind-dn-attr:attribute-name
```

3 **使目錄代理伺服器能夠遠端對映用戶端 ID。**

```
$ dpconf set-server-prop -h host -p port enable-remote-user-mapping:true
```

4 **配置預設對映。**

```
$ dpconf set-server-prop -h host -p port \
 user-mapping-default-bind-dn:default-mapping-bind-dn \
 user-mapping-default-bind-pwd-file:filename
```

如果遠端 LDAP 伺服器上找不到對映的身份識別，會將用戶端身份識別對映至預設的身份識別。

5 **為遠端 LDAP 伺服器上的用戶端在項目中配置使用者對映。**

如需有關在目錄伺服器中配置使用者對映的資訊，請參閱第 150 頁的 「代理授權」。

▼ 配置本機使用者對映

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **啓用要以替代使用者身份轉寄的作業。**

```
$ dpconf set-server-prop -h host -p port enable-user-mapping:true
```

2 **確保未將目錄代理伺服器配置為從遠端對映用戶端 ID。**

```
$ dpconf set-server-prop -h host -p port enable-remote-user-mapping:false
```

3 **配置預設對映。**

```
$ dpconf set-server-prop -h host -p port \
 user-mapping-default-bind-dn:default-mapping-bind-dn \
 user-mapping-default-bind-pwd-file:filename
```

如果遠端 LDAP 伺服器的對映失敗，用戶端 ID 會對映至此 DN。

4. **如果允許未認證的使用者執行作業，請配置未認證之用戶端的對映。**

```
$ dpconf set-server-prop -h host -p port \
 user-mapping-anonymous-bind-dn:anonymous-mapping-bind-dn \
 user-mapping-anonymous-bind-pwd-file:filename
```

如需有關如何允許未認證的使用者執行作業的資訊，請參閱第 441 頁的 「配置匿名存取」。

5. **配置用戶端的 ID。**

```
$ dpconf set-user-mapping-prop -h host -p port \
 user-bind-dn:client-bind-dn user-bind-pwd-file:filename
```

6. **配置替代使用者的 ID。**

```
$ dpconf set-user-mapping-prop -h host -p port \
 mapped-bind-dn:alt-user-bind-dn mapped-bind-pwd-file:filename
```

▼ 配置匿名用戶端的使用者對映

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **配置未認證的用戶端的對映。**

```
$ dpconf set-server-prop -h host -p port \
 user-mapping-anonymous-bind-dn:anonymous-mapping-bind-dn \
 user-mapping-anonymous-bind-pwd-file:filename
```

由於遠端 LDAP 伺服器不包含匿名用戶端的項目，因此在目錄代理伺服器中配置匿名用戶端的對映。

如需有關允許未認證的使用者執行作業的資訊，請參閱第 441 頁的 「配置匿名存取」。

◆ ◆ ◆ 第 26 章

用戶端與目錄代理伺服器之間的連線

如需用戶端與目錄代理伺服器之間的連線和連線處理程式之簡介，以及連線處理程式中使用之條件與策略的說明，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 20 章「Connections Between Clients and Directory Proxy Server」。

本章包含下列主題：

- 第 427 頁的「建立、配置與刪除連線處理程式」
- 第 431 頁的「建立與配置請求篩選策略與搜尋資料隱藏規則」
- 第 434 頁的「建立與配置資源限制策略」
- 第 436 頁的「配置目錄代理伺服器爲以連線爲基礎的路由器」

建立、配置與刪除連線處理程式

如需有關如何建立、配置與刪除連線處理程式的資訊，以及有關如何配置資料檢視相似性的資訊，請參閱下列程序。

▼ 建立連線處理程式

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1 **建立連線處理程式。**

```
$ dpconf create-connection-handler -h host -p port connection-handler-name
```

2 **(可選擇) 檢視連線處理程式的清單。**

```
$ dpconf list-connection-handlers -h host -p port
```

▼ 配置連線處理程式

開始之前 連線處理程式的特性必須依據針對目錄代理伺服器實例所定義的其他連線處理程式特性進行定義。請考量所有連線處理程式的特性，以確保這些連線處理程式指定不同的條件集，同時也正確地設定優先權。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1 檢視連線處理程式的詳細清單，以查看這些連線處理程式的主要特性與相關特性。

```
$ dpconf list-connection-handlers -h host -p port -v
Name                        is-enabled  priority  description
--------------------------  ----------  --------  --------------------------
anonymous                   false       99        unauthenticated connections
default connection handler  true        100       default connection handler
```

建立目錄代理伺服器實例時，即會建立連線處理程式 `anonymous` 與 `default connection handler`。

2 檢視單一連線處理程式的所有特性。

```
$ dpconf get-connection-handler-prop -h host -p port connection-handler-name
```

新連線處理程式的預設特性如下：

```
aci-source                    :  -
allowed-auth-methods          :  anonymous
allowed-auth-methods          :  sasl
allowed-auth-methods          :  simple
allowed-ldap-ports            :  ldap
allowed-ldap-ports            :  ldaps
bind-dn-filters               :  any
data-view-routing-custom-list :  -
data-view-routing-policy      :  all-routable
description                   :  -
domain-name-filters           :  any
enable-data-view-affinity     :  false
ip-address-filters            :  any
is-enabled                    :  false
is-ssl-mandatory              :  false
priority                      :  99
request-filtering-policy      :  no-filtering
resource-limits-policy        :  no-limits
schema-check-enabled          :  false
user-filter                   :  any
```

3 配置連線處理程式的優先權。

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name priority:value
```

優先權可以是從 1 到 100 的任何數字，其中 1 具有最高的優先權。對於目錄代理伺服器實例，會以優先權順序評估連線處理程式。

4 **(可選擇) 指定連線處理程式的 DN 篩選特性。**

此特性可讓您根據部分或全部的連結 DN 以控制存取。此特性的值是常規表示式。

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 bind-dn-filters:regular-expression
```

連結 DN 篩選器採用 Java™ 常規表示式的格式。如需有關建立 Java 常規表示式的資訊，請參閱 `http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html`。

例如，若要從 `ou=people,dc=example,dc=com` 下的使用者傳送所有連結至名爲 `secure-handler` 的連線處理程式，請依下列方式設定 `bind-dn-filters` 特性：

```
$ dpconf set-connection-handler-prop -h host1 -p 1389 secure-handler \
 bind-dn-filters:"uid=.*,ou=people,dc=example,dc=com"
```

5 **(可選擇) 指定搭配此連線處理程式使用的請求篩選策略名稱。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 request-filtering-policy:policy-name
```

其中 *policy-name* 是現有請求篩選策略的名稱。如需有關如何建立與配置請求篩選策略的資訊，請參閱第 431 頁的 「建立與配置請求篩選策略與搜尋資料隱藏規則」。

6 **(可選擇) 指定搭配此連線處理程式使用的資源限制策略名稱。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 resource-limits-policy:policy-name
```

其中 *policy-name* 是現有資源限制策略的名稱。如需有關如何建立與配置資源限制策略的資訊，請參閱第 434 頁的 「建立與配置資源限制策略」。

7 **配置步驟 2 中列出的所有其他特性。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 property:value [property:value ...]
```

例如，將連線處理程式配置爲僅接受 SSL 連線。

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 is-ssl-mandatory:true
```

如需特性的說明及其有效值清單，請執行此指令：

```
$ dpconf help-properties connection-handler
```

8 **啓用連線處理程式。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name is-enabled:true
```

9 **請視需要重新啓動目錄代理伺服器實例以使變更生效。**

如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的 「重新啓動目錄代理伺服器」。

▼ 刪除連線處理程式

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **(選用) 檢視連線處理程式的清單。**

```
$ dpconf list-connection-handlers -h host -p port
```

2 **刪除一或多個連線處理程式。**

```
$ dpconf delete-connection-handler -h host -p port connection-handler-name [connection-handler-name ... ]
```

▼ 配置資料檢視的相似性

配置連線給連線處理程式時，該連線上的請求會顯示在該連線處理程式所配置的資料檢視清單中，或顯示在所有已配置的資料檢視中。該連線上後續的請求僅會顯示在第一次請求所使用的資料檢視中。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **啓用資料檢視的相似性。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 enable-data-view-affinity:true
```

2 **(可選擇) 配置連線處理程式，以將請求路由至自訂的資料檢視清單。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name data-view-routing-policy:custom
```

3 **(可選擇) 配置資料檢視清單。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 data-view-routing-custom-list:view-name [data-view-routing-custom-list:view-name ...]
```

若要將資料檢視增加至現有的資料檢視清單中，請使用此指令：

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 data-view-routing-custom-list+:view-name
```

若要將資料檢視從現有的資料檢視清單中移除，請使用此指令：

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 data-view-routing-custom-list-:view-name
```

建立與配置請求篩選策略與搜尋資料隱藏規則

如需請求篩選策略的簡介，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Request Filtering Policies for Connection Handlers」。如需搜尋資料隱藏規則的簡介，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Search Data Hiding Rules in the Request Filtering Policy」。

如需有關如何建立與配置請求篩選策略和搜尋資料隱藏規則的資訊，請參閱下列程序。

▼ 建立請求篩選策略

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **建立請求篩選策略。**

```
$ dpconf create-request-filtering-policy policy-name
```

2 **建立請求篩選策略與連線處理程式的關聯。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 request-filtering-policy:policy-name
```

▼ 配置請求篩選策略

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **檢視請求篩選策略的特性。**

```
$ dpconf get-request-filtering-policy-prop -h host -p port policy-name
```

請求篩選策略的預設特性如下：

```
allow-add-operations                :  true
allow-bind-operations               :  true
allow-compare-operations            :  true
allow-delete-operations             :  true
allow-extended-operations           :  true
allow-inequality-search-operations  :  true
allow-modify-operations             :  true
allow-rename-operations             :  true
allow-search-operations             :  true
allowed-comparable-attrs            :  all
allowed-search-scopes               :  base
```

```
allowed-search-scopes               :  one-level
allowed-search-scopes               :  subtree
allowed-subtrees                    :  ""
description                         :  -
prohibited-comparable-attrs         :  none
prohibited-subtrees                 :  none
```

2 **透過設定步驟 1 中所列的一或多個特性，配置請求篩選策略。**

```
$ dpconf set-request-filtering-policy-prop -h host -p port policy-name \
  property:value [property:value ...]
```

透過設定步驟 1 中所列的特性，可以配置請求篩選策略的下列功能：

- 允許用戶端執行的作業類型
- 顯示給用戶端或向用戶端隱藏的子樹狀結構
- 搜尋作業的範圍
- 搜尋篩選的類型
- 搜尋和比較作業中可以比較或無法比較的屬性類型

▼ 建立搜尋資料隱藏規則

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **建立請求篩選策略的一或多個搜尋資料隱藏規則。**

```
$ dpconf create-search-data-hiding-rule -h host -p port policy-name rule-name \
 [rule-name ...]
```

2 **檢視搜尋資料隱藏規則的特性。**

```
$ dpconf get-search-data-hiding-rule-prop policy-name rule-name
```

搜尋資料隱藏規則的預設特性如下：

```
attrs                               :  -
rule-action                         :  hide-entry
target-attr-value-assertions        :  -
target-dn-regular-expressions       :  -
target-dns                          :  -
```

3 **透過設定步驟 2 中所列的一或多個特性，配置搜尋資料隱藏規則。**

```
$ dpconf set-search-data-hiding-rule-prop -h host -p port policy-name rule-name \
  property:value [property:value ...]
```

可以使用下列規則動作其中之一：

`hide-entry` 不傳回目標項目。

`hide-attributes` 傳回目標項目，但剔除指定的屬性。

`show-attributes` 傳回目標項目，但剔除未指定的屬性。

規則可套用至下列項目：

`target-dns` 具有指定 DN 的項目

`target-dn-regular-expressions` 具有指定 DN 模式的項目

`target-attr-value-assertions` 具有指定屬性名稱與成對屬性值組 (*attrName#attrValue*) 的項目

下列配置定義了隱藏 `inetorgperson` 類型項目的搜尋資料隱藏規則。

```
$ dpconf set-search-data-hiding-rule-prop -h host1 -p port my-policy my-rule \
  target-attr-value-assertions:objectclass#inetorgperson
```

請求篩選策略與搜尋資料隱藏規則範例

下列範例包含請求篩選策略與搜尋資料隱藏規則。將請求篩選策略與搜尋資料隱藏規則結合時，存取資料的限制如下：

- 不允許下列作業類型：增加、刪除、延伸、修改與重新命名。
- 僅能存取 `ou=people,dc=sun,dc=com` 子樹狀結構。
- 搜尋作業傳回 `inetorgperson` 類型以外的項目。

範例 26–1 請求篩選策略範例

```
allow-add-operations                :  false
allow-bind-operations               :  true
allow-compare-operations            :  true
allow-delete-operations             :  false
allow-extended-operations           :  false
allow-inequality-search-operations  :  true
allow-modify-operations             :  false
allow-rename-operations             :  false
allow-search-operations             :  true
allowed-comparable-attrs            :  all
allowed-search-scopes               :  base
allowed-search-scopes               :  one-level
allowed-search-scopes               :  subtree
allowed-subtrees                    :  ou=people,dc=sun,dc=com
description                         :  myRequestFilteringPolicy
prohibited-comparable-attrs         :  none
prohibited-subtrees                 :  none
```

範例 26–2 搜尋資料隱藏規則範例

```
attrs                              :  -
rule-action                        :  hide-entry
target-attr-value-assertions       :  objectclass:inetorgperson
target-dn-regular-expressions      :  -
target-dns                         :  -
```

建立與配置資源限制策略

如需資源限制策略的簡介，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Resource Limits Policies for Connection Handlers」。如需有關如何建立與配置資源限制策略以及如何自訂搜尋限制的資訊，請參閱下列程序。

▼ 建立資源限制策略

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **建立資源限制策略。**

```
$ dpconf create-resource-limits-policy -h host -p port policy-name
```

如需有關如何修改資源限制策略特性的資訊，請參閱第 434 頁的 「配置資源限制策略」。

2 **建立資源限制策略與連線處理程式的關聯。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 resource-limits-policy:policy-name
```

▼ 配置資源限制策略

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **檢視資源限制策略的特性。**

```
$ dpconf get-resource-limits-policy-prop -h host -p port policy-name
```

資源限制策略的預設特性如下：

```
description                              :  -
max-client-connections                   :  unlimited
```

```
max-connections                              :  unlimited
max-simultaneous-operations-per-connection   :  unlimited
max-total-operations-per-connection          :  unlimited
minimum-search-filter-substring-length       :  unlimited
referral-bind-policy                         :  default
referral-hop-limit                           :  default
referral-policy                              :  default
search-size-limit                            :  unlimited
search-time-limit                            :  unlimited
```

2 **透過設定步驟 1 中所列的一或多個特性，配置資源限制策略：**

```
$ dpconf set-resource-limits-policy-prop -h host -p port policy-name \
  property:value [property:value ...]
```

▼ 自訂搜尋限制

您可以根據搜尋基底和搜尋範圍，爲搜尋作業定義自訂的限制。如果搜尋作業的目標 DN 與範圍符合指定的條件，則限制搜尋結果的大小上限。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **建立一或多個自訂搜尋限制。**

```
$ dpconf create-custom-search-size-limit -h host -p port policy-name \
  custom-search-limit-name [custom-search-limit-name ...]
```

2 **設定自訂搜尋限制的條件。**

```
$ dpconf set-custom-search-size-limit-prop -h host -p port policy-name \
  custom-search-limit-name one-level-search-base-dn:value subtree-search-base-dn:value
```

3 **設定搜尋滿足步驟 2 的其中一個條件時，所傳回結果數的限制。**

```
$ dpconf set-custom-search-size-limit-prop -h host -p port policy-name \
 custom-search-limit-name search-size-limit:value
```

4 **檢視自訂搜尋限制的特性。**

```
$ dpconf get-custom-search-size-limit-prop -h host -p port policy-name \
  custom-search-limit-name
```

自訂搜尋限制的預設特性如下：

```
one-level-search-base-dn  :  -
search-size-limit         :  unlimited
subtree-search-base-dn    :  -
```

配置目錄代理伺服器為以連線為基礎的路由器

Directory Proxy Server 5.2 是以連線爲基礎的路由器。在 Directory Proxy Server 5.2 中，用戶端連線會路由至指定的目錄伺服器。該用戶端連線上的所有請求都會傳送至相同的目錄伺服器，直到連線中斷或用戶端解除連結爲止。

Directory Proxy Server 6.3 是以作業爲基礎的路由器。但是，此版本的目錄代理伺服器可能因爲相容性而設爲以連線爲基礎的路由器，如下列程序所述。

▼ 配置目錄代理伺服器為以連線為基礎的路由器

1. **如第 427 頁的 「建立、配置與刪除連線處理程式」中所述，建立與配置一或多個連線處理程式。**

   您也可以使用預設的連線處理程式。

2. **配置所有連線處理程式將請求僅路由至根資料檢視。**

   例如：

   ```
   $ dpconf set-connection-handler-prop -h host1 -p 1389 myConnectionHandler \
    data-view-routing-policy:custom data-view-routing-custom-list:"root data view"
   ```

3. **如第 321 頁的 「建立與配置 LDAP 資料來源」中所述，建立與配置每個後端 LDAP 伺服器的資料來源。**

   例如：

   ```
   $ dpconf create-ldap-data-source -h host1 -p 1389 myDataSource host2:2389
   ```

4. **如第 324 頁的 「建立與配置 LDAP 資料來源池」中所述，建立與配置資料來源池。**

   例如：

   ```
   $ dpconf create-ldap-data-source-pool -h host1 -p 1389 myDataSourcePool
   ```

5. **如第 325 頁的 「將 LDAP 資料來源附加至資料來源池」中所述，將所有資料來源附加至資料來源池。**

   例如，

   ```
   $ dpconf attach-ldap-data-source -h host1 -p 1389 myDataSourcePool myDataSource
   ```

6. **如第 421 頁的 「利用重新執行連結轉寄請求」中所述，配置每個資料來源透過使用 BIND 重新執行來認證用戶端。**

   例如：

   ```
   $ dpconf set-ldap-data-source-prop -h host1 -p 1389 myDataSource \
    client-cred-mode:use-client-identity
   ```

7 如第 360 頁的 「配置用戶端相似性」中所述，配置用戶端連線與資料來源池之間的相似性。

例如：

```
$ dpconf set-ldap-data-source-pool-prop -h host1 -p 1389 myDataSourcePool \
 enable-client-affinity:true client-affinity-policy:read-write-affinity-after-write
```

◆ ◆ ◆ 第 27 章

27 目錄代理伺服器用戶端認證

如需目錄代理伺服器中用戶端認證的簡介，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 21 章「Directory Proxy Server Client Authentication」。

本章包含下列主題：

- 第 439 頁的 「配置用戶端與目錄代理伺服器之間的偵聽程式」
- 第 440 頁的 「認證目錄代理伺服器的用戶端」

配置用戶端與目錄代理伺服器之間的偵聽程式

目錄代理伺服器提供安全偵聽程式與非安全偵聽程式以與用戶端通訊。如需有關目錄代理伺服器偵聽程式的相關資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Directory Proxy Server Client Listeners」。本節說明如何配置偵聽程式。

▼ 配置用戶端與目錄代理伺服器之間的偵聽程式

備註 – 本程序配置用戶端與目錄代理伺服器之間的非安全偵聽程式。若要配置安全偵聽程式，請執行相同的程序，但以 `ldaps` 取代 `ldap`。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。在 DSCC 中，您可以在 [效能] 標籤上配置此特性。

1 **檢視非安全偵聽程式的特性。**

```
$ dpconf get-ldap-listener-prop -h host -p port
```

非安全偵聽程式的預設特性如下：

```
connection-idle-timeout          :  1h
connection-read-data-timeout     :  2s
connection-write-data-timeout    :  1h
is-enabled                       :  true
listen-address                   :  0.0.0.0
listen-port                      :  port-number
max-connection-queue-size        :  128
max-ldap-message-size            :  unlimited
number-of-threads                :  2
use-tcp-no-delay                 :  true
```

2 **根據需求變更步驟 1 中所列的一或多個特性。**

```
$ dpconf set-ldap-listener-prop -h host -p port property:new-value
```

例如，若要停用 host1 上執行的目錄代理伺服器實例之非安全連接埠，請執行下列指令：

```
$ dpconf set-ldap-listener-prop -h host1 -p 1389 is-enabled:false
```

注意 – 如果您計劃使用未經授權的連接埠號碼，則必須以超級使用者的身份執行目錄代理伺服器。

若要變更非安全的連接埠號碼，請執行下列指令：

```
$ dpconf set-ldap-listener-prop -h host -p port listen-port:new-port-number
```

3 **請視需要重新啓動目錄代理伺服器實例以使變更生效。**

對某些偵聽程式特性所進行的變更，需要重新啓動伺服器才會生效。如果必須重新啓動伺服器，`dpconf` 會提供警示。如需有關重新啓動目錄代理伺服器的資訊，請參閱第 313 頁的 「重新啓動目錄代理伺服器」。

認證目錄代理伺服器的用戶端

依預設，目錄代理伺服器配置爲使用簡單連結認證。簡單連結認證不需要其他配置。

如需有關用戶端與目錄代理伺服器之間的認證資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Client Authentication Overview」。如需有關如何配置認證的資訊，請參閱下列程序。

▼ 配置憑證型認證

如需有關憑證型用戶端認證的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Configuring Certificates in Directory Proxy Server」。本節說明如何配置憑證型認證。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

備註 – 憑證型認證僅能在 SSL 連線上執行。

- **將目錄代理伺服器配置為用戶端建立 SSL 連線時必須出示憑證。**

```
$ dpconf set-server-prop -h host -p port allow-cert-based-auth:require
```

▼ 配置匿名存取

如需有關匿名存取的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Anonymous Access」。如需有關如何將匿名用戶端的識別對映到其他識別的資訊，請參閱第 423 頁的 「以替代使用者身份轉寄請求」。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **允許未認證的使用者執行作業。**

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:true
```

▼ 為 SASL 外部連結配置目錄代理伺服器

如需有關 SASL 外部連結的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Using SASL External Bind」。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **不允許未認證的作業。**

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:false
```

2. **要求用戶端建立連線時必須出示憑證。**

```
$ dpconf set-server-prop -h host -p port allow-cert-based-auth:require
```

用戶端提供包含 DN 的憑證。

3 **啓用 SASL 外部連結的用戶端認證。**

```
$ dpconf set-server-prop -h host -p port allow-sasl-external-authentication:true
```

4 **配置目錄代理伺服器使用的識別，以對映至後端 LDAP 伺服器上的用戶端憑證。**

```
$ dpconf set-server-prop -h host -p port cert-search-bind-dn:bind-DN \
 cert-search-bind-pwd-file:filename
```

5 **配置目錄代理伺服器搜尋的子樹狀結構之基底 DN。**

目錄代理伺服器搜尋子樹狀結構，以尋找對映至用戶端憑證的使用者項目。

```
$ dpconf set-server-prop -h host -p port cert-search-base-dn:base-DN
```

6 **將用戶端憑證中的資訊對映至 LDAP 伺服器上的憑證。**

a. **在包含憑證的 LDAP 伺服器上命名屬性。**

```
$ dpconf set-server-prop cert-search-user-attribute:attribute
```

b. **將用戶端憑證的屬性對映至包含憑證的 LDAP 伺服器上項目的 DN。**

```
$ dpconf set-server-prop -h host -p port \
 cert-search-attr-mappings:client-side-attribute-name:server-side-attribute-name
```

例如，若要將 DN 爲 `cn=user1,o=sun,c=us` 的用戶端憑證對映至 DN 爲 `uid=user1,o=sun` 的 LDAP 項目，請執行下列指令：

```
$ dpconf set-server-prop -h host1 -p 1389 cert-search-attr-mappings:cn:uid \
 cert-search-attr-mappings:o:o
```

7 **(可選擇) 將 SASL 外部連結作業的請求路由至所有資料檢視或自訂資料檢視清單。**

- 若要將請求路由至所有資料檢視，請執行此指令：

```
$ dpconf set-server-prop -h host -p port cert-data-view-routing-policy:all-routable
```

- 若要將請求路由至資料檢視清單，請執行此指令：

```
$ dpconf set-server-prop -h host -p port cert-data-view-routing-policy:custom \
 cert-data-view-routing-custom-list:view-name [view-name...]
```

◆ ◆ ◆ 第 28 章

28 目錄代理伺服器記錄

目錄代理伺服器會在存取記錄與錯誤記錄中記錄資訊。與目錄伺服器不同，目錄代理伺服器不具有稽核記錄。如需目錄代理伺服器中記錄的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的第 23 章「Directory Proxy Server Logging」。

本章包含下列主題：

- 第 443 頁的「檢視目錄代理伺服器記錄」
- 第 444 頁的「配置目錄代理伺服器記錄」
- 第 446 頁的「配置目錄代理伺服器記錄自動重建」
- 第 449 頁的「刪除目錄代理伺服器記錄」
- 第 450 頁的「將警示記錄至 `syslogd` 常駐程式」
- 第 453 頁的「經由目錄代理伺服器與目錄伺服器存取記錄追蹤用戶端請求」

檢視目錄代理伺服器記錄

您可以直接透過記錄檔或使用目錄服務控制中心 (DSCC) 檢視目錄代理伺服器記錄。

依預設，記錄儲存在此目錄中：

instance-path`/logs`

下圖顯示 DSCC 上目錄代理伺服器錯誤記錄的螢幕擷取。

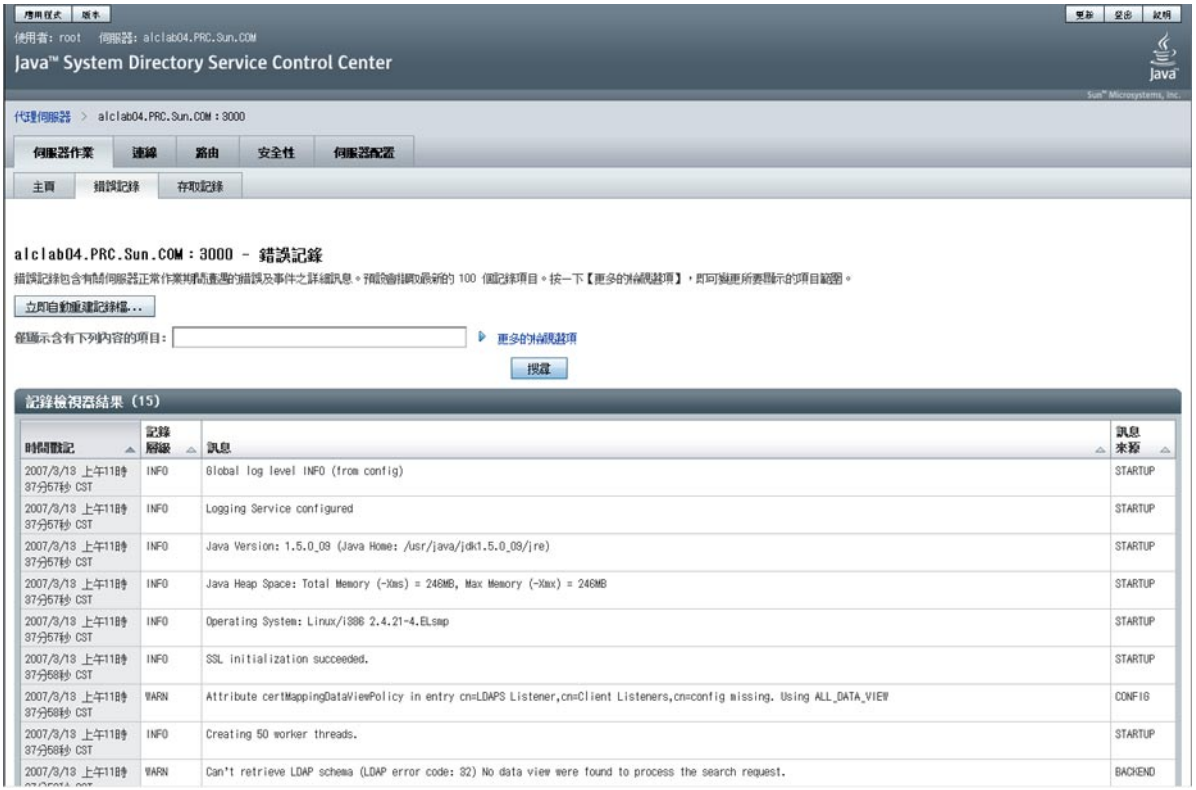


圖 28–1 目錄代理伺服器的錯誤記錄視窗

配置目錄代理伺服器記錄

透過使用 `dpconf` 指令或 DSCC，可以配置目錄代理伺服器錯誤記錄與存取記錄。如需有關如何使用 DSCC 配置記錄的資訊，請參閱目錄代理伺服器線上說明。本節說明如何透過使用 `dpconf` 指令配置目錄代理伺服器記錄。

您可以透過執行下列指令，擷取完整的配置選項清單以及允許的值和預設值：

```
$ dpconf help-properties error-log

$ dpconf help-properties access-log
```

▼ 配置目錄代理伺服器存取與錯誤記錄

此程序配置目錄代理伺服器存取記錄。若要配置目錄代理伺服器錯誤記錄，請執行相同的程序，但以 `error` 取代 `access`。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **檢視存取記錄的特性**。

```
$ dpconf get-access-log-prop -h host -p port
```

存取記錄的預設特性如下：

```
default-log-level                :  info
enable-log-rotation                 :  true
log-buffer-size                  :  9.8k
log-file-name                    :  logs/access
log-file-perm                    :  600
log-level-client-connections     :  -
log-level-client-disconnections  :  -
log-level-client-operations      :  -
log-level-connection-handlers    :  -
log-level-data-sources           :  -
log-level-data-sources-detailed  :  -
log-min-size                        :  100M
log-rotation-frequency           :  1h
log-rotation-policy              :  size
log-rotation-size                :  100M
log-rotation-start-day           :  1
log-rotation-start-time          :  0000
log-search-filters               :  false
max-age                          :  unlimited
max-log-files                    :  10
max-size                         :  unlimited
min-free-disk-space-size         :  1M
```

2 **變更步驟 1 中所列的一或多個特性**。

```
$ dpconf set-access-log-prop -h host -p port property:value \
 [property:value ...]
```

例如，若要將所有訊息種類的預設記錄層級設爲警告，請將 `default-log-level` 特性的值設爲 `warning`。

```
$ dpconf set-access-log-prop -h host1 -p 1389 default-log-level:warning
```

若要停用所有記錄，不論各訊息種類的記錄層級爲何，都將 `default-log-level` 特性的值設爲 `none`。

```
$ dpconf set-access-log-prop -h host1 -p 1389 default-log-level:none
```

若要將特定記錄層級重設爲預設記錄層級，請將該記錄層級的特性設爲 inherited。例如，若要重設用戶端連線的記錄層級，請執行下列指令：

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-level-client-connections:inherited
```

如需有關可由 set-access-log-prop 子指令設定之特性的資訊，請鍵入：

```
$ dpconf help-properties access-log
```

配置目錄代理伺服器記錄自動重建

依預設，記錄檔會在其大小達到 100 MB 時自動重建。依預設會保留十個記錄檔，超過十個之後，自動重建程序會從最舊的記錄檔開始覆寫。本節說明如何配置目錄代理伺服器記錄以排程自動重建、如何手動自動重建記錄，以及如何停用記錄自動重建。如需配置範例，請參閱第 448 頁的「記錄自動重建的配置範例」。

▼ 配置定期自動重建存取與錯誤記錄

此程序配置目錄代理伺服器存取記錄。若要配置目錄代理伺服器錯誤記錄，請執行相同的程序，但以 error 取代 access。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1. **(可選擇) 檢視存取記錄的特性。**

   ```
   $ dpconf get-access-log-prop -h host -p port
   ```

2. **(可選擇) 檢視存取記錄特性的有效值。**

   ```
   $ dpconf help-properties access-log
   ```

3. **若要在記錄到達特定大小時自動重建記錄，請設定下列特性：**

   ```
   $ dpconf set-access-log-prop -h host -p port \
    log-rotation-policy:size log-rotation-size:maximum file size
   ```

   若未指定最大檔案大小的單位，則使用**位元組**做爲預設單位。當記錄檔到達定義的大小時，即自動重建記錄。檔案大小必須最小爲 1 MB，最大不超過 2 GB。

   如需有關如何依大小自動重建記錄的範例，請參閱第 448 頁的「根據記錄大小自動重建記錄」。

4. **若不論記錄大小為何都要定期自動重建記錄，則設定下列特性：**

   ```
   $ dpconf set-access-log-prop -h host -p port \
    log-rotation-frequency:interval in months, weeks, hours, or minutes \
   ```

```
 log-rotation-policy:periodic \
 log-rotation-start-day:day in week (1-7) or day in the month (1-31) \
 log-rotation-start-time:time of day (hhmm)
```

如果將記錄配置爲在某月 31 日自動重建，但該月份少於 31 天，則在次月的第一天自動重建記錄。

如需有關如何定期自動重建記錄的範例，請參閱第 448 頁的 「根據時間自動重建記錄」。

5 **若要在記錄檔足夠大時定期自動重建記錄，請設定** `log-rotation-frequency` **與** `log-min-size` **特性**。

```
$ dpconf set-access-log-prop -h host -p port \
 log-rotation-frequency:interval in months, weeks, hours, or minutes \
 log-rotation-policy:periodic log-min-size:minimum file size
 log-rotation-start-day:day in week (1-7) or day in the month (1-31) \
 log-rotation-start-time:time of day (hhmm)
```

`log-min-size` 特性表示記錄的最小大小。自動重建作業僅在記錄檔大於指定大小時於排程時間執行。

如果將記錄配置爲在某月 31 日自動重建，但該月份少於 31 天，則在次月的第一天自動重建記錄。

如需有關如何在檔案大小足夠大時定期自動重建記錄的範例，請參閱第 449 頁的 「根據時間與記錄大小自動重建記錄」。

▼ 手動自動重建存取與錯誤記錄檔

此程序會自動重建目錄代理伺服器存取記錄。若要自動重建目錄代理伺服器錯誤記錄，請執行相同的程序，但以 `error` 取代 `access`。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **自動重建存取記錄**。

```
$ dpconf rotate-log-now -h host -p port access
```

▼ 停用存取與錯誤記錄自動重建

此程序會停用目錄代理伺服器存取記錄的自動重建。若要停用目錄代理伺服器錯誤記錄的自動重建，請執行相同的程序，但以 `error` 取代 `access`。

- **停用記錄檔自動重建**。

```
$ dpconf set-access-log-prop -h host -p port enable-log-rotation:false
```

記錄自動重建的配置範例

如何依記錄大小和/或時間配置記錄自動重建的範例。

根據記錄大小自動重建記錄

本節中的範例顯示如何僅依據記錄大小來配置記錄自動重建。不論上次自動重建記錄的時間爲何，此配置會在記錄到達 10 MB 時即自動重建。

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-policy:size \
  log-rotation-size:10M
```

根據時間自動重建記錄

本節中的範例顯示，不論記錄大小爲何，均依據上次自動重建的時間配置記錄自動重建。

- 此配置在當天 3:00 自動重建記錄，接著不論記錄檔的大小爲何，每 8 小時自動重建一次。

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:8h \
   log-rotation-policy:periodic log-rotation-start-time:0300
  ```

- 此配置不論記錄檔的大小爲何，在每天的 3:00、13:00 與 23:00 自動重建記錄。由於 `log-rotation-start-time` 參數優先於 `log-rotation-frequency` 參數，因此記錄會在 23:00 自動重建，然後在 4 小時之後再次自動重建。而不會在 23:00 自動重建，並於 10 小時後再自動重建一次。

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:10h \
   log-rotation-policy:periodic log-rotation-start-time:0300
  ```

- 此配置在星期一中午自動重建記錄，接著於每週同一時間自動重建，而不論記錄檔大小爲何。

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:1w \
   log-rotation-policy:periodic log-rotation-start-day:2 log-rotation-start-time:1200
  ```

- 此配置在星期一中午自動重建記錄，接著每三天自動重建一次，而不論記錄檔大小爲何。

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:3d \
   log-rotation-policy:periodic log-rotation-start-day:2 log-rotation-start-time:1200
  ```

  在下列幾天自動重建記錄：星期一、星期四、星期日、星期三等。請注意，`log-rotation-start-day` 參數僅適用於第一週。第二週的星期一不會自動重建記錄。

- 此配置在當月 22 日中午自動重建記錄，接著不論記錄大小爲何，會於每月同一時間自動重建。

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:1m \
 log-rotation-policy:periodic log-rotation-start-day:22 \
 log-rotation-start-time:1200
```

如果 `log-rotation-start-day` 設爲 31 而該月僅有 30 天，則在次月的第一天自動重建記錄。如果 `log-rotation-start-day` 設爲 31 而該月僅有 28 天 (二月)，則在次月第 3 天自動重建記錄。

根據時間與記錄大小自動重建記錄

本範例顯示如何配置在檔案大小足夠大時，依指定間隔自動重建記錄。

此配置在每天 3:00、11:00 與 19:00 當記錄檔大小超過 1 MB 時，自動重建記錄。如果記錄檔大小未超過 1 MB，則不自動重建記錄檔。

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:8h \
 log-rotation-policy:periodic log-min-size:1M log-rotation-start-time:0300
```

刪除目錄代理伺服器記錄

目錄代理伺服器可讓您依據時間、大小或可用磁碟空間 (預設值) 配置記錄刪除作業。如需有關這些刪除策略的更多資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Log File Deletion」。

下列程序配置存取記錄的記錄刪除。若要配置錯誤記錄的記錄刪除，請使用相同指令，但以 `error` 取代 `access`。

▼ 配置根據時間刪除存取與錯誤記錄

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **指定記錄檔的最長期限。**

  ```
  $ dpconf set-access-log-prop -h host -p port max-age:duration
  ```

  其中 *duration* 包含天 (`d`)、週 (`w`) 或月 (`M`) 等單位。例如，若要刪除五天前的備份記錄檔，請使用此指令：

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 max-age:5d
  ```

▼ 配置根據檔案大小刪除存取與錯誤記錄

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **指定記錄檔的大小上限。**

```
$ dpconf set-access-log-prop -h host -p port max-size:memory-size
```

例如，若要刪除大於 1 MB 的備份記錄檔，請使用此指令：

```
$ dpconf set-access-log-prop -h host1 -p 1389 max-size:1M
```

▼ 配置根據可用磁碟空間刪除存取與錯誤記錄

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

- **指定最小的可用磁碟空間。**

```
$ dpconf set-access-log-prop -h host -p port min-free-disk-space-size:memory-size
```

例如，若要在可用磁碟空間少於 2 MB 時刪除備份記錄檔，請使用此指令：

```
$ dpconf set-access-log-prop -h host1 -p 1389 min-free-disk-space-size:2M
```

將警示記錄至 syslogd 常駐程式

本節說明如何配置將警示訊息記錄至 syslogd 常駐程式，以及如何配置作業系統接受 syslog 警示。

▼ 配置目錄代理伺服器將警示記錄至 syslogd 常駐程式

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **(可選擇) 檢視系統記錄警示特性的目前值。**

```
$ dpconf get-server-prop -h host -p port syslog-alerts-enabled \
 syslog-alerts-facility syslog-alerts-host
```

系統記錄警示的預設特性如下：

```
syslog-alerts-enabled   :  false
syslog-alerts-facility  :  USER
syslog-alerts-host      :  localhost
```

syslog-alerts-host 特性將 syslogd 常駐程式主機名稱定義爲向其傳送訊息之主機的名稱。syslog-alerts-facility 特性爲唯讀，且可能導致將訊息傳送至系統記錄的 user 類別中。

2 **啓用將警示訊息記錄至 syslogd 常駐程式。**

```
$ dpconf set-server-prop -h host -p port syslog-alerts-enabled:true
```

3 **(可選擇) 傳送警示訊息至不同主機上的 syslogd 常駐程式。**

```
$ dpconf set-server-prop -h host -p port syslog-alerts-host:hostname
```

配置作業系統接受 syslog 警示

本節提供有關配置 Solaris™、Linux 與 HP-UX 作業系統接受 syslog 警示的指示。

▼ 配置 Solaris 作業系統接受 syslog 警示

1 **將適當的功能增加至 syslog 配置檔。**

例如，若要儲存所有使用 USER 功能的警示，請將以下行增加至 /etc/syslog.conf：

```
user.info       /var/adm/info
```

其中 /var/adm/info 是在其中儲存訊息的本機目錄範例。繼續之前，請確保存在 /var/adm/info。

2 **重新啓動 syslogd 常駐程式。**

a. **在 Solaris 8 與 9 上，鍵入以下內容重新啓動 syslogd：**

```
$ /etc/init.d/syslog stop | start
```

b. **在 Solaris 10 上，鍵入以下內容重新啓動 syslogd：**

```
$ svcadm restart system/system-log
```

3 **驗證是否將訊息記錄在 syslog 中。**

```
$ logger -p user.info "Test message"
$ cat /var/adm/info
 Jun 19 17:18:38 host user: [ID 12345 user.info] Test message
```

▼ 配置 Linux 接受 syslog 警示

1. **將適當的功能增加至 syslog 配置檔。**
   例如，若要儲存所有使用 USER 功能的警示，請將以下行增加至 /etc/syslog.conf：

   ```
   user.info       /var/adm/info
   ```

   其中 /var/adm/info 是在其中儲存訊息的本機目錄範例。繼續之前，請確保存在 /var/adm/info。

2. **將 syslogd 常駐程式配置為使用 -r 選項執行。**
   此選項允許 syslogd 接受來自網路的連線。依預設，未設定 -r 選項。

   若要設定 -r 選項，請將下行增加至 /etc/sysconfig/syslog：

   ```
   SYSLOGD_OPTIONS="-m 0 -r"
   ```

   如果 /etc/sysconfig/syslog 不存在，請將相同行增加至 /etc/init.d/syslog。

3. **重新啓動 syslogd 常駐程式。**

   ```
   $ /etc/init.d/syslog stop | start
   ```

4. **驗證是否將訊息記錄在 syslog 中。**

   ```
   $ logger -p user.info "Test message"
   $ cat /var/adm/info
    Jun 19 17:18:38 host user: [ID 12345 user.info] Test message
   ```

▼ 配置 HP-UX 接受 syslog 警示

1. **將適當的功能增加至 syslog 配置檔。**
   例如，若要儲存所有使用 USER 功能的警示，請將以下行增加至 /etc/syslog.conf：

   ```
   user.info       /var/adm/info
   ```

   其中 /var/adm/info 是在其中儲存訊息的本機目錄範例。繼續之前，請確保存在 /var/adm/info。

2. **重新啓動 syslogd 常駐程式。**

   ```
   $ /sbin/init.d/syslogd stop | start
   ```

3. **驗證是否將訊息記錄在 syslog 中。**

   ```
   $ logger -p user.info "Test message"
   $ cat /var/adm/info
    Jun 19 17:18:38 host user: [ID 12345 user.info] Test message
   ```

經由目錄代理伺服器與目錄伺服器存取記錄追蹤用戶端請求

若要追蹤用戶端請求的路徑，必須瞭解在目錄代理伺服器存取記錄與目錄伺服器存取記錄中記錄請求的方式。若要瞭解本章節，請先閱讀「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Tracking Client Requests Through Directory Proxy Server and Directory Server Access Logs」。

▼ 追蹤由目錄伺服器經目錄代理伺服器至用戶端應用程式的作業

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 **在目錄伺服器存取記錄中找到要追蹤之作業的連線編號。**

例如，存取記錄中的以下行顯示連線編號爲 conn=12839 的作業 op=2。

```
[20/Jul/2006:18:01:49 -0500] conn=12839 op=2 msgId=4 - SRCH base="dc=example,dc=com"
 scope=2 filter="(objectClass=organizationalunit)" attrs=ALL
```

2 **取得該連線的目錄代理伺服器連線資訊。**

若要取得此資訊，請搜尋目錄伺服器存取記錄，找出具有對應連線編號的所有作業。例如，在 UNIX 系統上，執行下列 grep 指令可找出目錄伺服器存取記錄中對應連線 conn=12839 的所有行：

```
$ grep conn=12839 access
```

顯示初始 LDAP 連線的行即是您要尋找的行，且與此類似：

```
[19/Jul/2006:16:32:51 -0500] conn=12839 op=-1 msgId=-1 - fd=27 slot=27
 LDAP connection from 129.153.160.175:57153 to 129.153.160.175
```

上面一行顯示存在自 **129.153.160.175:57153** 至目錄伺服器的 LDAP 連線。連接埠號碼 (**57153**) 是將連線連結回目錄代理伺服器存取記錄必要的資訊。連接埠號碼可讓您尋找目錄代理伺服器記錄中對應的連線，並從該連線找出用戶端資訊。

如果在初次建立連線之後已自動重建記錄檔，您必須同時搜尋歸檔的記錄檔與目前的存取記錄檔。

3 **找出目錄代理伺服器存取記錄中對應的連線。**

若要取得此資訊，請搜尋目錄代理伺服器存取記錄，找出具有對應連結埠號的所有作業。

您可能會在記錄檔中找到許多項目皆具有相同的連接埠號碼。若要確保找到正確的項目，請將目錄伺服器記錄項目的時間戳記納入搜尋中。

例如，在 UNIX 系統上執行下列 grep 指令，以在目錄伺服器記錄中找出與時間戳記和連接埠號碼對應的連線項目：

```
$ grep 19/Jul/2006:16:32 access | grep 57153
```

請注意，考量到伺服器時間會有些微差異，時間戳記會排除**秒**值。

目錄代理伺服器記錄中對應的行應與此類似：

```
[19/Jul/2006:16:32:51 -0500] - SERVER_OP  - INFO  - Created BIND  LDAP connection
 s_conn=sunds-d1m1-9389:34 client=0.0.0.0:57153
 server=idm160.central.sun.com:9389 main
```

此行顯示目錄代理伺服器已建立 BIND 連線至 s_conn=sunds-d1m1-9389:34。目錄代理伺服器將其本身識別爲 TCP 連接埠 57153 上的用戶端 client=0.0.0.0。

可從此記錄行擷取的重要資訊爲伺服器 ID 和連接埠號碼 (**s_conn=sunds-d1m1-9389:34**)。

4 **找出所有對應到上一步驟中識別的伺服器 ID 與連接埠號碼的作業。**

若要取得此資訊，請搜尋目錄代理伺服器存取記錄，找出具有對應伺服器 ID 與連結埠號的所有作業。

例如，在 UNIX 系統上，執行下列 grep 指令可找出上一個步驟中找到的伺服器 ID 對應的作業：

```
$ grep s_conn=sunds-d1m1-9389:34 access
```

在此狀況下，由於這些作業可能耗時數日，因此搜尋時間戳記毫無作用。但是，您必須確定搜尋是否傳回了正確的作業。如果有多個 Create 連線陳述式，請確保找到對應原始搜尋陳述式的作業。若要執行此項作業，請比較此時間戳記與步驟 1 中找到的時間戳記。

目錄代理伺服器存取記錄的下列擷取顯示針對 s_conn=sunds-d1m1-9389:34 傳回的所有作業。

```
[19/Jul/2006:16:32:51 -0500] - SERVER_OP  - INFO - Created BIND LDAP connection
 s_conn=sunds-d1m1-9389:34 client=0.0.0.0:57153 server=idm160.central.sun.com:9389 main
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=0 BIND dn="cn=directory manager"
 method="SIMPLE" s_msgid=3 s_conn=sunds-d1m1-9389:34
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=0 BIND RESPONSE err=0 msg=""
 s_conn=sunds-d1m1-9389:34
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=1 SEARCH base="dc=example,dc=com"
 scope=2 s_msgid=4 s_conn=sunds-d1m1-9389:34
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=1 SEARCH RESPONSE err=0 msg=""
 nentries=1 s_conn=sunds-d1m1-9389:34
```

此資訊表明，目錄代理伺服器上此搜尋作業的連線 ID 爲 31 (conn=31)。

5 **找出上一步驟中找到的連線 ID 對應的用戶端連線 IP 位址。**

若要取得此資訊，請搜尋目錄代理伺服器存取記錄，找出具有正確連線 ID 與時間戳記的所有作業。所使用的時間戳記是步驟 1 的原始搜尋陳述式中的時間戳記。

例如，在 UNIX 系統上，執行下列 `grep` 指令可找出用戶端連線 IP 位址：

```
$ grep "20/Jul/2006:18:01" access | grep conn=31
```

您關注的行應類似與此：

```
[20/Jul/2006:18:01:49 -0500] - CONNECT - INFO  - conn=31 client=129.150.64.156:2031
server=0.0.0.0:11389 protocol=LDAP
```

6 **確定上一步驟中找到的 IP 位址的擁有者。**

利用此資訊，您可以精確建立目錄伺服器上所執行作業的負責人。

◆ ◆ ◆ 第 29 章

29 目錄代理伺服器監視與警示

監視會偵測目錄代理伺服器和資料來源是否失敗。

如需目錄代理伺服器的監視架構描述和 cn=monitor 項目的詳細配置，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Monitoring Directory Proxy Server」。本章包含下列主題：

- 第 457 頁的 「擷取關於目錄代理伺服器的監視資料」
- 第 457 頁的 「擷取關於資料來源的監視資料」
- 第 460 頁的 「配置目錄代理伺服器的管理警示」
- 第 462 頁的 「透過使用 JVM 擷取關於目錄代理伺服器的監視資料」

擷取關於目錄代理伺服器的監視資料

若要擷取有關目錄代理伺服器的監視資料，請使用 cn=monitor 項目。此項目由目錄代理伺服器在本機常駐記憶體資料庫上進行管理。您可以在 cn=monitor 項目上執行 LDAP 搜尋，以擷取 cn=monitor 下的屬性。您必須以代理伺服器管理員身份連結，才能搜尋此項目。

如需有關使用 JVM 擷取監視資料的資訊，請參閱第 462 頁的 「透過使用 JVM 擷取關於目錄代理伺服器的監視資料」。

擷取關於資料來源的監視資料

如需目錄代理伺服器如何監視資料來源資料運作狀態的描述，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Reference」中的「Monitoring Data Sources」。本節說明如何配置對資料來源的監視。

▼ 透過偵聽錯誤監視資料來源

在此監視類型中，目錄代理伺服器偵聽目錄代理伺服器與資料來源之間通訊的錯誤。此監視類型稱爲被動監視，因爲目錄代理伺服器是在偵測到錯誤時做出反應，而不主動測試資料來源。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **將針對資料來源的監視模式設為** `reactive`。

```
$ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-mode:reactive
```

2 **如第 460 頁的 「配置目錄代理伺服器的管理警示」中所述，配置要在偵測到錯誤或資料來源離線或上線時傳送的警示。**

▼ 透過定期建立專屬連線監視資料來源

如果指定間隔內沒有發送至或來自某項資料來源的請求或回應，目錄代理伺服器即會建立此資料來源的專屬連線。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **將針對資料來源的監視模式設為** `proactive`。

```
$ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-mode:proactive
```

2 **配置監視搜尋請求由目錄代理伺服器執行。**

```
$ dpconf set-ldap-data-source-prop -h host -p port datasource \
  monitoring-bind-timeout:timeout monitoring-entry-dn:dn \
  monitoring-search-filter:filter monitoring-entry-timeout:timeout
```

搜尋請求中使用下列特性：

`monitoring-bind-timeout`	目錄代理伺服器與資料來源建立連線之前的等待時間長度。依預設，此特性的值爲 5 秒。
`monitoring-entry-dn`	搜尋請求中的目標項目的 DN。依預設，此特性爲根 DSE 項目 ("")。
`monitoring-search-filter`	搜尋篩選。
`monitoring-entry-timeout`	目錄代理伺服器等待搜尋回應的時間長度。依預設，此特性的值爲 5 秒。

3. **(可選擇)配置主動監視以特定使用者身份連結。**

   ```
   $ dpconf set-ldap-data-source-prop ldap-data-source \
   monitoring-bind-dn:uid=user-id monitoring-bind-pwd-file:password-file
   ```

   使用有效的 dn (如 `uid=bjensen,dc=example,dc=com`) 取代 `user-id`，以及使用包含密碼的檔案路徑取代 `password-file`。

   依預設，連結會以匿名身份執行，亦即 `monitoring-bind-dn` 與 `monitoring-bind-pwd` 屬性會同時設定爲 `none`。

4. **設定輪詢間隔。**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-interval:interval
   ```

   如果連線中斷，目錄代理伺服器會依此間隔輪詢連線以偵測其是否已回復。依預設，監視間隔爲 30 秒。

5. **如第 460 頁的 「配置目錄代理伺服器的管理警示」中所述，配置要在偵測到資料來源為離線或上線時傳送的警示。**

▼ 透過測試建立的連線監視資料來源

在此監視類型中，目錄代理伺服器會定期在每項資料來源的每個連線上執行搜尋。這樣，目錄代理伺服器可偵測關閉的連線，從而避免連線因無活動而遭中斷。

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **將針對資料來源的監視模式設為 `proactive`。**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-mode:proactive
   ```

2. **設定時間間隔，在此時間之後，目錄代理伺服器會將請求傳送至資料來源，以避免連線中斷。**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource \
    monitoring-inactivity-timeout:time
   ```

   依預設，無活動逾時爲 120 秒。

3. **(可選擇)配置主動監視以特定使用者身份連結。**

   ```
   $ dpconf set-ldap-data-source-prop ldap-data-source
   monitoring-bind-dn:uid=user-id monitoring-bind-pwd-file:password-file
   ```

   使用有效的 dn (如 `uid=bjensen,dc=example,dc=com`) 取代 `user-id`，以及使用包含密碼的檔案路徑取代 `password-file`。

依預設，連結會以匿名身份執行，亦即 monitoring-bind-dn 與 monitoring-bind-pwd 屬性會同時設定爲 none。

4 **如第 460 頁的「配置目錄代理伺服器的管理警示」中所述，配置要在偵測到資料來源為離線或上線時傳送的警示。**

配置目錄代理伺服器的管理警示

如需有關如何配置管理警示的資訊，請參閱下列程序。

▼ 啓用管理警示

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的「目錄服務控制中心介面」與 DSCC 線上說明。

1 **檢視啓用的警示。**

```
% dpconf get-server-prop -h host -p port enabled-admin-alerts
```

2 **啓用一或多個管理警示。**

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts:alert1 \
  [enabled-admin-alerts:alert2 ...]
```

例如，若要啓用所有可用的警示，請執行此指令：

```
% dpconf set-server-prop -h host -p port \
 enabled-admin-alerts:error-configuration-reload-failure-with-impact \
 enabled-admin-alerts:error-server-shutdown-abrupt \
 enabled-admin-alerts:info-configuration-reload \
 enabled-admin-alerts:info-data-source-available \
 enabled-admin-alerts:info-server-shutdown-clean \
 enabled-admin-alerts:info-server-startup \
 enabled-admin-alerts:warning-configuration-reload-failure-no-impact \
 enabled-admin-alerts:warning-data-source-unavailable \
 enabled-admin-alerts:warning-data-sources-inconsistent \
 enabled-admin-alerts:warning-listener-unavailable
```

若要停用所有警示，請執行此指令：

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts:none
```

若要將警示增加至現有已啓用的警示清單，請執行此指令：

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts+:alert-name
```

若要從現有已啓用的警示清單中移除警示，請執行此指令：

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts-:alert-name
```

依預設，不啓用警示。

另請參閱 如需詳細資訊，請參閱 `enabled-admin-alerts`(5dpconf)。

▼ 將管理警示配置為傳送至 Syslog

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **如第 460 頁的 「啓用管理警示」中所述，選取要傳送至 `syslog` 常駐程式的警示。**
2. **啓用要傳送至 `syslog` 常駐程式的警示。**

   ```
   $ dpconf set-server-prop -h host -p port syslog-alerts-enabled:true
   ```

   使用 `USER` 功能將所有警示傳送至 `syslog`。
3. **設定要向其傳送警示之 `syslog` 常駐程式的主機名稱。**

   ```
   $ dpconf set-server-prop -h host -p port syslog_hostname:hostname
   ```

▼ 將管理警示配置為傳送至電子郵件

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1. **如第 460 頁的 「啓用管理警示」中所述，選取要傳送至 `syslog` 的警示。**
2. **配置電子郵件位址與特性。**

   ```
   $ dpconf set-server-prop -h host -p port email-alerts-smtp-host:host-name \
     email-alerts-smtp-port:port-number \
     email-alerts-message-from-address:sender-email-address \
     email-alerts-message-to-address:receiver-email-address \
     [email-alerts-message-to-address:receiver-email-address ...] \
     email-alerts-message-subject:email-subject
   ```
3. **啓用要傳送至電子郵件的警示。**

   ```
   $ dpconf set-server-prop -h host -p port email-alerts-enabled:true
   ```

4 **(可選擇) 設定旗標以將警示碼納入電子郵件中**

```
$ dpconf set-server-prop -h host -p port \
 email-alerts-message-subject-includes-alert-code:true
```

▼ 將管理警示配置為執行程序檔

您可以使用 DSCC 執行此作業。如需相關資訊，請參閱第 47 頁的 「目錄服務控制中心介面」與 DSCC 線上說明。

1 **如第 460 頁的 「啓用管理警示」中所述，選取要傳送至 `syslog` 的警示。**

2 **啓用要執行程序檔的警示。**

```
$ dpconf set-server-prop -h host -p port scriptable-alerts-enabled:true
```

3 **設定所要執行程序檔的名稱。**

```
$ dpconf set-server-prop -h host -p port scriptable-alerts-command:script-name
```

透過使用 JVM 擷取關於目錄代理伺服器的監視資料

目錄代理伺服器在 Java 虛擬機器 (JVM) 內部執行，且依賴 JVM 機器的記憶體。若要確保目錄代理伺服器正常執行，您必須監視 JVM 機器的記憶體使用量。

如需有關如何微調 JVM 機器之參數的資訊，請參閱「Sun Java System Directory Server Enterprise Edition 6.3 Deployment Planning Guide」中的「Hardware Sizing For Directory Proxy Server」。

依預設，JVM 機器的堆疊大小爲 250 MB。如果目錄代理伺服器沒有足夠的實體記憶體，堆疊大小可能小於 250 MB。

目錄代理伺服器執行時，您可以監視 JVM 機器的堆疊大小，以確保足夠的記憶體。若要執行這項作業，請使用 Java 開發工具組 (JDK) 隨附的標準工具。這些工具位於下列目錄：`$JAVA_HOME/bin/jps` 與 `$JAVA_HOME/bin/jstat`。

▼ 檢視 JVM 堆疊大小

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

- **檢視 JVM 堆疊大小。**

```
$ dpadm get-flags instance-path jvm-args
jvm-args: -Xms250M  -Xmx250M
```

▼ 在目錄代理伺服器執行時監視 JVM 堆疊大小

無法使用 DSCC 執行此作業。請依照此程序中的說明使用指令行。

1 檢視目錄代理伺服器實例的 PID

```
$ jps
```

2 檢視 JVM 機器使用的記憶體。

```
$ jstat -gcutil PID
```

- 如果零欄接近 100%，則表示 JVM 機器的記憶體不足。
- FGC 是完整資源回收 (GC) 的事件數。資源回收佔用很大空間。
- GCT (資源回收時間) 是 GC 耗費的時間量。

索引

A

ACI
- 包含逗號的目標 DN，152
- 代理權限範例，150-151
- 使用巨集 ACI，156
- 使用回溯變更記錄，255
- 使用範例，141

ACI 儲存庫，377

`alternate-search-base-dn`, 配置，332

C

CoS
- 多值屬性 (merge-schemes)，211
- 角色型 CoS，215
- 建立
  - 從指令行的指標 CoS，213
  - 從指令行的間接 CoS，213
  - 從指令行的範本項目，212
  - 從指令行的類別 CoS，214
- 產生操作屬性，210
- 範本之間的優先權，212
- 覆寫實際屬性值，210

cosAttribute 屬性類型，210

cosClassicDefinition 物件類別，214

cosIndirectDefinition 物件類別，213

cosIndirectSpecifier 屬性類型，213

cosPointerDefinition 物件類別，213

cosPriority 屬性類型，212

cosSpecifier 屬性類型，214

cosSuperDefinition 物件類別，209

cosTemplateDN 屬性類型，214

D

db2ldif 公用程式, 匯出複本，236

DIGEST-MD5, 請參閱 SASL，118

dpadm
- create，311
- delete，314
- info，311
- restart，313
- start，312
- stop，313

dpconf
- get-server-prop，317
- LDAP 資料來源
  - `create-ldap-data-source`，321
  - `get-ldap-data-source-prop`，322
  - `list-ldap-data-sources`，322, 323
  - `set-ldap-data-source-prop`，323
- LDAP 資料來源池
  - `create-ldap-data-source-pool`，324
  - `get-ldap-data-source-pool-prop`，324
  - `list-ldap-data-source-pools`，324
  - `set-ldap-data-source-pool-prop`，325
- set-server-prop，317

`dpconf info`，314

dsadm，51
- 說明，53

dsadm create，57

dsadm delete，60

dsadm start，61-62
dsadm stop，61-62
DSCC，45, 47
　存取，47
　管理使用者，47
dsconf，51
　說明，53
　環境變數，52
`dsconf info`，67
dse.ldif 檔案
　從備份復原，191
　備份，188

E
`excluded-subtrees`, 配置，332

G
GSSAPI, 請參閱 SASL，121

I
install-path，27
instance-path，27
`isw-`*hostname* 目錄，27

J
Java Naming and Directory Interface，26
JDBC 表格, 關係，387
JDBC 表格, 屬性與物件類別，386
JDBC 資料檢視，392
　配置，392
　測試，394

K
Kerberos, 請參閱 SASL，121

L
LDAP 用戶端, 透過 SSL 進行認證，123
LDAP 資料來源
　附加至 LDAP 資料來源，325
　建立，321
　配置，322
LDAP 資料來源池
　附加 LDAP 資料來源，325
　建立，324
　配置，324
LDAP 資料檢視，326
　建立，327
　配置，327
　測試，391
ldapdelete 公用程式, 刪除項目，92
ldapmodify 公用程式, 修改項目，86
ldapsearch 公用程式，92
ldif2ldap 公用程式，194

M
Message Queue，26

N
nsComplexRoleDefinition 物件類別，205
nsFilteredRoleDefinition 物件類別，205
nsManagedRoleDefinition 物件類別，204
nsMatchingRule 屬性類型，281
nsNestedRoleDefinition 物件類別，206
nsRoleDefinition 物件類別，204
nsRoleDN 屬性類型，205, 206
nsRoleFilter 屬性類型，205
nsRoleScopeDN 屬性類型，206
nsSimpleRoleDefinition 物件類別，204

R
ref 屬性類型，97
referral 物件類別，97
rwd 關鍵字，319
rws 關鍵字，319

S

SASL，105
  DIGEST-MD5 的身份識別對映，119
  DIGEST-MD5 範圍，124
  GSSAPI，121
  GSSAPI 與 Kerberos 的身份識別對映，122
  Kerberos，121
  在用戶端中使用 Kerberos，125
  配置用戶端中的 DIGEST_MD5，123
  配置伺服器上的 DIGEST-MD5，118
  配置伺服器上的 GSSAPI，121
  配置伺服器上的 Kerberos，121
SLAMD Distributed Load Generation Engine，25
SSL，105
  用戶端認證，116
  安裝伺服器憑證，109
  信任憑證授權單位，109, 343
  配置用戶端使用 SSL，123
  搭配複寫，242
SSL 密碼, SSL 協定，420

T

TLS，105

U

UID 唯一性外掛程式，289

V

VLV 索引, 請參閱編製索引與瀏覽索引，285
子類型
  LDIF 更新陳述式中的語言，90
  二進位屬性，89
分佈，363
中央記錄目錄，28
用戶端相似性，360
  以連線爲基礎的路由，362
  複寫延遲，361
  驗證每個寫入作業，362
用戶端認證，439
用戶端請求, 追蹤，453
目標, 包含逗號的 DN，152
目錄代理伺服器實例，311
  刪除，313
  狀態，312
  建立，311
  重新啓動，313
  啓動, 停止，312
目錄代理伺服器實例
  備份，319
  復原，320
目錄伺服器
  使用 DSCC 修改項目，86
  配置，74
  控制存取，139
目錄伺服器管理員，47
目錄服務控制中心，45
目錄項目, 從指令行管理，86
目錄管理員，47
  配置，70, 317
  權限，70, 317
以連線爲基礎的路由器，436
代理授權，150
  ACI 範例，150-151
本機使用者對映，424
本機記錄目錄，27
巨集 ACI
  語法，158
  範例，156
  簡介，156
回溯變更記錄
  ACI，255
  修剪，254
  簡介，252
多重值特性, 設定，54
安全性，105
  用戶端認證，116
安裝, 目錄編輯器將於他處介紹，41
存取控制
  包含逗號的目標 DN，152
  匿名存取，149-150
  簡介，139
存取與錯誤記錄，444
自訂分佈演算法，365

自訂搜尋限制，435
自動重建記錄
  手動，447
  存取與錯誤記錄，446
尾碼，284
  刪除尾碼，66
  重新編製尾碼的索引，284
  設定尾碼層級的參照，64
  從指令行建立，62
  備份整個目錄，187
  暫時停用，64
  壓縮，66
角色，203
  角色型服務類別 (CoS)，215
  建立
    從指令行的巢式角色，206
    從指令行的管理角色，204
    從指令行篩選的角色，205
  篩選的
    範例，205-206
刪除記錄，449
  可用磁碟空間，450
  依據時間，449
  依據檔案大小，450
伺服器根目錄，27
每個帳戶的資源限制，83
串聯複寫, 請參閱複寫，240
物件類別
  另請參閱模式
  cosClassicDefinition，214
  cosIndirectDefinition，213
  cosPointerDefinition，213
  cosSuperDefinition，209
  nsComplexRoleDefinition，205
  nsFilteredRoleDefinition，205
  nsManagedRoleDefinition，204
  nsNestedRoleDefinition，206
  nsRoleDefinition，204
  nsSimpleRoleDefinition，204
  referral，97
使用者存取, 範例，144
重新初始化尾碼以重新編製索引，284
重新命名屬性, DN，330
指令行公用程式
  dsadm start，61-62
  dsadm stop，61-62
  ldapmodify，86
負載平衡，351
  配置加權，352
  容錯移轉演算法，358
負載平衡演算法，353
  比例演算法，353
  飽和演算法，354
後端 LDAP 伺服器，345
  SSL，419
  連線數，418
  匯出憑證，347
  增加憑證，346
根 DN, 請參閱目錄管理員，70, 317
記錄，293
  目錄代理伺服器，443
記錄自動重建，446
  停用，447
記錄警示，450
逗號, 在 DN 中, ACI 目標與，152
連接埠號碼, 目錄伺服器配置，74
連結規則，382
  使用者存取範例，144
  匿名存取
    範例，149-150
  群組存取範例，146
連結資料檢視，380
  建立，395
  測試，396
連結檢視, 輔助檢視，382
連線，417
  用戶端，427
連線池等待逾時，419
連線處理程式，427
  DN 篩選特性，429
連線逾時，418
配置
  目錄代理伺服器，315
  匯出用戶端憑證，347
配置特性，54
配置偵聽程式，439
配置項目, 存取，329

配置變更, 需要重新啓動，318
索引, 限制大小，282-284
索引清單臨界值, 限制大小，282-284
匿名用戶端使用者對映，425
匿名存取, 範例，149-150
動態群組, 請參閱群組，202
唯一屬性外掛程式, 配置，290
參照
  全域參照，96
  建立智慧型參照，96
  設定尾碼層級的參照，64
  預設參照，96
參照完整性
  記錄檔，217
  搭配複寫，242
  簡介，217
  屬性，218
階段作業逾時，72
國際化, 修改項目，90
堆疊大小，463
密碼策略
  工作表，167-168
  允許寬限認證，181-182
  安全密碼修改，179
  直接指定專用策略，174
  建立第一次登入策略，176-179
  建立專用策略，172-174
  使用角色與 CoS 指定專用策略，174-176
  重設密碼，180-181
  追蹤上次認證，167
  配置預設密碼策略，170-171
  密碼值，165-166
  密碼過期，166
  密碼變更，165
  帳號封鎖，164-165
  概念，164-168
  管理帳號封鎖，184-185
  檢視預設密碼策略，169-170
帳號封鎖，184-185
帳號啓用，184-185
  重新啓用帳號，185
  停用帳號，184-185
  帳號狀態，184
備份資料，187
備份資料 (續)
  dse.ldif 伺服器配置檔案，188
復原備份
  dse.ldif 伺服器配置檔案，191
  複寫注意事項，195
逾時延遲，72
運算屬性, 由角色產生，203
項目
  使用 DSCC 修改，86
  從指令行刪除，92
  從指令行修改，86
  從指令行管理，86
  尋找，92
虛擬，375
虛擬存取控制，378
虛擬配置，390
  LDAP 目錄, MySQL 資料庫，390
虛擬資料檢視，375
  存取控制，377
  模式檢查，379
虛擬轉換，407
  範例，408
匯入 LDIF，192
  從指令行，194
彙總資料，399
資料庫壓縮，66
資料檢視
  JDBC 資料檢視，383
  LDIF 資料檢視，375
  不同的資料來源
    上層與從屬子樹狀結構，338
    子樹狀結構，337
    子樹狀結構的各部分，370
  多個資料等效的來源，335
  相似性，430
  階層與分佈演算法，371
  路由所有請求，334
  預設資料檢視，333
資料儲存，397
資源限制策略，434
搜尋，92
搜尋資料隱藏規則，432
遠端使用者對映，424
預設位置，26-29

預設的自行簽署憑證，341
群組，202
　存取控制範例，146
　動態群組，202
　參照完整性管理，217
複寫，221
　初始化串聯複本，240
　建立複寫協議，231
　使用 SSL，242
　參照完整性配置，242
　經由 WAN，244
　與早期版本的相容性，252
　監視狀態，255
　確保同步化，250
管理簡介，45
管理警示，460
實例
　刪除，60
　建立，57
　啓動、停止與重新啓動，61
監視，457
　記錄檔，293
　從指令行，301
　複寫狀態，255
監視資料來源
　專屬連線，458
　測試建立的連線，459
認證，440
　SASL 外部連結，441
　匿名，441
　憑證型，441
認證方法, 代理授權，150
模式，261-278
　延伸與保留自訂檔案名稱，275
　刪除物件類別定義，273
　刪除屬性類型定義，270
　物件類別允許的 (可選擇) 屬性，271
　物件類別的必要 (必備的) 屬性，271
　建立物件類別定義，271-272
　建立屬性類型定義，268-269
　使用檔案與複寫延伸，276
　透過 LDAP 延伸，275-276
　檢查，261-262
　檢視物件類別定義，272
模式 *(續)*
　檢視屬性類型定義，269
範圍, 在 SASL DIGEST-MD5 中，124
編製索引
　刪除索引檔案，282
　建立用戶端搜尋的瀏覽索引，285
　重新初始化尾碼以重新編製索引，284
　重新編製尾碼的索引，284
　瀏覽索引，285
請求
　後端 LDAP 伺服器，421
　　用戶端身份識別，423
　　代理授權，422
　　連結轉寄，421
　　替代使用者，423
請求篩選策略，431
篩選的角色, 範例，205-206
操作相似性演算法
　全域帳號封鎖，355
　快取最佳化，357
憑證，342
　CA 簽署的憑證，342
　　安裝，343
　　更新，344
　存取資料庫，348
　　停用提示，348
　　提示輸入密碼，348
　非預設自行簽署，342
　清單，344
　備份與復原，347
憑證型認證，116
憑證資料庫, 預設路徑，27
憑證層級，116
環境變數，52
擷取監視資料
　目錄代理伺服器，457
　資料來源，457
舊有工具，55
瀏覽索引, 請參閱編製索引，285
屬性
　使用參照完整性，217
　從指令行增加二進位值，89
屬性唯一性, 請參閱 UID 唯一性外掛程式，289

屬性類型
另請參閱模式
cosAttribute，210
cosIndirectSpecifier，213
cosPriority，212
cosSpecifier，214
cosTemplateDN，214
nsMatchingRule，281
nsRoleDN，205, 206
nsRoleFilter，205
nsRoleScopeDN，206
ref，97