

DB2 Universal Database



Image、Audio 及 Video Extenders 管理與程式設計

版本 8

DB2 Universal Database



Image、Audio 及 Video Extenders 管理與程式設計

版本 8

使用本資訊及其支援的產品之前，請先閱讀注意事項下的一般資訊。

第一版 (2003 年 6 月)

本文件包含 IBM 的所有權資訊。本文出自於授權合約，受到著作權法的保護。本出版品中的資訊不包括任何產品保證，本手冊中的任何敘述亦不應該做這樣的解讀。

您可以訂購 IBM 線上出版品，或透過當地的 IBM 業務代表訂購。

- 若要訂購線上出版品，請至 IBM Publications Center，網址為 www.ibm.com/shop/publications/order
- 若要洽詢當地的 IBM 業務代表，請至 IBM Directory of Worldwide Contacts，網址為 www.ibm.com/planetwide

若要從美國或加拿大的 DB2 Marketing and Sales 訂購 DB2 出版品，請電洽 1-800-IBM-4YOU (426-4968)。

當您傳送資訊給 IBM 時，即表示您授與非專用權給 IBM，IBM 得以適當方式使用或分送此資訊，不必對您負責。

© Copyright International Business Machines Corporation 1998, 2003. All rights reserved.

目錄

圖	ix
表	xi
關於本書	xiii
適合本書的讀者	xiii
如何使用本書	xiii
平台專屬資訊	xiv
強調的慣例	xiv
如何閱讀語法圖	xv
相關資訊	xv
如何傳送意見	xvii

第 1 篇 簡介 1

第 1 章 簡介 3

管理 Extender 伺服器	3
建立 Extender 環境	3
新增和捨棄資料庫分割區 (僅適用於 EEE).	4
停止和啓動 Extender 伺服器	5
顯示伺服器狀態	5
建立及管理多重伺服器案例	6
清除管理支援表格	8
伺服器的管理命令	9
DMBICRT	10
DMBIDROP	12
DMBILIST	13
DMBIMIGR	14
DMBSTART	15
DMBSTAT	17
DMBSTOP	18
在已分割資料庫系統中重新分配 Extender 資料 (僅適用於 EEE)	19
重新分配 DB2 資料	19
重新分配 Extender 資料	19
偵測視訊場景變更	19
何謂視訊場景變更?	20
尋找及使用場景變更	21

第 2 章 概觀 41

開發 DB2	41
------------------	----

搜尋資訊的強力新方法	42
DB2 Extender	42
SDK 和執行環境	42
使用 Extender	43
範例	43
範例 1：根據特性擷取視訊	44
範例 2：依內容來搜尋影像	46
作業環境	49

第 3 章 DB2 Extender 概念 51

物件導向觀念	51
大型物件	52
使用者定義類型	52
使用者定義函數	53
UDF 和 UDT 名稱	53
觸發函式	54
Extender 資料結構	55
管理支援表	55
控點	56
QBIC 型錄	57
視訊索引	58
快照型錄	58
已分割資料庫概念 (僅適用於 EEE)	59
平行處理	61
可調整性	61
在分割的資料庫環境中使用 DB2 Extender	61
安全性和復原	61

第 4 章 Extender 的作業方式 63

Extender 實務範例	63
啓動 Extender 服務程式	64
準備資料庫	64
準備表格	66
變更表格	66
將資料插入表格中	68
從表格選取資料	70
顯示和播放物件	70
更新表格中的資料	71
從表格中刪除資料	72

第 2 篇 管理影像、音效和視訊資料 73

第 5 章 管理概觀	75
可以使用 DB2 Extender 來執行的管理作業	75
第 6 章 準備 Extender 資料的資料物件	79
啓用資料庫	79
範例	79
啓用表格	82
啓用直欄	85
停用資料物件	86
第 7 章 追蹤資料物件和媒體檔案	87
檢查資料物件的狀態	87
尋找用來參照檔案的表格項目	88
尋找表格項目所參照的檔案	89
檢查媒體檔案是否存在	90
第 8 章 授與及取消管理支援表格的專用權	91
第 3 篇 影像、音效和視訊資料的程式設計	93
第 9 章 程式設計概觀	95
使用 Extender UDF 和 API	95
您可以使用 Extender UDF 和 API 來執行的作業	96
Extender 範例的範例表格	96
DB2 Extender 程式設計的事前準備	97
併入 Extender 定義	99
指定 UDF 和 UDT 名稱	100
傳輸大型物件	100
處理回覆碼	103
Unicode 支援	104
第 10 章 儲存、擷取及更新物件	105
影像、音效及視訊格式	105
影像轉換選項	106
儲存影像、音效或視訊物件	107
DB2Image、DB2Audio 及 DB2Video UDF 格式	108
儲存常駐在從屬站的物件	111
儲存常駐在伺服器上的物件	112
指定資料庫或檔案儲存體	113
識別儲存體的格式	114
以使用者提供的屬性儲存物件	115
儲存縮圖 (影像與視訊)	117
儲存註解	118

擷取影像、音效或視訊物件	119
擷取作業的 Content UDF 格式	119
將物件擷取至從屬站	121
將物件擷取至伺服器檔案	122
擷取與使用屬性	124
擷取註解	126
更新影像、音效或視訊物件	126
用於更新的 Content UDF 格式	127
用於更新的 Replace UDF 格式	129
更新從屬站的物件	132
更新伺服器的物件	133
指定進行更新的資料庫或檔案儲存體	134
識別更新的格式	135
更新具有使用者提供之屬性的物件	136
更新縮圖 (影像與視訊)	137
更新註解	138
第 11 章 顯示或執行影像、音效或視訊物件	141
使用顯示或播放 API	141
識別顯示或播放程式	141
指定 BLOB 或檔案內容	142
指定等待指示符	143
顯示縮圖影像或視訊框	144
顯示最大圖影像或視訊框	144
播放音效或視訊	145
第 12 章 依內容來查詢影像	147
如何按影像內容查詢	147
管理 QBIC 型錄	148
建立 QBIC 型錄	149
開啓 QBIC 型錄	150
變更自動編目設定	151
將特性加入 QBIC 型錄	152
從 QBIC 型錄中移除特性	153
擷取 QBIC 型錄的相關資訊	154
手動編目影像	155
取消編目影像	156
重新編目影像	157
重新配置 QBIC 型錄 (僅適用於 EEE)	157
關閉 QBIC 型錄	158
刪除 QBIC 型錄	158
QBIC 型錄範例程式	159
建置查詢	163
指定查詢字串	163
使用查詢物件	166
按影像內容發出查詢	173

查詢影像	173
擷取影像分數	174
QBIC 查詢範例程式	176

第 4 篇 參照 183

第 13 章 使用者定義類型和使用者定義函數 187

綱目	187
使用者定義類型	187
使用者定義函數	187
AlignValue	191
AspectRatio	193
BitsPerSample	194
BytesPerSec	195
Comment	196
CompressType	198
Content	199
DB2Audio	204
DB2Image	207
DB2Video	211
Duration	215
Filename	216
FindInstrument	217
FindTrackName	218
Format	219
FrameRate	220
GetInstruments	221
GetTrackNames	222
Height	223
Importer	224
ImportTime	225
MaxBytesPerSec	226
NumAudioTracks	227
NumChannels	228
NumColors	229
NumFrames	230
NumVideoTracks	231
QbScoreFromName	232
QbScoreFromStr	234
QbScoreTBFromName	235
QbScoreTBFromStr	237
Replace	239
SamplingRate	242
Size	243
Thumbnail	244
TicksPerQNote	246

TicksPerSec	247
Updater	248
UpdateTime	249
Width	250

第 14 章 應用程式設計介面 251

DBaAdminGetInaccessibleFiles	252
DBaAdminGetReferencedFiles	254
DBaAdminIsFileReferenced	256
DBaAdminReorgMetadata	258
DBaDisableColumn	260
DBaDisableDatabase	262
DBaDisableTable	264
DBaEnableColumn	266
DBaEnableDatabase	268
DBaEnableTable	270
DBaGetError	272
DBaGetInaccessibleFiles	274
DBaGetReferencedFiles	276
DBaIsColumnEnabled	278
DBaIsDatabaseEnabled	280
DBaIsFileReferenced	282
DBaIsTableEnabled	284
DBaPlay	286
DBaPrepareAttr	289
DBaReorgMetadata	290
DBiAdminGetInaccessibleFiles	292
DBiAdminGetReferencedFiles	294
DBiAdminIsFileReferenced	296
DBiAdminReorgMetadata	298
DBiBrowse	300
DBiDisableColumn	303
DBiDisableDatabase	305
DBiDisableTable	307
DBiEnableColumn	309
DBiEnableDatabase	311
DBiEnableTable	313
DBiGetError	315
DBiGetInaccessibleFiles	317
DBiGetReferencedFiles	319
DBiIsColumnEnabled	321
DBiIsDatabaseEnabled	323
DBiIsFileReferenced	325
DBiIsTableEnabled	327
DBiPrepareAttr	329
DBiReorgMetadata	330

DBvAdminGetInaccessibleFiles	332	QbCreateCatalog	423
DBvAdminGetReferencedFiles	334	QbDeleteCatalog	425
DBvAdminIsFileReferenced	336	QbGetCatalogInfo	427
DBvAdminReorgMetadata.	338	QbListFeatures	429
DBvBuildStoryboardFile	340	QbOpenCatalog	431
DBvBuildStoryboardTable.	342	QbQueryAddFeature	433
DBvClose.	344	QbQueryCreate	435
DBvCreateIndex.	346	QbQueryDelete	437
DBvCreateIndexFromVideo	348	QbQueryGetFeatureCount	438
DBvCreateShotCatalog.	350	QbQueryGetString	440
DBvDeleteShot	352	QbQueryListFeatures	442
DBvDeleteShotCatalog.	354	QbQueryNameCreate	444
DBvDetectShot	356	QbQueryNameDelete	446
DBvDisableColumn.	358	QbQueryNameSearch	448
DBvDisableDatabase	360	QbQueryRemoveFeature	450
DBvDisableTable	362	QbQuerySearch	452
DBvEnableColumn	364	QbQuerySetFeatureData	454
DBvEnableDatabase	366	QbQuerySetFeatureWeight	456
DBvEnableTable	368	QbQueryStringSearch	458
DBvFrameDataTo24BitRGB	370	QbReCatalogColumn	460
DBvGetError.	372	QbRemoveFeature	462
DBvGetFrame	374	QbSetAutoCatalog	464
DBvGetInaccessibleFiles	376	QbUncatalogImage	466
DBvGetReferencedFiles	378		
DBvInitShotControl.	380	第 15 章 從屬站的管理命令	469
DBvInitStoryboardCtrl	382	輸入 DB2 Extender 管理命令	469
DBvInsertShot	384	取得 DB2 Extender 命令的線上解說	469
DBvIsColumnEnabled	386	ADD QBIC FEATURE	470
DBvIsDatabaseEnabled	388	CATALOG QBIC COLUMN	471
DBvIsFileReferenced	390	CLOSE QBIC CATALOG	472
DBvIsIndex	392	CONNECT	473
DBvIsTableEnabled.	394	CREATE QBIC CATALOG.	474
DBvMergeShots.	396	DELETE QBIC CATALOG.	476
DBvOpenFile	398	DISABLE COLUMN	477
DBvOpenHandle	400	DISABLE DATABASE	478
DBvPlay	402	DISABLE TABLE.	479
DBvPrepareAttrs	405	DISCONNECT SERVER AT NODENUM (僅 適用於 EEE)	480
DBvReorgMetadata.	406	DISCONNECT SERVER FOR DATABASE (僅適用於 EEE)	481
DBvSetFrameNumber	408	DISCONNECT SERVER FOR DATABASE AT NODENUM (僅適用於 EEE).	482
DBvSetShotComment	410	ENABLE COLUMN	483
DBvUpdateShot	412	ENABLE DATABASE	484
DMBRedistribute (僅適用於 EEE).	414	ENABLE TABLE	485
QbAddFeature	416	GET EXTENDER STATUS.	487
QbCatalogColumn	418		
QbCatalogImage.	420		
QbCloseCatalog	422		

GET INACCESSIBLE FILES	488	重新格式化追蹤資訊	542
GET QBIC CATALOG INFO	490	顯示追蹤狀態	542
GET REFERENCED FILES.	491		
GET SERVER STATUS	493	附錄 A. 設定 DB2 Extender 的環境變數	543
OPEN QBIC CATALOG.	494	如何使用環境變數來分辨檔名	543
QUIT	495	如何使用環境變數來識別顯示或播放程式	544
RECONNECT SERVER AT NODENUM (僅 適用於 EEE)	496	如何使用 DB2MMDATAPATH 環境變數 (僅 適用於 EEE)	544
RECONNECT SERVER FOR DATABASE (僅 適用於 EEE)	497	設定環境變數	545
RECONNECT SERVER FOR DATABASE AT NODENUM (僅適用於 EEE)	498	在 AIX、HP-UX、Solaris 伺服器與從屬站 中設定環境變數	545
REDISTRIBUTE NODEGROUP (僅適用於 EEE)	499	在 Windows 伺服器與從屬站中設定環境變 數	547
REMOVE QBIC FEATURE.	501	附錄 B. 範例程式和媒體檔案	549
REORG	502	範例程式	549
SET QBIC AUTOCATALOG	504	範例影像、音效及視訊檔案	551
START SERVER (僅適用於非 EEE).	505	範例 Net.Data 巨集檔	551
STOP SERVER (僅適用於非 EEE)	506		
TERMINATE	507	注意事項	559
		程式設計介面資訊	560
		商標	561
第 16 章 診斷資訊	509	名詞解釋	563
處理 UDF 回覆碼	509	索引	567
處理 API 回覆碼	510		
SQLSTATE 訊息碼	510	連絡 IBM	579
訊息	515	產品資訊	579
診斷追蹤	541		
啓動追蹤	541		
停止追蹤	541		



1. 清除管理支援表格的範例程式碼	8	18. 更新表格中的資料	72
2. 視訊 storyboard	20	19. 啓用資料庫的範例程式碼	80
3. 如何使用 DBvStoryboardCtrl 結構中的值	36	20. 啓用表格的範例程式碼	84
4. 多媒體資料庫表格	44	21. 啓用直欄的範例程式碼	85
5. 用來存取視訊的查詢	45	22. 檢查資料庫是否已啓用的範例程式碼	88
6. 用來存取和播放視訊的應用程式	45	23. 檢查檔案是否由使用者表格來參照的範例 程式碼	89
7. 依內容來搜尋影像	46	24. 可取得被參照的檔案清單的範例程式碼	90
8. 依內容來搜尋影像的應用程式	48	25. 使用於 DB2 Extender 程式設計範例的表 格	97
9. DB2 Extender 平台	50	26. 使用 DB2 Extender 的應用程式	98
10. 管理支援表格	56	27. 按影像內容來查詢	147
11. Handles	57	28. QBIC 型錄範例程式	160
12. 資料庫中的節點群組	60	29. QBIC 查詢範例程式	177
13. 員工表格	63	30. 執行範例 Net.Data 巨集檔的 Web 應用 程式	552
14. 加入音效直欄的員工表格	64	31. Net.Data 範例巨集檔	553
15. 將資料插入表格中	69		
16. 從表格選取資料	70		
17. 顯示和播放物件	71		

表

1. DBvShotControl 欄位	24	10. DB2 Extender 所管理的屬性	124
2. DBvStoryboardCtrl 欄位	25	11. QBIC 特性名稱	152
3. 快照型錄概略表中一些直欄	32	12. 特性值可在查詢字串中指定	164
4. Image Extender 建立的使用者定義函數	65	13. Image Extender 在 QbImageSource 中檢 查什麼	168
5. Audio Extender 建立的使用者定義函數	67	14. DB2 Extender 建立的使用者定義類型	187
6. DB2 Extender 的管理作業和機能	76	15. DB2 Extender UDF	188
7. 您可以使用 DB2 Extender API 來執行的 作業	96	16. SQLSTATE 碼及相關訊息碼	510
8. DB2 Extender 可處理的格式	105	17. DB2 Extender 的環境變數	543
9. 影像轉換選項	106		

關於本書

本書說明如何使用 DB2 Extender，來準備和維護 DB2® 資料庫的影像、音效或視訊資料。它也說明如何使用 DB2 Extender 提供的使用者定義函數 (UDF) 和應用程式設計介面 (API)，來存取和操作這些資料。將 UDF 合併至程式的 SQL 陳述式，然後合併 API 之後，您就可以存取非傳統資料，例如影像和視訊物件夾，以及傳統數值資料和字元資料。

本書提到的 "DB2" 是指 DB2 UDB。

適合本書的讀者

本書適用於熟悉 DB2 管理觀念、工具及技術的 DB2 資料庫管理員。

本書亦適用於 DB2 應用程式設計師，其應熟悉 SQL 及一種以上使用於 DB2 上的程式設計語言。

本書適用於使用 DB2 Image、Audio 及 Video Extender 來作業的人。使用 Text Extender 的人，請參閱 *DB2 Text Extender 管理與程式設計*。

如何使用本書

本書的結構如下：

『第一篇：簡介』

本篇提供 DB2 Extender 的概觀。若您是 DB2 Extender 管理或程式設計方面的新手，請閱讀本篇。

『第二篇：管理影像、音效及視訊資料』

本篇說明如何準備及維護 DB2 資料庫的影像、音效和視訊資料。若需要管理含有影像、音效或視訊資料的 DB2 資料庫，則請參閱本篇。

『第三篇：影像、音效或視訊資料的程式設計』

本篇說明如何使用 DB2 Extender UDF 和 API，來要求在影像、音效或視訊資料上的作業。若您需要在 DB2 應用程式中存取和操作影像、音效或視訊資料，則請閱讀本篇。

『第四篇：參考資料』

本篇包含 DB2 Extender UDF、API、管理命令及診斷資訊方面的參考資訊，例如訊息和程式碼。若您熟悉 DB2 Extender 觀念和作業，但需要特定 DB2 Extender UDF、API、命令、訊息或程式碼方面的資訊，則請閱讀本篇。

『附錄』

附錄說明：

- 如何設定 DB2 Extender 使用的環境變數，以便尋找檔案並識別用於影像、音效及視訊物件的顯示或播放程式
- 如何安裝和使用那些含有 Extender 的範例程式及媒體檔。

平台專屬資訊

DB2 Extender 可以與 DB2 Universal Database 的單一分割區資料庫環境或 DB2 Universal Database Enterprise-Extended Edition 的多重分割區資料庫環境一起使用。

兩種環境的 DB2 Extender 相關使用資訊都會在本書中說明。專屬於在 DB2 Universal Database Enterprise-Extended Edition 的多重分割區環境中使用 Extender 的資訊，會標示為「**僅適用於 EEE**」。專屬於在 DB2 Universal Database 的單一分割區環境中使用 Extender 的資訊，會標示為「**僅適用於非 EEE**」。非標示為屬於特定環境的資訊適用於這兩個環境。

強調的慣例

本書採用下列慣例：

粗體字 粗體字是用來表示新術語的定義。

Italics 斜體字表示那些被值取代的變數參數，或強調本文中的字詞。

UPPERCASE

大寫字母表示：

- 資料類型
- 目錄名稱
- 欄位名稱
- API 呼叫
- 命令
- 關鍵字
- 變數名稱

範例 範例文字表示系統訊息或您鍵入的值。範例文字也用於編碼範例。

如何閱讀語法圖

本書採用語法圖說明命令和 SQL 語法。閱讀語法圖的方式如下：

- 採取由左向右及從上到下的方向來閱讀語法圖，遵循線條指示的路徑。

- ▶▶ 符號表示陳述式的開始。
- ▶ 符號表示陳述式將續接下一行。
- ▶—— 符號表示陳述式接自上一行。
- ▶▶ 符號表示陳述式終止。

- 必要的項目會出現在水平線（主要路徑）上。

▶▶——必要項目——▶▶

- 可選用的項目會出現在主要路徑的下方。

▶▶
└─選用性項目┐——▶▶

- 如果您可從兩個或更多個項目中選取，則它們會以堆疊形式出現。

如果您必須選擇其中一個項目，則該堆疊的一個項目會出現在主要路徑上。

▶▶
└─必要選項1┐
└─必要選項2┐——▶▶

如果沒有選取任何項目，則整個堆疊會出現在主路徑下方。

▶▶
└─可選用的選項1┐
└─可選用的選項2┐——▶▶

堆疊上方的重覆箭頭表示您可從堆疊項目中選取一個以上的選項。

▶▶
└─可選用的選項1┐
└─可選用的選項2┐——▶▶

- 以大寫顯示關鍵字 (例如，/DB2IMAGE:)。它們的拼法必須與所顯示的一模一樣。
- 以小寫顯示變數 (例如，srcpath)。它們代表語法中由使用者提供的名稱或值。
- 如果有顯示標點符號、括弧、數學運算子或其他符號，則您必須將它們當作語法的一部份來輸入。

相關資訊

DB2 Universal Database 版本 8

版本 8 的 *DB2 Universal Database DB2 伺服器快速入門* (GC40-0731)、*DB2 從屬站快速入門* (GC40-0728)、*DB2 Connect Personal Edition 快速入門* (GC40-0730)、*DB2 Personal Edition 快速入門* (GC40-0733) 及 *IBM Data Links Manager 快速入門* (GC40-0727-00)。這些書說明如何在適當平台上規劃、安裝、架構和移轉 *DB2 Universal Database*。

IBM DB2 Universal Database Administration Guide Version 8 Planning (SC09-4822)、*Performance* (SC09-4821) 及 *Implementation* (SC09-4820)。這些書說明設計及施行 *DB2* 資料庫的方法。

IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1, Version 8 (SC09-4849)。本書說明如何使用 *DB2 CLI* 發展可存取 *DB2* 資料庫的應用程式，*DB2 CLI* 是一個可呼叫的 *SQL* 介面，它與 *Microsoft ODBC* 規格相容。

IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2, Version 8 (SC09-4850)。本書說明如何使用 *DB2 CLI* 發展可存取 *DB2* 資料庫的應用程式，*DB2 CLI* 是一個可呼叫的 *SQL* 介面，它與 *Microsoft ODBC* 規格相容。

IBM DB2 Universal Database Command Reference Version 8 (SC09-4829)。本書說明如何使用 *DB2* 命令行處理器，並提供關於 *DB2* 命令的參考資料。

DB2 Universal Database Text Extender

DB2 Universal Database Text Extender 管理與程式設計，版本 8(SC40-0526)。本書說明如何管理 *DB2* 資料庫的文字資料。它也說明如何使用 *DB2 Text Extender* 提供的應用程式設計介面，來存取和操作文字資料。

DB2 Universal Database XML Extender

DB2 Universal Database XML Extender Administration and Programming。本書說明如何管理 *DB2* 資料庫的 *XML* 文件。它也說明如何使用 *DB2 XML Extender* 提供的應用程式設計介面，來存取和操作 *XML* 文件和資料。

DB2 Universal Database Spatial Extender

Spatial Extender 使用與參考手冊 (SC40-0527)。本書提供有關 *Spatial Extender* 的安裝、架構、管理、程式設計及疑難排解等資訊。它也提供有關空間資料概念的重要說明及 *Spatial Extender* 特定的參考資訊 (訊息和 *SQL*)。

DB2 Universal Database for z/OS Image, Audio, and Video Extenders

DB2 Universal Database for z/OS Version 6 Image, Audio, and Video Extenders Administration and Programming(SC26-9650)。本書說明如何管理 *DB2 for z/OS* 資料庫伺服器的影像、音效和視訊資料。它也說明如何使用 *DB2 for*

z/OS Image, Audio and Video Extenders 所提供的使用者定義函數及應用程式設計介面，來存取和操作影像、音效及視訊資料。

DB2 Universal Database for z/OS Text Extender

DB2 Universal Database for z/OS Version 6 Text Extender Administration and Programming(SC26-9651)。本書說明如何管理 DB2 for z/OS 資料庫伺服器的文字資料。它也說明如何使用 DB2 for z/OS Text Extender 提供的使用者定義函數和應用程式設計介面，來存取和操作文字資料。

如何傳送意見

您的回饋可幫助 IBM 提供高品質資訊。您對本書或其它 DB2 Extender 文件若有任何意見，請來信指教。您可以透過下列管道提供意見：

- 透過電子郵件將意見傳送至 swsdid@de.ibm.com。請確實附上書名、書號和產品版本，可能的話請附上您的意見在文章中確實的位置（如頁碼或表格號碼）。

當您傳送資訊給 IBM 時，即表示您授與非專用權給 IBM，IBM 得以適當方式使用或分送此資訊，不必對您負責。

聯絡 IBM 的相關資訊，請參閱第 579 頁的『連絡 IBM』。

第 1 篇 簡介

第 1 章 簡介	3
管理 Extender 伺服器	3
建立 Extender 環境	3
新增和捨棄資料庫分割區 (僅適用於 EEE).	4
停止和啟動 Extender 伺服器	5
顯示伺服器狀態	5
建立及管理多重伺服器案例	6
建立多重 DB2 Extender 伺服器案例	6
列出案例	6
同時執行多重案例	7
設定現行案例	7
移除案例	7
移轉案例	8
清除管理支援表格	8
伺服器的管理命令	9
DMBICRT	10
DMBIDROP	12
DMBILIST	13
DMBIMIGR	14
DMBSTART	15
DMBSTAT	17
DMBSTOP	18
在已分割資料庫系統中重新分配 Extender 資料 (僅適用於 EEE)	19
重新分配 DB2 資料	19
重新分配 Extender 資料	19
偵測視訊場景變更	19
何謂視訊場景變更?	20
尋找及使用場景變更	21
快照偵測資料結構	22
取得快照或框	26
編目快照	31
第 2 章 概觀	41
開發 DB2	41
搜尋資訊的強力新方法	42
DB2 Extender	42
SDK 和執行環境	42
使用 Extender	43
範例	43
範例 1：根據特性擷取視訊	44

範例 2：依內容來搜尋影像	46
作業環境	49
第 3 章 DB2 Extender 概念	51
物件導向觀念	51
大型物件	52
使用者定義類型	52
使用者定義函數	53
UDF 和 UDT 名稱	53
函數路徑	54
超載函數名稱	54
觸發函式	54
Extender 資料結構	55
管理支援表	55
控點	56
QBIC 型錄	57
視訊索引	58
快照型錄	58
已分割資料庫概念 (僅適用於 EEE)	59
平行處理	61
可調整性	61
在分割的資料庫環境中使用 DB2 Extender 安全性和復原	61
第 4 章 Extender 的作業方式	63
Extender 實務範例	63
啟動 Extender 服務程式	64
準備資料庫	64
準備表格	66
變更表格	66
將資料插入表格中	68
從表格選取資料	70
顯示和播放物件	70
更新表格中的資料	71
從表格中刪除資料	72

第 1 章 簡介

管理 Extender 伺服器

DB2 Extender 執行於 DB2 主/從環境中。此環境由一部資料庫伺服器和一部以上的遠端資料庫從屬站所組成。DB2 Extender 服務程式執行於伺服器之上。在能夠存取它們之前，您必須先啟動它們。

設定好環境之後，您即可在從屬站上停止和重新啟動 Extender 服務程式。您可以從伺服器或從屬站上，取得 Extender 的狀態。

僅適用於 EEE：在多重分割區環境中，您也可以新增和捨棄資料庫分割區。

建立 Extender 環境

在伺服器的作業系統命令行上，輸入 DMBSTART 命令來啟動 Extender 服務程式：

```
dmbstart
```

DMBSTART 命令會為用來保留 Extender 資料的所有資料庫，啟動 Extender 服務程式。該命令亦啟動尚未執行的 DB2。您需要 SYSADM、SYSCTRL 或 SYSMAINT 權限，才能執行該命令。在 AIX 上，您必須以 Extender 案例擁有者的身份登入。

此時，若您的 C 語言應用程式有連接資料庫，則可以透過 API 來存取 Extender 服務程式。同樣地，若想使用 db2ext 命令行，則您必須連接所要使用的資料庫。db2ext 命令行連接必須與 DB2 命令行使用的連接不同。

在從屬站上開啓 db2ext 命令行處理器，並執行 DB2 Extender CONNECT 命令。下列範例中，該命令會連接 personnel 資料庫。它以 ANPASS 通行碼來存取限定元為 ANITAS 的表格。

```
connect to personnel user anitas using anpass
```

僅適用於 EEE：若在分割的資料庫環境中使用 DB2 Extender，則 DMBSTART 命令會在定義給該案例的所有資料庫分割區伺服器中，啟動 Extender 服務程式。若只想在一個資料庫分割區伺服器上啟動 Extender 服務程式，請在命令中指定所要啟動的節點。下列範例顯示您要在節點號碼 2 上啟動服務程式。

```
dmbstart nodenum 2
```

僅適用於 **EEE**：在啟動單一資料庫分割區伺服器之前，必須先在該節點上啟動 DB2。

現在您可以執行列示在第 469 頁的第 15 章,『從屬站的管理命令』中其餘的 DB2 Extender 命令。

新增和捨棄資料庫分割區 (僅適用於 EEE)

爲了在已分割資料庫環境中使用 Extender，對 Extender 和 DB2 所定義的分割區必須相配。DMBSTART 命令會在爲現行案例所定義的節點上啟動 Extender 伺服器。伺服器會自動偵測節點是否剛建立，且執行任何必要的起始設定。若從 DB2 捨棄節點，則該節點相關的 Extender 檔案必須以手動方式刪除。

關於新增和捨棄分割區的 DB2 命令之資訊，請參閱 *IBM DB2 Universal Database Enterprise Extended-Edition 快速入門*。

下列是新增資料庫分割區的必要步驟：

1. 使用 DB2NCRT 命令或 DB2START ADDNODE 命令建立 DB2 的分割區。
2. 使用 Extender 命令 DMBSTART NODENUM 建立 Extender 的分割區。
3. 使用 DB2 命令 REDISTRIBUTE NODEGROUP，來重新分配 DB2 資料，以利用新節點架構。
4. 使用 Extender 命令 REDISTRIBUTE NODEGROUP，來重新分配 Extender 資料，以利用新節點架構。

下列是捨棄資料庫分割區的必要步驟：

1. 使用 DB2 命令 REDISTRIBUTE NODEGROUP 來重新分配 DB2 資料，以從要捨棄的分割區上移除資料。
2. 使用 Extender 命令 REDISTRIBUTE NODEGROUP 來重新分配 Extender 資料，以從要捨棄的分割區上移除資料。
3. 使用 DB2 命令 DB2NDROP 或命令 DB2STOP DROP，來捨棄 DB2 的分割區。
4. 使用 Extender 命令 DMBSTART NODENUM 捨棄 Extender 的分割區。
5. 自行移除與捨棄的分割區相關的 Extender 檔案。

資料庫分割區的 Extender 資料，位於 *nodenum* 子目錄中，其中 *num* 代表對應至資料庫分割區的節點號碼。該子目錄位於 DB2MMDATAPATH 環境變數值所指定的目錄內。欲從被捨棄的資料庫分割區中移除 Extender 資料，請刪除適當的 *nodenum* 子目錄，及其更下層的所有子目錄。(有關 DB2MMDATAPATH 的詳細資訊，請參閱第 544 頁的『如何使用 DB2MMDATAPATH 環境變數 (僅適用於 EEE)』。)

停止和啟動 Extender 伺服器

當您將使用 Extender 服務程式的應用程式停止時，伺服器仍保持在作用中狀態，直到您明確地停止它或伺服器機器再次循環為止。您可以在伺服器機器上，於作業系統的命令行中輸入 **DMBSTOP** 命令，來停止所有的 Extender 伺服器。

欲從從屬站停止和重新啟動 Extender 服務程式，請在 **db2ext** 命令行上執行 **STOP SERVER** 和 **START SERVER** 命令。這些命令會停止和啟動現行資料庫的 Extender 服務程式。

僅適用於 EEE：在已分割資料庫環境中，可用 **DMBSTART** 來啟動為該案例所定義的所有單一伺服器，或只啟動單一資料庫分割區伺服器。**DMBSTART** 若不含參數，則會啟動所有資料庫分割區伺服器。若只想啟動一個資料庫分割區伺服器，請在命令中指定所要啟動的節點。

```
dmbstart nodenum 2
```

在特定節點啟動伺服器後，您必須將伺服器重新連接至資料庫。如下所示，使用 Extender 命令 **RECONNECT SERVER**：

```
reconnect server at nodenum 2
```

僅適用於 EEE：若在分割的資料庫環境中使用 **DB2 Extender**，沒有參數的 **DMBSTOP** 會停止為案例所定義的所有資料庫分割區伺服器。若只要停止某個資料庫分割區伺服器，您必須先切斷伺服器與資料庫的連線。如下所示，使用 Extender 命令 **DISCONNECT SERVER**：

```
disconnect server at nodenum 2
```

然後即可執行 **DMBSTOP**，並於命令中指定所要停止的節點。下列範例顯示在伺服器命令行上所要輸入的字串，以停止節點號碼 2 上的 Extender 服務程式。

```
dmbstop nodenum 2
```

僅適用於 EEE：除非您的資料庫是在維護模式中執行，否則請勿在特定節點上執行 **DMBSTOP**。此外，您要確定此節點關閉時，節點上不會觸發任何 Extender 活動。否則，可能會發生非預期的行為。

顯示伺服器狀態

您可以在伺服器上使用 **DMBSTAT** 命令，來顯示 Extender 伺服器狀態。例如，下列命令列出已啓用的資料庫，以及 Extender 是否正在執行。執行此命令之前，請先連接伺服器。

```
dmbstat
```

停止和啓動伺服器

您可使用 `GET SERVER STATUS` 命令，由從屬站取得資料庫的 Extender 伺服器狀態。例如，下列命令列出 `personnel` 資料庫的狀態：

```
get server status personnel
```

建立及管理多重伺服器案例

您可以建立及使用多重 DB2 Extender 伺服器案例。如果您有多重 DB2 伺服器案例，就應該建立多重案例。每一個 DB2 Extender 伺服器案例都有一個相關的 DB2 伺服器案例，它們具有相同名稱。您也可以列出系統上可使用的 DB2 Extender 伺服器案例、同時執行多重案例，以及移除案例。

建立多重 DB2 Extender 伺服器案例

當您安裝 DB2 Extender 時，將建立起始或預設 DB2 Extender 案例，其將與預設 DB2 案例同名。在 Windows 上，預設 DB2 Extender 案例名稱爲 `DB2`。在 UNIX 上，預設 DB2 Extender 案例與起始預設 DB2 案例同名。若要建立附加的 DB2 Extender 伺服器案例，您必須具有 `SYSADMIN` 權限，在 UNIX 上必須具有 `Root` 權限。

使用 `DMBICRT` 命令以建立附加的 DB2 Image、Audio、Video Extenders 伺服器案例。若要建立 DB2 案例 `DEVINST` 的 DB2 Extender 伺服器案例，在作業系統命令行中，請輸入：

```
dmbicrt devinst
```

執行 `DMBICRT` 命令時，將建立案例的子目錄，並將案例新增至 DB2 Extender 所維護的案例清單中。

僅適用於 EEE：

- 在 Windows 上，預設 DB2 Extender 伺服器案例名稱爲 `DB2MPP`。
- 當您使用 `DMBICRT` 以建立附加的 DB2 Image、Audio、Video Extenders 伺服器案例時，對已分割資料庫環境中的各作業，必須指定 DB2 Extender 使用的目錄。在 UNIX 上，這是 `DB2MMDATAPATH` 環境變數中所指定的目錄；在 Windows 上，這是在登錄項目中。它必須是共用目錄，並且所有節點必須已存在該案例。
- 在 Windows 上，您也必須指定 TCP/IP 埠範圍；在 UNIX 上，埠範圍必須新增至 `/etc/services` 檔案 (請參閱第 10 頁的『`DMBICRT`』)。

列出案例

使用 `DMBILIST` 命令可列出系統上可使用的 DB2 Extender 伺服器案例。若要尋找作用中案例，請輸入下列命令：

echo %DB2INSTANCE% (在 Windows 上)

echo \$DB2INSTANCE (在 UNIX)

同時執行多重案例

若要同時執行多重 DB2 Extender 伺服器案例，請執行下列步驟：

在 Windows 上

從命令行：

1. 將 DB2INSTANCE 變數設為所要啟動的案例名稱，方法是輸入：

```
set db2instance=instanceName
```

2. 啟動 Extender 服務。

在 UNIX 上

1. 以案例擁有者身分或具有系統管理權限的使用者，登入案例。
2. 設定環境。
3. 啟動資料庫管理程式。

設定現行案例

當您執行命令以啟動或停止某個案例的服務程式，命令將引用至現行案例。您可以指定所要使用的 DB2 Extender 伺服器案例，方法是將 DB2INSTANCE 變數設為案例名稱。

移除案例

若要移除 DB2 Extender 案例，請執行下列步驟：

1. 停止目前使用該案例的所有應用程式。
2. 使用 DMBSTOP 與 db2ext TERMINATE 命令，停止 Extender 服務與所有命令行處理器階段作業。
3. 對 DB2 Extender 案例目錄中所要儲存的檔案進行備份，如 QBIC 型錄檔。捨棄案例時，將移除此目錄中的檔案。
4. 輸入 DMBIDROP 命令，捨棄案例。例如，若要捨棄 DEVINST 案例，請輸入：

```
dmbidrop devinst
```

使用 DMBIDROP 命令移除 DB2 Extender 案例時，並不移除相關 DB2 案例。您必須個別移除相關 DB2 案例。當您捨棄某個 DB2 Extender 案例的相關 DB2 案例，並不會移除 DB2 Extender 案例。不過，您將無法使用它。

建立及管理多重伺服器案例

移轉案例

在 UNIX 系統上，當您安裝新版本的 DB2 UDB 及 DB2 Extender 後，您應該移轉 DB2 Extender 案例。

若要對使用舊版本建立的現存 DB2 Extender 案例進行移轉：

1. 移轉 DB2 Extender 案例的相關 DB2 UDB 案例。
2. 輸入 DMBIMIGR 命令以移轉案例。例如，若要移轉 OLDINST 案例，請輸入：

```
dmbimigr oldinst
```

清除管理支援表格

使用 DB2 Extender 時，作廢的項目最後會堆積在管理支援表格內。可能有人刪除媒體檔案，但未刪除資料庫中對該檔案的參照。刪除作廢的 meta 資料，可以改善效能和收回儲存空間。

使用 API：圖 1 中的會清除 ANITAS 擁有的所有使用者表格的影像 meta 資料。該範例包括一些錯誤檢查碼。完整範例程式位於 SAMPLES 子目錄下的 API.C 檔案。

```
/*---- query database using DBiAdminReorgMetadata API ----*/
step="DBiAdminReorgMetadata API";
rc = DBiAdminReorgMetadata("anitas");
if (rc < 0) {
    printf("%s: %s FAILED!\n", argv[0], step);
    printMsg(rc);
    DBiGetError(&sqlcode, errorMsgText);
    printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
    fail = TRUE;
} else if (rc > 0) {
    printf("%s: %s, warning detected.\n", argv[0], step);
    printMsg(rc);
    DBiGetError(&sqlcode, errorMsgText);
    printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
} else
    printf("%s: %s PASSED\n\n", argv[0], step);

/*---- end of query using DBiAdminReorgMetadata API ----*/
```

圖 1. 清除管理支援表格的範例程式碼

使用 db2ext 命令行：

```
reorg database user anitas for db2image
```

若您不是 DBA 但擁有 CONTROL 權限，則可以使用 DBxReorgMetadata API，來清除擁有表格中的 meta 資料。

伺服器的管理命令

本章的命令是在伺服器作業系統的命令行上執行。請勿在 DB2 命令行或 db2ext 命令行上執行。重新啟動您的伺服器系統時，請執行 DMBSTART 命令。

僅適用於 EEE：您也可以多重分割區資料庫環境中，發出 DMBSTART 和 DMBSTOP 伺服器命令。在多重分割區資料庫環境中發出伺服器命令時，該命令適用於所有節點，除非您包括節點號碼，此時，命令將只適用於該特定節點。

僅適用於 EEE：DMBSTAT 命令不能在多重分割區環境中執行。執行從屬站命令 GET SERVER STATUS ALL 時，可在多重分割區環境中檢查伺服器狀態。

DMBICRT

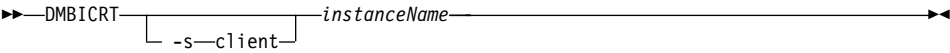
影像	音效	視訊
X	X	X

建立 DB2 Extender 案例。如果您有多重 DB2 案例，就應該建立多重 DB2 Extender 伺服器案例。在 UNIX 上，在安裝 DB2 Extender 從屬站時，需要建立從屬站案例；建立從屬站案例將設定可使用 DB2 Extender 的環境。

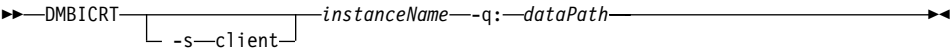
授權
SYSADM

在 UNIX 上，您必須具有 Root 權限。

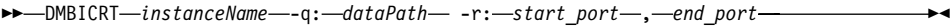
命令語法
在非分割資料庫環境中：



在 UNIX 已分割資料庫環境中：



在 Windows 已分割資料庫環境中：



命令參數

instanceName

現存的 DB2 案例的名稱。如果不存在這個名稱的 DB2 案例，將提示您是否要建立它。

-s client

指定要建立僅限單一從屬站案例。當您使用此參數，`instanceName` 即是從屬站的使用者 ID。建立從屬站案例將設定從屬站的環境。(僅適用於 UNIX)

dataPath

共用目錄或檔案系統的名稱；所有節點必須存在該目錄。在 UNIX，是設定於 `DB2MMDATAPATH` 環境變數；在 Windows，則是設定於系統登記。(僅適用於 EEE)

start_port, end_port

使用的 TCP/IP 埠範圍。埠範圍必須等於或大於您所要使用的節點數字。埠號將寫入 Windows 系統登記。(僅適用於 Windows EEE)

範例

在非分割資料庫環境中，建立 DB2 案例 DEVINST 的 DB2 Extender 伺服器案例：

```
dmbicrt devinst
```

用法注意事項

DMBICRT 命令建立案例所使用的檔案的 DB2 Extender 案例目錄。此目錄將指定為：

- *install_directory\INSTANCE\instance_name*，其中 *install_directory* 是安裝 DB2 Extender 所在的目錄 (Windows、OS/2)
- *INSTHOME/dmb* 其中 *INSTHOME* 是案例擁有者的起始目錄 (UNIX)

在使用 DMBICRT 命令時，如果不存在這個指定名稱的 DB2 案例，將提示您是否要建立它。

僅適用於 EEE：

雖然您可以從任何參與節點的 root 使用者 ID 來執行 DMBICRT，但建議您使用同一個節點來建立所有 DB2 Extender 伺服器案例。該節點應該是用來建立 DB2 案例，也是 DB2 案例目錄所在的相同節點。如果使用不同節點來建立 DB2 Extender 伺服器案例，儲存在節點上的案例清單可能不完整。

以 *dataPath* 所指定的共用目錄或檔案系統，在 UNIX 上的 *\$INSTHOME/dmb/dmbprofile* 將儲存為 DB2MMDATAPATH 環境變數值，以及 Windows 登錄機碼：

```
\\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2 Extenders\PROFILES
  \instance_name\DB2MMDATAPATH
```

在 UNIX，建立案例前，您必須將埠範圍新增至 */etc/services* 檔案。請使用下列語法，新增兩個登錄：

```
- DMB_instance_name start_port
- DMB_instance_name_END end_port
```

範圍必須足夠大，以容納已分割資料庫環境中的全部節點。

建立 DB2 Extender 伺服器案例前，您必須先建立 DB2 案例。

若要在分割的資料庫環境中建立 Text Extender 案例，請使用 TXTICRT 命令，說明於 *DB2 Universal Database Text Extender 管理與程式設計*。

DMBIDROP

影像	音效	視訊
X	X	X

捨棄 DB2 Extender 案例。

授權
SYSADM

在 UNIX 上，您必須具有 Root 權限。

命令語法
►►—DMBIDROP—*instanceName*————►◄

命令參數
instanceName
要捨棄的 DB2 Extender 案例名稱。

範例
捨棄名為 DEVINST 的 DB2 Extender 伺服器案例：
dmbidrop devinst

用法注意事項
執行此命令前：

- 停止所有使用此案例的應用程式及全部 db2ext 命令行處理器。
- 停止 Extender 服務程式。

DMBIDROP 命令將移除 DB2 Extender 案例目錄、從案例清單中移除案例登錄，並移除案例的相關資訊。

DMBIDROP 命令僅移除 DB2 Extender 案例，不移除相關的 DB2 案例。您必須明確地捨棄 DB2 案例。

當您捨棄某個 DB2 Extender 伺服器案例的相關 DB2 案例時，並不會捨棄 DB2 Extender 伺服器案例。不過，您將無法使用它。

(僅適用於 EEE) 如果您捨棄 DB2 Extender 案例，您必須從 UNIX 的 /etc/services 檔 (Windows 的 \WINNT\system32\drivers\etc\Services 檔) 移除開始與結束埠登錄。這些登錄是 DMB_*instanceName*mp 和 DMB_*instanceName*mp_END。

DMBILIST

影像	音效	視訊
X	X	X

列出所有的 DB2 Extender 案例。

授權
無。

命令語法
▶▶—DMBILIST—▶▶

命令參數
無。

範例
列出 DB2 Extender 案例：
dmbilist

DMBIMIGR

影像	音效	視訊
X	X	X

(僅適用於 **UNIX**) 將 DB2 Extender 案例從舊版本移轉至現行版本。

授權

您必須具有 Root 權限。

命令語法

►—DMBIMIGR—*instanceName*—◄◄

命令參數

instanceName

要移轉的 DB2 Extender 案例的名稱。

範例

移轉名為 OLDINST 的 DB2 Extender 案例名稱：

dmbimigr oldinst

用法注意事項

執行此命令前：

- 您必須已安裝現行版本的 DB2 Extender。
- 您必須移轉相關 DB2 案例。

每一個 DB2 Extender 案例執行一次 DMBIMIGR。請使用 DMBILIST，以列出案例。

DMBSTART

影像	音效	視訊
X	X	X

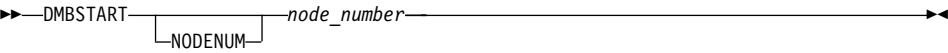
啓動 Extender 案例的所有 Extender 服務程式。

僅適用於 EEE：若指定節點，則命令只在該節點上啓動 Extender 服務程式。DMBSTART 也會在每一個節點上起始 Node Create/Drop 函數。Node Create/Drop 函數會檢查定義給 DB2 的節點，是否與定義給 Extender 的節點相配。若不相配，則 Node Create/Drop 函數視情況新增或移除節點。

授權

SYSADM

命令語法



命令參數

node_number

您要啓動 Extender 服務程式所在的節點。(僅適用於 EEE)

範例

啓動 Extender 服務程式：

dmbstart

在節點號碼 2 上啓動 Extender 服務程式：

dmbstart nodenum 2

用法注意事項

執行此命令的時機：

- 以案例擁有者登入時 (在 AIX、HP-UX 或 Solaris 上)。
- 從某個視窗中，其 DB2INSTANCE 環境變數與您要啓動的案例相同 (在 Windows 上)。
- 重新啓動您的伺服器系統時。

在單一分割區環境中，DMBSTART 也會啓動尚未執行的 DB2 案例。

DMBSTART

僅適用於 **EEE**：

在多重分割區環境中，DMBSTART 不會啟動 DB2 案例。在已分割環境中執行 DMBSTART 之前，必須先啟動 DB2。

如果 DMBSTART 失敗，檢查下列項目：

- 請確定 DBD2MMDATAPATH 變數值是正確的。
- 請確定已存在變數中的共用目錄或檔案系統，並且對每個節點皆可存取。

DMBSTAT

影像	音效	視訊
X	X	X

顯示哪些資料庫已啓用，以及這些資料庫的 Extender 服務程式是否正在執行。

授權

無

命令語法

►►—DMBSTAT—◄◄

命令參數

無。

範例

顯示 Extender 服務程式的狀態：

```
dmbstat
```

DMBSTOP

DMBSTOP

影像	音效	視訊
X	X	X

停止 Extender 案例的 Extender 服務程式。

僅適用於 **EEE**：若指定節點，則 DMBSTOP 只會停止該節點上的 Extender 服務程式。

授權
SYSADM



命令參數

node_number
您要停止 Extender 服務程式所在的節點。(僅適用於 **EEE**)

範例
停止 Extender 服務程式：

dmbstop

在節點號碼 2 上停止 Extender 服務程式：

dmbstop nodenum 2

用法注意事項

執行此命令的時機：

- 以案例擁有者登入時 (在 AIX、HP-UX 或 Solaris 上)。
- 從某個視窗中，其 DB2INSTANCE 環境變數與您要停止的案例相同 (在 Windows 上)。

DMBSTOP 不會停止 DB2 案例。

僅適用於 **EEE**：除非在維護模式中操作，否則請勿在特定節點上執行 DMBSTOP。此外，您要確定此節點關閉時，節點上不會觸發任何 Extender 活動。否則，可能會發生非預期的行為。

在已分割資料庫系統中重新分配 Extender 資料 (僅適用於 EEE)

DB2 Extended Enterprise Edition 可讓您在已分割資料庫環境中，新增和刪除資料庫分割區伺服器 (亦稱為節點)。新增節點之後 (或刪除節點之前)，現存的資料即可重新分配，以利用新架構。

若要重新分配 Extender 資料，您必須執行兩個步驟。首先，您必須重新分配 DB2 資料。然後就可以重新分配 DB2 Extender 資料。

重新分配 DB2 資料

在重新分配資料之前，您必須使用 DB2 命令 `REDISTRIBUTE NODEGROUP` 來重新分配 DB2 資料。

有關重新分配 DB2 資料的更多資訊，請參閱 *DB2 管理指南*。

重新分配 Extender 資料

重新分配 DB2 資料之後，您即可重新分配 Extender 資料。輸入 Extender 命令 `REDISTRIBUTE NODEGROUP`，來啟動 Extender 資料重新分配。

```
redistribute nodegroup
```

`REDISTRIBUTE NODEGROUP` 命令可重新分配音效、影像及視訊的 Extender 資料和 QBIC 特性資料，並將其置於相對應使用者資料的相同節點上。

若重新分配程序傳回錯誤，您可以重新執行命令。您可以根據命令回應的指示，決定是否使用 `CONTINUE` 參數來重新執行命令。此選項指示系統從停止處繼續執行，而非從頭開始執行。執行 DB2 的 `REDISTRIBUTE NODEGROUP` 命令之後，第一次執行 `REDISTRIBUTE NODEGROUP` 命令時，不能使用 `CONTINUE` 參數。

若要維護資料完整性，請一次重新配置一個節點群組。請先等待一個節點群組完成重新分配後，再開始另一個節點群組的分配。

使用此命令之前，請務必先連接資料庫。

您需要 `SYSADM` 或 `DBADM` 權限，才能執行此命令。

偵測視訊場景變更

本章說明如何在具有 DB2 Video Extender API 的視訊片段中偵測場景變更。這些 API 適用於所有 DB2 Video Extender 平台，而 Windows 3.1 除外。只有 MPEG-1 格式的視訊片段支援視訊場景變更偵測。

何謂視訊場景變更？

假定電視製片室記錄隨後播送的錄影帶節目。最近製片室已開始使用 Video Extender，將錄影帶片段存入 DB2 資料庫中。製片室專業人員可藉以查詢傳統類型節目的資訊，以及檢視節目內容。

製片室希望有預覽視訊片段的功能。他們想要有檢視視覺摘要 (稱為 storyboard) 的能力。storyboard 的範例顯示於圖 2 中。製片室人員可藉由 storyboard 來取得視訊片段清單，而不需要檢視整個視訊片段。它也可協助製片室人員來決定視訊片段是否即為所需要的正確視訊片段 (例如，它的優點是下載和檢視)。這對製片室而言是很重要的需求。檢視 storyboard 而非整個視訊片段，可減少下載和檢視所用的時間。有關以此方法來使用視訊場景變更檢視功能的詳細資訊，請參閱第 35 頁的『儲存視訊中所有快照的有關資訊』。

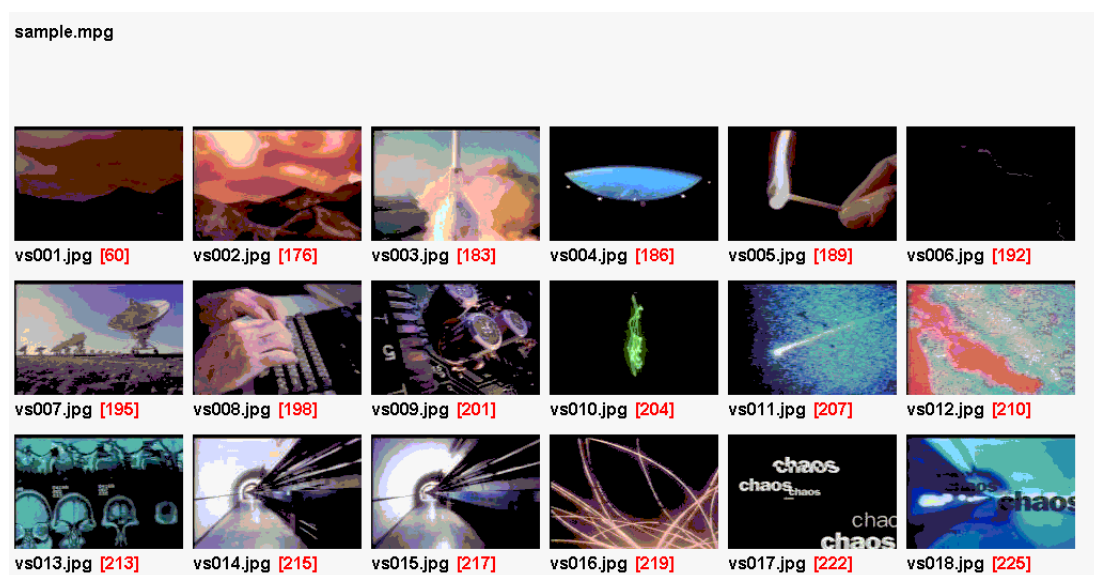


圖 2. 視訊 storyboard. 代表框摘要說明視訊的內容與流程。

製片室計劃使用 Video Extender 的視訊場景變更偵測功能，來擷取它們 storyboard 的代表框。

視訊場景變更是指視訊片段中一個點，兩個連續框在此點上有明顯的不同。例如，攝影機在錄影時改變焦點，就會出現這個點。兩個場景變更之間的框，構成一個快照。

當 Video Extender 檢視場景變更時¹，會記錄相關快照的資料。此資料包括起始快照的框號碼、結束快照的框號碼，以及快照內代表框的號碼。快照資料也包括代表框的圖點內容。

尋找及使用場景變更

Video Extender 提供一組應用程式設計介面，供您尋找視訊片段中的快照或框。找到快照或框之後，即可存取相關資料，例如快照的開始和結束框號碼，或框的圖點內容。然後，您可以將此資訊傳送到程式，作進一步處理。例如，您可以將框內容傳送到程式，以便顯示它。

Video Extender 亦提供 API，將快照資料存入**快照型錄**中。快照型錄可位於資料庫或檔案中。您可以存取檔案中的快照型錄，或存取資料庫中快照型錄的唯讀概略表。

快照型錄檔含有具下列資料的欄位：

- 快照型錄名稱
- 控制 Video Extender 偵測快照方法的一些值，例如一個快照中的最小框數
- 控制儲存多少框和哪些框作為快照代表框的一些值
- 快照號碼
- 起始框號
- 結束框號
- 代表框號碼
- 包含代表框內容的檔名

資料庫中快照型錄的概略表，含有一些具有下列資料的直欄：

- 快照控點
- 視訊表格名稱
- 視訊直欄名稱
- 視訊控點
- 視訊檔名
- 起始框號
- 結束框號
- 代表框號碼

1. 視訊場景變更偵測碼包括加州大學柏克萊分校 MPEG 解碼器，以及來自波士頓大學 Multimedia Communication Laboratory 的許多規格。

- 代表框資料
- 註解

當資料庫中有快照型錄時，您可以存取快照型錄檔中的資料或查詢資料。代表框資訊在顯示 storyboard 時特別有幫助。此外，如果資料庫中有快照型錄，您可以將快照資料與其它表格中的相關資料合併。例如，視訊製片室中的專業人員可以在資料庫中建立一個快照型錄。然後，他們可以將型錄資料與含有視訊片段及其相關資訊的表格合併。利用這個方法，他們即可使用單一查詢方式來擷取視訊片段和相關商業資訊，以及識別視訊片段內的快照。

快照偵測資料結構

快照偵測的相關資料儲存在併入於快照偵測標題檔 dmbshot.h 中的一些結構。某些快照偵測 API 需要您指出一或多個結構。所用的某些結構含有 Video Extender 作為輸入的資料。例如，快照控制結構含有控制快照偵測的資訊。Video Extender 使用部份結構來儲存從視訊片段中擷取的資料。例如，視訊框資料結構含有框的圖點內容。

快照偵測所用的結構有 DBvIOType、DBvShotControl、DBvShotType、DBvFrameData 及 DBvStoryboardCtrl。

DBvIOType: DBvIOType 資料結構含有視訊片段相關資料，例如視訊片段格式、維度及框數。定義資料結構的方法如下：

```
typedef struct {  
    FILE *hFile;                /* file handle for the video */  
    char vhandle[255];          /* video handle (if from database) */  
    char vtable[255];           /* video table name (if from database) */  
    char vcolumn[255];          /* video column name (if from database) */  
    char vFile[255];            /* name of video file */  
    char idxFile[255];          /* name of index file */  
    char isIdx;                 /* 1 if the index exists, 0 otherwise */  
    char isInDb;                /* 1 if from DB, 0 if from file */  
    int format;                 /* Format of the video */  
    unsigned long dx, dy;       /* Dimensions of the video */  
    unsigned long totalFrames;  /* TotalFrames in the video */  
    unsigned long markFrame;    /* used by shot detection */  
    unsigned long currentFrame; /* The current video frame */  
    DBvFrameData fd;            /* Frame data for current frame */  
    DBvDCFrameData fdDc;        /* Frame data for DC images */  
    unsigned char BGRValid;     /* reserved */  
    unsigned short usDeviceID;  /* reserved */  
    unsigned long hwnid;        /* reserved */  
    int videoReset;             /* Flag if video is opened or seeked */  
    int firstshot;              /* Used internally to indicate the first call */  
    void *reserved;             /* reserved */  
}  
DBvIOType;
```

DBvShotControl: DBvShotControl 資料結構含有控制快照偵測的資訊，例如偵測方法。定義資料結構的方法如下：

```
typedef struct {

    unsigned long reserved;
    unsigned long method;          /* detection method */

    #define DETECT_CORRELATION  0x00000001
    #define DETECT_HISTOGRAM    0x00000002
    #define DETECT_CORRHIST     0x00000003
    #define DETECT_CORRHISTDISS 0x00000004

    int normalCorrValue;           /* Correlation threshold */
    int sceneCutSkipXY;            /* reserved */
    int CorrHistThresh;           /* Histogram threshold */
    int DissThresh;               /* Dissolve threshold */
    int DissCacheSize;            /* Dissolve cache size */
    int DissNumCaches;            /* Dissolve cache number */
    int minShotSize;              /* Minimum frames in a shot */

} DBvShotControl;
```

表 1 說明 DBvShotControl 中每一個欄位，以及可接受和預設的設定。若要將這些欄位起始設定為預設值，請使用第 26 頁的『起始設定快照偵測資料結構中的值』中描述的 DBvInitShotControl API。

DBvShotControl 設定視視訊類型而定：數位化視訊變換中的場景變更視視訊片段內容和格式而定。另外，場景變更演算法的準確性是根據視訊片段而定。以整體框外觀之明顯差異所定義的場景變更，其被偵測到的程度，比更細微的變更類型(或整體顏色內容維持相同的變更類型)還要精確。雖然預設的一些 DBvShotControl 欄位設定較適合大部份的應用程式，但為了減少假案例或錯誤偵測情況的發生，您可能需要調整這些設定。

使用場景變更

表 1. DBvShotControl 欄位

欄位	意義
method	<p>識別 Video Extender 用來偵測場景變更的方法。您可以選擇下列其中一個方法：</p> <p>DETECT_CORRELATION. 比較兩個連續框中的圖點。如果差異值超出相關臨界值，則僅偵測一個場景變更。</p> <p>DETECT_HISTOGRAM. 比較兩個連續框的直方圖值。直方圖值可測量框中的顏色分配。如果差異值超出直方圖臨界值，則僅偵測一個場景變更。</p> <p>DETECT_CORRHIST. 使用相關方法識別可能的場景變更，然後使用直方圖方法將框標示為可能的場景變更。如果超出直方圖臨界值，則僅偵測一個場景變更。</p> <p>DETECT_CORRHISTDISS. 與 DETECT_CORRHIST 相同，但會檢查其它分解的框。</p> <p>預設方法是 DETECT_CORRHIST。</p>
normalCorrValue	指定相關臨界值的整數值 0 到 100。此提供兩個框的圖點之間的最小相關係數值。0 值表示一定偵測下一個框的場景變更。100 值表示僅在所有圖點從一個框變更到下一個框時，偵測場景變更。預設值是 60。
sceneCutSkipXY	保留。
CorrHistThresh	指定直方圖臨界值的整數值 0 到 100。此測量連續框直方圖值之間的差異。0 值表示僅在一個框與下一個框的直方圖值完全不同時，偵測場景變更。100 值表示一定偵測下一個框的場景變更。預設值是 10。
DissThresh	指定分解測試臨界值的整數值 0 到 100。此測量必須對所要偵測的分解傳遞分解測試的框中圖點的百分比。0 值表示一定偵測框的分解。100 值表示僅在框中所有圖點傳遞分解測試時，偵測分解。預設值是 15。
DissCacheSize	一種整數值，指定分解測試斜率部份使用的框數。預設值是 4。
DissNumCaches	一種整數值，指定分解測試一致性部份使用的框數。預設值是 7。
minShotSize	一種整數值，指定快照的最小框數。對於要偵測的快照而言，至少要有最小框數。預設值是 5。

DBvShotType: DBvShotType 資料結構含有快照的相關資訊，例如快照起始框號、結束框號及代表框號碼；以及代表框圖點內容的指標。定義資料結構的方法如下：

```
typedef struct {  
  
    unsigned long startFrame;    /* starting frame number */  
    unsigned long endFrame;      /* ending frame number */  
    unsigned long repFrame;      /* representative frame number */  
}
```

```
DBvFrameData fd;                /* data for representative shot */
unsigned long dx;                /* frame data width in pixels */
unsigned long dy;                /* frame data height in pixels */
char *comment;                  /* shot remark */

} DBvShotType;
```

DBvFrameData: DBvFrameData 資料結構含有框的圖點內容。定義資料結構的方法如下：

```
typedef struct                    /* video frame data */
{
    /* MPEG 1 pixels */
    unsigned char *luminance;     /* Luminance pixel plane (black and white) */
    unsigned char *Cr;           /* Cr pixel plane */
    unsigned char *Cb;           /* Cb pixel plane */
    unsigned char *reserved;
} DBvFrameData;
```

DBvStoryboardCtrl: DBvStoryboardCtrl 資料結構含有一些值，控制快照的哪些及多少代表框儲存在視訊型錄中。有關這些值的使用說明，請參閱第 36 頁的『建置 storyboard』。定義資料結構的方法如下：

```
typedef struct {
    int thresh1;                  /* threshold for small to medium scenes */
    int thresh2;                  /* threshold for medium to large scenes */
    int delta;                    /* offset used for representative frames */
} DBvStoryboardCtrl;
```

表 2說明 DBvStoryboardCtrl 中每一個欄位及其預設設定。若要將這些欄位起始設定為預設值，請使用第 26 頁的『起始設定快照偵測資料結構中的值』中描述的 DBvInitStoryboardCtrl API。

DBvStoryboardCtrl 設定依視訊類型而定：以 storyboard 而言，哪些及多少個代表框是最佳的，依不同的視訊類型會有不同。雖然預設的 DBvStoryboardCtrl 欄位設定較適合大部份視訊類型，但您可能要在視訊的測試子集上使用這些設定。然後，在為較寬的視訊設定建立 storyboard 之前，您可以調整設定來符合您的需求。

表 2. DBvStoryboardCtrl 欄位

欄位	意義
thresh1	識別短快照的臨界值。框數比 thresh1 值小的快照稱為短快照。如果編目，則短快照的資訊將併入一個代表框 (指中間框) 中。 預設值是 90。如果 thresh1 值設定為 -1，將快照將被視為一個短快照 (無論其實際長度)。

表 2. DBvStoryboardCtrl 欄位 (繼續)

欄位	意義
thresh2	<p>識別中至大型快照的臨界值。框數與 thresh2 值一樣或較小的快照，稱為中間快照。如果編目，則中間快照的資訊將併入兩個代表框中。代表框的位置由差異處(delta)欄位值控制。框數比 thresh2 值大的快照稱為長快照。如果編目，則長快照的資訊將併入三個代表框中。第一個及最後一個代表框的位置由差異處(delta)欄位值控制。第二個框是中間框。</p> <p>預設值是 150。如果 thresh2 值設定為 -1，則快照將被視為一個短快照 (無論其實際長度)。</p>
delta	<p>識別代表框使用的偏移。以中間和長快照而言，第一個代表框是藉由差異處的框數，從快照起始點偏移。最後一個代表框是藉由差異處的框數，從快照結束點偏移。</p> <p>預設值是 5。</p>

起始設定快照偵測資料結構中的值: DBvShotControl 資料結構中的值可控制快照偵測。 DBvStoryboardCtrl 資料結構中的值可控制 storyboard 的建置。您可以為這些資料結構中的欄位明確指定一些值。此外，您可以將這些結構中的值起始設定成預設值。有關 DBvShotControl 資料結構中的預設值，請參閱第 24 頁的表 1。有關 DBvStoryboardCtrl 資料結構中的預設值，請參閱第 25 頁的表 2。

使用 DBvInitShotControl API 來起始設定 DBvShotControl 資料結構中的值。使用該 API 時，您必須指定快照控制結構。例如，下列陳述式可將 DBvShotControl 結構中的欄位起始設定成預設值：

```
DBvShotControl    shotCtrl;

rc=DBvInitShotControl(
    shotCtrl);          /* pointer to shot control structure */
```

使用 DBvInitStoryboardCtrl API 來起始設定 DBvStoryboardCtrl 資料結構中的值。在使用該 API 時，您必須指定 storyboard 控制結構。例如，下列陳述式可將 DBvStoryboardCtrl 結構中的欄位起始設定成預設值：

```
DBvStoryboardCtrl    sbCtrl;

rc=DBvInitStoryboardCtrl(
    sbCtrl);          /* pointer to storyboard control structure */
```

取得快照或框

您可以使用 Video Extender 來取得視訊中的快照或框。在取得快照或框之前，您必須開啓用於快照偵測的視訊。Video Extender 使用索引來存取框和快照。在取得快照或框之前，您必須建立視訊索引。

在開啓視訊並建立索引之後，可透過快照號碼來取得視訊中下一個快照或框，或取得特定框。Video Extender 可處理 MPEG-1 格式的視訊片段。如果您計劃將所擷取的框使用在需要 RGB 格式的節目，可以使用 Video Extender API 將框轉換成該格式。

開啓用於快照偵測的視訊： 使用 DBvOpenFile API 來開啓儲存在檔案中的視訊。此檔案必須可由從屬站存取。使用 DBvOpenHandle API 來開啓儲存在資料庫表格中的視訊。應用程式必須先連接到資料庫。如果視訊是儲存在資料庫表格中，則 Video Extender 會將視訊複製到暫時檔案中。此暫時檔案存放在以從屬站環境變數 DB2VIDEOTEMP 指定的目錄中。為快照偵測開啓並起始設定一個視訊。Video Extender 在視訊起點設定一個指標，亦即框 0。

當您使用其中一個 API 時，您需指出一個區域，來包含指向視訊資料結構 (DBvIOType) 的指標。Video Extender 配置此結構來回應 API 呼叫，並使用此結構來儲存視訊的相關資訊。此結構亦指示框資料結構 (DBvFrameData)，它含有現行框的圖點內容。有關這些結構的說明，請參閱第 22 頁的『快照偵測資料結構』。以 DBvOpenFile API 而言，您也需要指定視訊檔的名稱。以 DBvOpenHandle API 而言，您也需要指定視訊控點。

例如，下列陳述式可為檔案中儲存的快照偵測，開啓一個視訊：

```
DBvIOType    *videoptr;

rc=DBvOpenFile (
    &videoptr,                                /* pointer to video structure pointer */
    "/employee/video/rsmith.mpg");          /* video file */
```

下列陳述式可為資料庫表格中儲存的快照偵測，開啓一個視訊：

```
EXEC SQL BEGIN DECLARE SECTION;
char Vid_hdl[251];
EXEC SQL END DECLARE SECTION;

DBvIOType    *videoptr;

EXEC SQL SELECT VIDEO INTO :Vid_hdl
FROM EMPLOYEE
WHERE NAME="Anita Jones";

rc=DBvOpenHandle(
    &videoptr,                                /* pointer to video structure pointer */
    vid_hdl);                                /* video handle*/
```

建立視訊索引： Video Extender 使用索引來存取視訊中的框和快照。在取得視訊的框或快照之前，您必須建立視訊索引 (MPEG 格式不提供框和快照的索引)。索引會將框號碼對映到含有 MPEG-1 視訊的位元串流。

您可以使用 `DBvCreateIndexFromVideo` API 或 `DBvCreateIndex` API，來建立視訊索引。不過，如果您已使用 `DBvOpenFile` API 或 `DBvOpenHandle` API，開啓用於快照偵測的視訊，則不需要明確地建立索引；`Video Extender` 會自動為您建立索引。（有關如何開啓視訊的資訊，請參閱第 27 頁的『開啓用於快照偵測的視訊』）。

建立索引後（明確或自動），`DB2 Video Extender` 嘗試將索引儲存在與視訊檔相同的路徑。它首先會嘗試將索引檔存成 `fname.ext.idx`，其中 `fname` 指視訊檔名，而 `ext` 是視訊檔副檔名。如果嘗試失敗，則 `Video Extender` 會嘗試在視訊檔同一個目錄中，將檔案存成 `fname.idx`。如果還是失敗，則會嘗試在本端目錄中，將索引檔先存成 `fname.ext.idx`，其次是存成 `fname.idx`。

當檔案被開啓時，`Video Extender` 會依下列次序尋找索引檔：

1. 先尋找可寫入版本的索引檔，再尋找唯讀版本的索引檔。
2. 先尋找與視訊檔相同路徑的索引檔，再尋找現行目錄中的索引檔。
3. 先尋找檔名為 `fname.ext.idx` 的索引檔，再尋找檔名為 `fname.idx` 的索引檔，其中 `fname` 指視訊檔名，而 `ext` 是視訊檔的副檔名。

例如，如果已為視訊檔 `myvideo.mpg` 建立索引，則 `Video Extender` 會先尋找與視訊檔相同路徑中檔名為 `myvideo.mpg.idx` 的可寫入式索引。

當您使用 `DBvCreateIndexFromVideo` API 時，請指定 `DBvIOType` 資料結構。`Video Extender` 會將索引檔名稱存入此結構中。有關此結構的說明，請參閱第 22 頁的『快照偵測資料結構』。例如，下列陳述式會對先前為快照偵測開啓的視訊建立索引：

```
DBvIOType      *video;

rc=DBvCreateIndexFromVideo(
    video);          /* pointer to video structure */
```

當您使用 `DBvCreateIndex` API 時，請指定視訊檔名。`Video Extender` 會將索引儲存在檔案中（此檔案位於視訊所在的相同目錄中）。例如，下列陳述式會建立視訊檔的索引（此檔案不是先前為快照偵測所開啓的檔案）：

```
rc=DBvCreateIndex(
    "/employee/video/rsmith.mpg");    /* video file */
```

您亦可判斷視訊的索引是否存在。使用 `DBvIsIndex` API 來檢查索引。如果視訊索引不存在，則 API 會將狀態變數設成 0，如果視訊索引存在，則設成 1。例如，下列陳述式可檢查視訊檔索引是否存在：

```
short *status

rc=DBvIsIndex(
    "/employee/video/rsmith.mpg",    /* video file */
    &status);                        /* status indicator */
```


備份視訊索引：備份視訊索引檔，以便需要時回復它。此檔案位於 Video Extender 安裝目錄中。

取得框：您可以取得視訊中的現行框。您亦可將現行框設成特定框號碼。使用 DBvGetFrame API，來取得視訊中的現行框。使用 DBvSetFrameNumber API，將現行框設成特定框號碼。

當您使用 DBvGetFrame API 時，請指定視訊結構。例如，下列陳述式可取得視訊中的現行框：

```
DBvIOType      *video;

rc=DBvGetFrame(
    video);          /* pointer to video structure */
```

當您使用 DBvSetFrameNumber API 時，請指定視訊結構和您要設定為現行框的框號碼。例如，下列陳述式可將現行框設成框號碼 85，然後取得框：

```
DBvIOType      *video;

rc=DBvSetFrameNumber(
    video,          /* pointer to video structure */
    85);           /* frame number */

rc=DBvGetFrame(
    video);          /* pointer to video structure */
```

輸出時，DBvSetFrameNumber API 會重設 DBvIOType 結構中的 currentFrame 欄位。DBvGetFrame API 會將框圖點內容置於 DBvFrameData 結構中。有關這些結構的說明，請參閱第 22 頁的『快照偵測資料結構』。

取得快照：使用 DBvDetectShot API 來取得視訊中下一個快照。當您使用 DBvDetectShot API 時，您必須指定下列資料結構：

- 視訊 (DBvIOType)
- 快照控制 (DBvShotControl)
- 快照類型 (DBvShotType)

您也需要指示搜尋使用的起始框。Video Extender 會依視訊中的指示開始搜尋下一個快照。

當 API 產生結果時，Video Extender 會設定 shotDetected 旗號，並指到下一個快照的起始框及其框資料。如果 shotDetected 旗號設成 1，則表示已偵測一個快照。在此情形下，Video Extender 會：

- 將 DBvIOType 中的 currentFrame 欄位設成下一個快照的起始框
- 將下一個快照的起始框資料置於 DBvIOType 中的 fd 欄位

使用場景變更

- 設定 DBvShotType，使包含起始框號、結束框號、代表框號碼、代表框資料以及下一個快照的註解

如果 shotDetected 旗號設成 0，則表示尚未偵測到快照。在此情形下，Video Extender 會傳回一個程式碼，指出已到達視訊結尾。

有關這些結構的說明，請參閱第 22 頁的『快照偵測資料結構』。

例如，下列陳述式要求視訊中的下一個快照：

```
DBvIOType      *video;
long start_frame = 1;
char shotDetected = 0;
DBvShotControl  shotCtrl;
DBvShotType     shot;

shotCtrl->method=DETECT_CORRHIST
shotCtrl->normalCorrValue=60;
shotCtrl->sceneCutSkipXY=1;
shotCtrl->CorrHistThresh=10;
shotCtrl->DissThresh=10;
shotCtrl->DissCacheSize=4;
shotCtrl->DissNumCaches=7;
shotCtrl->minShotSize=0;

rc=DBvDetectShot(
    video,                /* pointer to video structure */
    start_frame,          /* starting frame for search */
    &shotDetected,         /* shot detected flag */
                          /* 1=detected, 0=not detected */
    shotCtrl,             /* pointer to shot control structure */
    &shot);                /* pointer to shot type structure */
```

轉換擷取框的格式： MPEG-1 框內容格式為 YUV，格式包括框的亮度圖點平面、Cr 圖點平面及 Cb 圖點平面相關資訊。如果您要編輯視訊框，您會發現它適合從 YUV 格式轉換成 RGB 格式。Video Extender 提供 DBvFrameDataTo24BitRGB API，將擷取的 MPEG-1 框從 YUV 格式轉換成 24 位元 RGB 格式。若要使用 API，首先您必須配置目標緩衝區。

當發出 API 時，您必須指示所要轉換的框資料和目標緩衝區。您也要指定框的高度與寬度。(您可以從框的 DBvIOType 結構中取得框的資料、高度和寬度。) 例如，下列陳述式可將 MPEG-1 框轉換成 24 位元 RGB 格式：

```
char RGB[18000];
DBvIOType      *video;
DBvFrameData    fd;

rc=DBvGetNextFrame(
    video);                /* pointer to video structure */

fd=video.fd
```

```
dx=video.dx
dy=video.dy
```

```
rc=DBvFrameDataTo24BitRGB (
    RGB,                /* pointer to target buffer */
    &fd,                 /* pointer to frame data */
    dx,                  /* frame width */
    dy);                 /* frame height */
```

關閉視訊檔： 使用 DBvClose API 來關閉已為快照偵測開啓的視訊檔。使用該 API 時，您要對檔案的視訊結構指定一個指標。

例如，下列陳述式可關閉已為快照偵測開啓的視訊檔：

```
DBvIOType    *video;

rc=DBvClose (video);
```

顯示擷取框： 擷取 MPEG-1 框內容格式為 YUV；此格式不是大部份影像顯示程式可顯示的格式。若要顯示擷取視訊框，該框的格式必須可被影像顯示程式辨識，例如 BMP 格式。例如，要顯示 MPEG-1 框：

1. 您可以使用 DBvFrameDatato24BitRGB API，將擷取 MPEG-1 框的格式從 YUV 格式轉換成 24 位元 RGB 格式。有關使用 DBvFrameDatato24BitRGB API 的資訊，請參閱第 30 頁的『轉換擷取框的格式』。
2. 將適合的標頭加到轉換框上。例如，BMP 格式需要含有影像寬度與高度資訊的標頭。
3. 將框內容（連同其標頭）複製到檔案。
4. 使用 DBiBrowse API 來顯示該檔案。有關使用 DBiBrowse API 的資訊，請參閱第 141 頁的『使用顯示或播放 API』。

編目快照

您可以將快照相關資訊儲存在快照型錄中。Video Extender 提供 API 來：

- 在資料庫中建立及管理快照型錄。您可以使用 API 來：
 - 在資料庫中建立快照型錄
 - 將單一快照相關資訊存入快照型錄中
 - 將視訊中所有快照相關資訊存入快照型錄中
 - 變更快照型錄中所儲存的快照資訊
 - 將快照資訊合併到快照型錄中
 - 刪除快照型錄中的快照資訊
 - 刪除資料庫中的快照型錄

- 建立一個快照型錄檔，並將視訊中所有快照相關資訊存入該檔案中。提供一個 API 來建立型錄檔並填入快照資料。您可以存取及操作快照型錄檔中的資料，但所提供的 API 不具備這些功能。

為 storyboard 將提供輸入的快照編入型錄：將快照資訊存入快照型錄 (此型錄位於資料庫或檔案中) 後，您可以使用快照相關應用程式中的資訊。例如，您可以建立應用程式來取得視訊中所有快照的代表框，並在 storyboard 中顯示它們。

您僅需對資料庫建立快照型錄：只有在您要型錄常駐於資料庫時，才需要建立快照型錄。當您儲存視訊中快照的資料並指示要檔案中的輸出時，Video Extender 會自動建立一個快照型錄檔。

- 在建立型錄之前 (僅適用於資料庫):** 在資料庫中建立及使用型錄之前，您必須：
- 發出 SQLConnect 呼叫。DB2 資料庫中的快照型錄是由表格的集成所組成。在資料庫中建立快照型錄或執行作業之前，您必須透過 SQLConnect 呼叫來連接資料庫。(SQLConnect 是一種 DB2 CLI 呼叫。) 此呼叫會傳回在管理快照型錄的 API 中，您必須指定的連線控點。
 - 啓用影像資料使用的資料庫。在資料庫中建立快照型錄之前，您必須啓用 DB2Image 資料類型使用的資料庫。除了儲存在快照型錄中的其它資訊以外，Video Extender 會為每一個已編目快照儲存代表框資料。代表框資料的資料類型是 DB2Image。

建立快照型錄 (僅適用於資料庫): 使用 DBvCreateShotCatalog API，在資料庫中建立快照型錄。(當您儲存快照資料並指示要檔案中的輸出時，Video Extender 會自動建立一個快照型錄檔。). 此型錄是由儲存快照相關資訊的表格所組成。 您可以使用 SQL 來查詢表格的概略表。 表 3顯示概略表中一些直欄。

表 3. 快照型錄概略表中一些直欄

直欄名稱	資料類型	說明
SHOTHANDLE	CHAR(36)	快照控點
VIDEOHANDLE	VARCHAR(254)	視訊控點。僅在視訊使用 DBvOpenHandle API 開啓時，此直欄中才含有值。
VIDEOTABLE	VARCHAR(254)	包含視訊的表格。僅在視訊使用 DBvOpenHandle API 開啓時，此直欄中才含有值。

表 3. 快照型錄概略表中一些直欄 (繼續)

直欄名稱	資料類型	說明
VIDEOCOLUMN	VARCHAR(254)	包含視訊的表格直欄。僅在視訊使用 DBvOpenHandle API 開啓時，此直欄中才含有值。
VIDEOFILE	VARCHAR(254)	視訊檔名。僅在視訊使用 DBvOpenFile API 開啓時，此直欄才含有值。
STARTFRAME	INTEGER	起始框號
ENDFRAME	INTEGER	結束框號
REPFRAME	INTEGER	代表框號碼
REPFRAMEDATA	DB2IMAGE	代表框資料
COMMENTS	LONG VARCHAR	註解

您可以彈性地決定要在資料庫中建立多少個快照型錄，以及哪些快照要儲存資訊在每一個快照型錄中。您可以建立一個型錄來儲存許多視訊的快照資訊、將每一個視訊的快照資訊儲存在分開的型錄內、或將視訊內多個快照的資訊儲存在多個型錄內。

在使用該 API 時，您必須指定型錄的名稱。大於 16 個字元的名稱會被截斷。您也需要指定 SQLConnect 對資料庫呼叫後所傳回的資料庫連線控點。例如，下列陳述式建立一個快照型錄，名為 hotshots：

```
SQLHDBC hdbc;

rc = SQLConnect(hdbc,"hotshots",SQL_NTS,id,SQL_NTS,password,SQL_NTS);

rc=DBvCreateShotCatalog(
    "hotshots",          /* shot catalog name */
    hdbc);               /* database connection handle */
```

快照型錄概略表稱為 MMDBSYS.SVcatname，其中 catname 是指快照型錄的名稱。例如，型錄 hotshots 的概略表名稱是 MMDBSYS.SVHOTSHOTS。

儲存有關單一快照的資訊 (僅適用於資料庫): 使用 DBvInsertShot API，在快照型錄中儲存有關單一快照的資訊。只有在快照型錄位於資料庫中時，您才可以儲存視訊中單一快照的資訊。型錄中儲存的資訊包括：

- 快照控點
- 視訊表格名稱 (使用於儲存在表格中的視訊片段)
- 視訊直欄名稱 (使用於儲存在表格中的視訊片段)

使用場景變更

- 視訊控點 (使用於儲存在表格中的視訊片段)
- 視訊檔名 (使用於儲存在檔案中的視訊片段)
- 起始框號
- 結束框號
- 代表框號碼
- 代表框資料

但是，並不儲存快照的註解。有關如何新增註解到快照儲存的資訊中，請參閱第 37 頁的『指定的快照的註解 (僅適用於資料庫)』。

使用 `DBvInsertShot` API 時，您必須指定快照型錄名稱和快照指標。設定快照指標的一種方法，是取得下一個快照，請參閱第 29 頁的『取得快照』。您也需要指定 `SQLConnect` 對資料庫呼叫後所傳回的資料庫連線控點。例如，下列陳述式取得框號 1 之後的下一個快照，然後將快照的資訊儲存在名為 `hotshots` 的快照型錄中：

```
SQLHDBC  hdbc;
SQLHENV  henv;
DBvIOType *video;
long start_frame = 1;
char shotDetected = 0;
DBvShotControl shotCtrl;
DBvShotType shot;

shotCtrl->method=DETECT_CORRHIST
shotCtrl->normalCorrValue=60;
shotCtrl->sceneCutSkipXY=1;
shotCtrl->CorrHistThresh=10;
shotCtrl->DissThresh=10;
shotCtrl->DissCacheSize=4;
shotCtrl->DissNumCaches=7;
shotCtrl->minShotSize=0;

SQLAllocConnect(henv,&hdbc)

rc = SQLConnect(hdbc,"hotshots",SQL_NTS,id,SQL_NTS,password,SQL_NTS);

rc=DBvDetectShot(
    video,
    start_frame,
    &shotDetected,
    &shotCtrl,
    &shot)

rc=DBvInsertShot (
    "hotshots",          /*shot catalog name*/
    shot,                /*pointer to shot*/
    hdbc);               /*database connection handle*/
```

儲存視訊中所有快照的有關資訊： 使用 DBvBuildStoryboardTable API 或 DBvBuildStoryboardFile API，來將視訊中所有快照的資訊儲存在快照型錄中。DBvBuildStoryboardTable API 將資訊儲存在資料庫的快照型錄中。DBvBuildStoryboardFile API 會建立快照型錄檔，並在該檔中儲存快照資訊。

對任一個 API 而言，來源視訊可以位於資料庫表格或檔案中。

使用任一個 API 時，您必須：

- 指定快照型錄名稱。
- 指向視訊結構。
- 指向 DBvShotControl 資料結構。
- 指向 DBvStoryboardCtrl 資料結構。此資料結構中的值，控制哪些和多少視訊框要儲存為快照型錄內的代表框。有關設定這些值的資訊，請參閱 第 36 頁的『建置 storyboard』。

對 DBvBuildStoryboardTable API 而言，您還需要指定 SQLConnect 對資料庫呼叫後所傳回的資料庫連線控點。

例如，下列陳述式將視訊中所有快照的資訊儲存在快照型錄中：該快照型錄位於資料庫中。

```
SQLHDBC hdbc;
SQLHENV henv;
DBvIOType *video;
DBvShotControl shotCtrl;
DBvStoryboardCtrl sbCtrl;

sbCtrl->thresh1=50
sbCtrl->thresh2=500;
sbCtrl->delta=20;

SQLAllocConnect(henv,&hdbc)

rc = SQLConnect(hdbc,"hotshots",SQL_NTS,id,SQL_NTS,password,SQL_NTS);

rc=DBvBuildStoryboardTable (
    "hotshots",          /*shot catalog name*/
    video,               /*pointer to video structure*/
    shotCtrl,            /*pointer to shot control structure*/
    sbCtrl,              /*pointer to storyboard control structure*/
    hdbc);              /*database connection handle*/
```

下列陳述式建立一個快照型錄檔，並將視訊中所有快照的資訊儲存在該檔案：

```
DBvIOType *video;
DBvShotControl shotCtrl;
DBvStoryboardCtrl sbCtrl;
```

使用場景變更

```
sbCtrl→thresh1=50
sbCtrl→thresh2=500;
sbCtrl→delta=20;

rc=DBvBuildStoryboardFile (
    "hotshots",           /*shot catalog file name*/
    video,                /*pointer to video structure*/
    shotCtrl,             /*pointer to shot control structure*/
    sbctrl);              /*pointer to storyboard control structure*/
```

建置 storyboard: 顧名思義，DBvBuildStoryboardTable API 和 DBvBuildStoryboardFile API，特別適合來儲存 storyboard 中所用的資訊。**storyboard** 是視訊的視覺摘要。您可以藉由顯示快照型錄中某視訊所儲存的代表框，來建立一個 storyboard。

DBvBuildStoryboardTable API 和 DBvBuildStoryboardFile API，儲存快照的一個或多個代表框。在 DBvStoryboardCtrl 結構中指定的值，控制某快照所儲存的代表框數，以及要使用哪些框。有關 DBvStoryboardCtrl 結構的定義，請參閱第 22 頁的『快照偵測資料結構』。圖 3 說明如何使用 DBvStoryboardCtrl 欄位中的值。

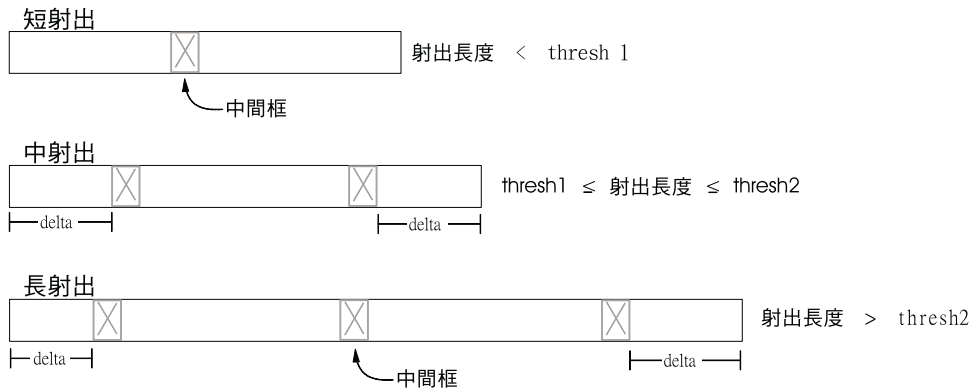


圖 3. 如何使用 DBvStoryboardCtrl 結構中的值

如圖 3 所示：

- 小型快照只儲存一個代表框。小型快照中的框數，小於 DBvStoryboardCtrl 資料結構中的 thresh1 值。代表框是快照的中間框。
- 中型快照儲存兩個代表框。中型快照內的框數，大於或等於 DBvStoryboardCtrl 資料結構中的 thresh2 值。DBvStoryboardCtrl 資料結構中的 delta 值，決定視訊開頭至第一個代表框之間的框數。delta 值亦決定第二個代表框至視訊結尾之間的框數。

- 大型快照儲存三個代表框。大型快照中的框數，大於 DBvStoryboardCtrl 資料結構中的 thresh2 值。DBvStoryboardCtrl 資料結構中的 delta 值，決定視訊開頭至第一個代表框之間的框數。第二個代表框是視訊的中間框。從視訊結尾到第三個代表框的距離，是由 delta 值所決定。

若 thresh1 或 thresh2 值設為 -1，則任何快照皆可當做小型快照來處理。此時，只有一個代表框，亦即中間框，會被儲存在快照型錄內。

除了 DBvStoryboardCtrl 資料結構中的值以外，DBvShotControl 資料結構中的許多欄位，會影響後來用以顯示於 storyboard 中的代表框。例如，DBvShotControl 資料結構中的 CorrHistThresh、normalcorrValue 及 minShotSize 欄位，指定快照偵測的臨界值，因此會影響哪些框要顯示於視訊的 storyboard 內。使用 DBvBuildStoryboardTable API 和 DBvBuildStoryboardFile API 來儲存 storyboard 中所用的快照資訊時，您可以先試著使用 DBvStoryboardCtrl 和 DBvShotControl 資料結構的起始設定值。然後，您可以在這些資料結構中變更各種欄位值，以調整您的結果。

顯示 storyboard: 您可以建立程式來顯示 storyboard。作法是存取視訊中快照型錄內所儲存的代表框。若使用 DBvBuildStoryboardFile API 來儲存視訊的快照，則快照型錄檔會指向代表框的 GIF 檔案。您可以使用瀏覽程式或顯示程式，來顯示這些 GIF 檔案。

若使用 DBvBuildStoryboardTable API 來儲存視訊的快照，則快照型錄 (儲存於資料庫中) 會包含代表框的資料。您可以在快照型錄概略表中，存取代表框資料 (有關概略表的說明，請參閱第 32 頁的表 3)。代表框資料是 YUV 格式；這種格式不是大多數顯示程式所能顯示的。若要顯示代表框，您可使用 DBvFrameDatato24BitRGB API 來轉換框資料，其有關資訊請參閱第 31 頁的『顯示擷取框』。然後，即可使用瀏覽程式或顯示程式來顯示代表框。

Storyboard 範例程式: SAMPLES 子目錄包含兩個範例程式，用以展示建置和顯示視訊的 storyboard。其中一個範例程式 makesf.exe 檔，使用 DBvBuildStoryboardFile API 來建立快照型錄檔，並將快照資料儲存於檔案中。另一個範例程式 makehtml.exe，存取快照型錄檔，並建立 Web 瀏覽程式所顯示的 HTML 網頁。

指定的快照的註解 (僅適用於資料庫): 對於快照型錄中的快照，您可以指定註解來與其它資訊儲存在一起。使用 DBvSetShotComment API 來指定註解。

使用該 API 時，您必須指定要儲存註解的快照型錄名稱、要新增註解的快照控點、以及該註解。您也需要指定 SQLConnect 對資料庫呼叫後所傳回的資料庫連線控點。例如，下列陳述式在名為 hotshots 的快照型錄中新增一個快照 (從框號 85 開始) 的註解：

使用場景變更

```
SQLHDBC hdbc;
SQLHENV henv;
char shothandle[37];

SQLAllocConnect(henv,&hdbc)

rc = SQLConnect(hdbc,"hotshots",SQL_NTS,id,SQL_NTS,password,SQL_NTS);

EXEC SQL SELECT SHOTHANDLE INTO :shothandle
FROM MMDBSYS.SVHOTSHOTS
WHERE STARTFRAME=85;

rc=DBvSetShotComment (
    "hotshots",          /*shot catalog name*/
    shothandle,          /*shot handle*/
    "shot of beach at sunset", /*comment*/
    hdbc);              /*database connection handle*/
```

變更快照所儲存的資訊 (僅適用於資料庫): 您可以變更快照型錄中某個快照所儲存的資訊。作法是使用 DBvUpdateShot API。將取代資訊放入 DBvShotType 結構中。您也需要指定其餘欄位的資訊，即使維持不變。當您使用 DBvUpdateShot API 時，請指定型錄名稱和一個指向 DBvShotType 結構的指標。您也需要指定 SQLConnect 對資料庫呼叫後所傳回的資料庫連線控點。

變更快照的資訊時，您可以變更資訊中儲存的任何註解。若您想變更註解，請在 DBvShotType 結構中指定。若您想保留以前的註解，請在 DBvShotType 結構中指定 NULL 值。

例如，下列陳述式在一個名為 hotshots 的型錄內，變更快照所儲存的資訊；該快照從框號 85 開始：

```
SQLHDBC hdbc;
SQLHENV henv;
char shothandle[37];
DBvShotType shot;
DBvFrameData fd110;

/* get shot handle */

EXEC SQL SELECT SHOTHANDLE INTO :shothandle
FROM MMDBSYS.SVHOTSHOTS
WHERE STARTFRAME=85;

/* change shot attribute */

shot.startFrame=110;
shot.endFrame=200;
shot.repframe=110;
shot.fd=fd110;
shot.comment=NULL;

/* update shot information */
```

```

SQLAllocConnect(henv,&hdbc)

rc = SQLConnect(hdbc,"hotshots",SQL_NTS,id,SQL_NTS,password,SQL_NTS);

rc=DBvUpdateShot (
    "hotshots",                /*shot catalog name*/
    shot,                      /*shot information*/
    hdbc);                     /*database connection handle*/

```

在快照型錄中合併快照資訊 (僅適用於資料庫): 您可以在一個快照型錄中，合併兩個快照所儲存的資訊。合併快照資訊時，您需指定合併次序，來識別第一個快照和第二個快照。第一個快照的起始框號，儲存為被合併快照的起始框號。第一個和第二個快照之間的最大框號，儲存為被合併快照的結束框號。合併動作會以被合併快照的資訊來取代第一個快照的資訊；第二個快照所儲存的資訊會從快照型錄中刪除。

使用 `DBvMergeShots` API 來合併快照型錄中兩個快照的資訊。使用該 API 時，請指定快照型錄名稱，之後接著要合併之第一個和第二個快照的控點。您也需要指定 `SQLConnect` 對資料庫呼叫後所傳回的資料庫連線控點。例如，下列陳述式將兩個快照所儲存的資訊合併到一個稱為 `hotshots` 的型錄內；第一個快照從框號 85 開始，第二個快照從框號 210 開始：

```

SQLHDBC hdbc;
SQLHENV henv;
char shothandle1[37];
char shothandle2[37];

EXEC SQL SELECT SHOTHANDLE INTO :shothandle1
FROM MMDBSYS.SVHOTSHOTS1
WHERE STARTFRAME=85;

EXEC SQL SELECT SHOTHANDLE INTO :shothandle2
FROM MMDBSYS.SVHOTSHOTS2
WHERE STARTFRAME=210;

SQLAllocConnect(henv,&hdbc)

rc = SQLConnect(hdbc,"hotshots",SQL_NTS,id,SQL_NTS,password,SQL_NTS);

rc=DBvMergeShots (
    "hotshots",                /*shot catalog name*/
    shothandle1,              /*shot handle for first shot*/
    shothandle2,              /*shot handle for second shot*/
    hdbc);                    /*database connection handle*/

```

刪除快照型錄中的快照資訊 (僅適用於資料庫): 使用 `DBvDeleteShot` API 來刪除快照型錄中有關某快照的資訊。使用該 API 時，請指定快照型錄名稱，再接著快

使用場景變更

照控點。您也需要指定 `SQLConnect` 對資料庫呼叫後所傳回的資料庫連線控點。例如，下列陳述式在名為 `hotshots` 的快照型錄中，刪除一個快照（從框號 85 開始）的相關資訊：

```
SQLHDBC hdbc;
SQLHENV henv;
char shothandle[37];

EXEC SQL SELECT shothandle INTO :shothandle
FROM mmdbsys.svhotshots
WHERE startframe=85;

rc=DBvDeleteShot (
    "hotshots",                /*shot catalog name*/
    shothandle,                /*shot handle*/
    hdbc);                     /*database connection handle*/
```

刪除快照型錄 (僅適用於資料庫): 使用 `DBvDeleteShotCatalog` API 來刪除快照型錄。使用該 API 時，請指定要刪除的快照型錄名稱，以及 `SQLConnect` 對資料庫呼叫後所傳回的資料庫連線控點。例如，下列陳述式刪除一個快照型錄，稱為 `hotshots`：

```
SQLHDBC hdbc;
SQLHENV henv;

rc=DBvDeleteShotCatalog (
    "hotshots",                /*shot catalog name*/
    hdbc);                     /*database connection handle*/
```

第 2 章 概觀

DB2 (DB2) Universal Database (UDB) 是功能強大的物件關聯式資料庫管理程式。它可儲存和保護傳統數字和字元資料，以及大型且複雜的物件 (LOB)。DB2 Extender 可協助您開發 DB2 的物件關聯式特性。Extender 定義影像、音效及文字物件的特定資料類型和特殊函數。好處是 Extender 可節省您在應用程式中定義這些資料類型和函數所花費的時間與精力。資料類型和函數皆可於 SQL 中使用。因此，Extender 提供一項存取利器給您的應用程式，來存取所有的資料類型，以及傳統數字和字元資料。此外，Extender 提供應用程式搜尋資訊的新方法。例如，您的應用程式可以使用顏色或結構的視覺範例，以視覺特性來搜尋影像。

開發 DB2

DB2 Extender 開發 DB2 物件導向特性。尤其，使用 DB2 您可以：

- 在 DB2 資料庫中儲存 2 GB 的 LOB。
- 定義特殊資料類型給這些大型、複雜的物件。您可使用這些使用者定義類型 (UDT)，來識別代表物件 (例如影像或音效) 的資料類型。
- 定義特定函數，以便於使用者定義類型的資料上來要求。例如，您可以定義函數來計算影像的顏色數目，或取得音效的取樣率。您可以在 SQL 陳述式中要求這些使用者定義函數 (UDF)，如同其它 SQL 函數一樣。

DB2 Extender 建立影像、音效、視訊及文字物件的 UDT 和 UDF。UDT 和 UDF 可幫助您：

- 開發應用程式。因為 Extender 定義資料類型和函數，所以您不必於應用程式中再定義。
- 確保一致性。同一組的 Extender UDT 和 UDF，可用於您所有的應用程式中。這提供現成的一致性程度，該一致性在處理大型物件的應用程式之間可能難以達成。
- 建立強力查詢。因為要求 UDF 的方式和其它 SQL 函數相同，所以應用程式可包含多重資料類型查詢。一個 SQL 陳述式可以存取影像、音效、視訊及文字物件，以及傳統數字和字元資料。您可以在內含的 SQL 陳述式中，以及 DB2 CLI 呼叫中，指定 UDF 和 UDT。

因為 Extender 所處理的物件是儲存在 DB2 資料庫中，所以這些物件的安全性、完整性及復原保護，皆與資料庫中儲存的傳統資料類型相同。

DB2 Extender

此外，DB2 Extender 使用 DB2 Universal Database Enterprise- Extended Edition 的分割資料庫環境。分割可讓應用程式使用對於單一電腦來說太大的資料庫。分割也可讓 SQL 作業以並行方式來執行，因此加速 SQL 查詢或公用程式。

搜尋資訊的強力新方法

DB2 Extender 在搜尋資訊方面，提供很大的彈性給您的應用程式。對於資料庫中儲存的傳統類型資料，您的應用程式可以搜尋與這些資料相關的物件。例如，可根據說明或錄製日期來搜尋音效片段。您的應用程式亦可根據先天的特性 (例如視訊片段的播放時間) 來搜尋物件。Extender 會自動決定和儲存這些特性，以便在搜尋中使用。

您的應用程式甚至可以根據內容來搜尋影像。假設應用程式使用視覺範例來搜尋影像。透過這個應用程式，使用者可以選取一個範例影像，並使應用程式去尋找與該範例有著相似顏色或結構的其它影像。您可以使用 DB2 Extender 的「依影像內容查詢」(QBIC) 功能，來建立以此視覺方式去搜尋影像的應用程式。

DB2 Extender

DB2 Extender 是由個別的 Image Extender、Audio Extender、Video Extender 及 Text Extender 組成。

本書涵蓋 Image、Audio 及 Video Extender。本書中提及『Extender』或『DB2 Extender』時，就是指 Image、Audio 及 Video Extender，除非另有指定。如需 Text Extender 的相關資訊，請參閱 *Text Extender 管理與程式設計*。如需 XML Extender 的相關資訊，請參閱 *XML Extender Administration and Programming*。

SDK 和執行環境

DB2 Extender 安裝套件提供 Software Developers Kit (SDK) 以及從屬站和伺服器執行環境。您可在已安裝 DB2 Extender SDK 的從屬站或伺服器機器上開發 DB2 Extender 應用程式。

您可在含有 DB2 Extender 從屬站執行碼和伺服器執行碼的伺服器機器中執行 DB2 Extender 應用程式。(安裝伺服器執行碼時會自動安裝從屬站執行碼。) 您也可以安裝 DB2 Extender 從屬站執行碼的從屬站機器上執行 DB2 Extender 應用程式。若在從屬站機器執行 Extender 應用程式，您必須確定可以連接至伺服器。

使用 Extender

您可在 DB2 應用程式中要求 Extender UDF，或者使用 DB2 命令行處理器，以交談式模式來要求 UDF。

Extender 也提供下列應用程式設計介面 (API)：

- 管理的 API，用來準備和維護資料庫，供影像、音效及視訊資料使用。
- 顯示和播放 API 以顯示影像和播放視訊和音效片段。
- QBIC API，用來準備影像及要求依內容來搜尋。(內容搜尋也可經由 UDF 來要求。)
- 視訊快照偵測 API 以識別那些以視訊中的場景變更為基礎的框順序。

DB2 Extender 也提供用來發出管理命令的命令行處理器。為了區別 Extender 提供的命令行處理器與 DB2 提供的命令行處理器，我們將前者稱為『db2ext 命令行處理器』，將後者稱為『DB2 命令行處理器』。

範例

廣告代理商維護其 DB2 廣告資料庫。在以前，該代理商儲存有關每一個廣告活動的數字和字元資料，例如客戶名稱和廣告結束的日期。安裝 DB2 UDB 以及 DB2 Extender 之後，代理商現在也可以將廣告內容儲存在資料庫。這包括印刷廣告影像、電視廣告視訊以及音效廣告錄音。如第 44 頁的圖 4 所示，所有相關廣告資訊都在一個稱為 ADS 的資料庫表格內。



圖 4. 多媒體資料庫表格. 該表格包含影像、音效及視訊資料，以及傳統資料類型。會顯示一段視訊、音效及影像。

範例 1：根據特性擷取視訊

廣告代理商的會計經理需要查看在 1997 年中為 IBM 帳戶製作、少於 30 秒的視訊廣告。

第 45 頁的圖 5顯示用來存取視訊的查詢。請注意到查詢中稱為 Filename 和 Duration 的 Video Extender UDF。


```

SELECT FILENAME(ADS_VIDEO)
  FROM ADS
 WHERE CLIENT='IBM' AND
    SHIP_DATE>='01/01/1997' AND
DURATION(ADS_VIDEO) <=30

```

圖 5. 用來存取視訊的查詢

該查詢將適當視訊的檔名傳回。然後，專案經理即可啟動其喜歡的視訊播放程式，來播放每一個視訊檔的內容。

圖 5 是專案經理可交互發出一個查詢。一般說來，專案經理會使用應用程式來尋找及播放視訊。例如，圖 6 顯示這種應用程式 (以 C 撰寫) 的一些重要元素。該應用程式在 DB2 主變數 hvVid_fname 中擷取視訊檔名。同時也請注意，應用程式使用一個稱為 DBvPlay 的播放 API 來播放視訊。

```

#include <dmbvideo.h>

int count = 0;

EXEC SQL BEGIN DECLARE SECTION;
char hvClient[30];           /*client name*/
char hvCampaign[30];         /*campaign name*/
char hvSdate[8];             /*ship date*/
char hvVid_fname [251]       /*video file name*/
EXEC SQL END DECLARE SECTION;

EXEC SQL DECLARE c1 CURSOR FOR
  SELECT CLIENT, CAMPAIGN, SHIP_DATE, FILENAME(ADS_VIDEO)
    FROM ADS
   WHERE CLIENT='IBM' AND
      SHIP_DATE>='01/01/1997' AND
      DURATION(ADS_VIDEO)≤30
FOR FETCH ONLY;

```

圖 6. 用來存取和播放視訊的應用程式 (1/2)

```

EXEC SQL OPEN c1;
for (;;) {
    EXEC SQL FETCH c1 INTO :hvClient, :hvCampaign,
                        :hvSdate, :hvVid_fname;

    if (SQLCODE != 0)
        break;

    printf("\nRecord %d:\n", ++count);
    printf("Client = '%s'\n", hvClient);
    printf("Campaign = '%s'\n", hvCampaign);
    printf("Sdate = '%s'\n", hvSdate);

    rc=DBvPlay(NULL,MMDB_PLAY_FILE,hvVid_fname,MMDB_PLAY_WAIT);
}
EXEC SQL CLOSE c1;

```

圖 6. 用來存取和播放視訊的應用程式 (2/2)

範例 2：依內容來搜尋影像

廣告代理商的插畫員正在設計某位客戶的新印刷廣告。插畫員想要在廣告背景中使用特定藍色陰影，並且想瞭解代理商先前建立的印刷廣告是否曾使用該顏色。插畫員的作法是執行應用程式，依內容去搜尋影像。影像儲存在資料庫表格中 (請參閱第 44 頁的圖 4)。該應用程式要求使用者提供一個視覺範例，亦即可用來示範指定顏色的影像。然後應用程式分析範例中的顏色，並尋找顏色與範例最相近的影像。

圖 7顯示一個視覺範例，以及最接近其顏色的擷取影像。

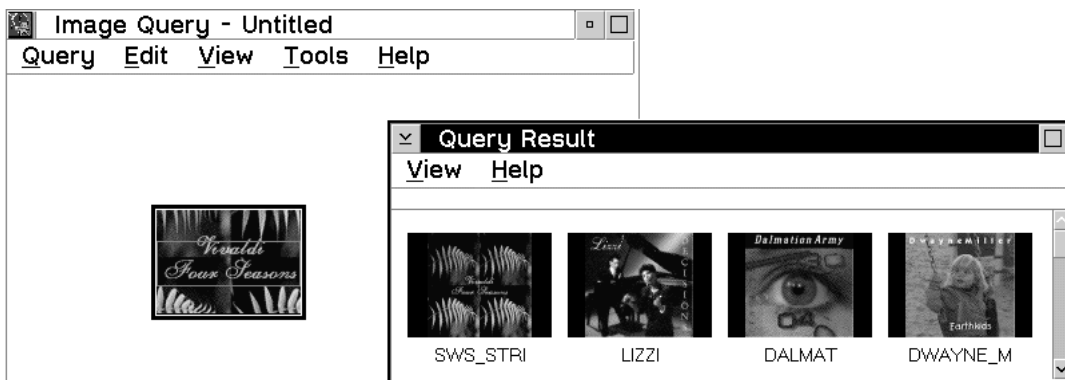


圖 7. 依內容來搜尋影像。視覺範例的用途是依平均色來搜尋影像。

第 48 頁的圖 8 顯示該應用程式的一些重要元素。請注意，該應用程式使用一個稱為 `QbQueryCreate` 的 QBIC API 來建立 QBIC 查詢、使用 `QbQueryAddFeature` 和 `QbQuerySetFeatureData` 來將顏色選擇新增至查詢、使用 `QbQuerySearch` 來發出查詢、及使用 `QbQueryDelete` 來刪除查詢。應用程式也使用一個稱為 `DBiBrowse` 的圖形 API 來顯示所擷取的影像。

```

#include <dmbqbqpi.h>

#define    MaxQueryReturns    10

static    SQLHENV    henv;
static    SQLHDBC    hdbc;
static    SQLHSTMT    hstmt;
static    SQLRETURN    rc;

void main(int argc, char* argv[])
{
    char        line[4000];
    char*        handles[MaxQueryReturns];
    QbQueryHandle    qhandle=0;
    QbResult    results[MaxQueryReturns];
    SQLINTEGER    count;
    SQLINTEGER    resultType=qbiArray;

    SQLAllocEnv(&henv);
    SQLAllocConnect(henv, &hdbc);
    rc = SQLConnect(hdbc, (SQLCHAR*)"qtest", SQL_NTS,
                    (SQLCHAR*)"", SQL_NTS, (SQLCHAR*)"", SQL_NTS);

    if (argc !=2) {
        printf("usage: query colorname\n");
        exit(1);
    }

    QbImageSource is;
        is.type = qbiSource_AverageColor;

    /* run the get color subroutine */
    getColor(argv[1], is.average.Color);

    QbQueryCreate(&qhandle);
    QbQueryAddFeature(qhandle, "QbColorFeatureClass");
    QbQuerySetFeatureData(qhandle, "QbColorFeatureClass",&is);
    QbQuerySearch(qhandle, "ADS", "ADS_IMAGE", 10, 0, resultType
                  &count, results);
    for (int j = 0; j <count; j++) {
        printf(j,":\n");

        DBiBrowse("usr/local/bin/xv %s", MMDB_PLAY_HANDLE, handles[j],
                  MMDB_PLAY_WAIT);
    }
}

```

圖 8. 依內容來搜尋影像的應用程式 (1/2)

```

QbQueryDelete(qhandle);

SQLDisconnect(hdbc);
SQLFreeConnect(hdbc);
SQLFreeEnv(henv);
}

```

圖 8. 依內容來搜尋影像的應用程式 (2/2)

作業環境

DB2 Extender 版本 7 在主/從環境中使用 DB2 Universal Database 版本 7.1 (或以上版本)。支援的平台所需的最低版本和版次與 DB2 Universal Database 版本 7.1 相同。

支援的從屬站平台如下：OS/2、AIX、Windows NT[®] 及更新版本、Windows 95、Windows 98、Solaris 作業環境及 HP-UX。

支援的伺服器如下：OS/2、AIX、Windows NT 及更新版本、Solaris 作業環境及 HP-UX。

第 50 頁的圖 9 顯示支援的平台。

DB2 Universal Database for z/OS Extender 也是屬於 DB2 Extender 產品，可支援 z/OS 從屬站和伺服器。如需 DB2 Universal Database for z/OS Extender 的相關資訊，請參閱 *DB2 Universal Database for z/OS Image, Audio, and Video Extenders Administration and Programming* 或 *DB2 Universal Database for z/OS Text Extender Administration and Programming*。

DB2 Extender 可以在單一分割區資料庫環境中作業。

僅適用於 EEE：在下列平台上，Extender 也可以在多重分割區資料庫環境中作業：AIX、Solaris 作業環境及 Windows NT (以上版本)。

DB2 Extender 必須與 DB2 Universal Database Enterprise-Extended Edition 一起使用，以便在多重分割區資料庫環境中作業。

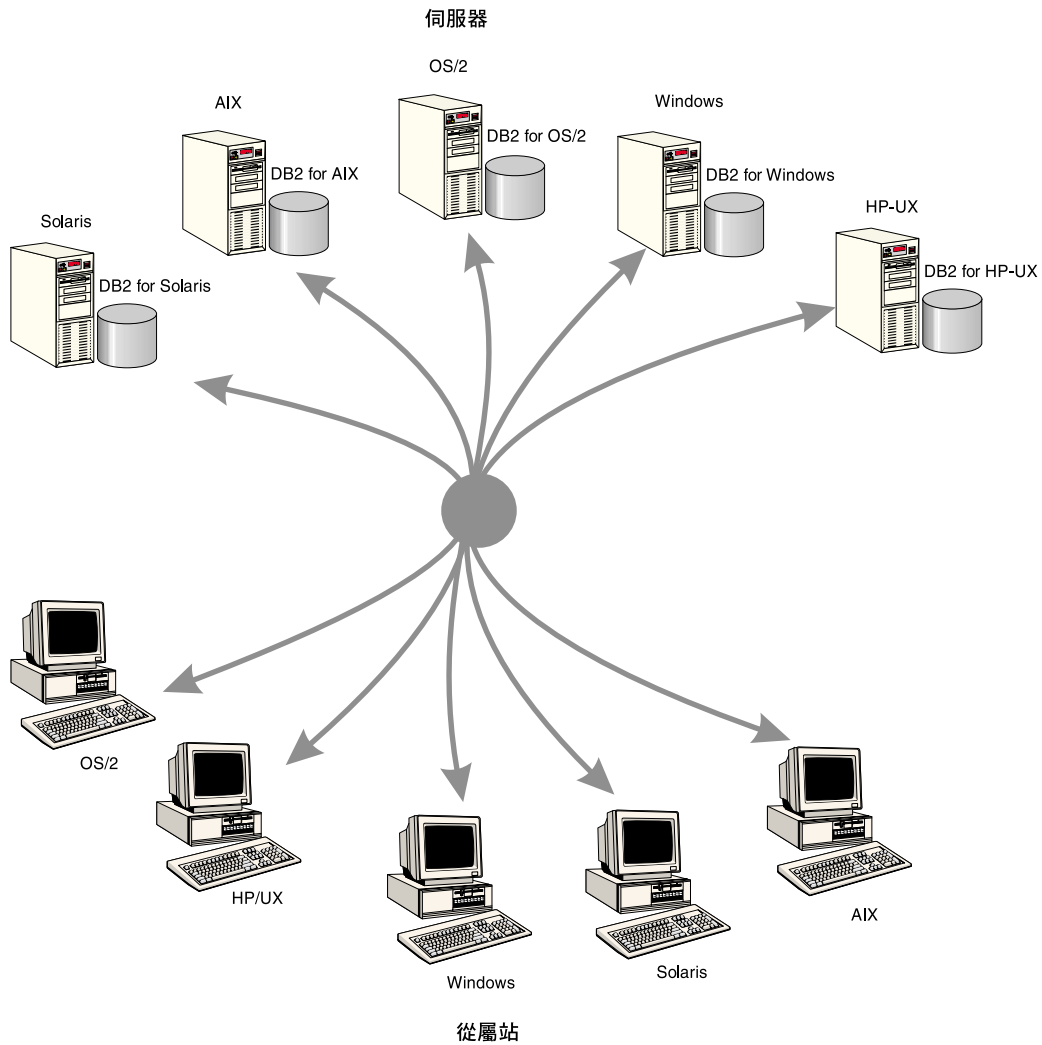


圖 9. DB2 Extender 平台

第 3 章 DB2 Extender 概念

本章說明在使用 DB2 Extender 之前，您需要了解的觀念。

主題	請參閱
物件導向觀念	第 51 頁
Extender 資料結構	第 55 頁
已分割資料庫概念	第 59 頁
Extender 安全性和復原	第 61 頁

關於物件導向概念的其它資訊，請參閱 *DB2 DB2 Application Development Guide*。

物件導向觀念

DB2 支援物件導向,此觀念指任何事物，不論實際或抽象，皆可在應用程式中以「物件」來表示，而物件是由一組作業和資料值組成。例如，文件可以使用文件物件來代表，該物件由文件資料和在文件上執行的作業所組成，這些作業包括歸檔、傳送及列印。視訊片段可以由視訊物件來代表，該物件由視訊資料和作業組成，這些作業包括播放視訊片段或尋找特定視訊框。如同真實世界一般，代表性物件亦有屬性。例如，可以指定屬性給視訊物件，例如壓縮類型和取樣頻率。

物件可以分類。相同類型的物件，擁有相同的屬性和行為，亦即因相同的作業而相關聯。例如，若一種視訊類型有定義壓縮類型屬性，則該視訊類型的所有物件皆有該屬性。若某視訊類型的一個物件能夠播放，則該視訊類型的所有物件皆可播放。

DB2 對物件導向的支援，容許您在表格直欄中儲存物件類型的案例，且在 SQL 陳述式中使用函數來操作。例如，您可以在表格直欄中儲存視訊物件，並使用 SQL 函數來操作。此外，還可以在應用程式之間，共用這些物件的屬性和行為。所有應用程式對於相同的物件類型，『視為』具有同組的屬性和行為。

視訊物件通常很大且複雜。影像和音效物件亦然。就因為支援物件導向，DB2 容許您在資料庫儲存大型物件 (LOB)。也提供您利用使用者定義類型 (UDT)、使用者定義函數 (UDF) 及觸發函式，來定義和操縱 LOB。

大型物件

DB2 容許您在資料庫將**大型物件 (LOB)** 儲存為下列物件：

- 二進位大型物件 (BLOB)
- 字元大型物件 (CLOB)
- 雙位元組大型物件 (DBCLOB)

BLOB 是二進位字串。影像、音效及視訊物件，在資料庫中儲存為 **BLOB**。**CLOB** 是由相關字碼頁的單一位元組字元所組成之字串。這種資料類型使用於含有單一位元組字元的文字物件。**DBCLOB** 是由相關字碼頁的雙位元組字元所組成之字串。這種資料類型使用於含有雙位元組字元的文字物件。

雖然每一個 **LOB** 的長度可達 2 GB；但 DB2 仍容許每一個表格中有許多 **LOB** 直欄。每一列可儲存多達 24 GB **LOB** 空間，每一個表格可儲存多達 4 TB **LOB** 空間。

因為大小的關係，**LOB** 的內容不直接儲存在使用者表格。取而代之，表格中的每一個 **LOB** 是由大型物件描述子來識別。描述子用來存取磁碟上其它位置的大型物件。

DB2 Extender 提供更多彈性，可讓您在檔案中保留 **LOB** 的內容，再從資料庫中指向它。使用 DB2 Extender 來儲存物件時，要完成這項指定。

使用者定義類型

影像、音效和視訊物件以 **BLOB** 形式出現在資料庫。**使用者定義類型 (UDT)** 亦稱為**特殊類型**，提供方法來區別 **BLOB**。例如，可為影像物件建立一個 **UDT**，而為音效物件建立另一個 **UDT**。雖然儲存為 **BLOB**，影像和音效物件仍被視為不同於 **BLOB** 的特殊類型，且彼此亦互不相同。

您可以使用 **SQL CREATE DISTINCT TYPE** 陳述式來建立 **UDT**。例如，假設您正在發展的應用程式，是用來處理地圖上的地理特性。您可以建立特殊類型 **map**，來代表地圖物件，如下所示：

```
CREATE DISTINCT TYPE map AS BLOB (1M)
```

地圖類型的物件在內部是以 1 MB 的 **BLOB** 來代表，但被視為特殊類型的物件。

您可以使用 **UDT** (如同 **SQL** 內建類型) 來說明表格直欄中儲存的資料。下列範例中，使用可保留地圖類型資料的直欄來建立表格：

```
CREATE TABLE places  
  (locid    INTEGER NOT NULL,  
   location CHAR (50),  
   grid     map)
```


每一個 DB2 Extender 為其類型建立 UDT，亦即影像、音效及視訊三種類型。

使用者定義函數

使用者定義函數 (UDF) 是建立 SQL 函數的一種方式，所以加入 DB2 提供的內建函數集內。特殊情況下，您可以為影像、音效及視訊物件建立 UDF，以執行專用作業。例如，建立 UDF 來取得視訊的壓縮格式，或傳回音效的取樣率。如此一來，即可定義特殊類型的物件之行爲。例如，視訊物件的行爲是根據視訊類型的函數，影像物件的行爲是根據影像類型的函數。

您可以使用 SQL CREATE FUNCTION 陳述式來建立 UDF。該陳述式特別指定 UDF 所適用的資料類型。例如，下列陳述式建立一個 UDF `map_scale`，用來計算地圖的比例尺。請注意，該 UDF 視 `map` 為可用的資料類型。施行函數的程式碼是以 C 寫成，並在 EXTERNAL NAME 子句中識別：

```
CREATE FUNCTION map_scale (map)
  RETURNS SMALLINT
  EXTERNAL NAME 'scale!map'
  LANGUAGE C
  PARAMETER STYLE DB2SQL
  NO SQL
  DETERMINISTIC
  NO EXTERNAL ACTION
```

UDF 可使用於 SQL 陳述式中，如同內建函數一樣。下列範例中，`map_scale` UDF 使用於 SQL SELECT 陳述式中，來傳回表格直欄 `grid` 中儲存的地圖比例尺：

```
SELECT map_scale (grid)
  FROM places
 WHERE location='SAN JOSE, CALIFORNIA'
```

每一個 DB2 Extender 為其類型建立一組 UDT，亦即影像、音效及視訊三種類型的 UDF。您可在 SQL 陳述式中使用這些 UDF 來要求 Extender 函數，例如在表格中儲存影像、取得視訊框掃描率、或新增音效註解。

UDF 和 UDT 名稱

DB2 函數的全名是 *schema-name.function-name*，其中 *schema-name* 是一個識別字，提供以邏輯分組的一組 SQL 物件。DB2 Extender UDF 的綱目名稱是 MMDBSYS。這個 MMDBSYS 綱目名稱也是 DB2 Extender UDT 的限定元。

您可依照喜好，對 UDF 或 UDT 使用完整名稱。例如，MMDBSYS.CONTENT 表示 UDF 的綱目名稱為 MMDBSYS，函數名稱為 CONTENT。MMDBSYS.DB2IMAGE 表示 UDT 的綱目為 MMDBSYS，而特殊類型名稱為 DB2IMAGE。在參照 UDF 或 UDT 時，您也可以省略綱目名稱；在此情況中，DB2 會使用函數路徑，判斷您要的函數或特殊資料類型。

函數路徑

函數名稱為已排序的綱目名稱清單。DB2 使用清單中綱目名稱的次序，來分辨指向函數和特殊資料類型的參照。您可以透過指定 SQL 陳述式 SET CURRENT FUNCTION PATH，來指定函數路徑。這樣可在 CURRENT FUNCTION PATH 特別暫存區中設定函數路徑。

就 DB2 Extender 而言，建議您將 mmdbsys 綱目新增至函數路徑中。這樣可讓您輸入 DB2 Extender UDF 和 UDT 名稱而不必在它們前面加上 mmdbsys。以下是將 mmdbsys 綱目新增至函數路徑的範例：

```
SET CURRENT FUNCTION PATH = mmdbsys, CURRENT FUNCTION PATH
```

若以 mmdbsys 身份登入，請勿新增 mmdbsys 作為函數路徑中的第一個綱目：若使用 mmdbsys 使用者 ID 登入，函數路徑中的第一個綱目會設定為 mmdbsys。若嘗試以 SET CURRENT FUNCTION PATH 陳述式，將函數路徑中的第一個綱目設定為 mmdbsys，則函數路徑會從兩個 mmdbsys 綱目開始 -- 這是一個錯誤狀況。

超載函數名稱

函數名稱可以**超載**。這表示即使在相同綱目中，多個 UDF 仍可共用相同名稱。但是，兩個函數不可有相同的**標記**。標記是指完整的函數名稱結合已定義資料類型的所有函數參數。

觸發函式

觸發函式可定義表格變動時所啟動的一組動作。您可使用觸發函式執行一些動作，例如驗證輸入資料，為新插入的橫列自動產生一值，讀取其它表格以便交互參照，或寫入其它表格以便審核。通常使用觸發函式檢查完整性或強制執行商務規則。

您可使用 SQL CREATE TRIGGER 陳述式來建立觸發函式。下列陳述式建立一個觸發函式，來強制有關零件庫存的商業規則。當現有的零件數量少於最大庫存量的 10% 時，該觸發函式會再訂購零件。

```
CREATE TRIGGER reorder
  AFTER UPDATE OF on_hand, max_stocked ON parts
  REFERENCING NEW AS n_row
  FOR EACH ROW MODE DB2SQL

  WHEN (n_row.on_hand < 0.10 * n_row.max_stocked)
  BEGIN ATOMIC
    VALUES(issue_ship_request(n_row.max_stocked -
                                n_row.on_hand,
                                n_row.partno));
  END
```

DB2 Extender 建立和維護管理支援表格，以記錄關於儲存於 資料庫的影像、音效和視訊的資訊。(有關這些表格的詳細資訊，請參閱『管理支援表』。) 在資料庫中插入、更新或刪除影像、音效或視訊資料時，Extender 會使用觸發函式更新這些表格。

Extender 資料結構

Image, Audio, and Video Extenders 建立及使用管理支援表格和控點以儲存及存取影像、音效和視訊資料。Image Extender 也建立及使用 QBIC 型錄，根據內容來存取影像。Video Extender 也使用索引檔和照片型錄來存取關於視訊中場景變更的資訊。

管理支援表

管理支援表格，亦稱為 meta 資料表格，包含必要資訊讓 Extender 處理影像、音效及視訊物件上的使用者要求。支援管理表格中的資訊，通常是指『meta 資料』。

如第 56 頁的圖 10 所示，有些管理支援表格會識別可用於 Extender 的使用者表格和直欄。這些表格參照其它管理支援表格，這些支援表格保留已啟用直欄中物件的屬性資訊。在這些表格中，Extender 會為特定 Extender 定義的資料類型之專用屬性，以及 Extender 資料類型之間的共用屬性，維護其相關的資訊。舉例而言，Image Extender 維護影像中有關寬度、高度及顏色數量的資訊，以及有關影像、音效及視訊物件之共用屬性的資訊，例如識別在資料庫中匯入物件的人，或最後更新物件的人。

對 BLOB 格式的物件，管理支援表格也可包含其內容。另外，物件可儲存在檔案中，再由管理支援表格來參照。例如，視訊片段在管理支援表格中可儲存為 BLOB，或保存在表格所參照的檔案內。

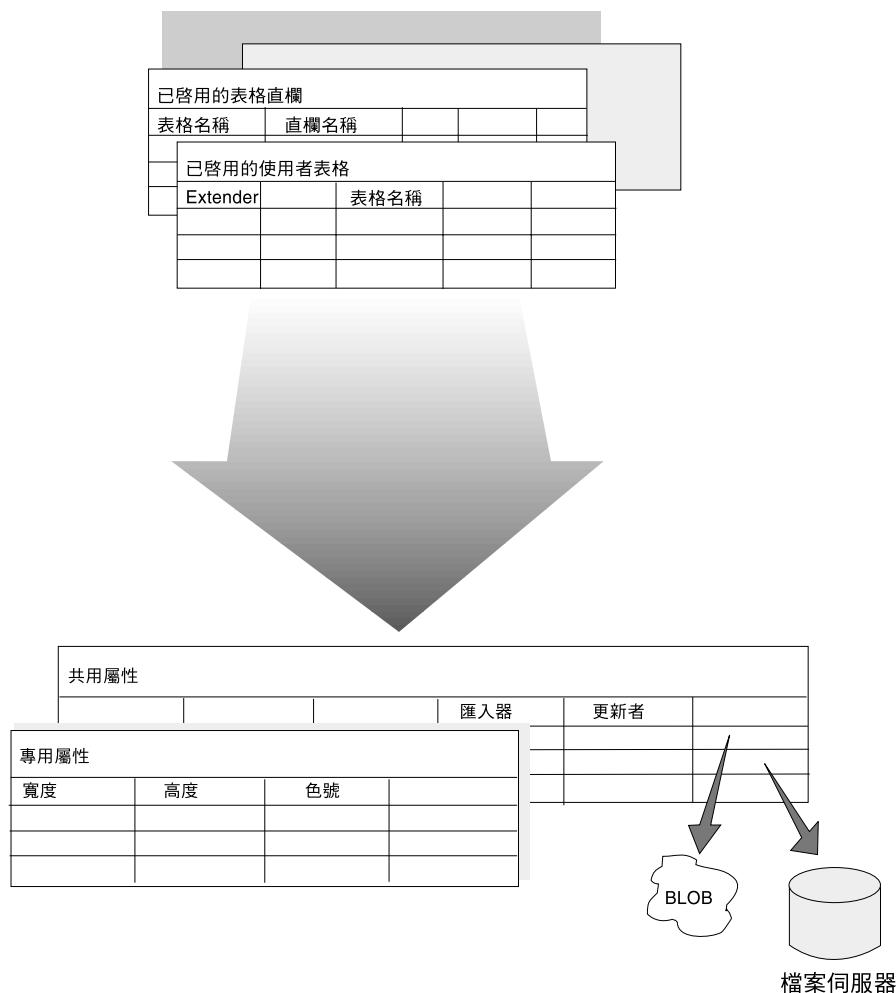


圖 10. 管理支援表格

控點

當您在使用者表格內儲存影像、音效或視訊物件時，物件並不真正儲存於該表格中。相反地，Extender 建立一個稱為 **handle** 的字串來代表物件，並將控點儲存在表格中。Extender 將物件儲存在管理支援表格中，或者，若您把檔案內容保存在檔案內，則 Extender 將檔案識別字儲存在管理支援表格中。它也將物件的屬性和控點儲存在管理支援表格內。如此，Extender 即可鏈結使用者表格中的控點和管理

支援表格中的物件資訊。圖 11 說明在使用者表格中儲存兩個影像的資訊。

使用者表格

ID	名稱	圖片
		Handle 1
		Handle 2

管理支援表



圖 11. Handles

QBIC 型錄

QBIC 型錄是一個檔案集，可保留有關影像視覺特性的資料。Image Extender 使用此資料來根據內容搜尋影像。

對於依內容搜尋的使用者表格，您可以為其中每一欄的影像建立一個 QBIC 型錄。建立 QBIC 型錄時，要識別 Image Extender 所分析、儲存及後來查詢資料的特性。您也可以先在 QBIC 型錄建立之後，再新增或捨棄特性。

QBIC 型錄可以保留有關下列影像特性的資料：

平均色

影像中所有圖點的顏色值總計，再除以影像中的圖點數量。(圖點是影像中的最小元素，可指定顏色和亮度。)例如，若影像的一半是藍色圖點，另一半是紅色圖點，則影像的平均色值是紫色。平均色用來搜尋具有主色的影像。若影像內含主色，則平均色會接近該主色。

直方圖色	對照 64 色譜，來測量影像中的顏色分配。對於 64 色中的每一種顏色，直方圖色可識別影像中該顏色的所佔圖點百分比。例如，影像的直方圖色可能是 40% 白色圖點、50% 藍色、及 10% 紅色；影像中無其他圖點是柱狀色譜中的其餘顏色。直方圖色用來搜尋具有各種不同顏的影像。
按位顏色	影像中特定區域的圖點平均色值。例如，影像的右上角顯示一個發亮的黃色太陽，則此區影像的按位顏色是亮黃色。按鈕顏色用來搜尋特定區內具有主色的影像。
結構	測量影像的粗細、對比及方向性。粗細表示影像中重覆項目的大小 (例如，水晶對石頭)。對比識別影像中的亮度差 (淡對暗)。方向性指出影像中是由方向控制 (例如柵欄的垂直方向) 或不受方向控制 (例如沙灘影像)。結構用來搜尋具有特殊型樣的影像。

要讓影像能夠以內容來搜尋，就先要對影像進行編目。編目影像時，Image Extender 會計算影像的特性值來分析影像，並將這些值儲存在 QBIC 型錄中。

按照內容搜尋影像時，您的查詢會識別搜尋的一個或多個特性 (如平均顏色)、每一個特性的來源 (如範例影像) 和型錄影像目標集。Image Extender 計算來源的特性值，並與目標影像的特性值做比較。然後計算得分，以指出目標影像的特性值與來源的相似程度。

您可以令 Image Extender 傳回特性最接近來源的影像。Image Extender 會傳回每一個影像的控點和影像分數。您也可使 Image Extender 只傳回單一影像的分數。

視訊索引

視訊索引是一個檔案，供 Video Extender 用於視訊片段中尋找特定場景或框。

Video Extender 可偵測視訊中的場景變更。**場景變更**是指視訊片段中的一個點，兩個連續框在此點上有著明顯的不同。例如，在記錄影像時若照相機變更焦點，則發生上述狀況。兩個場景變更之間的框，構成一個**快照**。

您可以使用 Video Extender 的場景偵測功能，在視訊片段中尋找快照、或個別的框。方法是 Extender 需要快照或框的索引資訊。此項索引資訊儲存於**索引檔**中。

快照型錄

快照型錄用來儲存視訊片段中有關快照的資料。快照型錄可儲存於資料庫或檔案中。

檔案中儲存的快照型錄，包含下列有關快照的資料：

- 快照型錄檔名稱

- 控制 Video Extender 偵測快照方法的一些值，例如一個快照中的最小框數
- 一些值控制著儲存多少框和哪些框，來代表快照的框。
- 快照號碼
- 起始框號
- 結束框號
- 代表框號碼
- 包含代表框內容的檔名

您可以存取快照型錄檔中的資料，或存取資料庫中快照型錄的概略表。此概略表包含下列快照相關資料的直欄：

- 快照控點
- 視訊表格名稱
- 視訊直欄名稱
- 視訊控點
- 視訊檔名
- 起始框號
- 結束框號
- 代表框號碼
- 代表框資料
- 註解

已分割資料庫概念 (僅適用於 EEE)

DB2 Extender 可搭配 DB2 Extended Enterprise Edition 使用，好處是可以利用 DB2 Extended Enterprise Edition 所提供的分割的資料庫支援。

已分割資料庫是在兩台以上的機器之間分散的一種資料庫。對一般使用者和應用程式發展者而言，該資料庫是單一機器上的單一資料庫。分割容許應用程式有效地使用一個資料庫，只因爲該資料庫太大而無法由一台機器來處理。

已分割資料庫由兩個以上的分割區組成。每一個分割區是由其所屬**資料庫分割區伺服器**來管理。資料庫分割區伺服器包含資料庫管理程式，及其所管理的資料和系統資源集成。一般而言，會對每一個機器指定一個資料庫分割區伺服器。但是，單一機器上也可能有多個資料庫分割區伺服器。每一個資料庫分割區伺服器，會保留整個資料庫的一部份。資料庫分割區伺服器有時亦稱爲**節點**。

已分割資料庫概念

如圖 12 所示，資料庫分割區可依邏輯方式分組並指定名稱。每一個組資料庫分割區稱為一個**節點群組**。定義節點群組容許您限制應用程式查詢已選定的資料庫分割區，以便加速異動時間。一個節點群組可以包含一個或多個資料庫分割區。若一個節點群組包含多個資料庫分割區，則稱為**多重分割區節點群組**。屬於一個多重分割區節點群組的所有資料庫分割區，必須位於相同的資料庫內。

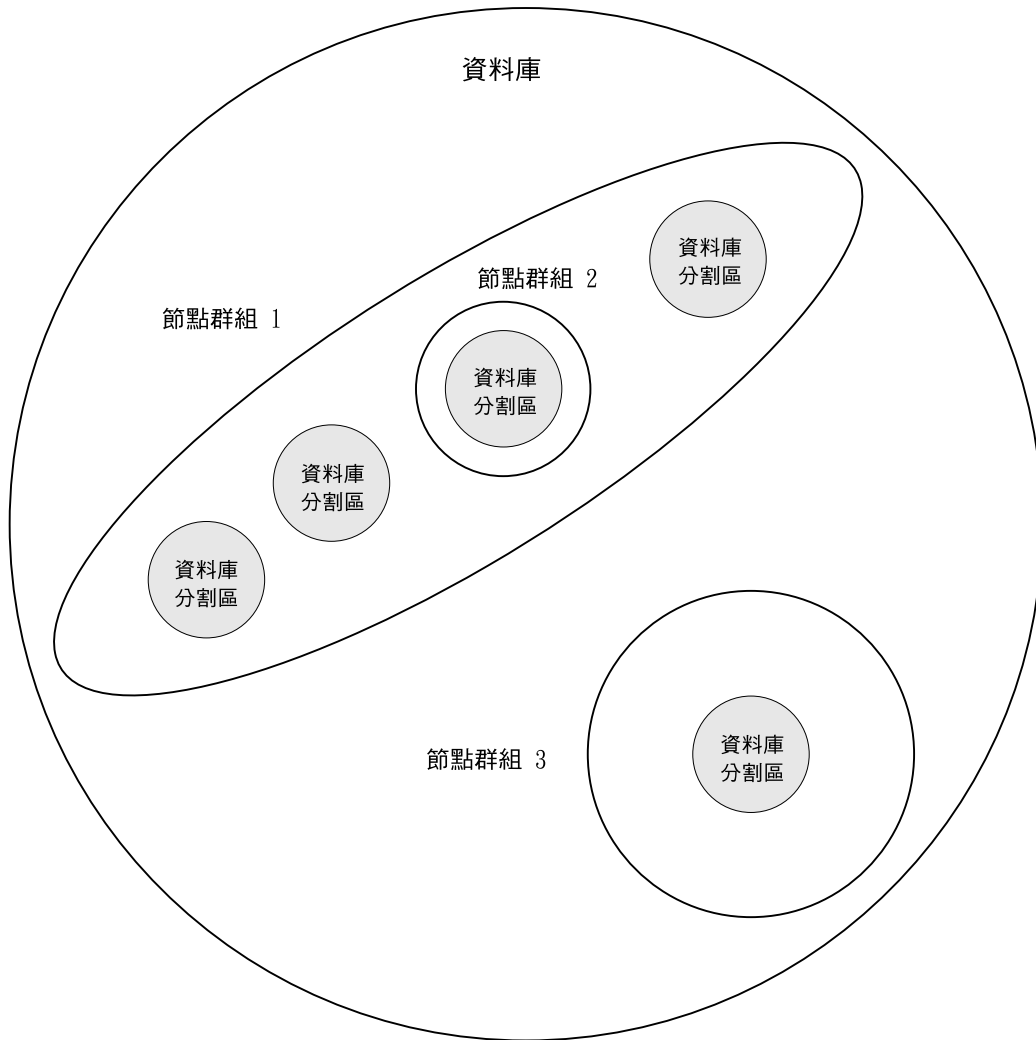


圖 12. 資料庫中的節點群組

在已分割資料庫系統中使用 Extender 的用途如下：

- 在多重分割之間分送資料，以減少輸入/輸出和處理瓶頸。
- 新增更多機器並重新分配資料，以增加您的資料庫大小。

平行處理

已分割資料庫可使用多 CPU 來滿足資訊要求。在每一台機器的資料庫分割區伺服器上，擷取和更新要求會自動地配置到後續要求內，並同步執行。

就已分割資料庫系統的強大處理能力而言，假設您有一億筆記錄要在單一分割區資料庫上掃描。這項掃描需要單一資料庫管理程式去搜尋一億筆記錄。現在假設這些記錄平均地分散在 20 個以上的資料庫分割區伺服器上；而每個資料庫管理程式只能掃描五百萬個記錄。若每一個資料庫管理程式同時且同速掃描，則完成掃描所需的時間，大約是單一分割區系統處理這項工作的 5%。

可調整性

當您的資料庫逐漸變大時，您可以新增資料庫分割區伺服器到資料庫系統中，以改善效能，這個程序稱為**調整**資料庫系統。

調整資料庫時，在您新增資料庫分割區伺服器後，它會再新增資料庫分割區給資料庫系統中的每一個現存資料庫。然後，您即可指派新的資料庫分割區至該資料庫的現存節點群組。最後，您可以重新分配該節點群組內的資料，以啓用這個新的資料庫分割區。

在分割的資料庫環境中使用 DB2 Extender

在分割的資料庫環境中使用 DB2 Extender，可讓您利用有效支援 LOB 操作的特性。大型儲存庫的 LOB (每一個的長度可達 2 GB) 可儲存於資料庫中，因為一個資料庫可分散於許多機器上。

而且，DB2 Extender 亦參與平行處理 DB2 Extended Enterprise Edition 所管理的 SQL 作業。當 DB2 Extended Enterprise Edition 平行執行查詢時，該查詢中的任何 DB2 Extender UDF 亦於個別資料庫分割區中平行執行。

安全性和復原

在資料庫中儲存為 BLOB 的影像、音效及視訊物件，如同傳統數字和字元資料一樣，有著相同的安全性和復原保護。儲存於 meta 資料表格中有關這些物件的資訊亦然。使用者必須擁有必要的專用權，才能選取、插入或更新物件。

使用者發出 UDF 以便從使用者表格選取、插入、更新或刪除物件。若要執行要求的作業，UDF 必須能夠存取 (如果需要，還要更新) 保留物件屬性資訊的管理支援表格。若使用者對使用者表格有適當專用權，Extender 可讓 UDF 對管理支援表格執行這些作業。

有些 Extender 相關管理作業需要 DBADM 權限。關於 DB2 Extender 管理 API 需要的權限，請參閱 第 251 頁的第 14 章，『應用程式設計介面』。關於 DB2 Extender 管理命令需要的權限，請參閱 第 469 頁的第 15 章，『從屬站的管理命令』。

若影像、音效或視訊的內容儲存於資料庫所參照的檔案內，則物件的 meta 資料由 DB2 來保護。檔案必須在 PUBLIC (亦即所有使用者) 可讀取的目錄。

BLOB 和 meta 資料可以備份和回復，方法與 DB2 中的其它資料相同。檔案中儲存的物件內容，可以使用非 DB2 工具來備份和回復。另外，可使用非 DB 工具備份和回復 QBIC 型錄和視訊索引。關於備份 QBIC 型錄的資訊，請參閱第 150 頁。關於備份視訊索引的資訊，請參閱 page 29。

第 4 章 Extender 的作業方式

DB2 Extender 的用途良多，幫忙您處理影像、音效及視訊資料的要求。揭開 Extender 作業方式的一個好辦法，是在使用時檢驗其作為。本章說明一個實務範例，包括 Image 和 Audio Extender。其中討論使用者執行的操作，以及 Extender 的回應方式。

Extender 實務範例

有一家公司的人事部門，想要建立一個人事資料庫 (使用 DB2 for AIX)，其中包括每一位員工的圖片。

含有照片的資料庫： 如圖 13 所示，資料庫中的員工表格，包含每一位員工的識別、名字及圖片。

ID	名稱	圖片
128557	Anita Jones	
843962	Robert Smith	

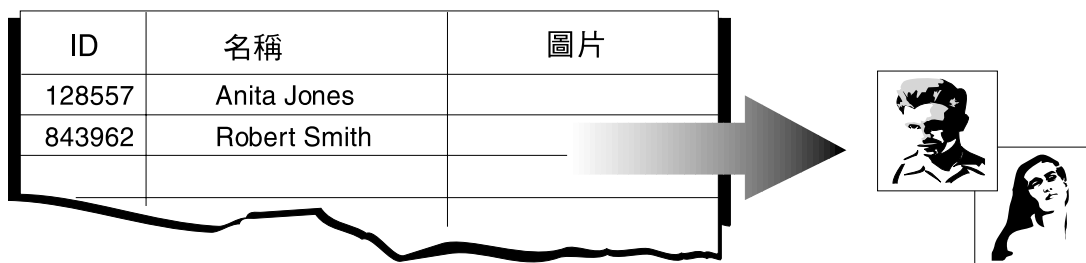
A diagram illustrating the relationship between a database table and employee photos. On the left is a table with three columns: 'ID', '名稱' (Name), and '圖片' (Image). The first two rows contain data: '128557' and 'Anita Jones', and '843962' and 'Robert Smith'. A large, thick, grey arrow points from the '圖片' column of the table towards two small, square-framed black and white photographs of a man and a woman, representing the images stored in the database.

圖 13. 員工表格

若要準備人事資料庫來處理影像，一位具有 SYSADM 權限的系統管理者，應從啓動 Extender 服務程式來著手。然後，系統管理者建立並啓用資料庫，供 Image Extender 使用。

資料庫管理員 (DBA) 或同等權限的人員，建立並啓用員工表格，並建立員工圖片直欄，以供 Image Extender 使用。

含有音效的資料庫： 備妥影像處理所用的人事資料庫和員工表格之後，人事部門決定將每一位員工的錄音新增至表格內。請參閱第 64 頁的圖 14。

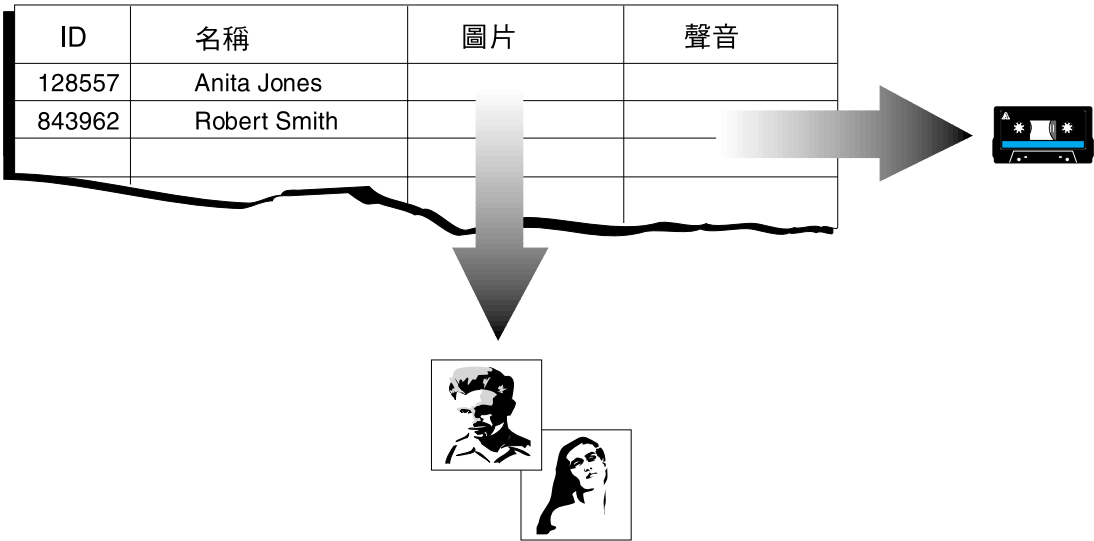


圖 14. 加入音效直欄的員工表格

經由新增直欄並啓用資料庫、表格及直欄供 Audio Extender 來使用，系統管理者即可變更表格。

然後，人事部門的使用者即可在表格上插入資料、選取和顯示資料、更新資料及刪除資料。

啓動 Extender 服務程式

Extender 使用伺服器中的服務程式，當做其作業的一部份。若這些服務程式尚未成為伺服器正常『啓動』作業的功能，則由系統管理者來啓動。

系統管理者的作法：系統管理者以案例擁有者的身份來登入 AIX 伺服器。然後，系統管理者在伺服器上發出下列命令：

```
DMBSTART
```

結果：Extender 服務程式啓動，供伺服器上的 Extender 案例使用。DMBSTART 命令亦啓動一個尚未執行的 DB2 案例。

準備資料庫

系統管理者建立並啓用人事資料庫，供 Image Extender 使用。

系統管理者的作法：系統管理者在 DB2 for AIX 中，使用下列 SQL 陳述式來建立人事資料庫：

```
CREATE DATABASE personnel          /*name of the database*/
      ON /persdb                   /*name of the database directory*/
      WITH "Personnel database"    /*comment*/
```

系統管理者連接並啓用資料庫，供 Image Extender 使用。系統管理者使用 dv2ext 命令行處理器，發出下列命令：

```
CONNECT TO personnel
ENABLE DATABASE FOR DB2IMAGE
```

結果： 爲了回應 ENABLE DATABASE 命令，Image Extender 的動作如下：

- 建立使用者定義類型，稱爲 DB2IMAGE，代表影像物件。
- 建立影像物件的管理支援表格。
- 建立影像物件的使用者定義函數。UDF 列示於 表 4。

表 4. Image Extender 建立的使用者定義函數

UDF 名稱	說明
Comment	取得或更新使用者註解
Content	取得或更新影像的內容
DB2Image	儲存影像的內容
Filename	取得含有影像的檔名
Format	取得影像格式 (例如 GIF)
Height	取得影像的圖點高度
Importer	取得影像匯入者的使用者 ID
ImportTime	取得影像匯入時的時間戳記
NumColors	取得影像中所用的顏色數目
QbScoreFromName	取得影像的相似性分數 (使用指定的查詢)
QbScoreFromStr	取得影像的相似性分數 (使用查詢字串)
QbScoreTBFromName	取得一欄影像的相似性分數表 (使用指定查詢)
QbScoreTBFromStr	取得一欄影像的相似性分數表 (使用查詢字串)
Replace	更新影像的內容和使用者的註解
Size	取得影像的位元組大小
Thumbnail	取得縮小版的影像
Updater	取得影像更新者的使用者 ID
UpdateTime	取得影像更新時的時間戳記
Width	取得影像的圖點寬度

準備表格

DBA 建立並啓用員工表格，並建立圖片直欄，以供 Image Extender 使用。

DBA 的作法： 爲了方便起見，DBA 使用下列 SQL 陳述式，將 mmdbsys 綱目新增至現行函數路徑：

```
SET CURRENT FUNCTION PATH = mmdbsys, CURRENT FUNCTION PATH
```

這容許指定 UDT 和 UDF 名稱時，不需要前置 mmdbsys 綱目名稱。(mmdbsys 綱目不一定是函數路徑中的第一個綱目。) 有關 UDT 和 UDF 名稱的詳細資訊，請參閱 第 53 頁的『UDF 和 UDT 名稱』。

DBA 建立員工表格。DBA 可使用 DB2 命令行處理器發出下列 SQL 陳述式：

```
CREATE TABLE    employee          /*name of the table*/  
  (id            CHAR(6)           /*employee identification*/  
   name          VARCHAR(40)       /*employee name*/  
   picture       DB2IMAGE)         /*employee picture*/
```

然後，DBA 使用 db2ext 命令行處理器，來發出下列命令：

```
ENABLE TABLE employee FOR DB2IMAGE  
ENABLE COLUMN employee picture FOR DB2IMAGE
```

結果：爲了回應 ENABLE TABLE 命令，Image Extender 的動作如下：

- 定義員工表格。
- 建立管理支援表，在啓用的直欄內保留影像物件的屬性資訊。

爲了回應 ENABLE COLUMN 命令，Image Extender 的作法如下：

- 識別所用使用的圖片直欄。
- 建立觸發函式。這些觸發函式會更新各種管理支援表格，以回應在員工表格上的插入、更新及刪除作業。

變更表格

DBA 將一個音效直欄新增至員工表格中，並啓用給 Audio Extender 使用。

DBA 的作法： DBA使用 db2ext 命令行處理器，啓用人事資料庫，供 Audio Extender 使用：

```
ENABLE DATABASE FOR DB2AUDIO
```

然後，DBA 發出下列 SQL 陳述式，變更員工表格。DBA 使用 DB2 命令行處理器，發出 SQL 陳述式。

```
ALTER TABLE employee          /*name of the table*/
      ADD voice DB2AUDIO        /*employee audio recording*/
```

DBA 使用 db2ext 命令行處理器，來啓用員工表格和啓用直欄，以供 Audio Extender 使用：

```
ENABLE TABLE employee FOR DB2AUDIO
ENABLE COLUMN employee voice FOR DB2AUDIO
```

結果： 爲了回應 ENABLE DATABASE 命令，Audio Extender 的動作如下：

- 建立使用者定義類型，稱爲 DB2AUDIO，代表音效物件。
- 建立音效物件的管理支援表格。
- 建立音效物件的使用者定義函數。UDF 列示於 表 5。

表 5. Audio Extender 建立的使用者定義函數

UDF 名稱	說明
AlignValue	取得每一個音效範例值的位元組
BitsPerSample	取得用來代表音效的位元數
BytesPerSec	取得每秒音效的平均位元組數
Comment	取得或更新使用者註解
Content	取得或更新音效的內容
DB2Audio	儲存音效的內容
Duration	取得音效的播放時間
Filename	取得含有音效的檔名
FindInstrument	取得磁軌的數目，用來記錄特定樂器的音效
FindTrackName	取得錄音中指定磁軌的磁軌號碼
Format	取得音效格式
GetInstruments	取得錄音的樂器名稱
GetTrackNames	取得音效中的磁軌名稱
Importer	取得音效匯入者的使用者 ID
ImportTime	取得音效匯入時的時間戳記
NumAudioTracks	取得音效中記錄磁軌的數目
NumChannels	取得聲道數目
Replace	更新錄音的內容和使用者的註解
SamplingRate	取得音效的取樣率
Size	取得音效的位元組大小
TicksPerQNote	取得錄音時每四分音符的節拍數。
TicksPerSec	取得錄音時每秒的節拍數。

表 5. *Audio Extender* 建立的使用者定義函數 (繼續)

UDF 名稱	說明
Updater	取得音效更新者的使用者 ID
UpdateTime	取得音效更新時的時間戳記

爲了回應 `ENABLE TABLE` 命令，*Audio Extender* 的作法如下：

- 定義員工表格。
- 建立管理支援表，在啓用的直欄內保留音效物件的屬性資訊。

爲了回應 `ENABLE COLUMN` 命令，*Audio Extender* 的作法如下：

- 識別所用使用的聲音直欄。
- 建立觸發函式。這些觸發函式會更新各種管理支援表格，以回應在員工表格上的插入、更新及刪除作業。

將資料插入表格中

使用者將一筆 *Anita Jones* 記錄插入員工表格中。該記錄包括 *Anita* 的識別 (128557)、名字、圖片及錄音。來源影像和音效內容位於伺服器的檔案中。影像在表格中儲存爲 `BLOB`；音效內容保留在伺服器檔案中 (表格項目參照伺服器檔案)。

使用者的作法：使用者利用應用程式來將記錄插入員工表格中，該應用程式包括的陳述式請參閱 第 69 頁的圖 15。


```

EXEC SQL BEGIN DECLARE SECTION;
long hvInt_Stor;
long hvExt_Stor;
EXEC SQL END DECLARE SECTION;

hvInt_Stor = MMDB_STORAGE_TYPE_INTERNAL;
hvExt_Stor = MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',                                /*id*/
    'Anita Jones',                          /*name*/
    DB2IMAGE(                               /*Image Extender UDF*/
        CURRENT SERVER,                    /*database server name in*/
                                           /*CURRENT SERVER register*/
        '/employee/images/ajones.bmp'      /*image source file*/
        'ASIS',                            /*keep the image format*/
        :hvInt_Stor,                       /*store image in DB as BLOB*/
        'Anita's picture'),               /*comment*/
    DB2AUDIO(                               /*Audio Extender UDF*/
        CURRENT SERVER,                    /*database server name in*/
                                           /*CURRENT SERVER register*/
        '/employee/sounds/ajones.wav',     /*audio source file*/
        'WAVE',                            /* audio format */
        :hvExt_Stor,                       /*retain content in server file*/
        'Anita's voice')                  /*comment*/
    );

```

圖 15. 將資料插入表格中

結果為了回應 INSERT 陳述式中的 DB2Image UDF，Image Extender的作法如下：

- 從來源影像檔標頭中，讀取影像的屬性，例如高度、寬度及顏色數目。
- 建立影像的專用控點，並記錄於管理支援表格中：
 - 影像的控點
 - 時間戳記
 - 影像的位元組大小
 - 註解 『Anita 的圖片』
 - 影像的內容

影像來源在檔名為 ajones.bmp 的伺服器檔案中。該檔案的內容以 BLOB 的形式，被插入至管理支援表格記錄中。影像儲存的格式和來源影像相同，不需任何轉換。

- 在管理支援表格中儲存一筆記錄。該記錄包含影像類型的屬性，例如顏色數目和影像的縮圖。

為了回應 INSERT 陳述式中的 DB2Audio UDF，Audio Extender 的作法如下：

- 從音效檔標頭中，讀取音效的屬性，例如磁軌和聲道數目。

插入資料

- 建立音效專用的控點
- 在管理支援表格中儲存一筆記錄。記錄包含：
 - 音效的控點
 - 時間戳記
 - 音效的位元組大小
 - 註解 『Anita 的聲音』

音效內容位於一個稱為 ajones.wav 的伺服器檔案中；管理支援表格記錄參照此檔案。

- 在管理支援表格中儲存另一筆記錄。該記錄包含音效類型的屬性，例如音效的取樣率。

觸發函式將影像和音效屬性插入各種管理支援表格內。

從表格選取資料

使用者擷取有關 Robert Smith 的影像和聲音目前在員工表格內的儲存情形。

使用者的作法： 使用者取得資訊是經由含有 SQL 陳述式 (如 圖 16 所示) 的應用程式。

```
EXEC SQL BEGIN DECLARE SECTION;
char[255] hvImg_Time;
char[255] hvAud_Time;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT IMPORTTIME(PICTURE),           /*when image was stored*/
               IMPORTTIME(VOICE)              /*when audio was stored*/
               INTO :hvImg_Time, :hvAud_Time
               FROM EMPLOYEE
               WHERE NAME='Robert Smith';
```

圖 16. 從表格選取資料

結果：為了回應圖片直欄的 ImportTime UDF，Image Extender 會傳回一個時間戳記，其中包含儲存影像的日期和時間。為了回應 VOICE 直欄的 ImportTime UDF，Audio Extender 會傳回一個時間戳記，其中包含儲存聲音的日期和時間。

顯示和播放物件

使用者在工作站從屬站上，顯示 Robert Smith 的影像，並播放 Robert Smith 的錄音。影像以 BLOB 的形式儲存在員工表格中；聲音的內容錄製在伺服器檔案中。

使用者的作法： 使用者顯示影像及播放錄音的方法，是經由含有 SQL 陳述式的應用程式，SQL 陳述式顯示於圖 17。

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_hdl [251];
char hvAud_hdl [251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT PICTURE,                               /*Get image handle*/
                VOICE                                   /*Get audio handle*/
INTO :hvImg_hdl, :hvAud_hdl
FROM EMPLOYEE
WHERE NAME='Robert Smith';

rc=DBiBrowse(
    NULL,                                              /*Use default image browser*/
    MMDB_PLAY_HANDLE,                                /*Use handle*/
    hvImg_hdl,                                        /*Image handle*/
    MMDB_PLAY_NO_WAIT);                              /*Run browser independently*/

rc=DBaPlay(
    NULL,                                              /*Use default audio player*/
    MMDB_PLAY_HANDLE,                                /*Use handle*/
    hvAud_hdl,                                        /*Audio handle*/
    MMDB_PLAY_WAIT);                                  /*Wait for player to end*/
                                                    /*before continuing*/
```

圖 17. 顯示和播放物件

結果： DB2 擷取 Robert Smith 的影像和錄音之控點。然後，為了回應 DBiBrowse API，Image Extender 使用所擷取的影像控點，來取得影像內容。Image Extender 從資料庫中擷取擷取內容，並放入暫時從屬站檔案中，供影像瀏覽程式來顯示。NULL 參數表示透過使用者系統的預設影像瀏覽程式。瀏覽程式獨立於所呼叫的程式之外，表示所呼叫的程式不會等待影像瀏覽程式完成之後才繼續執行。

為了回應 DBaPlay API，Audio Extender 取得與所擷取控點相關的音效檔名，並將該檔名傳遞給音效播放程式。NULL 參數表示透過使用者系統的預設音效播放程式。所呼叫的程式在繼續執行之前，會先等待使用者結束音效播放程式。

更新表格中的資料

Anita Jones 以最新的圖片來取代她在員工表格內的圖片。新圖片的內容位於伺服器檔案中。

使用者的作法： 使用者利用應用程式來取代員工表格內的圖片，該應用程式包括的 SQL 陳述式請參閱 第 72 頁的圖 18。

更新資料

```
EXEC SQL BEGIN DECLARE SECTION;
char hvComment[16385];
long hvStorageType;
EXEC SQL END DECLARE SECTION;

strcpy(hvComment, "Picture taken at Anita's promotion");
hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

EXEC SQL UPDATE EMPLOYEE
SET PICTURE=REPLACE(
    PICTURE,                                /*image handle*/
    '/myimages/newone.bmp',                /*source image content*/
    'BMP',                                  /*source format*/
    :hvStorageType,                        /*store image in table as BLOB*/
    :hvComment)                            /*replace comment*/
WHERE NAME='Anita Jones';
```

圖 18. 更新表格中的資料

結果：爲了回應 UPDATE 陳述式中的 Replace UDF，Image Extender 讀取新影像的屬性。Image Extender 使用新影像的屬性，更新管理支援表格內的舊影像屬性。影像來源在檔名爲 newone.bmp 的伺服器檔案中。該檔案的內容以 BLOB 的形式，被插入至管理支援表格記錄中，並取代舊影像的 BLOB 內容。

觸發函式取代各種管理支援表格內的影像屬性資料。

從表格中刪除資料

使用者從員工表格中刪除 Anita Jones 的記錄。

使用者的作法：使用者使用應用程式來刪除員工表格中的記錄，該應用程式包括下列的 SQL 陳述式：

```
DELETE FROM EMPLOYEE
WHERE NAME='Anita Jones';
```

結果：觸發函式會刪除 Anita Jones 在各種管理支援表格內的項目。

第 2 篇 管理影像、音效和視訊資料

第 5 章 管理概觀 75

可以使用 DB2 Extender 來執行的管理作業 . . 75

第 6 章 準備 Extender 資料的資料物件 . . 79

啓用資料庫 79

範例. 79

啓用表格 82

啓用直欄 85

停用資料物件. 86

第 7 章 追蹤資料物件和媒體檔案. 87

檢查資料物件的狀態 87

尋找用來參照檔案的表格項目 88

尋找表格項目所參照的檔案 89

檢查媒體檔案是否存在. 90

第 8 章 授與及取消管理支援表格的專用權. . 91

第 5 章 管理概觀

本章提供建立使用 DB2 Extender 的應用程式時的相關管理作業概觀。

DB2 Extender 提供兩種方式來執行大部份的管理作業：

- 管理應用程式設計介面 (API)。您可將 DB2 Extender API 併入 C 語言程式。關於這些 API 的參考資訊，請參閱第 251 頁的第 14 章，『應用程式設計介面』。
- 管理命令。您可向 db2ext 命令行處理器提出管理命令。DB2 命令行不會執行這些命令。關於輸入管理命令的指示和其它的參考資料，請參閱第 469 頁的第 15 章，『從屬站的管理命令』。

可以使用 DB2 Extender 來執行的管理作業

管理作業可分五類：

- 管理 Extender 服務程式。DB2 Extender 是在 DB2 上自己的伺服器上執行。應用程式使用 Extender 資料之前，系統管理者必須啟動 Extender 服務程式，而且使用者要先連接到保留 Extender 資料的資料庫。
- 準備 Extender 資料的資料物件。您可準備資料庫、表格和直欄，並啓用它們來保留 Extender 資料。啓用資料物件時，Extender 會建立和維護管理支援表格 (亦稱為 meta 資料表格)，來管理 Extender 資料。
- 僅適用於 EEE。在已分割環境中重新分配 Extender 資料。在已分割資料庫中新增或捨棄分割區時，您可以重新分配資料來利用新節點架構。
- 追蹤資料物件和媒體檔案。對使用 DB2 Extender 的應用程式進行除錯時，若能瞭解哪些資料物件會為 Extender 資料啓用，將會很有幫助。瞭解使用者表格和外部媒體檔的相互關係也很有幫助。
- 清除管理支援表格。使用 DB2 Extender 時，作廢的項目最後會堆積在管理支援表格內。刪除作廢的 meta 資料，可以改善效能和收回儲存空間。

第 76 頁的表 6 列出管理 Extender 資料時的所有作業。此表格指定提供執行每個作業的工具，以及可尋找更多資訊的位置。

在 **Extender API** 直欄中，x 代表每一個 API 陳述式的第三個字元。此字元會隨您使用的 Extender 而不同：

字元	Extender
a	音效
i	影像
v	視訊

例如，啓用表格供影像資料使用的 API 是 DBiEnableTable，啓用表格供音效使用的 API 是 DBaEnableTable，啓用表格供視訊使用的 API 是 DBvEnableTable。Extender API 中的 No 值表示作業沒有 Extender API。Extender Command 直欄中的 No 值表示作業沒有 Extender 命令。

QBIC 需要額外管理：若您想使用 Image Extender 的 Query by Image Content (QBIC) 功能，則需要執行額外的管理作業，例如建立 QBIC 型錄。有關這些作業的資訊，請參閱第 147 頁的第 12 章，『依內容來查詢影像』。

表 6. DB2 Extender 的管理作業和機能

作業	Extender API	Extender 命令	請參閱
管理 Extender 服務程式			
啓動 Extender 服務程式	否	DMBSTART	3 頁
取得 Extender 服務程式的狀態	否	DMBSTAT	5 頁
停止 Extender 服務程式	否	DMBSTOP	5 頁
連接資料庫	否	CONNECT	3 頁
啓動資料庫的 Extender 服務程式	否	START SERVER	5 頁
取得資料庫的 Extender 服務程式之狀態	否	GET SERVER STATUS	5 頁
停止資料庫的 Extender 服務程式	否	STOP SERVER	5 頁
建立與管理 Extender 案例	否	DMBICRT、DMBILIST、DMBIDROP、DMBIMIGR	6 頁
準備多媒體資料的資料物件			
啓用資料庫	DBxEnableDatabase	ENABLE DATABASE	79 頁
停用資料庫	DBxDisableDatabase	DISABLE DATABASE	86 頁
啓用表格	DBxEnableTable	ENABLE TABLE	82 頁
停用表格	DBxDisableTable	DISABLE TABLE	86 頁
啓用直欄	DBxEnableColumn	ENABLE COLUMN	85 頁
停用直欄	DBxDisableColumn	DISABLE COLUMN	86 頁

表 6. DB2 Extender 的管理作業和機能 (繼續)

作業	Extender API	Extender 命令	請參閱
在已分割環境中重新分配 Extender 資料 (僅適用於 EEE)			
根據新的節點群組架構來重新分配 Extender 資料	DMBRedistribute	REDISTRIBUTE NODEGROUP	19 頁
追蹤資料物件和媒體檔案			
判斷資料庫是否啟用	DBxIsDatabaseEnabled	GET EXTENDER STATUS	87 頁
判斷表格是否啟用	DBxIsTableEnabled	GET EXTENDER STATUS	87 頁
判斷直欄是否啟用	DBxIsColumnEnabled	GET EXTENDER STATUS	87 頁
在限定元為現行使用者 ID 的表格中，尋找用來參照檔案的表格項目。	DBxIsFileReferenced	否	88 頁
在特定限定元的所有表格中或資料庫的所有表格中，尋找參照檔案的表格項目。	DBxAdminIsFileReferenced	否	88 頁
尋找在限定元為現行使用者 ID 的表格中，表格項目所參照的檔案。	DBxGetReferencedFiles	GET REFERENCED FILES	89 頁
在特定限定元的所有表格中或資料庫的所有表格中，尋找表格項目所參照的檔案。	DBxAdminGetReferencedFiles	GET REFERENCED FILES	89 頁
在限定元為現行使用者 ID 的所有表格中，尋找表格項目所參照但無法存取的檔案。	DBxGetInaccessibleFiles	GET INACCESSIBLE FILES	90 頁
在特定限定元的所有表格中或資料庫的所有表格中，尋找表格項目所參照但無法存取的檔案。	DBxAdminGetInaccessibleFiles	GET INACCESSIBLE FILES	90 頁
清除管理支援 (meta 資料) 表格			
為特定使用者表格，或限定元為現行使用者 ID 的所有使用者表格，清除 meta 資料表格。	DBxReorgMetadata	REORG	8 頁

表 6. DB2 Extender 的管理作業和機能 (繼續)

作業	Extender API	Extender 命令	請參閱
為特定限定元的使用者表格，或資料庫中的所有使用者表格，清除 meta 資料表格。	DBxAdminReorgMetadata	REORG	8 頁

管理作業的順序：下列清單是您首次使用 Extender 時，執行的管理作業之已排序摘要。請使用 DB2 命令或陳述式執行某些作業。使用 DB2 Extender 執行其它作業。此順序假設您的 DB2 系統正在執行。

必要的作業：

- 1. 啟動 Extender 服務程式。
- 2. 建立資料庫 (使用 DB2)。
- 3. 連接資料庫資料庫伺服器。
- 4. 啟用資料庫。
- 5. 建立表格和直欄 (使用 DB2)。
- 6. 啟用資料庫中的表格。
- 7. 啟用表格中的直欄。

可選用的作業：

- 1. 追蹤資料物件和媒體檔案。
- 2. 設定函數使用 (使用 DB2)。
- 3. 清除管理支援表格。

範例： 以下五章的大部份範例都假設系統管理者 (SYSADM) 或資料庫管理員 (DBA) 在執行這些作業。少數作業不需要 DBA 或 SYSADM 權限。

這些範例假設 DBA 已在現行函數路徑中新增 MMDBSYS 綱目。這容許 DBA 指定 UDT 名稱時，不需要以 MMDBSYS 綱目名稱為開頭。有關 UDT 名稱的詳細資訊，請參閱第 53 頁的『UDF 和 UDT 名稱』。

本節的許多 API 範例是以 Extender 附帶的範例應用程式碼為基礎。範例程式碼放在從屬站的 SAMPLES 子目錄內。

第 6 章 準備 Extender 資料的資料物件

您可準備資料庫、表格和直欄，並啓用它們來保留 Extender 資料。首先請啓用資料庫。然後啓用資料庫中的表格。最後，在表格中啓用直欄。

當您在資料物件中不再需要 Extender 資料時，可以停用這些物件。

您可以使用以 C 語言程式撰寫的 API 或從 db2ext 命令行，來啓用和停用物件。本章提供每一種方法的範例。

啓用資料庫

使用 DBxEnableDatabase API (其中 x 若是 a 代表音效，i 代表影像，或 v 代表視訊) 或使用 ENABLE DATABASE 命令，來啓用 DB2 Extender 的資料庫。

啓用資料庫時，Extender 會：

- 爲您的資料物件建立使用者定義類型 (UDT)，稱爲 DB2xxxxx，其中 xxxxx 是指影像、音效或視訊。UDT 是用來定義使用者表格中的直欄，以保留該類型物件的控點。
- 建立資料庫的管理支援表格 (亦稱爲中間資料表格)。這些表格不是使用者表格 (使用者儲存商業資料的表格)。Extender 使用它們管理 Extender 資料。請勿以手動方式來編輯它們。
- 建立與 Extender 有關的使用者定義函數 (UDF)。UDF 列示於 第 187 頁的『使用者定義函數』中。

啓用資料庫時，您必須指定表格空間來保留資料庫的管理支援表格 (及其索引)。指定的一或多個表格空間可以是 NULL 值，而在該情況中會使用預設表格空間。

您需要 DBA 權限才能啓用資料庫。

僅適用於 EEE：在已分割環境中啓用 Extender 的資料庫時，您所指定的表格空間應該定義在節點群組中，而該節點群組包括已分割資料庫系統中的所有節點。而且，指定的表格空間應該與使用者表格位於相同的節點群組中。

範例

下列範例中，會使用預設表格空間，讓資料庫來保留影像資料。

使用 API：圖 19 中的程式碼在啓用之前，會連接到現存的資料庫。此範例是以 DB2 CLI (DB2 call level interface) 來撰寫。包括一些設置和錯誤檢查碼。完整範例程式位於 SAMPLES 子目錄中的 ENABLE.C 檔。

```
/*---- Set-up -----*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "dmbimage.h"      /* image extender function prototypes (DBi) */
#include "utility.h"       /* utility functions */

#define MMDB_ERROR_MSG_TEXT_LEN      1000
#define SERVER_IS_DB2390 (strcmp(dbms, "DB2")==0 || strcmp(dbms, "DSN06010")==0)

int
main(int argc, char *argv[])
{
    SQLHENV henv = SQL_NULL_HENV;
    SQLHDBC hdbc = SQL_NULL_HDBC;
    SQLHSTMT hstmt = SQL_NULL_HSTMT;
    SQLCHAR uid[18+1];
    SQLCHAR pwd[30+1];
    SQLCHAR dbname[SQL_MAX_DSN_LENGTH+1];
    SQLCHAR buffer[500];
    SQL SMALLINT dbms_sz = 0;
    char dbms[20];

    SQLRETURN rc = SQL_SUCCESS;
    SQLINTEGER sqlcode = 0;
    char errorMsgText[MMDB_ERROR_MSG_TEXT_LEN+1];
    char *program = "enable";
    char *step;
```

圖 19. 啓用資料庫的範例程式碼 (1/3)

```

/*----- Prompt for database name, userid, and password -----*/
if (argc > 5) || (argc >= 2 && strcmp(argv[1], "?") == 0)
{
    printf("Syntax for enable - enabling a DB2 UDB database: \n"
           "    enable database_name userid password\n");
    exit(0);
}

if (argc == 4) {
    strcpy((char *)dbname, argv[1]);
    strcpy((char *)uid, argv[2]);
    strcpy((char *)pwd, argv[3]);
}
else {
    printf("Enter database name:\n");
    gets((char *) dbName);
    printf("Enter userid:\n");
    gets((char *) uid);
    printf("Enter password:\n");
    gets((char *) pwd);
}

/*----- connect to the database -----*/
rc = cliInitialize(&henv, &hdbc, dbname, uid, pwd);
cliCheckError(henv, hdbc, SQL_NULL_HSTMT, rc);
if (rc < 0) goto SERROR;

/*----- find out if application is connected to DB2/UDB or DB2/390?-----*/
rc = SQLGetInfo(hdbc, SQL_DBMS_NAME, (SQLPOINTER) &dbms,
               sizeof(dbms), &dbms_sz);
cliCheckError(henv, hdbc, SQL_NULL_HSTMT, rc);
if (rc < 0) goto SERROR;

```

圖 19. 啓用資料庫的範例程式碼 (2/3)

```
/****** enable server for image extender *****/
if (!SERVER_IS_DB2390)
{
    printf("%s: Enabling database.....\n", program);
}
printf("%s: This may take a few minutes, please wait.....\n", program);

if (!SERVER_IS_DB2390)
{
    step="DBiEnableDatabase with NULL tablespace"
    rc=DBiEnableDatabase(NULL);
}
if (rc < 0) {
    printf ("%s: %s failed!\n", program, step);
    printMsg(rc);
    DBiGetError(&sqlcode, errorMsgText);
    if (sqlcode)
        printf("sqlcode=%i, ",sqlcode);
}
else if (rc > 0) {
    printf("%s: %s, warning detected.\n", program, step);
    printMsg(rc);
    DBiGetError(&sqlcode, errorMsgText);
    printf("warning MsgText=%s\n", errorMsgText);
}
else
    printf("%s: %s OK\n", program, step);
/****** end of enable server *****/
```

圖 19. 啓用資料庫的範例程式碼 (3/3)

使用 **db2ext** 命令行： 在本例中，已連接資料庫。

enable database for db2image

啓用表格

使用 DBxEnableTable API (其中 x 若是 a 代表音效，i 代表影像，或 v 代表視訊) 或 ENABLE TABLE 命令，以啓用 DB2 Extender 的表格。

啓用使用者表格時，您也必須指定表格空間來保留隨附的管理支援表格 (及其索引)。 指定的一或多個表格空間可以是 NULL 值，而在該情況中會使用預設表格空間。

僅適用於 EEE： 在已分割環境中啓用 Extender 的表格時，您指定的表格空間應定義於一個節點群組，而該節點群組包含已分割資料庫系統中的所有節點。 而且，指定的表格空間必須與使用者表格位於相同的節點群組中。

僅適用於 EEE： 在已分割資料庫環境中，DB2 Extender 直欄不能作為分割鍵值直欄使用。

對於使用者表格，您需要「控制」(Control) 或「變更」(Alter) 權限。在資料庫中啓用表格之前，必須先啓用資料庫。

下列範例中，會使用預設表格空間，讓表格來保留影像資料。 資料庫已啓用。

使用 API： 在第 84 頁的圖 20 中， 啓用表格之前，程式碼會建立表格和 COMMIT 變更。 該範例包括一些錯誤檢查碼。 完整範例程式位於 SAMPLES 子目錄中的 ENABLE.C 檔。

啓用表格

```
SQLCHAR szCreate_DB2UDB[]="CREATE TABLE %s(%s mmdbsys.DB2Image,
%s mmdbsys.DB2Video, %s mmdbsys.DB2Audio, artist varchar(25), title varchar(25)
stock_no char(11), tw char(10), price char(10))";

SQLRETURN rc = SQL_SUCCESS;
SQLINTEGER sqlcode = 0;
char errorMsgText[MMDDB_ERROR_MSG_TEXT_LEN+1];
char tableName[8+18+1] = "sobay_catalog";
char audioColumn[18+1] = "music";
char imageColumn[18+1] = "covers";
char videoColumn[18+1] = "video";
char *program = "enable";
char *step;

/*-----create table -----*/
printf("%s: Creating table .....\\n", program);
if (!SERVER_IS_DB2390)
    sprintf((char*) buffer, (char*) szCreate_DB2UDB,
            tableName, imageColumn, videoColumn, audioColumn);

rc = SQLAllocStmt(hdbc, &hstmt);
cliCheckError(SQL_NULL_HENV, hdbc, SQL_NULL_HSTMT, rc);
rc = SQLExecDirect(hstmt, buffer, SQL_NTS);
cliCheckError(SQL_NULL_HENV, SQL_NULL_HDBC, hstmt, rc);

/*---- enable table for image extender -----*/
printf("%s: Enabling table.....\\n", program);
step="DBiEnableTable";
if (!SERVER_IS_DB2390)
    rc = DBiEnableTable(NULL, tableName);
}
if (rc < 0) {
    printf("%s: %s failed!\\n", program, step);
    printMsg(rc);
    DBiGetError(&sqlcode, errorMsgText);
    if (sqlcode)
        printf("sqlcode=%i, \"sqlcode\");
    printf("errorMsgText=%s\\n", errorMsgText);
} else if (rc > 0) {
    printf("%s: %s, warning detected.\\n", program, step);
    printMsg(rc);
    DBiGetError(&sqlcode, errorMsgText);
    printf("warningMsgText=%s\\n", errorMsgText);
} else
    printf("%s: %s OK\\n", program, step)
/*---- end of enable table -----*/
```

圖 20. 啓用表格的範例程式碼

使用 **db2ext** 命令行：在本範例中，表格已存在，而且已啓用資料庫。

enable table employee for db2image

啓用直欄

使用 DBxEnableColumn API (其中 x 若是 a 代表音效，i 代表影像，或 v 代表視訊) 或使用 ENABLE COLUMN 命令，以啓用 DB2 Extender 的直欄。發出 API 或命令時，請指定所屬的表格和直欄。

啓用直欄時，Extender 會新增資訊至屬於使用者表格的管理支援表格內。對於含有此直欄的使用者表格來說，您需要「控制」權限或「變更」權限。必須啓用資料庫和表格之後才能啓用直欄。

下列範例中，啓用員工表格中的圖片直欄來保留影像資料。資料庫和表格已經啓用。

使用 API：本範例包括一些錯誤檢查碼。完整範例程式位於 SAMPLES 子目錄中的 ENABLE.C 檔。

```
char imageColumn[18+1] = "covers";

/*---- enable column for image extender ----*/
printf("%s: Enabling columns.....\n", program);
step="DBiEnableColumn";
rc = DBiEnableColumn(tableName, imageColumn);
if (rc < 0) {
    printf("%s: %s failed!\n", program, step);
    printMsg(rc);
    DBiGetError(&sqlcode, errorMsgText);
    if (sqlcode)
        printf("sqlcode=%i, ", sqlcode);
    printf("errorMsgText=%s\n", errorMsgText)

} else if (rc > 0) {
    printf("%s: %s, warning detected.\n", program, step);
    printMsg(rc);
    DBiGetError(&sqlcode, errorMsgText);
    printf("warningMsgText=%s\n", errorMsgText);
} else
    printf("%s: %s OK\n", program, step);
/*---- enable column for image extender ----*/
```

圖 21. 啓用直欄的範例程式碼

使用 db2ext 命令行：在本範例中，此直欄已存在，而且已啓用資料庫和表格。

enable column employee picture for db2image

停用資料物件

若從資料庫、表格或直欄移除 Extender 資料，您就不必再啓用它們。您有兩種方法可停用資料物件：DISABLE 命令和 API。有關 Extender 命令的更多資訊，請參閱第 469 頁的第 15 章，『從屬站的管理命令』。有關 Extender API 的更多資訊，請參閱第 251 頁的第 14 章，『應用程式設計介面』。

在捨棄內含 Extender 資料的表格或資料庫之前，請先停用它，並停止該資料庫的伺服器。

第 7 章 追蹤資料物件和媒體檔案

針對使用 DB2 Extender 的應用程式加以建立或進行除錯時，若能瞭解哪些資料物件會為 Extender 資料啟用，將會很有幫助。例如，若您確定某個表格可用於影像資料，您的應用程式就可以順利地在該表格內儲存影像。

瞭解使用者表格和外部媒體檔的相互關係也很有幫助，例如，哪些表格參照特定檔案、或特定表格參照哪些檔案。瞭解表格是否參照不存在於系統的檔案也很有幫助。

您需要適當專用權：您必須擁有某表格的存取權才能在該表格中追蹤資料。若您想要執行大範圍的追蹤作業，例如要找出在資料庫中的所有使用者表格有哪些項目參照某檔案，則您需要的是所有被搜尋的使用者表格及相關的管理支援表格中，已啟用直欄的 SYSADM 權限、DBADM 權限、或 SELECT 專用權。若沒有這些表格的存取權，Extender 只會為您可存取的表格傳回追蹤資訊。它們也會傳回程式碼，表示您對一些必要的表格沒有存取權。

檢查資料物件的狀態

您可以來檢查是否已啟用資料庫、表格和直欄來保留 Extender 資料。下列範例可判斷是否已為 Image Extender 啟用現行資料庫。已連接資料庫。完整範例程式位於 SAMPLES 子目錄中的 API.C 檔。

使用 API：第 88 頁的圖 22 中的範例程式碼包括一些錯誤檢查碼。

檢查啓用性

```
/*---- Query the database using DBIsDatabaseEnabled API. -----*/
step="DBIsDatabaseEnabled API";
rc = DBIsDatabaseEnabled(&status);
if (rc < 0) {
    printf("%s: %s FAILED!\n", argv[0], step);
    printMsg(rc);
    DBGetError(&sqlcode, errorMsgText);
    printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
    fail = TRUE;
} else if (rc > 0) {
    printf("%s: %s, warning detected.\n", argv[0], step);
    printMsg(rc);
    DBGetError(&sqlcode, errorMsgText);
    printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
} else {
    if (status == 1) {
        printf("%s: \"%s\" database is enabled for Image Extender\n",
            argv[0], dbName);
        printf("%s: %s PASSED\n\n", argv[0], step);
    } else if (status == 0) {
        printf("%s: \"%s\" database is not enabled for Image Extender\n",
            argv[0], dbName);
        printf("%s: %s PASSED\n\n", argv[0], step);
    } else
        printf("%s: %s FAILED, invalid status!\n", argv[0], step);
}
```

圖 22. 檢查資料庫是否已啓用的範例程式碼

使用 db2ext 命令行：

get extender status

檢查使用者表格和直欄的狀態，類似檢查資料庫的狀態。使用 DBxIsTableEnabled 和 DBxIsColumnEnabled API，或 GET EXTENDER STATUS 命令。

尋找用來參照檔案的表格項目

您可檢查使用者表格中有哪些項目參照外部媒體檔。使用 DBxAdminIsFileReferenced API，來檢查現行資料庫的所有使用者表格或使用者表格子集中，有哪些項目參照外部媒體檔。使用 DBxIsFileReferenced API，來檢查特定使用者表格有哪些項目參照外部媒體檔。

使用 API：第 89 頁的圖 23 中的範例程式碼會傳回檔案被參照的次數及被參照的位置。該範例包括一些錯誤檢查碼。完整範例程式位於 SAMPLES 子目錄中的 API.C 檔。

```

/*---- Query the database using DBiAdminIsFileReferenced API. -----*/step="DBiAdminIsF
rc = DBiAdminIsFileReferenced((char*) uid, filename, &count, &filelist);
if (rc < 0) {
    printf("%s: %s FAILED!\n", program, step);
    printMsg(rc);
    DBiGetError(&sqlcode, errorMsgText);
    printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
} else if (rc > 0) {
    printf("%s: %s, warning detected.\n", program, step);
    printMsg(rc);
    DBiGetError(&sqlcode, errorMsgText);
    printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
} else {
    if (count == 0)
        printf("%s: \"%s\" file is not referenced\n",
            program, filename);
    else {
        printf("%s: \"%s\" file is referenced %d times\n",
            program, filename);
        for (i=0; i < count; i++)
        {
            /* filename is NULL for any IsFileReferenced APIs */

            printf ("filename = %s\n", filelist[i].filename);
            printf ("\tqualifier = %s\n", filelist[i].tqualifier);
            printf ("\tttable = %s\n", filelist[i].tname);
            printf ("\thandle = %s\n", filelist[i].handle);
            printf ("\tcolumn = %s\n", filelist[i].column);
            if (filelist[i].filename)
                free (filelist[i].filename);
        }
    }
    if (filelist)
        free (filelist);
    printf("%s: %s PASSED\n\n", argv[0], step);
}

```

圖 23. 檢查檔案是否由使用者表格來參照的範例程式碼

尋找表格項目所參照的檔案

使用 DBxAdminGetReferencedFiles API 或 GET REFERENCED FILES 命令，來列出現行資料庫中所有使用者表格或使用者表格子集所參照的外部媒體檔。使用 DBxGetReferencedFiles API 或 GET REFERENCED FILES 命令，來列出在特定表格中參照的外部媒體檔。

使用 API： 第 90 頁的圖 24 中的範例程式碼會傳回找到的檔案數目及檔案清單。完整範例程式位於 SAMPLES 子目錄中的 API.C 檔。

列出參照的檔案

```
/*---- Query the database using DBiAdminGetReferencedFiles API. -----*/
step="DBiAdminGetReferencedFilesAPI"
rc = DBiAdminGetReferencedFiles((char*) uid, &count, &filelist);
if (rc < 0) {
    printf("%s: %s FAILED!\n", program, step);
    printMsg(rc);
    DBiGetError(&sqlcode, errorMsgText);
    printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
} else if (rc > 0) {
    printf("%s: %s, warning detected.\n", program, step);
    printMsg(rc);
    DBiGetError(&sqlcode, errorMsgText);
    printf("sqlcode=%i, errorMsgText=%s\n", sqlcode, errorMsgText);
} else {
    if (count == 0)
        printf("%s: no referenced files\n", program);
    else {
        printf("%s: %d referenced files\n", program, count);
        for (i=0; i < count; i++)
        {
            printf ("filename = %s\n", filelist[i].filename);
            printf ("\tqualifier = %s\n", filelist[i].tqualifier);
            printf ("\tttable = %s\n", filelist[i].tname);
            printf ("\thandle = %s\n", filelist[i].handle);
            printf ("\tcolumn = %s\n", filelist[i].column);
            if (filelist[i].filename)
                free (filelist[i].filename);
        }
    }
    if (filelist)
        free (filelist);
    printf("%s: %s PASSED\n\n", argv[0], step);
}
```

圖 24. 可取得被參照的檔案清單的範例程式碼

使用 db2ext 命令行：

get referenced files user anitas for db2image

檢查媒體檔案是否存在

假設有人自系統刪除某媒體檔，但並未更新參照該檔案的使用者表格。您可能想要列出使用者表格所參照之所有無法存取的媒體檔案。

使用 DBxAdminGetInaccessibleFiles API 或 GET INACCESSIBLE FILES 命令，來列示現行資料庫中的所有使用者表格或使用者表格子集參照的無法存取的媒體檔。使用 DBxGetInaccessibleFiles API 或 GET INACCESSIBLE FILES 命令，來列出特定表格參照的無法存取的媒體檔。

第 8 章 授與及取消管理支援表格的專用權

使用者發出 UDF，以便從使用者表格選取、插入、更新或刪除影像、音效與視訊物件。若要執行要求的作業，UDF 必須能夠存取 (如果需要，還要插入、更新及刪除) 保留物件屬性資訊的管理支援表格。就使用者表格的擁有者而言，Extender 會將 UDF 處理要求的作業所需的存取權提供給 UDF。但表格擁有者以外的使用者如需從使用者表格中選取某物件，必須被授與對管理支援表格的選取專用權。

此外，在使用者表格影像物件上執行 QBIC 作業的使用者，必須要有適當的管理支援表格 (由那些物件的 QBIC 型錄構成) 專用權。例如，要對影像直欄發出 QBIC 查詢的使用者，必須針對影像直欄的 QBIC 型錄表格具有 SELECT 專用權。要變更 QBIC 型錄的使用者，必須具有相關 QBIC 型錄表格的 SELECT、INSERT、UPDATE 與 DELETE 專用權。

使用者表格的擁有者或資料庫的 DBA (具有 GRANT 專用權)，可使用 DB2 Extender 命令 GRANT 來授與管理支援表格的專用權。當您發出 GRANT 命令時，必須指定：

- 必要的專用權，如 SELECT 或 UPDATE。
- Extender 名稱：DB2IMAGE、DB2AUDIO 或 DB2VIDEO。您也可以為三個 Extender 指定 ALL。
- 使用者表格名稱。
- 使用者的 ID。您可以在使用者 ID 前加上選用的關鍵字 USER。您也可以為所有使用者指定 PUBLIC。

若您指定 SELECT，則會針對與使用者表格相關的指名 Extender，授與管理支援表格的 SELECT 專用權給指定的使用者。若您指定 DB2IMAGE，則也會針對與使用者表格相關的 QBIC 型錄，授與管理支援表格的 SELECT 專用權。例如，下列命令可針對與員工表格相關的 Image Extender，授與管理支援表格的 SELECT 專用權。使用者 ID ajones 會被授與專用權。此命令也可將員工表格相關的 QBIC 型錄之 SELECT 專用權授與使用者 ID ajones：

```
grant select for db2image on employee to ajones
```

下列命令可針對與員工表格相關的 Image、Audio 與 Video Extender，授與管理支援表格的 SELECT 專用權。所有的使用者都會被授與專用權。此命令也可將員工表格相關的 QBIC 型錄之 SELECT 專用權授與所有的使用者：

```
grant select for all on employee to public
```

就插入、更新或刪除等作業而言，Extender 會檢查以判定使用者是否具有使用者表格上必要的 INSERT、UPDATE 或 DELETE 專用權。若使用者具有必要的專用權，Extender 就會容許 UDF 依需求存取管理支援表格。

若要授與「QBIC 型錄」表格的 INSERT、UPDATE 與 DELETE 專用權，請以 GRANT 命令來指定 UPDATE 及 DB2Image。例如，下列命令可將與員工表格相關的「QBIC 型錄」表格的 INSERT、UPDATE 與 DELETE 專用權授與使用者 ID ajones：

```
grant update for db2image on employee to user ajones
```

當使用者不適合再存取使用者表格中的物件時，使用者表格的擁有者或資料庫的 DBA (具有 GRANT 專用權)，可取消使用者的管理支援表格 SELECT 專用權。這也包括由 QBIC 型錄構成的管理支援表格。使用 DB2 Extender 命令 REVOKE，以取消管理支援表格和 QBIC 型錄表格的專用權。REVOKE 的格式與 GRANT 命令相似。例如，下列命令可針對與員工表格相關的 Image Extender，取消管理支援表格的 SELECT 專用權。使用者 ID ajones 會被取消專用權。此命令也可取消員工表格相關的「QBIC 型錄」表格之 SELECT 專用權：

```
revoke select for db2image on employee from ajones
```

您也可以針對由 QBIC 型錄構成的管理支援表格，取消 INSERT、UPDATE 與 DELETE 專用權。使用 REVOKE 命令上的 UPDATE 參數。例如，下列命令可取消與員工表格相關的「QBIC 型錄」表格的 INSERT、UPDATE 與 DELETE 專用權。使用者 ID ajones 會被取消專用權。

```
revoke update for db2image on employee from ajones
```

在新增所有特性後，授與 QBIC 型錄的專用權：授與的管理支援表格 (由 QBIC 型錄構成) 專用權包括 QBIC 特性表格的專用權，但僅限於已新增至型錄中的特性。若您在授與型錄專用權後，又對型錄新增特性，則必須再次授與型錄的專用權。因此，在型錄建立完成，且所有特性皆新增完畢後，才是授與 QBIC 型錄專用權的正確時機。

第 3 篇 影像、音效和視訊資料的程式設計

第 9 章 程式設計概觀	95	更新影像、音效或視訊物件	126
使用 Extender UDF 和 API	95	用於更新的 Content UDF 格式	127
您可以使用 Extender UDF 和 API 來執行的作業	96	用於更新的 Replace UDF 格式	129
Extender 範例的範例表格	96	更新從屬站的物件	132
DB2 Extender 程式設計的事前準備	97	更新伺服器的物件	133
併入 Extender 定義	99	指定進行更新的資料庫或檔案儲存體	134
指定 UDF 和 UDT 名稱	100	識別更新的格式	135
傳輸大型物件	100	識別更新的格式；不進行轉換	135
若物件是在表格和伺服器檔案之間傳輸	100	識別更新的格式及轉換選項；進行格式轉換	135
若物件是在從屬站緩衝區中來回傳輸	101	更新具有使用者提供之屬性的物件	136
使用 LOB 定位器	101	更新縮圖 (影像與視訊)	137
若物件是在從屬站檔案中來回傳輸	102	更新註解	138
傳輸物件時指定檔名	103		
處理回覆碼	103	第 11 章 顯示或執行影像、音效或視訊物件	141
Unicode 支援	104	使用顯示或播放 API	141
		識別顯示或播放程式	141
第 10 章 儲存、擷取及更新物件	105	指定 BLOB 或檔案內容	142
影像、音效及視訊格式	105	指定等待指示符	143
影像轉換選項	106	顯示縮圖影像或視訊框	144
儲存影像、音效或視訊物件	107	顯示最大圖影像或視訊框	144
DB2Image、DB2Audio 及 DB2Video UDF 格式	108	播放音效或視訊	145
儲存常駐在從屬站的物件	111		
儲存常駐於伺服器上的物件	112	第 12 章 依內容來查詢影像	147
指定資料庫或檔案儲存體	113	如何按影像內容查詢	147
識別儲存體的格式	114	管理 QBIC 型錄	148
識別儲存體的格式；不進行轉換	114	建立 QBIC 型錄	149
識別儲存體的格式與轉換選項；進行格式轉換	115	開啓 QBIC 型錄	150
以使用者提供的屬性儲存物件	115	變更自動編目設定	151
儲存縮圖 (影像與視訊)	117	將特性加入 QBIC 型錄	152
儲存註解	118	從 QBIC 型錄中移除特性	153
擷取影像、音效或視訊物件	119	擷取 QBIC 型錄的相關資訊	154
擷取作業的 Content UDF 格式	119	手動編目影像	155
將物件擷取至從屬站	121	手動編目單一影像	155
將物件擷取至從屬站；不進行格式轉換	121	手動編目影像直欄	156
將影像擷取至從屬站；進行轉換	122	取消編目影像	156
將物件擷取至伺服器檔案	122	重新編目影像	157
擷取與使用屬性	124	重新配置 QBIC 型錄 (僅適用於 EEE)	157
擷取註解	126	關閉 QBIC 型錄	158
		刪除 QBIC 型錄	158
		QBIC 型錄範例程式	159

建置查詢	163
指定查詢字串	163
特性值	163
特性比重	165
範例	165
使用查詢物件	166
建立查詢物件	166
將特性加入查詢物件	166
指定查詢物件中特性的資料來源	167
設定查詢物件中特性的比重	170
儲存並重覆使用查詢字串	170
擷取查詢物件的相關資訊	171
移除查詢物件中特性	172
刪除查詢物件	172
按影像內容發出查詢	173
查詢影像	173
擷取影像分數	174
擷取單一影像的分數	175
擷取多個影像的分數	175
QBIC 查詢範例程式	176

第 9 章 程式設計概觀

本章提供 DB2 Extender 程式設計的概觀。它提供您在著手設計 Extender 程式之前的必備資訊，並包含一個範例應用程式來說明撰寫 Extender 程式碼的方法。

使用 Extender UDF 和 API

DB2 Extender 提供使用者定義函數，可用來儲存、存取及操作資料庫中的影像、音效及視訊資料。您在應用程式中使用 SQL 陳述式來撰寫這些 UDF 的要求，方法與要求 SQL 內建函數相同。與內建函數一樣，UDF 是在資料庫伺服器中執行。

下列 C 應用程式中的 SQL 陳述式，要求一個稱為 DB2Image 的 Image Extender UDF，來將影像儲存於資料庫表格中；該來源影像的內容位於一個伺服器檔案中：

```
EXEC SQL BEGIN DECLARE SECTION;
        long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL

EXEC SQL INSERT INTO EMPLOYEE VALUES(
        '128557',                                /*id*/
        'Anita Jones',                          /*name*/
        DB2IMAGE(                               /*Image Extender UDF*/
                CURRENT SERVER,                 /*database */
                '/employee/images/ajones.bmp',  /*image content*/
                'ASIS',                        /*keep the image format*/
                :hvStorageType,                /*store image in DB as BLOB*/
                'Anita's picture')              /*comment*/
        );
```

您可使用 Extender 應用程式設計介面來顯示影像及播放音效或視訊物件。您可使用 C 語言中的 client 函數呼叫來撰寫這些 API。函數可在您的資料庫從屬站工作站中執行。

下列 C 陳述式包括一個名為 DBiBrowse 的 API。該 API 擷取影像控點的資料，並啟動瀏覽器來顯示影像：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_hdl [251];
EXEC SQL END DECLARE SECTION

EXEC SQL SELECT PICTURE INTO :hvImg_hdl
        WHERE NAME='Robert Smith';
```

```
rc=DBiBrowse(  
    "ib %s",           /*image browser*/  
    MMDB_PLAY_HANDLE, /*use image handle*/  
    hvImg_hdl,         /*image handle*/  
    MMDB_PLAY_NO_WAIT); /*run browser independently*/
```

UDF 必須以案例的使用者 ID 來執行： DB2 Extender UDF 必須以與 DB2 Extender 案例相同的使用者 ID 來執行。此外，若您建立 DB2 Extender 案例，或使用現存的 DB2 Extender 案例，則 UDF 必須以 DB2 相同的使用者 ID 來執行。

DB2 必須適當地架構：DB2 必須適當地架構，確保 DB2 Extender 能正常作業，尤其是指 DB2 Extender UDF 的正常作業。APP_CTL_HEAP_SZ 資料庫架構參數，特別要適當地設定。

您可以使用 Extender UDF 和 API 來執行的作業

表 7列出您可以使用 Extender UDF 和 API 來執行的作業，並顯示每一項作業說明的位置。

表 7. 您可以使用 DB2 Extender API 來執行的作業

作業	請參閱
儲存影像、音效或視訊物件	第 107 頁
擷取影像、音效或視訊物件	第 119 頁
擷取和使用影像、音效及視訊屬性	第 124 頁
擷取與影像、音效或視訊物件相關的註解	第 126 頁
更新影像、音效或視訊物件	第 126 頁
顯示影像物件	第 141 頁
顯示縮圖影像或視訊框	第 144 頁
播放音效或視訊物件	第 145 頁
依內容來查詢影像	第 147 頁
偵測視訊場景變更	第 19 頁

Extender 範例的範例表格

您將在本章中看到使用 DB2 Extender 的程式設計範例。這些範例假設您建立一個資料庫表格，稱為 EMPLOYEE，其中包含人事資訊。表格中的直欄包含員工的識別和名稱。依 Extender 的不同，表格也可包含員工照片、語音致詞及視訊片段。

第 97 頁的圖 25 說明員工表格的結構，並顯示建立表格所用的 SQL 陳述式。

```
CREATE TABLE 員工 (  
    ID          CHAR(6),  
    名稱        VARCHAR(40),  
    圖片        DB2Image,  
  
    音訊        DB2Audio,  
  
    視訊        DB2Video  
);
```

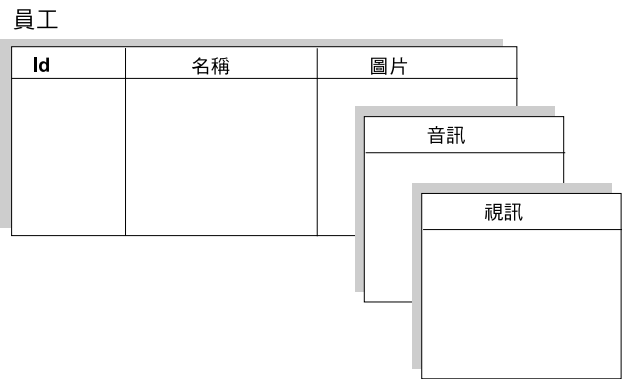


圖 25. 使用於 DB2 Extender 程式設計範例的表格

DB2 Extender 程式設計的事前準備

在開發使用 DB2 Extender 的程式之前，您應該熟悉 DB2 應用程式開發程序和程式設計技術，相關資訊請參閱 *DB2 Application Development Guide*。開發使用 DB2 Extender 的程式，其程序本質上與傳統 DB2 應用程式相同。

您的應用程式之程式碼與傳統 DB2 應用程式有所不同，是因為有 Extender 所定義的新資料類型和函數。例如，第 98 頁的圖 26中以 C 撰寫的應用程式使用了 Image Extender 來識別資料庫表格中儲存的 GIF 影像。識別影像之後，程式即呼叫影像瀏覽程式來顯示它們。

如範例所示，使用 DB2 Extender 的應用程式需要執行下列函數：

- 1** 併入 Extender 定義。範例中的 dmbimage.h 檔案，是 Image Extender 的併入檔（標頭檔）。併入檔中定義 Extender 的常數、變數及函數原型。

事前準備

2 依需要來定義主變數，來包含對 UDF 的輸出入或對 API 呼叫的輸入。範例中的 hvFormat、hvSize、hvWidth、 hvHeight 及 hvComment，是用來包含 Image Extender UDF 所擷取的資料。主變數 hvImg_hdl 是用來包含影像控點，該控點被指定為 Image Extender API 呼叫的輸入。

3 依需要指定 UDF 要求。在範例中，SIZE、 WIDTH、 HEIGHT、COMMENT 及 FORMAT 皆為 Image Extender UDF。

4 依需要指定 API 呼叫。在範例中，DBiBrowse 是呼叫本端 C 函數的 API，而函數顯示的影像，其控點是擷取自表格。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sqlenv.h>
#include <sqlcodes.h>
#include <dmbimage.h> 1

int count=0;

long
main(int argc,char *argv[])
{
EXEC SQL BEGIN DECLARE SECTION; 2
    char hvImg_hdl[251];           /* image handle */
    char hvDBName[19];             /* database name */      char hvName[40];
    char hvFormat[9];              /* image format */
    long hvSize;                   /* image size */
    long hvWidth;                  /* image width */
    long hvHeight;                 /* image height */
    struct {      short len;
                  char data[32700]
                } hvComment;        /* comment about the image */
EXEC SQL END DECLARE SECTION;

/* Connect to database */
strcpy(hvDBName, argv[1]);         /* copy the database name */

EXEC SQL CONNECT TO :hvDBName IN SHARE MODE;
/*
 * Set current function path*/
EXEC SQL SET CURRENT FUNCTION PATH = mmdbsys, CURRENT FUNCTION PATH;
```

圖 26. 使用 DB2 Extender 的應用程式 (1/2)

```

/*
 * Select (query) using Image Extender UDF
 *
 * The SQL statement below finds all images in GIF format.
 */
EXEC SQL DECLARE c1 CURSOR FOR
    SELECT PICTURE, NAME,
           SIZE(PICTURE), WIDTH(PICTURE),
           HEIGHT(PICTURE), COMMENT(PICTURE)
    FROM EMPLOYEE
    WHERE PICTURE IS NOT NULL AND
          FORMAT(PICTURE) LIKE 'GIF%'
FOR FETCH ONLY;

EXEC SQL OPEN c1;
for (;;) {
    EXEC SQL FETCH c1 INTO :hvImg_hdl, :hvName, :hvSize,
                           :hvWidth, :hvHeight, :hvComment;
                                if (SQLCODE != 0)
                                break;

    printf("\nRecord %d:\n", ++count);
    printf("employee name = '%s'\n", hvName);
    printf("image size = %d bytes, width=%d, height=%d\n",
           hvSize, hvWidth, hvHeight);
    hvComment.data[Comment.len]='\0';
    printf("comment = %s\n", hvComment.data);/*
 * The API call below displays the images
 */
    rc=DBiBrowse ("ib %s",MMDB_PLAY_HANDLE,hvImg_hdl,
                  MMDB_PLAY_WAIT);
}

EXEC SQL CLOSE c1;

/* end of program */

```

圖 26. 使用 DB2 Extender 的應用程式 (2/2)

併入 Extender 定義

對於所用的每一個 Extender，您的應用程式中需要一個併入檔（標頭檔）。每個併入檔中都會定義 Extender 所用的常數、變數及函數原型。併入檔的名稱是：

併入檔	Extender
dmbimage.h	影像
dmbqbapi.h	影像（依內容來查詢）
dmbaudio.h	音效

dmbvideo.h	視訊
dmbshot.h	視訊 (場景變更偵測)

您可使用 `#include` 指引，將併入檔帶入 C 程式內。例如，下列指引帶入 Image Extender 的併入檔：

```
#include <dmbimage.h>
```

指定 UDF 和 UDT 名稱

DB2 Extender UDF 的全名是 `mmdbsys.function-name`。DB2 Extender UDT 的全名是 `mmdbsys.type-name`，其中 `mmdbsys` 是指函數或特殊類型的綱目名稱。例如，Content UDF 的全名是 `mmdbsys.Content`；而 Image Extender 建立的 DB2Image 資料類型，其全名是 `mmdbsys.DB2Image`。若您之前的現行函數路徑設定為 `mmdbsys`，則可以忽略 `mmdbsys` 綱目名稱，例如：

```
SET CURRENT FUNCTION PATH = mmdbsys, CURRENT FUNCTION PATH
```

```
SET CURRENT PATH = mmdbsys, CURRENT PATH
```

傳輸大型物件

您可以使用各種方式，在應用程式和 DB2 資料庫之間傳輸大型物件，例如影像、音效片段及視訊片段。所用的方法，是依物件的傳輸是在檔案或記憶體緩衝區上，而有所不同。也依檔案是位於從屬站機器或資料庫伺服器機器中而定。

若物件是在表格和伺服器檔案之間傳輸

當您在資料庫表格和伺服器檔案之間傳輸物件時，請指定適當的 Extender UDF 要求中指定檔案路徑。因為 Extender UDF 和檔案兩者皆在伺服器上，所以 Extender 能夠找出該檔案。例如，在下列 SQL 陳述式中，內容位於伺服器檔案中的影像，是儲存於資料庫表格中：

```
EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2Image(
        CURRENT SERVER,
        '/employee/images/ajones.bmp',
        'ASIS',
        :hvStorageType,
        'Anita's picture')
    );
```


若物件是在從屬站緩衝區中來回傳輸

Extender 無法直接存取記憶體緩衝區。若您想在從屬站機器的緩衝區上，來回傳輸物件，則需要一種方法來完成，而不是指定緩衝區位置。其中一種方法是透過主變數。這種方法是一般在應用程式和 DB2 資料庫之間傳輸物件時所使用的。

定義和使用大型物件的主變數，其方式與傳統字元和數字物件相同。您可在 DECLARE 區段中宣告主變數、指派要傳輸的值、或存取傳輸給它們的值。

宣告影像、音效或視訊資料的主變數時，請指定 BLOB 的資料類型。當您使用 UDF 來儲存、擷取或更新物件時，請指定適當的主變數當做 UDF 要求中的引數。使用的格式，與 SQL 陳述式中指定的其它主變數相同。

例如，下列 SQL 陳述式宣告並使用一個主變數，稱為 hvaudio，來裁剪音效和傳輸音效至資料庫：

```
EXEC SQL BEGIN DECLARE SECTION;
      SQL TYPE IS BLOB (2M) hvaudio;
EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
      DB2Audio(
        CURRENT SERVER,
        :hvaudio,
        'WAVE',
        CAST(NULL as LONG VARCHAR),          'Anita''s voice')
      );
```

使用 LOB 定位器

例如音效和視訊片段之類的大型物件可能非常大，因此使用主變數來操作可能並非十分有效率。使用 **LOB 定位器**在應用程式中操作 LOB，也許是一種較佳的方式。

LOB 定位器是主變數中儲存的一個小值 (4 位元組)，供程式用來參照 DB2 資料庫中的大型 LOB。您的程式透過 LOB 定位器操作 LOB 時，LOB 就如同儲存在一般主變數中一樣。差異在於不需要在資料庫伺服器 and 從屬站機器的應用程式之間傳輸 LOB。例如，在資料庫表格中選取 LOB 時，該 LOB 仍保留在伺服器上，但 LOB 定位器會移至從屬站。

宣告 LOB 定位器是在 DECLARE 區段中，而使用方式與主變數相同。宣告影像、音效或視訊資料的 LOB 定位器時，請指定 BLOB LOCATOR 的資料類型。例如，下列 SQL 陳述式宣告並使用一個 LOB 定位器 (稱為 video_loc)，從資料庫表格中擷取視訊短片：

```
EXEC SQL BEGIN DECLARE SECTION;  
SQL TYPE IS BLOB_LOCATOR video_loc;  
EXEC SQL END DECLARE SECTION;  
  
EXEC SQL SELECT CONTENT(VIDEO)  
INTO :video_loc  
FROM EMPLOYEE  
WHERE NAME='Anita Jones';
```

UDF 使用 LOB 定位器：用以儲存、擷取及更新影像、音效及視訊物件的 DB2 Extender，使用 LOB 定位器。DB2 Extender V1 中的這些 UDF，並不使用 LOB 定位器，因此，無法處理大於 2 MB 的物件。這項限制強迫使用者在區段中傳輸大於 2 MB 的物件。因為現在這些 UDF 使用 LOB 定位器，所以就無 2 MB 的限制。

若物件是在從屬站檔案中來回傳輸

使用檔案參考變數，對從屬站透過檔案參考變數，可讓您不須在應用程式中配置大型物件的緩衝區空間。使用檔案參考變數和 UDF 時，DB2 會直接在檔案和 UDF 之間傳遞 BLOB 內容。

宣告檔案參考變數是在 DECLARE 區段中，而使用方式與主變數相同。宣告影像、音效或視訊資料的檔案參考變數時，請指定 BLOB_FILE 的資料類型。但是，不像含有物件內容的主變數，檔案參考變數會包含檔案名稱。檔案的大小不能超過 UDF 的 BLOB 大小。

使用檔案參考變數來輸入和輸出，有著各種不同的選項。在程式的檔案參考變數結構中，可以設定 FILE_OPTIONS 欄位來選擇您要的選項。您可以選擇下列選項：

輸入的選項：

SQL_FILE_READ。此檔案可以開啓、讀取及關閉。檔案中的資料長度 (位元組)，是在開啓檔案時決定。檔案參考變數的 data_length 欄位，保留檔案的長度 (位元組)。

輸出的選項：

SQL_FILE_CREATE。此選項建立新的檔案 (若尚未存在)。若檔案已存在，則傳回錯誤訊息。檔案參考變數的 data_length 欄位，保留檔案的長度 (位元組)。

SQL_FILE_OVERWRITE。此選項建立新的檔案 (若尚未存在)。若檔案已存在，則新資料會改寫檔案中的資料。檔案參考變數的 data_length 欄位，保留檔案的長度 (位元組)。

SQL_FILE_APPEND。此選項將輸出添加至已存在的檔案。若檔案尚未存在，則建立新檔。檔案參考變數的 data_length 欄位，保留新增至檔案的資料長度 (位元組)，而不是檔案的總長度。

例如，下列陳述式宣告一個檔案參考變數，稱為 `Img_file`，並用來將影像（其內容位於從屬站檔案中）儲存到資料庫表格內。請注意 `FILE_OPTIONS` 欄位中的 `SQL_FILE_READ` 指定：

```
EXEC SQL BEGIN DECLARE SECTION;
      SQL TYPE IS BLOB_FILE Img_file;
EXEC SQL END DECLARE SECTION;

strcpy (Img_file.name, "/employee/images/ajones.bmp");
Img_file.name_length=strlen(Img_file.name);
Img_file.file_options=SQL_FILE_READ;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
      DB2Image(
        CURRENT SERVER,
        :Img_file,
        'ASIS',
        CAST(NULL as LONG VARCHAR),          'Anita''s picture')
      );
```

傳輸物件時指定檔名

當您儲存、擷取或更新物件時，DB2 Extender 能讓您指定檔名時更有彈性。

雖然可以指定完整的檔名（亦即完整路徑後面接著檔名）來進行儲存、擷取及更新作業，但是建議您最好指定相對檔名。在 AIX、HP-UX 及 Solaris 中，相對檔名即為不以斜線開頭的檔名。在 Windows 中，相對檔名即為不以磁碟機字母加上冒號及反斜線開頭的檔名。

若指定相對檔名，則 Extender 會使用不同從屬站和伺服器環境環境中的目錄規格來分辨檔名。完整的路徑名稱包含前端部份及尾端部份，前者通常是關於裝載點，後者以唯一的方式來識別所需的檔案。尾端路徑名稱指定於 UDF 中。環境變數提供一連串前端路徑名稱，於嘗試分辨相對檔名時作為搜尋之用。有關 DB2 Extender 用來分辨檔名的環境變數的資訊，請參閱第 543 頁的附錄 A，『設定 DB2 Extender 的環境變數』。

Extender 也會適當地轉換檔名格式。當檔名傳遞到伺服器時，會轉換成適合伺服器作業系統的格式。

處理回覆碼

程式中之所有內含的 SQL 陳述式或 DB2 CLI 呼叫，包括要求 DB2 Extender UDF 的陳述式或 DB2 CLI 呼叫，會建立程式碼來指出內含的 SQL 陳述式或 DB2 CLI 呼叫是否成功執行。其它 DB2 Extender API，例如管理的 API，也有回覆碼來指出成功與否。您的程式應該檢查和回應內含的 SQL 陳述式、CLI 呼叫及 API 所傳回的回覆碼。

有關處理這些回覆碼的資訊，請參閱第 509 頁的第 16 章，『診斷資訊』。

在 Extender API 無法順利完成工作單元的狀況之下，則執行 ROLLBACK 作業。API 也會傳回錯誤碼。完成 ROLLBACK 作業之後，資料庫才能回到其先前的一致點。詳細資訊，請參閱第 251 頁的第 14 章，『應用程式設計介面』。

Unicode 支援

請遵循下列幾點關於 Image、Audio 及 Video Extender 的 Unicode 支援：

- 可能是 Unicode 字串只有下列 UDF 的註解欄位中的參數：
 - mmdbsys.db2image() 匯入影像
 - mmdbsys.db2audio() 匯入音效
 - mmdbsys.db2video() 匯入視訊
 - mmdbsys.replace() 取代影像、音效或視訊
 - mmdbsys.comment() 註解更新
- 如果您規劃要存取 Unicode 資料庫，則必須設定 DB2 Extender 以支援 Unicode。Unicode 案例只能用來處理 Unicode 資料庫。

若要 Extender 案例支援 Unicode，呼叫 DMBSTART 之前，您必須將環境變數 DB2CODEPAGE 設為 1208。

第 10 章 儲存、擷取及更新物件

本章節說明如何使用 DB2 Extender 使用者定義的功能來儲存、擷取及更新影像、音效或視訊。

影像、音效及視訊格式

表 8 列出可儲存、擷取或更新影像、音效及視訊物件的格式。若是影像物件，Image Extender 可在儲存、擷取或更新影像時轉換其格式。(但在儲存、擷取或更新音效及視訊物件時，並無法轉換其格式。)

表格中的「讀取」與「寫入」直欄分別表示可讀取何種格式，以及在寫入時可轉換何種格式。當表格中「讀取」直欄的項目是 x 時，表示在儲存、擷取或更新時可使用的對應的物件格式。如果表格中「寫入」直欄的項目為 x，則在儲存、擷取或更新影像物件時，可將其轉換成對應的格式。例如，使用 BMP 格式的影像在儲存、擷取或更新時，可轉換成 GIF 格式。使用 JPG 格式的影像可轉換成 TIF 格式。但使用 TIF 格式的影像無法轉換成 JPG 格式。

表格中所列出的格式為大寫，但實際上儲存、擷取或更新要求的格式規格並不區分大小寫。例如，規格 GIF、gif 及 Gif 是一樣的。

表 8. DB2 Extender 可處理的格式

格式	說明	讀取	寫入
影像格式			
_IM	PS/2 音效視訊連接 (AVC)	x	
BMP	OS/2 - Microsoft Windows 點陣圖 ²	x	x
EPS	封裝 PostScript		x
EP2	封裝層次 2 PostScript		x
GIF	Compuserve GIF89a (包括動態 GIF ³) 與 87	x	x
IMG	IOCA 影像	x	x
IPS	Brooktrout FAX 卡檔	x	x
JPG	JPEG ⁴ (JFIF 格式)	x	
PCX	PC 上色檔 (只限灰階)	x	x
PGM	可轉移式灰色對映 (來自 PBMPLUS)	x	x
PS	PostScript		x

格式

表 8. DB2 Extender 可處理的格式 (繼續)

格式	說明	讀取	寫入
PSC	壓縮的 PostScript 影像		x
PS2	PostScript 層次 2 (彩色)		x
TIF	所有 TIFF 5.0 格式	x	x
YUV	YUV 數位視訊	x	x
音效格式			
AIF 或 AIFF	音效交換檔案格式	x	
AIFFC	壓縮的音效交換檔案格式	x	
AU	Sun 音效檔案格式	x	
MIDI	樂器數位介面	x	
MPG1 或 MPEG1	第一類動態影像壓縮格式	x	
WAV 或 WAVE	Wave	x	
視訊格式			
AVI	音效/視訊交錯	x	
MPG1 或 MPEG1	第一類動態影像壓縮格式	x	
MPG2 或 MPEG2	第二類動態影像壓縮格式	x	
QT	Quicktime (AVI)	x	

影像轉換選項

表 9列出在儲存、擷取或更新影像時可對其指定的轉換選項 (格式轉換之外)。Image Extender 會將您的規格引用到目標影像；不會變更來源影像。

每一個轉換選項的指定格式為參數/值對。表格中列出各參數的容許值。

表 9. 影像轉換選項

參數	說明	值
-b	用來呈現各影像樣本的位元數	1 或 8 位元

- 2. Windows 版本 2、Windows 版本 3 及 Windows NT BMP 格式可支援讀取功能。
- 3. DB2 Image Extender 只對動態 GIF 檔中第一個影像儲存屬性資訊。
- 4. 支援使用的軟體是部分基於獨立 JPEG 群組之工作的軟體。

表 9. 影像轉換選項 (繼續)

參數	說明	值
-s ⁵	標度因數	任何大於零的十進位值。標度因數指定對影像原型轉換的大小比例。例如，標度因數 0.5 會將影像轉換成其原有大小的一半。標度因數 2.0 會將影像轉換成其原有大小的兩倍。
-p	光度測定 (影像反轉)。此選項會根據指定的值，會變更影像的表現方式。影像本身不會變更。此選項僅適用於黑白或灰階影像，而且不適用於使用 GIF 格式的影像。	0 = 黑 1 = 白
-n	光度測定 (影像反轉)。此選項藉由黑白對調來改變影像。此選項僅適用於黑白或灰階影像。	無
-r ⁵	旋轉	0 = 0 度 (無旋轉) 1 = 90 度 (逆時針方向) 2 = 90 度 (順時針方向) 3 = 180 度
-x ⁵	寬度 (以圖點計)	圖點數
-y ⁵	高度 (以圖點計)	圖點數
-c	壓縮類型	0 = IBM MMR 1 = CCITT 群組 3 1-D 2 = CCITT 群組 3 2-D (k=2) 3 = CCITT 群組 3 2-D (k=4) 4 = CCITT 群組 4 6 = TIFF 類型 2 10 = 未壓縮 14 = LZW 15 = TIFF Packbit 25 = JBIG

儲存影像、音效或視訊物件

在 SQL INSERT 陳述式中使用 DB2Image、DB2Audio 或 DB2Video UDF，將影像、音效或視訊物件儲存在資料庫。

5. 如果您對交錯式 GIF 影像指定此選項，則應同時指定 LZW 的壓縮類型。

您可以儲存來源是在從屬站機器緩衝區或檔案，或伺服器檔案中的物件。就這些來源而言，您可在資料庫表格中將物件儲存為 **BLOB**，或將物件儲存在資料庫伺服器上的某個檔案中。

當要求 **UDF** 時，您需要指定下列各項：

- 目前連接之資料庫伺服器的名稱；這包含在 **CURRENT SERVER** 特別暫存區中。
- 物件內容的來源；可以是從屬站緩衝區、從屬站檔案 或伺服器檔案。
- 要將物件儲存為資料庫表格中的 **BLOB**，或儲存在檔案伺服器中。
- 來源的格式。
- 要隨物件一起儲存的註解 (若不想儲存註解，則為 **NULL** 值或空字串)。

Image、**Audio** 和 **Video Extender** 可讓您儲存物件 (即使是它們無法識別的物件格式)。如果無法辨識其格式，您就需要指定該物件的屬性。當您連同使用者提供的屬性一起儲存影像或視訊時，您也可以儲存縮圖。縮圖是代表影像或視訊的小型影像。

如果是影像，您可以選擇在儲存影像時轉換其格式。如果您要求格式轉換，則需要指定影像的來源及目標格式。在格式轉換要求中，您也可以指定對影像作進一步的變更，如加以修剪或旋轉。指定轉換選項即可指示這些變更。

確定儲存作業：在資料庫中儲存影像、音效或視訊物件之後，需要確定工作單元。這樣會解除 **Extender** 所保留的鎖定，使您可以對儲存的物件執行更新作業。

DB2Image、DB2Audio 及 DB2Video UDF 格式

DB2Image、**DB2Audio** 及 **DB2Video** UDF 可被超載，也就是說其格式隨使用 UDF 的方式而有所不同。UDF 具有下列格式 (格式中顯示的 **xxxxx** 可以是 **Image**、**Audio** 或 **Video**)：

格式 1：儲存一個來自從屬站緩衝區或從屬站檔案的物件：

```
DB2xxxxx(  
    CURRENT SERVER,      /* database name name in CURRENT SERVER REGISTER */  
    content,              /* object content */  
    format,               /* source format */  
    target_file,          /* target file name for storage in file server */  
                        /* or NULL for storage in table as BLOB */  
    comment               /* user comment */  
);
```

格式 2：儲存來自伺服器檔案的物件：


```

DB2xxxxx(
    CURRENT SERVER,      /* database name in CURRENT SERVER REGISTER */
    source_file,         /* source file name */
    format,              /* source format */
    stortype,            /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                        /* in file server*/
                        /* MMDB_STORAGE_TYPE_INTERNAL=store */
                        /* as a BLOB*/
    comment              /* user comment */
);

```

格式 3：儲存從屬站緩衝區或檔案中具有使用者提供之屬性的物件：

```

DB2xxxxx(
    CURRENT SERVER,      /* database name in CURRENT SERVER REGISTER */
    content,             /* object content */
    target_file,         /* target file name for storage in file server */
                        /* or NULL for storage in table as BLOB */
    comment,             /* user comment */
    attrs,               /* user-supplied attributes */
    thumbnail            /* thumbnail (image and video only) */
);

```

格式 4：儲存伺服器檔案中具有使用者提供之屬性的物件：

```

DB2xxxxx(
    CURRENT SERVER,      /* database name in CURRENT SERVER REGISTER */
    source_file,         /* source file name */
    stortype,            /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                        /* in file server*/
                        /* MMDB_STORAGE_TYPE_INTERNAL=store */
                        /* as a BLOB*/
    comment,             /* user comment */
    attrs,               /* user-supplied attributes */
    thumbnail            /* thumbnail (image and video only) */
);

```

DB2Image UDF 包括下列附加的格式：

格式 5：儲存來自從屬站緩衝區或從屬站檔案的影像，同時做格式轉換：

```

DB2Image(
    CURRENT SERVER,      /* database name in CURRENT SERVER REGISTER */
    content,             /* object content */
    source_format,       /* source format */
    target_format,       /* target format */
    target_file,         /* target file name for storage in file server */
                        /* or NULL for storage in table as BLOB */
    comment              /* user comment */
);

```

格式 6：儲存來自伺服器檔案，並做格式轉換的影像：

```
DB2Image(  
    CURRENT SERVER,      /* database name in CURRENT SERVER REGISTER */  
    source_file,         /* server file name */  
    source_format,       /* source format */  
    target_format,       /* target format */  
    target_file,         /* target file name for storage in file server */  
                        /* or NULL for storage in table as BLOB */  
    comment              /* user comment */  
);
```

格式 7：儲存來自從屬站緩衝區從屬站檔案，並做格式轉換和額外變更的影像：

```
DB2Image(  
    CURRENT SERVER,      /* database name in CURRENT SERVER REGISTER */  
    content,             /* object content */  
    source_format,       /* source format */  
    target_format,       /* target format */  
    conversion_options,  /* Conversion options */  
    target_file,         /* target file name for storage in file server */  
                        /* or NULL for storage in table as BLOB */  
    comment              /* user comment */  
);
```

格式 8：儲存來自伺服器檔案，並做格式轉換和額外變更的影像：

```
DB2Image(  
    CURRENT SERVER,      /* database name in CURRENT SERVER REGISTER */  
    source_file,         /* server file name */  
    source_format,       /* source format */  
    target_format,       /* target format */  
    conversion_options   /* conversion options */  
    target_file,         /* target file name for storage in file server */  
                        /* or NULL for storage in table as BLOB */  
    comment              /* user comment */  
);
```

例如，下列在 C 應用程式中的陳述式插入了將影像併入員工表格的列。來源影像為名為 ajones.bmp 的伺服器檔案。影像以 BLOB 形式儲存在員工表格內。(它對應於上述格式 2。)

```
EXEC SQL BEGIN DECLARE SECTION;  
    long hvStorageType;  
EXEC SQL END DECLARE SECTION;
```

```
hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;
```

```
EXEC SQL INSERT INTO EMPLOYEE VALUES(  
    '128557',           /*id*/  
    'Anita Jones',     /*name*/  
    DB2IMAGE(          /*Image Extender UDF*/  
        CURRENT SERVER, /*database*/  
        '/employee/images/ajones.bmp', /*source file */  
    );
```

```

        'ASIS',                                /*keep the image format*/
        :hvStorageType                         /*store image in DB as BLOB*/
        'Anita's picture')                   /*comment */
);

```

下列在 C 應用程式中的陳述式將同一橫列儲存至前例中的員工表格。但此處在儲存影像時，其格式 BMP 被轉換成 GIF。(它對應於上述格式 6。)

```

EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',                                /*id*/
    'Anita Jones',                           /*name*/
    DB2IMAGE(                                /*Image Extender UDF*
        CURRENT SERVER,                     /*database*/
        '/employee/images/ajones.bmp',     /*source file */
        'ASIS',                             /*source image format*/
        'GIF',                              /*target image format*/
        'Anita's picture')                 /*comment*/
);

```

當您儲存影像音效或視訊物件時，Extender 會計算許多屬性，諸如影像中所使用的顏色數、音效播放時間或視訊壓縮格式。若儲存的物件含有無法辨識的格式，您必須提供這些屬性作為 UDF 的輸入。Extender 會連同其它屬性 (如有關物件的註解及儲存物件之使用者的識別) 將這些屬性儲存在資料庫中。然後您便可以在查詢中使用這些屬性。

儲存常駐在從屬站的物件

使用主變數或檔案參考變數，自從屬站緩衝區或從屬站檔案將影像、音效或視訊物件傳送到伺服器。

如果物件是在從屬站檔案中，請使用檔案參考變數來傳送其內容，以儲存在伺服器中。例如，下列 C 應用程式中的陳述式中，定義一個檔案參考變數 `Audio_file`，然後使用該變數傳送其內容是在從屬站檔案的音效片段。音效片段會被儲存在伺服器上的資料庫表格。請注意：檔案參考變數的 `file_options` 欄位會被設定為 `SQL_FILE_READ` 以供輸入。另請注意檔案參考變數被用來作為對 `DB2Audio UDF` 的內容引數。

```

EXEC SQL BEGIN DECLARE SECTION;
    SQL TYPE IS BLOB FILE Audio_file;
EXEC SQL END DECLARE SECTION;

strcpy (Audio_file.name, "/employee/sounds/ajones.wav");
Audio_file.name_length= strlen(Audio_file.name);
Audio_file.file_options= SQL_FILE_READ;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2AUDIO(
        CURRENT SERVER,

```

```
        :Audio_file,          /* file reference variable */
        'WAVE',
        CAST(NULL as LONG VARCHAR),

        'Anita''s voice')
    );
```

若物件位於從屬站緩衝區，請使用一個定義成 BLOB 或 BLOB_LOCATOR 的主變數來傳送物件內容，以便儲存於伺服器。在下列 C 應用程式陳述式中，主變數 Video_loc 用來傳送視訊片段內容以便儲存在伺服器。視訊片段被儲存為資料庫表格中的 BLOB。請注意：主變數被用來作為對 DB2Video UDF 的內容引數。

```
EXEC SQL BEGIN DECLARE SECTION;
    SQL TYPE IS BLOB_LOCATOR Video_loc;
EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2VIDEO(
        CURRENT SERVER,
        :Video_loc,          /* host variable */
        'MPEG1',
        '',
        'Anita''s video')
    );
```

請確定有足夠 UDF 記憶體：儲存一個其內容是在從屬站緩衝區的物件時，您必須確定「資料庫管理程式架構」中的 UDF_MEM_SZ 參數設定為 4 MB 或更大。您可以使用 DB2 命令 UPDATE DATABASE MANAGER CONFIGURATION 來更新 UDF_MEM_SZ 參數。如需有關 UPDATE DATABASE MANAGER 命令進一步的資訊，請參閱 *DB2 Command Reference*。

儲存常駐於伺服器上的物件

如果要儲存的影像、音效或視訊是在伺服器檔案中，請將該檔案的路徑指定為對 UDF 的內容引數。例如，下列在 C 應用程式中的陳述式會儲存一個將影像併入資料庫的橫列。影像內容是放在伺服器的檔案中。儲存的影像仍留在伺服器檔案中，且資料庫會指向它。

```
EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2IMAGE(
        CURRENT SERVER,
```

```

'/employee/images/ajones.bmp', /*source in server file */
'BMP',
:hvStorageType,
'Anita''s picture')
);

```

指定正確的路徑：當您儲存其來源是在伺服器檔案的物件時，您可以指定檔案的完整名稱或相對名稱。如果是指定相對名稱，您需要確定 DB2 伺服器中的適當環境變數包括檔案的正確路徑。有關設定這些環境變數的資訊，請參閱第 543 頁的附錄 A，『設定 DB2 Extender 的環境變數』。

指定資料庫或檔案儲存體

您可將影像、音效或視訊物件儲存為資料庫表格中的 BLOB，或儲存在伺服器檔案中。如果您將物件儲存在伺服器檔案中，資料庫會指向該檔案。

若儲存來自從屬站緩衝區或從屬站檔案的物件，請您依照 `target_file` 參數中指定的結果，指示 BLOB 或伺服器檔案儲存體。如果是指定檔名，表示您要將物件儲存在伺服器檔案中。若指定一個 NULL 值或空字串，則表示您要在資料庫表格中將物件儲存為 BLOB。 `target_file` 參數的資料類型是 LONG VARCHAR。若指定一個 NULL 值，請記得將該值強制轉型為 LONG VARCHAR 資料類型。

例如，下列在 C 應用程式中的陳述式會儲存一個將影像併入資料庫表格的橫列。影像來源是位於從屬站緩衝區中。影像會儲存在伺服器檔案。資料庫表格會指向此伺服器檔案：

```

EXEC SQL BEGIN DECLARE SECTION;
      SQL TYPE IS BLOB LOCATOR Img_buf
EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
      DB2IMAGE(
        CURRENT SERVER,
        :Img_buf,
        'ASIS',
        '/employee/images/ajones.bmp',    /* store image in server file */
        'Anita''s picture')
      );

```

如果是儲存伺服器檔案的物件，請指定常數

`MMDB_STORAGE_TYPE_INTERNAL`，將物件儲存為資料庫表格中的 BLOB。如果要儲存物件並將其內容存放在伺服器檔案中，請指定常數

`MMDB_STORAGE_TYPE_EXTERNAL`。`MMDB_STORAGE_TYPE_INTERNAL` 有整數值 1。`MMDB_STORAGE_TYPE_EXTERNAL` 有整數值 0。

例如，在下列 C 應用程式中會將音效片段儲存在伺服器檔案中。來源音效內容已存放在伺服器檔案中。儲存作業將檔名置於資料庫內，使您可透過 SQL 陳述式存取檔案。

```
EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2AUDIO(
        CURRENT SERVER,
        '/employee/sounds/ajones.wav',
        'WAVE',
        :hvStorageType,                /* store audio in server file */
        'Anita''s voice')
    );
```

識別儲存體的格式

當您儲存物件時，您需要識別其格式。您可以指定的格式列示於第 105 頁的表 8 中。**Extender** 將會以與來源相同的格式儲存影像、音效或視訊物件。如果是影像物件，您可以選擇讓 **Image Extender** 轉換儲存之影像的格式。若選擇轉換影像格式，您必須指定來源影像的格式以及目標影像的格式。目標影像是儲存時的影像。

識別儲存體的格式：不進行轉換

當您儲存物件而不進行格式轉換時，請指定來源影像、音效或視訊物件的格式。例如，下列在 C 應用程式中的陳述式會將點陣圖 (BMP) 影像儲存至資料庫表格中。來源的內容位於伺服器檔案中。目標影像的格式會與來源影像的格式相同。

```
EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2IMAGE(
        CURRENT SERVER,
        '/employee/images/ajones.bmp',
        'BMP',                        /*image in BMP format */
        '',
        'Anita''s picture')
    );
```

您也可以指定 **NULL** 值或空白字串為格式，或對 **Image Extender** 指定字串 **ASIS**。然後 **Extender** 會檢查來源來決定格式。

對可辨識的格式使用 NULL 或 ASIS： 只在格式可為 **Extender** 所辨識 (亦即 **Extender** 列示在 第 105 頁的表 8 中的其中一個格式) 的情況下，指定 **NULL** 值、空白字串或 **ASIS**。否則，**Extender** 將無法儲存物件。

識別儲存體的格式與轉換選項：進行格式轉換

當您進行格式轉換而儲存影像時，請指定來源與目標影像的格式。第 105 頁的表 8 列出容許的格式轉換。

此外，您可以指定轉換選項藉以識別諸如旋轉或壓縮等額外變更 (變更的對象為已儲存的影像)。您可透過參數及相關值來指定各轉換選項。第 106 頁的表 9 中會列出參數及容許的值。您可以指定多組參數/值配對，對儲存的影像要求多重變更。

在下列範例中，在將點陣圖 (BMP) 影像 (其內容在伺服器檔案中) 儲存於資料庫表格中時，其格式被轉換成 GIF 格式。

```
EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2IMAGE(
        CURRENT SERVER,
        '/employee/images/ajones.bmp',
        'BMP',                               /* source format */
        'GIF',                               /* target format */
        '',
        'Anita's picture')
    );
```

在下列範例中，前例中的影像在儲存於資料庫表格中時被轉換成 GIF 格式。此外，在儲存時該影像被裁剪成 寬 110 圖點、高 150 圖點，並使用 LZW 壓縮加以壓縮。

```
EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2IMAGE(
        CURRENT SERVER,
        '/employee/images/ajones.bmp',
        'BMP',                               /* source format */
        'GIF',                               /* target format */
        '-x 110 -y 150 -c 14',               /* conversion options */
        '/employee/images/ajones.gif',
        'Anita's picture')
    );
```

以使用者提供的屬性儲存物件

儲存影像、音效或視訊物件的格式不限於 Extender 所能瞭解的格式。您可以指定您自己的格式。由於 Extender 並不瞭解該格式，因此您必須指定來源物件的屬性。請在某個屬性結構中指定屬性值。屬性結構必須儲存在 UDF 的 LONG VARCHAR FOR BIT DATA 變數的資料欄位中。

伺服器上的 UDF 碼會預期資料為『big endian 格式』。Big endian 格式是大部份 UNIX 平台使用的格式。若使用『little endian 格式』儲存某物件，您必須準備使

用者提供的屬性資料，以便伺服器上的 UDF 碼能夠正確處理該物件。Little endian 格式是通常用於 Intel® 和其它微處理器平台的格式。(即使您沒有使用 little endian 格式儲存物件，仍然建議您準備使用者提供的屬性資料。) 使用 DBiPrepareAttr API 來準備影像物件的屬性。使用 DBaPrepareAttr API 來準備音效物件的屬性。使用 DBvPrepareAttr API 來準備視訊物件的屬性。

例如，下列在 C 應用程式中的陳述式會儲存一個將影像併入資料庫表格的橫列。位於伺服器檔案中的來源影像，其具有使用者定義的格式、高 640 圖點及寬 480 圖點。請注意：要在儲存影像之前準備屬性。

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
struct {
    short len;
    char data[400];
}hvImgattr;
EXEC SQL END DECLARE SECTION;

DB2IMAGEATTRS    *pimgattr;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

pimgattr = (DB2IMAGEATTRS *) hvImgattr.data;
strcpy(pimgattr->format,"FormatI");
pimgattr->width=640;
pimgattr->height=480;
hvImgattr.len=sizeof(DB2IMAGEATTRS);

DBiPrepareAttr(pimgattr);

DBEXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2IMAGE(
        CURRENT SERVER,
        '/employee/images/ajones.bmp',
        :hvStorageType,
        'Anita''s picture',
        :hvImgattr,
        CAST(NULL as LONG VARCHAR)
    ),
    /* user-specified attributes */
);
```

下列在 C 應用程式中的陳述式會儲存一個將音效片段併入資料庫表格的橫列。來源音效片段位於伺服器檔案中，具有使用者定義的格式、取樣率 44.1 kHz，並具有兩個記錄的通道。此音效片段不是 MIDI，所以請將磁軌名稱和樂器指定為空字串。

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
struct {
    short len;
    char data[600];
}hvAudattr;
```



```

EXEC SQL END DECLARE SECTION;

MMDBAudioAttrs          *paudiattr;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

paudioattr=(MMDBAudioAttrs *) hvAudattr.data;
strcpy(paudioAttr->cFormat,"FormatA");
paudioAttr->ulSamplingRate=44100;
paudioAttr->usNumChannels=2;
hvAudattr.len=sizeof(MMDBAudioAttrs);

DBaPrepareAttrs(paudioAttr);

EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2AUDIO(
        CURRENT SERVER,
        '/employee/sounds/ajones.aud',
        :hvStorageType,
        'Anita's voice',
        :hvAudattr)                /* user-specified attributes */
    );

```

儲存縮圖 (影像與視訊)

當您以自己的格式儲存影像時，您也可以儲存縮圖，其為影像的小型版本。您自行控制縮圖的大小及格式。當您以 Image Extender 可辨識的格式儲存影像時，它會自動建立及儲存物件的縮圖。Image Extender 會以 GIF 格式、112 x 84 圖點大小建立縮圖。

當您以自己的格式儲存視訊物件時，您也可以儲存縮圖，它將視訊物件加以符號化。當您以 Video Extender 可辨識的格式儲存視訊物件時，它會自動儲存物件的同屬縮圖。Video Extender 會以 GIF 格式、108 x 78 圖點大小建立縮圖。

若不要在儲存具有使用者提供的屬性的影像或視訊物件時儲存縮圖，請指定 NULL 值或空字串來代替縮圖。

請在您的程式中建立縮圖 -- Extender 並不會提供 API 來產生縮圖。在您的程式中建立縮圖的結構，並指定 UDF 中的縮圖結構。

下列在 C 應用程式中的陳述式會儲存一個將視訊片段併入資料庫表格的橫列。來源視訊裁剪 (內容位於伺服器檔案) 擁有使用者定義的格式。視訊內容將存放在伺服器中，表格會指向該內容。同時會儲存代表視訊框的縮圖。

```

EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
    struct {
        short len;

```

```

        char data[4000];
    }hvVidattrs;
struct {
    short len;
    char data[10000];
    }hvThumbnail;
EXEC SQL END DECLARE SECTION;

MMDBVideoAttrs          *pvideoAttr;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

pvideoAttr=(MMDBVideoAttrs *) hvVidattrs.data;
strcpy(pvideoAttr->cFormat,"Formatv");
hvVidattrs.len=sizeof(MMDBVideoAttrs);

/* Generate thumbnail and assign data in video structure */

EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2VIDEO(
        CURRENT SERVER,
        '/employee/videos/ajones.vid',
        :hvStorageType,
        'Anita's video',
        :hvVidattrs,
        :hvThumbnail)                /* Thumbnail*/
    );

```

儲存註解

在 UDF 要求中指定註解，便可將註解連同影像、音效或視訊物件一起儲存。註解是資料類型為 **LONG VARCHAR** 的自由形式文字，該文字可多達 32,700 位元組。若不要在儲存物件時儲存註解，請指定 **NULL** 值或空字串來代替註解。若指定一個 **NULL** 值，請記住將該值強制轉型為 **LONG VARCHAR** 資料類型。

例如，下列在 C 應用程式中的陳述式會將註解連同視訊片段一起儲存。

```

EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2VIDEO(
        CURRENT SERVER,
        '/employee/videos/ajones.mpg',

```

```

        'MPEG1',
        :hvStorageType,
        'Anita's video')          /* comment */
);

```

下列在 C 應用程式中的陳述式會儲存沒有註解的影像。

```

EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2IMAGE(
        CURRENT SERVER,
        '/employee/images/ajones.bmp',
        'GIF',
        :hvStorageType,
        Cast(NULL as LONG VARCHAR)          /* no comment */
    );

```

擷取影像、音效或視訊物件

在 SQL SELECT 陳述式上使用 Content UDF，從資料庫表格中擷取影像、音效或視訊物件。您可將物件擷取至從屬站緩衝區、從屬站檔案、或伺服器檔案。

擷取作業的 Content UDF 格式

Content UDF 已超載，這表示可根據 UDF 的使用方式，決定不同的格式。其格式如下：

格式 1：將物件擷取至從屬站緩衝區或從屬站檔案：

```

Content(
    handle,                      /* object handle */
);

```

格式 2：將物件的某區段擷取至從屬站緩衝區或從屬站檔案：

```

Content(
    handle,                      /* object handle */
    offset,                      /* offset where retrieval begins */
    size                         /* number of bytes to retrieve */
);

```

格式 3：將物件擷取至伺服器檔案：

```

Content(
    handle,                      /* object handle */
    target_file,                 /* server file name */
);

```

擷取

```
        overwrite                /* 0=Do not overwrite target file if it exists */
                                   /* 1=Overwrite target file */
    );
```

另外，Content UDF 對影像物件還包括下列格式：

格式 4：將影像擷取至從屬站緩衝區或檔案，並做格式轉換：

```
Content(
    handle,                /* object handle */
    target format          /* target format */
);
```

格式 5：將物件擷取至伺服器檔案，並做格式轉換：

```
Content(
    handle,                /* object handle */
    target_file,           /* server file name */
    overwrite,             /* 0=Do not overwrite target file if it exists */
                           /* 1=Overwrite target file */
    target format          /* target format */
);
```

格式 6：將物件擷取至從屬站緩衝區或檔案，並做格式轉換及額外變更：

```
Content(
    handle,                /* object handle */
    target format,         /* target format */
    conversion_options     /* conversion options */
);
```

格式 7：將物件擷取至伺服器檔案，並做格式轉換及額外變更：

```
Content(
    handle,                /* object handle */
    target_file,           /* server file name */
    overwrite,             /* 0=Do not overwrite target file if it exists */
                           /* 1=Overwrite target file */
    target format,         /* target format */
    conversion_options     /* conversion options */
);
```

例如，下列陳述式會將員工表格中的影像擷取至伺服器的檔案中。(它對應於格式 3。)

```
EXEC SQL SELECT CONTENT(                /* retrieval UDF */
    PICTURE,                            /* image handle */
    '/employee/images/ajones.bmp',     /* target file */
    1)                                  /* overwrite target file */
FROM EMPLOYEE
WHERE NAME = 'Anita Jones';
```

下列在 C 應用程式中的陳述式會將員工表格中的影像擷取至伺服器的檔案中。在擷取影像時會轉換其格式。(它對應於格式 5。)

```
EXEC SQL BEGIN DECLARE SECTION;
    char hvImg_fname[255];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT CONTENT(                /* retrieval UDF */
    PICTURE,                            /* image handle */
    '/employee/images/ajones.bmp',      /* target file */
    1,                                  /* overwrite target file */
    'GIF')                              /* target format */
INTO :hvImg_fname
FROM EMPLOYEE
WHERE NAME = 'Anita Jones';
```

將物件擷取至從屬站

您可以使用 Content UDF，不進行格式轉換而將影像、音效或視訊物件擷取至從屬站緩衝區或檔案中。此外，您可以選擇在擷取影像時，讓 Image Extender 轉換其格式。

將物件擷取至從屬站：不進行格式轉換

使用 LOB 定位器將影像、音效或視訊物件擷取至從屬站緩衝區或擷取 LOB。使用檔案參考變數將影像、音效或視訊物件擷取至 從屬站檔案。

當物件內容以 BLOB 的形式儲存在資料庫表格時，適合使用主變數將影像、音效或視訊物件擷取至從屬站緩衝區，或使用檔案參考變數將影像、音效或視訊物件擷取至從屬站檔案。如果內容是在伺服器檔案中，則從伺服器檔案將內容複製到從屬站檔案會更有效率。

請指定物件的控點。您也可選用從位元組 1 (開始擷取的位置) 開始指定偏移，以及指定要擷取的位元組數。

下列 C 應用程式的陳述式中使用名為 audio_loc 的 LOB 定位器，來將音效片段擷取至從屬站緩衝區。

```
EXEC SQL BEGIN DECLARE SECTION;
    SQL TYPE IS BLOB_LOCATOR audio_loc;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT CONTENT(
    SOUND)                                /* audio handle */
INTO :audio_loc
FROM EMPLOYEE
WHERE NAME = 'Anita Jones';
```

請確定有足夠 UDF 記憶體：將物件擷取至從屬站緩衝區時，您必須確定「資料庫管理程式架構」中的 UDF_MEM_SZ 參數設定為 4MB 或更大。您可以使用 DB2

命令 `UPDATE DATABASE MANAGER CONFIGURATION` 來更新 `UDF_MEM_SZ` 參數。欲獲取有關 `UPDATE DATABASE MANAGER` 命令的其它資訊，請參閱出版品 *DB2 命令參考文件*。

將影像擷取至從屬站：進行轉換

使用 `LOB` 定位器，進行格式轉換而將儲存的影像擷取至從屬站緩衝區中，或擷取 `LOB`。使用檔案參考變數將儲存的影像擷取至從屬站檔案，並做格式轉換。

當影像內容以 `BLOB` 的形式儲存在資料庫表格時，適合使用主變數將影像擷取至從屬站緩衝區，或使用檔案參考變數將影像擷取至從屬站檔案。如果內容是在伺服器檔案中，則從伺服器檔案將內容複製到從屬站檔案會更有效率。

當您進行格式轉換而擷取影像時，您需指定其目標格式，亦即轉換後的格式。第 105 頁的表 8 識別容許的格式轉換。您也可以指定識別諸如旋轉或定標等要引用到擷取影像之額外變更的轉換選項。第 106 頁的表 9 列出可指定的轉換選項。

例如，下列在 `C` 應用程式中的陳述式會將影像擷取至從屬站檔案。來源影像使用點陣圖格式而且以 `BLOB` 的形式儲存在資料庫表格。擷取的影像會被轉換成 `GIF` 格式，而且調整比例成為原來大小的 3 倍。

```
EXEC SQL BEGIN DECLARE SECTION;
    SQL TYPE IS BLOB_FILE Img_file;
EXEC SQL END DECLARE SECTION;

strcpy (Img_file.name, "/employee/images/ajones.gif");
Img_file.name_length= strlen(Img_file.name);
Img_file.file_options= SQL_FILE_CREATE;

EXEC SQL SELECT CONTENT(
    PICTURE,                                /* image handle */
    'GIF',                                  /* target format */
    '-s 3.0')                               /* conversion options */
    INTO :Img_file,
    FROM EMPLOYEE
    WHERE NAME = 'Anita Jones';
```

將物件擷取至伺服器檔案

您可以使用 `Content` UDF，不進行格式轉換而將影像、音效或視訊物件擷取至伺服器檔案。此外，您可以使用 `Content` UDF，進行格式轉換而將影像擷取至伺服器檔案。

當您不進行轉換而將影像、音效或視訊物件擷取至伺服器上的檔案時，請指定物件的控點、目標檔名以及改寫指示符。改寫指示符告知 `Extender`，如果目標檔案已存在伺服器上時，是否要以擷取的資料改寫目標檔案。若目標檔案不存在，`Extender` 會在伺服器建立目標檔案。

如果您指定改寫指示符值 1，則 Extender 會以擷取的資料改寫目標檔案。如果您指定改寫指示符值 0，則 Extender 不會改寫目標檔案，因此而不擷取資料。

如果要擷取的物件被儲存為資料庫表格中的 BLOB，則會忽略改寫指示符。無論指定的改寫指示符為何，都會建立或改寫目標檔案。

當您將物件擷取至伺服器檔案時，它會傳回伺服器檔案的名稱。例如，下列在 C 應用程式中的陳述式會將視訊擷取至伺服器的檔案中。伺服器檔案的檔名則會儲存在主變數 hvVid_fname 中。

```
EXEC SQL BEGIN DECLARE SECTION;
struct{
    short len;
    char data[250];
    }hvVid_fname[;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT CONTENT(
    VIDEO,                                /* video handle */
    '/employee/videos/ajones.mpg',      /* server file */
    1)                                    /* overwrite target file */
    INTO :hvVid_fname;
FROM EMPLOYEE
WHERE NAME = 'Anita Jones';
```

如果物件被儲存為資料庫表格中的 BLOB，則適合使用 Content UDF，不進行轉換而將物件擷取至伺服器檔案中。如果物件是儲存在伺服器檔案中，則將來源檔的內容複製到目標檔會更有效率。

當您進行格式轉換而將影像擷取至伺服器檔案時，請指定影像的控點、目標檔名、改寫目標指示符以及目標格式。第 105 頁的表 8 識別容許的格式轉換。您也可以選擇對目標格式指定 NULL 值或空字串或字串 ASIS。在此情況下，擷取影像的格式會與來源相同。

例如，下列在 C 應用程式中的陳述式會將影像擷取至伺服器的檔案中。來源影像的格式為點陣圖格式，且被儲存為資料庫表格中的 BLOB。擷取的影像被轉換為 GIF 格式。伺服器檔案的檔名則會儲存在主變數 hvImg_fname 中。

```
EXEC SQL BEGIN DECLARE SECTION;
struct{
    short len;
    char [400];
    }hvImg_fname[;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT CONTENT(
    PICTURE,                                /* image handle */
    '/employee/images/ajones.gif',        /* target file */
    1,                                    /* overwrite target file */
```

```
'GIF')                                /* target format */
INTO :hvImg_fname
FROM EMPLOYEE
WHERE NAME = 'Anita Jones';
```

必須是可存取的伺服器檔案：當您將物件擷取至伺服器檔案時，您必須指定目標檔案的完整名稱。另外您必須確定已經適當設定 DB2IMAGEEXPORT、DB2AUDIOEXPORT 和 DB2VIDEOEXPORT 環境變數，以便分辨不完整的檔名規格。

擷取與使用屬性

當您在資料庫中儲存影像、音效或視訊物件時， Extender 會同時在資料庫中儲存物件的屬性。當您更新某物件時，Extender 會更新儲存在資料庫的屬性。您可以在查詢中使用這些屬性。

Extender 會為其管理的每一個屬性建立 UDF。因而您可以在 SQL 陳述式中指定 UDF，來存取及使用物件屬性。表 10 列出 Extender 管理的屬性以及它們的 UDF。它也指出各屬性的物件類型。某些屬性 (如物件的格式和檔名) 是所有物件類型的公用屬性。這些屬性與影像、音效和視訊物件相關。取樣率或壓縮類型等其它屬性則是某些物件類型 (如音效與視訊) 的特定屬性。

表 10. DB2 Extender 所管理的屬性. 您可透過屬性的 UDF 來存取各個屬性。

屬性	UDF	影像	音效	視訊
儲存物件之伺服器檔案的名稱	Filename	x	x	x
儲存物件之人員的使用者 ID	Importer	x	x	x
儲存物件的日期與時間	ImportTime	x	x	x
物件的大小 (以位元組為單位)	Size	x	x	x
上次更新物件之人員的使用者 ID	Updater	x	x	x
上次更新物件的日期與時間	UpdateTime	x	x	x
物件的格式 (例如 GIF 或 MPEG1)	Format	x	x	x
關於物件的註解	Comment	x	x	x
物件的高度 (以圖點計)	Height	x		x
物件的寬度 (以圖點計)	Width	x		x
物件中的顏色數	NumColors	x		
物件的縮圖大小影像	Thumbnail	x		x
音效或視訊的音效磁軌中，每一樣本所傳回的位元組數	AlignValue		x	x
用來呈現各樣本的位元數	BitsPerSample		x	x

表 10. DB2 Extender 所管理的屬性 (繼續)。您可透過屬性的 UDF 來存取各個屬性。

屬性	UDF	影像	音效	視訊
紀錄的通道數	NumChannels		x	x
持續時間 (以秒數計)	Duration		x	x
取樣率 (樣本數/每秒)	SamplingRate		x	x
傳送時間每秒的平均位元組	BytesPerSec		x	
器具的音效磁軌數	FindInstrument		x	
已命名磁軌的磁軌號碼	FindTrackName		x	
記錄的器具名稱	GetInstruments		x	
記錄之器具的磁軌號碼及名稱	GetTrackNames		x	
音效的每秒時鐘滴答數	TicksPerSec		x	
音效的每四分之一註記時鐘滴答數	TicksPerQNote		x	
畫面長寬比	AspectRatio			x
視訊壓縮格式 (如 MPEG1)	CompressType			x
每秒框產量	FrameRate			x
最大產量 (位元組/秒)	MaxBytesPerSec			x
音效磁軌數	NumAudioTracks		x	x
框數	NumFrames			x
視訊磁軌數	NumVideoTracks			x

您可以在 SQL 陳述式 **SELECT** 子句表示式或 **WHERE** 子句搜尋條件中使用屬性 UDF。當您要求 UDF 時，須指定資料庫表格中包含物件控點的直欄名稱。

例如，下列陳述式在 SQL **SELECT** 陳述式的 **SELECT** 子句中使用 **Updater** UDF，來擷取前次更新員工表格中之影像的使用者 ID。

```
EXEC SQL BEGIN DECLARE SECTION;
char hvUpdatr[30];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT UPDATER(PICTURE)
      INTO :hvUpdatr
      FROM EMPLOYEE
      WHERE NAME = 'Anita Jones';
```

下列陳述式在 **SELECT** 陳述式的 **SELECT** 子句中使用 **Filename** UDF，並在 **WHERE** 子句中使用 **NumAudioTracks** UDF，尋找儲存在員工表格中具有音效磁軌的視訊：

使用屬性

```
EXEC SQL BEGIN DECLARE SECTION;
char hvVid_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(VIDEO)
      INTO :hvVid_fname
      FROM EMPLOYEE
      WHERE NUMAUDIOTRACKS(VIDEO)>0;
```

擷取註解

使用 **Comment UDF** 來擷取與影像、音效或視訊物件一起儲存的註解。當您擷取某個物件的註解時，須指定資料庫表格中包含物件控點的直欄。例如，下列陳述式會擷取與員工表格中的音效擷取同時儲存的註解。

```
EXEC SQL BEGIN DECLARE SECTION;
struct {
    short len;
    char data[32700];
}hvComment
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT COMMENT(SOUND)
      INTO :hvComment
      FROM EMPLOYEE
      WHERE NAME = 'Anita Jones';
```

您也可以使用 **Comment UDF** 作為 **SQL** 查詢之 **WHERE** 子句中的述詞。例如，下列陳述式會擷取員工表格中所有被註記為『touched up』之影像的檔名。

```
EXEC SQL BEGIN DECLARE SECTION;
struct {
    short len;
    char data[250];
}hvImg_fname
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
      INTO :hvImg_fname
      FROM EMPLOYEE
      WHERE COMMENT(PICTURE)
            LIKE '%touch%up';
```

更新影像、音效或視訊物件

在 **SQL UPDATE** 陳述式中使用 **Content UDF**，來更新資料庫表格中的影像、音效或視訊物件。在 **SQL UPDATE** 陳述式中使用 **Replace UDF**，來更新資料庫表格中的影像、音效或視訊，並且更新與物件相關的註解。在上述兩種情況之下，**Extender** 都會更新與物件相關的屬性。

您可以更新儲存為資料庫表格中之 **BLOB** 的物件，或更新儲存在伺服器檔案（以及從資料庫指向的檔案）中的物件。更新的來源可以位於緩衝區、從屬站檔案、或伺服器檔案。

第 105 頁的表 8 列出可更新影像、音效或視訊物件的格式。不過，您也可以更新其格式無法為 **Extender** 所識別的物件。在此情況下，使用者會在儲存物件時指定物件的屬性。以使用者指定的屬性更新某物件時，您必須指定該物件的更新屬性。在 **SQL UPDATE** 陳述式中使用 **ContentA** UDF，並以資料庫中使用者提供的屬性來更新影像、音效或視訊物件。在 **SQL UPDATE** 陳述式中使用 **ReplaceA** UDF，並以資料庫表格中的使用者提供的屬性來更新影像、音效或視訊，並且更新與物件相關的註解。當您以使用者指定的屬性更新某物件時，您必須指定該物件的屬性及格式，另外，就視訊物件而言，您還必須指定該物件的壓縮格式。

您也可以為儲存的影像或視訊更新縮圖。

確定更新作業：在資料庫中更新影像、音效或視訊物件之後，需要確定工作單元。這樣會解除 **Extender** 所保留的鎖定，使您可以對儲存的物件執行後續更新作業。

用於更新的 **Content** UDF 格式

Content UDF 已超載，這表示可根據 UDF 的使用方式，決定不同的格式。其格式如下：

格式 1：更新來自從屬站緩衝區或從屬站檔案的物件：

```
Content(
    handle,                /* object handle */
    content,               /* object content */
    source_format,         /* source format */
    target_file            /* target file name for storage in file */
                        /* server or NULL for storage in table as BLOB */
);
```

格式 2：更新來自伺服器檔案的物件：

```
Content(
    handle,                /* object handle */
    source_file,           /* server file name */
    source_format,         /* source format */
    stortype               /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                        /* in file server */
                        /* MMDB_STORAGE_TYPE_INTERNAL=store as a BLOB*/
);
```

格式 3：更新從屬站緩衝區或檔案中具有使用者提供之屬性的物件：

```
Content(  
    handle,                /* object handle */  
    content,               /* object content */  
    target_file,           /* target file name for storage in file server */  
                           /* or NULL for storage in table as BLOB */  
    attrs,                /* user-supplied attributes */  
    thumbnail              /* thumbnail (image and video only) */  
);
```

格式 4：更新伺服器檔案中具有使用者提供之屬性的物件：

```
Content(  
    handle,                /* object handle */  
    source_file,           /* source file name */  
    stortype,              /* MMDB_STORAGE_TYPE_EXTERNAL=store */  
                           /* in file server*/  
                           /* MMDB_STORAGE_TYPE_INTERNAL=store */  
                           /* as a BLOB*/  
    attrs,                /* user-supplied attributes */  
    thumbnail              /* thumbnail (image and video only) */  
);
```

單就影像物件而言，Content UDF 具有下列附加的格式：

格式 5：更新來自從屬站緩衝區或從屬站檔案的影像，同時做格式轉換：

```
Content(  
    handle,                /* object handle */  
    content,               /* object content */  
    source format,         /* source format */  
    target format,         /* target format */  
    target_file            /* target file name for storage in file server */  
                           /* or NULL for storage in table as BLOB */  
);
```

格式 6：更新來自伺服器檔案，並做格式轉換的物件：

```
Content(  
    handle,                /* object handle */  
    source_file,           /* server file name */  
    source format,         /* source format */  
    target format,         /* target format */  
    target_file            /* target file name for storage in file server */  
                           /* or NULL for storage in table as BLOB */  
);
```

格式 7：更新來自從屬站緩衝區或從屬站檔案的影像，同時做格式轉換及其他變更：

```
Content(  
    handle,                /* object handle */  
    content,               /* object content */  
    source format,         /* source format */  
    target format,         /* target format */
```

```

conversion_options,      /* conversion options */
target_file              /* target file name for storage in file server */
                          /* or NULL for storage in table as BLOB */
);

```

格式 8：更新來自伺服器檔案，並做格式轉換和額外變更的物件：

```

Content(
    handle,                /* object handle */
    source_file,           /* server file name */
    source_format,         /* source format */
    target_format,         /* target format */
    conversion_options,    /* conversion options */
    target_file            /* target file name for storage in file server */
                          /* or NULL for storage in table as BLOB */
);

```

例如，下列在 C 應用程式中的陳述式會更新在員工表格中的影像。更新的來源內容為名為 `ajones.bmp` 的伺服器檔案。更新的影像以 **BLOB** 的形式儲存在員工表格。(它對應於上述格式 2。)

```

EXEC SQL UPDATE EMPLOYEE
SET PICTURE=CONTENT(
    PICTURE,                /*image handle*/
    '/employee/newimg/ajones.bmp', /*source file */
    'ASIS',                /*keep the image format*/
    '');                  /*store image in DB as BLOB*/
WHERE NAME='Anita Jones';

```

下列在 C 應用程式中的陳述式會更新前例中的同一個影像。但此處在更新影像時，其格式 **BMP** 被轉換成 **GIF**。(它對應於上述格式 6。)

```

EXEC SQL UPDATE EMPLOYEE
SET PICTURE=CONTENT(
    PICTURE,                /*image handle*/
    '/employee/newimg/ajones.bmp', /*source file */
    'BMP',                  /*source format*/
    'GIF',                  /*target format*/
    '');                  /*store image in DB as BLOB*/
WHERE NAME='Anita Jones';

```

用於更新的 **Replace UDF** 格式

Replace UDF 已超載，也就是說，可根據使用 **UDF** 的方式，決定不同的格式。其格式如下：

格式 1：更新來自從屬站緩衝區或從屬站檔案的物件，並更新該物件的註解：

```

Replace(
    handle,                /* object handle */
    content,               /* object content */

```

更新

```
        source_format,          /* source format */
        target_file,           /* target file name for storage in file */
        comment                 /* user comment */
    );
```

格式 2：更新來自伺服器檔案的物件，並更新該物件的註解：

```
Replace(
    handle,                    /* object handle */
    source_file,               /* server file name */
    source_format,             /* source format */
    stortype,                  /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                                /* in file server*/
                                /* MMDB_STORAGE_TYPE_INTERNAL=store as a BLOB*/
    comment                    /* user comment */
);
```

格式 3：取代來自從屬站緩衝區或從屬站檔案，且具有使用者提供的屬性的物件，並更新該物件的註解：

```
Replace(
    handle,                    /* object handle */
    content,                   /* object content */
    target_file,               /* target file name for storage in file */
                                /* or NULL for storage in table as BLOB */
    comment,                   /* user comment */
    attrs,                     /* user-supplied attributes */
    thumbnail                  /* thumbnail */
);
```

格式 4：儲存伺服器檔案中具有使用者提供之屬性的物件：

```
Replace(
    handle,                    /* object handle */
    source_file,               /* server file name */
    stortype,                  /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                                /* in file server*/
                                /* MMDB_STORAGE_TYPE_INTERNAL=store as a BLOB*/
    comment,                   /* user comment */
    attrs,                     /* user-supplied attributes */
    thumbnail                  /* thumbnail */
);
```

單就影像物件而言，Replace UDF 具有下列附加的格式：

格式 5：更新來自從屬站緩衝區或從屬站檔案的影像，同時做格式轉換，並更新該影像的註解：

```
Replace(
    handle,                    /* object handle */
    content,                   /* object content */
    source_format,             /* source format */
    target_format,             /* target format */
);
```

```

        target_file,                /* target file name for storage in file server */
                                     /* or NULL for storage in table as BLOB */
        comment                     /* user comment */
    );

```

格式 6：更新來自伺服器檔案，並做格式轉換的物件，並更新該物件的註解：

```

Replace(
    handle,                        /* object handle */
    source_file,                   /* server file name */
    source_format,                 /* source format */
    target_format,                 /* target format */
    target_file,                   /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                                     /* in file server */
                                     /* MMDB_STORAGE_TYPE_INTERNAL=store as a BLOB*/
    comment                        /* user comment */
);

```

格式 7：更新來自從屬站緩衝區或從屬站檔案的影像，同時做格式轉換及其他變更，並更新該影像的註解：

```

Replace(
    handle,                        /* object handle */
    content,                       /* object content */
    source_format,                 /* source format */
    target_format,                 /* target format */
    conversion_options,            /* conversion options */
    target_file,                   /* target file name for storage in file server */
                                     /* or NULL for storage in table as BLOB */
    comment                        /* user comment */
);

```

格式 8：更新來自伺服器檔案，並做格式轉換和額外變更的物件，並更新該物件的註解：

```

Replace(
    handle,                        /* object handle */
    source_file,                   /* server file name */
    source_format,                 /* source format */
    target_format,                 /* target format */
    conversion_options,            /* conversion options */
    target_file,                   /* MMDB_STORAGE_TYPE_EXTERNAL=store */
                                     /* in file server */
                                     /* MMDB_STORAGE_TYPE_INTERNAL=store as a BLOB*/
    comment                        /* user comment */
);

```

例如，下列在 C 應用程式中的陳述式會更新員工表格中的音效片段，並更新其相關的註解。更新的來源內容為名為 ajones.wav 的伺服器檔案。更新的音效片段未經格式轉換，被儲存為員工表格中的 BLOB (Audio Extender 不支援格式轉換)。它對應於上述格式 2。

更新

```
EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

EXEC SQL UPDATE EMPLOYEE
    SET SOUND=REPLACE(
        SOUND,                                /*audio handle*/
        '/employee/newaud/ajones.wav',        /*source file */
        'WAV',                                /*keep the audio format*/
        :hvStorageType,                        /*store audio in DB as BLOB*/
        'Anita''s new greeting')              /*user comment*/
    WHERE NAME= 'Anita Jones';
```

在下列範例中，影像及其相關的註解都被更新。更新的來源內容在伺服器檔案中。更新的影像被儲存為員工表格中的 BLOB，且其格式在更新時從 BMP 被轉換成 GIF 格式。(它對應於上述格式 6。)

```
EXEC SQL UPDATE EMPLOYEE
    SET PICTURE=REPLACE(
        PICTURE,                                /*image handle*/
        '/employee/newimg/ajones.bmp',        /*source file */
        'BMP',                                /*source format*/
        'GIF',                                /*target format*/
        ''                                     /*store image in DB as BLOB*/
        'Anita''s new picture')
    WHERE NAME='Anita Jones';                /* user comment */
```

更新從屬站的物件

使用主變數或檔案參考變數，來更新從屬站緩衝區或從屬站檔案

如果更新的來源是在從屬站檔案中，請使用檔案參考變數來傳送其內容。例如，下列在 C 應用程式中的陳述式會定義名為 `Audio_file` 的檔案參考變數，並使用該變數來更新儲存為資料庫表格中之 BLOB 的音效片段。更新的來源在從屬站檔案中。請注意：檔案參考變數的 `file_options` 欄位被設定為 `SQL_FILE_READ`，也就是說專供輸入。另請注意檔案參考變數被用來作為對 Content UDF 的內容引數。

```
EXEC SQL BEGIN DECLARE SECTION;
    SQL TYPE IS BLOB FILE Audio_file;
EXEC SQL END DECLARE SECTION;

strcpy (Audio_file.name, "/employee/newsound/ajones.wav");
Audio_file.name_length= strlen(Audio_file.name);
Audio_file.file_options= SQL_FILE_READ;

EXEC SQL UPDATE EMPLOYEE
    SET SOUND=CONTENT(
        SOUND,
        :Audio_file                                /*file reference variable*/
```



```

        'WAVE',                                /*keep the image format*/
        CAST(NULL as LONG VARCHAR))

WHERE NAME='Anita Jones';

```

如果物件是在從屬站緩衝區中，請使用主變數來傳送其內容，以進行更新。在下列的 C 應用程式範例中，會使用名為 **Video_seg** 的主變數來傳送視訊片段的內容，以進行更新。同時也會更新與視訊片段相關的註解。視訊片段被儲存為資料庫表格中的 **BLOB**。請注意：主變數被用來作為對 **Replace** UDF 的內容引數。

```

EXEC SQL BEGIN DECLARE SECTION;
    SQL TYPE IS BLOB (2M) Video_seg
EXEC SQL END DECLARE SECTION;

EXEC SQL UPDATE EMPLOYEE
    SET VIDEO=REPLACE(
        VIDEO,
        :Video_seg                                /*host variable*/
        'MPEG1',
        CAST(NULL as LONG VARCHAR),

        'Anita''s new video')
WHERE NAME='Anita Jones';

```

請確定有足夠 UDF 記憶體：更新其內容是在從屬站緩衝區的物件時，您必須確定「資料庫管理程式架構」中的 **UDF_MEM_SZ** 參數設定為 **4MB** 或更大。您可以使用 **DB2** 命令 **UPDATE DATABASE MANAGER CONFIGURATION** 來更新 **UDF_MEM_SZ** 參數。

更新伺服器的物件

如果影像、音效或視訊物件更新的來源內容是在伺服器檔案中，請將該檔案路徑指定為對 **UDF** 的內容引數。例如，下列在 C 應用程式中的陳述式會更新資料庫中的影像。影像內容位於伺服器檔案。資料庫會指向伺服器檔案。更新的來源也在伺服器檔案中。

```

EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDb_STORAGE_TYPE_EXTERNAL;

EXEC SQL UPDATE EMPLOYEE
    SET PICTURE=CONTENT(
        PICTURE,                                /* image handle */
        '/employee/newimg/ajones.bmp',          /* source file */
        'ASIS',
        :hvStorageType)
WHERE NAME='Anita Jones';

```

指定正確的路徑：當您更新其來源在伺服器檔案的物件時，您可以指定檔案的完整名稱或相對名稱。如果是指定相對名稱，您需要確定 DB2 伺服器中的適當環境變數包括檔案的正確路徑。有關設定這些環境變數的資訊，請參閱第 543 頁的附錄 A，『設定 DB2 Extender 的環境變數』。

指定進行更新的資料庫或檔案儲存體

您可以更新儲存為資料庫表格中之 BLOB 或在伺服器檔案（以及從資料庫指向的檔案）中的影像、音效或視訊物件。

若您更新來自從屬站緩衝區或從屬站檔案的物件，請依照您在檔名參數中指定的結果，指示 BLOB 或伺服器檔案儲存體。如果是指定檔名，表示您要更新其內容是在伺服器檔案中的物件。如果是指定 NULL 檔名，表示您要更新儲存為資料庫表格中之 BLOB 的物件。

例如，下列在 C 應用程式中的陳述式會更新其內容是在伺服器檔案中的影像。更新來源在從屬站緩衝區中。同時也會更新影像註解。

```
EXEC SQL BEGIN DECLARE SECTION;
    SQL TYPE IS BLOB (2M) Img_buf
EXEC SQL END DECLARE SECTION;

EXEC SQL UPDATE EMPLOYEE
    SET PICTURE=REPLACE(
        PICTURE,
        :Img_buf,
        'ASIS',
        '/employee/newimg/ajones.bmp',      /*update image in*/
                                           /*server file*/
        'Anita's new picture')
    WHERE NAME='Anita Jones';
```

若更新來自伺服器檔案的物件，請指定 `MMDB_STORAGE_TYPE_INTERNAL`，更新以 BLOB 的形式儲存在資料庫表格中的物件。如果您要更新其內容是在伺服器檔案中的物件，請指定 `MMDB_STORAGE_TYPE_EXTERNAL`。

例如，在下列 C 應用程式中會更新音效片段。音效片段的內容在伺服器檔案中。更新的來源也在伺服器檔案中。

```
EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL UPDATE EMPLOYEE
    SET SOUND=CONTENT(
        SOUND,
```

```

        '/employee/newimg/ajones.wav',
        'WAVE',
        :hvStorageType)          /*update audio in server file*/
WHERE NAME='Anita Jones';

```

識別更新的格式

當您更新物件時，您需要識別其格式。Extender 將以與來源相同的格式儲存更新的影像、音效或視訊物件。如果是影像物件，您可以選擇讓 Image Extender 轉換更新之影像的格式。若要轉換影像格式，您必須指定更新來源的格式以及目標影像的格式。目標影像是儲存時的更新影像。

識別更新的格式；不進行轉換

當您更新物件而不進行格式轉換時，請指定來源影像、音效或視訊物件的格式。例如，下列在 C 應用程式中的陳述式會更新其內容是在伺服器檔案中的點陣圖 (BMP) 影像。將不會轉換更新之影像的格式。

```

EXEC SQL UPDATE EMPLOYEE
SET PICTURE=CONTENT(
    PICTURE,
    '/employee/newimg/ajones.bmp',
    'BMP',
    '')
WHERE NAME='Anita Jones';

```

您也可以指定 NULL 值或空白字串為格式，或對 Image Extender 指定字串 ASIS。然後 Extender 會檢查來源來決定格式。

對可辨識的格式使用 NULL 或 ASIS： 只在格式可為 Extender 所辨識 (亦即 Extender 列在第 105 頁的表 8 中的其中一個格式) 的情況下，指定 NULL 值、空白字串或 ASIS。否則，Extender 無法更新物件。

識別更新的格式及轉換選項；進行格式轉換

當您進行格式轉換而更新影像時，請指定來源與目標影像的格式。第 105 頁的表 8 列出容許的格式轉換。

此外，您可以指定轉換選項藉以識別諸如旋轉或壓縮等額外變更 (變更的對象為已更新的影像)。您可透過參數及相關值來指定各轉換選項。第 106 頁的表 9 中會列出參數及容許的值。您可以指定多組參數/值對，對更新的影像要求多重變更。

在下列範例中，會更新其內容是在伺服器檔案中的影像。更新之來源的格式為點陣圖 (BMP) 格式。在更新時，會從 BMP 格式轉換成 GIF 格式。

```

EXEC SQL UPDATE EMPLOYEE
SET PICTURE=CONTENT(
    PICTURE,
    '/employee/newimg/ajones.bmp',

```

```

        'BMP',                                /*source format*/
        'GIF',                                /*target format*/
        '')
WHERE NAME='Anita Jones';

```

在下列範例中，同一個影像會在更新時轉換成 GIF 格式。此外，影像會在更新時順時針旋轉 90 度。

```

EXEC SQL UPDATE EMPLOYEE
SET PICTURE=CONTENT(
    PICTURE,
    '/employee/newimg/ajones.bmp',
    'BMP',                                /*source format*/
    'GIF',                                /*target format*/
    '-r 1',                               /* conversion options */
    '')
WHERE NAME='Anita Jones';

```

更新具有使用者提供之屬性的物件

當您更新連同使用者提供之屬性一起儲存的影像、音效或視訊物件時，您必須指定更新內容的屬性。請在某個屬性結構中指定屬性值。屬性結構必須儲存在 UDF 之 LONG VARCHAR FOR BIT DATA 變數的資料欄位中。

伺服器上的 UDF 碼會預期資料為『big endian 格式』。Big endian 格式是大部份 UNIX 平台使用的格式。若使用『little endian 格式』儲存某物件，您必須準備使用者提供的屬性資料，以便伺服器上的 UDF 碼能夠正確處理該物件。Little endian 格式是通常用於 Intel 和其它微處理器平台的格式。（即使您沒有使用 little endian 格式儲存物件，仍然建議您準備使用者提供的屬性資料。）使用 DBiPrepareAttrs API 來準備影像物件的屬性。使用 DBaPrepareAttrs API 來準備音效物件的屬性。使用 DBvPrepareAttrs API 來準備視訊物件的屬性。

例如，下列在 C 應用程式中的陳述式會更新其內容是在伺服器檔案中的影像。該影像具有使用者定義的格式：高 640 圖點、寬 480 圖點。請注意：要在更新影像之前準備屬性。

```

EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
struct {
    short len;
    char data[400];
    }hvImgattrs;
EXEC SQL END DECLARE SECTION;

DB2IMAGEATTRS    *pimgattr;

hvStorageType=MMDB_STORAGE_TYPE_INTERNAL;

pimgattr = (DB2IMAGEATTRS *) hvImgattrs.data;
strcpy(pimgattr->Format,"FormatI");

```

```

pimgattr->width=640;
pimgattr->height=480;
hvImgattrs.len=sizeof(DB2IMAGEATTRS);

DBiPrepareAttrs(pimgattr);

EXEC SQL UPDATE EMPLOYEE
SET PICTURE=REPLACE(
    PICTURE,
    '/employee/newimg/ajones.bmp',
    :hvStorageType,
    'Anita's new picture',
    :ImgAttrs, /*user-supplied attributes*/
    CAST(NULL as LONG VARCHAR))
WHERE NAME='Anita Jones';

```

更新縮圖 (影像與視訊)

使用 Thumbnail UDF 來更新對影像或視訊物件所儲存的縮圖 (如果沒有與儲存之影像或視訊相關的縮圖，請新增一個)。當您使用 Thumbnail UDF 時，請指定已更新其縮圖的物件控點，並指定更新 (或新建) 之縮圖的內容。

請在您的程式中建立縮圖 -- Extender 並不會提供 API 來產生縮圖。您自行控制更新縮圖的大小及格式。在您的程式中建立縮圖的結構，並指定 UDF 中的縮圖結構。

例如，下列在 C 應用程式中的陳述式會更新與儲存之視訊片段相關的縮圖。

```

EXEC SQL BEGIN DECLARE SECTION;
    struct {
        short len;
        char data[10000];
    }hvThumbnail;
EXEC SQL END DECLARE SECTION;

/*Create thumbnail and store in hvThumbnail*/

EXEC SQL UPDATE employee
SET picture=Thumbnail(
    picture,
    :hvThumbnail)
WHERE name='Anita Jones';

```

當您更新具有使用者提供之屬性的影像或視訊物件時，您也可以更新縮圖。事實上，如果您更新具有使用者提供之屬性的影像或視訊，則您必須指定縮圖為輸入。若不要在更新物件時更新縮圖，請指定 NULL 值或空字串來代替縮圖。

下列在 C 應用程式中的陳述式會更新具有使用者提供之屬性的視訊片段，並更新與視訊相關的縮圖。

更新

```
EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
    struct {
        short len;
        char data[400];
    }hvVidattrs;
    struct {
        short len;
        char data[10000];
    }hvThumbnail;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

MMDBVideoAttrs      *pvideoAttr;
pvideoAttr=(MMDBVideoAttrs *)hvVidattrs.data;
strcpy(pvideoAttr->cformat,"Formatv");
hvVidattrs.len=sizeof(MMDBVideoAttrs);

/* Update video content and thumbnail */

EXEC SQL UPDATE EMPLOYEE
    SET VIDEO=REPLACE(
        VIDEO,
        '/employee/newvid/ajones.mpg',
        :hvStorageType,
        'Anita''s new video',
        :VidAttrs,
        :hvThumbnail)          /*thumbnail*/
    WHERE NAME='Anita Jones';
```

更新註解

您可以單獨更新註解，或在更新註解的相關物件時更新註解。

使用 **Comment UDF** 來單獨更新註解。指定更新之註解的內容，以及包含物件之控點的表格直欄。使用主變數將內容傳送到伺服器。例如，下列陳述式宣告主變數 **hvRemarks**，並使用它來更新儲存的視訊片段的現存註解。

```
EXEC SQL BEGIN DECLARE SECTION;
    struct {
        short len;
        char data [40];
    }hvRemarks;
EXEC SQL END DECLARE SECTION;

/* Get the old comment */

EXEC SQL SELECT COMMENT(VIDEO)
    INTO :hvRemarks
    FROM EMPLOYEE
    WHERE NAME = 'Anita Jones';

/* Append to old comment */
```

```

hvRemarks.data[Remarks.len]='\0';
hvRemarks.len=strlen(hvRemarks.data);
strcat (hvRemarks.data, "Updated video");
EXEC SQL UPDATE EMPLOYEE
    SET VIDEO=COMMENT(VIDEO, :hvRemarks)
    WHERE NAME = 'Anita Jones';

```

當您更新註解的相關物件時，請使用 **Replace UDF** 來更新註解。例如，下列陳述式更新儲存在伺服器檔案的視訊片段以及它的相關註解。

```

EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType=MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL UPDATE EMPLOYEE
    SET VIDEO=REPLACE(
        VIDEO,
        '/employee/newvid/ajones.mpg',
        'MPEG1',
        :hvStorageType,
        'Anita's new video')      /*updated comment*/
    WHERE NAME='Anita Jones';

```

更新

第 11 章 顯示或執行影像、音效或視訊物件

本章說明如何使用 DB2 Extender 應用程式設計介面，來顯示或播放儲存在資料庫中的影像、音效或視訊物件。

使用顯示或播放 API

您可以使用 Extender API 來顯示資料庫中儲存的影像或視訊框。您可以將影像或視訊框顯示成縮圖或最大圖。您也可以使用 Extender API 來播放資料庫中儲存的音效或視訊物件。

請使用下列 API 來顯示或播放物件：

使用此 API	來
DBiBrowse	顯示影像或視訊框
DBaPlay	播放音效片段
DBvPlay	播放視訊片段或顯示視訊框

若要求任何這些 API，則需要指定下列事項：

- 顯示或播放程式的名稱
- 欲顯示或播放的物件是在資料庫表格儲存為 BLOB，或是位於表格所參照的檔案中
- 來源檔名稱，或儲存在資料庫表格的控點
- 在繼續作業之前，是否讓您的應用程式等待使用者去關閉顯示或播放程式

識別顯示或播放程式

指定您要使用之影像瀏覽程式、音效或視訊播放程式的名稱。請在名稱後面輸入 %s，Extender 會以含有物件內容的檔案來取代 %s。

您也可以指定 NULL 值，而不用指名特定的顯示或播放程式。在此情形下，Extender 會啟動 DB2IMAGEBROWSER、DB2AUDIOPLAYER 或 DB2VIDEOPLAYER 環境變數中，指定的預設影像瀏覽程式、音效或視訊播放程式。關於 DB2 Extender 使用環境變數的方式，請參閱第 543 頁的附錄 A，『設定 DB2 Extender 的環境變數』。

使用顯示/播放 API

例如，下列 C 應用程式中的陳述式，會啟動 DB2AUDIOPLAYER 環境變數所指定的預設音效播放程式：

```
rc = DBaPlay(  
    NULL,                                /* use default audio player */  
    MMDB_PLAY_FILE,  
    "/employee/sounds/ajones.wav",  
    MMDB_PLAY_NO_WAIT  
);
```

環境變數必須指名一個程式：若要預設的顯示或播放程式 (指定 NULL 值)，請確定在適當的環境變數中指定顯示或播放程式。若未指定程式，則 API 將傳回錯誤碼。

指定 BLOB 或檔案內容

您可以顯示或播放放在資料庫表格中儲存為 BLOB 的物件，或其內容儲存於檔案中的物件 (從資料庫表格來參照)。若物件儲存為 BLOB，請指定 MMDB_PLAY_HANDLE。若物件內容儲存於檔案中，請指定 MMDB_PLAY_FILE。MMDB_PLAY_HANDLE 和 MMDB_PLAY_FILE 是 Extender 定義的常數。

例如，下列 C 應用程式中的陳述式，會播放內容位於檔案中的視訊：

```
rc = DBvPlay(  
    "explore %s",  
    MMDB_PLAY_FILE,                    /* content in file */  
    "/employee/videos/ajones.mpg",  
    MMDB_PLAY_NO_WAIT  
);
```

顯示或播放程式通常接受檔案的輸入。若指定 MMDB_PLAY_FILE，Extender 會使用環境變數中的值，來解析檔案的相對檔名和路徑。然後 Extender 會啟動瀏覽程式並將檔名傳給它。若指定 MMDB_PLAY_HANDLE，Extender 會從控點取出檔名 (假設檔名不是空字元)。若控點中的檔名是空字元，物件會儲存為 BLOB。Extender 會在從屬站中建立暫用檔，並將物件內容從資料庫表格複製到此從屬站檔案。然後 Extender 會啟動程式，並將保留內容的檔案 (或暫用檔) 名稱傳給程式。

例如，下列 C 應用程式中的陳述式，會取得儲存為 BLOB 的影像控點，並使用該控點來顯示影像：

```
EXEC SQL BEGIN DECLARE SECTION;  
char hvImg_hdl [251];  
EXEC SQL END DECLARE SECTION;
```

```
rc = DBiBrowse(  
    "ib %s",
```

```

MMDB_PLAY_HANDLE,                /* content is BLOB */
hvImg_hdl,
MMDB_PLAY_NO_WAIT
);

```

內容必須是可存取的：請確定顯示或播放程式能夠存取物件內容。若內容位於伺服器檔案中，但程式要求內容須位於從屬站上，則請將檔案複製到從屬站上，或使用 Content UDF。若內容儲存為 BLOB，則 Extender 將自動擷取至從屬站。

指定等待指示符

您可以指定在應用程式繼續執行之前，是否讓您的應用程式等待使用者去結束顯示或播放程式。若要讓您的應用程式等待，請指定 `MMDB_PLAY_WAIT`。若不要讓您的應用程式等待，請指定 `MMDB_PLAY_NO_WAIT`。`MMDB_PLAY_WAIT` 和 `MMDB_PLAY_NO_WAIT` 是 Extender 所定義的常數。

若指定 `MMDB_PLAY_WAIT`，則顯示或播放程式將於應用程式相同的緒或程序中執行。若指定 `MMDB_PLAY_NO_WAIT`，則顯示或播放程式將在不同於應用程式的緒或程序中獨立執行。

例如，下列陳述式的結果，應用程式在繼續執行之前，會等待使用者關閉影像瀏覽程式：

```

rc = DBiBrowse(
    "explore %s",
    MMDB_PLAY_FILE,
    "/employee/images/ajones.bmp",
    MMDB_PLAY_WAIT                /* wait for browser to close */
);

```

若指定 `DBxPlay` 和 `MMDB_PLAY_NO_WAIT` 時，請留意：發出 `DBaPlay` 或 `DBvPlay` 時，若下列任何一項為真，Extender 會建立暫用檔：

- 物件儲存為 BLOB
- 使用環境變數中的值無法解析相對檔名
- 在從屬站機器上無法存取檔案

該暫用檔會建立於 `TMP` 環境變數所指定的目錄內。若指定 `MMDB_PLAY_WAIT`，則 Extender 會於播放物件之後，刪除該暫用檔。相反地，若指定 `MMDB_PLAY_NO_WAIT`，則不會刪除暫用檔。您必須要親自刪除該暫用檔。

顯示縮圖影像或視訊框

縮圖是指影像或視訊框的縮小版。將影像儲存在資料庫時，Image Extender 會在屬性表格中儲存影像的縮圖。將視訊儲存在資料庫時，Video Extender 會將象徵視訊物件的同屬縮圖儲存在屬性表格。

在預設情況下，Image Extender 自動建立的影像縮圖大小大約是 112 x 84 圖點。Video Extender 插入的同屬視訊縮圖大小是 108 x 78 圖點。影像縮圖和同屬視訊縮圖都是以 GIF 格式儲存。根據影像或視訊框中的資料密度，縮圖的約為 4.5 KB 至 5 KB 的資料。若以使用者提供的屬性來儲存或更新影像或視訊，則您可以選擇縮圖的大小和格式。

在 SQL SELECT 陳述式中使用 Thumbnail UDF，可從資料庫擷取縮圖。使用檔案參考變數將縮圖傳送到檔案。指定 UDF 時，您需要指定資料庫表格中含有影像或視訊控點的直欄名稱。然後，使用 DBiBrowse API 來顯示影像或視訊框縮圖。

例如，下列陳述式會擷取縮圖影像，然後顯示出來：

```
long rc, outCount;
char Thumbnail_filename[254];
FILE *file_handle;

EXEC SQL BEGIN DECLARE SECTION;
struct {
    short len
    char data[10000];
}Thumbnail_buffer;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT THUMBNAIL(PICTURE)
    INTO :Thumbnail_buffer
    FROM EMPLOYEE
    WHERE NAME = 'Anita Jones';

strcpy (Thumbnail_filename, "/tmp/ajones.tmb");
file_handle=fopen(Thumbnail_filename,"wb+");
outCount=fwrite(Thumbnail_buffer.data, 1, Thumbnail_buffer.len, file_handle);
fclose(file_handle);
rc = DBiBrowse(
    NULL,                                /* use the default display program */
    MMDB_PLAY_FILE,                      /* thumbnail image in file */
    Thumbnail_filename,                  /* thumbnail image content */
    MMDB_PLAY_WAIT);                    /* wait for user to finish */
```

顯示最大圖影像或視訊框

使用 DBiBrowse API 顯示儲存在資料庫表格的影像。關於使用此 API 的詳細資訊，請參閱第 141 頁的『使用顯示或播放 API』。

請使用 DBvGetNextFrame API 或 DBvSeekFrame API 來取得最大圖視訊框。框是以 YUV 格式儲存於緩衝區中，可使用 DBvFrameDatato24BitRGB API 來轉換成 RGB 格式。添加標頭到已轉換的框 (例如，添加 BMP 檔案類型標頭)，並將標頭和框資料寫入檔案中。然後，使用 DBiBrowse API 來顯示檔案的內容。有關使用 DBvGetNextFrame、DBvSeekNextFrame 及 DBvFrameDatato24BitRGB API 的詳細資訊，以及有關顯示視訊框的進一步資訊，請參閱第 19 頁的『偵測視訊場景變更』。

播放音效或視訊

使用 DBaPlay API 播放儲存在資料庫表格的音效。使用 DBvPlay API 播放儲存在資料庫表格的視訊。關於使用這些 API 的詳細資訊，請參閱第 141 頁的『使用顯示或播放 API』。

第 12 章 依內容來查詢影像

圖 27顯示一種應用程式，可供使用者將視覺化範例當作搜尋準則，來搜尋資料庫中的影像，也就是說，示範主色或紋理型樣的影像。透過這類應用程式，使用者可提供影像作為搜尋的輸入檔。然後，應用程式使用來源影像的顏色或紋理來比對儲存影像的顏色或紋理，並傳回顏色或紋理最符合輸入檔的影像。

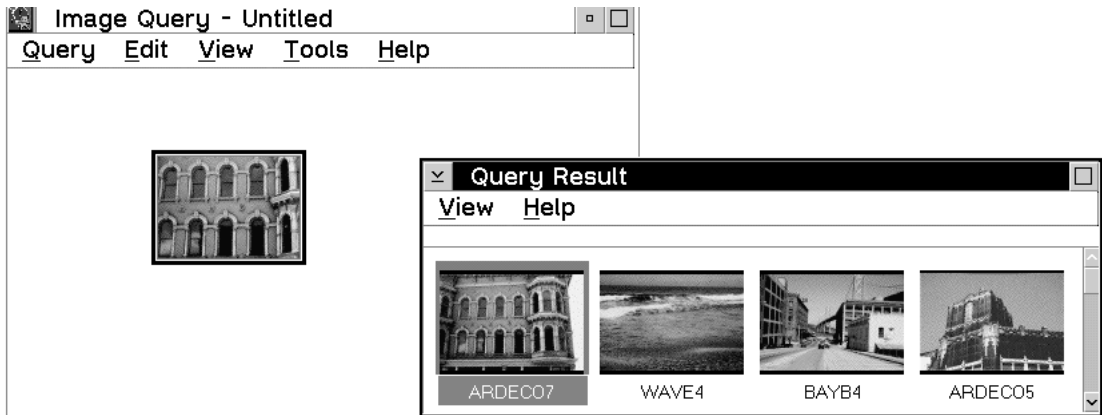


圖 27. 按影像內容來查詢。您可使用視覺化範例的顏色或紋理，來搜尋儲存在資料庫表格中的影像。

透過影像視覺化特性來查詢影像的功能，稱為**按影像內容查詢 (QBIC)**⁶。本章說明如何使用 Image Extender 提供的 API 和 UDF，來建立上述的應用程式。本章亦說明如何使用 Image Extender 提供的命令和 API，來執行 QBIC 管理作業。

如何按影像內容查詢

按影像內容查詢：

1. 建立影像的 QBIC 型錄。
 2. 將影像編入型錄中。意指將影像項目加入型錄中，以及儲存影像特性的值。
- 有關 QBIC 型錄及影像特性的說明，請參閱第 57 頁的『QBIC 型錄』。

6. Image Extender 包含加州大學柏克萊分校及其贊助者所研發的軟體。

如何按內容查詢

3. 建立查詢。此查詢會識別將用來當作搜尋準則的特性、它們的值及它們的比重(亦即，每個特性的強調點)。您可以在稱為查詢字串的字串中指定這些查詢屬性。另外，您可以建立查詢物件及與其相關的屬性。然後就可以儲存查詢子串並重覆使用。
4. 執行查詢。執行查詢時，您可以將查詢字串指定為輸出，或識別查詢使用的查詢物件。任一情形下，您都可以識別所要搜尋的影像。任一情形下，您都可以從 DB2 命令行或某程式內提出查詢。

為了回應，Image Extender 會計算查詢的特性值。它會針對目標影像，將該特性值與儲存在 QBIC 型錄中的值作比較。接著，Image Extender 會計算分數，指出每一個目標影像的特性值與來源的相似性。

您可以要求 Image Extender 將特性值最類似來源的影像傳回。您亦可要求 Image Extender 將一或多個影像的分數傳回。

管理 QBIC 型錄

在按內容查詢影像之前，這些影像必須編入 QBIC 型錄中。QBIC 型錄保留有關影像視覺特性的資料。

您可以在想按內容查詢的使用者表格中，為每一個影像直欄建立 QBIC 型錄。使用者表格中每一個影像直欄不能有一個以上的 QBIC 型錄，且多個直欄不能共用同一個 QBIC 型錄。

在建立 QBIC 型錄時，您要識別 Image Extender 儲存資料使用的特性。您也要指出是否要 Image Extender 自動將影像編入型錄中。所謂的自動編目，是指使用者表格中有影像時，Image Extender 會自動在型錄中為影像建立項目。如果影像未被自動編目，則您必須進行手動編目。這表示您要明確地告知 Image Extender 為影像在型錄中建立一些項目。

在建立 QBIC 型錄之後，您可以：

- 開啓型錄來執行一些後續動作
- 將自動編目的設定變更爲手動編目，或將手動編目變更爲自動編目
- 新增特性至型錄中，這可定義 Image Extender 要儲存的特性資料
- 從型錄中移除特性
- 擷取型錄的相關資訊，例如使用者表格名稱及與型錄關聯的直欄名稱，或資料儲存在型錄中的特性
- 以手動方式將影像編入型錄中
- 取消編目影像 (亦即，移除型錄中影像的項目)

- 重新編目影像
- 重新配置型錄 (亦即，在已分割資料庫系統中新增或捨棄節點)
- 關閉型錄
- 刪除型錄

您可以使用 Image Extender 提供的 API 來執行這些作業，包括建立 QBIC 型錄。您亦可使用 db2ext 命令行處理器，執行許多作業。

建立 QBIC 型錄

使用 QbCreateCatalog API 或 CREATE QBIC CATALOG 命令來建立 QBIC 型錄。若要建立型錄，您必須是要進行編目的影像的使用者表格擁有者。此外，您必須具有內含型錄的資料庫之 CREATE TABLE 權限。在直欄中建立影像的 QBIC 型錄之前，Image Extender 必須啟用使用者表格和影像直欄。

在建立 QBIC 型錄時，您要：

- 將含有要編目的影像之直欄和使用者表格命名。
- 指示影像是否自動編目。所謂自動編目，是指影像存入使用者表格之後，Image Extender 將該影像編入型錄。該 Extender 會定期檢查是否有影像在等待編目。您可以設定環境變數 DB2CATALOGDELAY 的值，以秒為單位指定檢查週期。此值可設定的範圍從 1 秒到極大的值。預設值是 60 秒。

手動編目是指您明確要求 Image Extender 將影像編入型錄。(有關如何手動將影像編入型錄的資訊，請參閱第 155 頁的『手動編目影像』。)

啟用使用者表格和直欄： 在直欄中建立影像的 QBIC 型錄之前，Image Extender 必須啟用使用者表格和直欄。(有關對 Image Extender 啟用使用者表格和直欄的資訊，請參閱第 79 頁的第 6 章，『準備 Extender 資料的資料物件』。)

使用 API： 在使用 QbCreateCatalog API 時，您要指定自動編目值來決定自動或手動編目。1 表示自動編目；0 表示手動編目。

例如，使用下列陳述式，可在員工表格的圖形直欄中建立影像的 QBIC 型錄。當影像儲存在員工表格中時，這些影像將會被自動編目：

```
SQLINTEGER autoCatalog=1;                                /* automatic cataloging */

rc=QbCreateCatalog(
    "employee",                                           /* user table */
    "picture",                                           /* image column */
    autoCatalog);                                       /* auto catalog setting */
```

使用命令行： 當發出 CREATE QBIC CATALOG 命令時，指定 ON 可自動編目。指定 OFF 表示手動編目。OFF 是預設值。

例如，使用下列命令，可建立如 API 範例中相同的 QBIC 型錄：

```
CREATE QBIC CATALOG employee picture on
```

備份 QBIC 型錄：Image Extender 會將 QBIC 型錄存入檔案中。您應該定期備份這些檔案，以便在需要時作為回復型錄之用。在 AIX、HP-UX 或 Sun Solaris 伺服器上，這些檔案是位於 `/home/instance_owner/dmb/qbic` 目錄中，其中 *instance_owner* 是案例擁有者的使用者 ID。在 Windows 伺服器中，這些檔案是位於 `\destination\instance\instance_name\qbic` 目錄中，其中 *destination* 是安裝 Image Extender 的目錄，而 *instance_name* 是 Extender 案例的名稱。

開啓 QBIC 型錄

您必須開啓 QBIC 型錄，才能執行變更型錄的後續動作。例如，在新增特性到型錄之前，必須先開啓 QBIC 型錄。

若要開啓 QBIC 型錄，請使用 `QbOpenCatalog` API 呼叫或 `OPEN QBIC CATALOG` 命令。開啓 QBIC 型錄時，您可以：

- 為型錄命名使用者表格和影像直欄。
- 指定要開啓型錄的模式（當您使用 `OPEN QBIC CATALOG` 命令時，它是隱含的）。您可以開啓一個型錄供作業讀取，例如搜尋影像內容。或者，您可以開啓一個型錄供作業更新，例如新增特性。您必須擁有使用者表格的 `SELECT` 權限，才能開啓型錄來讀取作業。您必須擁有使用者表格的 `UPDATE` 權限，才能開啓型錄來更新作業。

若型錄已開啓，則情形如何？如果型錄已開啓，供另一個階段作業更新之用，則您無法再開啓型錄供更新作業使用。當您開啓某 QBIC 型錄時，Image Extender 會關閉現行階段作業中已開啓的任何 QBIC 型錄。

使用 API：當您使用 `QbOpenCatalog` API 時，請明確指定要開啓型錄的模式。請指定：

- API 參數 `qbiRead`，用來開啓讀取作業的型錄。
- API 參數 `qbiUpdate`，用來開啓更新作業的型錄。

`QbiRead` 和 `QbiUpdate` 是定義於 QBIC 併入檔（標頭檔）`dmbqbapi.h` 中的常數。

您也需要指向型錄控點。型錄控點有一種 QBIC 特定資料類型 `QbCatalogHandle`。此資料類型也定義在 `dmbqbapi.h` 中。Image Extender 會傳回型錄控點值作為來自 API 的輸出值。

例如，下列 API 呼叫可開啓 QBIC 型錄，供讀取自該型錄的作業使用：

```

SQLINTEGER mode;
QbCatalogHandle *CatHdl;

mode=qbiRead;                                     /* open catalog for */
                                                  /* read operations */

rc=QbOpenCatalog(
    "employee",                                     /* user table */
    "picture",                                     /* image column */
    mode,                                          /* open catalog mode */
    &CatHdl);                                     /* catalog handle */

```

使用命令行：當您使用 OPEN QBIC CATALOG 命令時，Image Extender 會嘗試開啓更新作業使用的型錄。如果所開啓的型錄目前被另一個階段作業作為更新之用，Image Extender 會開啓讀取作業的型錄。

例如，下列命令可開啓 QBIC 型錄；Image Extender 會嘗試開啓它供更新作業使用：

```
OPEN QBIC CATALOG employee picture
```

完成 QBIC 相關的活動後關閉型錄：開啓 QBIC 型錄時，Image Extender 會配置一些資源給它，如記憶體。完成 QBIC 相關活動後，關閉型錄。這可釋放所配置的資源。

變更自動編目設定

QbSetAutoCatalog API 或 SET QBIC AUTOCATALOG 命令，可用來將自動編目變更為手動編目，或從手動變為自動。在變更型錄設定之前，必須開啓用於更新的 QBIC 型錄。

變更不會追溯既往：當您變更自動編目設定時，僅在變更後才引用到加入使用者表格直欄中的影像。已儲存在使用者表格直欄中的影像不會受到影響。例如，如果您將設定從手動編目變更為自動編目，則只有變更後加入使用者表格直欄中的影像會自動被編目。如果要將已存在於表格直欄中的影像編入型錄，您必須以手動方式編目它們。(有關如何手動將影像編入型錄的資訊，請參閱第 155 頁的『手動編目影像』。)

使用 API：在使用 QbSetAutoCatalog API 時，指定 QBIC 型錄的控點（使用 QbOpenCatalog API 開啓型錄時，將傳回該控點）。並指定自動編目值 1 代表自動編目，或 0 值代表手動編目。

在下列範例中，為 QBIC 型錄指定手動編目，此型錄與員工表格圖形直欄中的影像相關聯。請注意：首先開啓的是更新 QBIC 型錄的作業所使用的型錄。

```

SQLINTEGER mode;
SQLINTEGER autoCatalog=0;                         /* manual cataloging */

```

管理 QBIC 型錄

```
QbCatalogHandle *CatHdl;

mode=qbiUpdate;                                /* open catalog for */
                                                /* update */

/* Open a QBIC catalog */
rc=QbOpenCatalog(
    "employee",                                /* user table */
    "picture",                                /* image column */
    mode,                                     /* open catalog mode */
    &CatHdl);                                /* catalog handle */

/* Change the auto catalog setting */
rc=QbSetAutoCatalog(
    CatHdl,                                    /* catalog handle */
    autoCatalog);                            /* auto catalog flag */
```

使用命令行：在發出 SET QBIC AUTOCATALOG 命令時，您可以指定 ON 來指示自動編目。指定 OFF 表示手動編目。此命令可作用於目前開啓的型錄。

例如，下列命令可爲目前開啓的 QBIC 型錄，將自動編目設成 off：

```
SET QBIC AUTOCATALOG off
```

將特性加入 QBIC 型錄

使用 QbAddFeature API 或 ADD QBIC FEATURE 命令，將特性加入 QBIC 型錄。在將影像編入 QBIC 型錄之前，至少要先將一個特性加入該型錄中。在新增特性之前，必須先開啓用於更新的 QBIC 型錄。

新增特性到型錄時，指定您要新增的特性名稱（這些特性名稱列示於表 11）。

表 11. QBIC 特性名稱

特性名稱	說明
QbColorFeatureClass	平均色
QbColorHistogramFeatureClass	直方圖色
QbDrawFeatureClass	按位顏色
QbTextureFeatureClass	紋理

您可能需要重新編目影像：若新增特性到 QBIC 型錄中，則 Image Extender 並不會自動儲存已編目影像的新特性相關資料，即使自動編目已設爲 on。若要併入有關已編目影像的新特性資料，您需要將影像重新編目（請參閱第 157 頁的『重新編目影像』）。

使用 API：在使用 QbAddFeature API 時，除了特性名稱之外，您還需要指定 QBIC 型錄的控點。請注意特性名稱長度使用的常數 qbiMaxFeatureName 用法。此常數定義在 QBIC 併入（標頭）檔 dmbqbapi.h 中，值是 50。

在下列範例中，可使用 QbAddFeature API 將直方圖色特性加入 QBIC 型錄：

```
char featureName[qbiMaxFeatureName];

QbCatalogHandle CatHdl;

strcpy(featureName,"QbColorHistogramFeatureClass");

rc=QbAddFeature(
    CatHdl,                                /* catalog handle */
    featureName);                          /* feature name */
```

使用命令行：ADD QBIC FEATURE 命令可作用於目前開啓的型錄。在下列範例中，此命令用來將按位顏色特性加入目前開啓的型錄：

```
ADD QBIC FEATURE QbDrawFeatureClass
```

從 QBIC 型錄中移除特性

使用 QbRemoveFeature API 或 REMOVE QBIC FEATURE 命令，從 QBIC 型錄中移除特性。Image Extender 會刪除特性使用的型錄表格。因此，當您在編目影像時，特性的資料不會被儲存。在移除特性之前，必須先開啓用於更新的 QBIC 型錄。

在從型錄移除特性時，指定您要移除的特性名稱。

使用 API：在使用 QbRemoveFeature API 時，除了特性名稱之外，您還需要指定 QBIC 型錄的控點。

在下列範例中，可使用 QbRemoveFeature API，從 QBIC 型錄中將直方圖色特性移除：

```
char featureName[qbiMaxFeatureName];

QbCatalogHandle CatHdl;

strcpy(featureName,"QbColorHistogramFeatureClass");

rc=QbRemoveFeature(
    CatHdl,                                /* catalog handle */
    featureName);                          /* feature name */
```

使用命令行：REMOVE QBIC FEATURE 命令可作用於目前開啓的型錄。在下列範例中，使用此命令從目前開啓的 QBIC 型錄中移除按位顏色特性：

```
REMOVE QBIC FEATURE QbDrawFeatureClass
```

擷取 QBIC 型錄的相關資訊

您可以擷取下列有關 QBIC 型錄的資訊：

- 與型錄相關的影像直欄和使用者表格之名稱。
- 資料存在型錄中的特性數目，以及它們的特性名稱。
- 當影像儲存在使用者表格中時，Image Extender 是否自動編目影像。

您可使用 QbGetCatalogInfo API 來擷取使用者表格和直欄名稱、特性數目、以及自動編目設定。使用 QbListFeatures API 來擷取特性名稱。或使用 GET QBIC CATALOG INFO 命令來擷取所有資訊。

在擷取資訊之前，您必須開啓 QBIC 型錄。

使用 API：當使用 QbGetCatalogInfo API 時，您必須指定 QBIC 型錄的控點。您也需要指示 Image Extender 傳回型錄資訊的結構。型錄資訊結構定義在 QBIC 併入 (標頭) 檔 dmbqapi.h 中，如下所示：

```
typedef struct{
    char      tableName[qbiMaxTableName+1] /* user table */
    char      columnName[qbiMaxColumnName+1] /* image column */
    SQLINTEGER featureCount; /* number of features */
    SQLINTEGER autoCatalog; /* auto catalog flag */
} QbCatalogInfo;
```

發出 QbListFeatures API 呼叫時，您需要配置緩衝區來保留傳回的特性名稱。緩衝區中儲存的特性名稱是以一個空白字元來分隔。您也需要為傳回的特性名稱指定緩衝區大小和型錄控點。若須估計所需的緩衝區大小，可以使用 QbGetCatalogInfo API 傳回的特性計數，乘以最長特性名稱的計數。您可使用常數 qbiMaxFeatureName 作為最長特性名稱大小。

下列範例中的 API 呼叫，擷取有關 QBIC 型錄的資訊。請注意 QbGetCatalogInfo API 如何傳回特性計數，以及 qbiMaxFeature 名稱常數如何被用來計算 QbListFeatures API 的緩衝區大小：

```
long  bufSize;
long  count;
char  *featureNames;

QbCatalogHandle  CatHdl;
QbCatalogInfo    catInfo;

/* Get user table name, image column name, feature count, */
/* and auto catalog setting */

rc=QbGetCatalogInfo(
    CatHdl, /* catalog handle */
    &catInfo); /* catalog info. structure */
```

```

/* List feature names */

bufSize=catInfo.featureCount*qbiMaxFeatureName;
featureNames=malloc(bufSize);

rc=QbListFeatures(
    CatHdl,                /* catalog handle */
    bufSize                /* size of buffer */
    count,                 /* feature count */
    featureNames);         /* buffer for feature names */

```

使用命令行：GET QBIC CATALOG INFO 命令可改變目前開啓的型錄。在下列範例中，此命令用來擷取目前開啓的 QBIC 型錄相關資訊：

```
GET QBIC CATALOG INFO
```

手動編目影像

在建立型錄時，若使用者表格中已有影像，指示是否要 Image Extender 自動編目影像。如果影像未被自動編目，則在影像存入使用者表格後，您必須手動進行編目。您可以手動編目單一影像或整個直欄的影像。

手動編目單一影像

使用 QbCatalogImage API 來手動編目單一影像。您不能經由命令來編目單一影像，因為沒有任何方法可識別命令行中個別的影像。在使用 API 時，指定型錄控點和影像控點 (您可以從使用者表格中擷取影像控點)。手動編目影像之前，您必須開啓 QBIC 型錄。

例如，下列陳述式可從使用者表格中擷取影像，然後編目影像：

```

/* Retrieve the image handle */

EXEC SQL BEGIN DECLARE SECTION;
char Img_hdl[251];
EXEC SQL END DECLARE SECTION;

QbCatalogHandle CatHdl;

EXEC SQL SELECT PICTURE INTO :Img_hdl
    FROM EMPLOYEE
    WHERE NAME='Anita Jones';

/* Catalog the image*/

rc=QbCatalogImage(
    CatHdl,                /* catalog handle */
    Img_hdl);              /* image handle */

```

手動編目影像直欄

使用 QbCatalogColumn API 或 CATALOG QBIC COLUMN 命令，手動編目影像直欄。Image Extender 只會對在上一次直欄編目後，新近在直欄中插入、更新或刪除的影像進行編目。Image Extender 對型錄中所有特性的影像進行編目。手動編目影像直欄之前，您必須開啓用於更新的 QBIC 型錄。

使用 API：當您使用 QbCatalogColumn API 時，指定型錄控點。Image Extender 使用的是與特定型錄有關的使用者表格直欄中的影像。

例如，下列 API 呼叫可在與特定型錄有關的使用者表格直欄中，對尚未編目的影像進行編目。這些影像已對型錄中所有特性編目：

```
QbCatalogHandle CatHdl;  
  
rc=QbCatalogColumn(  
    CatHdl);                                /* catalog handle */
```

使用命令行：使用 CATALOG QBIC COLUMN 命令來手動編目影像直欄。您亦可使用此命令來重新編目影像（請參閱第 157 頁的『重新編目影像』）。指定參數 FOR 和 NEW。（FOR 和 NEW 是預設參數。）

在下列範例中，此命令可在與目前開啓的型錄有關之表格直欄中，用來對尚未編目的影像進行編目。這些影像已對型錄中所有特性編目：

```
CATALOG QBIC COLUMN FOR NEW
```

取消編目影像

取消編目影像意指從 QBIC 型錄中移除影像的項目。使用 QbUncatalogImage API 來取消影像的編目。您不能經由命令來取消影像編目，因為沒有任何方法可識別命令行中個別の影像。在使用 API 時，指定型錄控點和影像控點（您可以從使用者表格中擷取影像控點）。在取消影像編目之前，您必須開啓用於更新的 QBIC 型錄。

例如，下列陳述式可從使用者表格中擷取影像控點，然後取消影像編目：

```
/* Retrieve the image handle */  
  
EXEC SQL BEGIN DECLARE SECTION;  
char Img_hdl[251];  
EXEC SQL END DECLARE SECTION;  
  
QbCatalogHandle CatHdl;  
  
EXEC SQL SELECT PICTURE INTO :Img_hdl  
    FROM EMPLOYEE  
    WHERE NAME='Anita Jones';  
  
/* Uncatalog the image */
```



```
rc=QbUncatalogImage(
    CatHdl,                      /* catalog handle */
    Img_hdl);                   /* image handle */
```

重新編目影像

當您編目影像時，Image Extender 會分析 QBIC 型錄所識別的影像特性，並將那些特性值存入型錄中。當您新增特性到 QBIC 型錄時，Image Extender 不會自動分析已編目影像的新特性。若要將新特性的值新增到型錄，您需要重新編目所有影像。

使用 QbReCatalogColumn API 或 CATALOG QBIC COLUMN 命令，來重新編目 QBIC 型錄中的影像。Image Extender 移除目前在型錄中的所有特性資料。然後分析影像的所有特性，包括新的特性，最後將影像編目。在重新編目影像之前，您必須開啓 QBIC 型錄。

使用 API：當您使用 QbReCatalogColumn API 時，指定型錄控點。

在下列範例中，已重新分析 QBIC 型錄中的影像：

```
QbCatalogHandle CatHdl;

rc=QbReCatalogColumn(
    CatHdl);                      /* catalog handle */
```

使用命令行：使用 CATALOG QBIC COLUMN 命令來重新編目影像。此命令可作用於目前開啓的型錄。您亦可使用此命令，手動將影像編目（請參閱第 155 頁的『手動編目影像』）。

當您發出此命令時，請指定參數 FOR 和 ALL。這樣會告知 Image Extender 您要重新編目所有影像。

在下列範例中，重新分析目前開啓的 QBIC 型錄中已編目的影像：

```
CATALOG QBIC COLUMN FOR ALL
```

重新配置 QBIC 型錄 (僅適用於 EEE)

使用 DMBRedistribute API 或 REDISTRIBUTE NODEGROUP 命令，當加入節點到節點群組或從節點群組移除節點時，重新配置 QBIC 特性資料。此命令會將 QBIC 特性資料存放在與相關使用者資料的同一個節點。

若重新分配程序傳回錯誤，則您可以根據命令回應的指示，決定是否使用 CONTINUE 參數來重新執行命令。此選項指示系統從停止處繼續執行，而非從頭開始執行。在執行 DB2 的 REDISTRIBUTE 命令之後，在第一次執行 REDISTRIBUTE NODEGROUP 命令時，您不應該使用 CONTINUE 參數。

管理 QBIC 型錄

若要維護資料完整性，請一次重新配置一個節點群組。請先等待一個節點群組完成重新分配後，再開始另一個節點群組的分配。

使用 API：下列範例顯示如何在 groupone 節點群組中重新分配 QBIC 特性資料：

```
#include <dmbdst.h>

rc = DMBRedistribute(groupone,"continue");
```

使用命令行：下列範例顯示如何使用 REDISTRIBUTE NODEGROUP 命令，利用 CONTINUE 參數來重新配置 my_nodegroup 節點的資料：

```
redistribute nodegroup my_nodegroup continue
```

關閉 QBIC 型錄

使用 QbCloseCatalog API 或 CLOSE QBIC CATALOG 命令，關閉 QBIC 型錄。關閉此型錄前必須先開啓它。

使用 API：當您發出 QbCloseCatalog API 呼叫時，指定型錄控點。例如：

```
QbCatalogHandle CatHdl;

rc=QbCloseCatalog(
    CatHdl);                                /* catalog handle */
```

使用命令行：CLOSE QBIC CATALOG 命令可作用於目前開啓的型錄。在下列範例中，此命令用來關閉目前開啓的 QBIC 型錄：

```
CLOSE QBIC CATALOG
```

刪除 QBIC 型錄

刪除 QBIC 型錄即會刪除型錄表格中所有特性資料。因此，相關的影像無法再按內容查詢。若要刪除 QBIC 型錄，您必須具備與型錄相關的表格之 ALTER 或 CONTROL 權限。刪除型錄之前必須先開啓型錄。

使用 QbDeleteCatalog API 或 DELETE QBIC CATALOG 命令，刪除 QBIC 型錄。當您刪除 QBIC 型錄時，請命名與型錄相關的使用者表格和直欄。

使用 API：在下列範例中，使用 QbDeleteCatalog API 來刪除 QBIC 型錄：

```
rc=QbDeleteCatalog(
    "employee",                                /* user table */
    "picture");                                /* image column */
```

使用命令行：DELETE QBIC CATALOG 命令可作用於目前開啓的型錄。在下列範例中，此命令用來刪除目前開啓的 QBIC 型錄：

```
DELETE QBIC CATALOG employee picture
```

QBIC 型錄範例程式

第 160 頁的圖 28 顯示以 C 語言撰寫的部份程式碼，此程式碼可用來建立 QBIC 型錄。此程式也會將影像直欄編入 QBIC 型錄中。您可以在 SAMPLES 子目錄中的 QBCATDMO.C 檔找到完整程式。在執行完整程式之前，您必須先執行 ENABLE 和 POPULATE 範例程式（這兩個範例程式也位於 SAMPLES 子目錄中）。有關範例程式的詳細資訊，請參閱第 549 頁的附錄 B，『範例程式和媒體檔案』。

請注意程式中下列各點：

- 1** 併入 dmbqbapi 標頭檔。
- 2** 連接到資料庫。
- 3** 建立型錄。此型錄在自動編目關閉時建立。
- 4** 開啓用於更新的型錄。
- 5** 將平均色特性加入型錄。
- 6** 將影像直欄編入型錄。
- 7** 關閉型錄。

管理 QBIC 型錄

```
#include <sql.h>
#include <sqlcli.h>
#include <sqlcli1.h>
#include <dmbqbqpi.h> 1
#include <stdio.h>

/*****
/* Define the function prototypes                                     */
*****/

void printError(SQLHSTMT hstmt);
void createCatalog();
void openCatalog();
void closeCatalog();
void addFeature();
void catalogImageColumn();

QbCatalogHandle cHdl = 0;

static SQLHENV    henv;
static SQLHDBC    hdbc;
static SQLHSTMT   hstmt;
static SQLRETURN  rc;
char tableName[] = "sobay_catalog";
char columnName[] = "covers";

SQLCHAR uid[18+1];
SQLCHAR pwd[30+1];
SQLCHAR dbnName[SQL_MAX_DSN_LENGTH+1];

void main ()
{
/*---- prompt for database name, userid, and password ----*/
    printf("Enter database name:\n");
    gets((char *) dbName);
    printf("Enter userid:\n");
    gets((char *) pwd);
/* set up the SQL CLI environment */
    SQLAllocEnv(&henv);
    SQLAllocConnect(henv, &hdbc);
    rc = SQLConnect(hdbc, dbName, SQL_NTS, uid, SQL_NTS, pwd, SQL_NTS); 2
    if (rc != SQL_SUCCESS)
    {
        printError(SQL_NULL_HSTMT);
        exit(1);
    }
}
```

圖 28. QBIC 型錄範例程式 (1/4)

```

    createCatalog();
    openCatalog();
    addFeature();
    getCatalogInfo();
    listFeatures();
    catalogImageColumn();
    closeCatalog();

    SQLDisconnect(hdbc);
    SQLFreeConnect(hdbc);
    SQLFreeEnv(henv);
}
/*****
void createCatalog()
{
    SQLINTEGER autoCatalog = 0;
    SQLINTEGER retLen;
    SQLINTEGER errCode = 0;
    char errMsg[500];

    QbCreateCatalog( 3
        (char *) tableName,
        (char *) columnName,
        autoCatalog,
        0
    );

    DBiGetError(&errCode, errMsg);
    if(errCode) printf("Error code is %d Error Message %s", errCode, errMsg);
}
/*****
void openCatalog()
{
    SQLINTEGER errCode = 0;
    char errMsg[500];
    SQLINTEGER mode = qbiUpdate;

    QbOpenCatalog( 4
        (char *) tableName,
        (char *) columnName,
        mode,
        &cHdl
    );

    DBiGetError(&errCode, errMsg);
    if(errCode) printf("Error code is %d Error Message %s", errCode, errMsg);
}

```

圖 28. QBIC 型錄範例程式 (2/4)

管理 QBIC 型錄

```

/*****
void addFeature()
{
    SQLINTEGER errCode=0;
    char errMsg[512];
    if(cHdl) /* if we have an open catalog, else do nothing */
    {
        char featureName[512] = "QbColorFeatureClass"; 5
        QbaddFeature(
            cHdl,
            featureName
        );

        DBiGetError(&errCode, errMsg);
        if(errCode) printf("Error code is %d Error Message %s", errCode, errMsg);
    }
    else
    {
        exit(1);
    }
}
*****/
void catalogImageColumn()
{
    SQLINTEGER errCode = 0;
    char errMsg[512];

    if(cHdl) /* if we have an open catalog, else do nothing */
    {
        SQLRETURN rc;
        QbCatalogColumn( 6
            cHdl,
        );

        DBiGetError(&errCode, errMsg);
        if(errCode) printf("Error code is %d Error Message %s", errCode, errMsg);
    }
    else
    {
        exit(1);
    }
}

```

圖 28. QBIC 型錄範例程式 (3/4)

```

/*****
void closeCatalog()
{
    if(cHdl) /* if we have an open catalog, else do nothing */
    {
        QbCloseCatalog( 7
                        cHdl,
                        );
    }
}
*****/

```

圖 28. QBIC 型錄範例程式 (4/4)

建置查詢

在按內容查詢影像時，您要識別查詢的輸入檔及編目影像的目標設定。查詢的輸入檔指定查詢所用的特性名稱、特性值及特性比重 (亦即每個特性的強調點)。

您有兩個方法提供此輸入檔：

- 在查詢中指定查詢字串。查詢字串是一種字串，可用來指定查詢的特性、特性值以及特性比重。
- 建立查詢物件並在您的查詢中參照它。查詢物件可指定特性和特性比重。它也會識別每一個特性的資料來源。資料來源提供每一個特性的值。

指定查詢字串

您可使用查詢字串來識別查詢的特性、特性值和特性比重。**查詢字串**是一種字串，其格式為 *feature_name value*，其中 *feature_name* 指 QBIC 特性名稱，*value* 是與特性相關的值。

您可以在一個查詢中指定多個特性。然後，為每一個特性指定特性名稱值配對，如『特性值』中所述。每一個配對是以子句 AND 隔開。當您在一個查詢中指定多個特性時，您亦可指定比重給一或多個特性，如第 165 頁的『特性比重』中所述。接著查詢字串的格式為 *feature_name value weight*，其中 *weight* 是指定給特性的比重。

Image Extender 提供共用一個查詢字串的 API (QbQueryStringSearch) 和兩個 UDF (QbScoreFromStr 和 QbScoreTBFromStr)。當您發出查詢時，請使用適當的 API 或 UDF，並指定查詢字串作為輸入參數。(詳細資訊，請參閱第 173 頁的『按影像內容發出查詢』。)

特性值

在查詢字串中，為查詢中每一個特性指定一個特性值。

當傳遞查詢到 DB2 命令內時，您必須遵守某些檔案命名慣例，查詢才能正常運作。包含空格或直角括弧 (>) 的檔名必須以雙引號含括；其它檔名則可選擇性地以雙引號含括。如果檔名以引號括住，每一個引號前都必須有一個跳出字元 (\)。如果查詢未傳遞到 DB2 命令內，則不需要以引號括住跳出字元。

在下列範例中，查詢字串會透過 DB2 命令傳送：

```
db2 "select image_id from table
(mmdbsys.QbScoreTBFromStr
('texture file=<server,patterns/ptrn07.gif>',
'fabric',
'swatch_img',
10))
as T1"
```

表 12 列出每一個特性可指定的值。下面指示的每一個特性名稱是可替代的簡短名稱。

表 12. 特性值可在查詢字串中指定

特性名稱	值
averageColor、average 或 QbColorFeatureClass	<p>顏色=<Rvalue、Gvalue、Bvalue></p> <p>顏色值範圍為 0 到 255 的整數，可用來識別影像的紅色值 (Rvalue)、綠色值 (Gvalue) 和藍色值 (Bvalue)。</p> <p>檔案=<file_location, filename></p> <p>對伺服器檔案，file_location 是 server。filename 是以檔案所在的系統，其適合的格式來指定之完整檔案路徑或相對檔名。DB2 Extender 將使用環境變數解析相對檔名 (請參閱第 543 頁的『如何使用環境變數來分辨檔名』)。</p>
histogram、histogramcolor 或 QbColorHistogramFeatureClass	<p>直方圖=<(hist_value、Rvalue、Gvalue、Bvalue)>，...</p> <p>每一個直方圖色值都指定在一個子句中，此子句用來識別直方圖 (hist_value) 中顏色的百分比 (1 到 100)，以及顏色的紅色值 (Rvalue)、綠色值 (Gvalue) 和藍色值 (Bvalue)。</p> <p>檔案=<file_location, filename></p> <p>對伺服器檔案，file_location 是 server。filename 是以檔案所在的系統，其適合的格式來指定之完整檔案路徑或相對檔名。DB2 Extender 將使用環境變數解析相對檔名。</p>

表 12. 特性值可在查詢字串中指定 (繼續)

特性名稱	值
draw、positional 或 QbDrawFeatureClass	檔案=<file_location, filename> handle=<image_handle> 對伺服器檔案，file_location 是 server。filename 是以檔案所在的系統，其適合的格式來指定之完整檔案路徑或相對檔名。DB2 Extender 將使用環境變數解析相對檔名。
texture 或 QbTextureFeatureClass	檔案=<file_location, filename> handle=<image_handle> 對伺服器檔案，file_location 是 server。filename 是以檔案所在的系統，其適合的格式來指定之完整檔案路徑或相對檔名。DB2 Extender 將使用環境變數解析相對檔名。

特性比重

如果在查詢字串中指定多個特性，您也可以為一或多個特性指定比重。特性的比重指出 Image Extender 在計算相似性點數及傳回影像內容的查詢結果時，對特性的強調。您為特性指定的比重越高，則在查詢中對該特性的強調就越大。比重指大於 0.0 的實際數字，例如 2.5 或 10.0。如果您沒有在查詢字串中指定比重，則 Image Extender 將使用特性的預設比重。如果該特性是查詢字串中唯一指定的特性，則不需要指定比重。(在查詢中，該特性將一直保有完整的比重。)

特性的比重與查詢中指定的其它特性有關。例如，假定您指定平均色和紋理特性，並對平均色指定比重值 2.0。它告知 Image Extender 給予平均色值相較於紋理值兩倍的強調。

範例

下列查詢字串指定紅色的平均色：

averageColor color=<255, 0, 0>

下列查詢字串指定紅色 10%、綠色 50% 及藍色 40% 所構成的直方圖：

histogram histogram=<(10, 255, 0, 0), (50, 0, 255, 0),
(40, 0, 0, 255)>

下列查詢字串指定平均色值和紋理值。紋理值由伺服器檔案中的影像提供。紋理比平均色加倍重要：

averageColor color=<30, 200, 25> and
texture file=<server, "\patterns\pattern7.gif"> weight=2.0

使用查詢物件

查詢物件可用來識別您查詢的特性、特性值及特性比重。建立查詢物件並將特性加入其中。然後指定每一個特性的資料來源。資料來源提供特性的值。例如，資料來源可以是檔案中的影像。如果平均色是相關特性，則影像的平均色與查詢物件有關。如果將多個特性加入查詢物件，則您可以指定比重給一或多個特性。

Image Extender 提供三個 API (QbQuerySearch、QbQueryStringSearch 及 QbQueryNameSearch) 以及兩個 UDF (QbScoreFromName 和 QbScoreTBFromName)，皆共用一個查詢物件。當發出某個查詢時，請使用適當的 API 或 UDF，並指定查詢物件作為輸入參數。(詳細資訊，請參閱第 173 頁的『按影像內容發出查詢』。)

建立查詢物件

使用 QbQueryCreate API 來建立查詢物件。在回應時，Image Extender 會傳回查詢物件的控點。控點具有 QBIC 特定資料類型 QbQueryHandle，其定義在 QBIC 併入 (標頭) 檔 dmbqbapi.h 中。

在使用 API 時，您需要指出查詢物件控點。您也需要在查詢物件上執行其它作業 (如新增特性) 的 API 中指定控點。

例如，下列 API 呼叫可建立查詢物件：

```
QbQueryHandle qHandle;  
  
rc=QbQueryCreate(  
    &qHandle);                               /* query object handle */
```

將特性加入查詢物件

藉由將特性加入查詢物件，可識別您要 Image Extender 查詢的影像特性。

使用 QbQueryAddFeature API 來將特性加入查詢物件。在使用 API 時，請指定查詢物件控點。您也要將特性命名。您只能在 API 中指定一個特性。您必須對每一個要加入查詢物件的特性發出個別的 API 呼叫。

在下列範例中，QbQueryAddFeature API 用來加入平均色特性到查詢物件：

```
char featureName[qbiMaxFeatureName];  
QbQueryHandle qHandle;  
  
rc=QbQueryAddFeature(  
    qHandle,                                   /* query object handle */  
    "QbColorFeatureClass");                  /* feature name */
```

指定查詢物件中特性的資料來源

使用 `QbQuerySetFeatureData` API 來指定查詢物件中特性的資料來源。資料來源檔可以是：

- 使用者表格直欄中已編目或取消編目的影像
- 從屬站工作站上的影像
- 從屬站工作站上緩衝區中的影像

此外，您可以明確地指定平均色或直方圖色特性的資料。例如，您可以指定平均色的紅色、綠色及藍色值。

當使用 API 時：

- 請指定查詢物件控點。
- 將特性命名。
- 指出 `QbImageSource` 結構 (詳細資訊，請參閱第 167 頁)。

使用資料來源結構： 使用各種結構來提供查詢物件的資料來源資訊。這些結構如下：

- `QbImageSource`
- `QbColor`
- `QbHistogramColor`

QbImageSource： `QbImageSource` 結構可識別查詢物件中特性的來源類型。此結構定義在 QBIC 併入 (標頭) 檔 `dmbqbapi.h` 中，如下所示：

```
typedef struct{
    SQLINTEGER    type;
    union {
        char          imageHandle[MMDB_BASE_HANDLE_LEN+1];
        QbImageFile   clientFile;
        QbImageBuffer  buffer;
        QbSampleSource reserved;
        QbColor        averageColor;
        QbHistogramColor histogramColor[qbiHistogramCount];
    };
} QbImageSource;
```

`QbImageSource` 結構中的類型欄位指示來源類型。您可以設定欄位中的值，如下所示：

值	意義
<code>qbiSource_ImageHandle</code>	來源位於使用者表格直欄中
<code>qbiSource_ClientFile</code>	來源位於從屬站工作站檔案中
<code>qbiSource_Buffer</code>	來源位於從屬站工作站緩衝區中

值	意義
qbiSource_ServerFile	來源位於伺服器檔案
qbiSource_AverageColor	來源為平均色規格
qbiSource_HistogramColor	來源為直方圖色規格

這些設定僅對適當特性有效。例如，qbiSource_AverageColor 僅對平均色特性有效。

若類型欄位設為 qbiSource_ServerFile，對伺服器上檔案的名稱及類型，請使用 clientFile。

根據來源類型，Image Extender 也會檢查您指定的其它資訊。它會顯示於表 13。

表 13. Image Extender 在 QbImageSource 中檢查什麼

來源	Image Extender 檢查什麼	在何處指定
使用者表格	影像控點	QbImageSource 的影像控點欄位
檔案	檔名 檔案格式	QbImageSource 的 clientFile 欄位
緩衝區	檔名	QbImageBuffer (關於使用此結構的詳細資訊，請參閱第 168 頁)
平均色規格	紅色、綠色和藍色值	QbColor 結構 (有關使用此結構的細節，請參閱第 168 頁)
直方圖色規格	顏色值和百分比	QbHistogramColor 結構 (有關使用此結構的細節，請參閱第 169 頁)

QbImageBuffer：當資料來源位於緩衝區中時，使用 QbImageBuffer 結構來指定影像格式、長度和內容。此結構定義在 QBIC 併入 (標頭) 檔 dmbqbapi.h 中，如下所示：

```
typedef struct{
    char          format[qbiImageFormatLength+1];
    SQLINTEGER    length;
    char*         image;
} QbImageBuffer;
```

QbColor：當資料來源是平均色規格時，可使用 QbColor 結構來指定平均色的紅色、綠色及藍色值。此結構定義在 QBIC 併入 (標頭) 檔 dmbqbapi.h 中，如下所示：

```
typedef struct{
    SQLUSMALLINT  red;          /*0 off - 65535 (fully on) */
    SQLUSMALLINT  green;        /*0 off - 65535 (fully on) */
    SQLUSMALLINT  blue;         /*0 off - 65535 (fully on) */
} QbColor;
```

設定 `QbColor` 中的值，指示平均值計算中要分解為因素的紅色、綠色及藍色圖點數量。這些值的範圍從 0 到 65535。0 值表示忽略項目。

QbHistogramColor：使用 `QbHistogramColor` 結構，指定直方圖色規格的每一個顏色元件。直方圖陣列的完整規格包含在 `QbHistogramColor` 結構的陣列中。每一個結構含有一個顏色值和一個百分比。此顏色值由紅色、綠色和藍色圖點值構成。此百分比指定目標影像中所需顏色所佔的百分比。

此結構定義在 `QBIC` 併入 (標頭) 檔 `dmbqbapi.h` 中，如下所示：

```
typedef struct{
    QbColor      color;
    SQLUSMALLINT percentage; /*0 - 100 */
} QbHistogramColor;
```

設定 `QbColor` 中的值，指示顏色的紅色、綠色和藍色圖點數量。這些值的範圍從 0 到 65535。設定百分比，以指示目標影像中所需的指定顏色所佔的百分比。此值範圍從 1 到 100。直方圖色中顏色元件的百分比總和必須不大於 100。

範例： 下列範例中的 API 指定查詢物件中直方圖色特性的資料來源。資料來源檔是從屬站工作站中的檔案。

```
char          featureName[qbiMaxFeatureName];
QbQueryHandle qHandle;
QbImageSource imgSource;

imgSource.type=qbiSource_ClientFile;
strcpy(imgSource.clientFile.fileName, "/tmp/image.gif");
strcpy(imgSource.clientFile.format, "GIF");

rc=QbQuerySetFeatureData(
    qHandle,                                     /* query object handle */
    "QbColorHistogramFeatureClass",             /* feature name */
    &imgSource);                                /* feature data source */
```

在下列範例中，資料來源是紅色的平均色規格：

```
char          featureName[qbiMaxFeatureName];
QbColor       avgColor;
QbImageSource imgSource;

imgSource.type=qbSource_AverageColor;
avgColor.red=255;
avgColor.green=0;
avgColor.blue=0;
strcpy(featureName, "QbColorFeatureClass");

rc=QbQuerySetFeatureData(
    qHandle,                                     /* query object handle */
    featureName,                                /* feature name */
    &imgSource);                                /* feature data source */
```

設定查詢物件中特性的比重

如果您已將一個以上特性加入查詢物件，可以指定查詢中所提供一個以上特性的比重。使用 `QbQuerySetFeatureWeight` API 來指定特性的比重。特性的比重指出 `Image Extender` 在計算相似性點數及傳回影像內容的查詢結果時，對特性的強調。對特性指定了越高的比重，表示越強調查詢物件中的該特性。

雖然，每次您發出 `QbQuerySetFeatureWeight` API 時，只能對一個特性指定比重，但您可以對查詢物件中一或多個特性指定比重。如果您沒有指定比重給查詢物件中的特性，則 `Image Extender` 將使用此特性的預設比重。如果特性是查詢物件中唯一的特性，則不需要指定比重給該特性。(在查詢物件中，此特性將一直保有完整的比重。)

當使用 API 時：

- 指定查詢物件控點。
- 指定特性名稱。
- 指出特性比重。您可以將比重設成大於 0 的實際數字，例如，2.5 或 10.0。指定的值越高，表示越強調該特性。此設定可變更先前為查詢物件中特性所設定的任何比重。

在下列範例中，一個查詢物件含有平均色特性及至少一個其它特性。您可使用 `QbQuerySetFeatureWeight` API 來指定查詢物件中平均色特性的比重：

```
char          featureName[qbiMaxFeatureName];
double        weight;
QbQueryObjectHandle qoHandle;

strcpy(featureName,"QbColorFeatureClass");
weight=5.00;

rc=QbQuerySetFeatureWeight(
    qoHandle,                      /* query object handle */
    featureName,                  /* feature name */
    &weight);                     /* feature weight */
```

儲存並重覆使用查詢字串

除非您儲存查詢物件，否則它們只是暫時性狀態。這些查詢物件只存在於單一資料庫連接期間。您可以儲存來自查詢的查詢字串，再次用於程式，或即使現行資料庫連接捨棄後的程式呼叫中使用。

`Image Extender` 提供 `QbQueryGetString` API，可傳回來自查詢物件的查詢字串。然後，該查詢字串可以輸入 `QbQueryStringSearch` API 或 `QbScoreFromStr`，及按影像內容的其它字串 `QbScoreTBFromStr` UDF。(請參閱第 173 頁的『按影像內容發出查詢』)。

當您使用下列命令來建置字串時，可建置查詢字串：

- QbQueryCreate
- QbQueryAddFeature
- QbQuerySetFeatureData
- QbQuerySetFeatureWeight
- QbQueryRemoveFeature

建立查詢後，您可以呼叫 `QbQueryGetString` 來取得字串。在該程式內呼叫中可以使用此查詢字串，或將它儲存至其他檔案，以使用於應用程式後續的呼叫及其它資料庫連接。使用 `QbQueryGetString` 所傳回的查詢字串後，您必須明確地釋放空間。

在下列範例中，`QbQueryGetString` 用來擷取查詢物件的查詢字串：

```
SQLRETURN rc;
char* qryString;
QbQueryHandle qHandle;

.....          /* 這裡建立並使用查詢 */

rc = QbQueryGetString(qHandle, &qryString);
if ( rc == 0 ) {
    ...          /* 這裡查詢字串作為輸入 */
    free((void *)qryString);
    qryString=(char *)0;
}
```

限制：當您使用從屬站檔案指定資料來源給某特性時，查詢字串不會反映特性資料。

擷取查詢物件的相關資訊

您可以判斷已加入查詢物件的特性 (若有的話)。您亦可判斷特性的現行比重。

使用此 API

`QbQueryGetFeatureCount`
`QbQueryListFeatures`

擷取

查詢物件中的特性數目
 查詢物件中的特性名稱

當您發出 `QbQueryGetFeatureCount` API 時，請指定查詢物件控點。您也需要指向計數器。`Image Extender` 會傳回計數器中的特性計數。

在下列範例中，`QbQueryGetFeatureCount` API 用來判斷查詢物件中特性數目：

```
SQLINTEGER    count;
QbQueryHandle qHandle;
```

建置查詢

```
rc=QbQueryGetFeatureCount(  
    qHandle,                                /* query object handle */  
    &count);                                /* feature count */
```

當您發出 `QbQueryListFeatures` API 呼叫時，您必須配置緩衝區來保留傳回的特性名稱。您也需要為傳回的特性名稱指定型錄控點和緩衝區大小。

在下列範例中，`QbQueryListFeatures` API 用來擷取查詢物件中每一個特性的名稱：

```
SQLINTEGER    retCount,bufSize;  
char*         featureName;  
QbQueryHandle qHandle;  
  
bufSize=qbiMaxFeatureName;  
featureName=(char*)malloc(bufSize);  
  
rc=QbQueryListFeatures(  
    qHandle,                                /* query object handle */  
    bufSize                                /* size of buffer */  
    &retCount,                              /* feature count */  
    featureName);                          /* buffer for feature names */
```

移除查詢物件中特性

使用 `QbQueryRemoveFeature` API，移除查詢物件中的特性。當您使用 API 時，請指定查詢物件控點及為特性命名。

在下列範例中，`QbQueryRemoveFeature` API 用來移除查詢物件中直方圖色特性：

```
char          featureName[qbiMaxFeatureName];  
QbQueryHandle qHandle;  
  
strcpy(featureName,"QbColorHistogramFeatureClass");  
  
rc=QbQueryRemoveFeature(  
    qHandle,                                /* query object handle */  
    featureName);                          /* feature name */
```

刪除查詢物件

使用 `QbQueryDelete` API 刪除未命名的查詢物件。Image Extender 會從目前連接的資料庫中刪除查詢。

當您使用 `QbQueryDelete` API 時，請指定查詢物件控點。

在下列範例中，`QbQueryDelete` API 用來刪除查詢物件：

```
QbQueryHandle qHandle;  
  
rc=QbQueryDelete(  
    qHandle);                                /* query object handle */
```

如果您使用指名的查詢，請使用 `QbQueryNameDelete` API 來刪除查詢物件。

按影像內容發出查詢

在編目影像之後，您可以按內容查詢一或多個影像。當您按內容查詢影像時，您要識別查詢的輸入檔，以及編目影像的目標設定。您可以在查詢字串中指定輸入檔 (請參閱第 163 頁的『指定查詢字串』)，或在查詢物件中指定輸入檔 (請參閱第 166 頁的『使用查詢物件』)。

如果您使用的是查詢字串，您可以從 DB2 命令行或程式內提出查詢。如果您使用的是查詢物件，您可以透過參照其控點從程式內提出查詢。

Image Extender 會比較查詢中指定的特性值與目標影像作的特性值一比較，並計算每一個影像的分數。此分數指示目標影像的特性值與查詢中指定的特性值之相似性。

您可以擷取特性值與查詢最相似的那些影像。您亦可查詢單一的編目影像並取得它的分數，或取得表格直欄中所有編目影像的分數。

查詢影像

Image Extender 提供三個 API 來查詢表格直欄中已編目的影像。這些 API 僅在需要查詢字串或將查詢物件當作輸入時不同：

API	輸入
QbQueryStringSearch	查詢字串
QbQuerySearch	查詢物件控點
QbQueryNameSearch	查詢物件名稱

在這三個 API 中，您也要：

- 對含有所要搜尋影像的直欄和表格命名。您必須將影像編入 QBIC 型錄中。
- 指定所要傳回的最大結果數目。
- 指示指定了查詢範圍的結構。將指標設成 0、NULL 值或空白字串。這會指定搜尋表格直欄中所有已編目的影像。
- 指定常數 qbiArray 來指示陣列中儲存的結果。此 qbiArray 常數定義在 QBIC 併入 (標頭) 檔 dmbqbapi.h 中。

您也要指示輸出結構陣列包含搜尋結果。因此，Image Extender 會將特性值與查詢特性值最相似的目標影像之控點傳回這些結構中。也會傳回每一個影像的分數，其指示影像的特性值與查詢的相似性。此結構定義在 QBIC 併入 (標頭) 檔 dmbqbapi.h 中，如下所示：

發出 QBIC 查詢

```
typedef struct{
    char        imageHandle[MMDB_BASE_HANDLE_LEN+1];
    SQLDOUBLE   SCORE
} QbResult;
```

您必須配置足夠保留所指定最大結果數目的陣列，並在 API 中指示陣列。您也要指示計數器；Image Extender 將計數器值設成傳回的結果數。

在下列範例中，QbQueryStringSearch API 用來按內容查詢表格直欄中已編目的影像。請注意：查詢的指標是設成零值。

```
QbResult      returns[MaxQueryReturns];
SQLINTEGER    maxResults=qbiMaxQueryReturns;
SQLINTEGER    count;
QbQueryHandle qHandle;
QbResult      results[qbiMaxQueryReturns];

rc=QbQueryStringSearch(
    "QbColorFeatureClass color=<255, 0, 0>" /*query string */
    "employee",                               /* user table */
    "picture",                               /* image column */
    maxResults,                               /* maximum number of results */
    0,                                       /* query scope pointer */
    qbiArray,                                /* store results in an array */
    &count,                                  /* count of returned images */
    results);                                /* array of returned results */
```

此處為使用 QbQuerySearch API 的要求。請注意：查詢物件控點被指定為輸入。

```
QbResult      returns[MaxQueryReturns];
SQLINTEGER    maxResults=qbiMaxQueryReturns;
SQLINTEGER    count;
QbQueryHandle qHandle;
QbResult      results[qbiMaxQueryReturns];

rc=QbQuerySearch(
    qHandle,                                /* query object handle */
    "employee",                             /* user table */
    "picture",                             /* image column */
    maxResults,                             /* maximum number of results */
    0,                                     /* query scope pointer */
    qbiArray,                              /* store results in an array */
    &count,                               /* count of returned images */
    results);                              /* array of returned results */
```

擷取影像分數

Image Extender 提供四個可用於 SQL 陳述式的 UDF，來擷取表格直欄中已編目影像的分數。score 指範圍從 0.0 到無限大的加倍精準度浮點值。score 越低表示與符合查詢中所指定特性值的影像特性值越接近。score 值 0.0 表示影像完全相符。

UDF 如下：

- QbScorefromStr
- QbScoreTBfromStr
- QbScoreFromName
- QbScoreTBFromName

建議： 使用 `QbScoreFromStr` UDF，取得單一已編目影像的分數。使用 `QbScoreTBfromStr` UDF，以取得表格直欄中多個已編目影像的分數。

擷取單一影像的分數

使用 `QbScoreFromStr` UDF，取得表格直欄中單一已編目影像的分數。將作為輸入的查詢字串指定給 `QbScoreFromStr` UDF。如果您使用 `QbScoreFromName` UDF，可以指定作為輸入的查詢物件名稱給 `QbScoreFromName` UDF。使用其中一個 UDF，您也可以指定含有目標影像的表格直欄名稱。

在下列查詢中，可使用 `QbScoreFromStr` UDF，尋找表格直欄中平均色分數最接近紅色的已編目影像。

```
SELECT name, description
decimal (QbScoreFromStr(swatch_img,
                        'QbColorFeatureClass color=<255, 0, 0>'), /* query string *
                        10, 5) AS score
FROM fabric /* table column */
ORDER BY score
```

擷取多個影像的分數

使用 `QbScoreTBfromStr` UDF，以取得表格直欄中多個已編目影像的分數。如果您有指名的查詢，可以使用 `QbScoreTBfromName` UDF。這兩個 UDF 會傳回影像控點和分數的兩個直欄表格；而表格中橫列是以分數的遞增次序排列。結果表格中的控點直欄名稱是 `IMAGE_ID`，分數直欄名稱是 `SCORE`。

將作為輸入的查詢字串指定給 `QbScoreTBfromStr` UDF。指定作為輸入的查詢物件名稱給 `QbScoreTBfromName` UDF。使用其中一個 UDF，您也可以指定含有目標影像的直欄和表格名稱。您也可以指定傳回結果表格中的最大橫列數目。如果您沒有指定最大結果數目，則 UDF 會為目標表格直欄中每一個已編目影像傳回一橫列。

在下列查詢中，可使用 `QbScoreTBfromStr` UDF，尋找表格直欄中紋理最接近伺服器檔中影像的 10 個已編目影像。

```
SELECT name, description
FROM fabric
WHERE CAST (swatch_img as varchar(250)) IN
      SELECT CAST (image_id as varchar(25)) FROM TABLE
      (QbScoreTBfromStr
      (QbTextureFeatureClass file=<server,"patterns/ptrn07.gif">' /*query string */
```

```
'fabric',                                     /* table */
'swatch_img',                                /* table column */
10))                                          /* maximum number of results */
AS T1));
```

QBIC 查詢範例程式

第 177 頁的圖 29 顯示在建立 QBIC 查詢中，以 C 撰寫的部份程式碼。圖中的程式碼可按平均色來查詢影像。它提示使用者輸入顏色或影像的名稱。使用者也可以使用查詢傳回的影像，作為後續查詢的範例影像。然後，程式使用已命名的顏色或影像顏色，作為查詢影像直欄的平均色。

您可以在 SAMPLES 子目錄中的 QBICDEMO.C 檔中找到完整程式。完整程式可用來透過直方圖色或位置顏色，以及平均色來查詢影像。若要執行完整程式，您必須執行 ENABLE、POPULATE 和 QBCATDMO 範例程式（也位於 SAMPLES 子目錄中）。有關範例程式的詳細資訊，請參閱第 549 頁的附錄 B，『範例程式和媒體檔案』。

請注意第 177 頁的圖 29：

- 1** 併入 dmbqbapi 標頭檔。
- 2** 提示使用者資料庫資訊。
- 3** 連接資料庫。
- 4** 建立查詢物件。
- 5** 將特性新增到查詢物件。
- 6** 提示使用者鍵入輸入類型（顏色名稱、影像或先前擷取的影像）。
- 7** 指定特性的資料來源。資料來源指平均色的明確規格。
- 8** 發出查詢。Image Extender 會搜尋整個影像直欄。它也指定 10 作為即將傳回的最大影像數目。
- 9** 顯示所傳回之影像集合中的下一張影像。有關顯示影像的進一步資訊，請參閱第 144 頁的『顯示最大圖影像或視訊框』。
- 10** 刪除查詢物件。

SAMPLES 子目錄包括示範如何建立及使用 QBIC 查詢的另一個程式。程式 QbicQry.java 顯示如何以圖形方式指定 QBIC 查詢的搜尋準則。例如，程式會顯示一個顏色選取元來選擇平均色。程式會將選擇轉換成查詢字串。

```

#include <sql.h>
#include <sqlcli.h>
#include <sqlcli1.h>
#include <dmbqbqpi.h> 1
#include <stdio.h>
#include <string.h>
#include <malloc.h>
#include <color.h>
#include <ctype.h>

#define    MaxQueryReturns        10

#define    MaxDatabaseNameLength  SQL_SH_IDENT
#define    MaxUserIdLength        SQL_SH_IDENT
#define    MaxPasswordLength      SQL_SH_IDENT
#define    MaxTableNameLength     SQL_LG_IDENT
#define    MaxColumnNameLength    SQL_LG_IDENT

static char    databaseName[MaxDatabaseNameLength+1];
static char    userID[MaxUserIdLength+1];
static char    password[MaxPasswordLength+1];

static char    tableName[MaxTableNameLength+1];
static char    columnName[MaxColumnNameLength+1];

static char    line[4000];

static QbResult    results[MaxQueryReturns];
static long        currentImage    = -1;
static long        imageCount    = 0;

static char*      tableAndColumn;

static QbQueryHandle    averageHandle = 0;
static QbQueryHandle    histogramHandle = 0;
static QbQueryHandle    drawHandle = 0;
static QbQueryHandle    lastHandle = 0;

static SQLHENV    henv;
static SQLHDBC    hdbc;
static SQLHSTMT    hstmt;
static SQLRETURN    rc;

static char*    listQueries =
    "SELECT NAME,DESCRIPTION FROM MMDBSYS.QBICQUERIES ORDER BY NAME";

static char*    menu[] = {

```

圖 29. QBIC 查詢範例程式 (1/6)

發出 QBIC 查詢

```
/*
1234567890123456789012345678901234567890123456789012345678901234567890 */
""
",
"+-----+",
" | AVERAGE COLOR colorname | ",
" | AVERAGE FILE filename format | ",
" | AVERAGE LAST | ",
" | Press Enter to display the next image in the series | ",
"+-----+",
""
",
0
};

static char* help[] = {
""
",
"AVERAGE Execute an average color query",
" COLOR Specifies the color to query for",
" FILE Specifies the file to compute the average color from",
" LAST Specifies the last displayed image be used to compute the color",
" NEXT Displays the next image from the current query or nothing if",
" all of the image have been displayed."
""
",
">>pause<<",
0
};
/*****
/* doNext() */
*****/
static void doNext(void)
{
    int ret;
    if (currentImage < imageCount)
        currentImage++;
    if (currentImage < imageCount)
        ret = DBiBrowse("/usr/local/bin/xv %s", MMDB_PLAY_HANDLE,
            results[currentImage].imageHandle, MMDB_PLAY_NO_WAIT); 9
}
```

圖 29. QBIC 查詢範例程式 (2/6)

```

/*****
/* doAverage()
/*****
static void doAverage(void)
{
    QbQueryHandle qohandle = 0; QbImageSource is; char* type;
    char* arg1; char* arg2;

    type = nextWord(0);
    if (abbrev(type, "color", 1)) {
        is.type = qbiSource_AverageColor;
        arg1 = nextWord(0);
        if (arg1 == 0) {
            printf("AVERAGE COLOR command requires a colorname argument.\n");
            return;
        }
        if (getColor(arg1, &is.averageColor) == 0) {
            printf("The colorname entered was not recognized.\n");
            return;
        }
    }
    else if (abbrev(type, "file", 1)) {
        is.type = qbiSource_ClientFile;
        arg1 = nextWord(0);
        if (arg1 == 0) {
            printf("AVERAGE FILE command requires a filename argument.\n");
            return;
        }
        arg2 = nextWord(0);
        if (arg2 == 0) {
            printf("AVERAGE FILE command requires a file format argument.\n");
            return;
        }
        strcpy(is.clientFile.fileName, arg1);
        strcpy(is.clientFile.format, arg2);
    }
    else if (abbrev(type, "last", 1)) {
        is.type = qbiSource_ImageHandle;
        if (0 <= currentImage && currentImage < imageCount)
            strcpy(is.imageHandle, results[currentImage]imageHandle);
        else {
            printf("No last image for AVERAGE LAST command\n");
            return;
        }
    }
}

```

圖 29. QBIC 查詢範例程式 (3/6)

發出 QBIC 查詢

```
else {
    printf("AVERAGE command only supports COLOR, FILE, and LAST types.\n");
    return;
}

_QbQuerySetFeatureData(averageHandle, "QbColorFeatureClass", &is); 7
_QbQuerySearch(averageHandle, tableAndColumn, "IMAGE",
    MaxQueryReturns, 0, 0, &imageCount, results); 8
lastHandle = averageHandle;

currentImage = -1;
}
/*****
/* commandLoop() */
*****/
void commandLoop(void)
{
    int    done = 0;

    while (!done) { 6
        displayText(menu);
        printf("%d", currentImage + 1);
        if (0 <= currentImage && currentImage < imageCount)
            printf(" %8.6f", results[currentImage].score);
        printf("> ");
        gets(line);
        done = processCommand(line);
    }
}
```

圖 29. QBIC 查詢範例程式 (4/6)


```

/*****
*/ main()
/*****
void main(void)
{
    char*    inst;
    int      i;

    printf("\n\n");
    printf("Please enter: database_name [user_id] [password] "\n"); 2
    gets(line);
    if (copyWord(line, databaseName, sizeof(databaseName)) == 0)
        exit(0);
    copyWord(0, userid, sizeof(userid));
    copyWord(0, password, sizeof(password));
    printf("\n");

    if (SQLAllocEnv(&henv) != SQL_SUCCESS)
        sqlError(SQL_NULL_HSTMT);
    if (SQLAllocConnect(henv, &hdbc) != SQL_SUCCESS)
        sqlError(SQL_NULL_HSTMT);
    if (SQLConnect(hdbc, 3
        (SQLCHAR*)databaseName,
        SQL_NTS,
        (SQLCHAR*)userid,
        SQL_NTS,
        (SQLCHAR*)password,
        SQL_NTS) != SQL_SUCCESS)
        sqlError(SQL_NULL_HSTMT);

    printf("Initializing . . .\n");

```

圖 29. QBIC 查詢範例程式 (5/6)

發出 QBIC 查詢

```
inst = getenv("DB2INSTANCE");
if (inst != 0 && strcmp(inst, "keesity") == 0)
    tableAndColumn = "KEESEY.TEST";
else
    tableAndColumn = "QBICDEMO.TEST";

_QbQueryCreate(&averageHandle); 4
_QbQueryAddFeature(averageHandle, "QbColorFeatureClass");
_QbQueryCreate(&histogramHandle);
_QbQueryAddFeature(histogramHandle, "QbColorHistogramFeatureClass");
_QbQueryCreate(&drawHandle);
_QbQueryAddFeature(drawHandle, "QbDrawFeatureClass"); 5

commandLoop();

_QbQueryDelete(drawHandle);
_QbQueryDelete(histogramHandle); 10
_QbQueryDelete(averageHandle);

if (SQLDisconnect(hdbc) != SQL_SUCCESS)
    sqlError(SQL_NULL_HSTMT);
if (SQLFreeConnect(hdbc) != SQL_SUCCESS)
    sqlError(SQL_NULL_HSTMT);
if (SQLFreeEnv(henv) != SQL_SUCCESS)
    sqlError(SQL_NULL_HSTMT);
}
```

圖 29. QBIC 查詢範例程式 (6/6)

第 4 篇 參照

第 13 章 使用者定義類型和使用者定義函數	187
綱目	187
使用者定義類型	187
使用者定義函數	187
AlignValue	191
AspectRatio	193
BitsPerSample	194
BytesPerSec	195
Comment	196
CompressType	198
Content	199
DB2Audio	204
DB2Image	207
DB2Video	211
Duration	215
Filename	216
FindInstrument	217
FindTrackName	218
Format	219
FrameRate	220
GetInstruments	221
GetTrackNames	222
Height	223
Importer	224
ImportTime	225
MaxBytesPerSec	226
NumAudioTracks	227
NumChannels	228
NumColors	229
NumFrames	230
NumVideoTracks	231
QbScoreFromName	232
QbScoreFromStr	234
QbScoreTBFromName	235
QbScoreTBFromStr	237
Replace	239
SamplingRate	242
Size	243
Thumbnail	244
TicksPerQNote	246
TicksPerSec	247

Updater	248
UpdateTime	249
Width	250

第 14 章 應用程式設計介面	251
DBaAdminGetInaccessibleFiles	252
DBaAdminGetReferencedFiles	254
DBaAdminIsFileReferenced	256
DBaAdminReorgMetadata	258
DBaDisableColumn	260
DBaDisableDatabase	262
DBaDisableTable	264
DBaEnableColumn	266
DBaEnableDatabase	268
DBaEnableTable	270
DBaGetError	272
DBaGetInaccessibleFiles	274
DBaGetReferencedFiles	276
DBaIsColumnEnabled	278
DBaIsDatabaseEnabled	280
DBaIsFileReferenced	282
DBaIsTableEnabled	284
DBaPlay	286
DBaPrepareAttr	289
DBaReorgMetadata	290
DBiAdminGetInaccessibleFiles	292
DBiAdminGetReferencedFiles	294
DBiAdminIsFileReferenced	296
DBiAdminReorgMetadata	298
DBiBrowse	300
DBiDisableColumn	303
DBiDisableDatabase	305
DBiDisableTable	307
DBiEnableColumn	309
DBiEnableDatabase	311
DBiEnableTable	313
DBiGetError	315
DBiGetInaccessibleFiles	317
DBiGetReferencedFiles	319
DBiIsColumnEnabled	321
DBiIsDatabaseEnabled	323

DBIsFileReferenced	325	QbAddFeature	416
DBIsTableEnabled	327	QbCatalogColumn	418
DBiPrepareAttrs	329	QbCatalogImage.	420
DBiReorgMetadata	330	QbCloseCatalog	422
DBvAdminGetInaccessibleFiles	332	QbCreateCatalog	423
DBvAdminGetReferencedFiles	334	QbDeleteCatalog	425
DBvAdminIsFileReferenced	336	QbGetCatalogInfo	427
DBvAdminReorgMetadata.	338	QbListFeatures	429
DBvBuildStoryboardFile	340	QbOpenCatalog	431
DBvBuildStoryboardTable.	342	QbQueryAddFeature	433
DBvClose.	344	QbQueryCreate	435
DBvCreateIndex.	346	QbQueryDelete	437
DBvCreateIndexFromVideo	348	QbQueryGetFeatureCount	438
DBvCreateShotCatalog.	350	QbQueryGetString	440
DBvDeleteShot	352	QbQueryListFeatures	442
DBvDeleteShotCatalog.	354	QbQueryNameCreate	444
DBvDetectShot	356	QbQueryNameDelete	446
DBvDisableColumn.	358	QbQueryNameSearch	448
DBvDisableDatabase	360	QbQueryRemoveFeature	450
DBvDisableTable	362	QbQuerySearch	452
DBvEnableColumn	364	QbQuerySetFeatureData	454
DBvEnableDatabase	366	QbQuerySetFeatureWeight	456
DBvEnableTable	368	QbQueryStringSearch	458
DBvFrameDataTo24BitRGB	370	QbReCatalogColumn	460
DBvGetError.	372	QbRemoveFeature	462
DBvGetFrame	374	QbSetAutoCatalog	464
DBvGetInaccessibleFiles	376	QbUncatalogImage	466
DBvGetReferencedFiles	378		
DBvInitShotControl.	380	第 15 章 從屬站的管理命令	469
DBvInitStoryboardCtrl	382	輸入 DB2 Extender 管理命令	469
DBvInsertShot	384	取得 DB2 Extender 命令的線上解說.	469
DBvIsColumnEnabled	386	ADD QBIC FEATURE	470
DBvIsDatabaseEnabled	388	CATALOG QBIC COLUMN	471
DBvIsFileReferenced	390	CLOSE QBIC CATALOG	472
DBvIsIndex	392	CONNECT	473
DBvIsTableEnabled.	394	CREATE QBIC CATALOG.	474
DBvMergeShots.	396	DELETE QBIC CATALOG.	476
DBvOpenFile	398	DISABLE COLUMN	477
DBvOpenHandle	400	DISABLE DATABASE	478
DBvPlay	402	DISABLE TABLE.	479
DBvPrepareAttrs	405	DISCONNECT SERVER AT NODENUM (僅 適用於 EEE)	480
DBvReorgMetadata.	406	DISCONNECT SERVER FOR DATABASE (僅適用於 EEE)	481
DBvSetFrameNumber	408	DISCONNECT SERVER FOR DATABASE AT NODENUM (僅適用於 EEE)	482
DBvSetShotComment	410		
DBvUpdateShot	412		
DMBRedistribute (僅適用於 EEE).	414		

ENABLE COLUMN	483
ENABLE DATABASE	484
ENABLE TABLE	485
GET EXTENDER STATUS.	487
GET INACCESSIBLE FILES	488
GET QBIC CATALOG INFO	490
GET REFERENCED FILES.	491
GET SERVER STATUS.	493
OPEN QBIC CATALOG.	494
QUIT	495
RECONNECT SERVER AT NODENUM (僅 適用於 EEE)	496
RECONNECT SERVER FOR DATABASE (僅 適用於 EEE)	497
RECONNECT SERVER FOR DATABASE AT NODENUM (僅適用於 EEE)	498
REDISTRIBUTE NODEGROUP (僅適用於 EEE)	499
REMOVE QBIC FEATURE.	501
REORG	502
SET QBIC AUTOCATALOG	504
START SERVER (僅適用於非 EEE).	505
STOP SERVER (僅適用於非 EEE)	506
TERMINATE	507
 第 16 章 診斷資訊	 509
處理 UDF 回覆碼.	509
處理 API 回覆碼	510
SQLSTATE 訊息碼	510
訊息	515
診斷追蹤.	541
啓動追蹤.	541
停止追蹤.	541
重新格式化追蹤資訊	542
顯示追蹤狀態	542

第 13 章 使用者定義類型和使用者定義函數

本章提供 DB2 Extender 建立的 UDT 及 UDF 參考資料。

綱目

Extender 使用 MMDBSYS 綱目來處理其所有的物件關聯物件，包括 UDT 及 UDF。

使用者定義類型

表 14 列示並說明 DB2 Extender 建立的使用者定義類型。並且列出每一個 UDT 的 DB2 來源資料類型。

表 14. DB2 Extender 建立的使用者定義類型

UDT	來源資料類型	說明
DB2IMAGE	VARCHAR(250)	影像控點。變數長度字串含有存取影像物件所需的資訊。影像控點儲存在為 Image Extender 啓用的使用者表格直欄中。
DB2AUDIO	VARCHAR(250)	音效控點。變數長度字串含有存取音效物件所需的資訊。音效控點儲存在為 Audio Extender 啓用的使用者表格直欄中。
DB2VIDEO	VARCHAR(250)	視訊控點。變數長度字串含有存取視訊物件所需的資訊。視訊控點儲存在為 Video Extender 啓用的使用者表格直欄中。

使用者定義函數

本節提供 DB2 Extender 的參照資訊。UDF 按字母順序清單。

下列資訊會為每個 UDF 提供：

- 提供 UDF 的 Extender
- 簡要說明
- UDF 的併入 (標頭) 檔
- UDF 的 SQL 語法
- UDF 參數的說明，包括其資料類型

使用者定義函數

- UDF 傳回的值，包括其資料類型
- 使用範例

表 15列出 UDF，並識別提供每個 UDF 的 Extender。本表格也指出可讓您取得有關每一個 UDF 詳細資訊的位置。本表格中的 UDF 可使用內含的 SQL 陳述式或 DB2 CLI 呼叫來編碼。

表 15. DB2 Extender UDF

UDF	說明	影像	音效	視訊	請參閱頁面
AlignValue	傳回 WAVE 音效或視訊的音效磁軌中，每一取樣的位元組數目。		x	x	191
AspectRatio	傳回 MPEG1 和 MPEG2 視訊第一個磁軌的縱橫比。			x	193
BitsPerSample	傳回用來代表音效中 WAVE 或 AIFF 音效，或視訊的音效磁軌中每一個範例的資料位元數目。		x	x	194
BytesPerSec	傳回 WAVE 音效的資料傳送速率，以每秒平均位元組為單位。		x		195
Comment	傳回或更新以影像、音效或視訊所儲存的註解。	x	x	x	196
CompressType	傳回視訊壓縮格式，例如 MPEG-1。			x	198
Content	擷取或更新資料庫中影像、音效或視訊的內容。	x	x	x	199
DB2Audio	將音效內容儲存在資料庫表格中。		x		204
DB2Image	將影像內容儲存在資料庫表格中。	x			207
DB2Video	將視訊內容儲存在資料庫表格中。			x	211
Duration	傳回 WAVE 或 AIFF 音效，或視訊的持續時間 (亦即，以秒為單位的播放時間)。		x	x	215
Filename	傳回含有影像、音效或視訊內容的伺服器檔案名稱。	x	x	x	216
FindInstrument	傳回 MIDI 音效中指定樂器第一次出現的磁軌號碼。		x		217
FindTrackName	傳回 MIDI 音效中指定磁軌的號碼。		x		218
Format	傳回影像、音效或視訊的格式。	x	x	x	219

表 15. DB2 Extender UDF (繼續)

UDF	說明	影像	音效	視訊	請參閱頁面
FrameRate	傳回視訊產量 (以每秒框數為單位)。			x	220
GetInstruments	傳回 MIDI 音效中所有樂器的樂器名稱。		x		221
GetTrackNames	傳回 MIDI 音效中所有磁軌的名稱。		x		222
Height	傳回影像或視訊框的高度 (以圖點為單位)。	x		x	223
Importer	傳回將影像、音效或視訊儲存在資料庫表格中的使用者之使用者 ID。	x	x	x	224
ImportTime	傳回影像、音效或視訊儲存在資料庫表格中時指示的時間戳記。	x	x	x	225
MaxBytesPerSec	傳回最大視訊產量 (以每秒位元組為單位)。			x	226
NumAudioTracks	傳回視訊或 MIDI 音效中音效磁軌的數目。		x	x	227
NumChannels	傳回 WAVE 或 AIFF 音效，或視訊中記錄的音效通道數目。		x	x	228
NumColors	傳回影像中顏色的數目。	x			229
NumFrames	傳回視訊中的框數。			x	230
NumVideoTracks	傳回視訊中視訊磁軌的數目。			x	231
QbScoreFromName	傳回影像的分數 (使用已命名查詢物件)。(取代 QbScore。)	x			232
QbScoreFromStr	傳回影像的分數 (使用查詢字串)。	x			234
QbScoreTBFromName	傳回影像直欄中分數的表格 (使用已命名查詢物件)。	x			235
QbScoreTBFromStr	傳回影像直欄中分數的表格 (使用查詢字串)。	x			237
Replace	更新資料庫中儲存的影像、音效或視訊內容，並更新它的註解。	x	x	x	239
SamplingRate	傳回 WAVE 或 AIFF 音效，或視訊中音效磁軌的取樣率 (以每秒範例數目為單位)。		x	x	242

使用者定義函數

表 15. DB2 Extender UDF (繼續)

UDF	說明	影像	音效	視訊	請參閱頁面
Size	傳回影像、音效或視訊的大小 (以位元組為單位)。	x	x	x	243
Thumbnail	傳回或更新資料庫中儲存的視訊框或影像縮圖大小版本。	x		x	244
TicksPerQNote	傳回記錄的 MIDI 音效之計時速度 (以每四分之一音符刻點數為單位)。		x		246
TicksPerSec	傳回記錄的 MIDI 音效之計時速度 (以每秒刻點數為單位)。		x		247
Updater	傳回最後更新資料庫表格中影像、音效或視訊的使用者之使用者 ID。	x	x	x	248
UpdateTime	傳回最後更新資料庫表格中影像、音效或視訊時指示的時間戳記。	x	x	x	249
Width	傳回影像或視訊框的寬度 (以圖點為單位)。	x		x	250

AlignValue

影像	音效	視訊
	X	X

傳回 WAVE 音效或視訊的音效磁軌中，每一取樣的位元組數目。 WAVE 音效可利用下列方式儲存它的資料：每一取樣一個位元組 (8-bit 單音，稱為『以位元組為準』)、每一取樣兩個位元組 (8-bit 立體音 16-bit 單音，稱為『以字為準』)，或每一取樣四個位元組 (16-bit 立體音，稱為『以雙字為準』)。

併入檔

音效 dmbaudio.h
視訊 dmbvideo.h

語法

▶▶AlignValue(—handle—)————▶▶

參數 (資料類型)

handle (DB2AUDIO 或 DB2VIDEO)
含有音效控點的主變數或直欄名稱。

回覆值 (資料類型)

WAVE 音效或視訊中音效磁軌每一取樣值的位元組數目 (SMALLINT)。這些值可以是：

- 1 以位元組為準
- 2 以字為準
- 4 以雙字為準
- NULL 值** 其它格式的音效

範例

取得員工表格的聲音直欄中所有以字為準的音效檔名：

```
EXEC SQL BEGIN DECLARE SECTION;  
char hvAud_fname[251];  
EXEC SQL END DECLARE SECTION;  
  
EXEC SQL SELECT FILENAME (SOUND)  
INTO :hvAud_fname  
FROM EMPLOYEE  
WHERE ALIGNVALUE(SOUND) = 2;
```

AlignValue

尋找視訊中音效磁軌每一取樣值的位元組數目；此視訊儲存在 Anita Jones 的員工表格的視訊直欄中：

```
EXEC SQL BEGIN DECLARE SECTION;
      short hvAlign_val;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT ALIGNVALUE(VIDEO)
      INTO :hvAlign_val
      FROM EMPLOYEE
      WHERE NAME='Anita Jones';
```

AspectRatio

影像	音效	視訊
		X

傳回 MPEG 視訊第一個磁軌的縱橫比。

併入檔

dmbvideo.h

語法

►—AspectRatio—(—handle—)—————►

參數 (資料類型)

handle (DB2VIDEO)

含有視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

MPEG 視訊第一個磁軌的縱橫比，或其它格式中視訊的 NULL 值 (SMALLINT)

範例

取得員工表格的視訊直欄中，儲存在 Robert Smith 的視訊縱橫比：

```
EXEC SQL BEGIN DECLARE SECTION;
      short hvAsp_ratio;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT ASPECTRATIO(VIDEO)
      INTO :hvAsp_ratio
      FROM EMPLOYEE
      WHERE NAME='Robert Smith';
```

BitsPerSample

影像	音效	視訊
	X	X

傳回用來代表音效中 WAVE 或 AIFF 音效，或視訊的音效磁軌中每一個範例的資料位元數目。

併入檔

音效 dmbaudio.h

視訊 dmbvideo.h

語法

►►—BitsPerSample—(—handle—)————►◄

參數 (資料類型)

handle (DB2AUDIO 或 DB2VIDEO)

含有視訊或音效控點的主變數或直欄名稱。

回覆值 (資料類型)

用來代表視訊，WAVE 或 AIFF 音效每一取樣的資料位元數目 (SMALLINT)。以其它格式傳回音效的 NULL 值

範例

取得儲存在員工表格的聲音直欄中，每一取樣位元數等於 8 的所有 WAVE 音效之檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvAud_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (SOUND)
      INTO :hvAud_fname
      FROM EMPLOYEE
      WHERE FORMAT(SOUND)='WAVE'
      AND BITSPERSAMPLE(SOUND) = 8;
```

BytesPerSec

影像	音效	視訊
	X	

傳回 WAVE 音效的資料傳送速率，以每秒平均位元組為單位。

併入檔

dmbaudio.h

語法

►►BytesPerSec(—handle—)—————►►

參數 (資料類型)

handle (DB2AUDIO)

含有音效控點的主變數或直欄名稱。

回覆值 (資料類型)

資料傳送速率 (INTEGER)。以其它格式傳回音效的 NULL 值。

範例

取得儲存在員工表格的聲音直欄中，傳送速率 (以每秒平均位元組數目為單位) 小於 44100 的所有音效檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvAud_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (SOUND)
      INTO :hvAud_fname
      FROM EMPLOYEE
      WHERE BYTESPERSEC(SOUND) < 44100;
```

Comment

影像	音效	視訊
X	X	X

傳回或更新以影像、音效或視訊所儲存的註解。

併入檔

影像	dmbimage.h
音效	dmbaudio.h
視訊	dmbvideo.h

語法

擷取註解

►►Comment—(—handle—)————►►

語法

更新註解

►►Comment—(—handle—,—new_comment—)————►►

參數 (資料類型)

handle (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)

含有影像、音效或視訊控點的主變數或直欄名稱。

new_comment (LONG VARCHAR)

用來更新的新註解。NULL 值或空字串會刪除現存的註解。

回覆值 (資料類型)

以更新而言，指影像、音效或視訊的控點 (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)。以擷取而言，指註解 (LONG VARCHAR)。

範例

取得員工表格的圖片直欄中，具有相關註解中 『confidential』 這個字的所有影像檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_fname[255;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
```



```

        INTO :hvImg_fname
        FROM EMPLOYEE
        WHERE COMMENT(PICTURE)
              LIKE '%confidential%';

```

更新員工表格的視訊直欄中，與 Anita Jones 的視訊片段相關的註解：

```

EXEC SQL BEGIN DECLARE SECTION;
struct{
    short len;
    char data[4000];
}hvRemarks;
EXEC SQL END DECLARE SECTION;

/* Get the old comment */

EXEC SQL SELECT COMMENT(VIDEO)
        INTO :hvRemarks
        FROM EMPLOYEE
        WHERE NAME = 'Anita Jones';

/* Update the comment */

hvRemarks.data[hvRemarks.len]='\0';
strcat (hvRemarks.data, "Updated video");
hvRemarks.len=strlen(hvRemarks.data);

EXEC SQL UPDATE EMPLOYEE
        SET VIDEO=COMMENT(VIDEO, :hvRemarks)
        WHERE NAME = 'Anita Jones';

```

CompressType

CompressType

影像	音效	視訊
		X

傳回視訊壓縮格式，例如 MPEG-1。

併入檔

dmbvideo.h

語法

►—CompressType—(—handle—)————►◄

參數 (資料類型)

handle (DB2VIDEO)

含有視訊控點的主變數或直欄名稱

回覆值 (資料類型)

視訊的壓縮格式 (VARCHAR(8))

範例

取得儲存在員工表格的視訊直欄中，其壓縮格式為 MPEG-1 的所有視訊名稱：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvVid_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (VIDEO)
      INTO :hvVid_fname
      FROM EMPLOYEE
      WHERE COMPRESSTYPE(VIDEO) = 'MPEG1';
```

Content

影像	音效	視訊
X	X	X

擷取或更新資料庫中影像、音效或視訊的內容。 此內容可被擷取到從屬站緩衝區、從屬站檔案、或伺服器檔案。

併入檔

影像	dmbimage.h
音效	dmbaudio.h
視訊	dmbvideo.h

語法

將內容擷取到緩衝區或從屬站檔案

►►Content—(—handle—)—————►◄

語法

將內容區段擷取到緩衝區或從屬站檔案

►►Content—(—handle—,—offset—,—size—)—————►◄

語法

將內容擷取到伺服器檔案

►►Content—(—handle—,—target_file—,—overwrite—)—————►◄

語法

將內容擷取到緩衝區或 從屬站檔案，並做格式轉換 -- 僅適用於影像

►►Content—(—handle—,—target_format—)—————►◄

語法

將內容擷取到伺服器檔案，並做格式轉換 -- 僅適用於影像

►►Content—(—handle—,—target_file—,—overwrite—,—target_format—)—————►◄

語法

將內容擷取到緩衝區或從屬站檔案，並做格式轉換及其他變更 -- 僅適用於影像

►►Content—(—handle—,—target_format—,—conversion_options—)—————►◄

語法

將內容擷取到伺服器檔案，並做格式轉換及其他變更 -- 僅適用於影像

```
►►Content—(—handle—,—target_file—,—overwrite—,—  
►target_format—,—conversion_options—)
```

語法

更新緩衝區或從屬站檔案的內容

```
►►Content—(—handle—,—content—,—source_format—,—target_file—)
```

語法

更新伺服器檔案的內容

```
►►Content—(—handle—,—source_file—,—source_format—,—stortype—)
```

語法

更新緩衝區或從屬站檔案中含有使用者提供屬性的內容

```
►►Content—(—handle—,—content—,—target_file—,—attrs—,—thumbnail—)
```

語法

更新伺服器檔案中含有使用者提供屬性的內容

```
►►Content—(—handle—,—source_file—,—stortype—,—attrs—,—thumbnail—)
```

語法

更新緩衝區或從屬站檔案的內容，並做格式轉換 -- 僅適用於影像

```
►►Content—(—handle—,—content—,—source_format—,—  
►target_format—,—target_file—)
```

語法

更新伺服器檔案的內容，並做格式轉換 -- 僅適用於影像

```
►►Content—(—handle—,—source_file—,—source_format—,—  
►target_format—,—target_file—)
```

語法

更新緩衝區或從屬站檔案的內容，並做格式轉換及其他變更 -- 僅適用於影像

```
►►Content—(—handle—,—content—,—source_format—,—  
►target_format—,—conversion_options—,—target_file—)
```

語法

更新伺服器檔案的內容，並做格式轉換及其他變更 -- 僅適用於影像

```
►—Content—(—handle—,—source_file—,—source_format—,——————►
►—target_format—,—conversion_options—,—target_file—)—————►
```

參數 (資料類型)**handle (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)**

含有影像、音效或視訊控點的主變數或直欄名稱。

offset (INTEGER)

啟動所要擷取的視訊、音效或影像的偏移 (原點 1)。

size (INTEGER)

所要擷取視訊、音效或影像的位元組數目。

source_file (LONG VARCHAR)

含有影像、音效或視訊更新內容的檔案名稱。

target_file (LONG VARCHAR)

以擷取而言，指所要擷取視訊、音效或影像的檔名。以更新而言，指含有所要更新視訊、音效或影像的檔名。

stortype (INTEGER)

一種指示將儲存已更新影像、音效或視訊位置的值。常數 MMDB_STORAGE_TYPE_INTERNAL (值=1)，指示已更新物件將被存入 BLOB 資料庫。常數 MMDB_STORAGE_TYPE_EXTERNAL (值=0)，指示已更新物件將被存入伺服器檔案。

overwrite (INTEGER)

一種指示是否改寫已存在的目標檔案之值。此值可以是 0 或 1。0 值表示目標檔案不會被改寫 (實際上，擷取並不會發生)。1 值表示已存在的目標檔案會被改寫。

target_format (VARCHAR(8))

擷取或更新後的影像格式。來源影像格式將被轉換成合適的格式。以伺服器檔案的影像擷取而言，如果 target_file 與 source_file 相同，則目標格式必須與來源格式相同。若是 MPG1 格式，您可以指定 MPG1、mpeg1、MPEG1 或 mpeg1。若是 MPG2 格式，您可以指定 MPG2、mpeg2、MPEG2 或 mpeg2。

conversion_options (VARCHAR(100))

指定擷取或更新影像時所要引用的一些變更，例如旋轉和壓縮。有關支援的轉換選項，請參閱第 106 頁的表 9。

content (BLOB(2G) AS LOCATOR)

含有影像、音效或視訊更新內容的主變數。主變數可以是類型 BLOB、BLOB_FILE 或 BLOB_LOCATOR。DB2 會將內容的資料類型提升為 BLOB_LOCATOR，並將 LOB 定位器傳遞到「Content UDF」。

source_format (VARCHAR(8))

更新影像、音效或視訊的來源格式。您可以指定 NULL 值或空白字串，或僅對影像指定字串 ASIS；在這三種情形下，Extender 會嘗試自動決定格式。若是 MPG1 格式，您可以指定 MPG1、mpg1、MPEG1 或 mpeg1。若是 MPG2 格式，您可以指定 MPG2、mpg2、MPEG2 或 mpeg2。

attrs (LONG VARCHAR FOR BIT)

影像、音效或視訊的屬性

thumbnail (LONG VARCHAR FOR BIT DATA)

影像或視訊框的縮圖（僅適用於影像和視訊）

回覆值 (資料類型)

如果擷取到緩衝區，則是擷取的影像、音效或視訊的內容 (BLOB(2G) AS LOCATOR)。如果是擷取到檔案，則是 VARCHAR(254)。

以更新而言，指所要更新的影像、音效或視訊的控點 (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)。

範例

將員工表格的圖片直欄中，儲存在 Anita Jones 的影像擷取到伺服器檔案：

```
struct{
    short len;
    char data[250];
}hvImg_fname;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT CONTENT (PICTURE,
    '/employee/images/ajones.bmp',1)
    INTO :hvImg_fname
    FROM EMPLOYEE
    WHERE NAME='Anita Jones';
```

將儲存在員工表格的聲音直欄中 Robert Smith 的 1-MB 音效片段擷取到從屬站緩衝區：

```
EXEC SQL BEGIN DECLARE SECTION;
    SQL TYPE IS BLOB LOCATOR audio_loc;
EXEC SQL END DECLARE SECTION;
```

```
EXEC SQL SELECT CONTENT (SOUND, 1, 1000000)
      INTO :audio_loc
      FROM EMPLOYEE
      WHERE NAME='Robert Smith';
```

更新員工表格的圖片直欄中 Anita Jones 的影像；將影像格式從 BMP 轉換成 GIF，並將影像縮減成其原始大小的 50%：

```
EXEC SQL UPDATE EMPLOYEE
      SET picture = CONTENT(PICTURE,
          '/employee/newimg/ajones.bmp',
          'BMP',
          'GIF',
          '-s 0.5',
          '');
      WHERE NAME='Anita Jones';
```

DB2Audio

影像	音效	視訊
	X	

將音效內容儲存在資料庫表格中。音效來源可存放於從屬站緩衝區、從屬站檔案或伺服器檔案。音效可以 BLOB 的形式儲存在資料庫表格或伺服器檔案中 (供資料庫表格參照)。音效來源可以在支援的格式中，這種情形下， DB2Audio Extender 會識別儲存體的屬性，或在不支援的格式中，必須在 UDF 中指定這些屬性。

併入檔

dmbaudio.h

語法

儲存緩衝區或從屬站檔案的內容

```
►►DB2Audio—(—dbname—,—content—,—format—,—target_file—,—comment—)————►◄
```

語法

儲存伺服器檔案的內容

```
►►DB2Audio—(—dbname—,—source_file—,—format—,—stortype—,—comment—)————►◄
```

語法

儲存緩衝區或從屬站檔案中含有使用者提供屬性的內容

```
►►DB2Audio—(—dbname—,—content—,—target_file—,—comment—,—attrs—)————►◄
```

語法

儲存伺服器檔案中含有使用者提供屬性的內容

```
►►DB2Audio—(—dbname—,—source_file—,—stortype—,—comment—,—attrs—)————►◄
```

參數 (資料類型)

dbname (VARCHAR(18))

目前連接的資料庫名稱，可由 CURRENT SERVER 特別暫存區指示。

content (BLOB(2G) AS LOCATOR)

含有音效內容的主變數。主變數可以是類型 BLOB、BLOB_FILE 或 BLOB_LOCATOR。DB2 會將內容的資料類型提升為 BLOB-LOCATOR，並將 LOB 定位器傳遞到 DB2Audio UDF。

format (VARCHAR(8))

來源音效的格式。在指定 NULL 值或空白字串的情形下， Audio Extender

會嘗試自動決定來源格式。音效將以其和來源相同的格式存檔。有關支援的音效格式，請參閱第 105 頁的表 8。

target_file (LONG VARCHAR)

目標伺服器檔案的名稱 (用於伺服器檔案的儲存體)，或 NULL 值或空白字串 (用於作為 BLOB 的資料庫表格中儲存體)。目標檔案可以是完整名稱。如果名稱不完整，則使用伺服器上的 DB2AUDIOSTORE 及 DB2MMSTORE 環境變數來尋找檔案。

source_file (LONG VARCHAR)

來源伺服器檔案的名稱。來源檔名稱可以是完整名稱或不完整名稱；但不能是 NULL 值或空白字串。如果名稱不完整，則使用伺服器上的 DB2AUDIOPATH 及 DB2MMPATH 環境變數來尋找檔案。

stortype (INTEGER)

一種指示音效儲存位置的值。常數 MMDB_STORAGE_TYPE_INTERNAL (值=1)，指示音效將被存入 BLOB 資料庫中；常數 MMDB_STORAGE_TYPE_EXTERNAL (值=0)，指示音效內容將被存入伺服器檔案中 (指示從資料庫中)。

comment (LONG VARCHAR)

使用音效儲存的註解。

attrs (LONG VARCHAR FOR BIT DATA)

音效屬性。

回覆值 (資料類型)

音效控點 (DB2AUDIO)

範例

將含有 Anita Jones 的音效片段之記錄插入員工表格。音效來源位於從屬站緩衝區中。將音效片段存入表格作為 BLOB：

```
EXEC SQL BEGIN DECLARE SECTION;
      SQL TYPE IS BLOB (5M) aud_seg;
EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
      DB2AUDIO(
        CURRENT SERVER,
        :aud_seg,
        'WAVE',
        CAST(NULL as LONG VARCHAR),
        'Anita''s voice'));;
```

DB2Audio

將含有 Robert Smith 的音效片段之記錄插入員工表格。音效來源位於伺服器檔案中。員工表格記錄會指向檔案。

```
EXEC SQL BEGIN DECLARE SECTION;
      long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType = MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '384779',
      'Robert Smith',
      DB2AUDIO(
        CURRENT SERVER,
        '/employee/sounds/rsmith.wav',
        'WAV',
        :hvStorageType,
        'Robert''s voice'));;
```

將含有 Anita Jones 的音效片段之記錄插入員工表格。將音效片段存成 BLOB。位於伺服器檔案中的來源音效片段，其具有使用者定義的格式、取樣率 44.1 KHz 以及兩個記錄通道。

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
struct {
    short len;
    char data[600];
} hvAudattr;
EXEC SQL END DECLARE SECTION;

MMDBAudioAttrs      *paudiattr;

hvStorageType = MMDB_STORAGE_TYPE_INTERNAL;

paudioattr=(MMDBAudioAttrs *) hvAudattr.data;
strcpy(paudioAttr->cFormat,"cFormatA");
paudioAttr->u1SamplingRate=44100;
paudioAttr->usNumChannels=2;
hvAudattr.len=sizeof(MMDBAudioAttrs);

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
      DB2AUDIO(
        CURRENT SERVER,
        '/employee/sounds/ajones.aud',
        :hvStorageType,
        'Anita''s voice',
        :hvAudattr)
      );;
```

DB2Image

影像	音效	視訊
X		

將影像內容儲存在資料庫表格中。影像來源可存放於從屬站緩衝區、從屬站檔案、或伺服器檔案中。影像可以 BLOB 的形式儲存在資料庫表格或伺服器檔案中 (供資料庫表格參照)。影像來源可以位於所支援的格式中，這種情形下，DB2Image Extender 會識別儲存體的屬性，或在不支援的格式中，必須在 UDF 中指定這些屬性。

併入檔

dmbimage.h

語法

儲存緩衝區或從屬站檔案的內容

```
►►—DB2Image—(—dbname—,—content—,—source_format—,——————►
►—target_file—,—comment—)——————►◄◄
```

語法

儲存伺服器檔案的內容

```
►►—DB2Image—(—dbname—,—source_file—,—source_format—,——————►
►—stortype—,—comment—)——————►◄◄
```

語法

儲存緩衝區或從屬站檔案中含有使用者提供屬性的內容

```
►►—DB2Image—(—dbname—,—content—,—target_file—,——————►
►—comment—,—attrs—,—thumbnail—)——————►◄◄
```

語法

儲存伺服器檔案中含有使用者提供屬性的內容

```
►►—DB2Image—(—dbname—,—source_file—,—stortype—,—comment—,——————►
►—attrs—,—thumbnail—)——————►◄◄
```

語法

儲存緩衝區或從屬站檔案的內容，並做格式轉換

DB2Image

```
►►DB2Image—(—dbname—,—content—,—source_format—,——————►
►—target_format—,—target_file—,—comment—)—————►◄
```

語法

儲存伺服器檔案內容，並做格式轉換

```
►►DB2Image—(—dbname—,—source_file—,—source_format—,——————►
►—target_format—,—target_file—,—comment—)—————►◄
```

語法

儲存緩衝區或從屬站檔案的內容，並做格式轉換及其他變更

```
►►DB2Image—(—dbname—,—content—,—source_format—,——————►
►—target_format—,—conversion_options—,—target_file—,—comment—)—————►◄
```

語法

儲存伺服器檔案內容，並做格式轉換及其他變更

```
►►DB2Image—(—dbname—,—source_file—,—source_format—,——————►
►—target_format—,—conversion_options—,—target_file—,—comment—)—————►◄
```

參數 (資料類型)

dbname (VARCHAR(18))

目前連接的資料庫名稱，可由 CURRENT SERVER 特別暫存區指示。

content (BLOB(2G) AS LOCATOR)

含有影像內容的主變數。主變數可以是類型 BLOB、BLOB_FILE 或 BLOB_LOCATOR。DB2 會將內容的資料類型提升為 BLOB_LOCATOR，並將 LOB 定位器傳遞到 DB2Image UDF。

source_format (VARCHAR(8))

來源影像格式。您可以指定 NULL 值、空白字串或字串 ASIS；在其中一種情形下，Image Extender 會嘗試自動決定來源格式。影像將以其和來源相同的格式存檔。有關支援的影像格式，請參閱第 105 頁的表 8。

target_format (VARCHAR(8))

儲存後的影像格式。來源影像格式將被轉換成合適的格式。

target_file (LONG VARCHAR)

目標伺服器檔案的名稱 (用於伺服器檔案的儲存體)，或 NULL 值或空白字串 (用於作為 BLOB 的資料庫表格中儲存體)。目標檔名可以是完整名稱。如果名稱不完整，則使用伺服器上的 DB2IMAGESTORE 和

DB2MMSTORE 環境變數來尋找檔案。如果使用格式轉換儲存影像，則必須在 DB2IMAGEPATH 和 DB2MMPATH 環境變數中指定目標檔的路徑。

source_file (LONG VARCHAR)

來源伺服器檔案的名稱。來源檔名稱可以是完整名稱或不完整名稱；但不能是 NULL 值或空白字串。如果名稱不完整，則使用伺服器上的 DB2IMAGEPATH 及 DB2MMPATH 環境變數來尋找檔案。

stortype (INTEGER)

一種指示影像儲存位置的值。常數 MMDB_STORAGE_TYPE_INTERNAL (值=1)，指示影像將被存入 BLOB 資料庫中；常數 MMDB_STORAGE_TYPE_EXTERNAL (值=0)，指示影像內容將被存入伺服器檔案中 (指示從資料庫中)。

comment (LONG VARCHAR)

使用影像儲存的註解。

attrs (LONG VARCHAR FOR BIT DATA)

影像屬性。

thumbnail (LONG VARCHAR FOR BIT DATA)

影像的縮圖。

conversion_options (VARCHAR(100))

指定儲存影像時所要引用的一些變更，例如旋轉和壓縮。有關支援的轉換選項，請參閱第 106 頁的表 9。

回覆值 (資料類型)

影像控點 (DB2IMAGE)

範例

將含有 Anita Jones 的影像記錄插入員工表格。影像來源是位於從屬站緩衝區中。將影像存入表格作為 BLOB：

```
EXEC SQL BEGIN DECLARE SECTION
      SQL TYPE IS BLOB (2M) hvImg
EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
      '128557',
      'Anita Jones',
      DB2IMAGE(
        CURRENT SERVER,
        :hvImg,
        'ASIS',
        CAST(NULL as LONG VARCHAR),
        'Anita''s picture'));;
```

DB2Image

將含有 Robert Smith 的影像記錄插入員工表格。影像來源是位於伺服器檔案中。員工表格記錄會指向檔案。存檔時，將影像格式從 BMP 轉換成 GIF。並將影像修剪成寬 110 圖點和高 150 圖點，以及使用 LZW 類型壓縮法來壓縮影像：

```
EXEC SQL INSERT INTO EMPLOYEE VALUES(  
    '384779',  
    'Robert Smith',  
    DB2IMAGE(  
        CURRENT SERVER,  
        '/employee/pictures/rsmith.bmp',  
        'BMP',  
        'GIF',  
        '-x 110 -y 150 -c 14',  
        '',  
        'Robert"s picture'));
```

將含有 Robert Smith 的影像記錄插入員工表格。位於伺服器檔案中的來源影像，其具有使用者定義的格式、高 640 圖點及寬 480 圖點。將影像存成 BLOB：

```
EXEC SQL BEGIN DECLARE SECTION;  
long hvStorageType;  
struct {  
    short len;  
    char data[400];  
} hvImgattrs;  
EXEC SQL END DECLARE SECTION;  
  
DB2IMAGEATTRS    *pimgattr;  
  
hvStorageType = MMDB_STORAGE_TYPE_INTERNAL;  
  
pimgattr = (DB2IMAGEATTRS *) hvImgattrs.data;  
strcpy(pimgattr->cFormat, "FormatI");  
pimgattr->width=640;  
pimgattr->height=480;  
hvImgattrs.len=sizeof(DB2IMAGEATTRS);  
  
EXEC SQL INSERT INTO EMPLOYEE VALUES(  
    '128557',  
    'Anita Jones',  
    DB2IMAGE(  
        CURRENT SERVER,  
        '/employee/images/ajones.bmp',  
        :hvStorageType,  
        'Anita"s picture',  
        :hvImgattrs,  
        CAST(NULL as LONG VARCHAR))  
    );
```

DB2Video

影像	音效	視訊
		X

將視訊內容儲存在資料庫表格中。視訊來源可存放於從屬站緩衝區、從屬站檔案或伺服器檔案。視訊可以 BLOB 的形式儲存在資料庫表格或伺服器檔案中 (供資料庫表格參照)。視訊來源可以位於所支援的格式中，這種情形下，DB2Video Extender 會識別儲存體的屬性，或在不支援的格式中，必須在 UDF 中指定這些屬性。

併入檔

dmbvideo.h

語法

儲存緩衝區或從屬站檔案的內容

```
►►DB2Video—(—dbname—,—content—,—format—,—target_file—,—comment—)————►◄
```

語法

儲存伺服器檔案的內容

```
►►DB2Video—(—dbname—,—source_file—,—format—,—stortype—,—comment—)————►◄
```

語法

儲存緩衝區或從屬站檔案中含有使用者提供屬性的內容

```
►►DB2Video—(—dbname—,—content—,—target_file—,—————►
►—comment—,—attrs—,—thumbnail—)————►◄
```

語法

儲存伺服器檔案中含有使用者提供屬性的內容

```
►►DB2Video—(—dbname—,—source_file—,—stortype—,—comment—,—————►
►—attrs—,—thumbnail—)————►◄
```

參數 (資料類型)

dbname (VARCHAR(18))

目前連接的資料庫名稱，可由 CURRENT SERVER 特別暫存區指示。

content (BLOB(2G) AS LOCATOR)

含有視訊內容的主變數。主變數可以是資料類型 BLOB、BLOB_FILE 或 BLOB_LOCATOR。DB2 會將內容提升為 BLOB_LOCATOR，並將 LOB

定位器傳遞到 DB2Video UDF。如果內容位於從屬站緩衝區中，則緩衝區至少要有 640 KB 內容，以確保整個視訊標頭可被讀取。

format (VARCHAR(8))

來源視訊格式。若指定 NULL 值或空白字串，Video Extender 會嘗試自動決定來源格式。視訊將以其和來源相同的格式存檔。關於支援的視訊格式，請參閱第 105 頁的表 8。若是 MPG1 格式，您可以指定 MPG1、mpg1、MPEG1 或 mpeg1。若是 MPG2 格式，您可以指定 MPG2、mpg2、MPEG2 或 mpeg2。

target_file (LONG VARCHAR)

目標伺服器檔案的名稱 (用於伺服器檔案的儲存體)，或 NULL 值或空白字串 (用於作為 BLOB 的資料庫表格中儲存體)。伺服器檔案必須是完整名稱。如果檔名不完整，則使用伺服器上的 DB2VIDEOSTORE 及 DB2MMSTORE 環境變數來尋找檔案。

source_file (LONG VARCHAR)

來源伺服器檔案的名稱。來源檔名稱可以是完整名稱或不完整名稱；但不能是 NULL 值或空白字串。如果名稱不完整，則使用伺服器上的 DB2VIDEOPATH 及 DB2MMPATH 環境變數來尋找檔案。

stortype (INTEGER)

一種指示視訊儲存位置的值。常數 MMDB_STORAGE_TYPE_INTERNAL (值=1)，指示視訊將被存入 BLOB 資料庫中；常數 MMDB_STORAGE_TYPE_EXTERNAL (值=0)，指示視訊內容將被存入伺服器檔案中 (指示從資料庫中)。

comment (LONG VARCHAR)

使用視訊儲存的註解。

attrs (LONG VARCHAR FOR BIT DATA)

視訊屬性。

thumbnail (LONG VARCHAR FOR BIT DATA)

代表視訊的縮圖影像。

回覆值 (資料類型)

視訊控點 (DB2VIDEO)

範例

將含有 Anita Jones 的視訊片段之記錄插入員工表格。視訊來源是位於從屬站緩衝區中。將視訊存入表格作為 BLOB：

```
EXEC SQL BEGIN DECLARE SECTION
      SQL TYPE IS BLOB (8M) vid;
EXEC SQL END DECLARE SECTION;
```



```
EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2VIDEO(
        CURRENT SERVER,
        :vid,
        'MPEG1',
        CAST(NULL as LONG VARCHAR),
        'Anita''s video'));
```

將含有 Robert Smith 的視訊片段之記錄插入員工表格。視訊來源是位於伺服器檔案中。員工表格記錄會指向檔案：

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType = MMDB_STORAGE_TYPE_EXTERNAL;

EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '384779',
    'Robert Smith',
    DB2VIDEO(
        CURRENT SERVER,
        '/employee/videos/rsmith.mpg',
        'MPEG1',
        :hvStorageType,
        'Robert''s video'));
```

將含有視訊片段的記錄插入資料庫表格。位於伺服器檔案中的來源視訊片段，具有使用者定義的格式。將視訊內容保存在伺服器檔案中（資料庫表格記錄會指向檔案）。並儲存代表視訊的縮圖：

```
EXEC SQL BEGIN DECLARE SECTION;
long hvStorageType;
struct {
    short len;
    char data[400];
}hvVidattrs;
struct {
    short len;
    char data[10000];
}hvThumbnail;
EXEC SQL END DECLARE SECTION;

MMDBVideoAttrs      *pvideoAttr;

hvStorageType = MMDB_STORAGE_TYPE_EXTERNAL;

pvideoAttr=(MMDBVideoAttrs *)hvVidattrs.data;
strcpy(pvideoAttr->cFormat,"Formatv");
pvideoAttr->len=sizeof(MMDBVideoAttrs);
```

DB2Video

```

:
:
/* Generate thumbnail and assign data */
/* in video structure */
:
EXEC SQL INSERT INTO EMPLOYEE VALUES(
    '128557',
    'Anita Jones',
    DB2VIDEO(
        CURRENT SERVER,
        '/employee/videos/ajones.vid',
        :hvStorageType,
        'Anita's video',
        :hvVidattrs,
        :hvThumbnail)
    );
```

Duration

影像	音效	視訊
	X	X

傳回 WAVE 或 AIFF 音效，或視訊的持續時間 (亦即，以秒為單位的播放時間)。

併入檔

音效 dmbaudio.h

視訊 dmbvideo.h

語法

►►—Duration—(—handle—)————►

參數 (資料類型)

handle (DB2AUDIO 或 DB2VIDEO)

含有視訊或音效控點的主變數或直欄名稱。

回覆值 (資料類型)

視訊、WAVE、AIFF 或使用者定義格式音效的持續時間 (INTEGER)，是以秒為單位。以其它格式傳回音效的 NULL 值。

範例

顯示儲存在員工表格的視訊直欄中，所有視訊的持續時間：

```
EXEC SQL BEGIN DECLARE SECTION;
long hvDur_vid;
EXEC SQL END DECLARE SECTION;
```

```
EXEC SQL SELECT DURATION(VIDEO)
      INTO :hvDur_vid
      FROM EMPLOYEE;
```

Filename

Filename

影像	音效	視訊
X	X	X

如果物件內容儲存在某檔案中，傳回含有影像、音效或視訊內容的伺服器檔案名稱（指示從資料庫表格中）。如果如果將影像、音效或視訊在資料庫中儲存為 BLOB，則會傳回 NULL 值。

併入檔

影像 dmbimage.h
音效 dmbaudio.h
視訊 dmbvideo.h

語法

►►Filename(—handle—)—————►◄

參數 (資料類型)

handle (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)
含有影像、音效或視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

如果物件內容位於伺服器檔案中，則傳回此伺服器檔名 (VARCHAR(250))；如果物件被儲存為 BLOB，則傳回 NULL 值。

範例

顯示 Robert Smith 項目在員工表格中的視訊檔名：

```
EXEC SQL BEGIN DECLARE SECTION;  
char hvVid_fname[251];  
EXEC SQL END DECLARE SECTION;  
  
EXEC SQL SELECT FILENAME(VIDEO)  
INTO :hvVid_fname  
FROM EMPLOYEE  
WHERE NAME='Robert Smith';
```

FindInstrument

影像	音效	視訊
	X	

傳回 MIDI 音效中指定樂器第一次出現的磁軌號碼。

併入檔

dmbaudio.h

語法

►—FindInstrument—(—handle—,—instrument—)—————►

參數 (資料類型)

handle (DB2AUDIO)

含有音效控點的主變數或直欄名稱。

instrument (VARCHAR(255))

所要搜尋的樂器名稱。Audio Extender 會尋找名稱完全符合所提供名稱的樂器。

回覆值 (資料類型)

磁軌號碼含有第一個出現的指定樂器名稱 (SMALLINT)；如果找不到樂器的指定名稱，則傳回 -1 值。以其它音效格式傳回 NULL。

範例

尋找員工表格的聲音直欄中所儲存 Robert Smith 的 MIDI 音效記錄，第一個出現的 PIANO：

```
EXEC SQL BEGIN DECLARE SECTION;
      short hvInstr;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FINDINSTRUMENT(SOUND, 'PIANO')
      INTO :hvInstr
      FROM EMPLOYEE
      WHERE NAME = 'Robert Smith';
```

FindTrackName

FindTrackName

影像	音效	視訊
	X	

傳回 MIDI 音效中指定磁軌的號碼。

併入檔

dmbaudio.h

語法

►►FindTrackName(—handle—,—trackname—)————►

參數 (資料類型)

handle (DB2AUDIO)

含有音效控點的主變數或直欄名稱。

trackname (VARCHAR(255))

所要搜尋的磁軌名稱。Audio Extender 會尋找名稱完全符合所提供名稱的磁軌。

回覆值 (資料類型)

指定樂器名稱的已命名磁軌名稱 (SMALLINT)。如果找不到指定名稱的磁軌，則傳回 -1 值。以其它格式傳回 NULL 值。

範例

查看 Robert Smith 的 MIDI 錄音內，是否有名為 WELCOME 的磁軌：音效記錄儲存於員工表格的聲音直欄中：

```
EXEC SQL BEGIN DECLARE SECTION;
    short hvTrack;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FINDTRACKNAME(SOUND,
    'WELCOME')
    INTO :hvTrack
    FROM EMPLOYEE
    WHERE NAME = 'Robert Smith';
```

Format

影像	音效	視訊
X	X	X

傳回影像、音效或視訊的格式。

併入檔

影像	dmbimage.h
音效	dmbaudio.h
視訊	dmbvideo.h

語法

►►Format(—handle—)————►

參數 (資料類型)

handle (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)

含有影像、音效或視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

影像、音效或視訊的格式 (VARCHAR(8))。請參閱第 105 頁的表 8，來取得支援的影像、音效和視訊格式。

範例

取得儲存在員工表格的圖片直欄中格式為 GIF 的影像，其所有員工的名稱：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvName[30];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT NAME
      INTO :hvName
      FROM EMPLOYEE
      WHERE FORMAT(PICTURE)='GIF';
```

FrameRate

FrameRate

影像	音效	視訊
		X

傳回視訊產量 (以每秒框數為單位)。

併入檔
dmbvideo.h

語法
►►FrameRate(—handle—)————►◄

參數 (資料類型)

handle (DB2VIDEO)
含有視訊控點的主變數或直欄名稱。

回覆值 (資料類型)
視訊的框掃描率 (SMALLINT)。如果產量掃描率是變數，則傳回 NULL 值。

範例
取得員工表格的視訊直欄中，儲存在 Anita Jones 的視訊框掃描率：

```
EXEC SQL BEGIN DECLARE SECTION;
short hvFm_rate;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FRAMERATE (VIDEO)
FROM EMPLOYEE
INTO :hvFm_rate
WHERE NAME='Anita Jones';
```


GetInstruments

影像	音效	視訊
	X	

傳回 MIDI 音效中所有樂器的樂器名稱。

併入檔

dmbaudio.h

語法

►►GetInstruments—(—handle—)—————►►

參數 (資料類型)

handle (DB2AUDIO)

含有音效控點的主變數或直欄名稱。

回覆值 (資料類型)

MIDI 音效中所有樂器的樂器名稱 (VARCHAR(1536))。這些值會以磁軌號碼順序傳回 (例如，PIANO、TRUMPET、BASS)。結果會分成 *n* 個欄位，其中 *n* 是 MIDI 音效中磁軌的號碼。如果磁軌沒有相關的樂器，則其欄位為空白。以 MIDI 以外的音效格式傳回 NULL 值。

範例

尋找 Robert Smith 的 MIDI 錄音中的所有樂器 (亦即，磁軌號碼和樂器名稱)。音效記錄儲存於員工表格的聲音直欄中：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvAud_Instr[1536];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT GETINSTRUMENTS(SOUND)
INTO :hvAud_Instr
FROM EMPLOYEE
WHERE NAME = 'Robert Smith';
```

GetTrackNames

影像	音效	視訊
	X	

傳回 MIDI 音效中所有磁軌的名稱。

併入檔
dmbaudio.h

語法
►►GetTrackNames—(—handle—)————►◄

參數 (資料類型)

handle (DB2AUDIO)
含有音效控點的主變數或直欄名稱。

回覆值 (資料類型)
MIDI 音效中所有磁軌的名稱 (VARCHAR(1536))。這些值會以磁軌號碼順序傳回 (例如，PIANO TUNE、TRUMPET FANFARE)。結果會分成 *n* 個欄位，其中 *n* 是 MIDI 音效中磁軌的號碼。如果磁軌沒有名稱，則它的欄位是空白。以 MIDI 以外的音效格式傳回 NULL 值。

範例
取得員工表格的聲音直欄中所儲存 Robert Smith 的 MIDI 音效記錄內，所有的磁軌號碼和磁軌名稱：

```
EXEC SQL BEGIN DECLARE SECTION;  
  char hvTracks[1536];  
EXEC SQL END DECLARE SECTION;  
  
EXEC SQL SELECT GETTRACKNAMES(SOUND)  
  INTO :hvTracks  
 FROM EMPLOYEE  
 WHERE NAME = 'Robert Smith';
```

Height

影像	音效	視訊
X		X

傳回影像或視訊框的高度（以圖點為單位）。

併入檔

影像 dmbimage.h

視訊 dmbvideo.h

語法

►►—Height—(—*handle*—)—————►◄

參數 (資料類型)

handle (DB2IMAGE 或 DB2VIDEO)

含有影像或視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

高度的單位是圖點 (INTEGER)

範例

取得員工表格的圖片直欄中小於 500 圖點的所有影像之檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
      char hvImg_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
      INTO :hvImg_fname
      FROM EMPLOYEE
      WHERE HEIGHT(PICTURE)<500;
```

Importer

影像	音效	視訊
X	X	X

傳回將影像、音效或視訊儲存在資料庫表格中的使用者之使用者 ID。

併入檔

影像 dmbimage.h
音效 dmbaudio.h
視訊 dmbvideo.h

語法

►► Importer (—handle—) ◀◀

參數 (資料類型)

handle (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)
含有影像、音效或視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

importer 的使用者 ID (CHAR(8))

範例

以使用者 ID rsmith，取得員工表格的聲音直欄中所儲存音效的所有檔名：

```
EXEC SQL BEGIN DECLARE SECTION;  
  char hvAud_fname[251];  
EXEC SQL END DECLARE SECTION;  
  
EXEC SQL SELECT FILENAME(SOUND)  
  INTO :hvAud_fname  
  FROM EMPLOYEE  
  WHERE IMPORTER(SOUND)='rsmith';
```

ImportTime

影像	音效	視訊
X	X	X

傳回影像、音效或視訊儲存在資料庫表格中時指示的時間戳記。

併入檔

影像 dmbimage.h

音效 dmbaudio.h

視訊 dmbvideo.h

語法

►►—ImportTime—(—handle—)—————►◄

參數 (資料類型)

handle (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)

含有影像、音效或視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

儲存影像、音效或視訊的時間戳記 (TIMESTAMP)

範例

取得員工表格的圖片直欄中一年以上的影像之所有檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
  char hvImg_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
  INTO :hvImg_fname
  FROM EMPLOYEE
 WHERE (CURRENT TIMESTAMP -
        IMPORTTIME(PICTURE))>365;
```

MaxBytesPerSec

影像	音效	視訊
		X

傳回最大視訊產量 (以每秒位元組為單位)。

併入檔

dmbvideo.h

語法

►►MaxBytesPerSec(—*handle*—)————►◄

參數 (資料類型)

handle (DB2VIDEO)

含有視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

視訊產量 (INTEGER)。如果產量掃描率是變數，則傳回 NULL 值。

範例

取得員工表格的視訊直欄中，儲存在 Anita Jones 的最大視訊產量：

```
EXEC SQL BEGIN DECLARE SECTION;
long hvMax_BytesPS;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT MAXBYTESPERSEC(VIDEO)
      INTO :hvMax_BytesPS
      FROM EMPLOYEE
      WHERE NAME='Anita Jones';
```

NumAudioTracks

影像	音效	視訊
	X	X

傳回視訊或 MIDI 音效中音效磁軌的數目。

併入檔

音效 dmbaudio.h

視訊 dmbvideo.h

語法

►►—NumAudioTracks—(—handle—)—————►►

參數 (資料類型)

handle (DB2AUDIO 或 DB2VIDEO)

含有視訊或音效控點的主變數或直欄名稱。

回覆值 (資料類型)

視訊或 MIDI 音效中音效磁軌數目 (SMALLINT)。以其它格式傳回音效的 NULL 值。

範例

取得員工表格的視訊直欄中不含任何音效磁軌的任何視訊檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvVid_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(VIDEO)
INTO :hvVid_fname
FROM EMPLOYEE
WHERE NUMAUDIOTRACKS(VIDEO) = 0;
```

NumChannels

影像	音效	視訊
	X	X

傳回 WAVE 或 AIFF 音效，或視訊中記錄的音效通道數目。

併入檔

音效 dmbaudio.h

視訊 dmbvideo.h

語法

►►NumChannels—(—handle—)————►◄

參數 (資料類型)

handle (DB2AUDIO 或 DB2VIDEO)

含有視訊或音效控點的主變數或直欄名稱。

回覆值 (資料類型)

視訊、WAVE 或 AIFF 音效中記錄的音效通道數目 (SMALLINT)。以其它格式傳回音效的 NULL 值。

範例

取得員工表格的聲音直欄中以立體音 (亦即 2 個通道) 記錄的所有音效檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvAud_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(SOUND)
INTO :hvAud_fname
FROM EMPLOYEE
WHERE NUMCHANNELS(SOUND) = 2;
```


NumColors

影像	音效	視訊
X		

傳回影像中顏色的數目。

併入檔

dmbimage.h

語法

►►—NumColors—(—*handle*—)—————►◄

參數 (資料類型)

handle (DB2IMAGE)

含有影像控點的主變數或直欄名稱。

回覆值 (資料類型)

影像中的顏色數目 (INTEGER)

範例

取得員工表格的圖片直欄中影像少於 16 色的影像名：

```
EXEC SQL BEGIN DECLARE SECTION;
      char hvImg_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
      INTO :hvImg_fname
      FROM EMPLOYEE
      WHERE NUMCOLORS(PICTURE) < 16;
```

NumFrames

NumFrames

影像	音效	視訊
		X

傳回視訊中的框數。

併入檔

dmbvideo.h

語法

►►NumFrames—(—*handle*—)————►◄

參數 (資料類型)

handle (DB2VIDEO)

含有視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

視訊中的框數 (INTEGER)。如果產量掃描率是變數，則傳回 NULL 值。

範例

取得員工表格的視訊直欄中，儲存在 Robert Smith 的視訊中的框數：

```
EXEC SQL BEGIN DECLARE SECTION;
long hvNum_Frames;
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT NUMFRAMES (VIDEO)
      INTO :hvNum_Frames
      FROM EMPLOYEE
      WHERE NAME='Robert Smith';
```

NumVideoTracks

影像	音效	視訊
		X

傳回視訊中視訊磁軌的數目。

併入檔

dmbvideo.h

語法

►►—NumVideoTracks—(—handle—)—————►►

參數 (資料類型)

handle (DB2VIDEO)

含有視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

視訊磁軌數目 (SMALLINT)

範例

取得員工表格的視訊直欄中具有一個以上視訊磁軌的所有視訊之檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvVid_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (VIDEO)
INTO :hvVid_fname
FROM EMPLOYEE
WHERE NUMVIDEOTRACKS(VIDEO) > 1;
```

QbScoreFromName

影像	音效	視訊
X		

傳回影像的分數，它是一個表示如何讓影像特性符合查詢物件特性的數字。與屬於影像控點的直欄相關的 QBIC 型錄，可用來計算影像的分數。分數越低，表示影像特性越符合指定查詢物件的特性。（QbScoreFromName 會取代 QbScore，但 QbScore 仍會被接受。）

註:

- 1. 僅適用於 **EEE**：已分割資料庫環境中不支援 QbScoreFromName。在使用 QbQueryGetString API 查詢字串後，請改用 QbScoreFromStr UDF。
- 2. 在未來版本中，非分割資料庫環境也不支援 QbScoreFromName。若要重覆使用查詢，您應該使用 QbQueryGetString API 以取得查詢字串，並儲存該字串供稍後應用程式之使用。

併入檔
無

語法
►►QbScoreFromName(—imgHandle—,—queryName—)◄◄

語法
不支援的版本
►►QbScoreFromName(—queryName—,—imgHandle—)◄◄

參數 (資料類型)

imgHandle (DB2Image)
影像的控點。

queryName (varchar(18))
查詢物件的名稱。

回覆值 (資料類型)
影像的分數 (DOUBLE)。分數的範圍可從 0.0 到無限大。分數越低，表示與符合查詢中所指定特性值的目標影像特性值越接近。分數 0.0 表示完全相符。分數 NULL 值表示影像尚未被編目；此舊版 UDF 在影像尚未被編目時傳回分數 -1。

範例
尋找表格直欄中平均色非常接近紅色的已編目影像：

```
EXEC SQL BEGIN DECLARE SECTION;
char Img_fnd[100];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT NAME
      INTO :Img_fnd
FROM FABRIC
WHERE (QBSCOREFROMNAME(SWATCH_IMG,
      'fshavgcol'))<0.1;
```

QbScoreFromStr

影像	音效	視訊
X		

傳回影像的分數，它是一個表示如何讓影像特性符合查詢字串特性的數字。與屬於影像控點的直欄相關的 QBIC 型錄，可用來計算影像的分數。分數越低，表示影像特性越符合查詢的特性。

併入檔
無

語法
►►QbScoreFromStr(—imgHandle—,—query—)————►◄

語法
不支援的版本
►►QbScoreFromStr(—query—,—imgHandle—)————►◄

參數 (資料類型)

imgHandle (DB2Image)
影像的控點。

query (VARCHAR(1024))
查詢字串。

回覆值 (資料類型)
影像的分數 (DOUBLE)。分數的範圍可從 0.0 到無限大。 分數越低，表示與符合查詢中所指定特性值的目標影像特性值越接近。 分數 0.0 表示完全相符。分數 NULL 值表示影像尚未被編目；此舊版 UDF 在影像尚未被編目時傳回分數 -1。

範例
尋找表格直欄中平均色非常接近紅色的已編目影像：

```
SELECT name
  FROM fabric
 WHERE (QbScoreFromStr(Swatch_Img,
    'QbColorFeatureClass color=<255, 0, 0>'))<0.1
```

QbScoreTBFromName

影像	音效	視訊
X		

傳回影像直欄的分數表格。每一個分數都是表示如何讓符合查詢物件特性的影像特性越接近的一個數字。與屬於影像控點的指定表格和直欄相關的 QBIC 型錄，可用來計算影像的分數。任何影像的分數越低，則表示與符合查詢物件特性的影像特性越接近。

註:

- 1. 僅適用於 **EEE**：已分割資料庫環境中不支援 QbScoreTBFromName。在使用 QbQueryGetString API 查詢字串後，請改用 QbScoreFromStr UDF。
- 2. 在未來版本中，非分割資料庫環境也不支援 QbScoreTBFromName。若要重覆使用查詢，您應該使用 QbQueryGetString API 以取得查詢字串，並儲存該字串供稍後應用程式之使用。

併入檔
無

語法
傳回直欄中所有已編目影像的分數
▶▶—QbScoreTBFromName—(—queryName—,—table—,—column—)————▶▶

語法
傳回直欄中指定已編目影像數目的分數
▶▶—QbScoreTBFromName—(—queryName—,—table—,—column—,—maxReturns—)————▶▶

參數 (資料類型)

queryName (VARCHAR(18))
查詢物件的名稱。

table (CHAR(18))
內含影像直欄的完整表格名稱。如果表格綱目與啓動 DB2 Extender 服務所用的使用者 ID 相同時，您可以使用不完整的表格名稱。

column (CHAR(18))
影像直欄的名稱。

maxReturns (INTEGER)
結果表格傳回的最大控點數目。如果未指定一個值，則傳回的最大控點數目是 100。

QbScoreTBFromName

回覆值 (資料類型)

直欄中影像的影像控點和分數表格。結果表格有兩個直欄： IMAGE_ID (DB2Image) 含有影像控點，及 SCORE (DOUBLE) 含有分數。結果表格會依分數的升序次序來排列。 分數的範圍可從 0.0 到無限大。 分數越低，表示與符合查詢中所指定特性值的目標影像特性值越接近。 分數 0.0 表示完全相符。分數 -1 表示影像尚未被編目。

範例

將表格直欄中影像的紋理與查詢物件中指定的紋理作一比較；傳回影像控點及它們的分數：

```
SELECT name, description
INTO :hvName, :hvDesc
FROM fabric
WHERE CAST (swatch_img as varchar(250)) IN
  (SELECT CAST (image_id as varchar(250)) FROM TABLE
   (QbScoreTBFromName
    'fstxtr',
    'clothes.fabric',
    'swatch_img'))
AS T1));
```


QbScoreTBFromStr

影像	音效	視訊
X		

傳回影像直欄中分數的表格。每一個分數都是一個數字，它表示如何讓影像特性與查詢字串中指定的特性更接近。與屬於影像控點的表格和直欄相關的 QBIC 型錄，可用來計算影像的分數。影像的分數越低，則表示與符合查詢特性的影像特性越接近。

併入檔
無

語法
傳回直欄中所有已編目影像的分數
▶▶QbScoreTBFromStr(—query—,—table—,—column—)▶▶

語法
傳回直欄中指定已編目影像數目的分數
▶▶QbScoreTBFromStr(—query—,—table—,—column—,—maxReturns—)▶▶

參數 (資料類型)

query (VARCHAR(1024))
查詢字串。

table (CHAR(18))
內含影像直欄的完整表格名稱。如果表格綱目與啓動 DB2 Extender 服務所用的使用者 ID 相同時，您可以使用不完整的表格名稱。

column (CHAR(18))
要查詢的影像直欄。

maxReturns (INTEGER)
結果表格傳回的最大控點數目。如果未指定一個值，則傳回的最大影像控點數目是 100。

回覆值 (資料類型)
直欄中影像的影像控點和分數表格。結果表格有兩個直欄： IMAGE_ID (DB2Image) 含有影像控點，及 SCORE (DOUBLE) 含有分數。結果表格會依分數的升序次序來排列。分數的範圍可從 0.0 到無限大。分數越低，表示與符合查詢中所指定特性值的目標影像特性值越接近。分數 0.0 表示完全相符。分數 -1 表示影像尚未被編目。

QbScoreTBFromStr

範例

尋找表格直欄中紋理最接近伺服器檔案中影像紋理的 10 個已編目影像：

```
SELECT name, description
FROM fabric
WHERE CAST (swatch_img as varchar(250)) IN
      (SELECT CAST (image_id as varchar(250)) FROM TABLE
        (QbScoreTBFromStr
         (QbTextureFeatureClass file=<server,"patterns/ptrn07.gif">'
          'clothes.fabric',
          'swatch_img',
          10))
        AS T1));
```

Replace

影像	音效	視訊
X	X	X

更新資料庫中儲存的影像、音效或視訊的內容，並更新它的註解。

併入檔

影像 dmbimage.h

音效 dmbaudio.h

視訊 dmbvideo.h

語法

更新緩衝區或從屬站檔案的內容，並且更新註解

```
►►—Replace—(—handle—,—content—,—source_format—,—target_file—,—comment—)——►◄
```

語法

更新伺服器檔案的內容，並更新註解

```
►►—Replace—(—handle—,—source_file—,—source_format—,—stortype—,——————►
►—comment—)——►◄
```

併入檔

更新緩衝區或從屬站檔案中具有使用者提供屬性的內容，並更新註解

```
►►—Replace—(—handle—,—content—,—target_file—,——————►
►—comment—,—attrs—,—thumbnail—)——►◄
```

併入檔

更新伺服器檔案中具有使用者提供屬性的內容，並更新註解

```
►►—Replace—(—handle—,—source_file—,—stortype—,—comment—,——————►
►—attrs—,—thumbnail—)——►◄
```

語法

更新緩衝區或從屬站檔案的內容，並做格式轉換及更新註解 -- 僅適用於影像

```
►►—Replace—(—handle—,—content—,—source_format—,——————►
►—target_format—,—target_file—,—comment—)——►◄
```

Replace

語法

使用格式轉換更新伺服器檔案內容及更新註解 -- 僅適用於影像

```
►►Replace—(—handle—,—source_file—,—source_format—,——————►
                                     ►
►target_format—,—target_file—,—comment—)——————►◄
```

語法

更新緩衝區或從屬站檔案的內容，並做格式轉換及其他變更，同時更新註解 -- 僅適用於影像

```
►►Replace—(—handle—,—content—,—source_format—,——————►
                                     ►
►target_format—,—target_file—,—conversion_options—,—comment—)——————►◄
```

語法

更新伺服器檔案內容，並做格式轉換及其他變更，同時更新註解 -- 僅適用於影像

```
►►Replace—(—handle—,—source_file—,—source_format—,——————►
                                     ►
►target_format—,—conversion_options—,—target_file—,—comment—)——————►◄
```

參數 (資料類型)

handle (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)

含有影像、音效或視訊控點的主變數或直欄名稱。

source_file (LONG VARCHAR)

含有影像、音效或視訊更新內容的檔案名稱。

target_file (LONG VARCHAR)

含有要更新的影像、音效或視訊內容的檔案名稱。

create_target (INTEGER)

一種指示是否在來源內容位於伺服器檔案中時建立目標檔案。此值可以是 0 或 1。0 值表示目標檔案不會被建立 (實際上，擷取並不會發生)。1 值表示目標檔案會被建立 (如果目標檔案已存在，則此值會改寫檔案)。如果來源內容是 BLOB，則會建立目標檔案 (如果檔案已存在，則會被改寫)。

target_format (VARCHAR(8))

擷取後的影像格式。來源影像格式將被轉換成合適的格式。如果內容已使用格式轉換更新過，則必須在 DB2IMAGEPATH 和 DB2MMPATH 環境變數中指定目標檔案路徑。若是 MPG1 格式，您可以指定 MPG1、mpg1、MPEG1 或 mpeg1。若是 MPG2 格式，您可以指定 MPG2、mpg2、MPEG2 或 mpeg2。

content (BLOB(2G) AS LOCATOR)

含有影像、音效或視訊更新內容的主變數。主變數可以是類型 BLOB、BLOB_FILE 或 BLOB_LOCATOR。DB2 會將資料類型提升為 BLOB_LOCATOR，並將 LOB 定位器傳遞到 Replace UDF。

source_format (VARCHAR(8))

更新影像、音效或視訊的來源格式。您可以指定 NULL 值或空白字串，或僅對影像指定字串 ASIS；在這三種情形下，Extender 會嘗試自動決定格式。若是 MPG1 格式，您可以指定 MPG1、mpg1、MPEG1 或 mpeg1。若是 MPG2 格式，您可以指定 MPG2、mpg2、MPEG2 或 mpeg2。

comment (LONG VARCHAR)

註解。

attrs (LONG VARCHAR FOR BIT DATA)

影像、音效或視訊的屬性

thumbnail (LONG VARCHAR FOR BIT DATA)

影像或視訊框的縮圖 (僅適用於影像和視訊)

conversion_options (VARCHAR(100))

指定更新影像時所要引用的一些變更，例如旋轉和壓縮。有關支援的轉換選項，請參閱第 106 頁的表 9。

回覆值 (資料類型)

要更新的影像、音效或視訊的控點 (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)。

範例

更新員工表格的圖片直欄中 Anita Jones 的影像、將影像格式從 BMP 轉換成 GIF 及更新註解：

```
EXEC SQL BEGIN DECLARE SECTION;
    long hvStorageType;
EXEC SQL END DECLARE SECTION;

hvStorageType = MMDB_STORAGE_TYPE_INTERNAL;

EXEC SQL UPDATE EMPLOYEE
    SET PICTURE = REPLACE(PICTURE,
        '/employee/newimg/ajones.bmp',
        'BMP',
        'GIF',
        :hvStorageType,
        'Anita''s new picture')
    WHERE NAME='Anita Jones';
```

SamplingRate

影像	音效	視訊
	X	X

傳回 WAVE 或 AIFF 音效，或視訊中音效磁軌的取樣率（以每秒範例數目為單位）。

併入檔

音效 dmbaudio.h

視訊 dmbvideo.h

語法

►►SamplingRate(—handle—)◄◄

參數 (資料類型)

handle (DB2AUDIO 或 DB2VIDEO)

含有視訊或音效控點的主變數或直欄名稱。

回覆值 (資料類型)

視訊、WAVE 或 AIFF 音效的取樣率 (INTEGER)。以其它格式傳回音效的 NULL 值。

範例

取得員工表格的聲音直欄中取樣率為 44.1 KHz 的所有音效之檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvAud_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME (SOUND)
INTO :hvAud_fname
FROM EMPLOYEE
WHERE SAMPLINGRATE(SOUND) = 44100;
```

Size

影像	音效	視訊
X	X	X

傳回影像、音效或視訊的大小 (以位元組為單位)。

併入檔

影像	dmbimage.h
音效	dmbaudio.h
視訊	dmbvideo.h

語法

►►Size(—handle—)—————►

參數 (資料類型)

handle (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)

含有影像、音效或視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

以位元組為單位，影像、音效或視訊的大小 (INTEGER)。

範例

取得員工表格的圖片直欄中大大於 310 KB 的所有影像檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
INTO :hvImg_fname
FROM EMPLOYEE
WHERE SIZE(PICTURE) > 310000;
```

Thumbnail

影像	音效	視訊
X		X

傳回或更新資料庫中儲存的視訊框或影像縮圖大小版本。

併入檔

影像 dmbimage.h
視訊 dmbvideo.h

語法

擷取縮圖

►►Thumbnail—(—handle—)————►◄

語法

更新縮圖

►►Thumbnail—(—handle—,—new_thumbnail—)————►◄

參數 (資料類型)

handle (DB2IMAGE 或 DB2VIDEO)

含有影像或視訊控點的主變數或直欄名稱。

new_thumbnail (LONG VARCHAR FOR BIT DATA)

更新縮圖的來源內容

回覆值 (資料類型)

以擷取而言，是指被擷取的縮圖內容 (LONG VARCHAR FOR BIT DATA)；以更新而言，是指影像或視訊的控點 (DB2IMAGE 或 DB2VIDEO)。

範例

取得員工表格中，儲存在 Anita Jones 的影像縮圖：

```
EXEC SQL BEGIN DECLARE SECTION;  
struct{  
    short len;  
    char data [32000];  
}hvThumbnail;  
EXEC SQL END DECLARE SECTION;  
  
EXEC SQL SELECT THUMBNAIL(PICTURE)  
INTO :hvThumbnail  
FROM EMPLOYEE  
WHERE NAME = 'Anita Jones';
```


更新員工表格中，與 Anita Jones 的視訊相關的縮圖：

```
EXEC SQL BEGIN DECLARE SECTION;
    struct {
        short len;
        char data[10000];
    }hvThumbnail;
EXEC SQL END DECLARE SECTION;

/* Create thumbnail and */
/* store in hvThumbnail */

EXEC SQL UPDATE EMPLOYEE
    SET VIDEO=THUMBNAIL(
        VIDEO,
        :hvThumbnail)
    WHERE NAME='Anita Jones';
```

TicksPerQNote

影像	音效	視訊
	X	

傳回記錄的 MIDI 音效之計時速度 (以每四分之一音符刻點數為單位)。

併入檔

dmbaudio.h

語法

►—TicksPerQNote—(*—handle—*)——►◄

參數 (資料類型)

handle (DB2AUDIO)

含有音效控點的主變數或直欄名稱。

回覆值 (資料類型)

MIDI 音效每四分之一音符的計時刻點數目 (SMALLINT)。以其它格式傳回音效的 NULL 值。

範例

取得員工表格的聲音直欄中，以每四分之一音符超過 200 計時刻點的速度記錄，所有 MIDI 音效的檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvAud_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(SOUND)
INTO :hvAud_fname
FROM EMPLOYEE
WHERE FORMAT(SOUND)='MIDI'
AND TICKSPERQNOTE(SOUND)>200;
```

TicksPerSec

影像	音效	視訊
	X	

傳回記錄的 MIDI 音效之計時速度 (以每秒刻點數為單位)。

併入檔

dmbaudio.h

語法

►—TicksPerSec—(—*handle*—)—————►◄

參數 (資料類型)

handle (DB2AUDIO)

含有音效控點的主變數或直欄名稱。

回覆值 (資料類型)

MIDI 音效每秒的計時刻點數目 (SMALLINT)。以其它格式傳回音效的 NULL 值。

範例

取得員工表格的聲音直欄中，以每秒低於 50 計時刻點的速度記錄，所有 MIDI 音效的檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvAud_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(SOUND)
INTO :hvAud_fname
FROM EMPLOYEE
WHERE FORMAT(SOUND)='MIDI'
AND TICKSPERSEC(SOUND)<50;
```

Updater

影像	音效	視訊
X	X	X

傳回最後更新資料庫表格中影像、音效或視訊的使用者之使用者 ID。

併入檔

影像	dmbimage.h
音效	dmbaudio.h
視訊	dmbvideo.h

語法

►►—Updater—(—handle—)—————►◄

參數 (資料類型)

handle (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)
含有影像、音效或視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

最後更新影像、音效或視訊者的使用者 ID (CHAR(8))

範例

取得儲存在員工表格的 Robert Smith 視訊直欄中，最後更新視訊者的使用者 ID。

```
EXEC SQL BEGIN DECLARE SECTION;
char hvUpdater[30];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT UPDATER(VIDEO)
      INTO :hvUpdater
      FROM EMPLOYEE
      WHERE NAME='rsmith';
```

UpdateTime

影像	音效	視訊
X	X	X

傳回最後更新資料庫表格中影像、音效或視訊時指示的時間戳記。

併入檔

影像 dmbimage.h

音效 dmbaudio.h

視訊 dmbvideo.h

語法

►►—UpdateTime—(—handle—)—————►◄

參數 (資料類型)

handle (DB2IMAGE、DB2AUDIO 或 DB2VIDEO)

含有影像、音效或視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

最後更新影像、音效或視訊時的時間戳記 (TIMESTAMP)

範例

取得員工表格的圖片直欄中最後 2 天更新的影像之檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
INTO :hvImg_fname
FROM EMPLOYEE
WHERE(CURRENT TIMESTAMP -
      UPDATETIME(PICTURE))< 2;
```

Width

Width

影像	音效	視訊
X		X

傳回影像或視訊框的寬度 (以圖點為單位)。

併入檔

影像 dmbimage.h

視訊 dmbvideo.h

語法

►►Width(—handle—)————►◄

參數 (資料類型)

handle (DB2IMAGE 或 DB2VIDEO)

含有影像或視訊控點的主變數或直欄名稱。

回覆值 (資料類型)

寬度的單位是圖點 (INTEGER)

範例

取得員工表格的圖片直欄中寬度小於 300 圖點的所有影像之檔名：

```
EXEC SQL BEGIN DECLARE SECTION;
char hvImg_fname[251];
EXEC SQL END DECLARE SECTION;

EXEC SQL SELECT FILENAME(PICTURE)
      INTO :hvImg_fname
      FROM EMPLOYEE
      WHERE WIDTH(PICTURE)<300;
```

第 14 章 應用程式設計介面

本章是提供 DB2 Extender 管理 API 的參考資訊。這些 API 是按字母順序列出。

在每一項 API 中會顯示如下資訊：

- 提供 API 的 Extender
- 簡要說明
- 使用該 API 所需具備的權限
- API 的檔案庫名稱
- API 的併入 (標頭) 檔
- API 呼叫的 C 語法
- API 參數的說明
- API 所傳回的值
- 使用範例

DBaAdminGetInaccessibleFiles

影像	音效	視訊
	X	

傳回在使用者表格的音效直欄中所參照到的無法存取的檔案名稱。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放檔案清單資料結構，以及檔案清單中每個項目中的檔名欄位。

授權

SYSADM、SYSCTRL、SYSMAINT

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaAdminGetInaccessibleFiles(  
    char *qualifier,  
    long *count,  
    FILEREF *(*fileList)  
);
```

參數

- qualifier (in)**
有效的使用者 ID 或 NULL 值。若您指定的是使用者 ID，則會搜尋所有採用所指限定元的表格；若指定 NULL 值，則會搜尋現行資料庫中的所有表格。
- count (out)**
輸出清單中的登錄數目。

fileList (out)

表格中所參照到的無法存取的檔案清單。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

SQL_ERROR 或其它 SQL 回覆碼

DB2 傳回錯誤。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

MMDB_RC_NO_AUTH

使用者沒有呼叫此 API 的適當權限。

範例

列出使用者 ID rsmith 擁有的表格音效直欄中所參照到的無法存取的檔案：

```
#include <dmbaudio.h>
long idx;

rc = DBaAdminGetInaccessibleFiles("rsmith",
    &count, &filelist);
```

DBaAdminGetReferencedFiles

影像	音效	視訊
	X	

傳回在使用者表格的音效直欄中所參照到的檔名。若某檔案無法存取 (例如，無法使用環境變數規格來分辨檔名)，則檔名前面會有一個星號。由於此 API 不會用到 **FILEREF** 資料結構的 **FILENAME** 欄位，因此請將此值設為 **NULL**。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放 **filelist** 資料結構。

授權

SYSADM、SYSCTRL、SYSMAINT

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaAdminGetReferencedFiles(
    char *qualifier,
    long *count,
    FILEREF *(*fileList)
);
```

參數

qualifier (in)
有效的使用者 ID 或 **NULL** 值。若您指定的是使用者 ID，則會搜尋所有採用所指限定元的表格；若指定 **NULL** 值，則會搜尋現行資料庫中的所有表格。

count (out)

輸出清單中的登錄數目。

fileList (out)

表格中所參照的檔案清單。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

MMDB_RC_NO_AUTH

使用者沒有呼叫此 API 的適當權限。

範例

列出 ajones 擁有的表格音效直欄中所參照到的檔案：

```
#include <dmbaudio.h>
long idx;

rc = DBaAdminGetReferencedFiles("ajones",
    &count, &fileList);
```

DBaAdminIsFileReferenced

影像	音效	視訊
	X	

傳回使用者表格中參照到指定檔案的音效直欄項目清單。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放檔案清單資料結構，以及檔案清單中每個項目中的檔名欄位。

授權

SYSADM、SYSCTRL、SYSMAINT

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaAdminIsFileReferenced(  
    char *qualifier,  
    char *fileName,  
    long *count,  
    FILEREF *(*tableList)  
);
```

參數

- qualifier (in)**
有效的使用者 ID 或 NULL 值。若您指定的是使用者 ID，則會搜尋所有採用所指限定元的表格；若指定 NULL 值，則會搜尋現行資料庫中的所有表格。
- fileName (in)**
被參照到的檔案名稱。

count (out)

輸出清單中的登錄數目。

tableList (out)

參照指定檔案的表格項目清單。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

MMDB_RC_NO_AUTH

使用者在呼叫此 API 上並無適當權限。

範例

列出現行資料庫的所有表格中，參照到檔案 /audios/asmith.wav 的音效直欄項目：

```
#include <dmbaudio.h>
long idx;

rc = DBaAdminIsFileReferenced(NULL,
    "/audios/asmith.wav",
    &count, &tableList);
```

DBaAdminReorgMetadata

影像	音效	視訊
	X	

『清除』和音效有關的 meta 資料表格，例如：

- 回收音效 meta 資料表格中不再使用的空間。
- 刪除音效 meta 資料表格中會參照不復存在之音效檔的項目。

呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

SYSADM、SYSCTRL、SYSMAINT

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaAdminReorgMetadata(  
    char *qualifier  
);
```

參數

qualifier (in)
有效的使用者 ID 或 NULL 值。若您指定的是使用者 ID，則會清除所有採用所指限定元的表格；若您指定的是 NULL 值，則會清除現行資料庫中的所有表格。

錯誤碼

MMDB_SUCCESS
API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_NO_AUTH

使用者在呼叫此 API 上並無適當權限。

範例

清除使用者 ID rsmith 擁有的表格中，音效直欄的 meta 資料表格：

```
#include <dmbaudio.h>
```

```
rc = DBaAdminReorgMetadata("rsmith");
```

DBaDisableColumn

影像	音效	視訊
	X	

爲音效 (DB2Audio 資料) 停用某直欄，使該直欄無法保有音效資料。直欄項目的內容會設爲 NULL，且會捨棄和此直欄有關的 meta 資料。此外，音效 Extender 定義給此直欄的所有觸發函式，亦會遭到捨棄。新橫列仍可插入到內含停用直欄的表格中，且新橫列中仍可包含以 DB2Audio 類型定義的資料，但是，新橫列將無相關的 meta 資料 (在管理支援表格中)。應用程式必須先連接資料庫之後，才能呼叫此 API。建議您呼叫此 API 之後，發出 SQL COMMIT 陳述式。

授權

控制、改變、SYSADM、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaDisableColumn(  
    char *tableName,  
    char *colName,  
);
```

參數

- tableName (in)**
內含音效直欄的表格名稱。
- colName (in)**
音效直欄的名稱。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

為音效 (DB2Audio 資料) 停用員工表格中的聲音直欄：

```
#include <dmbaudio.h>
```

```
rc = DBaDisableColumn("employee", "sound");
```

DBaDisableDatabase

影像	音效	視訊
	X	

爲音效 (DB2Audio 資料) 停用某資料庫，使該資料庫無法保有音效資料。並且也停用爲 DB2Audio 所定義的資料庫中的所有表格。而 Audio Extender 定義給資料庫的 meta 資料與 UDF 亦會遭到捨棄。新橫列仍可插入到以 DB2Audio 類型定義的資料庫的表格中，但新橫列將無相關的 meta 資料 (在管理支援表格中)。建議呼叫此 API 之後，再發出 SQL COMMIT 陳述式。

授權

DBADM、SYSADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaDisableDatabase(  
    );
```

參數

DBaDisableDatabase 無相關參數。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_RC_NO_AUTH**
呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

爲音效 (DB2Audio 資料) 停用現行資料庫：

```
#include <dmbaudio.h>
```

```
rc = DBaDisableDatabase();
```

DBaDisableTable

影像	音效	視訊
	X	

停用音效 (DB2Audio 資料) 的某個表格，使該表格無法保有音效資料。也會停用為 DB2Audio 所定義的表格中的所有直欄。針對表格以 Audio Extender 定義的部份 meta 資料也會被捨棄。新橫列仍可插入到以 DB2Audio 類型定義的表格中，但新橫列將無相關的 meta 資料 (在管理支援表格中)。應用程式必須先連接資料庫之後，才能呼叫此 API。建議您呼叫此 API 之後，發出 SQL COMMIT 陳述式。

授權

控制、改變、SYSADM、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaDisableTable(  
    char *tableName  
);
```

參數

tableName (in)
 內含某音效直欄的表格名稱。

錯誤碼

MMDB_SUCCESS
 API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

停用音效 (DB2Audio 資料) 的員工表格：

```
#include <dmbaudio.h>
```

```
rc = DBaDisableTable("employee");
```

DBaEnableColumn

影像	音效	視訊
	X	

啓用音效 (DB2Audio 資料) 的某個直欄。 API 會定義與管理此直欄和 meta 資料表格間的關係。呼叫此 API 之前，應用程式必須先連接到資料庫。建議呼叫此 API 之後，再發出 SQL COMMIT 陳述式。

授權

控制、改變、SYSADM、DBADM

在 API 參數中指定的表格空間和緩衝池也需要使用專用權。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaEnableColumn(  
    char *tableName,  
    char *colName,  
);
```

參數

tableName (in)
 內含音效直欄的表格名稱。

colName (in)
 音效直欄的名稱。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_WARN_ALREADY_ENABLED

直欄已啟用。

MMDB_RC_WRONG_SIGNATURE

為指定直欄指定的資料類型不正確。只接受使用者定義的資料類型 MMDBSYS.DB2AUDIO。

MMDB_RC_COLUMN_DOESNOT_EXIST

指定表格中未定義直欄。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_NOT_ENABLED

未啟用資料庫或表格。

範例

啟用音效 (DB2Audio 資料) 的員工表格中的聲音直欄：

```
#include <dmbaudio.h>
```

```
rc = DBaEnableColumn("employee", "sound");
```

DBaEnableDatabase

影像	音效	視訊
	X	

啓用音效 (DB2Audio 資料) 的某資料庫。會爲每個資料庫各呼叫一次 API。其會定義 DB2 使用者定義的類型 (DB2Audio) 給資料庫管理程式。同時，它也建立會操作 DB2Audio 資料的所有 UDF。建議呼叫此 API 之後，再發出 SQL COMMIT 陳述式。

授權

DBADM、SYSADM、SYSCTRL

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaEnableDatabase(  
    char *tableSpace  
);
```

參數

tableSpace (in)

可謂配置區之集成 (管理表格的儲存所在) 的表格空間名稱。表格空間規格具有下列三個部份：*datats*、*indexts*、*longts*，其中 *datats* 爲其中建有 meta 資料表格的表格空間；*indexts* 爲其中建有 meta 資料表格上之索引的表格空間；*longts* 爲 meta 資料表格 (像是內含 LONG VARCHAR 與 LOB 資料類型者) 中長直欄之值儲存所在的表格空間。 假設表格空間規格中有任一部份的值是設爲 NULL，則會爲該部份採用預設的表格空間。

僅適用於 EEE： 在您爲 Extender 啓用某資料庫時，您指定的表格空間，也應定義在節點群組 (內含已分割資料庫系統中的所有節點) 中。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_WARN_ALREADY_ENABLED

資料庫已啟用。

MMDB_RC_API_NOT_SUPPORTED_FOR_SERVER

所連接的伺服器不支援此命令。

MMDB_WARN_NOT_ALL_NODES

指定的表格空間中不含有 Extender 的所有相關節點。(僅適用於 EEE)

MMDB_RC_NOT_SAME_NODEGROUP

指定的表格空間未位於同一節點群組中。(僅適用於 EEE)

範例

為表格空間 MYTS 中的音效 (DB2Audio 資料)，啟用現行的資料庫。在索引與長表格空間方面，則採用預設值：

```
#include <dmbaudio.h>
```

```
rc = DBaEnableDatabase("myts,,");
```

為音效 (DB2Audio 資料庫) 啟用現行的資料庫；採用預設的表格空間：

```
#include <dmbaudio.h>
```

```
rc = DBaEnableDatabase(NULL);
```

DBaEnableTable

影像	音效	視訊
	X	

啓用音效 (DB2Audio 資料) 的某個表格。會爲每個表格各呼叫一次 API。它會建立 meta 資料表格，以儲存並管理表格中音效直欄的相關屬性。爲了避免發生鎖定之虞，在呼叫此 API 之前，應用程式應先確定異動。呼叫此 API之前，應用程式必須先連接資料庫。建議您呼叫此 API 之後，發出 SQL COMMIT 陳述式。

授權

控制、改變、SYSADM、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaEnableTable(  
    char *tableSpace,  
    char *tableName  
);
```

參數

tableSpace (in)
表格空間的名稱，它是儲存管理表格的配置區集合。表格空間規格具有下列三個部份：*datats*、*indexts*、*longts*，其中 *datats* 爲其中建有 meta 資料表格的表格空間；*indexts* 爲其中建有 meta 資料表格上之索引的表格空間；*longts* 爲 meta 資料表格 (像是內含 LONG VARCHAR 與 LOB 資料類型者) 中長直欄之值儲存所在的表格空間。 若提供表格空間規格的任何部份一個 NULL 值，則會使用該部份的預設表格空間。

假設表格空間規格中有任一部份的值是設為 NULL，則會為該部份採用預設的表格空間。

僅適用於 EEE：指定的表格空間必須位於與使用者表格相同的節點群組。

tableName (in)

將包含音效直欄的表格名稱。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_WARN_ALREADY_ENABLED

表格已啟用。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_TABLE_DOESNOT_EXIST

表格不存在。

MMDB_RC_TABLESPACE_NOT_SAME_NODEGROUP

您指定的表格空間，和使用者表格所在的節點群組不同。 **(僅適用於 EEE)**

範例

為表格空間 MYTS 中的音效 (DB2Audio 資料)，啟用員工表格。使用索引和長表格空間的預設值：

```
#include <dmbaudio.h>
```

```
rc = DBaEnableTable("myts,,",
    "employee");
```

為音效 (DB2Audio 資料) 啟用員工表格。使用預設表格空間：

```
#include <dmbaudio.h>
```

```
rc = DBaEnableTable(NULL,
    "employee");
```

DBaGetError

影像	音效	視訊
	X	

傳回有關前次錯誤的說明。請在其它任何 API 傳回錯誤碼之後，再呼叫 API。

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaGetError(  
    SQLINTEGER *sqlcode,  
    char *errorMsgText  
);
```

參數

sqlcode (out)
同屬 SQL 錯誤碼。

errorMsgText (out)
SQL 錯誤訊息文字。

錯誤碼

MMDB_SUCCESS
API 呼叫程序執行成功。

範例

取得前次的錯誤，而此錯誤的 SQL 錯誤碼儲存在 `errCode` 中，訊息文字儲存在 `errMsg` 中：

```
#include <dmbaudio.h>
```

```
rc = DBaGetError(&errCode, &errMsg);
```

DBaGetInaccessibleFiles

影像	音效	視訊
	X	

傳回在使用者表格的音效直欄中所參照到的無法存取的檔案名稱。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放檔案清單資料結構，以及檔案清單中每個項目中的檔名欄位。

授權

在所有已搜尋的使用者表格和相關管理支援表格中，已啓用音效直欄的 SELECT 專用權。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaGetInaccessibleFiles(  
    char *tableName,  
    long *count,  
    FILEREF *(*fileList)  
);
```

參數

- tableName (in)**
完整、不完整、或 NULL 的表格名稱。 假設您指定的是表格名稱，則會搜尋該表格，以找出會參照無法存取型檔案的項目。若您指定的是 NULL 值，則會搜尋所有採用所指限定元的表格。
- count (out)**
輸出清單中的登錄數目。

fileList (out)

表格中所參照到的無法存取的檔案清單。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

呼叫此 API 之前，應用程式與資料庫之間無有效連接。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出在員工表格的音效直欄中所參照到的無法存取的檔案：

```
long idx;  
#include <dmbaudio.h>  
  
rc = DBaGetInaccessibleFiles("employee",  
    &count, &filelist);
```

DBaGetReferencedFiles

影像	音效	視訊
	X	

傳回在使用者表格的音效直欄中所參照到的檔名。若某檔案無法存取 (例如，無法使用環境變數規格來分辨檔名)，則檔名前面會有一個星號。由於此 API 不會用到 **FILEREF** 資料結構的 **FILENAME** 欄位，因此請將此值設為 **NULL**。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放 **filelist** 資料結構。

授權

在所有已搜尋的使用者表格和相關管理支援表格中，已啟用音效直欄的 **SELECT** 專用權。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
<code>dmbaudio.lib</code>	<code>libdmbaudio.a</code> (AIX) <code>libdmbaudio.sl</code> (HP-UX) <code>libdmbaudio.so</code> (Solaris)

併入檔

`dmbaudio.h`

語法

```
long DBaGetReferencedFiles(  
    char *tableName,  
    long *count,  
    FILEREF *(*fileList)  
);
```

參數

tableName (in)
完整、不完整、或 **NULL** 的表格名稱。 假設您指定的是表格名稱，則會搜尋該表格，以找出會參照檔案的項目。若您指定的是 **NULL** 值，則會搜尋現行使用者 **ID** 資料庫所擁有的所有表格。

count (out)

輸出清單中的登錄數目。

fileList (out)

表格中所參照的檔案清單。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出在員工表格的音效直欄中所有參照到的檔案：

```
#include <dmbaudio.h>
long idx;

rc = DBaGetReferencedFiles("employee",
    &count, &filelist);
```

DBalsColumnEnabled

影像	音效	視訊
	X	

判斷是否已為音效 (DB2Audio 資料) 啓用某直欄。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

使用者表格上的 SYSADM、DBADM、表格擁有者或 SELECT 專用權

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaIsColumnEnabled(
    char *tableName,
    char *colName,
    short *status
);
```

參數

- tableName (in)**
完整或不完整的表格名稱。
- colName (in)**
直欄的名稱。
- status (out)**
指出是否已啓用直欄。此參數會傳回一個數值。Extender 也會傳回一個指出其狀態的常數。 其值與常數如下：

1 MMDB_IS_ENABLED

0	MMDB_IS_NOT_ENABLED
-1	MMDB_INVALID_DATATYPE

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

判斷是否已為音效啟用員工表格中的聲音直欄：

```
#include <dmbaudio.h>
```

```
rc = DBaIsColumnEnabled("employee",  
    "sound", &status);
```

DBalsDatabaseEnabled

影像	音效	視訊
	X	

判斷是否已為音效 (DB2Audio 資料) 啓用某資料庫。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBalsDatabaseEnabled(  
short *status  
);
```

參數

- status (out)**
指出是否已啓用資料庫。此參數會傳回一個數值。Extender 也會傳回一個指出其狀態的常數。 其值與常數如下：
- 1 MMDB_IS_ENABLED
 - 0 MMDB_IS_NOT_ENABLED

錯誤碼

MMDB_SUCCESS
API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

判斷是否已為音效啟用 personnel 資料庫：

```
#include <dmbaudio.h>
```

```
rc = DBIsDatabaseEnabled(&status);
```

DBalsFileReferenced

影像	音效	視訊
	X	

傳回參照特定檔案的表格項目清單。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放檔案清單資料結構，以及檔案清單中每個項目中的檔名欄位。

授權

在所有已搜尋的使用者表格和相關管理支援表格中，已啓用音效直欄的 SELECT 專用權。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBalsFileReferenced(
    char *tableName,
    char *fileName,
    long *count,
    FILEREF *(*tableList)
);
```

參數

tableName (in)
完整、不完整、或 NULL 的表格名稱。 假設您指定的是表格名稱，則會搜尋該表格，以找出會參照所指檔案的項目。若指定 NULL 值，則會搜尋現行使用者 ID 擁有的所有表格。

fileName (in)

被參照的檔案名稱。

count (out)

輸出清單中的登錄數目。

tableList (out)

參照指定檔案的表格項目清單。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出參照檔案 `/audios/ajones.wav` 的員工表格音效直欄中的項目：

```
#include <dmbaudio.h>
long idx;

rc = DBaIsFileReferenced(NULL,
    "/audios/ajones.wav",
    &count, &tableList);
```

DBalsTableEnabled

影像	音效	視訊
	X	

判斷是否已為音效 (DB2Audio 資料) 啓用某表格。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBalsTableEnabled(  
    char *tableName,  
    short *status  
);
```

參數

tableName (in)
表格名稱。

status (out)
指出是否已啓用表格。此參數會傳回一個數值。Extender 也會傳回一個指出其狀態的常數。 其值與常數如下：

- 1 MMDB_IS_ENABLED
- 0 MMDB_IS_NOT_ENABLED
- 1 MMDB_INVALID_DATATYPE

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

判斷是否已為音效 (DB2Audio 資料) 啓用員工表格：

```
#include <dmbaudio.h>
```

```
rc = DBaIsTableEnabled("employee", &status);
```

DBaPlay

影像	音效	視訊
	X	

開啓從屬站上的音效播放程式，並播放一段音效片段。此片段可儲存在音效直欄或外部檔中：

- 若音效片段是儲存在外部檔中，則您可以將檔名或音效控點傳遞給此 API。API 會利用從屬站環境變數 DB2AUDIOPATH，分辨出檔案的位置。必須可由從屬站工作站存取此檔案。
- 假設音效片段是儲存在直欄中，則您必須將音效控點傳遞給 API。應用程式必須連接資料庫，而且具有儲存音效片段的表格的讀取權。

若此音效儲存在某直欄，則 Extender 會建立暫用檔，並從該直欄將物件內容複製到檔案。若音效儲存在外部檔，而且無法使用環境變數中的值分辨它的相對檔名，或無法在從屬站機器上存取此檔案，Extender 也可能會建立暫用檔。暫用檔會建立在使用 DB2AUDIOTEMP 環境變數所指定的目錄中。然後 Extender 會從暫用檔播放音效。

授權

假設您所播放的是直欄中的音效片段，請選取使用者表格上的權限。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

播放儲存於直欄中的一段音效

```
long DBaPlay(
    char *playerName,
    MMDB_PLAY_HANDLE,
    DB2Audio *audioHandle,
    waitFlag
);
```

語法

播放儲存成檔案的一段音效

```
long DBaPlay(
    char *playerName,
    MMDB_PLAY_FILE,
    char *fileName,
    waitFlag
);
```

參數

playerName (in)

音效播放程式的名稱。假設您設為 NULL，則會採用 DB2AUDIOPLAYER 環境變數所指定的預設音效播放程式。

MMDB_PLAY_HANDLE (in)

此常數指出將音效儲存成 BLOB。

MMDB_PLAY_FILE (in)

此常數指出將音效儲存成可自從屬站存取的檔案。

audioHandle (in)

音效的控點。在您播放直欄中的音效片段時，必須傳遞此參數。假設音效控點所呈現的是外部檔，則會以從屬站環境變數 DB2VIDEOPATH 來分辨出檔案的位置。

fileName (in)

內含音效的檔案名稱。

waitFlag (in)

此常數是指出您的程式在繼續執行前，得花多久時間等待使用者關閉播放程式。指定 MMDB_PLAY_WAIT，則會在和您的應用程式一樣的緒中執行播放程式；指定 MMDB_PLAY_NO_WAIT，則是在個別緒中執行播放程式。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

DBaPlay

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

播放由 `audioHandle` 識別的音效。在和應用程式一樣的緒中執行預設播放程式：

```
#include <dmbaudio.h>
```

```
rc = DBaPlay(NULL, MMDB_PLAY_HANDLE,  
             audioHandle, MMDB_PLAY_WAIT);
```

DBaPrepareAttrs

影像	音效	視訊
	X	

準備使用者所提供的音效屬性。當儲存或更新帶有使用者提供之屬性的音效物件時，即會使用此 API。在伺服器上執行的 UDF 碼會預期資料為『big endian』格式，而此格式可為大部份 UNIX 平台所使用。若使用『little endian』格式儲存或更新音效物件 (亦即，在非 UNIX 從屬站中)，則必須在使用 DBaPrepare API 之後，才能提出儲存或更新要求。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudio.lib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
void DBaPrepareAttrs(  
    MMDBAudioAttrs *audAttr  
);
```

參數

audAttr (in)
音效的屬性 (使用者所提供)。

範例

```
準備使用者提供的音效屬性：  
  
#include <dmbaudio.h>  
  
DBaPrepareAttrs(&imgattr);
```

DBaReorgMetadata

影像	音效	視訊
	X	

『清除』和音效有關的 meta 資料表格，例如：

- 回收音效 meta 資料表格中不再使用的空間。
- 刪除音效 meta 資料表格中會參照不復存在之音效檔的項目。

呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

改變、控制、SYSADM、SYSCTRL、SYSMAINT、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbaudiolib	libdmbaudio.a (AIX) libdmbaudio.sl (HP-UX) libdmbaudio.so (Solaris)

併入檔

dmbaudio.h

語法

```
long DBaReorgMetadata(  
    char *tableName  
);
```

參數

tableName (in)
完整、不完整、或 NULL 的表格名稱。若有指定表格名稱，會清除與指定的使用者表格相關的音效 meta 資料表格。若有指定 NULL 值，會清除現行使用者 ID 擁有的表格中的音效直欄的 meta 資料表格。

錯誤碼

MMDB_SUCCESS
API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

為員工表格中的音效直欄，清除其相關的 meta 資料表格：

```
#include <dmbaudio.h>
```

```
rc = DBaReorgMetadata("employee");
```

DBiAdminGetInaccessibleFiles

影像	音效	視訊
X		

傳回在使用者表格的影像直欄中所參照到的無法存取的檔案名稱。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放檔案清單資料結構，以及檔案清單中每個項目中的檔名欄位。

授權

SYSADM、SYSCTRL、SYSMAINT

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiAdminGetInaccessibleFiles(  
    char *qualifier,  
    long *count,  
    FILEREF *(*fileList)  
);
```

參數

- qualifier (in)**
有效的使用者 ID 或 NULL 值。若您指定的是使用者 ID，則會搜尋所有採用所指限定元的表格；若您指定的是 NULL 值，則會搜尋現行資料庫中的所有表格。
- count (out)**
輸出清單中的登錄數目。

fileList (out)

表格中所參照到的無法存取的檔案清單。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_NO_AUTH

使用者沒有呼叫此 API 的適當權限。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出使用者 ID rjones 擁有的表格影像直欄中所參照到的無法存取的檔案：

```
#include <dmbimage.h>
long idx;

rc = DBiAdminGetInaccessibleFiles
    ("rjones", &count, &filelist);
```

DBiAdminGetReferencedFiles

影像	音效	視訊
X		

傳回使用者表格的影像直欄中所參照到的檔名。若某檔案無法存取 (例如，無法使用環境變數規格來分辨檔名)，則檔名前面會有一個星號。由於此 API 不會用到 **FILEREF** 資料結構的 **FILENAME** 欄位，因此請將此值設為 **NULL**。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放 **filelist** 資料結構。

授權

SYSADM、SYSCTRL、SYSMAINT

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiAdminGetReferencedFiles(
    char *qualifier,
    long *count,
    FILEREF *(*fileList)
);
```

參數

qualifier (in)

有效的使用者 ID 或 **NULL** 值。若您指定的是使用者 ID，則會搜尋所有採用所指限定元的表格；若指定 **NULL** 值，則會搜尋現行資料庫中的所有表格。

count (out)

輸出清單中的登錄數目。

fileList (out)

表格中所參照的檔案清單。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_NO_AUTH

使用者沒有呼叫此 API 的適當權限。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出 ajones 擁有的表格影像直欄中所參照到的所有檔案：

```
#include <dmbimage.h>
long idx;

rc = DBiAdminGetReferencedFiles("ajones",
    &count, &filelist);
```

DBiAdminIsFileReferenced

影像	音效	視訊
X		

傳回參照指定檔案的使用者表格中的影像直欄項目清單。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放檔案清單資料結構，以及檔案清單中每個項目中的檔名欄位。

授權

SYSADM、SYSCTRL、SYSMAINT

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiAdminIsFileReferenced(
    char *qualifier,
    char *fileName,
    long *count,
    FILEREF *(*tableList)
);
```

參數

- qualifier (in)**
有效的使用者 ID 或 NULL 值。若您指定的是使用者 ID，則會搜尋所有採用所指限定元的表格；若指定 NULL 值，則會搜尋現行資料庫中的所有表格。
- fileName (in)**
被參照的檔案名稱。

count (out)

輸出清單中的登錄數目。

tableList (out)

參照指定檔案的表格項目清單。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_NO_AUTH

使用者在呼叫此 API 上並無適當權限。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出參照檔案 /images/asmith.bmp 的現行資料庫中，所有表格的影像直欄項目：

```
#include <dmbimage.h>
long idx;

rc = DBiAdminIsFileReferenced(NULL,
    "/images/asmith.bmp",
    &count, &tableList);
```

DBiAdminReorgMetadata

影像	音效	視訊
X		

『清除』和影像有關的 meta 資料表格：

- 回收影像 meta 資料表格中不再使用的空間。
- 刪除影像 meta 資料表格中會參照不復存在之影像檔的項目。

呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

SYSADM、SYSCTRL、SYSMAINT

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiAdminReorgMetadata(  
    char *qualifier  
);
```

參數

qualifier (in)
有效的使用者 ID 或 NULL 值。若您指定的是使用者 ID，則會清除所有採用所指限定元的表格；若您指定的是 NULL 值，則會清除現行資料庫中的所有表格。

錯誤碼

MMDB_SUCCESS
API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_NO_AUTH

使用者在呼叫此 API 上並無適當權限。

範例

清除使用者 ID rsmith 擁有的表格的影像直欄的 meta 資料表格：

```
#include <dmimage.h>
```

```
rc = DBiAdminReorgMetadata("rsmith");
```

DBiBrowse

影像	音效	視訊
X		

開啓從屬站上的影像瀏覽器，並顯示影像。此影像可儲存在影像直欄或外部檔中：

- 若影像是儲存在外部檔中，則您可以將檔名或影像控點傳遞給此 API。API 會利用從屬站環境變數 DB2IMAGEPATH，分辨出檔案的位置。您必須自從屬站工作站來存取該檔案。
- 假設影像是儲存在直欄中，則您必須將影像控點傳遞給 API。應用程式必須先連接到資料庫，而且必須有儲存影像的表格的讀取權。

若瀏覽器無法直接存取影像，則 Extender 會在使用 DB2IMAGETEMP 環境變數指定的目錄中建立暫用檔。然後，該 Extender 會由暫用檔中顯示該影像。

授權

假設您所瀏覽的是直欄中的影像，請選取使用者表格的權限。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

瀏覽儲存於直欄中的影像

```
long DBiBrowse(  
    char *browserName,  
    MMDB_PLAY_HANDLE,  
    DB2Image *imageHandle,  
    waitFlag  
);
```


語法

瀏覽儲存成檔案的影像

```
long DBiBrowse(
    char *browserName,
    MMDB_PLAY_FILE,
    char *fileName,
    waitFlag
);
```

參數

browserName (in)

影像瀏覽器的名稱。假設您設為 NULL，則會採用 DB2IMAGEBROWSER 環境變數所指定的預設影像瀏覽器。

MMDB_PLAY_HANDLE (in)

此常數指出將影像儲存成 BLOB。

MMDB_PLAY_FILE (in)

此常數指出將影像儲存成可自從屬站存取的檔案。

imageHandle (in)

影像的控點。在您瀏覽直欄中的影像時，必須傳遞此參數。假設影像控點所呈現的是外部檔，則會以從屬站環境變數 DB2IMAGEPATH 來分辨出檔案的位置。

fileName (in)

內含影像的檔案名稱。

waitFlag (in)

此常數是指出您的程式在繼續執行前，得花多久時間等待使用者關閉瀏覽器。指定 MMDB_PLAY_WAIT，則會在和您的應用程式一樣的緒中執行瀏覽器；指定 MMDB_PLAY_NO_WAIT，則是在個別緒中執行瀏覽器。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

DBiBrowse

範例

顯示 imageHandle 所識別的影像。在和應用程式相同的緒中執行預設瀏覽器：

```
#include <dmbimage.h>
```

```
rc = DBiBrowse(NULL, MMDB_PLAY_HANDLE,  
               imageHandle, MMDB_PLAY_WAIT);
```

DBiDisableColumn

影像	音效	視訊
X		

爲影像 (DB2Image 資料) 停用某直欄，使該直欄無法保有影像資料。直欄項目的內容會設爲 NULL，且會捨棄和此直欄有關的 meta 資料。與此直欄相關的 QBIC 型錄也會被刪除。此外，影像 Extender 定義給此直欄的所有觸發函式，亦會遭到捨棄。新橫列仍可插入到內含停用直欄的表格中，且新橫列中仍可包含以 DB2Image 類型定義的資料，但是，新橫列將無相關的 meta 資料 (在管理支援表格中)。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

控制、改變、SYSADM、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiDisableColumn(  
    char *tableName,  
    char *colName,  
    );
```

參數

- tableName (in)**
內含影像直欄的表格名稱。
- colName (in)**
影像直欄的名稱。

DBiDisableColumn

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

為影像 (DB2Image 資料) 停用員工表格中的圖片直欄：

```
#include <dmbimage.h>

rc = DBiDisableColumn("employee",
    "picture");
```

DBiDisableDatabase

影像	音效	視訊
X		

爲影像 (DB2Image 資料) 停用某資料庫，使該資料庫無法保有影像資料。也會停用爲 DB2Image 所定義的資料庫中的所有表格。會捨棄資料庫的 Image Extender 定義的 meta 資料和 UDF。新橫列仍可插入到以 DB2Image 類型定義的資料庫的表格中，但新橫列將無相關的 meta 資料 (在管理支援表格中)。

授權

DBADM、SYSADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiDisableDatabase(  
    );
```

參數

DBiDisableDatabase 無相關參數。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_RC_NO_AUTH**
呼叫程式不具備適當的存取權限。
- MMDB_RC_NOT_CONNECTED**
應用程式與資料庫之間無有效連接。

DBiDisableDatabase

範例

爲影像 (DB2Image 資料) 停用現行的資料庫：

```
#include <dmbimage.h>

rc = DBiDisableDatabase();
```

DBiDisableTable

影像	音效	視訊
X		

停用影像 (DB2Image 資料) 的某個表格，使該表格無法保有影像資料。也會停用為 DB2Image 所定義的表格的所有直欄。會捨棄表格的 Image Extender 定義的部份 meta 資料。也會刪除與表格的影像直欄相關的 QBIC 型錄。新橫列仍可插入到以 DB2Image 類型定義的表格中，但新橫列將無相關的 meta 資料 (在管理支援表格中)。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

控制、改變、SYSADM、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiDisableTable(  
    char *tableName  
);
```

參數

tableName (in)
 內含某影像直欄的表格名稱。

錯誤碼

MMDB_SUCCESS
 API 呼叫程序執行成功。

DBiDisableTable

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

為影像 (DB2Image 資料) 停用員工表格：

```
#include <dmbimage.h>
```

```
rc = DBiDisableTable("employee");
```


DBiEnableColumn

影像	音效	視訊
X		

為影像 (DB2Image 資料) 啟用某直欄。 API 會定義與管理此直欄和 meta 資料表格間的關係。呼叫此 API 之前，應用程式必須先連接到資料庫並確認使用者表格。

授權

控制、改變、SYSADM、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiEnableColumn(  
    char *tableName,  
    char *colName,  
    );
```

參數

tableName (in)
 內含影像直欄的表格名稱。

colName (in)
 影像直欄的名稱。

錯誤碼

MMDB_SUCCESS
 API 呼叫程序執行成功。

DBiEnableColumn

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_WARN_ALREADY_ENABLED

直欄已啟用。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_WRONG_SIGNATURE

為指定直欄指定的資料類型不正確。期望的使用者定義類型為 MMDBSYS.DB2IMAGE。

MMDB_RC_COLUMN_DOESNOT_EXIST

指定表格中未定義直欄。

MMDB_RC_NOT_ENABLED

未啟用資料庫或表格。

範例

為影像啟用員工表格中的圖片直欄：

```
#include <dmbimage.h>
```

```
rc = DBiEnableColumn("employee",  
    "picture");
```

DBiEnableDatabase

影像	音效	視訊
X		

爲影像 (DB2Image 資料) 啓用某資料庫。會爲每個資料庫各呼叫一次 API。其會定義 DB2 使用者定義的類型 (DB2Image) 給資料庫管理程式。同時會建立操作 DB2Image 資料的所有 UDF。

授權

DBADM、SYSADM、SYSCTRL

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiEnableDatabase(  
    char *tableSpace  
);
```

參數

tableSpace (in)
可謂配置區之集成 (管理表格的儲存所在) 的表格空間名稱。表格空間規格具有下列三個部份：*datats*、*indexts*、*longts*，其中 *datats* 爲其中建有 meta 資料表格的表格空間；*indexts* 爲其中建有 meta 資料表格上之索引的表格空間；*longts* 爲 meta 資料表格 (像是內含 LONG VARCHAR 與 LOB 資料類型者) 中長直欄之值儲存所在的表格空間。 假設表格空間規格中有任一部份的值是設爲 NULL，則會爲該部份採用預設的表格空間。
僅適用於 EEE： 在您爲 Extender 啓用某資料庫時，您指定的表格空間，也應定義在節點群組 (內含已分割資料庫系統中的所有節點) 中。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_WARN_ALREADY_ENABLED

資料庫已啟用。

MMDB_RC_API_NOT_SUPPORTED_FOR_SERVER

所連接的伺服器不支援此命令。

MMDB_WARN_NOT_ALL_NODES

指定的表格空間中不含有 Extender 的所有相關節點。(僅適用於 EEE)

MMDB_RC_NOT_SAME_NODEGROUP

指定的表格空間未位於同一節點群組中。(僅適用於 EEE)

範例

為表格空間 MYTS 中的影像 (DB2Image 資料) 啟用現行資料庫。在索引與長表格空間方面，則採用預設值：

```
#include <dmbimage.h>
```

```
rc = DBiEnableDatabase("myts,,");
```

為影像 (DB2Image 資料) 啟用現行的資料庫。使用預設表格空間：

```
#include <dmbimage.h>
```

```
rc = DBiEnableDatabase(NULL);
```

DBiEnableTable

影像	音效	視訊
X		

爲影像 (DB2Image 資料) 啓用某表格。會爲每個表格各呼叫一次 API。其會建立 meta 資料表格，以儲存並管理表格中影像直欄的相關屬性。爲了避免發生鎖定之虞，在呼叫此 API 之前，應用程式應先確定異動。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

控制、改變、SYSADM、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiEnableTable(  
    char *tableSpace,  
    char *tableName  
);
```

參數

tableSpace (in)

表格空間的名稱，它是儲存管理表格的配置區集合。表格空間規格具有下列三個部份：*datats*、*indexts*、*longts*，其中 *datats* 爲其中建有 meta 資料表格的表格空間；*indexts* 爲其中建有 meta 資料表格上之索引的表格空間；*longts* 爲 meta 資料表格 (像是內含 LONG VARCHAR 與 LOB 資料類型者) 中長直欄之值儲存所在的表格空間。若提供表格空間規格的任何部份一個 NULL 值，則會使用該部份的預設表格空間。

DBiEnableTable

假設表格空間規格中有任一部份的值是設為 NULL，則會為該部份採用預設的表格空間。

僅適用於 EEE：指定的表格空間必須位於與使用者表格相同的節點群組。

tableName (in)

將包含某影像直欄的表格名稱。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_WARN_ALREADY_ENABLED

表格已啟用。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_TABLE_DOESNOT_EXIST

表格不存在。

MMDB_RC_TABLESPACE_NOT_SAME_NODEGROUP

您指定的表格空間，和使用者表格所在的節點群組不同。 (僅適用於 EEE)

範例

為表格空間 MYTS 中的影像 (DB2Image 資料) 啟用員工表格。使用索引和長表格空間的預設值：

```
#include <dbmimage.h>

rc = DBiEnableTable("myts,,",
    "employee");
```

為影像 (DB2Image 資料) 啟用員工表格。使用預設表格空間：

```
#include <dbmimage.h>

rc = DBiEnableTable(NULL,
    "employee");
```

DBiGetError

影像	音效	視訊
X		

傳回有關前次錯誤的說明。請在其它任何 API 傳回錯誤碼之後，再呼叫 API。

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiGetError(  
    SQLINTEGER *sqlcode,  
    char *errorMsgText  
);
```

參數

sqlcode (out)
同屬 SQL 錯誤碼。

errorMsgText (out)
SQL 錯誤訊息文字。

錯誤碼

MMDB_SUCCESS
API 呼叫程序執行成功。

DBiGetError

範例

取得前次的錯誤，而此錯誤的 SQL 錯誤碼儲存在 `errCode` 中，訊息文字儲存在 `errMsg` 中：

```
#include <dmbimage.h>

rc = DBiGetError(&errCode, &errMsg);
```


DBiGetInaccessibleFiles

影像	音效	視訊
X		

傳回在使用者表格的影像直欄中所參照到的無法存取的檔案名稱。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放檔案清單資料結構，以及檔案清單中每個項目中的檔名欄位。

授權

在所有已搜尋的使用者表格和相關管理支援表格中，已啓用影像直欄的 SELECT 專用權

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiGetInaccessibleFiles(  
    char *tableName,  
    long *count,  
    FILEREF *(*fileList)  
);
```

參數

- tableName (in)**
完整、不完整、或 NULL 的表格名稱。 假設您指定的是表格名稱，則會搜尋該表格，以找出會參照無法存取型檔案的項目。若您指定的是 NULL 值，則會搜尋所有採用所指限定元的表格。
- count (out)**
輸出清單中的登錄數目。

DBiGetInaccessibleFiles

fileList (out)

表格中所參照到的無法存取的檔案清單。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出所有在員工表格的影像直欄中，參照到的無法存取的檔案：

```
#include <dmbimage.h>
long idx;

rc = DBiGetInaccessibleFiles("employee",
    &count, &filelist);
```

DBiGetReferencedFiles

影像	音效	視訊
X		

傳回使用者表格的影像直欄中所參照到的檔名。若某檔案無法存取 (例如，無法使用環境變數規格來分辨檔名)，則檔名前面會有一個星號。由於此 API 不會用到 **FILEREF** 資料結構的 **FILENAME** 欄位，因此請將此值設為 **NULL**。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放 **filelist** 資料結構。

授權

在所有已搜尋的使用者表格和相關管理支援表格中，已啟用影像直欄的 **SELECT** 專用權

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiGetReferencedFiles(  
    char *tableName,  
    long *count,  
    FILEREF *(*fileList)  
);
```

參數

tableName (in)
完整、不完整、或 **NULL** 的表格名稱。 假設您指定的是表格名稱，則會搜尋該表格，以找出會參照檔案的項目。若指定 **NULL** 值，則會搜尋現行使用者 **ID** 擁有的所有表格。

DBiGetReferencedFiles

count (out)

輸出清單中的登錄數目。

fileList (out)

表格中所參照的檔案清單。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出所有在員工表格的影像直欄中所參照到的檔案：

```
#include <dmbimage.h>
long idx;

rc = DBiGetReferencedFiles("employee",
    &count, &filelist);
```

DBIsColumnEnabled

影像	音效	視訊
X		

判斷是否已為影像 (DB2Image 資料)，啓用某直欄。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

使用者表格上的 SYSADM、DBADM、表格擁有者或 SELECT 專用權

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBIsColumnEnabled(  
    char *tableName,  
    char *colName,  
    short *status  
);
```

參數

tableName (in)
完整或不完整的表格名稱。

colName (in)
直欄的名稱。

status (out)
指出是否已啓用直欄。此參數會傳回一個數值。Extender 也會傳回一個指出其狀態的常數。 其值與常數如下：

- 1 MMDB_IS_ENABLED

DBIsColumnEnabled

0	MMDB_IS_NOT_ENABLED
-1	MMDB_INVALID_DATATYPE

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_WARN_ALREADY_ENABLED

直欄已啓用。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

判斷是否已為影像啓用員工表格中的圖片直欄：

```
#include <dmbimage.h>
```

```
rc = DBIsColumnEnabled("employee",  
    "picture", &status);
```

DBIsDatabaseEnabled

影像	音效	視訊
X		

判斷是否已為影像 (DB2Image 資料) 啟用某資料庫。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBIsDatabaseEnabled(  
short *status  
);
```

參數

status (out)
指出是否已啟用資料庫。此參數會傳回一個數值。Extender 也會傳回一個指出其狀態的常數。 其值與常數如下：

1 MMDB_IS_ENABLED
0 MMDB_IS_NOT_ENABLED

錯誤碼

MMDB_SUCCESS
API 呼叫程序執行成功。

DBIsDatabaseEnabled

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

判斷是否已為影像啓用 `personnel` 資料庫：

```
#include <dmbimage.h>
```

```
rc = DBIsDatabaseEnabled(&status);
```


DBiIsFileReferenced

影像	音效	視訊
X		

傳回參照指定檔案的影像直欄中的表格項目清單。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放檔案清單資料結構，以及檔案清單中每個項目中的檔名欄位。

授權

在所有已搜尋的使用者表格和相關管理支援表格中，已啟用影像直欄的 SELECT 專用權

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiIsFileReferenced(
    char *tableName,
    char *fileName,
    long *count,
    FILEREF *(*tableList)
);
```

參數

tableName (in)

完整、不完整、或 NULL 的表格名稱。 假設您指定的是表格名稱，則會搜尋該表格，以找出會參照所指檔案的項目。若指定 NULL 值，則會搜尋現行使用者 ID 擁有的所有表格。

DBiIsFileReferenced

fileName (in)

被參照的檔案名稱。

count (out)

輸出清單中的項目數目。

tableList (out)

參照指定檔案的表格項目清單。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出參照檔案 /images/ajones.bmp 的員工表格影像直欄的項目：

```
#include <dmbimage.h>
long idx;

rc = DBiIsFileReferenced(NULL,
    "/images/ajones.bmp",
    &count, &tableList);
```

DBIsTableEnabled

影像	音效	視訊
X		

判斷是否已為影像 (DB2Image 資料) 啓用某表格。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBIsTableEnabled(  
    char *tableName,  
    short *status  
);
```

參數

tableName (in)

表格名稱。

status (out)

指出是否已啓用表格。此參數會傳回一個數值。Extender 也會傳回一個指出其狀態的常數。其值與常數如下：

- 1 MMDB_IS_ENABLED

DBIsTableEnabled

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

判斷是否已為影像啟用員工表格：

```
#include <dmbimage.h>
```

```
rc = DBIsTableEnabled("employee",  
    &status);
```

DBiPrepareAttrs

影像	音效	視訊
X		

準備使用者提供的影像屬性。當儲存或更新帶有使用者提供之屬性的影像物件時，即會使用此 API。在伺服器上執行的 UDF 碼會預期資料為『big endian』格式，而此格式可為大部份 UNIX 平台所使用。若使用『little endian』格式儲存或更新影像物件（亦即，在非 UNIX 從屬站中），則必須在使用 DBiPrepare API 之後，才能提出儲存或更新要求。

授權

無

檔案庫

OS/2 與 Windows

dmbimage.lib

AIX、HP-UX 及 Solaris

libdmbimage.a (AIX)

libdmbimage.sl (HP-UX)

libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
void DBiPrepareAttrs(
    MMDImageAttrs *imgAttr
);
```

參數

imgAttr (in)

影像的屬性（使用者所提供）。

範例

準備使用者提供的影像屬性：

```
#include <dmbimage.h>
```

```
DBiPrepareAttrs(&imgattr);
```

DBiReorgMetadata

影像	音效	視訊
X		

『清除』和影像有關的 meta 資料表格，例如：

- 回收影像 meta 資料表格中不再使用的空間。
- 刪除影像 meta 資料表格中會參照不復存在之影像檔的項目。

呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

改變、控制、SYSADM、SYSCTRL、SYSMAINT、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbimage.lib	libdmbimage.a (AIX) libdmbimage.sl (HP-UX) libdmbimage.so (Solaris)

併入檔

dmbimage.h

語法

```
long DBiReorgMetadata(  
    char *tableName  
);
```

參數

tableName (in)
完整、不完整、或 NULL 的表格名稱。 假設您指定的是表格名稱，則會為與您指定的使用者表格有關的影像 meta 資料表格，執行清除動作。若有指定 NULL 值，會清除現行使用者 ID 擁有的表格的影像直欄的 meta 資料。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

為員工表格中的影像直欄，清除其相關的 meta 資料表格：

```
#include <dmbimage.h>
```

```
rc = DBiReorgMetadata("employee");
```

DBvAdminGetInaccessibleFiles

影像	音效	視訊
		X

傳回在使用者表格的視訊直欄中所參照到的無法存取的檔案名稱。應用程式必須先連接資料庫，才能呼叫此 API。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放檔案清單資料結構，以及檔案清單中每個項目中的檔名欄位。

授權

SYSADM、SYSCTRL、SYSMAINT

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvAdminGetInaccessibleFiles(  
    char *qualifier,  
    long *count,  
    FILEREF *(*fileList)  
);
```

參數

- qualifier (in)**
有效的使用者 ID 或 NULL 值。若您指定的是使用者 ID，則會搜尋所有採用所指限定元的表格；若指定 NULL 值，則會搜尋現行資料庫中的所有表格。
- count (out)**
輸出清單中的登錄數目。

fileList (out)

表格中所參照到的無法存取的檔案清單。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_NO_AUTH

使用者沒有呼叫此 API 的適當權限。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出使用者 ID rsmith 擁有的表格的視訊直欄中所參照到的無法存取的檔案：

```
#include <dmbvideo.h>
long idx;

rc = DBvAdminGetInaccessibleFiles
    ("rsmith", &count,
    &filelist);
```

DBvAdminGetReferencedFiles

影像	音效	視訊
		X

傳回在使用者表格的視訊直欄中所參照到的檔名。若某檔案無法存取 (例如，無法使用環境變數規格來分辨檔名)，則檔名前面會有一個星號。由於此 API 不會用到 **FILEREF** 資料結構的 **FILENAME** 欄位，因此請將此值設為 **NULL**。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放 **filelist** 資料結構。

授權

SYSADM、SYSCTRL、SYSMAINT

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvAdminGetReferencedFiles(  
    char *qualifier,  
    long *count,  
    FILEREF *(*fileList)  
);
```

參數

qualifier (in)

有效的使用者 ID 或 **NULL** 值。若您指定的是使用者 ID，則會搜尋所有採用所指限定元的表格；若指定 **NULL** 值，則會搜尋現行資料庫中的所有表格。

count (out)

輸出清單中的登錄數目。

fileList (out)

表格中所參照的檔案清單。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_NO_AUTH

使用者沒有呼叫此 API 的適當權限。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出在 ajones 擁有的表格的視訊直欄中所參照到的檔案：

```
#include <dmbvideo.h>
long idx;

rc = DBvAdminGetReferencedFiles
    ("ajones", &count,
    &filelist);
```

DBvAdminIsFileReferenced

影像	音效	視訊
		X

傳回參照指定檔案的使用者表格中的視訊直欄項目清單。呼叫此 API 之前，應用程式必須先連接到資料庫。

呼叫此 API 之後，一定要釋放此 API 配置的資源。更明確的說法是，您必須釋放檔案清單資料結構，以及檔案清單中每個項目中的檔名欄位。

授權

SYSADM、SYSCTRL、SYSMAINT

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvAdminIsFileReferenced(  
    char *qualifier,  
    char *fileName,  
    long *count,  
    FILEREF *(*tableList)  
);
```

參數

- qualifier (in)**
有效的使用者 ID 或 NULL 值。若您指定的是使用者 ID，則會搜尋所有採用所指限定元的表格；若指定 NULL 值，則會搜尋現行資料庫中的所有表格。
- fileName (in)**
被參照的檔案名稱。

count (out)

輸出清單中的登錄數目。

tableList (out)

參照指定檔案的表格項目清單。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_NO_AUTH

使用者在呼叫此 API 上並無適當權限。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出參照檔案 /videos/asmith.mpg 的現行資料庫中，所有表格的視訊直欄項目：

```
#include <dmbvideo.h>
long idx;

rc = DBvAdminIsFileReferenced(NULL,
    "/videos/asmith.mpg",
    &count, &tableList);
```

DBvAdminReorgMetadata

影像	音效	視訊
		X

『清除』和視訊有關的 meta 資料表格，例如：

- 回收視訊 meta 資料表格中不再使用的空間。
- 刪除視訊 meta 資料表格中會參照不復存在之視訊檔的項目。

呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

SYSADM、SYSCTRL、SYSMAINT

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvAdminReorgMetadata(  
    char *qualifier  
);
```

參數

qualifier (in)
有效的使用者 ID 或 NULL 值。若您指定的是使用者 ID，則會清除所有採用所指限定元的表格；若您指定的是 NULL 值，則會清除現行資料庫中的所有表格。

錯誤碼

MMDB_SUCCESS
API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_NO_AUTH

使用者在呼叫此 API 上並無適當權限。

範例

清除使用者 ID rsmith 擁有的表格的視訊直欄的 meta 資料表格：

```
#include <dmbvideo.h>
```

```
rc = DBvAdminReorgMetadata("rsmith");
```

DBvBuildStoryboardFile

影像	音效	視訊
		X

建立一個快照型錄檔，並在檔案內建置視訊中之所有快照的相關項目。來源視訊可放在資料庫或檔案中。就每一個快照而言，API 儲存至少一個代表框的快照號碼、起始框號碼、結束框號碼和資訊。 DBvStoryboardCtrl 資料結構中的值，用以決定要用多少個呈現畫面來認定一個快照。對於其長度低於 DBvStoryboardCtrl 中之臨界值的快照而言， API 會視同一個呈現畫面；對於其長度界於 DBvStoryboardCtrl 中之高低臨界值間的快照而言， API 會視同兩個呈現畫面；就長度大於 DBvStoryboardCtrl 中的上限值的快照而言，API 可識別三個代表框。呈現畫面資訊中會包括其畫面編號、以及內含畫面內容的檔案名稱。此資訊可用來顯示故事板，也就是顯示一則看得見的視訊摘要。

授權

插入、控制

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvBuildStoryboardFile(
    char *fileName,
    DBvIOType *video,
    DBvShotControl *shotCtrl,
    DBvStoryboardCtrl *sbCtrl
);
```

參數

catalogName (in)
指向快照型錄檔名稱的指標。

video (in)

指向視訊結構的指標。

shotCtrl (in)

指向快照控制結構的指標。

sbCtrl (in)

指向故事板控制結構的指標。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_INVALID_CATALOG

型錄無效或不存在。

範例

建立一個名為 `hotshots` 的快照型錄檔，然後以視訊中的所有快照的資料填入該檔案：

```
#include <dmbshot.h>

rc = DBvBuildStoryboardFile("hotshots",
    video, &shotCtrl, &sbCtrl);
```

DBvBuildStoryboardTable

影像	音效	視訊
		X

在快照型錄內建置視訊中之所有快照的相關項目。來源視訊可放在資料庫或檔案中。該快照型錄位於資料庫中。 API 會為每一個快照儲存來源視訊的相關控點或檔案資訊。它也儲存至少一個代表框的快照號碼、起始框號碼、結束框號碼和資訊。 DBvStoryboardCtrl 資料結構中的值，用以決定要用多少個呈現畫面來認定一個快照。對於其長度低於 DBvStoryboardCtrl 中之臨界值的快照而言， API 會視同一個呈現畫面；對於其長度界於 DBvStoryboardCtrl 中之高低臨界值間的快照而言， API 會視同兩個呈現畫面；就長度大於 DBvStoryboardCtrl 中的上限值的快照而言，API 可識別三個代表框。呈現畫面資訊中會包括其畫面編號以及畫面資料。儲存在快照型錄中的呈現畫面資訊，可用來顯示故事板，也就是顯示一則看得見的視訊摘要。

呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

插入、控制

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvBuildStoryboardTable(  
    char *catalogName,  
    DBvIOType *video,  
    DBvShotControl *shotCtrl,  
    DBvStoryboardCtrl *sbCtrl,  
    SQLHDBC hdbc  
);
```

參數

catalogName (in)

指向快照型錄名稱的指標。

video (in)

指向視訊結構的指標。

shotCtrl (in)

指向快照控制結構的指標。

sbCtrl (in)

指向故事板控制結構的指標。

hdbc (in)

取自 SQLConnect 中的資料庫控點。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_INVALID_CATALOG

型錄無效或不存在。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

在快照型錄 hotshots 中，建立視訊的相關項目：

```
#include <dmbshot.h>

rc = DBvBuildStoryboardTable("hotshots",
    video, &shotCtrl, &sbCtrl, hdbc);
```

DBvClose

影像	音效	視訊
		X

關閉一個已開啓的視訊檔，以進行場景變更偵測。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvClose(  
    DB2vIOType *video  
);
```

參數

video (in)
指向視訊結構的指標。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_RC_CANNOT_CLOSE**
未能關閉視訊檔。

範例

關閉先前為視訊場景變更偵測所開啓的視訊檔：

```
#include <dmbshot.h>  
  
rc = DBvClose(video);
```

DBvCreateIndex

影像	音效	視訊
		X

建立儲存在檔案的視訊的索引。 Video Extender 會利用此索引來存取視訊中的快照與畫面。 索引會儲存在和來源視訊檔同一目錄中的某個純文字檔中。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvCreateIndex(  
    char *fileName  
);
```

參數

fileName (in)
指向視訊檔名稱的指標。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_RC_OPEN_VIDEO**
未能關閉視訊檔以進行處理。
- MMDB_RC_INDEX_FAIL**
未能建置索引。

範例

爲 \videos\ajones.mpg 檔中的視訊建置一則索引：

```
#include <dmbshot.h>
```

```
rc = DBvCreateIndex("\\videos\\ajones.mpg");
```

DBvCreateIndexFromVideo

影像	音效	視訊
		X

爲視訊建立一則索引。視訊必須先行開啓才能進行快照偵測。 Video Extender 會利用此索引來存取視訊中的快照與畫面。索引會儲存在純文字檔中。而其檔名會儲存在 DBvIOType 資料結構中。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvCreateIndexFromVideo(  
    DBvIOType *video  
);
```

參數

video (update)
指向視訊結構的指標。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_RC_OPEN_VIDEO**
未能關閉視訊檔以進行處理。

MMDB_RC_INDEX_FAIL

未能建置索引。

範例

爲視訊建立索引：

```
#include <dmbshot.h>
```

```
rc = DBvCreateIndexFromVideo(video);
```

DBvCreateShotCatalog

影像	音效	視訊
		X

建立一份實為一組表格的快照型錄，在這些表格中，會提供快照的相關資訊，像是：畫面編號等。

應用程式必須連接一個同時啓用給 db2video 與 db2image 使用的資料庫。

授權

建立、SYSADM、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvCreateShotCatalog(  
    char *catalogName,  
    SQLHDBC hdbc  
);
```

參數

catalogName (in)
所要建立之快照型錄的名稱。

hdbc (in)
取自 SQLConnect 中的資料庫控點。

錯誤碼

MMDB_SUCCESS
API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

建立一個名為 hotshots 的快照型錄：

```
#include <dmbshot.h>
```

```
rc = DBvCreateShotCatalog("hotshots", hdbc);
```

DBvDeleteShot

影像	音效	視訊
		X

刪除型錄中的某個快照。

授權

插入、控制

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvDeleteShot(  
    char *catalogName,  
    char *shotHandle,  
    SQLHDBC hdbc  
);
```

參數

catalogName (in)
型錄的名稱。

shotHandle (in)
快照控點。

hdbc (in)
取自 SQLConnect 中的資料庫控點。

錯誤碼

MMDB_SUCCESS
API 呼叫程序執行成功。

MMDB_RC_ACCESS

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_INVALID_CATALOG

型錄無效或不存在。

範例

使用快照的控點，刪除 hotshots 型錄中的某個快照：

```
#include <dmbshot.h>
```

```
rc = DBvDeleteShot("hotshots", shot,  
                   hdbc);
```

DBvDeleteShotCatalog

影像	音效	視訊
		X

刪除快照型錄。

授權

控制、SYSADM、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvDeleteShotCatalog(  
    char *catalogName,  
    SQLHDBC hdbc  
);
```

參數

- catalogName (in)**
所要刪除之快照型錄的名稱。
- hdbc (in)**
取自 SQLConnect 中的資料庫控點。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_RC_ACCESS**
呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_INVALID_CATALOG

型錄無效或不存在。

刪除快照型錄 hotshots：

範例

```
#include <dmbshot.h>

rc = DBvDeleteShotCatalog("hotshots",
    hdbc);
```

DBvDetectShot

影像	音效	視訊
		X

在視訊檔中搜尋下一個快照。假設有偵測到快照，則會在所偵測到的快照中，記下畫面編號與第一個畫面的畫面資料。您必須檢查 `shotDetected` 旗號，以判斷有否偵測到快照。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvDetectShot(
    DBvIOType *video,
    unsigned long *start_frame,
    char *shotDetected,
    DBvShotControl *shotCtrl,
    DBvShotType *shot,
    );
```

參數

- video (update)**
指向視訊結構的指標。
- start_frame (in/out)**
做為搜尋起點的畫面編號。在返回時，會以下個快照的搜尋起點處來更新參數。
- shotDetected (out)**
快照的偵測旗號：1= 偵測到畫面，0= 未偵測到任何畫面。

shotCtrl (in)

指向快照控制資料的指標。

shot (out)

指向快照與快照資料的指標。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_EOF

已抵達檔案尾端。

MMDB_NO_INDEX

視訊索引不存在。

範例

在視訊檔中，由畫面 1 開始搜尋下個快照：

```
#include <dmbshot.h>

long start_frame=1;

rc = DBvDetectShot(video, start_frame&Detected,
    &shotCtrl, &shot);
```

DBvDisableColumn

影像	音效	視訊
		X

爲視訊 (DB2Video 資料) 停用某直欄，使該直欄無法保有視訊資料。直欄項目的內容會設爲 NULL，且會捨棄和此直欄有關的 meta 資料。此外，視訊 Extender 定義給此直欄的所有觸發函式，亦會遭到捨棄。新橫列仍可插入到內含停用直欄的表格中，且新橫列中仍可包含以 DB2Video 類型定義的資料，但是，新橫列將無相關的 meta 資料 (在管理支援表格中)。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

控制、改變、SYSADM、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvDisableColumn(  
    char *tableName,  
    char *colName,  
);
```

參數

- tableName (in)**
內含視訊直欄的表格名稱。
- colName (in)**
視訊直欄的名稱。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

為視訊 (DB2Video 資料) 停用員工表格中的 tv_ads 直欄：

```
#include <dmbvideo.h>
```

```
rc = DBvDisableColumn("employee",  
    "tv_ads");
```

DBvDisableDatabase

影像	音效	視訊
		X

爲視訊 (DB2Video 資料) 停用某資料庫，使該資料庫無法保有視訊資料。此外，定義給 DB2Video 的相關資料庫內的所有表格，亦全數停用。而 Video Extender 定義給資料庫的 meta 資料與 UDF 亦會遭到捨棄。新橫列仍可插入到以 DB2Video 類型定義的資料庫的表格中，但新橫列將無相關的 meta 資料 (在管理支援表格中)。

授權

DBADM、SYSADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvDisableDatabase(  
    );
```

參數

DBvDisableDatabase 無相關參數。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_RC_NO_AUTH**
呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

爲視訊 (DB2Video 資料) 停用現行的資料庫：

```
#include <dmbvideo.h>
```

```
rc = DBvDisableDatabase();
```

DBvDisableTable

影像	音效	視訊
		X

停用視訊 (DB2Video 資料) 的某個表格，使該表格無法保有視訊資料。此外，定義給 DB2Video 的相關表格內的所有直欄，亦會全數停用。而 Video Extender 定義給表格的部份 meta 資料亦會遭到捨棄。新橫列仍可插入到以 DB2Video 類型定義的表格中，但新橫列將無相關的 meta 資料 (在管理支援表格中)。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

控制、改變、SYSADM、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvDisableTable(  
    char *tableName  
);
```

參數

tableName (in)
內含某視訊直欄的表格名稱。

錯誤碼

MMDB_SUCCESS
API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

為視訊 (DB2Video 資料) 停用員工表格：

```
#include <dmbvideo.h>
```

```
rc = DBvDisableTable("employee");
```

DBvEnableColumn

影像	音效	視訊
		X

爲視訊 (DB2Video 資料) 啓用某直欄。 API 會定義與管理此直欄和 meta 資料表格間的關係。呼叫此 API 之前，應用程式必須先連接到資料庫並確認使用者表格。

授權

控制、改變、SYSADM、DBADM

使用 API 參數指定的表格空間和緩衝池也需要使用專用權。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvEnableColumn(  
    char *tableName,  
    char *colName,  
);
```

參數

- tableName (in)**
內含視訊直欄的表格名稱。
- colName (in)**
視訊直欄的名稱。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_WARN_ALREADY_ENABLED

直欄已啟用。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_WRONG_SIGNATURE

為指定直欄指定的資料類型不正確。只接受使用者定義的資料類型
MMDBSYS.DB2VIDEO。

MMDB_RC_COLUMN_DOESNOT_EXIST

指定表格中未定義直欄。

MMDB_RC_NOT_ENABLED

未啟用資料庫或表格。

範例

為視訊啟用員工表格中的視訊直欄：

```
#include <dmbvideo.h>
```

```
rc = DBvEnableColumn("employee",  
    "video");
```

DBvEnableDatabase

影像	音效	視訊
		X

爲視訊 (DB2Video 資料) 啓用某資料庫。會爲每個資料庫各呼叫一次 API。其會定義 DB2 使用者定義的類型 (DB2Video) 給資料庫管理程式。同時會建立操作 DB2Video 資料的所有 UDF。

授權

DBADM、SYSADM、SYSCTRL

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvEnableDatabase(  
    char *tableSpace  
);
```

參數

tableSpace (in)
可謂配置區之集成 (管理表格的儲存所在) 的表格空間名稱。表格空間規格具有下列三個部份：*datats*、*indexts*、*longts*，其中 *datats* 爲其中建有 meta 資料表格的表格空間；*indexts* 爲其中建有 meta 資料表格上之索引的表格空間；*longts* 爲 meta 資料表格 (像是內含 LONG VARCHAR 與 LOB 資料類型者) 中長直欄之值儲存所在的表格空間。 假設表格空間規格中有任一部份的值是設爲 NULL，則會爲該部份採用預設的表格空間。
僅適用於 EEE： 在您爲 Extender 啓用某資料庫時，您指定的表格空間，也應定義在節點群組 (內含已分割資料庫系統中的所有節點) 中。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_WARN_ALREADY_ENABLED

資料庫已啟用。

MMDB_RC_API_NOT_SUPPORTED_FOR_SERVER

所連接的伺服器不支援此命令。

MMDB_WARN_NOT_ALL_NODES

指定的表格空間中不含有 Extender 的所有相關節點。(僅適用於 EEE)

MMDB_RC_NOT_SAME_NODEGROUP

指定的表格空間未位於同一節點群組中。(僅適用於 EEE)

範例

爲表格空間 MYTS 中的視訊 (DB2Video 資料)，啟用現行的資料庫。在索引與長表格空間方面，則採用預設值：

```
#include <dmbvideo.h>
```

```
rc = DBvEnableDatabase("myts,,");
```

爲視訊 (DB2Video 資料) 啟用現行的資料庫。使用預設表格空間：

```
#include <dmbvideo.h>
```

```
rc = DBvEnableDatabase(NULL);
```

DBvEnableTable

影像	音效	視訊
		X

爲視訊 (DB2Video 資料) 啓用某表格。會爲每個表格各呼叫一次 API。會建立 meta 資料表格，以儲存並管理表格中視訊直欄的相關屬性。爲了避免發生鎖定之虞，在呼叫此 API 之前，應用程式應先確定異動。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

控制、改變、SYSADM、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvEnableTable(  
    char *tableSpace,  
    char *tableName  
);
```

參數

tableSpace (in)
表格空間的名稱，它是儲存管理表格的配置區集合。表格空間規格具有下列三個部份：*datats*、*indexts*、*longts*，其中 *datats* 爲其中建有 meta 資料表格的表格空間；*indexts* 爲其中建有 meta 資料表格上之索引的表格空間；*longts* 爲 meta 資料表格 (像是內含 LONG VARCHAR 與 LOB 資料類型者) 中長直欄之值儲存所在的表格空間。若提供表格空間規格的任何部份一個 NULL 值，則會使用該部份的預設表格空間。

假設表格空間規格中有任一部份的值是設為 NULL，則會為該部份採用預設的表格空間。

僅適用於 EEE：指定的表格空間必須位於與使用者表格相同的節點群組。

tableName (in)

內含某視訊直欄的表格名稱。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_WARN_ALREADY_ENABLED

表格已啟用。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_TABLE_DOESNOT_EXIST

表格不存在。

MMDB_RC_TABLESPACE_NOT_SAME_NODEGROUP

您指定的表格空間，和使用者表格所在的節點群組不同。 **(僅適用於 EEE)**

範例

為表格空間 MYTS 中的視訊 (DB2Video 資料) 啟用員工表格。使用索引和長表格空間的預設值：

```
#include <dmbvideo.h>

rc = DBvEnableTable("myts,,",
    "employee");
```

為視訊 (DB2Video 資料) 啟用員工表格。使用預設表格空間：

```
#include <dmbvideo.h>

rc = DBvEnableTable(NULL,
    "employee");
```

DBvFrameDataTo24BitRGB

影像	音效	視訊
		X

將視訊框 (如 MPEG) 從 YUV 色彩值格式轉換成 24 位元 RGB 格式。使用者必須在發出 API 呼叫前，先配置一個目標緩衝區。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvFrameDataTo24BitRGB(  
    unsigned char *RGB,  
    DBvFrameData *fd,  
    unsigned long dx,  
    unsigned long dy  
);
```

參數

- RGB (out)** 指向目標 RGB 緩衝區的指標。
- fd (in)** 指向所要轉換之畫面資料的指標。
- dx (in)** 畫面寬度
- dy (in)** 畫面高度

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

範例

將視訊畫面由 MPEG 轉換成 24 位元 RGB：

```
#include <dmbshot.h>
```

```
rc = DBvFrameDataTo24BitRGB(RGB, &video->fd,  
                             video->dx, video->dy);
```

DBvGetError

影像	音效	視訊
		X

傳回有關前次錯誤的說明。請在其它任何 API 傳回錯誤碼之後，再呼叫 API。

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvGetError(  
    SQLINTEGER *sqlcode,  
    char *errorMsgText  
);
```

參數

- sqlcode (out)**
同屬 SQL 錯誤碼。
- errorMsgText (out)**
SQL 錯誤訊息文字。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。

範例

取得前次的錯誤，而此錯誤的 SQL 錯誤碼儲存在 `errCode` 中，訊息文字儲存在 `errMsg` 中：

```
#include <dmbvideo.h>

rc = DBvGetError(&errCode, &errMsg);
```

DBvGetFrame

影像	音效	視訊
		X

取得視訊檔中的現行畫面。 此畫面資料會傳回到 DBvFrameData 視訊結構中。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvGetFrame(  
    DBvIOType *video  
);
```

參數

video (update)
指向視訊結構的指標。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_RC_EOF**
已抵達檔案尾端。

範例

取得視訊檔中的現行畫面：

```
#include <dmbshot.h>

rc = DBvGetFrame(video);
```

DBvGetInaccessibleFiles

影像	音效	視訊
		X

傳回使用者表格之視訊直欄中所參照之無法存取型檔案的名稱。呼叫此 API 之前，應用程式必須先連接到資料庫。

在呼叫此 API 後，請務必釋放此 API 所配置的資源。更明確的說法是，您必須釋放檔案清單資料結構，以及檔案清單中每個項目中的檔名欄位。

授權

在所有已搜尋的使用者表格和相關管理支援表格中，已啓用視訊直欄的 SELECT 專用權

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvGetInaccessibleFiles(  
    char *tableName,  
    long *count,  
    FILEREF *(*fileList)  
);
```

參數

- tableName (in)**
完整、不完整、或 NULL 的表格名稱。 假設您指定的是表格名稱，則會搜尋該表格，以找出會參照無法存取型檔案的項目。若您指定的是 NULL 值，則會搜尋所有採用所指限定元的表格。
- count (out)**
輸出清單中的登錄數目。

fileList (out)

列出表格中所參照的無法存取型檔案。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出員工表格之視訊直欄中所參照的所有無法存取型檔案：

```
#include <dmbvideo.h>
long idx;

rc = DBvGetInaccessibleFiles("employee",
    &count, &filelist);
```

DBvGetReferencedFiles

影像	音效	視訊
		X

傳回使用者表格之視訊直欄中所參照的檔案名稱。若某檔案無法存取 (例如，無法使用環境變數規格來分辨檔名)，則檔名前面會有一個星號。由於此 API 不會用到 **FILEREF** 資料結構的 **FILENAME** 欄位，因此請將此值設為 **NULL**。呼叫此 API 之前，應用程式必須先連接到資料庫。

在呼叫此 API 後，請務必釋放此 API 所配置的資源。更明確的說法是，您必須釋放 **filelist** 資料結構。

授權

在所有已搜尋的使用者表格和相關管理支援表格中，已啟用視訊直欄的 **SELECT** 專用權

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvGetReferencedFiles(  
    char *tableName,  
    long *count,  
    FILEREF *(*fileList)  
);
```

參數

tableName (in)
完整、不完整、或 **NULL** 的表格名稱。 假設您指定的是表格名稱，則會搜尋該表格，以找出會參照檔案的項目。若指定 **NULL** 值，則會搜尋現行使用者 ID 擁有的所有表格。

count (out)

輸出清單中的登錄數目。

fileList (out)

列出表格中所參照的檔案。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出員工表格之視訊直欄中所參照的所有檔案：

```
#include <dmbvideo.h>
long idx;

rc = DBvGetReferencedFiles("employee",
    &count, &filelist);
```

DBvInitShotControl

影像	音效	視訊
		X

起始設定快照控制資料結構中的值。

授權

插入、控制

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvInitShotControl(  
    DBvShotControl *shotCtrl,  
);
```

參數

shotCtrl (in)
指向快照控制資料結構的指標。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_RC_ACCESS**
呼叫程式不具備適當的存取權限。
- MMDB_RC_NOT_CONNECTED**
應用程式與資料庫之間無有效連接。

範例

起始設定快照控制資料結構中的值：

```
#include <dmbshot.h>
```

```
rc = DBvInitShotControl(shotCtrl);
```

DBvInitStoryboardCtrl

影像	音效	視訊
		X

起始設定故事板控制資料結構中的值。

授權

插入、控制

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvInitStoryboardCtrl(  
    DBvStoryboardCtrl *sbCtrl,  
);
```

參數

shotCtrl (in)
指向快照控制資料結構的指標。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_RC_ACCESS**
呼叫程式不具備適當的存取權限。
- MMDB_RC_NOT_CONNECTED**
應用程式與資料庫之間無有效連接。

範例

起始設定故事板控制資料結構中的值：

```
#include <dmbshot.h>
```

```
rc = DBvInitStoryboardCtrl(shotCtrl);
```

DBvInsertShot

影像	音效	視訊
		X

在快照型錄中插入一個快照。

授權

插入、控制

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvInsertShot(  
    char *catalogName,  
    DBvShotType *shot,  
    DBvIOType *video,  
    char *shotHandle,  
    SQLHDBC hdbc  
);
```

參數

- catalogName (in)**
型錄的名稱。
- shot (in)**
指向要插入到型錄中之擴充快照的指標。
- shotHandle (in)**
快照控點。
- hdbc (in)**
取自 SQLConnect 中的資料庫控點。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_ACCESS

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_INVALID_CATALOG

型錄無效或不存在。

範例

在快照型錄 hotshots 中插入一個快照：

```
rc = DBvInsertShot(  
    "hotshots",          /* shot catalog name */  
    shot,                /* pointer to shot structure */  
    video,               /* pointer to video structure */  
    shotHandle,          /* pointer to shot handle */  
    hdbc);               /* database connection handle */
```

DBvIsColumnEnabled

影像	音效	視訊
		X

判斷是否已為視訊 (DB2Video 資料) 啓用某直欄。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

使用者表格上的 SYSADM、DBADM、表格擁有者或 SELECT 專用權

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvIsColumnEnabled(
    char *tableName,
    char *colName,
    short *status
);
```

參數

tableName (in)
完整或不完整的表格名稱。

colName (in)
直欄的名稱。

status (out)
指出是否已啓用直欄。此參數會傳回一個數值。Extender 也會傳回一個指出其狀態的常數。 其值與常數如下：

- 1 MMDB_IS_ENABLED

0	MMDB_IS_NOT_ENABLED
-1	MMDB_INVALID_DATATYPE

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

判斷是否已為視訊啟用員工表格中的視訊直欄：

```
#include <dmbvideo.h>
```

```
rc = DBvIsColumnEnabled("employee",  
    "video", &status);
```

DBvIsDatabaseEnabled

影像	音效	視訊
		X

判斷是否已為視訊 (DB2Video 資料) 啓用某資料庫。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvIsDatabaseEnabled(  
short *status  
);
```

參數

- status (out)**
指出是否已啓用資料庫。此參數會傳回一個數值。Extender 也會傳回一個指出其狀態的常數。 其值與常數如下：
- 1 MMDB_IS_ENABLED
 - 0 MMDB_IS_NOT_ENABLED

錯誤碼

MMDB_SUCCESS
API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

判斷是否已為視訊啟用 personnel 資料庫：

```
#include <dmbvideo.h>
```

```
rc = DBvIsDatabaseEnabled(&status);
```

DBvIsFileReferenced

影像	音效	視訊
		X

傳回清單，而此清單中列有視訊直欄中會參照所指檔案的表格項目。呼叫此 API 之前，應用程式必須先連接到資料庫。

在呼叫此 API 後，請務必釋放此 API 所配置的資源。更明確的說法是，您必須釋放檔案清單資料結構，以及檔案清單中每個項目中的檔名欄位。

授權

在所有已搜尋的使用者表格和相關管理支援表格中，已啓用視訊直欄的 SELECT 專用權

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvIsFileReferenced(
    char *tableName,
    char *fileName,
    long *count,
    FILEREF *(*tableList)
);
```

參數

tableName (in)
完整、不完整、或 NULL 的表格名稱。 假設您指定的是表格名稱，則會搜尋該表格，以找出會參照所指檔案的項目。若指定 NULL 值，則會搜尋現行使用者 ID 擁有的所有表格。

fileName (in)

所參照到的檔案名稱。

count (out)

輸出清單中的登錄數目。

tableList (out)

列出參照指定檔案的表格項目。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_MALLOC

系統無法配置記憶體供傳回結果時使用。

範例

列出參照檔案 /videos/ajones.mpg 的員工表格中的視訊直欄項目：

```
#include <dmbvideo.h>
long idx;

rc = DBvIsFileReferenced(NULL,
    "/videos/ajones.mpg",
    &count, &tableList);
```

DBvIsIndex

影像	音效	視訊
		X

檢查視訊索引是否存在。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvIsIndex(  
    char *fileName,  
    short *status  
);
```

參數

- fileName (in)**
所參照到的檔案名稱。
- status (out)**
指出索引是否存在。值 1 表示索引存在；值 0 表示索引不存在。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_ERROR**
狀態無效。

範例

檢查視訊檔 \videos\ajones.mpg 的索引是否存在：

```
#include <dmbshot.h>
```

```
rc = DBvIsIndex("\\videos\\ajones.mpg", &status);
```

DBvIsTableEnabled

影像	音效	視訊
		X

判斷是否已啓用視訊 (DB2Video 資料) 的某個表格。呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvIsTableEnabled(  
    char *tableName,  
    short *status  
);
```

參數

tableName (in)
表格名稱。

status (out)
指出是否已啓用表格。此參數會傳回一個數值。Extender 也會傳回一個指出其狀態的常數。 其值與常數如下：

- 1 MMDB_IS_ENABLED
- 0 MMDB_IS_NOT_ENABLED

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

判斷是否已為視訊啓用員工表格：

```
#include <dmbvideo.h>
```

```
rc = DBvIsTableEnabled("employee",  
    &status);
```

DBvMergeShots

影像	音效	視訊
		X

將兩個快照合併成一個。產生的快照會採用第一個快照的快照控點與起始畫面。並將這兩個快照中結束畫面較大者，用於產生的快照中。第二個快照控點指向的橫列會被刪除。

授權

控制、選取、刪除、更新

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvMergeShots(  
    char *catalogName,  
    char *shotHandle1,  
    char *shotHandle2,  
    SQLHDBC hdbc  
);
```

參數

- catalogName (in)**
快照型錄的名稱。
- shotHandle1 (in)**
第一個快照的控點。
- shotHandle2 (in)**
第二個快照的控點。

hdbc (in)

取自 SQLConnect 中的資料庫控點。

錯誤碼**MMDB_SUCCESS**

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_CANNOT_MERGE

無法合併快照。

MMDB_RC_INVALID_CATALOG

型錄無效或不存在。

範例

將其控點分別是 shotHandle1 與 shotHandle2 的快照，合併到 hotshots 型錄中：

```
#include <dmbshot.h>
```

```
rc = DBvMergeShots("hotshots", shotHandle1,  
    shotHandle2, hdbc);
```

DBvOpenFile

影像	音效	視訊
		X

配置空間給 DBvIOType 結構，並開啓視訊檔以存取圖點。當順利開啓視訊時，其會指向第一個畫面編號（畫面 0）。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvOpenFile(
    DBvIOType **video,
    char *fileName,
);
```

參數

- video (out)**
指向視訊結構指標的指標。
- fileName (in)**
所要開啓之視訊檔的名稱。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。

MMDB_RC_CANNOT_OPEN

無法開啓視訊檔。

MMDB_RC_NO_MEMORY

記憶體不足。

MMDB_RC_NO_INDEX

無視訊隨機存取索引。

範例

開啓視訊檔 \videos\ajones.mpg：

```
#include <dmbshot.h>
```

```
rc = DBvOpenFile(&videoa,  
                "\videos\ajones.mpg");
```

DBvOpenHandle

影像	音效	視訊
		X

配置空間給 DBvIOType 結構，並開啓視訊控點以存取圖點。結構會指向第一個畫面編號 (畫面 0)。視訊可以是 BLOB。視訊會複製到 DB2VIDEOTEMP 環境變數指定之目錄內的暫用檔中。並且會根據隨機存取索引的存在與否，來設定 isIdx 旗號。

授權

選取

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvOpenHandle(
    DBvIOType **video,
    DB2Video *videoHandle
    SQLHDBC hdbc
);
```

參數

video (out)
指向視訊結構的指標。

videoHandle (in)
視訊控點。

hdbc (in)
取自 SQLConnect 中的資料庫控點。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_CANNOT_OPEN

無法開啓視訊檔。

MMDB_RC_NO_MEMORY

記憶體不足。

MMDB_RC_NO_INDEX

無視訊隨機存取索引。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_INVALID_HANDLE

視訊控點無效。

範例

透過 videoa 指標開啓 videoHandle：

```
#include <dmbshot.h>
```

```
rc = DBvOpenHandle(&oa, videoHandle, hdbc);
```

DBvPlay

影像	音效	視訊
		X

開啓從屬站上的視訊播放程式，並播放一段視訊。此視訊可儲存在視訊直欄或外部檔中：

- 若視訊是儲存在外部檔中，則您可以將檔名或視訊控點傳遞給此 API。API 會利用從屬站環境變數 DB2VIDEOPATH，分辨出檔案的位置。您必須自從屬站工作站來存取該檔案。
- 假設視訊是儲存在直欄中，則您必須將視訊控點傳遞給 API。應用程式必須連接資料庫，而且必須有儲存視訊的表格的讀取權。

若此視訊儲存在直欄中，Extender 會建立暫用檔，並從該直欄將物件內容複製到此暫用檔。若視訊儲存在外部檔，而且無法使用環境變數中的值分辨它的相對檔名，或無法在從屬站機器上存取此檔案，則 Extender 也可能會建立暫用檔。此暫用檔會在使用 DB2VIDEOTEMP 環境變數指定的目錄中建立。然後，Extender 會由暫用檔中播放該視訊。

授權

假設您所播放的是直欄中的視訊，請選取使用者表格上的權限。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

播放儲存在直欄中的視訊

```
long DBvPlay(
    char *playerName,
    MMDB_PLAY_HANDLE,
    DB2Video *videoHandle,
    waitFlag
);
```

語法

播放儲存成檔案的視訊

```
long DBvPlay(
    char *playerName,
    MMDB_PLAY_FILE,
    char *fileName,
    waitFlag
);
```

參數

playerName (in)

視訊播放程式的名稱。假設您設為 NULL，則會採用 DB2VIDEOPAYER 環境變數所指定的預設視訊播放程式。

MMDB_PLAY_HANDLE (in)

此常數指出將視訊儲存成 BLOB。

MMDB_PLAY_FILE (in)

此常數指出將視訊儲存成可自從屬站存取的檔案。

videoHandle (in)

視訊的控點。在您播放直欄中的視訊時，必須傳遞此參數。假設視訊控點所呈現的是外部檔，則會以從屬站環境變數 DB2VIDEOPATH 來分辨出檔案的位置。

fileName (in)

內含視訊的檔案名稱。API 會利用從屬站環境變數 DB2VIDEOPATH，分辨出檔案的位置。您必須自從屬站工作站來存取該檔案。

waitFlag (in)

此常數是指出您的程式在繼續執行前，得花多久時間等待使用者關閉播放程式。指定 MMDB_PLAY_WAIT，則會在和您的應用程式一樣的緒中執行播放程式；指定 MMDB_PLAY_NO_WAIT，則是在個別緒中執行播放程式。

DBvPlay

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

播放 videoHandle 所識別的視訊。在和應用程式一樣的緒中執行預設播放程式：

```
#include <dmbvideo.h>
```

```
rc = DBvPlay(NULL, MMDB_PLAY_HANDLE, videoHandle,  
              MMDB_PLAY_WAIT);
```


DBvPrepareAttrs

影像	音效	視訊
		X

準備使用者提供的視訊屬性。當儲存或更新帶有使用者提供之屬性的視訊物件時，即會使用此 API。在伺服器上執行的 UDF 碼會預期資料為『big endian』格式，而此格式可為大部份 UNIX 平台所使用。若使用『little endian』格式儲存或更新視訊物件 (亦即，在非 UNIX 從屬站中)，則必須在使用 DBvPrepare API 之後，才能提出儲存或更新要求。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
void DBvPrepareAttrs(  
    MMDBVideoAttrs *vidAttr  
);
```

參數

vidAttr (in)
 視訊的屬性 (使用者所提供)。

範例

```
準備使用者提供的視訊屬性：  
  
#include <dmbvideo.h>  
  
DBvPrepareAttrs(&vidattr);
```

DBvReorgMetadata

影像	音效	視訊
		X

- 『清除』和視訊有關的 meta 資料表格，例如：
- 回收視訊 meta 資料表格中不再使用的空間。
 - 刪除視訊 meta 資料表格中會參照不復存在之視訊檔的項目。

呼叫此 API 之前，應用程式必須先連接到資料庫。

授權

改變、控制、SYSADM、SYSCTRL、SYSMAINT、DBADM

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbvideo.lib	libdmbvideo.a (AIX) libdmbvideo.sl (HP-UX) libdmbvideo.so (Solaris)

併入檔

dmbvideo.h

語法

```
long DBvReorgMetadata(  
    char *tableName,  
);
```

參數

tableName (in)
完整、不完整、或 NULL 的表格名稱。 假設您指定的是表格名稱，則會為與您指定的使用者表格有關的視訊 meta 資料表格，執行清除動作。若您指定的是 NULL 值，則會為現行使用者 ID 所擁有之所有表格中的視訊直欄，清除其相關的 meta 資料表格。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NO_AUTH

呼叫程式不具備適當的存取權限。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

範例

為員工表格中的視訊直欄，清除其相關的 meta 資料表格：

```
#include <dmbvideo.h>
```

```
rc = DBvReorgMetadata("employee");
```

DBvSetFrameNumber

影像	音效	視訊
		X

將目前畫面設定成指定的畫面編號。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbmpeg.lib	libdmbmpeg.a (AIX) libdmbmpeg.sl (HP-UX) libdmbmpeg.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvSetFrameNumber(
    DBvIOType *video
    unsigned long frameNumber
);
```

參數

- video (in)**
指向視訊結構的指標。
- frameNumber (in)**
所要求之畫面的編號。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_FRAME_NOT_FOUND**
找不到您所要的畫面。

MMDB_NO_INDEX

視訊索引不存在。

範例

將目前畫面設定成視訊檔中的畫面編號 85：

```
#include <dmbshot.h>
```

```
rc = DBvSetFrameNumber(video, 85);
```

DBvSetShotComment

影像	音效	視訊
		X

更新快照中的唯讀註解。

授權

控制、更新

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvSetShotComment(  
    char *catalogName,  
    char *shotHandle,  
    char *comment,  
    SQLHDBC hdbc  
);
```

參數

- catalogName (in)**
型錄的名稱。
- shotHandle (in)**
所要更新之快照的控點。
- comment (in)**
快照的新註解。
- hdbc (in)**
取自 SQLConnect 中的資料庫控點。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_CANNOT_UPDATE

API 無法更新快照。

MMDB_RC_INVALID_CATALOG

型錄無效或不存在。

範例

用型錄 hotshots 中的 shotHandle 來變更說明快照的註解：

```
#include <dmbshot.h>
```

```
rc = DBvSetShotComment("hotshot", shotHandle,  
    "This is a hot shot.", hdbc);
```

DBvUpdateShot

影像	音效	視訊
		X

取代型錄中某個視訊快照的屬性。除了註解外，所有的屬性均會換成 DBvShotType 結構中的屬性。假設標註指標為 NULL，則現存的標註仍維持不變。

授權

控制、更新

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbshot.lib	libdmbshot.a (AIX) libdmbshot.sl (HP-UX) libdmbshot.so (Solaris)

併入檔

dmbshot.h

語法

```
long DBvUpdateShot(  
    char *catalogName,  
    DBvShotType *shot,  
    char *shotHandle,  
    SQLHDBC hdbc  
);
```

參數

- catalogName (in)**
型錄的名稱。
- shot (in)**
指向內含快照屬性之快照資訊結構的指標。
- shotHandle (in)**
快照控點。
- hdbc (in)**
取自 SQLConnect 中的資料庫控點。

錯誤碼

MMDB_SUCCESS

API 呼叫程序執行成功。

MMDB_RC_NOT_CONNECTED

應用程式與資料庫之間無有效連接。

MMDB_RC_CANNOT_UPDATE

API 無法更新快照。

MMDB_RC_NO_SHOT

快照不存在。

MMDB_RC_INVALID_CATALOG

型錄無效或不存在。

範例

更新 hotshots 型錄某個快照的屬性：

```
#include <dmbshot.h>
```

```
rc = DBvUpdateShot("hotshots", shot,  
    shothandle, hdbc);
```

DMBRedistribute (僅適用於 EEE)

影像	音效	視訊
X		

當在節點群組中新增或移除某節點時，或者當為某節點群組建立一個新分割對映表時，則會重新分送 QBIC 特性。

授權

API 必須由擁有該案例之 ID 來執行：

檔案庫

Windows	AIX 與 Solaris
dmbrd.lib	libdmbrd.a (AIX) libdmbrd.so (Solaris)

併入檔

dmbrdst.h

語法

```
long DMBRedistribute (  
  char *pNodeGroupName,  
  char DataRedistOption /* ""continue"" use CONTINUE parameter */  
);  
/* blank:start redistribution */
```

參數

pNodeGroupName (in)
要重新分送之節點群組的名稱。

錯誤碼

- MMDB_SUCCESS**
API 呼叫程序執行成功。
- MMDB_RD_NO_CONTINUE**
在不使用 CONTINUE 參數的情況下重新提出。
- MMDB_RD_CONTINUE**
使用 CONTINUE 參數重新提出。

範例

重新分送 groupone 節點群組中的 QBIC Extender 資料：

```
#include <dmbdst.h>
```

```
rc = DMBRedistribute(groupone,"continue");
```

QbAddFeature

影像	音效	視訊
X		

在目前所開啓的型錄中新增一項特性。QbAddFeature 建立資料庫內指定特性的特性表。 將影像新增至使用者表格的影像直欄之後，請使用 QbReCatalogColumn API，將影像的項目新增至特性表格並分析影像。

授權

改變

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbAddFeature(  
    QbCatalogHandle cHdl,  
    char *featureName  
);
```

參數

cHdl (in)
指向型錄之控點的指標。

featureName (in)
特性的名稱。 Image Extender 具有下列特性：

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

錯誤碼

qbicECInvalidHandle

型錄控點無效。

qbicECCatalogReadOnly

型錄開啓後僅供讀取。

qbicECDupFeature

型錄中已有該項特性。

qbiECInvalidFeatureClass

您指定之特性的名稱格式無效。

範例

在控點 CatHdl 所識別的型錄中，新增 QbColorFeatureClass 特性：

```
#include <dmbqbapi.h>
```

```
rc = QbAddFeature(CatHdl,  
                  QbColorFeatureClass);
```

QbCatalogColumn

影像	音效	視訊
X		

爲您的使用者表格之影像直欄中尚未編目的影像加以編目。API 會在特性表格中新增各個影像的相關項目，並分析影像。在 API 分析影像時，其會建立影像資料並將之儲存在特性表格中影像的項目中。在特性方面會採用預設參數。型錄必須呈開啓狀態。

授權

插入

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbCatalogColumn(  
    QbCatalogHandle cHdl  
);
```

參數

cHdl (in)
指向型錄之控點的指標。

錯誤碼

qbicECInvalidHandle
型錄控點無效。

qbicECInvalidCatalog
對於型錄來說，指定的控點或表格直欄無效。

qbicECCatalog Errors

建立個別影像的型錄時發生錯誤，這些錯誤會被記錄下來。沒有發生 ROLLBACK。

qbicECImageNotFound

該影像找不到或者無法存取。

qbicECCatalogRO

該型錄為唯讀的。

qbicECSQLError

發生 SQL 錯誤。

範例

```
#include <dmbqbapi.h>

rc = QbCatalogColumn(CatHd1);
```

QbCatalogImage

影像	音效	視訊
X		

編目整個影像。API 會在特性表格中新增影像的相關項目，並分析影像。在 API 分析影像時，其會建立影像資料並將之儲存在特性表格中影像的項目中。影像控點必須取自現行 QBIC 型錄的相關影像直欄中。在影像的編目上，會根據目前所定義的特性類別而定。在型錄中的特性方面會採用預設參數。

授權

插入

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbCatalogImage(  
    QbCatalogHandle cHdl,  
    char *imgHandle  
);
```

參數

- cHdl (in)**
指向型錄之控點的指標。
- imgHandle (in)**
影像的控點。

錯誤碼

qbicECInvalidHandle
型錄控點無效。

qbicECImageNotFound

該影像找不到或者無法存取。

qbicECCatalogRO

該型錄為唯讀的。

範例

對控點 `Img_hdl` 所識別的影像加以編目：

```
#include <dmbqbapi.h>
```

```
rc = QbCatalogColumn(CatHdl, Img_hdl);
```

QbCloseCatalog

影像	音效	視訊
X		

關閉型錄。API 會釋放所開啓的型錄控點以及所配置的資源。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbCloseCatalog(  
    QbCatalogHandle cHdl  
);
```

參數

cHdl (in)
指向型錄之控點的指標。

錯誤碼

qbicECInvalidHandle
型錄控點無效。

範例

```
關閉控點 CatHdl 所識別的型錄：  
#include <dmbqbapi.h>  
  
rc = QbCloseCatalog(CatHdl);
```

QbCreateCatalog

影像	音效	視訊
X		

在現行連接的資料庫中為指定的影像直欄建立型錄。 必須為影像資料啟用直欄。 API 會建立型錄的名稱，以做為其限定元。

授權

改變

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbCreateCatalog(  
    char *tableName,  
    char *columnName,  
    SQLINTEGER autoCatalog,  
    char *reserved  
);
```

參數

tableName (in)
 內含某影像直欄的表格名稱。

columnName (in)
 影像直欄的名稱，以便為其建立型錄。

autoCatalog (in)
 指示是否要自動編目新增至影像直欄的影像，亦即，新增至特性表格並加以分析。指定 1，表示啟用自動編目特性，指定 0 則表示停用。若沒有將

QbCreateCatalog

自動編目設定為開啓狀態，請使用 `QbCatalogColumn` 或 `QbCatalogImage` API，針對您新增至影像直欄的影像進行編目。

reserved (in)

目前不使用。

錯誤碼

qbicECSqlError

發生 SQL 錯誤。

qbicECNotEnabled

未對 `DB2Image` 資料類型啓用資料庫、表格或直欄。

qbicECDupCatalog

型錄已存在。

qbicECunsupportedOption

指定未支援的選項。

qbicECerrorParameterTooLong

參數太長無法處理。

qbicECqerr

發生 QBIC 錯誤，並產生訊息。

qbicECqerrUnknown

發生內部 QBIC 錯誤，並產生適用的錯誤訊息。

範例

為員工表格之圖片直欄中的影像建立一份型錄。啓用自動編目特性：

```
#include <dmbqbapi.h>

rc = QbCreateCatalog("employee",
    "picture", 1);
```

QbDeleteCatalog

影像	音效	視訊
X		

從現行資料庫刪除指定的型錄。

授權

改變

檔案庫

OS/2 與 Windows

dmbqbapi.lib

AIX、HP-UX 及 Solaris

libdmbqbapi.a (AIX)
libdmbqbapi.sl (HP-UX)
libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbDeleteCatalog(  
    char *tableName,  
    char *columnName  
);
```

參數

tableName (in)

內含影像直欄的表格名稱。

columnName (in)

和型錄有關之影像直欄的名稱。

錯誤碼

qbicECInvalidHandle

型錄控點無效。

qbicECCatalogInUse

其他人正在使用該型錄。

QbDeleteCatalog

qbicECCatalogRO

該型錄為唯讀的。

qbicECSystem

發生系統錯誤。

qbicECSqlError

發生 SQL 錯誤。

範例

刪除和員工表格之圖片直欄有關的 QBIC 型錄：

```
#include <dmbqbapi.h>
```

```
rc=QbDeleteCatalog("employee", "picture");
```

QbGetCatalogInfo

影像	音效	視訊
X		

傳回 QbCatalogInfo 結構，而其中所含的資訊如下：

- 型錄所隸屬之使用者表格與影像直欄的名稱。
- 型錄所含的特性數目。
- 有否啓用自動編目特性。

授權

選取

檔案庫

OS/2 與 Windows

dmbqbapi.lib

AIX、HP-UX 及 Solaris

libdmbqbapi.a (AIX)
libdmbqbapi.sl (HP-UX)
libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbGetCatalogInfo(
    QbCatalogHandle cHdl,
    QbCatalogInfo *catInfo
);
```

參數

cHdl (in)

指向型錄之控點的指標。

catInfo (out)

型錄資訊結構。

QbGetCatalogInfo

錯誤碼

qbicECInvalidHandle

型錄控點無效。

範例

取得控點 CatHdl 所識別之型錄的相關資訊，並將其傳回 catInfo 結構中：

```
#include <dmbqbapi.h>
```

```
rc = QbGetCatalogInfo(CatHdl, &catInfo);
```


QbListFeatures

影像	音效	視訊
X		

傳回其中列有型錄中目前所含之作用中特性的清單。而該清單會傳回到您配置的緩衝區中。

授權

選取

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbListFeatures(  
    QbCatalogHandle cHdl,  
    SQLINTEGER bufSize,  
    SQLINTEGER *count,  
    char *featureNames  
);
```

參數

- cHdl (in)**
指向型錄之控點的指標。
- bufSize (in)**
您緩衝區的大小。要預估所需的緩衝區大小時，您可以將 QbGetCatalogInfo API 所傳回的特性數目，乘上最長特性名稱之長度值，來預估其大小。儲存在緩衝區中的特性名稱，彼此間會以空白字元隔開。
- count (out)**
所傳回之特性名稱的數目。

QbListFeatures

featureNames (out)

您緩衝區中的特性名稱陣列。

錯誤碼

qbicECInvalidHandle

型錄控點無效。

qbicECTruncateData

傳回的資料遭到截斷，理由是傳回緩衝區太小。

範例

取得清單，而此清單中列有控點 `CatHdl` 所識別之型錄內的作用中特性。並且將資訊儲存在 `featureNames` 陣列中。

首先，先計算 `bufSize`，此為放置清單時應具備的緩衝區大小。使用 `QbGetCatalogInfo` API，將特性數目傳回到 `catInfo` 結構中。然後將該數目值乘上常數 `qbiMaxFeatureName` 值 (此為最長之特性名稱的大小)。

```
#include <dmbqbapi.h>

rc = QbGetCatalogInfo(CatHdl, &catInfo);

bufSize =
    catInfo.featureCount*qbiMaxFeatureName;

rc = QbListFeatures(CatHdl, bufSize,
    count, featureNames);
```

QbOpenCatalog

影像	音效	視訊
X		

開啓某特定影像直欄的相關 QBIC 型錄。您可以在讀取模式或更新模式下開啓型錄。API 會傳回所開啓型錄的相關控點。然後，您可以將此控點用於其它 API 中，來管理與移入型錄。

在您使用完後，請記得關閉該型錄。

授權

無

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbOpenCatalog(  
    char *tableName,  
    char *columnName,  
    SQLINTEGER mode,  
    QbCatalogHandle *cHdl  
);
```

參數

- tableName (in)**
內含影像直欄的表格名稱。
- columnName (in)**
影像直欄的名稱。
- mode (in)**
指出您想在何種模式下開啓型錄。有效值爲 qbiRead 與 qbiUpdate。

QbOpenCatalog

cHdl (out)

指向型錄之控點的指標。

錯誤碼

qbicECCatalogNotFound

找不到該型錄。

qbicECCatalogInUse

其他人正在使用該型錄。

qbicECOpenFailed

無法開啓該型錄。

qbicECNotEnabled

未啓用該型錄。

qbicECNoCatalogFound

找不到任何型錄。

qbicECSQLException

發生 SQL 錯誤。

qbicECSystem

發生系統錯誤。

範例

在讀取模式下開啓員工表格之圖片直欄的相關型錄：

```
#include <dmbqbapi.h>

rc=QbOpenCatalog("employee", "picture",
    qbiread, &CatHdl);
```

QbQueryAddFeature

影像	音效	視訊
X		

將指定特性新增至 QBIC 型錄。

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQueryAddFeature(  
    QbQueryHandle qObj,  
    char *featureName  
);
```

參數

qObj (in)
查詢物件的控點。

featureName (in)
所要新增之查詢特性的名稱。影像 Extender 所提供的特性如下：

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

QbQueryAddFeature

錯誤碼

qbiECinvalidQueryHandle

您指定的查詢物件控點所參照的查詢物件無效。

qbiECunknownFeatureClass

您指定的特性不是一個可辨識的特性類別名稱。

qbiECinvalidFeatureClass

您指定之特性的名稱格式無效。

qbiECfeaturePresent

您指定的特性早已是查詢物件內的成員。

qbiECallocation

系統無法配置足夠的記憶體。

範例

將 QbColorFeatureClass 特性，新增到 qoHandle 控點所識別的查詢物件中。

```
#include <dmbqbapi.h>
```

```
rc = QbQueryAddFeature(qoHandle,  
    "QbColorFeatureClass");
```

QbQueryCreate

影像	音效	視訊
X		

建立一個查詢物件，並傳回控點。您可以將此控點用於其它 API 上，來操作查詢物件。

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQueryCreate(  
    QbQueryHandle *qObj  
);
```

參數

qObj (out)
指向查詢控點的指標。若未成功，則此控點會設為 0。

錯誤碼

qbiEAllocation
系統無法配置足夠的記憶體。

範例

建立一個查詢物件，並傳回 qoHandle 中的控點：

QbQueryCreate

```
#include <dmbqbapi.h>

rc = QbQueryCreate(&qoHandle);
```


QbQueryDelete

影像	音效	視訊
X		

刪除未命名的查詢物件。 API 會將查詢物件與任何新增特性所用的記憶體全數釋放。

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQueryDelete(  
    QbQueryHandle qObj  
);
```

參數

qObj (in)
查詢物件 的控點。

錯誤碼

qbiECInvalidQueryHandle
您指定之查詢物件控點所參照的查詢無效。

範例

```
刪除控點 qoHandle 所識別的查詢物件：  
  
#include <dmbqbapi.h>  
rc = QbQueryDelete(qoHandle);
```

QbQueryGetFeatureCount

影像	音效	視訊
X		

傳回新增到查詢物件中之特性的數目。 Image Extender 所提供的特性如下：

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQueryGetFeatureCount(  
    QbQueryHandle qObj,  
    SQLINTEGER* count  
);
```

參數

- qObj (in)**
查詢物件的控點。
- count (out)**
指向變數的指標，而此變數是用以設定所出現之特性的數目。

錯誤碼

qbiECInvalidQueryHandle

您指定的查詢物件控點所參照的查詢物件無效。

範例

傳回控點 qoHandle 所識別之查詢物件中的特性數目：

```
#include <dmbqbapi.h>
```

```
rc = QbQueryGetFeatureCount(qoHandle,  
                             &count);
```

QbQueryGetString

影像	音效	視訊
X		

傳回查詢中的查詢字串。您可以將查詢字串用來輸入您應用程式中的 UDF，例如 UDF QbScoreFromStr 或 API QbQueryStringSearch。

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQueryGetString(  
    QbQueryHandle qObj,  
    (char*)* queryString  
);
```

參數

- qObj (in)**
查詢物件的控點。
- queryString (out)**
指向查詢物件的查詢字串。

錯誤碼

- qbiEInvalidQueryHandle**
您指定之查詢控點所參照的查詢無效。

範例

傳回控點 qrHandle 所識別之查詢物件的查詢字串。

```
#include <dmbqbapi.h>

SQLRETURN rc;
char *queryString;
QbQueryHandle qrHandle

rc = QbQueryGetString(qrHandle, &queryString);
if (rc == 0) {
    ...                /* 傳回的 queryString 將用來輸入 UDF */
    free((void*)queryString); /* 您必須釋放 queryString */
    queryString=(char*)0;
}
```

QbQueryListFeatures

影像	音效	視訊
X		

傳回查詢物件中現有特性的清單。 API 將清單傳回您配置的緩衝區。 Image Extender 所提供的特性如下：

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQueryListFeatures(  
    QbQueryHandle qObj,  
    SQLINTEGER bufSize,  
    SQLINTEGER* count,  
    char *featureNames  
);
```

參數

qObj (in)
查詢物件的控點。

bufSize (in)

featureNames 緩衝區的大小。請以 qbiMaxFeatureName 常數做為作為緩衝區的大小。查詢物件特性是以字串名稱做為識別。

count (out)

所傳回之特性名稱的數目。

featureNames (out)

指向查詢物件之特性名稱的陣列。此名稱陣列會儲存在您配置的緩衝區中。

錯誤碼**qbiECInvalidQueryHandle**

您指定之查詢控點所參照的查詢無效。

範例

傳回控點 qoHandle 所識別之查詢物件中的特性數目。使用 qbiMaxFeatureName 常數，來判斷您需要多大的緩衝區。將特性名稱傳回到 feats 緩衝區中，並將特性的數目傳回給 retCount 變數：

```
#include <dmbqbapi.h>

bufSize = qbiMaxFeatureName;

rc = QbQueryListFeatures(qoHandle, bufSize,
    &retCount, feats);
```

QbQueryNameCreate

影像	音效	視訊
X		

儲存查詢物件並加以命名，使該查詢物件能用於 UDF 中。您提供此名稱並可提供查詢物件的說明。

註:

- 1. 僅適用於 **EEE**： 已分割資料庫環境中不支援 QbQueryNameCreate。
- 2. 在未來版本中，非分割資料庫環境也不支援 QbQueryNameCreate。若要儲存查詢，您應該使用 QbQueryGetString 以取得查詢字串，並儲存該字串供稍後應用程式之使用。

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQueryNameCreate(  
    QbQueryHandle qObj,  
    char *name,  
    char *description  
);
```

參數

- qObj (in)**
查詢物件的控點。
- name (in)**
查詢物件的名稱。 此名稱長度不可超過 18 個字元。

description (in)

查詢物件的簡短說明，長度可在 250 個字元內。

錯誤碼**qbiECinvalidQueryHandle**

您指定的查詢物件控點，所參照的查詢無效。

範例

針對以 QbQueryCreate API 所建的查詢物件，提供該查詢物件的名稱與說明：

```
#include <dmbqbapi.h>

rc = QbQueryNameCreate(qHandle,
    "fshavgcol",
    "average color query, 10/15/96");
```

QbQueryNameDelete

影像	音效	視訊
X		

刪除查詢物件。該查詢物件必須已透過 QbQueryNameCreate API 命名並儲存之。

註:

- 1. 僅適用於 **EEE**：已分割資料庫環境中不支援 QbQueryNameDelete。
- 2. 在未來版本中，非分割資料庫環境也不支援 QbQueryNameDelete。

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQueryNameDelete(  
    char *name  
);
```

參數

name (in)
您所要刪除之查詢物件的名稱。

錯誤碼

qbiECinvalidQueryHandle
您指定的查詢物件控點，所參照的查詢物件無效。

範例

刪除 fshavgcol 查詢物件：

```
#include <dmbqbapi.h>
```

```
rc = QbQueryNameDelete("fshavgcol",);
```

QbQueryNameSearch

影像	音效	視訊
X		

搜尋 QBIC 型錄，以找出符合查詢物件中所含之搜尋準則的影像。查詢物件是以其名稱做為識別。 搜尋結果 (內含影像控點與其 QBIC 搜尋範圍) 會儲存在從屬站記憶體的结果陣列中。 並且會根據結果的範圍加以排序。

註:

- 1. 僅適用於 **EEE**： 已分割資料庫環境中不支援 QbQueryNameSearch。
- 2. 在未來版本中，非分割資料庫環境也不支援 QbQueryNameSearch。若要儲存查詢，您應該使用 QbQueryGetString 以取得查詢字串，並儲存該字串供稍後應用程式之使用。

授權

選取

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQueryNameSearch(  
    char *qName,  
    char *tableName,  
    char *columnName,  
    SQLINTEGER maxReturns,  
    QbQueryScope* scope,  
    SQLINTEGER resultType,  
    SQLINTEGER* count,  
    QbResult* returns  
);
```

參數

qName (in)

查詢物件的名稱。

tableName (in)

內含您要搜尋之影像直欄的表格名稱。

columnName (in)

影像直欄的名稱。必須為影像資料啟用直欄。

maxReturns (in)

您希望傳回之影像的數目上限。

scope (in) (保留)

必須設為 0 (NULL)

resultType (in) (保留)

必須設為 qbiArray。

count (out)

指向所傳回之影像數目的指標。假設傳回 0，請確定您已對查詢物件中的所有特性，編目影像直欄。

returns (out)

指向保有傳回結果之 QbResult 結構陣列的指標。請確定您所配置的緩衝區大小放得下您所要的全部結果。

錯誤碼

qbiECInvalidQueryHandle

您指定的查詢物件控點，所參照的查詢物件無效。

範例

為員工表格之圖片直欄中的已編目影像，執行查詢 FSHAVGCOL。確定所傳回的影像不超過 6 個：

```
#include <dmbqbapi.h>
```

```
rc = QbQueryNameSearch("fshavgcol",
    "employee", "picture",
    6, 0, qbiArray, &count, &returns);
```

QbQueryRemoveFeature

影像	音效	視訊
X		

將查詢特性自查詢物件中移除，並解除相關記憶體的配置： Image Extender 所提供的特性如下：

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQueryRemoveFeature(  
    QbQueryHandle qObj,  
    char *featureName  
);
```

參數

- qObj (in)**
查詢物件的控點。
- featureName (in)**
所要移除之特性的名稱。

錯誤碼

qbiECinvalidQueryHandle

您指定的查詢物件控點所參照的查詢物件無效。

qbiECinvalidFeatureClass

您指定之特性的名稱格式無效。

qbiECfeatureNotPresent

您指定的特性並非查詢物件內的成員。

範例

在控點 `qoHandle` 所識別的查詢物件中，移除 `QbColorFeatureClass` 特性：

```
#include <dmbqbapi.h>
```

```
rc = QbQueryRemoveFeature(qoHandle,  
    "QbColorFeatureClass");
```

QbQuerySearch

影像	音效	視訊
X		

搜尋 QBIC 型錄，以找出符合查詢物件中所含之搜尋準則的影像。查詢物件是以查詢物件控點做為識別。搜尋結果 (內含影像控點與其 QBIC 搜尋範圍) 會儲存在從屬站記憶體的结果陣列中。並且會根據其範圍加以排序。

授權

選取

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQuerySearch(  
    QbQueryHandle qObj,  
    char *tableName,  
    char *columnName,  
    SQLINTEGER maxReturns,  
    QbQueryScope* scope,  
    SQLINTEGER resultType,  
    SQLINTEGER* count,  
    QbResult* returns  
);
```

參數

- qObj (in)**
查詢物件的控點。
- tableName (in)**
內含您要搜尋之影像直欄的表格名稱。

columnName (in)

影像直欄的名稱。必須為影像資料啟用直欄。

maxReturns (in)

您希望傳回之影像的數目上限。

scope (in) (保留)

必須設定為 0 (NULL)。

resultType (in) (保留)

必須設為 qbiArray。

count (out)

指向所傳回之影像數目的指標。假設傳回 0，請確定您已對查詢物件中的所有特性，編目影像直欄。

returns (out)

指向保有傳回結果之 QbResult 結構陣列的指標。請確定您配置的緩衝區足夠保留預期的全部結果。

錯誤碼

qbiEInvalidQueryHandle

您指定的查詢物件控點所參照的查詢物件無效。

範例

查詢員工表格之圖片直欄中的已編目影像。確定所傳回的影像不超過 6 個：

```
#include <dmbqbapi.h>

rc = QbQuerySearch(qHandle, "employee",
    "picture", 6, 0, qbiArray,
    &count, &returns);
```

QbQuerySetFeatureData

影像	音效	視訊
X		

設定查詢物件中某特性之相關影像資料的來源。 您只能在新增特性到查詢物件中後，才能設定資料的來源。 資料來源可以是使用者表格、檔案或工作站緩衝區的影像。唯有在非分割資料庫環境中，才能使用從屬站檔案或工作站緩衝區做為資料來源。此外，您可以明確地指定平均色或直方圖色特性的資料。

使用 QbQuerySetFeatureData 於伺服器檔案中設定影像資料的來源後，使用 QbQueryStringSearch。 QbQuerySearch 不使用 QbQuerySetFeatureData 於伺服器檔案中設定影像資料的來源。

Image Extender 具有下列特性：

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQuerySetFeatureData(  
    QbQueryHandle qObj,  
    char *featureName,  
    QbImageSource* imgSource  
);
```

參數

qObj (in)

查詢物件的控點。

featureName (in)

所要設定的特性名稱。

imgSource (in)

指向影像來源結構的指標。假設您將 `imgSource` 的值設為 0 (NULL)，則表示不應變更特性中的資訊。進一步資訊請參閱第 167 頁的『使用資料來源結構』。

錯誤碼

qbiEInvalidQueryHandle

您指定的查詢物件控點所參照的查詢物件無效。

qbiEUnknownFeatureClass

您指定的特性不是一個可辨識的特性類別名稱。

qbiEInvalidFeatureClass

您指定之特性的名稱格式無效。

qbiEFeatureNotPresent

您指定的特性並非查詢物件內的成員。

qbiEFileUnreadable

找不到或無法讀取影像來源檔。

範例

在查詢物件中設定直方圖顏色特性的資料來源。此特性的資料來源是從屬站工作站上的檔案：

```
#include <dmbqbapi.h>

QbQueryHandle qoHandle;
QbImageSource imgSource;

imgSource.sourceType = qbiSource_ClientFile;
strcpy(featureName, "QbColorHistogramFeatureClass");
strcpy(imgSource.clientFile, "/tmp/image.gif");

rc = QbQuerySetFeatureData(qoHandle, featureName, &imgSource);
```

QbQuerySetFeatureWeight

影像	音效	視訊
X		

設定查詢物件中您指定之特性的加權。

授權

無。

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQuerySetFeatureWeight(  
    QbQueryHandle qObj,  
    sqldouble* weight  
);
```

參數

qObj (in)
查詢物件 的控點。

weight (out)
指向用以設定特性加權之變數的指標。

錯誤碼

qbIECinvalidQueryHandle
您指定的查詢物件控點，所參照的查詢物件無效。

範例

在控點 qoHandle 識別的查詢物件中設定平均色彩特性的比重：

```
#include <dmbqbapi.h>
```

```
weight=2.0
```

```
rc = QbQuerySetFeatureWeight(qoHandle, "QbColorFeatureClass", &weight);
```

QbQueryStringSearch

影像	音效	視訊
X		

搜尋 QBIC 型錄，以找出符合某查詢字串中所含之搜尋準則的影像。搜尋結果 (內含影像控點與其 QBIC 搜尋範圍) 會儲存在從屬站記憶體的结果陣列中。並且會根據其範圍加以排序。

授權

選取

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqqry.lib	libdmbqqry.a (AIX) libdmbqqry.sl (HP-UX) libdmbqqry.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbQueryStringSearch(  
    char *queryString,  
    char *tableName,  
    char *columnName,  
    SQLINTEGER maxReturns,  
    QbQueryScope* scope,  
    SQLINTEGER resultType,  
    SQLINTEGER* count,  
    QbResult* returns  
);
```

參數

- queryString (in)**
查詢字串。
- tableName (in)**
內含您要搜尋之影像直欄的表格名稱。

columnName (in)

影像直欄的名稱。必須為影像資料啟用直欄。

maxReturns (in)

您希望傳回之影像的數目上限。

scope (in) (保留)

必須設定為 0 (NULL)。

resultType (in) (保留)

必須設為 qbiArray。

count (out)

指向所傳回之影像數目的指標。假設傳回 0，請確定您已對查詢字串中的所有特性，編目影像直欄。

returns (out)

指向保有傳回結果之 QbResult 結構陣列的指標。請確定您所配置的緩衝區大小放得下您所要的全部結果。

錯誤碼**qbiEInvalidQueryString**

您指定的查詢字串無效。

範例

查詢員工表格之圖片直欄中的已編目影像。確定所傳回的影像不超過 6 個：

```
#include <dmbqbapi.h>
```

```
rc = QbQueryStringSearch("QbColorFeatureClass color=<255, 0, 0>"
    "employee",
    "picture", 6, 0, qbiArray,
    &count, &returns);
```

QbReCatalogColumn

影像	音效	視訊
X		

爲新特性重新分析已開啓的 QBIC 型錄中的現存影像。會採用特性的預設參數。當您在已含有影像的型錄中新增一項特性時，請使用此 API。

授權

更新、插入

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbReCatalogColumn (  
    QbCatalogHandle cHdl  
);
```

參數

cHdl (in)
指向型錄之控點的指標。

錯誤碼

qbicECInvalidHandle
型錄控點無效。

qbicECInvalidCatalog
對於型錄來說，指定的控點或表格直欄無效。

qbicECCatalog Errors

建立個別影像的型錄時發生錯誤，這些錯誤會被記錄下來。沒有發生 ROLLBACK。

qbicECImageNotFound

該影像找不到或者無法存取。

qbicECCatalogRO

該型錄為唯讀的。

qbicECSQLError

發生 SQL 錯誤。

範例

重新分析已開啓的 QBIC 型錄中現存影像的新特性：

```
#include <dmbqbapi.h>
```

```
rc = QbReCatalogColumn(CatHdl);
```

QbRemoveFeature

影像	音效	視訊
X		

從已開啓的型錄刪除指定特性。

授權

改變

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbRemoveFeature(  
    QbCatalogHandle cHdl,  
    char *featureName  
);
```

參數

- cHdl (in)**
指向型錄之控點的指標。
- featureName (in)**
特性的名稱。

錯誤碼

- qbicECInvalidHandle**
型錄控點無效。
- qbicECCatalogReadOnly**
型錄開啓後僅供讀取。

qbicECFeatureNotFound

型錄中無該項特性。

qbiECInvalidFeatureClass

您指定之特性的名稱格式無效。

範例

在控點 CatHdl 所識別的型錄中，移除 QbColorHistogramFeatureClass 特性：

```
#include <dmbqbapi.h>
```

```
rc=QbRemoveFeature(CatHdl,  
    "QbColorHistogramFeatureClass");
```

QbSetAutoCatalog

影像	音效	視訊
X		

對匯入到影像直欄中的影像，自動進行編目。API 會在特性表格中新增各個影像的相關項目，並分析影像。在 API 分析影像時，其會建立影像資料並將之儲存在特性表格中影像的項目中。

假設您未啓用自動編目特性，當您將影像新增到影像直欄中後，您可以使用 QbCatalogColumn 或 QbCatalogImage API，來編目這些影像。

授權

改變

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbSetAutoCatalog(  
    QbCatalogHandle cHdl  
    SQLINTEGER autoCatalog  
);
```

參數

cHdl (in)
指向型錄之控點的指標。

autoCatalog (in)
指出是否要將新增到影像直欄中的影像，自動新增到特性表格中並加以分析。指定 1，表示啓用自動編目特性，指定 0 則表示停用。

錯誤碼

qbicECInvalidHandle

型錄控點無效。

範例

對控點 CatHdl 所識別的型錄，啓用自動編目特性：

```
#include <dmbqbapi.h>
```

```
rc=QbSetAutoCatalog(CatHdl, 1);
```

QbUncatalogImage

影像	音效	視訊
X		

移除型錄中的某個影像。影像控點必須取自和已開啓之 QBIC 型錄有關的影像直欄中。影像將自己開啓的型錄中移除。影像屬性表格中的相對應橫列，指出影像尚未進行編目。

授權

刪除

檔案庫

OS/2 與 Windows	AIX、HP-UX 及 Solaris
dmbqbapi.lib	libdmbqbapi.a (AIX) libdmbqbapi.sl (HP-UX) libdmbqbapi.so (Solaris)

併入檔

dmbqbapi.h

語法

```
SQLRETURN QbUncatalogImage(  
    QbCatalogHandle cHdl,  
    char *imgHandle  
);
```

參數

- cHdl (in)**
指向型錄之控點的指標。
- imgHandle (in)**
影像的控點。 您可以自使用者表格中擷取到此控點。

錯誤碼

- qbicECInvalidHandle**
型錄控點無效。

qbicECImageNotFound

該影像找不到或者無法存取。

qbicECCatalogRO

該型錄為唯讀的。

範例

在控點 CatHdl 所識別的型錄中，移除 handle Img_hdl 所識別的影像：

```
#include <dmbqbapi.h>
```

```
rc=QbUncatalogImage(CatHdl, Img_hdl);
```

QbUncatalogImage

第 15 章 從屬站的管理命令

本章說明輸入從屬站的 DB2 Extender 管理命令的方法。它也提供從屬站的每個 DB2 Extender 管理命令的參考資訊。

輸入 DB2 Extender 管理命令

您可以在交談式模式中，或在命令模式中向 `db2ext` 命令行處理器提出 DB2 Extender 管理命令。交談式模式以 `db2ext` 為提示字元。在此模式下，您只能輸入 DB2 Extender 管理命令。在命令模式下，您可以在作業系統命令提示字元下輸入 DB2 Extender 命令，以及 DB2 命令和作業系統命令。

請勿在 DB2 命令提示字元下輸入 DB2 Extender 命令。

請依照下列處理以交談式模式啟動 `db2ext` 命令行處理器：

從屬站	動作
AIX、HP-UX、Solaris	在作業系統提示字元上輸入 DB2EXT 命令。
Windows	在 DB2 Extender 資料夾中，按兩下「DB2EXT 命令行處理器」圖示，或在 DB2 命令視窗中輸入 DB2EXT 命令。

輸入 `quit` 或 `terminate` 命令，即可結束交談式模式。`quit` 命令會結束交談式模式，但保持目前與 DB2 的連線。而 `terminate` 命令會結束交談式模式，並捨棄目前與 DB2 的連線。

若要在命令模式中提出 DB2 Extender 命令，請在作業系統命令行上輸入命令。每一個 DB2 Extender 命令都必須以 `db2ext` 為開頭，例如：

```
db2ext enable database for db2image using mydataspace, myindexspace, mylongspace
```

取得 DB2 Extender 命令的線上解說

若要取得所有 DB2 Extender 命令的線上說明，請輸入：

```
db2ext ?
```

ADD QBIC FEATURE

ADD QBIC FEATURE

影像	音效	視訊
X		

建立現行型錄內指定特性的特性表。型錄內現存的影像不會由 Image Extender 自動重新分析。

授權

Alter, Control, SYSADM, DBADM

命令語法

►►—ADD QBIC FEATURE—*feature_name*—◄◄

命令參數

feature_name

欲新增至 QBIC 型錄內的特性名稱。Image Extender 具有下列特性：

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

範例

新增 QbColorFeatureClass 特性至目前開啓的型錄：

```
add qbic feature qbcolorfeatureclass
```

用法注意事項

使用此命令之前，請先連接到資料庫。

型錄必須呈開啓狀態。

CATALOG QBIC COLUMN

影像	音效	視訊
X		

將影像直欄中的影像進行編目，並以特性資料來更新目前開啓的 QBIC 型錄。您可以爲影像直欄中的所有影像更新型錄，或只爲上次分析型錄後所新增的影像更新型錄。

授權

Insert、Control、SYSADM、DBADM

命令語法



命令參數

無。

範例

將新的影像編至現行的型錄，亦即尚未編目的影像。

```
catalog qbic column for new
```

用法注意事項

指定 NEW 時，Image Extender 只會以尚未編目的影像更新型錄。指定 ALL 時，Image Extender 會分析現行型錄的影像直欄中的每一個影像。NEW 是預設值。

使用此命令之前，請先連接到資料庫。

型錄必須呈開啓狀態。

CLOSE QBIC CATALOG

CLOSE QBIC CATALOG

影像	音效	視訊
X		

關閉 QBIC 型錄。

授權

無。

命令語法

►►CLOSE QBIC CATALOG◄◄

命令參數

無。

範例

關閉現行型錄：
close qbic catalog

用法注意事項

QBIC 型錄必須是開啓的。

CONNECT

影像	音效	視訊
X	X	X

連接資料庫。Extender 要單獨連接資料庫，需與 DB2 連接分開。

授權

Connect

命令語法

```

▶▶—CONNECT TO—db_name—[USER—user_ID—][USING—password—]————▶▶
▶▶—CONNECT RESET————▶▶

```

命令參數

db_name

資料庫名稱。

user_ID

授權連接資料庫的使用者 ID。

password

使用者 ID 的通行碼。

RESET

確定任何擱置變更之後，切斷資料庫。

範例

連接 personnel 資料庫。使用者 ID 是 anita，通行碼是 anitapas：

```

connect to personnel user anita
using anitapas

```

用法注意事項

資料庫是以 SHARE 模式來連接。

執行其它 Extender 命令之前，請先執行此命令。

CREATE QBIC CATALOG

CREATE QBIC CATALOG

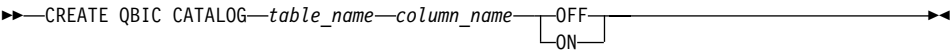
影像	音效	視訊
X		

在現行資料庫中為特定的 DB2IMAGE 直欄建立 QBIC 型錄。Extender 會自動建立型錄名稱。

授權

Alter, Control, SYSADM, DBADM

命令語法



命令參數

table_name
啓用 DB2IMAGE 的表格名稱。

column_name
啓用 DB2IMAGE 的直欄名稱。

OFF 手動編目影像。

ON 自動編目影像。

tablespace_name
「QBIC 型錄」的表格空間規格和索引選項。此規格包含四個部份：

- 型錄表格的表格空間名稱，其中的型錄表格包含特性資料。必須指定表格空間。表格空間應該是區段表格空間。
- 對型錄表格上建立的索引而言，則為使用中區塊、可用區塊、gbpcache 區塊的任意組合，或類型 2 非分割索引的索引選項。此規格為選用性。若不指定此部份，則使用預設值。
- 型錄日誌表的表格空間名稱。表格空間可以是簡式表格空間或區段表格空間。此規格為選用性。若沒有為日誌表格指定表格空間，則會使用指定給特性資料表的表格空間。
- 以日誌資料表上建立的索引而言，類型 2 非分割索引的使用中區塊、可用區塊、gbpcache 區塊或索引選項的任何組合。此規格為選用性。若不指定此部份，則使用預設值。

範例

在員工表格中，建立圖片直欄的 QBIC 型錄，且自動編目設定為 ON：

```
create qbic catalog employee picture on
```

用法注意事項

若指定 ON，則匯入直欄中的影像，會自動在相關的 QBIC 型錄中編目。預設為 OFF。

使用此命令之前，請先連接到資料庫。

DELETE QBIC CATALOG

DELETE QBIC CATALOG

影像	音效	視訊
X		

刪除 QBIC 型錄，包括所有的 QBIC-search 支援資料。

授權

Alter, Control, SYSADM, DBADM

命令語法

►►—DELETE QBIC CATALOG—*table_name*—*column_name*—————►◄

命令參數

- table_name**
啓用 DB2IMAGE 的表格名稱。
- column_name**
啓用 DB2IMAGE 的直欄名稱。

範例

在員工表格中，刪除與圖片直欄相關的型錄：
delete qbic catalog employee picture

用法注意事項

使用此命令之前，請先連接到資料庫。

DISABLE COLUMN

影像	音效	視訊
X	X	X

使特定直欄無法儲存指定的媒體資料。

授權

SYSADM、DBADM、Control、Alter

命令語法

►►—DISABLE COLUMN—*table_name*—*col_name*—FOR—*extender_name*—————►►

命令參數

table_name

現行資料庫中的表格名稱。

col_name

欲停用的直欄名稱。

extender_name

欲停用直欄的 Extender 名稱。有效的 Extender 名稱包括 db2image、db2audio 及 db2video。

範例

停用員工表格中的 photo 直欄，使之不能保留影像資料：

disable column employee photo for db2image

用法注意事項

使用此命令之前，請先連接到資料庫。

停用直欄時：

- 直欄無法儲存特定 Extender 的資料。這不會影響表格中的其它直欄是否能夠接受多媒體資料類型。
- 直欄項目的內容會設為 NULL，且管理表格中相對應的橫列會被刪除。
- 與直欄相關的觸發函式會被捨棄。

DISABLE DATABASE

DISABLE DATABASE

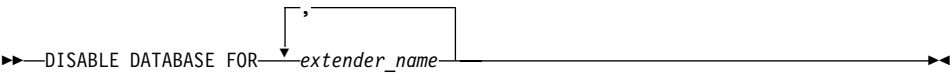
影像	音效	視訊
X	X	X

停用現行資料庫，使其無法儲存媒體資料。

授權

SYSADM、DBADM

命令語法



命令參數

extender_name
欲停用現行資料庫的 Extender 名稱。有效的 Extender 名稱包括 db2image、db2audio 及 db2video。

範例

停用現行資料庫，使其無法保留影像資料：
`disable database for db2image`

用法注意事項

使用此命令之前，請先連接到資料庫。

停用資料庫時，系統會：

- 停用為特定 Extender 啓用的所有表格。
- 捨棄特定 Extender 的 UDF 管理支援表格。

DISABLE TABLE

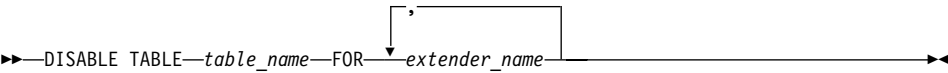
影像	音效	視訊
X	X	X

停用特定表格，使其無法儲存媒體資料。

授權

SYSADM、DBADM、Control、Alter

命令語法



命令參數

table_name
您要在現行資料庫停用的表格名稱。

extender_name
停用表格的 Extender 名稱。有效的 Extender 名稱包括 db2image、db2audio 及 db2video。

範例

停用員工表格，使其無法保留影像資料：

```
disable table employee for db2image
```

用法注意事項

使用此命令之前，請先連接到資料庫。

停用表格時，系統會：

- 停用表格內為特定 Extender 啟用的所有直欄。
- 捨棄與此表格相關的管理支援表格。

DISCONNECT SERVER AT NODENUM

DISCONNECT SERVER AT NODENUM (僅適用於 EEE)

影像	音效	視訊
X	X	X

切斷所有資料庫上伺服器與特定節點的連線。

授權

SYSADM、SYSCTRL、SYSMAINT、DBADM

命令語法

►►DISCONNECT SERVER AT NODENUM—*node_number*————►◄

命令參數

node_number
欲切斷與伺服器連線的節點。

範例

切斷伺服器與節點號碼 2 上所有資料庫的連線：
disconnect server at nodenum 2

用法注意事項

欲切斷伺服器與所有節點上的所有資料庫的連線，請使用 DMBSTOP 命令。

DISCONNECT SERVER FOR DATABASE (僅適用於 EEE)

影像	音效	視訊
X	X	X

切斷特定資料庫上的伺服器與所有節點之間的連線。

授權

SYSADM、SYSCTRL、SYSMAINT、DBADM

命令語法

▶▶—DISCONNECT SERVER FOR DATABASE—*database_name*————▶▶

命令參數

database_name
欲切斷與伺服器連線的資料庫。

範例

切斷伺服器與稱為 MY_DATABASE 資料庫之間的連線：
disconnect server for database my_database

用法注意事項

欲切斷伺服器與所有節點上的所有資料庫的連線，請使用 DMBSTOP 命令。

DISCONNECT SERVER FOR DATABASE AT NODENUM

DISCONNECT SERVER FOR DATABASE AT NODENUM (僅適用於 EEE)

影像	音效	視訊
X	X	X

切斷伺服器與指定節點上的特定資料庫之間的連線。

授權

SYSADM、SYSCTRL、SYSMAINT、DBADM

命令語法

►►DISCONNECT SERVER FOR DATABASE—*database_name*—AT NODENUM—*node_number*—►◄

命令參數

database_name
欲切斷與伺服器連線的資料庫。

node_number
欲切斷與伺服器連線的節點。

範例

切斷伺服器與節點號碼 2 上稱為 MY_DATABASE 資料庫之間的連線：
disconnect server for database my_database at nodenum 2

用法注意事項

欲切斷伺服器與所有節點上的所有資料庫的連線，請使用 DMBSTOP 命令。

ENABLE COLUMN

影像	音效	視訊
X	X	X

啓用指定的直欄，使其能夠儲存媒體資料。

授權

SYSADM、DBADM、Control、Alter

命令語法

```
▶▶—ENABLE COLUMN—table_name—col_name—FOR—extender_name————▶▶
```

命令參數

table_name
現行資料庫中的表格名稱。

col_name
欲啓用的直欄名稱。

extender_name
欲啓用表格的 Extender 名稱。有效的 Extender 名稱包括 db2image、db2audio 及 db2video。

範例

啓用員工表格中的 photo 直欄，使其可以保留影像資料：
enable column employee photo for db2image

用法注意事項

使用此命令之前，請先連接到資料庫。

ENABLE DATABASE

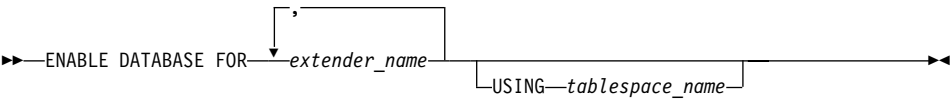
影像	音效	視訊
X	X	X

啓用現行資料庫，來儲存使用指定表格空間的媒體資料。

授權

SYSADM、SYSCTRL、DBADM

命令語法



命令參數

extender_name

欲啓用現行資料庫的 Extender 名稱。有效的 Extender 名稱包括 db2image、db2audio 及 db2video。

tablespace_name

表格空間的名稱，用來儲存管理表格的配置區集成。表格空間名稱有下列三個部份：datats、indexts、longts，其中 datats 是用來建立 meta 資料表格的表格空間名稱，indexts 是在 meta 資料表格上建立索引的表格空間名稱，longts 是 meta 資料表格中儲存 long 直欄值 (例如包含 LONG VARCHAR 和 LOB 資料類型的值) 的表格空間名稱。若在表格空間名稱的任一部份提供 NULL 值，則會使用該部份預設表格空間的名稱。指定的表格空間，應該在含有已分割資料庫系統中所有節點的節點群組上定義。

範例

啓用現行資料庫，以保留影像資料：

```
enable database for db2image using mydataspace, myindxspace, mylongspace
```

用法注意事項

使用此命令之前，請先連接到資料庫。

若未指定表格空間，則系統以 USERSPACE1 表格空間供管理表格使用。

ENABLE TABLE

啓用員工表格，以保留影像資料。 採用預設的表格空間：

```
enable table employee for db2image
```

用法注意事項

使用此命令之前，請先連接到資料庫。

若未指定表格空間，則系統使用現行資料庫被啓用時所定義的表格空間。

GET EXTENDER STATUS

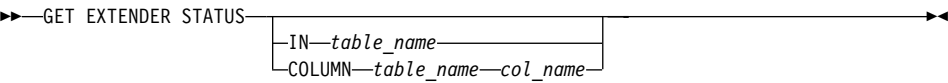
影像	音效	視訊
X	X	X

如果有的話，顯示已啓用直欄、表格或現行資料庫的 Extender 名稱。

授權

無

命令語法



命令參數

table_name
現行資料庫中的表格名稱。

col_name
直欄的名稱。

範例

顯示資料庫中已啓用的 Extender 名稱：

```
get extender status
```

顯示員工表格的狀態：

```
get extender status in employee
```

顯示員工表格中 ADDRESS 直欄的狀態：

```
get extender status column employee address
```

用法注意事項

使用此命令之前，請先連接到資料庫。

GET INACCESSIBLE FILES

影像	音效	視訊
X	X	X

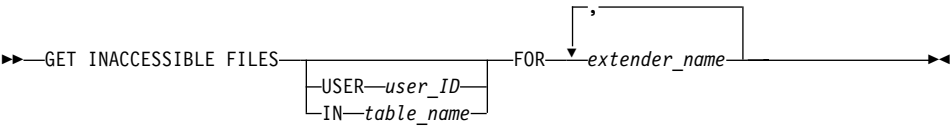
列出現行資料庫中，所有無法存取且為某個表格、含指定限定元的表格、或所有表格所參照的媒體檔。

授權

對現行資料庫中的所有表格，亦即未指定 `USER` 或 `IN`：使用 `SYSADM`、`SYSCTRL`、`SYSMAINT`、`DBADM`

對於特定表格（指定 `IN`）或屬於某限定元的表格（指定 `USER`）：使用 `SELECT`

命令語法



命令參數

user_ID
現行資料庫中的表格限定元，您想列出其中無法存取的檔案。

table_name
現行資料庫中的表格名稱，您想列出其中無法存取的檔案。

extender_name
Extender 的名稱。有效的 Extender 名稱包括 `db2image`、`db2audio` 及 `db2video`。

範例

列出為資料庫中的表格所參照，但是無法存取的影像檔：

```
get inaccessible files
  for db2image
```

列出表格中以限定元 `anita` 來參照，但無法存取的所有影像檔。

```
get inaccessible files
  user anita for db2image
```

列出員工表格項目所參照，但無法存取的影像檔。

```
get inaccessible files  
in employee FOR db2image
```

用法注意事項

使用此命令之前，請先連接到資料庫。

若指定表格，則命令會列出該表格無法存取的檔案。若指定限定元，則命令只會列出該限定元相關表格無法存取的檔案。若兩者都沒有指定，則此命令會列出現行資料庫的所有表格中無法存取的檔案。

GET QBIC CATALOG INFO

影像	音效	視訊
X		

- 傳回下列有關目前所開啓型錄的資訊：
- 與型錄相關的使用者表格和影像直欄之名稱。
 - 型錄中的特性名稱。
 - 型錄中的特性數目。
 - 自動分析是否開啓。

授權

Select、Control、SYSADM、DBADM

命令語法

▶▶—GET QBIC CATALOG INFO—◀◀

命令參數

無。

範例

取得有關目前所開啓 QBIC 型錄的資訊：
get qbic catalog info

用法注意事項

- 使用此命令之前，請先連接到資料庫。
- 型錄必須呈開啓狀態。

GET REFERENCED FILES

影像	音效	視訊
X	X	X

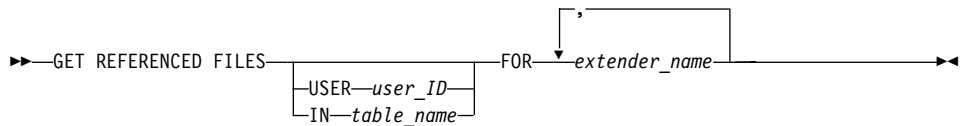
列出現行資料庫中，所有含指定限定元的表格，或所有表格所參照的媒體檔和直欄名稱。

授權

就現行資料庫中的所有表格而言，若未指定 `USER` 或 `IN`：即為 `SYSADM`、`SYSCTRL`、`SYSMAINT`、`DBADM`

對於特定表格 (指定 `IN`) 或屬於某限定元的表格 (指定 `USER`)：使用 `SELECT`

命令語法



命令參數

user_ID

現行資料庫中的表格的限定元，您想列出其中的參考檔。該命令只會搜尋具有該限定元的表格。

table_name

現行資料庫中表格的名稱，其參照的檔案是您想列出的。該命令只會搜尋該表格。

extender_name

Extender 的名稱。有效的 Extender 名稱包括 `db2image`、`db2audio` 及 `db2video`。

範例

列出資料庫中的所有表格登錄所參照的影像檔：

```
get referenced files
  for db2image
```

列出表格項目的限定元為 `anita` 所參照之所有影像檔：

GET REFERENCED FILES

```
get referenced files  
  user anita for db2image
```

列出員工表格項目所參照的所有影像檔：

```
get referenced files  
  in employee for db2image
```

用法注意事項

使用此命令之前，請先連接到資料庫。

若沒有指定參數，則此命令會搜尋資料庫中的所有表格。

GET SERVER STATUS

影像	音效	視訊
X	X	X

顯示現行資料庫或所有資料庫中的 Extender 伺服器狀態。

僅適用於 EEE 若指定節點，則該命令僅會為現行資料庫或所有資料庫，顯示該節點的 Extender 伺服器狀態。

授權

無

命令語法

▶▶ GET SERVER STATUS [ALL] [NODENUM *node_number*] ◀◀

命令參數

ALL 顯示所有資料庫的狀態。

node_number
節點的數目。該命令會顯示此節點的狀態。 **(僅適用於 EEE)**

範例

對現行資料庫顯示 Extender 伺服器的狀態：

```
get server status
```

對所有資料庫顯示 Extender 伺服器的狀態：

```
get server status all
```

對所有資料庫，顯示節點號碼 2 的 Extender 伺服器狀態：

```
get server status all nodenum 2
```

用法注意事項

使用此命令之前，請先連接到資料庫。

若未指定任何參數，則該命令會對現行資料庫，顯示 db2nodes.cfg 檔案中所有節點的狀態。

OPEN QBIC CATALOG

影像	音效	視訊
X		

開啓指定 DB2IMAGE 直欄的型錄。資料庫會嘗試以更新模式來開啓型錄。若型錄已經是更新模式，則以真實模式來開啓。

授權

Connect

命令語法

►►—OPEN QBIC CATALOG—*table_name*—*column_name*—————►◄

命令參數

table_name
啓用 DB2IMAGE 的表格名稱。

column_name
啓用 DB2IMAGE 的直欄名稱。

範例

在員工表格中，開啓圖片直欄的 QBIC 型錄：
open qbic catalog employee picture

用法注意事項

- 使用此命令之前，請先連接到資料庫。
- 此命令會關閉任何已開啓的型錄。

QUIT

影像	音效	視訊
X	X	X

關閉交談式模式中命令輸入所用的 db2ext 命令行處理器。仍維持與 DB2 的連線，所以您可以在命令模式下，繼續提出命令給 db2ext 命令行處理器。

授權

無

命令語法

▶▶—QUIT—◀◀

命令參數

無。

範例

關閉交談式模式的命令行介面：

quit

用法注意事項

QUIT 維持與資料庫的連線。

RECONNECT SERVER AT NODENUM

RECONNECT SERVER AT NODENUM (僅適用於 EEE)

影像	音效	視訊
X	X	X

為所有資料庫重新連接伺服器與特定節點。

授權

SYSADM、SYSCTRL、SYSMAINT、DBADM

命令語法

➡➡RECONNECT SERVER AT NODENUM—*node_number*————➡➡

命令參數

node_number
欲重新與伺服器連接的節點。

範例

重新連接伺服器與節點號碼 2 上的所有資料庫：
reconnect server at nodenum 2

用法注意事項

欲重新連接伺服器與所有節點上的所有資料庫，請使用 DMBSTART 命令。

RECONNECT SERVER FOR DATABASE (僅適用於 EEE)

影像	音效	視訊
X	X	X

重新連接伺服器與指定資料庫的所有節點。

授權

SYSADM、SYSCTRL、SYSMAINT、DBADM

命令語法

►►—RECONNECT SERVER FOR DATABASE—*database_name*—————►◄

命令參數

database_name

欲重新與伺服器連接的資料庫。

範例

重新連接伺服器與稱為 MY_DATABASE 的資料庫：

```
disconnect server for database my_database
```

用法注意事項

欲重新連接伺服器與全部節點上的全部資料庫，請使用 DMBSTART 命令。

RECONNECT SERVER FOR DATABASE AT NODENUM (僅適用於 EEE)

影像	音效	視訊
X	X	X

重接連接伺服器與指定節點上的指定資料庫。

授權

SYSADM、SYSCTRL、SYSMAINT、DBADM

命令語法

►►RECONNECT SERVER FOR DATABASE—*database_name*—AT NODENUM—*node_number*—►◄

命令參數

database_name
欲重新與伺服器連接的資料庫。

node_number
欲重新與伺服器連接的節點。

範例

重新連接伺服器與節點號碼 2 上稱為 MY-DATABASE 的資料庫：
reconnect server for database my_database at nodenum 2

用法注意事項

欲重新連接伺服器與全部節點上的全部資料庫，請使用 DMBSTART 命令。

REDISTRIBUTE NODEGROUP (僅適用於 EEE)

影像	音效	視訊
X		

當節點群組上新增或移除節點，或節點群組上建立新的分割對映表時，重新分配 Extender 資料。

授權

SYSADM、DBADM

命令語法



命令參數

nodegroup (節點群組)
欲重新分配的節點群組名稱。

CONTINUE
若重新分配程序傳回錯誤，則您可以根據命令回應的指示，決定是否使用 CONTINUE 參數來重新執行命令。此選項指示系統從停止處繼續執行，而非從頭開始執行。

範例

重新分配稱為 my_nodegroup 的節點群組：
redistribute nodegroup my_nodegroup

用法注意事項

- 使用此命令之前，請先連接到資料庫。
- 在第一次執行 REDISTRIBUTE NODEGROUP 命令時，在 DB2 的 REDISTRIBUTE 命令後面不應該使用 CONTINUE 命令。若第一次即使用，則會記載錯誤，且重新分配會重頭開始。
- 欲維護資料完整性，必須一次分配一個節點群組。請先等待一個節點群組完成重新分配後，再開始另一個節點群組的分配。

REDISTRIBUTE NODEGROUP

若 REDISTRIBUTE NODEGROUP 失敗，請參照下列其中一個目錄的 "redist.log" 檔案，內含詳細的說明：

- **Unix:** /<home-instance>/dmb/redist
- **Windows:** \\<instance_owning_machine>\DB2<instance_name>\<instance_name>\dmb\redist

REMOVE QBIC FEATURE

影像	音效	視訊
X		

從開啓的型錄中，刪除指定特性的特性表格。

授權

Alter, Control, SYSADM, DBADM

命令語法

►►—REMOVE QBIC FEATURE—*feature_name*—————►◄

命令參數

feature_name
欲從 QBIC 型錄內移除的特性名稱。Image Extender 具有下列特性：

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

範例

從目前開啓的型錄內移除 QbColorFeatureClass 特性：

```
remove qbic feature qbcolorfeatureclass
```

用法注意事項

使用此命令之前，請先連接到資料庫。

型錄必須呈開啓狀態。

REORG

影像	音效	視訊
X	X	X

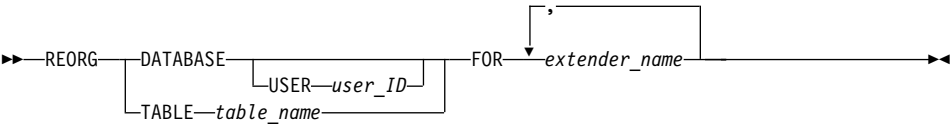
清除與指定表格、指定限定元的表格、或現行資料庫中所有表格有關的管理表格 (管理表格和屬性表格)。

授權

指定的表格 (執行 REORG TABLE) 或特定限定元的表格 (執行 REORG DATABASE)：使用 SYSADM、SYSCTRL、SYSMAINT、DBADM、Control

資料庫中的所有表格 (執行 REORG DATABASE)：使用 SYSADM、SYSCTRL、SYSMAINT、DBADM

命令語法



命令參數

user_ID
表格的限定元。

table_name
現行資料庫中表格的名稱，而其管理表格是您要清除的。

extender_name
Extender 的名稱。有效的 Extender 名稱包括 db2image、db2audio 及 db2video。

範例

重組和清除現行資料庫的影像管理表格。
reorg database for db2image

重組和清除現行表格中含有限定元 anita 的影像管理表格。
reorg database user anita for db2image

重組和清除員工表格的影像管理表格。

```
reorg table employee for db2image
```

用法注意事項

使用此命令之前，請先連接到資料庫。

SET QBIC AUTOCATALOG

SET QBIC AUTOCATALOG

影像	音效	視訊
X		

在影像匯入至直欄時自動將影像編目。影像會新增至與該直欄相關的 QBIC 型錄內。

授權

Alter, Control, SYSADM, DBADM

命令語法



命令參數

無。

範例

啓用自動編目特性：
set qbic autocatalog on

用法注意事項

QBIC 型錄必須是開啓的。

START SERVER (僅適用於非 EEE)

影像	音效	視訊
X	X	X

啓動現行資料庫的 Extender 伺服器。

授權

SYSADM、SYSCTRL、SYSMAINT、DBADM

命令語法

▶▶—START SERVER——▶▶

命令參數

無。

範例

啓動現行資料庫的 Extender 伺服器：

```
start server
```

用法注意事項

使用此命令之前，請先連接到資料庫。

STOP SERVER

STOP SERVER (僅適用於非 EEE)

影像	音效	視訊
X	X	X

停止現行資料庫的 Extender 伺服器。

授權

SYSADM、SYSCTRL、SYSMAINT、DBADM

命令語法

▶▶STOP SERVER————▶◀

命令參數

無。

範例

停止現行資料庫的 Extender 伺服器：

```
stop server
```

用法注意事項

使用此命令之前，請先連接到資料庫。

TERMINATE

影像	音效	視訊
X	X	X

關閉 db2ext 命令行處理器，並捨棄與 DB2 的連線。

授權

無

命令語法

►►—TERMINATE—►►

命令參數

無。

範例

關閉 db2ext 命令行處理器：

```
quit
```

用法注意事項

TERMINATE 會捨棄與資料庫的連接。

TERMINATE

第 16 章 診斷資訊

在您程式中的所有內含的 SQL 及 DB2 CLI 呼叫 (包括呼叫 DB2 Extender UDF 者) 會建立字碼，指出是否已順利執行 內含的 SQL 陳述式或 DB2 CLI 呼叫。其它 DB2 Extender API，例如管理的 API，也有回覆碼來指出成功與否。您的程式應檢查並回應這些回覆碼。

您的程式也可擷取補充這些回覆碼的資訊。它包括 SQLSTATE 資訊和錯誤訊息。您可使用此診斷資訊來找出和修正程式中的問題。

有時無法輕易地診斷出問題的來源。在這些情況下，您可能需要對服務人員提供資訊以找出並解決問題。DB2 Extender 含有記錄 Extender 活動的追蹤機能。追蹤資訊對服務人員來說可能是相當有價值的資訊。請遵照 IBM 服務人員的指示使用追蹤機能。

本章說明如何存取此診斷資訊。內容包括：

- 如何處理 DB2 Extender UDF 回覆碼及 API 回覆碼。
- 如何控制追蹤

它也列示及說明 Extender 可能會傳回的 SQLSTATE 及錯誤訊息。

處理 UDF 回覆碼

內含的 SQL 陳述式傳回 SQLCA 結構中 SQLCODE、SQLWARN、及 SQLSTATE 欄位內的訊息碼。此結構定義於 SQLCA 併入檔中。(關於 SQLCA 結構和 SQLCA include 檔案的詳細資訊，請參閱 *DB2 Application Development Guide*。)

DB2 CLI 呼叫傳回 SQLCODE 和 SQLSTATE 值，您可使用 SQLError 函數來擷取。(關於使用 SQLError 函數擷取錯誤資訊的其他資訊，請參閱 *CLI Guide and Reference*。)

SQLCODE 值 0 表示陳述式已順利執行 (可能附帶警告狀況)。SQLCODE 正值表示順利執行陳述式但帶有警告。(內含的 SQL 陳述式所傳回的警告會附隨於 SQLWARN 欄位中的 0 或正 SQLCODE 值。) SQLCODE 負值表示發生錯誤狀況。

DB2 在每一個 SQLCODE 值上結合一則訊息。如果 DB2 Extender UDF 發現警告或錯誤狀況，它會將包含在 SQLCODE 訊息中的相關資訊傳遞到 DB2。

處理 UDF 回覆碼

SQLSTATE 值包含補充 SQLCODE 訊息的字碼。請參閱『SQLSTATE 訊息碼』，以取得 DB2 Extender 所傳回之各 SQLSTATE 碼的說明。

呼叫 DB2 Extender UDF 的內含的 SQL 陳述式及 DB2 CLI 呼叫但 DB2 會以與傳回內含的 SQL 陳述式 或其它 DB2 CLI 呼叫的相同方式傳回這些值。因此，您存取這些值的方法與存取不啟動 DB2 Extender UDF 的內含 SQL 陳述式或 DB2 CLI 呼叫的方法一樣。

請參閱『SQLSTATE 訊息碼』，以取得 Extender 可能會傳回之相關訊息的 SQLSTATE 值及訊息號碼。關於每一則訊息的資訊，請參閱第 515 頁的『訊息』。

處理 API 回覆碼

每一個 DB2 Extender API 呼叫都會傳回一個回覆碼。回覆碼 0 表示順利處理 API 呼叫。非 0 的回覆碼表示已順利處理 API 呼叫，但發現警告狀況，或由於錯誤狀況而無法順利處理。

第 251 頁的第 14 章，『應用程式設計介面』會列出回覆碼的符號值，並說明 DB2 Extender API 會傳回的每一個回覆碼。

您可以擷取有關 API 所發現之錯誤的附加資訊。使用 DBxGetError API 來擷取此附加資訊，其中 *x* 表示 Audio Extender，*i* 表示 Image Extender，*v* 表示 Video Extender。DBxGetError API 會傳回前次發現錯誤之 DB2 Extender API 的 SQL 錯誤碼及相關訊息。關於 SQL 錯誤碼的資訊，請參閱 DB2 訊息參照。請參閱第 515 頁的『訊息』，以取得有關 DBxGetError API 會傳回之各個訊息的資訊。

例如，下列在 C 應用程式中的陳述式會對 Audio Extender 啟用表格，然後檢查錯誤。

```
rc=DBaEnableTable((char *)NULL, "employee");

rc=DBaGetError(&errCode, &errMsg);
```

SQLSTATE 訊息碼

表 16 列示並說明 DB2 Extender 可能會傳回的 SQLSTATE 值。每一個 SQLSTATE 值的說明包含其象徵性意義。表格也會列出與各 SQLSTATE 值相關的訊息號碼。關於每一則訊息的資訊，請參閱第 515 頁的『訊息』。

表 16. SQLSTATE 碼及相關訊息碼

SQLSTATE	訊息碼	說明
00000		MMDB_SQLSTATE_OK 成功

表 16. *SQLSTATE* 碼及相關訊息碼 (繼續)

SQLSTATE	訊息碼	說明
01H01	DMB0211W	MMDB_SQLSTATE_WARN_NO_OVERWRITE 未進行檔案改寫
38A00	DMB0101E	MMDB_SQLSTATE_AUDIO_NULL_PARM 對 UDF 的輸入參數不可為 NULL
38A02	DMB0209E	MMDB_SQLSTATE_AUDIO_OPEN_HDR_ERROR 開啓音效檔標頭時發生錯誤
38A03	DMB0209E	MMDB_SQLSTATE_AUDIO_BAD_WAVE_HDR 提供的 wave 檔無效
38V00	DMB0101E	MMDB_SQLSTATE_VIDEO_NULL_PARM 對 UDF 的輸入參數不可為 NULL
38V02	DMB0051E	MMDB_SQLSTATE_VIDEO_OPEN_HDR_ERROR 開啓影像檔標頭時發生錯誤
38V03	DMB0105E	MMDB_SQLSTATE_VIDEO_BAD_MPEG1_HDR 提供的 mpeg1 檔無效
38V04	DMB0104E	MMDB_SQLSTATE_VIDEO_BLOB_TOO_SHORT 提供的影像緩衝區太小
38V05	DMB0106E	MMDB_SQLSTATE_VIDEO_BAD_AVI_HDR 提供的 AVI 檔無效
38V07	DMB0106E	MMDB_SQLSTATE_VIDEO_BAD_QT_HDR 提供的 Quicktime 檔無效
38600	DMB0075E DMB0101E DMB0102E DMB0103E DMB0210E	MMDB_SQLSTATE_INVALID_INPUT 對 UDF 的輸入參數無效
38601	DMB0009E	MMDB_SQLSTATE_MALLOC_FAIL 記憶體配置失敗
38602	DMB0386E	MMDB_SQLSTATE_CANNOT_COLLOCATE 無法配置使用者資料
38603	DMB0077E	MMDB_SQLSTATE_READ_FILE_FAIL 無法讀取檔案
38604	DMB0080E	MMDB_SQLSTATE_WRITE_FILE_FAIL 無法寫入檔案
38610	DMB0070E	MMDB_SQLSTATE_INVALID_HANDLE 媒體直欄包含無效的資料

SQLSTATE

表 16. SQLSTATE 碼及相關訊息碼 (繼續)

SQLSTATE	訊息碼	說明
38611	DMB0073E	MMDB_SQLSTATE_INVALID_SESSION_HANDLE UDF 階段作業控點無效
38612	DMB0074E	MMDB_SQLSTATE_INVALID_STATEMENT_HANDLE UDF 陳述式控點無效
38613	DMB0083E	MMDB_SQLSTATE_INVALID_IMPORT_REQUEST 匯入的要求無效
38615	DMB0071E	MMDB_SQLSTATE_CONNECT_FAIL 連接資料庫時發生錯誤
38617	DMB0071E	MMDB_SQLSTATE_ALLOC_STMT_FAIL 配置新的陳述式控點時發生錯誤
38618	DMB0208E DMB0138E	MMDB_SQLSTATE_FREE_STMT_FAIL 在解除陳述式時發生錯誤
38619	DMB0208E DMB0132E	MMDB_SQLSTATE_BIND_FAIL 連結時發生錯誤
38620	DMB0208E	MMDB_SQLSTATE_BIND_COLUMN_FAIL 連結直欄時發生錯誤
38621	DMB0208E	MMDB_SQLSTATE_BIND_FILE_FAIL 連結檔案時發生錯誤
38622	DMB0208E DMB0132E	MMDB_SQLSTATE_SET_PARAM_FAIL 設定參數時發生錯誤
38623	DMB0208E DMB0131E	MMDB_SQLSTATE_PREPARE_FAIL 準備 SQL 陳述式時發生錯誤
38624	DMB0208E DMB0133E DMB0172E	MMDB_SQLSTATE_EXECUTE_FAIL 執行陳述式時發生錯誤
38625	DMB0208E DMB0133E	MMDB_SQLSTATE_EXEC_DIRECT_FAIL 直接在 UDF 中執行 SQL 陳述式時發生錯誤
38626	DMB0208E DMB0133E	MMDB_SQLSTATE_FETCH_FAIL 擷取下一列資料時發生錯誤
38627	DMB0208E	MMDB_SQLSTATE_COMMIT_FAIL 確定異動時發生錯誤
38628	DMB0208E	MMDB_SQLSTATE_GET_LENGTH_FAIL 擷取字串值的長度時發生錯誤
38629	DMB0208E	MMDB_SQLSTATE_GET_SUBSTRING_FAIL 擷取字串值的一部份時發生錯誤

表 16. *SQLSTATE* 碼及相關訊息碼 (繼續)

SQLSTATE	訊息碼	說明
38650	DMB0077E DMB0079E	MMDB_SQLSTATE_COPY_BLOB_2_FILE_FAIL 將 BLOB 複製到檔案時發生錯誤
38651	DMB0086E	MMDB_SQLSTATE_BLOB_BUFFER_TOO_SMALL 提供的緩衝區太小
38652	DMB0082E	MMDB_SQLSTATE_BUILD_HANDLE 建構媒體直欄資料時發生錯誤
38653	DMB0083E	MMDB_SQLSTATE_INVALID_INSERT_VIA_SELECT 透過選取進行插入的要求無效
38654	DMB0081E	MMDB_SQLSTATE_INVALID_OFFSET_SIZE 偏移大小無效
38655	DMB0068E	MMDB_SQLSTATE_METATABLE_DOESNOT_EXIST 所需的 meta 資料表不存在
38670	DMB0134E DMB0103E	MMDB_SQLSTATE_UNKNOWN_FORMAT 儲存之媒體的格式不明
38671	DMB0135E	MMDB_SQLSTATE_CREATE_THUMBNAIL_FAIL 建立縮圖時發生錯誤
38672	DMB0114E	MMDB_SQLSTATE_FORMAT_CONVERSION_FAIL 轉換檔案格式時發生錯誤
38673	DMB0363E	MMDB_SQLSTATE_INVALID_UPDATE 在未參照表格下呼叫更新 UDF 時 發生錯誤
38674	DMB0361E	MMDB_SQLSTATE_NOT_ENABLED 引用某匯入 UDF 的直欄不是 為該 Extender 啓用的直欄， 因而發生錯誤
38675	DMB0129E	MMDB_SQLSTATE_VIDEO_INTERNAL 在 Video Extender UDF 中發生內部錯誤
38676	DMB0129E	MMDB_SQLSTATE_AUDIO_INTERNAL 在 Audio Extender UDF 中發生內部錯誤
38677	DMB0129E	MMDB_SQLSTATE_IMAGE_INTERNAL
38678	DMB0089E DMB0208E	MMDB_SQLSTATE_BASE_INTERNAL_ERROR 在共用層中發生內部錯誤
38681	DMB0108E	MMDB_SQLSTATE_IMPORT_ENV_NOT_SETUP 匯入的環境變數設定不正確
38682	DMB0111E	MMDB_SQLSTATE_STORE_ENV_NOT_SETUP 儲存作業的環境變數設定不正確

SQLSTATE

表 16. SQLSTATE 碼及相關訊息碼 (繼續)

SQLSTATE	訊息碼	說明
38683	DMB0107E	MMDB_SQLSTATE_EXPORT_ENV_NOT_SETUP 匯出作業的環境變數設定不正確
38684	DMB0117E	MMDB_SQLSTATE_TEMP_ENV_NOT_SETUP 建立暫用檔之環境變數設定不正確
38686	DMB0109E	MMDB_SQLSTATE_CANT_RESOLVE_IMPORT_FILE 解析匯入檔名稱時發生錯誤
38687	DMB0112E	MMDB_SQLSTATE_CANT_RESOLVE_STORE_FILE 解析儲存檔名稱時發生錯誤
38688	DMB0110E	MMDB_SQLSTATE_CANT_RESOLVE_EXPORT_FILE 解析匯出檔名稱時發生錯誤
38689	DMB0116E	MMDB_SQLSTATE_CANT_CREATE_TMP_FILE 建立暫用檔時發生錯誤
38690	DMB0076E	MMDB_SQLSTATE_OPEN_IMPORT_FILE_FAIL 無法開啓匯入檔
38691	DMB0115E	MMDB_SQLSTATE_OPEN_STORE_FILE_FAIL 無法開啓匯入檔
38692	DMB0114E	MMDB_SQLSTATE_OPEN_EXPORT_FILE_FAIL 無法開啓匯出檔
38693	DMB0118E	MMDB_SQLSTATE_OPEN_TEMP_FILE_FAIL 無法開啓暫存檔
38694	DMB0117E	MMDB_SQLSTATE_OPEN_CONTENT_FILE_FAIL 無法開啓內容檔
38695	DMB0135E	MMDB_SQLSTATE_OPEN_THUMBNAIL_FILE_FAIL 無法開啓縮圖檔
38696	DMB0135E	MMDB_SQLSTATE_READ_THUMBNAIL_FILE_FAIL 無法讀取縮圖檔
38697	DMB0207E	MMDB_SQLSTATE_OVERWRITE_NOT_ALLOWED 無法執行改寫作業
38699	DMB0171E	MMDB_SQLSTATE_QUERY_NAME_NOT_FOUND 找不到具有該名稱的查詢
38700		MMDB_SQLSTATE_NO_MANAGEBLOB
38701		MMDB_SQLSTATE_UDFLOCATOR_FAIL
38702		MMDB_SQLSTATE_SQL_FAIL
38703		MMDB_SQLSTATE_INVALID_UPDATE

表 16. *SQLSTATE* 碼及相關訊息碼 (繼續)

SQLSTATE	訊息碼	說明
38704		MMDB_SQLSTATE_NOT_ENABLED
38705	DMB0366E DMB0382E	MMDB_SQLSTATE_QBIC_QUERY_FAIL_TO_BUILD 建置查詢時失敗
38706	DMB0205E	MMDB_SQLSTATE_QBIC_TABLE_COLUMN_PAIR_NOT_VALID 嘗試存取 QBIC 型錄時失敗。 可能型錄中找不到影像控點， 或表格名稱和直欄名稱組合沒有型錄。
38707	DMB0383E	MMDB_SQLSTATE_QBIC_QUERY_EXECUTE_FAILED 執行查詢時失敗
38708		MMDB_SQLSTATE_QBIC_UNKNOWN_ERROR QBIC 發生不明失敗
38709	DMB0208E	MMDB_COPY_FILE_TO_LOCATOR_FAILURE 將檔案複製到 LOB 定位器時發生失敗
38710	DMB0534E	MMDB_SQLSTATE_QBIC_UNSUPPORTED_UDF 不支援 UDF。

訊息

DMB0001E 未連接 DB2 Extender 伺服器。原因："`<code>`"。

原因： 必須執行 DB2 Extender 服務程式才能進行所嘗試的作業。

動作： 在伺服器上，於作業系統的命令行上執行 DMBSTART 命令。

DMB0003W 此階段作業正在執行 DB2 Extender 追蹤機能。

原因： 此追蹤機能用盡了系統資源。

動作： 如果系統的效能受到影響，則應關閉追蹤。

DMB0004I 只有下列的案例擁有者可執行此程式："`<name>`"。

原因： 必須以當初建立案例的使用者 ID 來啟動 DB2 Extender 伺服器。

動作： 以當初建立案例的使用者 ID 來執行 DMBSTART 命令。

DMB0005E 未對 "`<extender-name>`" Extender 啓用現行資料庫。

原因： 嘗試了需要對某個特定 DB2 Extender 啓用資料庫的作業。例如，如果您要對 DB2IMAGE 資料啓用表格，您必須先啓用其表格儲存 DB2IMAGE 資料的資料庫。

動作： 請對您要的 Extender 資料類型啓用資料庫，然後重試。

DMB0006E 使用者 "`<name>`" 未經授權呼叫此 API。

原因： 某個並無對該 API 所需的權限層次之使用者 ID 試圖呼叫應用程式設計介面。

動作： 請以另一個使用者 ID 執行應用程式，或讓

資料庫管理員變更初始使用者 ID 的權限層次。

DMB0007E 未對 Extender
"**<extender-name>**" 啟用使用者表格 "**<table-name>**"。

原因: 未對 DB2 Extender 啟用嘗試對其進行作業的表格。例如，必須先啟用表格來存放音效資料，才能對音效啟用表格中的直欄。

動作: 請確定已先為 Extender 啟用表格。然後啟用直欄。

DMB0008E 在執行儲存程序 "<name>**" 時發生錯誤。**

原因: 訊息識別的儲存程序可能有錯誤，或環境有問題。

動作: 請驗證您的應用程式，然後重試。

DMB0009E 記憶體配置錯誤。

原因: 系統無法配置支援嘗試作業所需的記憶體。

動作: 請驗證您的系統具有足夠的記憶體來完成該項作業。

DMB0010E 已預先對 UDT "<name>**" 定義了 "**<extender-name>**" Extender。**

原因: 使用者定義類型 (UDT) 的名稱已用於 UDT，該 UDT 是為另一個 DB2 Extender 所定義的。

動作: 請選擇另一個 UDT 名稱。

DMB0011E 無法對 "<extender-name>**" Extender 啟用使用者直欄 "**<column-name>**"。使用者直欄的定義和與 Extender 相關的特殊類型 "**MMDBSYS.<name>**" 不相容。**

原因: 此直欄不是為訊息中的資料類型而定義，所以無法為該 Extender 啟用這個直欄。

動作: 請確定已使用對應於 Extender 的資料類型，來定義您要啟用的直欄。

DMB0012E 所指定的使用者表格 "<table-name>**" 不存在。**

原因: 不存在所指定名稱的表格。

動作: 請驗證表格的名稱及表格存在與否。

DMB0013E 未對表格 "<table-name>**" 定義直欄 "**<column-name>**"。**

原因: 嘗試執行的作業參照到一個識別表格內所沒有的直欄名稱。

動作: 請檢查表格及直欄的名稱。

DMB0014W 已對 "<extender-name>**" Extender 啟用直欄 "**<column-name>**" (在使用者表格 "**<table-name>**" 中)。**

原因: 試圖對已啟用直欄的 Extender 啟用直欄。

動作: 不需要任何動作。

DMB0015W 已對 Extender "<extender-name>**" 啟用資料庫。**

原因: 試圖對已啟用資料庫的 Extender 啟用資料庫。

動作: 不需要任何動作。

DMB0016W 已對 "<extender-name>**" Extender 啟用使用者表格 "**<table-name>**"。**

原因: 試圖對已啟用表格的 Extender 啟用表格。

動作: 不需要任何動作。

DMB0017E 已對 "<extender-name>" Extender 啓用使用者表格 "<table-name>"。但其中至少有一個 meta 資料表 "<table-name>" 或 "<table-name>" 不存在。

原因: 與此表格相關的一或多個管理支援 (meta 資料) 表格已損壞或損毀。如果沒有這些 meta 資料表, 使用者表格將無法用於該 Extender 類型的資料。

動作: 請停用使用者表格, 並對 Extender 重新加以啓用。

DMB0018E 系統無法為直欄 "<column-name>" (在表格 "<table-name>" 中) 建立一個唯一的觸發函式名稱。

原因: 系統嘗試啓用使用者表格中的直欄時, 在建立 DB2 Extender 使用的觸發函式期間發生錯誤。

動作: 請重複該作業。如果再度發生錯誤, 請連絡資料庫管理員, 然後再連絡 IBM 服務部門。

DMB0019I "<Count>" 檔參照 Extender "<extender-name>" 的表格 "<table-name>"。

原因: 此訊息顯示特定 Extender 的使用者表格所參照的外部媒體檔數目。

動作: 不需要任何動作。

DMB0020I 在 "<extender-name>" Extender 的表格綱目 "<name>" 表格中參照了 "<Count>" 個檔案。

原因: 此訊息顯示具有特定綱目名稱的使用者表格所參照的外部媒體檔數目。

動作: 不需要任何動作。

DMB0021I "<extender-name>" Extender 的表格 "<table-name>" 中參照了 "<count>" 個無法存取的檔案。

原因: 此訊息顯示特定 Extender 的使用者表格所參照但無法存取的外部媒體檔數目。那些檔案可能已被消除。

動作: 不需要任何動作。

DMB0022I "<extender-name>" Extender 參照了 "<count>" 個無法存取的檔案。

原因: 此訊息顯示有下列情況的外部媒體檔數目:

- 為現行資料庫中的使用者表格所參照。
- 具有特定 Extender 媒體類型 (如視訊)。
- 無法存取; 例如, 檔案可能已被消除。

動作: 不需要任何動作。

DMB0023I Extender "<extender-name>" 的表格綱目 "<name>" 表格中參照了 "<count>" 個無法存取的檔案。

原因: 此訊息顯示具有特定綱目名稱的使用者表格所參照, 但無法存取的外部媒體檔數目。那些檔案可能已被消除。此訊息同時也指出表格對其啓用的 Extender 數。

動作: 不需要任何動作。

DMB0024I 已對 "<count>" 個 Extender 啓用現行資料庫。

原因: 此訊息列出現行資料庫已針對多少個 DB2 Extender 而啓用。

動作: 不需要任何動作。

DMB0025I 已對 "<count>" 個 Extender 啓用表格 "<table-name>"。

原因: 此訊息列出所指出的表格已針對多少個 DB2 Extender 而啓用。

訊息

動作: 不需要任何動作。

DMB0026I 已對 "<count>" 個 Extender 啟用直欄 "<column-name>" (在表格 "<table-name>" 中)。

原因: 此訊息列出所指出的直欄已針對多少個 DB2 Extender 而啟用。

動作: 不需要任何動作。

DMB0027I 已對 Extender "<extender-name>" 啟用現行資料庫。

原因: 此訊息指出現行資料庫對其啟用的 DB2 Extender。

動作: 不需要任何動作。

DMB0028I 已對 Extender "<extender-name>" 啟用表格 "<table-name>"。

原因: 此訊息指出啟用使用者表格所要存放的媒體資料類型。

動作: 不需要任何動作。

DMB0029I 已對 Extender "<extender-name>" 啟用直欄 "<column-name>" (在表格 "<table-name>" 中)。

原因: 此訊息指出啟用使用者直欄所要存放的媒體資料類型。

動作: 不需要任何動作。

DMB0030E 無法對 "<extender-name>" Extender 啟用現行資料庫。RC = "<code>."。

原因: 可能該資料庫不存在，或您未經授權啟用該資料庫。

動作: 請確定該資料庫存在，並且您被授權啟用該資料庫。

DMB0031E 無法對 "<extender-name>" Extender 啟用表格。RC = "<code>."。

原因: 可能此資料庫不存在或未啟用表格，或您未經授權啟用該表格。

動作: 請確定此資料庫存在，並且已對 Extender 啟用了資料庫及表格。請確定您有啟用此表格的權限。

DMB0032E 無法對 "<extender-name>" Extender 啟用直欄。RC = "<code>."。

原因: 未使用此 Extender 的資料類型定義此直欄，或此直欄不存在、未啟用表格，或您未經授權啟用該直欄。

動作: 請確定已使用正確的資料類型定義直欄。請確定已啟用表格，而且您有啟用此直欄的權限。

DMB0033E 您未經授權執行此命令。

原因: 您的使用者 ID 並無執行該命令所需的權限層次。

動作: 請以另一個使用者 ID 執行命令，或讓資料庫管理員變更您現行使用者 ID 的權限層次。

DMB0034I 已順利啟動資料庫 "<database-name>" 的 DB2 Extender Server。

原因: 已順利啟動現行資料庫的伺服器。

動作: 不需要任何動作。

DMB0035I 已停止資料庫 "<database-name>" 的 DB2 Extender Server。

原因: 已順利停止現行資料庫的伺服器。

動作: 不需要任何動作。

DMB0036E 無法啟動或停止 DB2 Extender 伺服器。DB2 Extender 伺服器常駐程式可能不在執行中。請連絡您的資料庫管理員。

原因: 在啟動或停止 DB2 Extender 伺服器時發生錯誤。DB2 Extender 伺服器常駐程式可能不在執行中。

動作: 請連絡您的資料庫管理員。

DMB0037E USE 階段作業控點無效。

原因: 發生了內部錯誤。

動作: 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0038E USE 陳述式控點無效。

原因: 發生了內部錯誤。

動作: 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0039E USE 錯誤："<error>"。

原因: 發生了內部錯誤。

動作: 請遵循相關錯誤訊息所包含的指示並重複作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0040E SQL 錯誤："<error>"

原因: 發生了內部錯誤。

動作: 請遵循相關錯誤訊息所包含的指示並重複作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0041W 已使用新指定的表格空間，對 "<extender-name>" Extender 重新啓用了現行資料庫。

原因: 當現行資料庫被預先啓用時，它使用的是不同的表格空間。現在已以管理支援表格的新表格空間啓用了資料庫。

動作: 不需要任何動作。

DMB0042E 未對 "<extender-name>" Extender 啓用直欄 "<column-name>" (在表格 "<table-name>" 中)。

原因: 未對嘗試對其進行作業的 Extender 啓用指定的直欄。例如，您可能已嘗試停用目前未對指定之 Extender 啓用的直欄。

動作: 請確定已對訊息指示的 Extender 啓用直欄。

DMB0043I 已對 "<extender-name>" Extender 停用現行資料庫。

原因: 停用作業成功。

動作: 不需要任何動作。

DMB0044I 已對 "<extender-name>" Extender 停用表格 "<table-name>"。

原因: 停用作業成功。

動作: 不需要任何動作。

DMB0045I 已對 "<extender-name>" Extender 停用直欄 "<column-name>" (在表格 "<table-name>" 中)。

原因: 停用作業成功。

動作: 不需要任何動作。

DMB0046E 無法對 "<extender-name>" Extender 停用現行資料庫。RC = "<code>."

原因: 可能此資料庫不存在或未對 Extender 而啓用，或您未經授權停用該資料庫。

動作: 請確定此資料庫存在，而且已為 Extender 啓用它。請確定有授權您停用資料庫。

DMB0047E 無法對 "<extender-name>"
Extender 停用表格。RC =
"<code>."

原因: 可能此表格不存在或未對 Extender 而啟用，
或您未經授權停用該表格。

動作: 請確定此表格存在，而且已為 Extender 啟用
它。請確定您有停用表格的權限。

DMB0048E 無法對 "<extender-name>"
Extender 停用直欄。回覆碼 =
"<code>"

原因: 沒有為訊息指示的 Extender 啟用此直欄， 所
以無法為該 Extender 停用此直欄。

動作: 請驗證 Extender 的名稱，並驗證使用者直欄
是否是您要停用的直欄。

DMB0049E 您未經授權執行此命令。

原因: 您的使用者 ID 並無執行該命令所需的權限
層次。

動作: 請以另一個使用者 ID 執行應用程式，或讓
資料庫管理員變更您現行使用者 ID 的權限層次。

DMB0050E 您對表格 "<table-name>" 並無
"<authority-level>" 權限。

原因: 作業所需的權限層次高於嘗試作業的其中一
個使用者 ID 的權限層次。

動作: 請以擁有合適授權的使用者 ID 執行作業，
或讓資料庫管理員變更您現行使用者 ID 的權限層
次。

DMB0051E 媒體檔標頭無效。

原因: 系統無法讀取或開啓媒體檔的標頭。檔案可
能損壞，或它並非媒體檔。

動作: 請驗證檔案是媒體檔，而且沒有損壞。

DMB0052I 資料庫 "<database-name>" 的
DB2 Extender 伺服器已順利啓
動。

原因: 伺服器順利啓動。

動作: 不需要任何動作。

DMB0053I 資料庫 "<database-name>" 的
DB2 Extender 伺服器已順利停
止。

原因: 伺服器順利停止。

動作: 不需要任何動作。

DMB0054E DB2 Extender 伺服器無法連接資
料庫，或無法配置 DB2 陳述式控
點。資料庫 "<database-name>"
的 DB2 Extender 伺服器可能不在
執行中。

原因: DB2 Extender 伺服器無法連接資料庫，或無
法配置 DB2 陳述式控點。資料庫的 DB2 Extender
伺服器可能不在執行中。

動作: 請確定正在執行資料庫的 DB2 Extender 伺
服器。如果不在執行中，請對資料庫啟動特定的
Extender 伺服器，或要求您的系統管理者重新啟動
Extender 服務程式。

DMB0055I 順利完成 "command-name" 命
令。

原因: 順利完成命令。

動作: 不需要任何動作。

DMB0056E 在 "<keyword>" 之後發現一個非預
期的符記 "<token>"。預期的符記
可包括: <extendername>。

原因: 此命令預期 DB2 Extender 名稱，而不是訊
息指示的符記。

動作: 請遵循命令的語法，然後重試該命令。

DMB0057E 表格空間 "**<table-space-name>**" 無效。

原因: 訊息中的表格空間不存在。

動作: 請驗證表格空間的名稱，並驗證它是否存在。

DMB0058I 有 "**<Count>**" 個檔為 "**<extender-name>**" Extender 所參照。

原因: 此訊息顯示特定 Extender 所參照的外部媒體檔數目。

動作: 不需要任何動作。

DMB0059E "**<Name>**" 不是一個有效的 DB2 Extender 名稱。有效的 Extender 名稱包括 "**<extender-name,>**" DB2VIDEO、DB2AUDIO 及 DB2IMAGE。

原因: Extender 名稱的拼字有誤。

動作: 請驗證 Extender 名稱。

DMB0060E "**<function>**" 的現行語法是: "**<syntax>**"。

原因: 您輸入的命令語法錯誤。

動作: 請遵循此訊息中所敘述的語法。

DMB0061E "**<keyword>**" 之後的表格名稱 "**<name>**" 無效。

原因: 該命令預期會有表格的名稱。

動作: 請遵循命令的語法，然後重試該命令。

DMB0062E "**<keyword>**" 之後的直欄名稱 "**<name>**" 無效。

原因: 該命令預期會有直欄的名稱。

動作: 請遵循命令的語法，然後重試該命令。

DMB0064E 系統無法辨識 "**<keyword>**" 之後的 "**<token>**"。

原因: 此命令預期的並非訊息所指示的符記。

動作: 請遵循命令的語法，然後重試該命令。

DMB0065E "**<keyword>**" 之後的使用者 ID "**<identifier>**" 無效。

原因: 該命令預期會有一個有效的使用者 ID。

動作: 請驗證您要的使用者 ID，然後重試。

DMB0066E "**<keyword>**" 之後的通行碼 "**<password>**" 無效。

原因: 此命令預期的是有效的使用者 ID，而不是訊息指示的符記。

動作: 請驗證通行碼，然後重試。

DMB0067E 您輸入的命令無效。

原因: 命令名稱拼字錯誤，或語法錯誤。

動作: 請遵循命令的語法，然後重試該命令。

DMB0068E meta 資料表不存在。

原因: 函數嘗試使用管理支援 (meta 資料) 表，資料物件應有該表格存在。meta 資料表格可能已損壞或被清除。

動作: 請檢查名稱，並驗證 meta 資料表是否存在。若意外清除或損壞 meta 資料表格，請停用該表格，然後重新啓用資料物件。

DMB0069E 資料庫名稱 "**<name>**" 無效。

原因: 不存在該名稱的資料庫。

動作: 請檢查名稱，並驗證資料庫是否存在。

DMB0070E 控點無效。

原因： 您傳遞的應用程式控點值可能已損壞。

動作： 驗證應用程式以確定 Extender 控點值沒有被變更。

DMB0071E 無法連接 "<database-name>"。

原因： 資料庫的 DB2 Extender 伺服器可能未啟動。

動作： 請檢查伺服器的狀態。如果該伺服器不在執行中，請在 DMB 命令行上使用 START SERVER 命令，重新加以啟動。

DMB0072E UDF SQL 伺服器無法切斷與資料庫的連線。

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0073E USE 階段作業控點無效。

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0074E USE 陳述式控點無效。

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0075E 請指定一個檔名。

原因： 此作業需要一個媒體檔名稱。

動作： 請輸入一個媒體檔名稱。

DMB0076E 無法開啓匯入檔。

原因： 匯入檔案可能遺失或已損壞。

動作： 請驗證匯入檔的名稱及其是否存在。

DMB0077E 無法開啓/讀取內容檔。

原因： Extender 控點指向的檔案不存在或訛誤。Extender 已無法存取該檔案。

動作： 請使用 FILENAME UDF 來尋找檔案名稱，或驗證該內容檔是否存在。

DMB0078E 無法建立匯出檔。

原因： 匯出檔遺失或訛誤。

動作： 請驗證匯出檔的名稱及其是否存在。

DMB0079E 無法將 BLOB 複製到檔案。

原因： 該檔案無法接受 BLOB。可能沒有足夠的儲存體空間來容納 BLOB。

動作： 請將 BLOB 的大小與可用的儲存體加以比較，並在必要時增加儲存體。

DMB0080E 無法寫入檔案。

原因： 檔案已損壞或不存在，或名稱的拼法有誤。

動作： 請驗證該檔案的名稱及其是否存在。

DMB0081E 偏移或大小無效。

原因： 作業未在資料結構中找到預期的資料。可能是欄位的大小或偏移不正確。

動作： 請驗證偏移及欄位的大小。

DMB0082E 無法建置控點。

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0083E "<extender-name>" 與 "<extender-name>" 不相容。

原因： 訊息中所指定的兩個 Extender 在此用途上並不相容。依所有或次選取的插入作業無效。

動作: 請確定已對同一個 Extender 啓用目標物件，其原始物件是啓用的。

DMB0084E 匯入要求無效：檔名、內容、儲存體類型。

原因: 匯入作業失敗。可能是檔名、內容或儲存體類型無效。

動作: 請檢查資料，然後重試。

DMB0085E 更新要求無效：檔名、內容、儲存體類型。

原因: 更新作業失敗。可能是檔名、內容或儲存體類型無效。

動作: 請檢查資料，然後重試。

DMB0086E 所要求的大小太大。

原因: 您所要求的大小超過 UDF 的最大 BLOB 大小。

動作: 請要求較小的尺寸。

DMB0087E 檔名無效。

原因: 沒有該名稱的檔案存在。

動作: 請驗證該檔案的名稱及其是否存在。

DMB0088E 控點值為 NULL。

原因: UDF 預期會有一個非 NULL 的控點。

動作: 請確定應用程式取得有效的控點，並且將它傳給 UDF。

DMB0089E 控點值不存在。

原因: 傳給 UDF 的控點無效。

動作: 請確定該應用程式傳遞的是有效的控點。

DMB0090E 資料被截斷。

原因: 檔案或緩衝區太小而無法接受該資料。

動作: 請增加檔案或緩衝區的大小。

DMB0091W 內容已在檔案中。

原因: 該檔案中已經有內容。該內容將被改寫。

動作: 不需要任何動作。

DMB0092E 對直欄 "<column-name>" 所嘗試的插入作業無效。已對 "<extender-name>" Extender 啓用了直欄。

原因: 插入的資料類型與已啓用直欄的 Extender 不同。

動作: 請確定已對同一個 Extender 啓用目標物件，其原始物件是啓用的。

DMB0093E 對直欄 "<column-name>" 所嘗試的更新作業無效。已對 "<extender-name>" Extender 啓用了直欄。

原因: 更新的資料類型與已啓用直欄的 Extender 不同。

動作: 請確定已對同一個 Extender 啓用目標物件，其原始物件是啓用的。

DMB0094I 表格 "<table-name>" 不存在。

原因: 系統找不到該名稱的表格。該表格可能存在另一個資料庫中。

動作: 不需要任何動作。

DMB0095W 未對 "<extender-name>" 啓用表格 "<table-name>"。

原因: 未對 Extender 啓用表格。

動作: 不需要任何動作。

DMB0096W 未對 "<extender-name>"
Extender 啟用直欄
"<column-name>" (在表格
"<table-name>" 中)。

原因: 系統預期會啟用直欄。

動作: 不需要任何動作。

DMB0097W 未對 "<extender-name>"
Extender 啟用現行資料庫。

原因: 系統預期會啟用資料庫。

動作: 啟用訊息所指示的 Extender 資料庫。

DMB0098E 使用者對表格 "<table-name>" 沒
有 "<authority-level>" 權限。

原因: 作業所需的權限層次高於嘗試作業的其中一個使用者 ID 的權限層次。

動作: 請以擁有該表格的使用者 ID 執行作業，或要求資料庫管理員變更您現行使用者 ID 的權限層次。

DMB0099E 無法確定異動。

原因: 現行資料庫的 Extender 伺服器可能已停止。

動作: 請檢查伺服器的狀態。如果該伺服器不在執行中，請在 db2ext 命令行上使用 START SERVER 命令，重新加以啟動。

DMB0100E "<name>" 不是一個有效的表格名稱。

原因: 不存在該名稱的表格。

動作: 請驗證表格的名稱及其是否存在，然後重試。

DMB0101E NULL 參數無效。

原因: 該命令預期會有一個非 NULL 的參數。

動作: 請檢查語法，然後重試。

DMB0102E 儲存體類型無效。

原因: 在 DB2 Extender 中，儲存體類型會指出媒體資料的儲存位置。

動作: 指定 0 表示外部 (在檔案中)，1 表示內部 (在資料庫中)。

DMB0103E 不受支援的格式。

原因: DB2 Extender 不支援此物件的格式。

動作: 請將該物件轉換成受支援的格式。

DMB0104E 影像內容緩衝區太小。

原因: 視訊片段過大，無法放入為它配置的緩衝區。

動作: 請配置較大的緩衝區。

DMB0105E MPEG1 標頭無效。

原因: MPEG1 檔的標頭遺失或訛誤。

動作: 請驗證檔案是 MPEG1 檔。

DMB0106E AVI 標頭無效。

原因: AVI 檔的標頭遺失或訛誤。

動作: 請驗證檔案是 AVI 檔。

DMB0107E 未設定匯出環境。

原因: 在 DB2 Extender 中，匯出環境的環境變數未正確設定。

動作: 請確定已按照第 543 頁的附錄 A, 『設定 DB2 Extender 的環境變數』說明，設定好環境變數。

DMB0108E 未設定匯入環境。

原因: 在 DB2 Extender 中，匯入環境的環境變數未正確設定。

動作: 請確定已按照第 543 頁的附錄 A, 『設定

DB2 Extender 的環境變數』說明，設定好環境變數。

DMB0109E 無法分辨匯入檔。

原因： 沒有該名稱的匯入檔存在。

動作： 請驗證檔名和檔案是否存在，並請確定已按照第 543 頁的附錄 A，『設定 DB2 Extender 的環境變數』說明，設定好環境變數

DMB0110E 無法分辨匯出檔。

原因： 沒有該名稱的匯出檔存在。

動作： 請驗證檔名和檔案是否存在，並請確定已按照第 543 頁的附錄 A，『設定 DB2 Extender 的環境變數』說明，設定好環境變數

DMB0111E 未設定儲存環境。

原因： 儲存環境的環境變數未正確設定。

動作： 請確定已如第 543 頁的附錄 A，『設定 DB2 Extender 的環境變數』中所述，正確設定環境變數。

DMB0112E 無法分辨儲存檔。

原因： 沒有該名稱的儲存檔存在。

動作： 請驗證檔名和檔案是否存在，並請確定已按照第 543 頁的附錄 A，『設定 DB2 Extender 的環境變數』說明，設定好環境變數

DMB0113E 無法開啓匯入檔。

原因： 該檔案可能被他人鎖定，或者檔案遺失或訛誤。

動作： 請驗證該檔案的名稱、是否存在及其狀態，以及您的授權層次。

DMB0114E 無法開啓匯出檔。

原因： 該檔案可能被他人鎖定，或者檔案遺失或訛誤。

動作： 請驗證該檔案的名稱、是否存在及其狀態，以及您的授權層次。

DMB0115E 無法開啓儲存檔。

原因： 系統嘗試寫入某檔案，但該檔案已存在。伺服器並無改寫該檔案的權限。

動作： 請驗證該檔案的名稱、是否存在及其狀態，以及您的授權層次。

DMB0116E 無法建立暫用檔。

原因： 可能沒有足夠的儲存體空間來建立暫用檔。

動作： 請確定已設定好 Extender 的暫用環境變數。必要時可增加儲存體。

DMB0117E 未設定暫用環境。

原因： 暫用環境的環境變數未正確設定。

動作： 請確定已按照第 543 頁的附錄 A，『設定 DB2 Extender 的環境變數』說明，設定好環境變數。

DMB0118E 無法開啓暫用檔。

原因： 暫用檔可能已被改寫或有損壞。

動作： 請確定已按照第 543 頁的附錄 A，『設定 DB2 Extender 的環境變數』說明，設定好環境變數。

DMB0119I dmbsrv 伺服器正對 "<name>" 啓動 "<count>" 個連線。

原因： 此訊息指出啓動伺服器時建立的連線數。

動作： 不需要任何動作。

DMB0120E dmbsrv 伺服器無法對 "<name>" 啓動 "<count>" 個連線。

原因： 可能 DB2 尚未啓動或資料庫不存在，或您的系統已用盡授權連線。

動作： 請確定已啓動 DB2 而且資料庫存在。如果問

訊息

題仍未解決，請連絡 IBM，取得更多授權。

DMB0121I dmbsrv 伺服器已對 "<name>" 啟動 "<count>" 個連線。

原因： 此訊息指出啟動伺服器時建立的連線數。

動作： 不需要任何動作。

DMB0122I dmbssd 伺服器已備妥。

原因： 伺服器已準備好執行應用程式。

動作： 不需要任何動作。

DMB0129E 無效的作業： "<operation-name>"。

原因： 不存在該名稱的命令或 API。

動作： 驗證該命令或 API，然後重試。

DMB0130E 直欄 "<column-name>" 無法為 SQL 陳述式所連結。

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0131E SQL 準備陳述式失敗。

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0132E SQL 設定參數失敗。

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0133E SQL 執行陳述式失敗。

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0134E 檔案格式轉換失敗。

原因： 所儲存之多媒體資料的格式不支援格式轉換。

動作： 您無法轉換此檔案的格式。

DMB0135E 無法開啓/讀取縮圖。

原因： 縮圖檔可能遺失或已損壞。

動作： 請驗證該縮圖檔的名稱、是否存在及完整性。

DMB0136E 找不到連結檔案。

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0137E 無法連接資料庫 "<database-name>"

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0138E 無法解除 SQL 陳述式。

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0139E "<keyword>" 之後的特性名稱 "<name>" 無效。

原因： Image Extender 預期在此命令中會有一個有效的特性名稱。

動作: 請以有效的特性名稱重試該命令。有效的特性名稱如下：

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

DMB0141E "<keyword>" 之後的限定元 "<identifier>" 無效。

原因: 系統無法識別命令中的限定元。

動作: 請檢查該限定元，然後重試。

DMB0142E 未開啓任何型錄。

原因: 在 DB2 Extender 中，現行命令需要 QBIC 型錄開啓。

動作: 請以 OPEN QBIC CATALOG 命令開啓 QBIC 型錄。

DMB0143I 在直欄 "<column-name>" (表格 "<table-name>" 中) 的 QBIC 型錄中，自動編目已被設定為 "<status>"。共有 "<count>" 個特性：

原因: 此訊息指出在 QBIC 型錄中為特定影像直欄所定義的特性數目，以及是否已開啓自動編目。

動作: 不需要任何動作。

DMB0145E 查詢控點無效。

原因: 您在 API 呼叫中所使用的查詢控點無效。

動作: 請查看應用程式，瞭解您是否獲取正確的查詢控點。

DMB0146E 特性類別名稱 "<feature-class>" 無效。

原因: 沒有該名稱的特性類別。有效的特性名稱如下：

- QbColorFeatureClass
- QbColorHistogramFeatureClass

- QbDrawFeatureClass
- QbTextureFeatureClass

動作: 請更正特性的名稱，然後重試。

DMB0147E 特性類別名稱 "<feature-class>" 遺漏或無效。

原因: 有效的特性名稱如下：

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

動作: 請更正特性的名稱，然後重試。

DMB0148E 特性 "<feature-name>" 已經是查詢的一個成員。

原因: 查詢原已支援訊息中所指出的特性。

動作: 不需要任何動作。

DMB0149E 特性 "<feature-name>" 不是查詢的一個成員。

原因: 查詢不包括指定的特性名稱。

動作: 若要將此特性新增至查詢之後，才呼叫其它存取此特性的 API，請使用 QbQueryAddFeature API。

DMB0150E 系統無法配置記憶體。

原因: 系統無法配置支援所嘗試之作業所需的記憶體。

動作: 請驗證您的系統具有足夠的記憶體來完成該項作業。

DMB0151E 指向回覆值的指標為 NULL。

原因: API 呼叫未順利完成，因為指向回覆值的指標不可為 NULL。

動作: 請確定對 API 呼叫提供了有效的參數，並遵循正確的語法。

DMB0152E 指向清單回覆值的指標為 NULL。

原因: API 呼叫未順利完成，因為指向回覆值的指標不可為 NULL。

動作: 請確定對 API 呼叫提供了有效的參數，並遵循正確的語法。

DMB0153E scope 參數已被保留，且必須為 0。

原因: 該參數已被保留供未來使用。

動作: 請將範圍設定為 0。

DMB0154E 指向特性類別名稱的指標無效。

原因: API 呼叫預期會有一個指向輸入特性類別名稱的有效指標。

動作: 請確定對 API 呼叫提供了有效的參數，並遵循正確的語法。

DMB0155I 對 "<function-name>" 函數傳遞了大小為零的緩衝區。

原因: API 呼叫需要緩衝區來傳回資訊。

動作: 不需要任何動作。

DMB0156E QbImageSource 指標為 NULL。

原因: NULL 值表示不可變更結構。

動作: 不需要任何動作。

DMB0157E QbImageSource 類型 "<type>" 無效。

原因: 此 DB2 Extender API 參照的資料結構是錯誤的資料類型。

動作: 該結構的資料類型應為 QbImageSource。

DMB0159E 指向 QbImageSource 壓縮檔緩衝區的指標為 NULL。

原因: API 呼叫預期會有一個指標傳回。

動作: 請查看應用程式，瞭解 API 呼叫及緩衝區的指定是否正確。

DMB0160I 壓縮檔緩衝區或檔案的長度為零。

原因: 長度為零。

動作: 不需要任何動作。

DMB0161E 指向表格及/或直欄名稱的指標為 NULL。

原因: API 呼叫預期會提供一個指標。

動作: 請查看應用程式，瞭解對 API 呼叫之輸入的指定是否正確。

DMB0162I 您將 requestedHits 設定為零。

原因: requestedHits 若設為零，您將無法取回任何事項。

動作: 不需要任何動作。

DMB0163I 尚不支援該功能。

原因: 尚不支援該功能。

動作: 不需要任何動作。

DMB0164E 系統無法處理該查詢 (<query-name>)。

原因: 當建立查詢時發生錯誤。

動作: 請驗證對命令或 API 的輸入，然後重試。

DMB0165E 系統無法執行查詢 (<query-name>)。

原因: 當建立查詢時發生錯誤。

動作: 請驗證對命令或 API 的輸入，然後重試。

<p>DMB0166E 執行 "<name>" 時在 "<name>" 中發現陳述式錯誤： "<error>"</p> <p>原因: 發生了內部 IBM 錯誤。</p> <p>動作: 請連絡您的資料庫管理員。</p>	<p>DMB0172E SQL 錯誤： "<error>"。</p> <p>原因: 發生了內部錯誤。</p> <p>動作: 請遵循相關錯誤訊息所包含的指示並重複作業。如果再度發生錯誤，請連絡 IBM 服務部門。</p>
<p>DMB0167E 在讀取 QbGenericImageDataClass 時發生錯誤 (<error>)。</p> <p>原因: 發生了內部 IBM 錯誤。</p> <p>動作: 請連絡您的資料庫管理員。</p>	<p>DMB0173E 型錄已開啓，但供唯讀： "<catalog-name>"。</p> <p>原因: 您無法更新該型錄，因為已有人加以開啓為寫入模式，或您對該型錄沒有寫入權限。</p> <p>動作: 請等到其它使用者完成作業後，以另一個使用者 ID 執行應用程式，或讓資料庫管理員變更您現行使用者 ID 的權限層次。</p>
<p>DMB0168E 在搜尋查詢的特性 "<feature-name>" 之前未先加以設定。</p> <p>原因: 該查詢並未執行，因為未對其指定特性。</p> <p>動作: 請使用 QbAddFeature API 或 ADD QBIC FEATURE 命令，將特性新增至查詢中。</p>	<p>DMB0174E 發生了系統錯誤： "<error>"。</p> <p>原因: 發生了內部 IBM 錯誤。</p> <p>動作: 請遵循相關錯誤訊息所包含的指示並重複作業。如果再度發生錯誤，請連絡 IBM 服務部門。</p>
<p>DMB0169E 在 Call-Level 介面中發生了下列錯誤： "<error>"。</p> <p>原因: CLI 錯誤。</p> <p>動作: 請遵循訊息本文中的指示。</p>	<p>DMB0175I 找不到影像： "<information>"。</p> <p>原因: 找不到符合查詢的影像。。資料庫可能是空的。</p> <p>動作: 不需要任何動作。</p>
<p>DMB0170E 查詢名稱 "<query-name>" 已在使用中。</p> <p>原因: 有另一個該名稱的查詢存在。</p> <p>動作: 請選取另一個名稱。</p>	<p>DMB0176I 直欄已經具有 QBIC 型錄： "<table-name column-name>"。</p> <p>原因: 有另一個該名稱的型錄存在。</p> <p>動作: 不需要任何動作。</p>
<p>DMB0171E 尚未儲存查詢名稱 "<query-name>"。</p> <p>原因: 系統在建立了該查詢之後，未能加以儲存。</p> <p>動作: 請確定您有寫入的權限，而且有足夠的空間來儲存此查詢。</p>	<p>DMB0177E 系統無法開啓該型錄； 錯誤訊息如下： "<error>"。</p> <p>原因: 型錄已損壞。</p> <p>動作: 請遵循訊息本文中的指示。</p>

DMB0178E 系統無法刪除該型錄；錯誤訊息如下："<error>"。

原因： 型錄不存在或已損壞。

動作： 請驗證型錄的名稱及是否存在，然後重試。

DMB0179E 型錄控點無效："<error>"。

原因： 您在 API 呼叫中所使用的型錄控點無效。

動作： 請查看應用程式，瞭解您是否獲取正確的型錄控點。

DMB0180I 對型錄的存取被拒絕："<error>"。

原因： 存取被拒絕。

動作： 不需要任何動作。

DMB0181I 型錄在使用中 "<error>"。

原因： 另一項作業正在使用此型錄。

動作： 不需要任何動作。

DMB0184I 已啟動追蹤：

原因： 已啟動追蹤：

動作： 不需要任何動作。

DMB0185I 尚未啟動追蹤。

原因： 尚未啟動追蹤。

動作： 不需要任何動作。

DMB0186I 已在 "<time>" 時從目錄 "<directory-name>" 開啟追蹤。追蹤檔為 "<file-name>"。已寫入 "<Count> 位元組的追蹤資料。

原因： 已開啟追蹤。

動作： 不需要任何動作。

DMB0187E 無法建立通信，因為系統無法開啟舊檔 "<file-name>" 以供寫入。

原因： 您可能不是環境變數 DB2INSTANCE 敘述的現行案例擁有者，或未正確設定環境變數，如 DB2MMTOP。

動作： 請以擁有該案例的使用者 ID 登入。驗證環境變數已正確設定。

DMB0188I 在建立追蹤常駐程式時發生錯誤："<error>"

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0189I 已順利啟動追蹤：

原因： 已啟動追蹤：

動作： 不需要任何動作。

DMB0190E 無法啟動追蹤。

原因： 發生了內部錯誤。

動作： 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0191E 需要設定環境變數 "<name>"。

原因： 系統架構不正確。

動作： 請設定變數，然後重試。

DMB0192I 已順利關閉追蹤。

原因： 已關閉追蹤。

動作： 不需要任何動作。

DMB0193E 系統無法寫入檔案 "<file-name>"。

原因： 您對所指定之檔案的目錄沒有寫入權限。

動作： 請連絡您的資料庫管理員，取得權限。

DMB0194E 系統無法讀取檔案 "<file-name>"。

原因: 可能是該檔案不存在，或您對該檔案沒有讀取權限。

動作: 請確定該檔案存在，並且您對該檔案具有讀取權限。

DMB0198E 輸入檔中的追蹤碼 "<code>" 不明。輸入檔可能損壞。

原因: 發生了內部錯誤。

動作: 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0199E 您對任何參照的表格並無 "<authority-level>" 權限。

原因: 您的使用者 ID 並無該作業所需的權限層次。

動作: 請以另一個使用者 ID 執行作業，或讓資料庫管理員變更您現行使用者 ID 的權限層次。

DMB0200W 您對其中至少一個參照的表格都沒有 "<authority-level>" 權限。

原因: 您的使用者 ID 對某些表格並無所需的權限層次。

若正在列出參照檔案，則所列出的檔案會由您具有 Select 權限的表格來參照。若系統有您無其 Select 權限的表格，就不會列出這些表格參照的檔案。

如果要重組 meta 資料，系統只重組那些您對其具有 Control 權限之表格的 meta 資料。

動作: 若要取得所有的檔案，請以另一個使用者 ID 執行作業，或讓資料庫管理員變更您現行使用者 ID 的權限層次。

DMB0201I 已存在該名稱的特性：
"**<feature-name>**"。

原因: 該名稱的特性已包含在 QBIC 型錄中。

動作: 不需要任何動作。

DMB0202E 特性名稱無效：
"**<feature-name>**"。

原因: 沒有該名稱的特性類別。有效的特性名稱如下：

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

動作: 請更正特性的名稱，然後重試。

DMB0203E 找不到下列特性：
"**<feature-name>**"。

原因: 沒有該名稱的特性類別，或特性類別不包含在 QBIC 型錄中。有效的特性名稱如下：

- QbColorFeatureClass
- QbColorHistogramFeatureClass
- QbDrawFeatureClass
- QbTextureFeatureClass

動作: 請更正特性的名稱，然後重試。

DMB0204E 未對 DB2IMAGE 啟用直欄：
"**<column-name>**"。

原因: 未對 Image Extender 啟用直欄。

動作: 請確定已為 DB2 Image Extender 啟用此直欄。

DMB0205E 未找到 "<table-name column-name>" 的型錄。

原因: 沒有與指定之直欄相關的 QBIC 型錄。

動作: 請在執行任何其它 QBIC 作業之前，先建立直欄的 QBIC 型錄。

DMB0206W 未對任何 Extender 啓用指定的直欄。

原因: 該直欄可能不存在，或其資料類型可能與 Extender 不相容。

動作: 請驗證已使用正確資料類型來定義此直欄。

DMB0207E 無法改寫檔案。

原因: 此檔案已存在，但 EXPORT UDF 無法改寫它。

動作: 請將該檔案匯出成另一個檔名，或容許 EXPORT UDF 可改寫該檔案。

**DMB0208E sqlcode=<code>
cliState=<code>。**

原因: 發生了內部錯誤。

動作: 請重複該作業。如果再度發生錯誤，請連絡 IBM 服務部門。

DMB0209E 音效標頭無效。

原因: 音效檔的標頭遺失或訛誤。

動作: 請驗證 DB2 Extender 是否支援該音效檔的格式。

DMB0211W 檔案存在，未經改寫。

原因: 指定的目標檔已存在，且未經改寫。

動作: 不需要任何動作。

DMB0212E resultType 參數已被保留，且必須為 0。

原因: 該參數已被保留供未來使用。

動作: 請將 resultType 設定為 0。

DMB0214E 指向查詢名稱的指標無效。

原因: API 呼叫預期會有一個指向輸入查詢名稱的有效指標。

動作: 請確定對 API 呼叫提供了有效的參數，並遵循正確的語法。

DMB0352E 未起始設定命令行環境。

原因: 未起始設定執行 db2ext 命令行處理器的命令行環境。（此訊息只適用於 Windows NT 及 Windows 95 環境。）

動作: 請發出 db2cmd 命令來開啓 DB2CLP 視窗，然後發出 db2ext 命令，在該視窗執行 db2 命令行處理器。

DMB0353E 無法與 db2ext 命令行處理器的背景處理通信。

原因: db2ext 命令行處理器的背景處理正在執行中，但 db2ext 命令行處理器無法與之通信。

動作: 請嘗試在另一個視窗發出 db2ext 命令。

DMB0354E 無法啓動 db2ext 命令行處理器的背景處理。

原因: db2ext 命令行處理器的背景處理正在執行中，但 db2ext 命令行處理器無法與之通信。

動作: 請查看背景處理的可執行模組 (db2extb 或 db2extb.exe) 是否存在，以及其目錄是否在 PATH 環境變數中。

DMB0355E db2ext 命令行處理器的背景處理逾時。

原因: db2ext 命令行處理器的背景處理已順利啓動，但 db2ext 命令行處理器無法在容許的時間限制內與之通信。

動作: 請嘗試在另一個視窗發出 db2ext 命令。

DMB0356E 無法與 db2ext 命令行處理器的背景處理通信。

原因: db2ext 命令行處理器對其背景處理傳送了一個要求，但未收到要求。

動作: 請確定 db2ext 命令行處理器的背景處理仍在執行中。

DMB0357E db2ext 命令行處理器的背景處理沒有回應。

原因: db2ext 命令行處理器對其背景處理傳送了一個要求，但該背景處理未在容許的時間限制內回應。

動作: 請確定 db2ext 命令行處理器的背景處理仍在執行中。

DMB0359E 未在逾時週期內建立 db2ext 命令行處理器背景處理要求佇列或輸入佇列。

原因: db2ext 命令行處理器的背景處理未在容許的時間限制內建立訊息佇列。(此訊息只適用於 UNIX 環境。)

動作: 請確定 DB2 案例主目錄常駐的磁碟未滿載(背景處理需要此主目錄來建立訊息佇列的檔案)。如果磁碟未滿載，請查看您是否啟動了太多 db2extb 處理。如果您在多個視窗中執行 db2ext 命令行處理器，便可能發生這種情況。當您第一次在命令模式下發出 db2ext 命令行處理器要求時，會在一個視窗中啟動背景處理。請確定當您不再需要 db2ext 命令行處理器時，有發出命令 db2ext terminate 來終止它。在您發出 terminate 命令後，才會刪除後端處理的訊息佇列。

DMB0361E 未啟用直欄或表格。

原因: 已指定匯入 UDF，但未對 Extender 啟用指定的表格直欄。

動作: 請啟用表格直欄，然後重試。

DMB0363E 遺漏表格及直欄名稱。

原因: 已呼叫更新 UDF，但未指定表格。

動作: 請指定表格，然後重試。

DMB0364E 先前已為表格空間 "<tablespace-name>" 定義了 Extender "<extender-name>"。

原因: 已使用非指定的表格空間，對 Extender 啓用了指定的資料庫、表格或直欄。

動作: 請檢查表格空間規格是否正確。

DMB0365E 您對 "<metadata-table-name>" 及 "<metadata-table-name >" (其為 "<schema-name>."<table-name>" 的 meta 資料表) 並無 CONTROL 權限。

原因: 您的要求已被拒絕，因為您對指定之使用者表格的 meta 資料表並無必要的 CONTROL 權限。

動作: 請讓資料庫管理員授與您對 meta 資料表的 CONTROL 權限。

DMB0366E 預期特性名稱。

原因: 查詢字串中應有特性名稱。

動作: 請更正查詢字串，然後重試。

DMB0367E 預期顏色|color histogram|檔。

原因: 查詢字串中應有 『color』、 『histogram』或 『file』。

動作: 請更正查詢字串，然後重試。

DMB0368E 預期 '、'。

原因: 查詢字串中應有 '、'。

動作: 請更正查詢字串，然後重試。

DMB0369E 檔案無效。

原因: 在查詢字串中指定的檔案無效。

動作: 請更正查詢字串，然後重試。

DMB0370E 預期檔名。

原因: 查詢字串中應有檔名。

動作: 請更正查詢字串，然後重試。

DMB0371E 預期 server|client。

原因: 查詢字串中應有『server』或『client』。

動作: 請更正查詢字串，然後重試。

DMB0372E 預期 '('。

原因: 查詢字串中應有 '('。

動作: 請更正查詢字串，然後重試。

DMB0373E 預期 ')'。

原因: 查詢字串中應有 ')'。

動作: 請更正查詢字串，然後重試。

DMB0374E 預期 percentage。

原因: 查詢字串中應有百分比值。

動作: 請更正查詢字串，然後重試。

DMB0375E 預期 color。

原因: 查詢字串中應有紅、綠及藍等值。

動作: 請更正查詢字串，然後重試。

DMB0376E 預期 '='。

原因: 查詢字串中應有 '='。

動作: 請更正查詢字串，然後重試。

DMB0377E 預期 '<'。

原因: 查詢字串中應有 '<'。

動作: 請更正查詢字串，然後重試。

DMB0378E 預期 '>'。

原因: 查詢字串中應有 '>'。

動作: 請更正查詢字串，然後重試。

DMB0379E 預期 'and'。

原因: 查詢字串中應有 'and'。

動作: 請更正查詢字串，然後重試。

DMB0380E 預期 weight。

原因: 查詢字串中應有加權。

動作: 請更正查詢字串，然後重試。

DMB0381E 未設定特性。

原因: 未將特性新增至 QBIC 型錄中。

動作: 請將特性新增至 QBIC 型錄中，然後將影像重新編目。

DMB0382E 無法建置查詢。

原因: 現行資料庫的 Extender 伺服器可能已停止。

動作: 請檢查伺服器的狀態。如果該伺服器不在執行中，請在 db2ext 命令行上使用 START SERVER 命令，重新加以啟動。

DMB0383E 無法執行查詢。

原因: 現行資料庫的 Extender 伺服器可能已停止。

動作: 請檢查伺服器的狀態。如果該伺服器不在執行中，請在 db2ext 命令行上使用 START SERVER 命令，重新加以啟動。

DMB0384E 無法取得下一個項目。

原因: 已達清單的尾端。

動作: 請確定您的應用程式未嘗試擷取清單結尾以外的項目。

DMB0386E 無法配置使用者資料。

原因: SQL API sqluihsh() 傳回了一個非零的回覆碼。

動作: 請重試。如果問題仍未解決，請尋求 IBM 支援。

DMB0387E 所指定之表格空間的節點群組與使用者表格的節點群組不同。

原因: 作為啓用表格輸入資料傳遞的一或多個表格空間 (亦即為 meta 資料表、索引或 BLOB) 定義於某個節點群組，該節點群組與定義使用者表格的節點群組不同。

動作: 請使用定義於相同節點群組 (定義使用者表格的節點群組) 的表格空間。

DMB0388E 一般、長型或索引表格空間並非定義於同一個節點群組。

原因: 作為啓用表格輸入資料傳遞的一或多個表格空間 (亦即為 meta 資料表、索引或 BLOB) 定義於某個節點群組，該節點群組與定義其它表格空間的節點群組不同。

動作: 請使用定義於相同節點群組的表格空間。

DMB0389W 所指定之表格空間的節點群組不包括所有的分割伺服器。

原因: 已在不包括所有分割伺服器的節點群組上定義了作為輸入資料傳遞的表格空間。

動作: 不需要任何動作。但是，如果表格空間定義於支援所有分割伺服器的節點群組上，則匯入及更新 UDF 的執行效能會更好。特別是在 Extender 應用程式以 BLOB 儲存媒體內容時更為明顯。

DMB0391I 只有當 DB2 UDB 從屬站正在存取 DB2 UDB 伺服器時才可執行此命令。

原因: db2ext 命令行處理器可能未連接到 DB2 UDB 伺服器，或 DB2 UDB 從屬站沒有啓動 db2ext

命令行處理器。例如，只有在 db2ext 命令行處理器已連接到 DB2 非 Extended Enterprise Edition 伺服器的情況下，才能使用命令 START SERVER。

動作: 請勿在現行的主/從架構下發出此命令。

DMB0392I 只有當 DB2 UDB 從屬站正在存取 DB2 UDB Extended Enterprise Edition 伺服器時才可執行此命令。
例如，只有在 db2ext 命令行處理器已連接到 DB2 Extended Enterprise Edition 伺服器的情況下，才能使用命令 DISCONNECT SERVER。

原因: db2ext 命令行處理器可能未連接到 DB2 UDB Extended Enterprise Edition 伺服器，或未自 DB2 UDB 從屬站啓動 db2ext 命令行處理器。

動作: 請勿在現行的主/從架構下發出此命令。

DMB0402E 只有在應用程式已連接到 DB2 "<server-type>" 伺服器的情況下，才可使用命令 "<command-name>" 的選項 "<option-name>"。

原因: 所指定的參數無效，因為 db2ext 命令行處理器未連接到支援該選項之類的伺服器。例如，只有在 db2ext 命令行處理器已連接到 DB2 Extended Enterprise Edition 的情況下，才能以參數 NODENUM <nodenum> 指定命令 GET SERVER STATUS。

動作: 請勿在現行的主/從架構下發出命令與參數組。

DMB0411E 基埠無效

原因: 在案例建立期間輸入了某個無效的 TCP/IP 埠號作為基埠。

動作: 正確的語法為 dmbicrt -r:base_port,end_port -t:base_port,end_port。請更正參數，然後重試該命令。

DMB0412E 端點埠無效

原因: 在案例建立期間已輸入一個錯誤 TCP/IP 埠號作為結束終點埠。

動作: 正確的語法為 `dmbicrt -r:base_port,end_port -t:base_port,end_port`。請更正參數，然後重試該命令。

DMB0413E 無法分辨 DB2 Extender 安裝路徑。

原因: 案例建立程式找不到環境變數 "DMBPATH" 的值。

動作: 請設定變數 "DMBPATH"，然後重試應用程式。

DMB0414E 無法分辨主電腦名稱。

原因: 在嘗試分辨電腦的名稱時發生內部錯誤。

動作: 請尋求 IBM 支援。

DMB0415E 無法分辨此機器的節點號碼。

原因: 正在執行案例建立作業上的機器未列示於檔案 "db2nodes.cfg" 中。

動作: 將機器新增至 "db2nodes.cfg"，然後重試該應用程式。

DMB0416E 必須由 root 啟動本程式。無法繼續執行。

原因: 正在執行之程式下的使用者 ID 沒有 root 權限。

動作: 以 root 身分登入，然後重試應用程式。

DMB0417E 本程式必須由具有管理者權限的使用者執行。無法繼續執行。

原因: 執行程式時使用的使用者 ID 沒有管理者權限。

動作: 以具有管理權限的使用者 ID 登入，然後重試應用程式。

DMB0418E 無法取得有關下列使用者的資訊： "<userid>"。

原因: 嘗試取得與建立的案例相關的使用者資訊時發生內部錯誤。

動作: 請確定有與建立之案例相同名稱的有效使用者 ID，然後重試應用程式

DMB0419E 無法建立 AIV Extender 目錄 "<directory_name>"。回覆碼 = <code>

原因: 在嘗試建立指定的目錄時發生錯誤。回覆碼代表作業系統所傳回的錯誤。

動作: 請確定目錄名稱中指定的檔案系統/磁碟機存在，且許可建立目錄。

DMB0420E 無法建立對 AIV Extender 目錄 "<directory_name>" 的鏈結。回覆 碼 = <code>

原因: 在嘗試建立指定的符號式鏈結時發生錯誤。回覆碼代表從作業系統傳回的錯誤。

動作: 請確定目錄名稱中指定的檔案系統/磁碟機存在，且許可建立鏈結。

DMB0421E 無法開啓舊檔："<file_name>"。回 覆碼 = <code>

原因: 在嘗試開啓指定的檔案時發生錯誤。回覆碼代表從作業系統傳回的錯誤。

動作: 請確定該檔案存在，且容許開啓該檔案。

DMB0422E 無法寫入檔案："<file_name>"。回 覆碼 = <code>

原因: 在嘗試寫入指定的檔案時發生錯誤。回覆碼代表從作業系統傳回的錯誤。

動作: 請確定該檔案存在，且容許寫入該檔案。

DMB0424E 找不到 "db2nodes.cfg" 檔。

原因: 找不到 DB2 檔案 "db2nodes.cfg"。

動作: 請確定已安裝正確版本的 DB2 UDB Extended Enterprise Edition，然後重試應用程式。

DMB0426E 錯誤："<error_code>" 開啓登記鍵 "<registry_key>"。

原因: 在嘗試開啓指定的登記鍵時發生錯誤。

動作: 請紀錄回覆碼，然後尋求 IBM 支援。

DMB0427E 在設定檔登記中未設定變數 "<variable>"。

原因: 在 Windows NT 登記中未找到指定的值。

動作: 請確定已指定有效 DB2 Extender 變數的名稱。

DMB0430E 找不到 DB2 登記值

原因: 找不到 DB2 所使用的登記值。

動作: 請確定已安裝正確版本的 DB2 UDB Extended Enterprise Edition，然後重試應用程式。

DMB0431E 無法建立 Extender 登記鍵："<registry_key>"。

原因: 在嘗試建立 Extender 登記鍵時發生內部錯誤。

動作: 請尋求 IBM 支援。

DMB0432E 無法設定 Extender 登記鍵的值："<registry_key>"。

原因: 在嘗試設定 Extender 登記鍵值時發生內部錯誤。

動作: 請尋求 IBM 支援。

DMB0435E 無法存取控制檔 "<control_file>"。

原因: 找不到指定的控制檔。

動作: 請尋求 IBM 支援。

DMB0443E 無法開啓目錄 "<directory_name>"。回覆碼 = <code>

原因: 在嘗試開啓指定的目錄時發生錯誤。回覆碼代表從作業系統傳回的錯誤。

動作: 請確定目錄名稱中指定的檔案系統/磁碟機存在，且許可開啓目錄。

DMB0449W DB2 Extender 案例建立作業必須有 -q:datapath。

原因: 在嘗試建立 DB2 extender 案例時未指定 -q 參數。

動作: 請指定參數，然後重試該應用程式。

DMB0450W 一或多個指定的 "<port>" 埠已在使用中。

原因: 指定供 DB2 Extender 使用的埠已列示在正被使用中之服務程式檔案中。

動作: 請指定不在使用中之一或多個埠，然後重試應用程式。

DMB0452E 在 "db2nodes.cfg" 檔中找不到節點號碼 "<node_num>"。

原因: 在 db2nodes.cfg 檔中找不到此機器的節點號碼。

動作: 將節點號碼新增至 db2nodes.cfg 檔中，然後重試應用程式。

DMB0460W 無法判斷 TCP/IP 埠是否可用。

原因: 在嘗試驗證指定的 TCP/IP 埠是否已在使用中時發生錯誤。

動作: 請確定所指定的埠未列示在另一個應用程式

正在使用之服務程式檔案中。

DMB0462E 無法起始設定此節點。回覆碼 = <code>

原因: 在嘗試起始設定現行節點時 Extender 啟動發生錯誤。

動作: 請尋求 IBM 支援。

DMB0495E 此版本的 AIV Extender 不支援長檔名。

原因: 當您呼叫 Extender 管理 API 或發出 db2ext 命令行命令時，您指定了長識別字。這版的 AIV Extender 所支援的識別字最大長度為：

- 區域授權 ID (AUTHID) -- 8 個字元
- 表格綱目 (TABSCHEMA) -- 8 個字元
- 表格名稱 (TABNAME) -- 18 個字元
- 直欄名稱 -- 18 個字元

請檢查 API 呼叫或命令，以確定使用短識別字。

DMB0496E 指定的表格名稱或直欄名稱無效。

原因: 當您呼叫 Extender 管理 API 或發出 db2ext 命令行命令時，您指定了無效識別字。可能的原因是，識別字名稱太長。請參閱快速入門一書，以取得 UDB Db2 名稱長度的資訊。

請檢查 API 呼叫或命令，以確定使用短識別字。

DMB497E DB2MMDATAPATH 拒絕存取。

原因: (僅適用於 EEE) 並非全部節點皆可存取您所指定的目錄或共用名稱。建立 DB2 Extender 案例時所指定的目錄或共用名稱必須已經存在於所有節點上，並可存取。請檢查所有節點存在有您建立 DB2 Extender 案例時所指定的目錄或共用名稱，並且可存取它。

DMB498E 至少 DB2MMDATAPATH 路徑中的一部份不是目錄。

原因: (僅適用於 EEE) 您所指定的目錄或共用名稱並非節點上的目錄。建立 DB2 Extender 案例時所指定的目錄或共用名稱必須已經存在於所有節點上，並可存取。請檢查所有節點存在有您建立 DB2 Extender 案例時所指定的目錄或共用名稱，並且可存取它。

DMB499E DB2MMDATAPATH 字串太長。

原因: (僅適用於 EEE) 您所指定的目錄或共用名稱使得變數 DB2MMDATAPATH 太長。建立 DB2 Extender 案例時所指定的目錄或共用名稱必須已經存在於所有節點上，並可存取。請檢查您建立 DB2 Extender 案例時所指定的目錄或共用名稱是正確的且存在，並可在所有節點上存取。

DMB500E DB2MMDATAPATH 目錄不存在。

原因: (僅適用於 EEE) 節點上不存在您所指定的目錄或共用名稱。建立 DB2 Extender 案例時所指定的目錄或共用名稱必須已經存在於所有節點上，並可存取。請檢查所有節點存在有您建立 DB2 Extender 案例時所指定的目錄或共用名稱，並且可存取它。

DMB501E DB2MMDATAPATH 發生未知 stat() 錯誤。

原因: (僅適用於 EEE) 嘗試存取此環境變數中的目錄或共用名稱時發生問題。所有節點都必須要有建立 DB2 Extender 案例時所指定的目錄或共用名稱，並且可加以存取。請檢查所有節點存在有您建立 DB2 Extender 案例時所指定的目錄或共用名稱，並且可存取它。

DMB502E DB2MMDATAPATH 存在，但非目錄。

原因: (僅適用於 EEE) 您已指定目錄或共用名稱的名稱，但該名稱並非節點上的目錄或共用名稱。所有節點都必須要有建立 DB2 Extender 案例時所指定的目錄或共用名稱，並且可加以存取。請檢查所有

節點存在有您建立 DB2 Extender 案例時所指定的目錄或共用名稱，並且可存取它。

DMB503E DB2MMDATAPATH 存在，但無法讀取。

原因: (僅適用於 EEE) 並非全部節點皆可讀取您所指定的目錄或共用名稱。所有節點都必須要有建立 DB2 Extender 案例時所指定的目錄或共用名稱，並且可加以存取。請檢查所有節點存在有您建立 DB2 Extender 案例時所指定的目錄或共用名稱，並且可存取它。

DMB504E DB2MMDATAPATH 存在，但無法寫入。

原因: (僅適用於 EEE) 並非全部節點上皆可寫入您所指定的目錄或共用名稱。所有節點都必須要有建立 DB2 Extender 案例時所指定的目錄或共用名稱，並且可加以存取及讀取/寫入。請檢查所有節點存在有您建立 DB2 Extender 案例時所指定的目錄或共用名稱，並且可存取它。

DMB504E 未設定 DB2MMDATAPATH。

原因: (僅適用於 EEE) 當您建立 DB2 Extender 案例時，未設定環境變數 DB2MMDATAPATH。如果這是新的 DB2 Extender 案例，使用 DMBIDROP 捨棄該案例；然後再指定-q 選項，正確地將它重建。

如果這不是新的 DB2 Extender 案例，則：

- 在 UNIX 環境中：
 1. 檢查目錄正確，而且全部節點都存在該目錄並可存取它。
 2. 修改 \$INSTHOME/dmb/dmbprofile，將 DB2MMDATAPATH 匯出為目錄。
- 在 Windows 環境中：
 1. 請檢查目錄地共用名稱是正確的，而且所有節點存在該目錄並可存取它。
 2. 使用該共用名稱作為值，將登錄項目 DB2MMDATAPATH 新增至 IAV Extender 案例登記中。機碼為

\\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2
Extenders
\\PROFILE\instance_name\DB2MMDATAPATH.

DMB506E 未設定案例名稱。

原因: 當您執行 DMBSTART 時未設定環境變數 DB2INSTANCE。使用 DMBSTART 來啟動 DB2 Extender 服務前，請確定 DB2START 可適當地運作。

DMB507E dmbssd name 節點 arguments

原因: 內部錯誤。請聯絡 IBM 業務代表。

DMB508E 節點號碼必須大於或等於 0。

原因: 內部錯誤。請聯絡 IBM 業務代表。

DMB509E 不應手動啟動此程式。

原因: 內部錯誤。請聯絡 IBM 業務代表。

DMB512E 用法： arguments dmbInstName。

原因: 內部錯誤。請聯絡 IBM 業務代表。

DMB513E Name 不是有效的案例名稱。

原因: 嘗試捨棄 DB2 Extender 案例時所指定的名稱無法辨識為案例名稱。請檢查所指定的案例名稱正確，並且具有該目錄名稱的 \$INSTHOME/dmb 目錄

DMB514I 此案例上既未安裝伺服器，也未安裝從屬站。

原因: 您嘗試捨棄 DB2 Extender 案例，但未安裝此 Extender。請檢查以確定安裝正確，並且未更名安裝目錄。

DMB515I 未捨棄 DB2 案例。您可呼叫 DB2IDROP 來捨棄 DB2 案例。

原因: 當您捨棄 DB2 Extender 案例時，將不會捨棄相關的 DB2 案例。請使用 DB2IDROP 以捨棄 DB2 案例。

DMB518E 異常錯誤。函數 = *function name*, 回覆碼 = *return_code*。

原因: 當您嘗試建立或捨棄 DB2 Extender 案例時，發生異常錯誤。請檢查您的安裝和設定是正確的。

DMB520E 您不可以 Root 身分執行此程式。

原因: 請檢查您具有用來執行此動作的正確權限。

DMB521E 變更 *name* 許可權的嘗試失敗。

原因: 請檢查您具有用來變更許可權的正確權限。

DMB522E 變更 *name* 所有權的嘗試失敗。

原因: 請檢查您具有用來變更所有權的正確權限。

DMB523E 變更 *name* 群組所有權的嘗試失敗。

原因: 請檢查您具有用來變更群組所有權的正確權限。

DMB524E 檔案或目錄 *name* 已存在。

原因: 您所指定的名稱的檔案或目錄已存在。請選擇不同名稱，並重新執行命令。

DMB525E 建立 *name* 的嘗試失敗。

原因: 請檢查您具有用來執行此動作的正確權限。

DMB526E 檔案或目錄 *name* 遺漏。

原因: 找不到所指示的檔案或目錄。請檢查所指定的檔案或目錄名稱為有效的。

DMB527E 將檔案或目錄 *name* 複製到 *name* 的嘗試失敗。

原因: 請檢查您具有用來複製檔案或目錄的正確權限；並檢查有足夠的空間可進行複製。

DMB528E 使用者 ID *user_ID* 無效。

原因: 所指定的使用者 ID 無效。請檢查使用者 ID，並重新執行命令。

DMB529E 使用者 ID *user_ID* 的主群組 *group* 無效。

原因: 指定給使用者 ID 的主群組無效。請檢查主群組是否正確，並重新執行命令。

DMB530E 案例名稱 *name* 無效。

原因: 當您嘗試建立或使用案例時，所指定的名稱無效。請檢查案例名稱是否有效，並重新執行命令。

DMB531E 未支援的作業系統 *name* 版本 *version_number*。

原因: 您嘗試在未支援版本的作業系統上執行此命令。請檢查執行此作業的需求。

DMB535E 無法存取指定的檔案

原因: 執行此命令前，請檢查您具有對此檔案的存取權。

DMB0533E 在已分割資料庫伺服器環境中，不支援 API <API name>。

原因: 在已分割資料庫環境中，無法使用所指定的 API。

動作: 請參閱所指定的 API 區段，以取得有關如何在應用程式中處理此函數的資訊。

DMB0534E 不支援 UDF。

原因: 在已分割資料庫環境中，無法使用使用者定義的函數。

動作: 請檢查訊息 SQL0443N，以決定發生錯誤的 UDF。請參閱該 UDF 區段，以取得有關如何在應用程式中處理此函數的資訊。

診斷追蹤

DB2 Extender 含有記錄 Extender 伺服器活動的追蹤機能。請遵照 IBM 服務人員的指示使用追蹤機能。

追蹤機能會在伺服器檔案中紀錄各種事件的資訊，如登錄或結束 DB2 Extender 元件，或 DB2 Extender 元件傳回的錯誤碼。因為記錄許多事件的資訊，所以應該在必要時才使用追蹤機能，例如，需要調查錯誤狀況時。此外，使用追蹤機能時，應該限制作用中的應用程式數目。限制作用中的應用程式數有助於找出發生問題的原因。

請使用 DMBTRC 命令來控制追蹤。您可在 AIX 伺服器或 Windows NT 或更新版伺服器的命令行上發出此命令。您必須具備 SYSADM、SYSCTRL 或 SYSMINT 權限才能發出此命令。

使用 DMBTRC 命令，可以：

- 啟動追蹤
- 停止追蹤
- 重新製作追蹤資訊的格式，使它易於閱讀
- 顯示追蹤狀態

啟動追蹤

您可輸入下列命令來啟動追蹤：

```
dmbtrc on path
```

其中，*path* 是含有追蹤資訊的伺服器檔案路徑。

例如，下列命令啟動追蹤：

```
dmbtrc on /tmp/trace.txt
```

停止追蹤

您可輸入下列命令來停止追蹤：

```
dmbtrc off
```

重新格式化追蹤資訊

追蹤資訊的記錄格式為二進位。您可輸入下列命令，重新製作資訊的格式，使它更易於閱讀：

```
dmbtrc format input_file output_file
```

其中 *input_file* 是包含二進位格式追蹤資訊的檔案，而 *output_file* 是將包含重新格式化資訊的檔案。*output_file* 參數是可選用的；若不指定，則重新製作格式的資訊會顯示在螢幕上。

例如，下列命令重新製作追蹤資訊的格式：

```
dmbtrc format /tmp/trace.txt /tmp/fmttrace.txt
```

顯示追蹤狀態

使用以下命令：

```
dmbtrc info
```

來顯示下列追蹤狀態資訊：

- 追蹤機能開啓或關閉
- 包含追蹤資訊的檔案路徑

附錄 A. 設定 DB2 Extender 的環境變數

當您儲存、擷取或更新影像、音效或視訊物件時，DB2 Extender 能讓您指定檔名時更有彈性。您也可以有彈性地指定程式如何顯示或播放擷取自資料庫表格的影像、音效和視訊物件。

如何使用環境變數來分辨檔名

雖然可以指定完整的檔名 (亦即完整路徑後面接著檔名) 來進行儲存、擷取及更新作業，但是建議您最好指定相對檔名。在 AIX、HP-UX 或 Solaris 中，相對檔名即為不以斜線開頭的檔名；在 Windows 中，相對檔名即為不以磁碟機字母加上冒號及反斜線當作開頭的檔名。

若指定相對檔名，則 Extender 會使用不同從屬站和伺服器環境環境中的目錄規格，來分辨檔名。這可讓檔案在主/從環境中移動，而不需變更檔名。完整的檔名在每次移動時，就需要變更。

表 17 列示和說明可設定的環境變數，以供 Image、Audio 及 Video Extender 用來分辨檔名。

表 17. DB2 Extender 的環境變數

Image Extender	Audio Extender	Video Extender	說明
伺服器環境變數			
DB2IMAGEPATH	DB2AUDIOPATH	DB2VIDEOPATH	用來分辨來源檔名稱，以便於伺服器檔案上執行儲存、擷取及更新作業。
DB2IMAGESTORE	DB2AUDIOSTORE	DB2VIDEOSTORE	用來分辨目標檔名稱，以便對伺服器檔案執行儲存和更新作業。
DB2IMAGEEXPORT	DB2AUDIOEXPORT	DB2VIDEOEXPORT	用來分辨目標檔名稱，以便對伺服器檔案執行擷取作業。
DB2IMAGETEMP			用來分辨目標檔名稱，以使用於建立暫時伺服器檔案。但是，若指定 TMP 環境變數，則使用目錄 TMP 來分辨檔名。
從屬站環境變數			
DB2IMAGEPATH	DB2AUDIOPATH	DB2VIDEOPATH	用來分辨來源檔名稱，以便在從屬站檔案上進行顯示和播放作業。

環境變數

表 17. DB2 Extender 的環境變數 (繼續)

Image Extender	Audio Extender	Video Extender	說明
DB2IMAGETEMP	DB2AUDIOTEMP	DB2VIDEOTEMP	用來分辨目標檔名稱，以使用於建立暫時從屬站檔案。但是，若指定 TMP 環境變數，則使用目錄 TMP 來分辨檔名。

若未設定適當的環境變數給特定 Extender 使用，則 Extender 會使用下列環境變數來分辨檔名：

環境變數	說明
DB2MMPATH	用來分辨來源檔名稱，以便進行儲存、擷取及更新作業。
DB2MMSTORE	用來分辨目標檔名稱，以便進行儲存和更新作業。
DB2MMEXPORT	用來分辨目標檔名稱，以便進行擷取作業。
DB2MMTEMP	用來分辨檔名，以使用於建立暫用檔。

如何使用環境變數來識別顯示或播放程式

環境變數除了用來分辨檔名之外，尚可用來識別程式，以顯示 Image Extender 所擷取的影像物件，以及播放 Audio 和 Video Extender 所擷取的音效或視訊物件。您可以使用 DBiBrowse、DBaPlay 及 DBvPlay API，來個別顯示或播放這些物件。使用每一個 API 時，您可以指定顯示或播放程式，或指出想要用預設程式來顯示或播放物件。

DB2 Extender 在從屬站中使用下列環境變數，來識別預設的顯示或播放程式：

環境變數	說明
DB2IMAGEBROWSER	用來識別預設的影像顯示程式。
DB2AUDIOPLAYER	用來識別預設的音效播放程式。
DB2VIDEOPLAYER	用來識別預設的視訊播放程式。

如何使用 DB2MMDATAPATH 環境變數 (僅適用於 EEE)

DB2 Extender 使用 DB2MMDATAPATH 環境變數來分辨位置，以供分割的資料庫環境中的各種作業使用。例如，DB2 Image Extender 使用 DB2MMDATAPATH 的值，在已分割資料庫環境中儲存 QBIC 資料。

建立 DB2 Extender 案例時即可設定 DB2MMDATAPATH，請參閱第 10 頁的『DMBICRT』及下列安裝『Readme』檔：

- aixeee 目錄中的 install.txt (在 AIX 中安裝 DB2 Extender，供 DB2 Extended Enterprise Edition 使用)
- soleee 目錄中的 install.txt (在 Solaris 作業環境中安裝 DB2 Extender，供 DB2 Extended Enterprise Edition 使用)

DB2MMDATAPATH 的用法範例，是說明儲存 QBIC 特性和索引資料。在 UNIX 中，DB2 Image Extender 將此 QBIC 資料儲存於下列目錄中：

```
db2mmdatapath /NODEnode_num/QBIC/database_name
```

其中 *db2mmdatapath* 是 DB2MMDATAPATH 環境變數的值、*node_num* 是節點號碼、*database_name* 是資料庫名稱。

請看下列 AIX 範例。假設 DB2MMDATAPATH 設為 /localfs/dmbdata。也假設一個名為 sample 的資料庫在節點 0、2 及 5 上被分割。該 sample 資料庫的 QBIC 資料，將儲存在下列目錄中：

Node 0: /localfs/dmbdata/NODE0000/QBIC/sample

Node 2: /localfs/dmbdata/NODE0002/QBIC/sample

Node 5: /localfs/dmbdata/NODE0005/QBIC/sample

設定環境變數

您可以在 AIX、HP-UX、Solaris、OS/2 及 Windows 中設定環境變數。

在 AIX、HP-UX、Solaris 伺服器與從屬站中設定環境變數

在 AIX、HP-UX 與 Solaris 中，環境變數指定於 C shell、Korn shell 與 Bourne Shell Script 中。安裝 DB2 Extender 之後，會設定伺服器的環境變數為：

C shell

```
setenv DB2MMPATH /usr/lpp/db2ext/samples:/tmp
setenv DB2MMTEMP /tmp
setenv DB2MMSTORE /tmp
setenv DB2MMEEXPORT /tmp
```

Korn 和 Bourne shell

```
DB2MMPATH=/usr/lpp/db2ext/samples:/tmp
export DB2MMPATH
```

```
DB2MMSTORE=/tmp
export DB2MMSTORE
```

環境變數

```
DB2MMEXPORT=/tmp
export DB2MMEXPORT
```

```
DB2MMTEMP=/tmp
export DB2MMTEMP
```

起始伺服器的環境變數是設定為可讓您存取媒體檔的值，這些檔案與 DB2 Extender 一起分送的範例程式一起使用。(關於範例程式和媒體檔案的資訊，請參閱 第 549 頁的附錄 B, 『範例程式和媒體檔案』。)

在 AIX、HP-UX 或 Solaris 從屬站中安裝 DB2 Extender 時，會設定下列從屬站環境變數：

C shell

```
setenv DB2MMPATH /tmp
setenv DB2MMTEMP /tmp
```

Korn 和 Bourne shell

```
DB2MMPATH=/tmp
export DB2MMPATH
```

```
DB2MMTEMP=/tmp
export DB2MMTEMP
```

設定用來分辨檔名的伺服器 and 從屬站環境變數。指定適用於自己環境的值。對於以 `PATH` 結尾的環境變數，您可以使用定界符號來指定多個目錄。以 `STORE`、`EXPORT` 及 `TEMP` 結尾的環境變數，只能設定一個目錄。

分別在 `DB2IMAGEBROWSER`、`DB2AUDIOPLAYER` 及 `DB2VIDEOPLAYER` 從屬站環境變數中，設定影像顯示、音效播放及視訊播放程式的適當名稱。

您可以變更環境變數的起始設定值，如下所示：

C shell

使用 `SETENV` 命令來設定環境變數：

```
setenv env-var directory
```

例如：

```
setenv DB2MMPATH /usr/lpp/db2ext/samples:/media
setenv DB2IMAGEPATH /employee/pictures:/images
setenv DB2AUDIOSTORE /employee/sounds
setenv DB2IMAGEBROWSER 'xv %s'
```

Bourne shell

使用 `EXPORT` 命令來設定環境變數：

```
env-var=directory
export env-var
```

例如：

```
DB2MMPATH=/usr/lpp/db2ext/samples:/media
export DB2MMPATH
```

```
DB2IMAGEPATH=/employee/pictures:/images
export DB2IMAGEPATH
```

```
DB2AUDIOSTORE=/employee/sounds
export DB2AUDIOSTORE
```

Korn shell

使用 `EXPORT` 命令來設定環境變數：

```
export env-var=directory
```

例如：

```
export DB2MMPATH=/usr/lpp/db2ext/samples:/media
export DB2IMAGEPATH=/employee/pictures:/images
export DB2AUDIOSTORE=/employee/sounds
```

在 Windows 伺服器與從屬站中設定環境變數

在 Windows 中，設定環境變數的方式，乃根據 DB2 Extender 的使用是在非分割環境中或已分割資料庫環境中而定 (亦即，使用 DB2 Extended Enterprise Edition for Windows)。

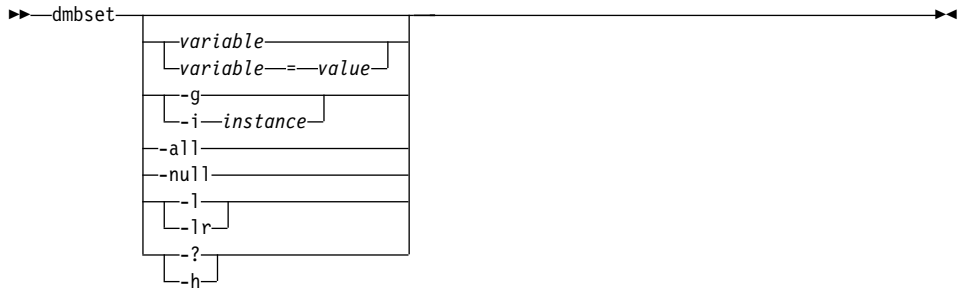
在 Windows 非已分割資料庫環境中設定環境變數 (僅適用於非 EEE)

在 Windows 中，環境變數儲存於系統登記中。開啓 Windows 控制台，並選取系統圖示，即可設定變數。從「系統性質」對話中，選取「環境」標籤。有兩個視窗包含環境變數及其值。頂端視窗顯示對所有使用者皆有效的變數。底端視窗顯示只對現行使用者有效的變數。

在 Windows 已分割資料庫環境中設定環境變數 (僅適用於 EEE)

在 Windows 分割的環境中，DB2 Extender 所用的所有變數，皆儲存於系統登錄的專用區域內。提供程式 `DMBSET` 來檢查和變更 Extender 變數。

該程式的語法如下：



欲查詢變數的值，請鍵入 `dmbset variable_name`。例如：

```
dmbset DB2MMPATH
```

欲設定變數的值，請鍵入 `dmbset variable_name=value`。例如：

```
dmbset DB2MMPATH=C:\DMB\SAMPLES
```

欲顯示已定義案例的所有變數值，請鍵入 `dmbset -i instance_name`。例如：

```
dmbset -i dmbinst1
```

欲設定一個 NULL 值，請鍵入 `mbset variable_name -null`。例如：

```
dmbset DB2MMPATH -null
```

若要顯示所有案例使用的變數值，請鍵入：`dmbset -g`

若要列出 DB2 Extender 使用的所有變數名稱，請鍵入：`dmbset -lr`

若要列出定義於此登記的所有案例設定檔名稱，請鍵入：`dmbset -l`

在分割的資料庫環境中設定 DB2 Extender 的環境變數時，有很大的彈性空間。例如，除了 DB2MMDATAPATH 之外，您可以使用下列格式來指定任何環境變數的值：

- 「通用命名慣例」(UNC) 名稱：`\\machine_name\share_name`。例如：
`\\harmony\JimsShr`
- 磁碟機:路徑。例如：
`f:\media`
- 其它格式：`share_name\directory_name`。例如：
`JimsShr\images`

附錄 B. 範例程式和媒體檔案

DB2 Extender 包括的檔案，有各種不同的範例程式。範例程式使用影像、音效及視訊檔案，這些檔案亦由 Extender 提供。大部份範例程式皆以 C 語言撰寫。所有 C 範例程式皆為「呼叫層次介面」(CLI) 格式。亦提供數個 Java 範例程式及一個 Net.Data 範例巨集檔。

安裝 DB2 Extender 時，範例程式安裝於目標目錄的 SAMPLES 子目錄中。安裝 DB2 Extender 時，影像、音效及視訊檔亦安裝於目標目錄的 SAMPLES 子目錄中。在安裝期間，Extender 環境變數會設定為指向目標目錄的範例子目錄。

範例程式

組成 DB2 Extender 範例程式的檔案，包括：

檔案	說明
enable.c	啟用 Audio、Image 及 Video Extender 所用的資料庫、建立表格、啟用表格及其直欄。
populate.c	將資料匯入表格中(程式為 C 格式)
Populate.java	將資料匯入表格中 (程式為 Java 格式)
query.c	查詢表格中的資料(程式為 C 格式)
Query.java	查詢表格中的資料 (程式為 Java 格式)
api.c	使用 Extender API 來查詢資料庫
handle.c	示範控點在 UDF 中的用法，以及 SELECT 陳述式中製作 where 子句比較的用法。
qbcatdemo.c	建立 QBIC 型錄，以及將影像直欄編目至型錄內
qbicdemo.c	查詢 QBIC 型錄
color.c	製作 qbicdemo.c 的色表宣告
QbicQry.java	呈現 QBIC 查詢的平均顏色和直方圖顏色選取元
makesf.c	建立 makehtml.exe 所用的快照型錄檔。
makehtml.c	存取快照型錄及建立 Web 瀏覽器所顯示的 HTML 網頁
storybrd.java	makehtml.c 所建立的 HTML 頁面呼叫時、可顯示快照的 applet

範例程式

utility.c	公用常式
utility.h	公用常式的標頭檔
makefile.aix	在 AIX 中建立程式所用的 Makefile
makefile.os2	在 OS/2 中建立程式所用的 Makefile
makefile.iva	在 Windows NT (或以上版本) 中以 IBM VisualAge C++ 來建立程式時所用的 Makefile
makefile.mvc	在 Windows 中以 Microsoft Visual C++ 建立程式所用的 Makefile
makefile.sun	在 Solaris 中建立程式所用的 Makefile
makefile.hp	在 HP-UX 中建立程式所用的 Makefile

下列範例程式有提供可執行檔。這些範例程式是以顯示的次序來執行。

1. Enable
2. Populate
3. Query
4. API
5. Handle
6. Qbcatdmo
7. Qbicdemo
8. QbicQry
9. Makesf
10. Makehtml

範例 JAVA 程式提供一些可執行的類別檔案 (Populate.class、Query.class、QbicQry.class 及 storybrd.class)。

在執行範例程式之前，您應該在伺服器上建立資料庫。伺服器上也應該已啟動 Extender 服務程式。欲執行範例程式，請鍵入程式名稱 (如此可啟動的可執行檔)。您必須輸入資料庫名稱、使用者 ID 和密碼。請使用建立該資料庫的使用者 ID 及其密碼。

您也可以建置自己的範例程式可執行檔。所需作法如下：

1. 將範例程式複製到可寫入的目錄內。
2. 編輯 makefile 來指定系統上安裝 DB2、Extender 及編譯程式的位置。
3. 使用 make 或 nmake 將檔案編譯成可執行的程式。

關安裝及使用範例程式的進一步資訊，請參閱範例程式目錄中的 README.CNT 檔案。

範例影像、音效及視訊檔案

DB2 Extender 提供的範例影像、音效及視訊檔如下：

- 影像檔
 - lizzi.bmp
 - sws_stri.bmp
 - nitecry.bmp
 - ranger_r.bmp
 - fuzzblue.bmp
- 音效檔
 - lizzi.wav
 - sws_stri.wav
 - nitecry.wav
 - ranger_r.wav
 - fuzzblue.wav
- 視訊檔
 - nitecry.avi
 - sample.mpg

範例 Net.Data 巨集檔

DB2 Extender 包含一個名為 `extender.d2w` 的 Net.Data 巨集檔。Web 伺服器執行巨集檔時，巨集檔會執行 SQL 陳述式來呼叫 DB2 Extender UDF。Web 瀏覽器顯示巨集檔傳回的結果。如第 552 頁的圖 30 所示，每一頁結果亦顯示產生該結果的 SQL 陳述式。第 553 頁的圖 31 顯示範例 Net.Data 巨集檔的內容。

若要執行範例 Net.Data 巨集檔，請在 Web 瀏覽器中輸入下列 URL：`http://your server/cgi-bin/db2www/extender.d2w/startHere`

其中 *your server* 是指您的 Web 伺服器名稱。

```
select cast(mmdbsys.thumbnail(covers) as blob(10000), cast(mmdbsys.thumbnail(video)
blob(3000)), mmdbsys.comment(music), artist, title, price, stock_no from sobay_catalog
```

Cover	Video	Audio	Artist	Title	Price
		[Listen]	Lizzi	Decisions	25.00
		[Listen]	SWS Strings	Vivaldi: Four Seasons	25.50
		[Listen]	Nitecry	Run for Cover	15.00
		[Listen]	Ranger Rick	Handy Sue	12.25
		[Listen]	Fuzzy Blues	Aurora	22.00

圖 30. 執行範例 *Net.Data* 巨集檔的 Web 應用程式。每一頁結果會顯示產生該結果的 SQL 陳述式。

```

%{ ----- }
%{ Copyright International Business Machines Corporation, 1998. }
%{ All rights reserved. }
%{ }
%{ 範例 Net.Data 巨集，顯示如何呼叫影像、音效及視訊 }
%{ Extender UDF。 }
%{ }
%{ 執行前，請將此巨集放在您的 MACRO_PATH Root 中，確定 tmplobs }
%{ 目錄存在於 web 伺服器的 document root 之下，並建立 }
%{ 資料庫，供執行 Extender 範例程式 }
%{ 'enable' 及 'populate' 時使用。執行 'enable' 及 'populate'。若您的 }
%{ 資料庫名稱不是為 'testdb2'，則需要變更 }
%{ 下列 DATABASE 的定義。Extender 環境變數 }
%{ DB2MEXPORT 需要設定，讓 Net.Data 所用的案例指向 }
%{ web 伺服器的 <document root>/tmplobs 目錄。然後重新啟動 DB2 }
%{ 及 Extender，使變數生效。 }
%{ 若未執行 Net.Data 的 Connection Manager，您就必須要 }
%{ 提供 LOGIN 及 PASSWORD 給資料庫。若您不熟悉這些指示， }
%{ 請至下列網址參考 Net.Data 文件： }
%{ http://www.software.ibm.com/data/netdata/docs (或 Extender 範例程式的 }
%{ Extender 文件)。 }
%{ }
%{ 若要不要顯示 SQL 陳述式，請將下列 SHOWSQL 的值 }
%{ 變更為 "no"。 }
%{ ----- }

%{ ----- }
%{ Definitions section }
%{ ----- }
%define{
    DATABASE="testdb2"
    SHOWSQL="yes"
%}

```

圖 31. Net.Data 範例巨集檔 (1/5)

範例媒體檔案

```
%{ ----- %}
%{ SQL functions %}
%{ ----- %}

%function (DTW_SQL) startHereSQL(){
    select artist, title, stock_no, price from sobay_catalog

    %REPORT{
        <table border="2" bgcolor="#b1b1b1">
            <tr><th>Artist <th> Title <th> Stock<th>
Number <th> Price </tr>
            %ROW{ <tr><td> $(V_artist) <td> $(V_title) <td>
$(V_stock_no) <td> $(V_price) <tr>
            %}
        </table>
    %}
%}

%function (DTW_SQL) addThumbsSQL(){
    select cast(mmdbsys.thumbnail(covers) as blob(10000)),
        cast(mmdbsys.thumbnail(video) as blob(3000)),
        mmdbsys.comment(music), artist, title, price, stock_no
    from sobay_catalog

    %REPORT{
        <table border="2" bgcolor="#b1b1b1">
            <tr><th>Cover <th>Video <th>Audio
<th>Artist <th>Title <th>Price </tr>
            %ROW{ <tr><td>< a
href="showCover?stock_no=$(V_stock_no)"></a>
            <td>< a href="getVideo?stock_no=$(V_stock_no)"></a>
            <td>< a
href="getAudio?stock_no=$(V_stock_no)&filename=$V3">[Listen]</a>
            <td> $(V_artist) <td> $(V_title) <td> $(V_price)
        </tr>
            %}
        </table>
    %}
%}

%function (DTW_SQL) showCoverSQL(){
    select cast(mmdbsys.content(covers, 'GIF') as blob(150000)),
    mmdbsys.format(covers)
    from sobay_catalog
    where stock_no = '$(stock_no)'

    %REPORT{
        %ROW{  <br><br><b>Original image
format: $(V2)</b>%}
        %}
    %}
```

圖 31. Net.Data 範例巨集檔 (2/5)

```
%{ 下列 Content 呼叫依賴於適當地設定 DB2MMEXPORT ,
    才能指向 web 伺服器文件根目錄下的 tmplobs 目錄。%}

%function (DTW_SQL) showVideoSQL(){
    select mmdbsys.comment(video), mmdbsys.content(video, mmdbsys.comment(video), 1),
        mmdbsys.format(video)
    from sobay_catalog
    where stock_no = '$(stock_no)'

    %REPORT{
        %ROW{ <a href="/tmplobs/$(V1)"><i><b> Play Video Clip</b></i></a>
            <br><br><b>Format: $(V3) <br>(Note: NT/Win95 may not come with
                a decompressor<br>for this video format.)</br>
        %}
    %}
%}

%{ 下列 Content 呼叫依賴於適當地設定 DB2MMEXPORT ,
    才能指向 web 伺服器文件根目錄下的 tmplobs 目錄。%}

%function (DTW_SQL) showAudioSQL(){
    select mmdbsys.comment(music), mmdbsys.content(music, mmdbsys.comment(music), 1),
        mmdbsys.format(music)
    from sobay_catalog
    where stock_no = '$(stock_no)'

    %REPORT{
        %ROW{ <a href="/tmplobs/$(V1) "><i><b>Play Audio Clip</b></i></a>
            <br><br><b>Format: $(V3)</b>
        %}
    %}
%}
```

圖 31. Net.Data 範例巨集檔 (3/5)

範例媒體檔案

```
%{ ----- %}
%{ HTML 區段 %}
%{ 例如, http://<your server>/cgi-bin/db2www/extender.d2w/startHere %}
%{ ----- %}
%{ 例如, http://
%{ 例如, http://
%HTML(startHere){
<html>
  <head><title>UDB Extender 巨集範例：簡式橫列報表
</title></head>
  <body bgcolor="#ffffff">
    <font color="#3300ff" size="3"><b>若下方未出現資料，您可能需要
    執行 UDB Extender 範例程式 <i>enable</i> 及 <i>populate</i>。
    extender.d2w 巨集的第一個 HTML 區段，只是擷取
    UDB Extender 範例資料庫中所有橫列的所有傳統資料。
    %if ( "${SHOWSQL}" == "yes" || "${SHOWSQL}" == "YES" )
    <br><br> 根據預設值，此巨集產生的每一頁中顯示建立
    此頁所用的 SQL。這裡列出此頁的 SQL 陳述式：
    %else
    <br>
    %endif
    </b></font>
    <br>@startHereSQL()
    <br><b>按一下< a href="addThumbs"><i>此處 </i></a>，顯示
    影像/音效/視訊資料的縮圖及鏈結。</b>
  </body>
</html>
%}

%HTML(addThumbs){
<html>
  <head><title>UDB Extender 巨集範例：新增縮圖
</title></head>
  <body bgcolor="#ffffff">
    <font color="#3300ff" size="3"><b>此頁增加專輯封面縮圖
    及鏈結，以顯示資料庫的多媒體內容。若要存取
    多媒體內容：
    <ul>
      <li> 按一下 CD 封面縮圖，以檢視原來大小的圖片
      <li> 按一下 "視訊縮圖"，以檢視視訊
      <li> 按一下 "[收聽]" 鏈結，以收聽音效片段
    </ul>
    </b></font> @addThumbsSQL()
    <br><b>按一下< a href="startHere"><i>此處
    </i></a>，回到首頁。</b>
  </body>
</html>
%}
```

圖 31. Net.Data 範例巨集檔 (4/5)


```

%HTML(showCover){
<html>
  <head><title>UDB Extender 巨集範例：項目封面 $(stock_no)</title></head>
  <body bgcolor="#ffffff">
    <font color="#3300ff" size="3"><b>此頁中，巨集取得全形封面
影像，並將影像轉換成瀏覽器可顯示的 GIF 格式：
    </b></font><br><br>
    <table width="400" border="2" bgcolor="#b1b1b1" cellpadding="5">
      <tr><td align=center> @showCoverSQL()
      <tr><td align=center> <b>庫存號碼：$(stock_no)</b>
    </table>
    <br><b>回到<a href="addThumbs"><i>前頁 </i></a>.</b>
  </body>
</html>
%}

%HTML(getVideo){
<html>
  <head><title>UDB Extender 巨集範例：項目的視訊片段
$(stock_no)</title></head>
  <body bgcolor="#ffffff">
    <font color="#3300ff" size="3"><b>此頁中，您可以觀賞視訊片段：
    </b></font><br><br>
    <table width="400" border="2" bgcolor="#b1b1b1" cellpadding="5">
      <tr><td align=center> @showVideoSQL()
      <tr><td align=center> <b>庫存號碼：$(stock_no)</b>
    </table>
    <br><b>回到<a href="addThumbs"><i>前頁 </i></a>.</b>
  </body>
</html>
%}

%HTML(getAudio){
<html>
  <head><title>UDB Extender 巨集範例：項目的音效片段
$(stock_no)</title></head>
  <body bgcolor="#ffffff">
    <font color="#3300ff" size="3"><b>您可以從此頁收聽音效片段：
    </b></font><br><br>
    <table width="400" border="2" bgcolor="#b1b1b1" cellpadding="5">
      <tr><td align=center> @showAudioSQL()
      <tr><td align=center> <b>庫存號碼：$(stock_no)</b>
    </table>
    <br><b>回到<a href="addThumbs"><i>前頁 </i></a>.</b>
  </body>
</html>
%}

```

圖 31. Net.Data 範例巨集檔 (5/5)

注意事項

本資訊是針對 IBM 在美國所提供之產品與服務開發出來的。而在其它國家中，IBM 不見得有提供本書中所提的各項產品、服務或功能。要知道在您所在地區是否可用到這些產品與服務時，請向當地的 IBM 服務代表查詢。本書在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，其他非 IBM 產品、程式、或服務在運作上的評價與驗證，其責任屬於使用者。

本文件中包含著 IBM 所擁有之專利或暫准專利。使用者不因取得本書而享有本書內容之專利權。您可以用書面方式來查詢授權，來函請寄到：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

若要查詢有關二位元組 (DBCS) 資訊的特許權限事宜，請聯絡當地國家的 IBM 智慧財產部門，或者用書面信函寄到：

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

下列段落不適用於英國，或與該國之法律條款抵觸，即不適用：IBM 以『交附時之現況』供應本書，而不提供任何明示或默示之保證，包括但不限於如默示保證之適售性或符合客戶之特殊使用目的；某些地區不允許在某些交易行為中，不作明示或暗示的保證，在此情況下，本聲明亦不適用。

本資訊可能會有技術上或排版印刷上的訛誤。因此，IBM 公司會定期修訂，並將修訂後的內容納入新版中。同時，IBM 會隨時改進並 (或) 變動本書中所提及的產品及 (或) 程式，而不另行通知。

本程式之獲授權者若希望取得相關資料，以便使用下列資訊者可洽詢 IBM。其下列資訊指的是：(1) 獨立建立的程式與其他程式 (包括此程式) 之間更換資訊的方式 (2) 相互使用已交換之資訊方法若有任何問題請聯絡：

IBM Corporation

J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於雙方之 [IBM 客戶合約、 IBM International Program 授權合約或任何同等合約中的條款約定，提供本書中所說的授權程式與其所有適用的授權材料。

本書僅供規劃使用。因此，在上述產品上市之前若有變更，這些資訊亦會隨著變更。

本書中含有日常商業活動所用的資料及報告範例。爲了盡可能詳細，範例中涵蓋了個人、公司、品牌及產品的名稱。所有名稱純屬虛構，如有雷同純屬巧合。

著作權授權：

本書包含原始語言的範例應用程式，用以說明各種作業平台上的程式設計技術。您可以基於研發、使用、銷售或散布符合作業平台應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式而不必向 IBM 付費。這些範例應用程式並未經過徹底的測試，因此，IBM 不提供明示或默示其可靠性、有用性或符合特定效用的保證。

這些範例程式或是任何衍生著作的每一份拷貝或任何部份，都必須具有下列的著作權聲明：

© (您的公司名稱) (年份)。Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _請在此輸入年份或年數。 All rights reserved.

當您檢視本書的軟本時，可能不會顯示照片與彩色圖例。

程式設計介面資訊

本出版品旨在協助您管理 DB2 Extenders 以及開發搭配 DB2 Extender 使用的程式。本出版品記載 DB2 Extender 所提供的「一般使用程式設計介面」與「相關指引」等資訊。

一般使用程式設計介面可讓您撰寫程式，以獲取 DB2 Extender 的服務。您可以將您發展的應用程式所需的 DB2 Extender 執行時間特性複製到從屬站或伺服器機器上。若要安裝執行時間特性，請參閱 DB2 Extender CD-ROM 上適用您作業系統的 README.TXT 檔案中所提供的安裝指示。

商標

下列專有名詞是 IBM Corporation 在美國及 (或) 其它國家的商標：

AIX	DB2 Universal Database	PS/2
DB2	IBM	QBIC
DB2 Extenders	OS/2	VisualAge

Microsoft、Windows、Windows NT 以及 Windows 商標是 Microsoft Corporation 在美國及 (或) 其它國家的商標或註冊商標。

UNIX 是 The Open Group 在美國及其它國家的註冊商標。

Intel 是 Intel 的註冊商標。

其它公司、產品或服務名稱可能是其它公司的商標或服務標誌。

名詞解釋

管理支援表 (administrative support table). DB2 擴充元用來處理使用者向影像檔、音效和視訊物件提出要求的其中一種表格。有些管理支援表識別為擴充元啓用的使用者表格和直欄。有些管理支援表含有關於啓用的直欄中的物件之屬性資訊。亦稱為中間資料表。

分析 (analyze). 計算映像檔的特性的數值以及將該數值新增至 QBIC 型錄。

API. 請參閱應用程式設計介面。

應用程式設計介面 (application programming interface, API).

(1) 作業系統或可單獨購買的授權程式所提供的功能介面。API 可讓使用高階語言撰寫的應用程式使用作業系統或授權程式的特定資料或函數。

(2) 在 DB2 中，介面內的函數。例如，取得錯誤訊息 API。

(3) DB2 Extender 提供 API 來要求使用者定義的函數、管理作業、顯示作業和影像場景變更偵測。

音效 (audio). 有關錄製資訊中可聽到的部份。

平均顏色 (average color). 一種顏色測量，它被計算成內含於映像檔圖點的顏色值平均值。

音效剪輯 (audio clip). 一段錄音音效材料。

二進位大型物件 (BLOB). 長度可達 2 GB 的二進位字串。映像檔、音效和影像物件以 BLOB 形式儲存在 DB2 資料庫。

字元大型物件 (CLOB). 單一位元組字串，字串可多達 2 GB。CLOB 有一個相關的字碼頁。含有單一位元組字元的文字物件以 CLOB 形式儲存在 DB2 資料庫。

粗糙度 (coarseness). 材質的一種屬性，可測量材質的質地 (質地的顆粒大小及粗細)。

對比 (contrast). 材質的一種屬性，它指花紋的明亮度，是灰階直方圖的變異函數。

資料庫分割區. 資料庫的一部份，它是由自己的使用者資料、索引、架構檔和異動日誌組成。有時稱為節點或資料庫節點。

資料庫分割區伺服器. 管理資料庫分割區。資料庫分割區伺服器包含一個資料庫管理程式和資料集合以及它管理的系統資源。一般而言，會對每一個機器指定一個資料庫分割區伺服器。

DB2 extender. 一組程式中的一個程式，您可使用該程式儲存和擷取傳統數值和字元資料以外的資料類型，例如映像檔、音效、影像資料以及複合文件。

方向性. 材質的一種屬性，它說明影像是否有偏好的方向 (如草) 或是像平滑物件一樣 (如玻璃)。

分解. 隨著下一個影像框的信號強度增強而減少上一個影像框的信號強度。

特殊類型. 請參閱使用者定義類型。

雙位元組大型物件 (DBCLOB). 一個雙位元組字串或單一位元組和雙位元組字元組合，其中字串可多達 2 GB。DBCLOB 有一個相關的字碼頁。含有雙位元組字元的文字物件以 DBCLOB 形式儲存在 DB2 資料庫。

環境變數. 用來說明 DB2 Extenders 作業環境及對此環境的值提供預設值的一個變數。

擴充元. 請參閱 DB2 Extender。

特性. 影像的一種視覺屬性，例如平均顏色。

檔案參考變數。 一個程式設計變數，它對於在從屬工作站的檔案中來回移動 LOB 來說很有用。

十億位元組 (GB)。 十億 (10^9) 位元組。若指記憶體容量，則為 1 073 741 824 位元組。

控點。 擴充元建立的一個字串，用來代表表格中的映像檔、音效或影像物件。物件的控點是儲存在使用者表格和管理支援表中。以此方式，**extender** 可鏈結儲存在使用者表格的控點以及關於儲存在管理支援表的物件資訊。

直方圖顏色。 影像中不同顏色的測量。每一種顏色的資料分別儲存在 **QBIC** 型錄。

主變數。 應用程式中的一個變數，在內含的 SQL 陳述式中可以參照該變數。主變數是在資料庫和應用程式工作區之間傳輸資料的主要機制。

影像。 圖片的電子呈現。

索引檔。 一個檔案，含有 Video Extender 用來在視訊短片中尋找鏡頭或個別畫面的索引資訊。

案例。 邏輯 DB2 擴充元伺服器環境。在同一個工作站中可以有數個 DB2 Extender 伺服器案例，但每一個 DB2 案例只能有一個案例。您可使用這些案例執行下列動作：

分隔開發環境與生產環境

敏感資訊僅限於特定人群存取。

千位元組 (KB)。 一千 (10^3) 位元組。若指記憶體容量，則為 1024 位元組。

大型物件 (LOB)。 一連串位元組，其長度最多 2 GB。LOB 可以有三種類型：二進位大型物件 (BLOB)、字元大型物件 (CLOB) 或雙位元組大型物件 (DBCLOB)。

LOB 定位器。 儲存在主變數的一個小值 (4 位元組)，在程式中可使用這個值參照 DB2 資料庫中的大型 LOB。使用 LOB 定位器時，使用者可操作 LOB 好像它是儲存在一般主變數，而且不必在用戶端機器和資料庫伺服器上的應用程式之間傳送 LOB。

百萬位元組 (MB)。 一百萬 (10^6) 位元組。若指記憶體容量，則為 1 048 576 位元組。

中間資料表。 請參閱管理支援表。

多分割區節點群組。 一個節點群組，它含有一個以上的資料庫分割區伺服器。

節點。 在資料庫分割中，與資料庫分割區同義。

nodegroup (節點群組)。 已命名的一個或多個資料庫分割區群組。

物件。 在物件導向程式設計中，由資料以及與該資料相關的作業組成的抽象物。

物件方向。 一種程式設計方法，在本方法中應用程式中的任何項目 (真實或抽象) 都可以顯示成由一組作業和資料值組成的物件。例如，文件可以使用文件物件來代表，該物件由文件資料和在文件上執行的作業所組成，這些作業包括歸檔、傳送及列印。視訊片段可以由視訊物件來代表，該物件由視訊資料和作業組成，這些作業包括播放視訊片段或尋找特定視訊框。

分割的資料庫。 具有兩個或更多個資料庫分割區的資料庫。使用者表格中的資料可放在一個或多個資料庫分割區。多個分割區有同一個表格時，該表格的部份橫列會儲存在一個分割區而其它橫列是儲存在其它分割區。

圖點。 螢幕可顯示的影像的最小元素。

位置顏色。 影像中某個指定區域的圖點之平均顏色值。

Query by Image Content (QBIC)。 Image Extender 提供的一種功能，可讓使用者根據影像的視覺特性 (例如平均顏色和材質) 來搜尋影像。

QBIC 型錄。 一個儲存庫，它儲存關於影像視覺特性的資料。

查詢物件 (query object). 一個指定 QBIC 查詢的特性、值和特性比重的物件。您可以在 QBIC 查詢中命名及儲存此物件供以後使用。與查詢字串相對照

query_string. 一個指定 QBIC 查詢的特性、值和特性比重的字串。您可從 DB2 命令行在查詢中輸入此查詢字串。與查詢物件相對照

調整比例. 將節點新增至資料庫以增加儲存體空間和效能。

場景變更. 視訊短片中的一個點，在這個點中兩個連續畫面有明顯不同。例如，在記錄影像時若照相機變更焦點，則發生上述狀況。

分數. 一個計算值，該值反映特性值與在查詢中根據影像內容指定的值的類似程度。此數字愈高，就愈相符。此分數是根據影像內容排序查詢結果。

照片. 兩個場景變更之間的畫面。

照片型錄. 一種資料庫表格或檔案，它是用來將照片的資料 (例如照片的開始和結束框號碼) 儲存在視訊短片。使用者可透過 SQL 查詢存取表格的檢視畫面，或存取檔案中的資料。

故事板 (storyboard). 影像的視覺化摘要。Video Extender 中有一些特性，可以用來識別及儲存代表影像中圖像的影像框。您可使用這些代表框建置 storyboard。

萬億位元組 (terabyte). 一萬億 (10^{12}) 位元組。10 的 12 次方位元組。若指記憶體容量，則為 1 099 511 627 776 位元組。

材質. 您可以根據影像內容在查詢中使用的其中一個特性。它指影像的粗糙度、對比或方向。

縮圖. 小型影像。

觸發函式. 變更表格時要執行的一組動作的定義。您可使用觸發函式執行一些動作，例如驗證輸入資料，為新插入的橫列自動產生一值，讀取其它表格以便交互參照，或寫入其它表格以便審核。通常使用觸發函式檢查完整性或強制執行商務規則。

UDF. 請參閱使用者定義的函數。

UDT. 請參閱使用者定義類型。

使用者定義的函數 (UDF). DB2 的使用者定義的一種函數。一旦定義之後，函數即可使用於 SQL 查詢和視訊物件中。例如，您可建立 UDF 以取得影像的壓縮格式或傳回音效的取樣率。如此一來，即可定義特殊類型的物件之行爲。

使用者定義的類型 (UDT). DB2 使用者定義的一種資料類型。UDT 是用來區別 LOB。例如，您可以為影像物件建立一個 UDT，為音效物件建立另一個 UDT。雖然儲存為 BLOB，影像和音效物件仍被視為不同於 BLOB 的特殊類型，且彼此亦互不相同。

視訊. 有關記錄資訊中可見的部份。

視訊短片. 影片或錄影帶的一小段。

影像索引. Video Extender 用來在視訊短片中尋找特定 shot 或畫面的檔案。

索引

索引順序以中文字，英文字，及特殊符號之次序排列。

〔一劃〕

一致性測試 (視訊場景變更) 24

〔二劃〕

二進位大型物件 (BLOB) 52

安全性 61

更新 134

將物件儲存為 113

復原 61

說明 52

〔三劃〕

大型物件 (LOB) 52

傳輸 100

說明 52

播放 141

顯示 141

已分割資料庫 59

說明 59

〔四劃〕

分解測試臨界值 24

分數, 影像 (QBIC) 174

方向性 58

〔五劃〕

可調整性 61

平行處理 61

說明 61

平均色 57

特性名稱 152

說明 57

〔六劃〕

光度測定 (影像反轉) 106

回覆碼 509

回覆碼 (SQLSTATE) 510

字元大型物件 (CLOB) 52

字串, QBIC 查詢 163

存取專用權 61

安全性 61

自動編目設定 (QBIC) 151

〔七劃〕

伺服器 3

多重案例 6

取得狀態 5

取得資料庫的狀態 6

停止資料庫 5

啟動 3

啟動資料庫 5

連接資料庫 3

伺服器上的管理命令 9

DMBICRT 10

DMBIDROP 12

DMBILIST 13

DMBIMIGR 14

DMBSTART 15

DMBSTAT 17

DMBSTOP 18

伺服器案例 6

列出 6

建立 6

執行 7

移除 7

移轉 8

設定 7

伺服器檔案 100

更新自 133

物件傳輸是在表格和 100

傳輸物件到 101

儲存自 112

伺服器檔案 (繼續)

擷取至 122

呈現影像的位元數 106

快照 20

說明 20

儲存 33

擷取中 26

快照型錄 58

建立 32

連線控點 32

說明 58

快照型錄的 SQLConnect 呼叫 32

快照型錄的連線控點 32

改寫指示符 122

更新物件 126

〔八劃〕

依影像內容查詢 (QBIC) 57

步驟 147

型錄 57

QbAddFeature API 416

QbCatalogColumn API 418

QbCatalogImage API 420

QbCloseCatalog API 422

QbCreateCatalog API 423

QbDeleteCatalog API 425

QbGetCatalogInfo API 427

QbListFeatures API 429

QbOpenCatalog API 431

QbQueryAddFeature API 433

QbQueryCreate API 435

QbQueryDelete API 437

QbQueryGetFeatureCount API 438

QbQueryGetString API 440

QbQueryListFeatures API 442

QbQueryNameCreate API 444

QbQueryNameDelete API 446

QbQueryNameSearch API 448

QbQueryRemoveFeature API 450

QbQuerySearch API 452

依影像內容查詢 (QBIC) (繼續)

QbQuerySetFeatureData API 454
 QbQuerySetFeatureWeight
 API 456
 QbQueryStringSearch API 458
 QbReCatalogColumn API 460
 QbRemoveFeature API 462
 QbSetAutoCatalog API 464
 QbUncatalogImage API 466

使用者定義函數 53

名稱 53
 函數路徑 54
 參照 187
 超載 54
 說明 53
 標記 54
 AlignValue 191
 AspectRatio 193
 BitsPerSample 194
 BytesPerSec 195
 Comment 196
 CompressType 198
 Content 199
 DB2Audio 204
 DB2Image 207
 DB2Video 211
 Duration 215
 Filename 216
 FindInstrument 217
 FindTrackName 218
 Format 219
 FrameRate 220
 GetInstruments 221
 GetTrackNames 222
 Height 223
 Importer 224
 ImportTime 225
 MaxBytesPerSec 226
 NumAudioTracks 227
 NumChannels 228
 NumColors 229
 NumFrames 230
 NumVideoTracks 231
 QbScoreFromName 232
 QbScoreFromStr 234
 QbScoreTBFromName 235

使用者定義函數 (繼續)

QbScoreTBFromStr 237
 Replace 239
 SamplingRate 242
 Size 243
 Thumbnail 244
 TicksPerQNote 246
 TicksPerSec 247
 Updater 248
 UpdateTime 249
 Width 250

使用者定義類型 (UDT) 52

名稱 53
 說明 52

併入檔 99

說明 99
 dmbaudio.h 99
 dmbimage.h 99
 dmbqbapi.h 99
 dmbshot.h 100
 dmbvideo.h 100

函數路徑 54

命令 469

ADD QBIC FEATURE 470
 CATALOG QBIC COLUMN 471
 CLOSE QBIC CATALOG 472
 CONNECT 473
 CREATE QBIC CATALOG 474
 DELETE QBIC CATALOG 476
 DISABLE COLUMN 477
 DISABLE DATABASE 478
 DISABLE TABLE 479
 DISCONNECT SERVER AT
 NODENUM 480
 DISCONNECT SERVER FOR
 DATABASE 481
 DISCONNECT SERVER FOR
 DATABASE AT
 NODENUM 482
 DMBICRT 10
 DMBIDROP 12
 DMBILIST 13
 DMBIMIGR 14
 DMBSTART 15
 DMBSTAT 17
 DMBSTOP 18

命令 (繼續)

ENABLE COLUMN 483
 ENABLE DATABASE 484
 ENABLE TABLE 485
 GET EXTENDER STATUS 487
 GET INACCESSIBLE FILES 488
 GET QBIC CATALOG
 INFO 490
 GET REFERENCED FILES 491
 GET SERVER STATUS 493
 OPEN QBIC CATALOG 494
 QUIT 495
 RECONNECT SERVER AT
 NODENUM 496
 RECONNECT SERVER FOR
 DATABASE 497
 RECONNECT SERVER FOR
 DATABASE AT
 NODENUM 498
 REDISTRIBUTE
 NODEGROUP 499
 REMOVE QBIC FEATURE 501
 REORG 502
 SET QBIC AUTOCATALOG 504
 START SERVER 505
 STOP SERVER 506
 TERMINATE 507

定位器 101

注意事項 559

物件 51

安全性 61
 更新 126
 更新者的使用者 ID 248
 更新時間 249
 每一音效取樣的位元數目 194
 音效或視訊的持續時間 215
 音效或視訊的播放時間 215
 音效的取樣率 242
 音效的資料傳送速率 195
 音效產量 195
 音效通道 (數目) 228
 音效通道數目 228
 音效磁軌 (數目) 227
 音效磁軌數目 227
 格式 105
 復原 61

物件 (繼續)

- 視訊中框 (數目) 230
- 視訊中框數 230
- 視訊的框掃描率 220
- 視訊的資料傳送速率 226
- 視訊產量 220, 226
- 視訊磁軌 (數目) 231
- 視訊磁軌數目 231
- 視訊壓縮格式 198
- 傳輸 100
- 匯入時間 225
- 對齊 191
- 說明 51
- 影像中顏色 (數目) 229
- 影像中顏色數目 229
- 播放 141
- 儲存 107
- 儲存的使用者之使用者 ID 224
- 儲存時間 225
- 縱橫比 193
- 擷取中 119
- 屬性，擷取 124
- 顯示 141
- comment 196
- file name 216
- format 219
- height 223
- importer 224
- size 243
- thumbnail 244
- updater 248
- width 250
- 物件大小 243
- 物件的格式 105
 - 由 DB2 Extender 處理 105
 - 對更新使用您自己的格式 136
 - 對儲存體使用您自己的格式 115
 - 擷取視訊 198
 - 轉換視訊框 30
 - 識別更新 135
 - 識別儲存體 114
- 物件寬度 250
- 物件導向 51
- 直方圖方法 (視訊場景變更) 24
- 直方圖方法臨界值 24
- 直方圖色 58

直方圖色 (繼續)

- 特性名稱 152
- 說明 58
- 直欄 85
 - 停用 86
 - 啟用 85
- 表格 82
 - 停用 86
 - 啟用 82

〔九劃〕

- 型錄 (QBIC) 57
 - 加入特性 152
 - 自動設定 151
 - 刪除 158
 - 取消編目影像 156
 - 建立 149
 - 重新編目影像 157
 - 移除特性 153
 - 開啓 150
 - 管理 148
 - 說明 57
 - 編目影像 155
 - 擷取相關資訊 154
 - 關閉 158
- 按位顏色 58
 - 特性名稱 152
 - 說明 58
- 持續時間 UDF 215
- 查詢字串, QBIC 163
 - 重覆使用 170
- 查詢, QBIC 163
 - 建置 163
 - 發出 173
- 相對檔名 103
- 相關方法 (視訊場景變更) 24
- 相關方法臨界值 24
- 重新分配資料 19
- 音效 41
 - 更新 126
 - 更新者的使用者 ID 248
 - 更新時間 249
 - 每一取樣位元數目 194
 - 取樣率 242
 - 所有 MIDI 樂器的磁軌號碼 221

音效 (繼續)

- 持續時間 215
- 計時速度, MIDI 246, 247
- 格式 105
- 格式屬性 219
- 產量 195
- 通道 (數目) 228
- 通道數目 228
- 註解屬性 196
- 匯入時間 225
- 資料傳送速率 195
- 對齊 191
- 磁軌 (數目) 227
- 磁軌名稱, MIDI 218, 222
- 磁軌數目 227
- 播放 141
- 播放時間 215
- 儲存 107
- 儲存的使用者之使用者 ID 224
- 儲存時間 225
- 擷取中 119
- 識別 更新的格式 135
- 識別儲存體 的格式 114
- file name 216
- importer 224
- MIDI 樂器的磁軌號碼 217
- size 243
- updater 248
- 音效或視訊的對齊值 191
- 音效或視訊的播放時間 215
- 音效的取樣率 242
- 音效的資料傳送速率 195
- 音效產量 195

〔十劃〕

- 案例 6
 - 列出 6
 - 建立 6
 - 執行 7
 - 移除 7
 - 移轉 8
 - 設定 7
- 框, 視訊 26
 - 掃描率 220
 - 產量 220

框, 視訊 (繼續)

擷取中 26

特性, QBIC 查詢 163

特殊類型 52

索引檔 58

追蹤機能 541

〔十一劃〕

區段 102

參考變數, 檔案 102

執行環境 42

從表格中刪除資料 72

從屬站上的管理命令 469

ADD QBIC FEATURE 470

CATALOG QBIC COLUMN 471

CLOSE QBIC CATALOG 472

CONNECT 473

CREATE QBIC CATALOG 474

DELETE QBIC CATALOG 476

DISABLE COLUMN 477

DISABLE DATABASE 478

DISABLE TABLE 479

DISCONNECT SERVER AT

NODENUM 480

DISCONNECT SERVER FOR

DATABASE 481

DISCONNECT SERVER FOR

DATABASE AT

NODENUM 482

ENABLE COLUMN 483

ENABLE DATABASE 484

ENABLE TABLE 485

GET EXTENDER STATUS 487

GET INACCESSIBLE FILES 488

GET QBIC CATALOG

INFO 490

GET REFERENCED FILES 491

GET SERVER STATUS 493

OPEN QBIC CATALOG 494

QUIT 495

RECONNECT SERVER AT

NODENUM 496

RECONNECT SERVER FOR

DATABASE 497

從屬站上的管理命令 (繼續)

RECONNECT SERVER FOR

DATABASE AT

NODENUM 498

REDISTRIBUTE

NODEGROUP 499

REMOVE QBIC FEATURE 501

REORG 502

SET QBIC AUTOCATALOG 504

START SERVER 505

STOP SERVER 506

TERMINATE 507

從屬站緩衝區 101

更新自 132

傳輸物件來回於 101

儲存自 111

擷取至; 不進行格式轉換 121

擷取至; 進行轉換 122

從屬站檔案 102

更新自 132

傳輸物件來回於 102

儲存自 111

擷取至 122

控點 56

授權 62

啓用資料庫 79

斜率 (視訊場景變更) 24

粗細 58

〔十二劃〕

場景變更, 視訊 19

偵測 19

說明 20

媒體檔案 549

復原 61

等待指示符 143

結構 58

特性名稱 152

說明 58

視訊 41

更新 126

更新者的使用者 ID 248

更新時間 249

持續時間 215

為快照偵測開啓 27

視訊 (繼續)

音效通道 (數目) 228

音效通道數目 228

音效磁軌 (數目) 227

音效磁軌數目 227

框 (數目) 230

框掃描率 220

框數 230

格式 105

格式屬性 219

產量 (以每秒位元組為單位) 226

產量 (框掃描率) 220

視訊磁軌 (數目) 231

視訊磁軌數目 231

註解屬性 196

匯入時間 225

資料傳送速率 226

對齊 191

播放 141

播放時間 215

儲存 107

儲存的使用者之使用者 ID 224

儲存時間 225

壓縮格式 198

縱橫比 193

擷取中 119

識別 更新的格式 135

識別儲存體 的格式 114

file name 216

height 223

importer 224

size 243

thumbnail 244

updater 248

width 250

視訊的資料傳送速率 226

視訊索引 58

視訊產量 226

視訊場景變更 19

偵測 19

資料結構 22

說明 20

視訊壓縮格式 198

視訊縱橫比 193

註解 118

更新 138

註解 (繼續)

儲存 118

擷取中 126

診斷資訊 509

超載函數名稱 54

階層式檔案系統 (HFS) 52

〔十三劃〕

傳輸大型物件 100

資料庫 79

啓用 79

清除 meta 資料 8

連接 3

檢查是否啓用 87

資料結構 55

快照型錄 58

快照偵測 22

控點 56

視訊索引 58

管理支援表格 55

QBIC 型錄 57

〔十四劃〕

圖點 57

對比 58

磁軌 227

音效號碼 227

視訊數目 231

磁軌名稱, MIDI 222

磁軌號碼, MIDI 218

管理支援表 55

安全性 61

清除 8

說明 55

管理作業概觀 75

綱目名稱 53

〔十五劃〕

影像 41

分數 (QBIC) 174

平均色 57

更新 126

更新者的使用者 ID 248

影像 (繼續)

更新時間 249

直方圖色 58

按內容查詢 147

按位顏色 58

格式 105

格式屬性 219

高度 轉換 106

旋轉 106

結構 58

註解屬性 196

匯入時間 225

圖點 57

寬度轉換 106

儲存 107

儲存的使用者之使用者 ID 224

儲存時間 225

壓縮類型 106

擷取中 119

轉換選項 106

顏色 (數目) 229

顏色數目 229

識別 更新的格式 135

識別儲存體 的格式 114

顯示 141

file name 216

height 223

importer 224

size 243

updater 248

width 250

影像旋轉 106

播放音效 141

播放視訊 141

標度因數 106

標記, 函數 54

標頭檔 99

碼, 回覆 509

範例媒體檔案 549

範例程式 549

緩衝區, 從屬站 101

更新自 132

傳輸物件來回於 101

儲存自 111

擷取至; 不進行格式轉換 121

擷取至; 進行轉換 122

調整 61

說明 61

〔十七劃〕

儲存快照 33

儲存物件 107

壓縮類型 106

應用程式設計介面 (API) 251

DBaAdminGetInaccessibleFiles 252

DBaAdminGetReferencedFiles 254

DBaAdminIsFileReferenced 256

DBaAdminReorgMetadata 258

DBaDisableColumn 260

DBaDisableDatabase 262

DBaDisableTable 264

DBaEnableColumn 266

DBaEnableDatabase 268

DBaEnableTable 270

DBaGetError 272

DBaGetInaccessibleFiles 274

DBaGetReferencedFiles 276

DBaIsColumnEnabled 278

DBaIsDatabaseEnabled 280

DBaIsFileReferenced 282

DBaIsTableEnabled 284

DBaPlay 286

DBaPrepareAttrs 289

DBaReorgMetadata 290

DBiAdminGetInaccessibleFiles 292

DBiAdminGetReferencedFiles 294

DBiAdminIsFileReferenced 296

DBiAdminReorgMetadata 298

DBiBrowse 300

DBiDisableColumn 303

DBiDisableDatabase 305

DBiDisableTable 307

DBiEnableColumn 309

DBiEnableDatabase 311

DBiEnableTable 313

DBiGetError 315

DBiGetInaccessibleFiles 317

DBiGetReferencedFiles 319

DBiIsColumnEnabled 321

DBiIsDatabaseEnabled 323

DBiIsFileReferenced 325

應用程式設計介面 (API) (繼續)

DBIsTableEnabled 327
DBiPrepareAttrs 329
DBiReorgMetadata 330
DBvAdminGetInaccessibleFiles 332
DBvAdminGetReferencedFiles 334
DBvAdminIsFileReferenced 336
DBvAdminReorgMetadata 338
DBvBuildStoryboardFile 340
DBvBuildStoryboardTable 342
DBvClose 344
DBvCreateIndex 346
DBvCreateIndexFromVideo 348
DBvCreateShotCatalog 350
DBvDeleteShot 352
DBvDeleteShotCatalog 354
DBvDetectShot 356
DBvDisableColumn 358
DBvDisableDatabase 360
DBvDisableTable 362
DBvEnableColumn 364
DBvEnableDatabase 366
DBvEnableTable 368
DBvFrameDataTo24BitRGB 370
DBvGetError 372
DBvGetFrame 374
DBvGetInaccessibleFiles 376
DBvGetReferencedFiles 378
DBvInitShotControl 380
DBvInitStoryboardCtrl 382
DBvInsertShot 384
DBvIsColumnEnabled 386
DBvIsDatabaseEnabled 388
DBvIsFileReferenced 390
DBvIsIndex 392
DBvIsTableEnabled 394
DBvMergeShots 396
DBvOpenFile 398
DBvOpenHandle 400
DBvPlay 402
DBvPrepareAttrs 405
DBvReorgMetadata 406
DBvSetFrameNumber 408
DBvSetShotComment 410
DBvUpdateShot 412
QbAddFeature 416

應用程式設計介面 (API) (繼續)

QbCatalogColumn 418
QbCatalogImage 420
QbCloseCatalog 422
QbCreateCatalog 423
QbDeleteCatalog 425
QbGetCatalogInfo 427
QbListFeatures 429
QbOpenCatalog 431
QbQueryAddFeature 433
QbQueryCreate 435
QbQueryDelete 437
QbQueryGetFeatureCount 438
QbQueryGetString 440
QbQueryListFeatures 442
QbQueryNameCreate 444
QbQueryNameDelete 446
QbQueryNameSearch 448
QbQueryRemoveFeature 450
QbQuerySearch 452
QbQuerySetFeatureData 454
QbQuerySetFeatureWeight 456
QbQueryStringSearch 458
QbReCatalogColumn 460
QbRemoveFeature 462
QbSetAutoCatalog 464
QbUncatalogImage 466

檔案 100

名稱 (含物件) 216
名稱, 指定 103
名稱, 相對 103
更新自從屬站 132
物件傳輸是在表格和 100
尋找表格所參照的檔案 89
傳輸物件來回於從屬站 102
儲存自從屬站 111

檔案參考變數 102

環境變數 141

DB2AUDIOEXPORT 543
DB2AUDIOPATH 543
DB2AUDIOPLAYER 141
DB2AUDIOSTORE 543
DB2AUDIOTEMP 543
DB2CATALOGDELAY 149
DB2IMAGEBROWSER 141
DB2IMAGEEXPORT 543

環境變數 (繼續)

DB2IMAGEPATH 543
DB2IMAGESTORE 543
DB2IMAGETEMP 543
DB2MMDATAPATH 11, 544
DB2VIDEOEXPORT 543
DB2VIDEOPATH 543
DB2VIDEOPLAYER 141
DB2VIDEOSTORE 543
DB2VIDEOTEMP 543

縮圖 117

更新 137
儲存 117
顯示 144

〔十八劃〕

擷取物件 119
轉換選項, 影像 106
顏色, 數目 (在影像中) 229

〔二十劃〕

觸發函式 54

〔二十一劃〕

屬性, 物件 124
更新者的使用者 ID 248
更新時間 249
每一音效取樣的位元數目 194
每四分之一音符的計時速度 246
每秒計時速度 247
所有 MIDI 樂器的磁軌號碼 221
音效或視訊的持續時間 215
音效或視訊的播放時間 215
音效的取樣率 242
音效的資料傳送速率 195
音效產量 195
音效通道 (數目) 228
音效通道數目 228
音效磁軌數目 227
視訊中框 (數目) 230
視訊中框數 230
視訊的框掃描率 220
視訊的資料傳送速率 226

屬性，物件 (繼續)

視訊產量 220, 226
視訊磁軌 (數目) 231
視訊磁軌數目 231
視訊壓縮格式 198
匯入時間 225
對齊值 191
磁軌名稱, MIDI 218, 222
說明 124
影像中顏色 (數目) 229
影像中顏色數目 229
儲存的使用者之使用者 ID 224
儲存時間 225
縱橫比 193
comment 196
file name 216
format 219
height 223
importer 224
MIDI 樂器的磁軌號碼 217
size 243
updater 248
width 250

〔二十三劃〕

顯示視訊框 141
顯示影像 141
顯示縮圖 144

〔二十五劃〕

觀念 51

A

ADD QBIC FEATURE 命令 152, 470
AlignValue UDF 191
AspectRatio UDF 193
Audio Extender 42
 概觀 42
 DBaAdminGetInaccessibleFiles
 API 252
 DBaAdminGetReferencedFiles
 API 254

Audio Extender (繼續)

DBaAdminIsFileReferenced
 API 256
DBaAdminReorgMetadata
 API 258
DBaDisableColumn API 260
DBaDisableDatabase API 262
DBaDisableTable API 264
DBaEnableColumn API 266
DBaEnableDatabase API 268
DBaEnableTable API 270
DBaGetError API 272
DBaGetInaccessibleFiles API 274
DBaGetReferencedFiles API 276
DBalsColumnEnabled API 278
DBalsDatabaseEnabled API 280
DBalsFileReferenced API 282
DBalsTableEnabled API 284
DBaPlay API 286
DBaReorgMetadata API 290
UDF 187
UDT 187

B

BitsPerSample UDF 194
BytesPerSec UDF 195

C

CATALOG QBIC COLUMN 命令
 156, 471
 重新編目影像 157
 編目直欄 156
Cb 圖點平面 30
channels, 音效號碼 228
CLOB (字元大型物件) 52
CLOSE QBIC CATALOG 命令 158, 472
Comment UDF 196
CompressType UDF 198
CONNECT 命令 473
Content UDF 199
Cr 圖點平面 30
CREATE QBIC CATALOG 命令
 149, 474

CURRENT SERVER 特別 暫存區
108

D

DB2 Extender 41
 回覆碼 509
 安全性 61
 作業環境 49
 更新物件使用 105
 系列 42
 追蹤機能 541
 執行環境 42
 您可以執行的作業 96
 復原 61
 程式設計概觀 95
 概觀 41
 資料結構 55
 實務範例 63
 碼 509, 510
 範例媒體檔案 549
 範例程式 549
 儲存物件使用 105
 擷取物件使用 105
 觀念 51
 Software Developers Kit
 (SDK) 42
 SQLSTATE 訊息碼 510
 UDF 187
 UDT 187
DB2 Extender 的主/從平台 49
DB2 Extender 的平台 49
DB2 Extender 的作業環境 49
DB2 Extender 的概觀 41
DB2 命令行處理器 43
DB2Audio UDF 204
DB2AUDIO 資料類型 187
DB2AUDIOEXPORT 環境變數 543
DB2AUDIOPATH 環境變數 543
DB2AUDIOPLAYER 環境變數 141
DB2AUDIOSTORE 環境變數 543
DB2AUDIOTEMP 環境變數 543
DB2CATALOGDELAY 環境變數
 149
db2ext 命令行處理器 43
DB2Image UDF 207

DB2IMAGE 資料類型 187
 DB2IMAGEBROWSER 環境變數 141
 DB2IMAGEEXPORT 環境變數 543
 DB2IMAGEPATH 環境變數 543
 DB2IMAGESTORE 環境變數 543
 DB2IMAGETEMP 環境變數 543
 DB2MMDATAPATH 11, 544
 DB2Video UDF 211
 DB2VIDEO 資料類型 187
 DB2VIDEOEXPORT 環境變數 543
 DB2VIDEOPATH 環境變數 543
 DB2VIDEOPLAYER 環境變數 141
 DB2VIDEOSTORE 環境變數 543
 DB2VIDEOTEMP 環境變數 543
 DBaAdminGetInaccessibleFiles API 252
 DBaAdminGetReferencedFiles API 254
 DBaAdminIsFileReferenced API 256
 DBaAdminReorgMetadata API 258
 DBaDisableColumn API 260
 DBaDisableDatabase API 262
 DBaDisableTable API 264
 DBaEnableColumn API 266
 DBaEnableDatabase API 268
 DBaEnableTable API 270
 DBaGetError API 272
 DBaGetInaccessibleFiles API 274
 DBaGetReferencedFiles API 276
 DBaIsColumnEnabled API 278
 DBaIsDatabaseEnabled API 280
 DBaIsFileReferenced API 282
 DBaIsTableEnabled API 284
 DBaPlay API 286
 DBaPrepareAttrs API 289
 DBaReorgMetadata API 290
 DBCLOB (雙位元組大型物件) 52
 DBiAdminGetInaccessibleFiles API 292
 DBiAdminGetReferencedFiles API 294
 DBiAdminIsFileReferenced API 296
 DBiAdminReorgMetadata API 298
 DBiBrowse API 300
 DBiDisableColumn API 303
 DBiDisableDatabase API 305
 DBiDisableTable API 307
 DBiEnableColumn API 309
 DBiEnableDatabase API 311
 DBiEnableTable API 313
 DBiGetError API 315
 DBiGetInaccessibleFiles API 317
 DBiGetReferencedFiles API 319
 DBiIsColumnEnabled API 321
 DBiIsDatabaseEnabled API 323
 DBiIsFileReferenced API 325
 DBiIsTableEnabled API 327
 DBiPrepareAttrs API 329
 DBiReorgMetadata API 330
 DBvAdminGetInaccessibleFiles API 332
 DBvAdminGetReferencedFiles API 334
 DBvAdminIsFileReferenced API 336
 DBvAdminReorgMetadata API 338
 DBvBuildStoryboardFile API 340
 DBvBuildStoryboardTable API 342
 DBvClose API 344
 DBvCreateIndex API 346
 DBvCreateIndexFromVideo API 348
 DBvCreateShotCatalog API 350
 DBvDeleteShot API 352
 DBvDeleteShotCatalog API 354
 DBvDetectShot API 356
 DBvDisableColumn API 358
 DBvDisableDatabase API 360
 DBvDisableTable API 362
 DBvEnableColumn API 364
 DBvEnableDatabase API 366
 DBvEnableTable API 368
 DBvFrameData 資料結構 25
 DBvFrameDataTo24BitRGB API 370
 DBvGetError API 372
 DBvGetFrame API 374
 DBvGetInaccessibleFiles API 376
 DBvGetReferencedFiles API 378
 DBvInitShotControl API 380, 382
 DBvInsertShot API 384
 DBvIOType 資料結構 22
 DBvIsColumnEnabled API 386
 DBvIsDatabaseEnabled API 388
 DBvIsFileReferenced API 390
 DBvIsIndex API 392
 DBvIsTableEnabled API 394
 DBvMergeShots API 396
 DBvOpenFile API 398
 DBvOpenHandle API 400
 DBvPlay API 402
 DBvPrepareAttrs API 405
 DBvReorgMetadata API 406
 DBvSetFrameNumber API 408
 DBvSetShotComment API 410
 DBvShotControl 資料結構 23
 DBvShotType 資料結構 24
 DBvStoryboardCtrl 資料結構 25
 DBvUpdateShot API 412
 DELETE QBIC CATALOG 命令 158, 476
 DISABLE COLUMN 命令 477
 DISABLE DATABASE 命令 478
 DISABLE TABLE 命令 479
 DISCONNECT SERVER AT NODENUM 命令 480
 DISCONNECT SERVER FOR DATABASE AT NODENUM 命令 482
 DISCONNECT SERVER FOR DATABASE 命令 481
 dmbaudio.h 併入檔 99
 DMBICRT 命令 10
 DMBIDROP 命令 12
 DMBILIST 命令 13
 dmbimage.h 併入檔 99
 DMBIMIGR 命令 14
 dmbqbapi.h 併入檔 99
 dmbshot.h 併入檔 100
 DMBSTART 命令 15
 DMBSTAT 命令 17
 DMBSTOP 命令 18
 DMBTRC 命令 541
 dmbvideo.h 併入檔 100

E
 ENABLE COLUMN 命令 483
 ENABLE DATABASE 484
 ENABLE TABLE 命令 485

F

Filename UDF 216
FindInstrument UDF 217
FindTrackName UDF 218
Format UDF 219
FrameRate UDF 220

G

GET EXTENDER STATUS 命令
487
GET INACCESSIBLE FILES 命令
488
GET QBIC CATALOG INFO 命令
154, 490
GET REFERENCED FILES 命令
491
GET SERVER STATUS 命令 493
GetInstruments UDF 221
GetTrackNames UDF 222

H

Height UDF 223

I

Image Extender 42
概觀 42
DBaPrepareAttrs API 289
DBiAdminGetInaccessibleFiles
API 292
DBiAdminGetReferencedFiles
API 294
DBiAdminIsFileReferenced
API 296
DBiAdminReorgMetadata API 298
DBiBrowse API 300
DBiDisableColumn API 303
DBiDisableDatabase API 305
DBiDisableTable API 307
DBiEnableColumn API 309
DBiEnableDatabase API 311
DBiEnableTable API 313
DBiGetError API 315

Image Extender (繼續)

DBiGetInaccessibleFiles API 317
DBiGetReferencedFiles API 319
DBiIsColumnEnabled API 321
DBiIsDatabaseEnabled API 323
DBiIsFileReferenced API 325
DBiIsTableEnabled API 327
DBiPrepareAttrs API 329
DBiReorgMetadata API 330
DBvPrepareAttrs API 405
UDF 187
UDT 187

Importer UDF 224
ImportTime UDF 225

L

LOB (大型物件) 52
定位器 101
傳輸 100
說明 52
播放 141
顯示 141

M

MaxBytesPerSec UDF 226
meta 資料表格 55
安全性 61
說明 55
MIDI 樂器 221
MIDI 樂器的磁軌號碼 217
MMDB_STORAGE_TYPE_EXTERNAL 113
更新時 134
儲存時 113
MMDB_STORAGE_TYPE_INTERNAL 113
更新時 134
儲存時 113

MPEG-1 視訊格式 30

N

Net.Data 範例 552
NumAudioTracks UDF 227
NumChannels UDF 228
NumColors UDF 229

NumFrames UDF 230
NumVideoTracks UDF 231

O

OPEN QBIC CATALOG 命令 150,
494

Q

QbAddFeature API 152, 416
QbCatalogColumn API 156, 418
QbCatalogImage API 155, 420
QbCloseCatalog API 158, 422
QbColor 168
QbColorFeatureClass 152
QbColorHistogramFeatureClass 152
QbCreateCatalog API 149, 423
QbDeleteCatalog API 158, 425
QbDrawFeatureClass 152
QbGetCatalogInfo API 154, 427
QbHistogramColor 169
QBIC 型錄 57
QBIC 查詢 163
加入特性 166
字串 163
刪除 172
物件 166
建立 166
移除特性 172
發出 173
資料來源 167
儲存 170
擷取相關資訊 171

QbImageBuffer 168
QbImageSource 167
QbListFeatures 154
QbListFeatures API 429
QbOpenCatalog API 150, 431
QbQueryAddFeature API 166, 433
QbQueryCreate API 166, 435
QbQueryDelete API 172, 437
QbQueryGetFeatureCount API 171,
438
QbQueryGetString API 170, 440

QbQueryListFeatures API 171, 442
QbQueryNameCreate API 444
QbQueryNameDelete API 172, 446
QbQueryNameSearch API 173, 448
QbQueryRemoveFeature API 172, 450
QbQuerySearch API 173, 452
QbQuerySetFeatureData API 167, 454
QbQuerySetFeatureWeight API 456
QbQueryStringSearch API 173, 458
QbReCatalogColumn API 157, 460
QbRemoveFeature API 153, 462
QbScoreFromName UDF 174, 232
QbScoreFromStr UDF 174, 234
QbScoreTBFromName UDF 174, 235
QbScoreTBFromStr UDF 174, 237
QbSetAutoCatalog API 151, 464
QbTextureFeatureClass 152
QbUncatalogImage API 156, 466
QUIT 命令 495

R

RECONNECT SERVER AT
NODENUM 命令 496
RECONNECT SERVER FOR
DATABASE AT NODENUM 命令 498
RECONNECT SERVER FOR
DATABASE 命令 497
REDISTRIBUTE NODEGROUP 命令 499
REMOVE QBIC FEATURE 命令 153, 501
REORG 命令 502
Replace UDF 239
RGB 視訊格式 30

S

SamplingRate UDF 242
SET CURRENT FUNCTION PATH
陳述式 54
SET QBIC AUTOCATALOG 命令 151, 504
Size UDF 243

Software Developers Kit (SDK) 42
SQLSTATE 訊息碼 510
START SERVER 命令 505
STOP SERVER 命令 506
storyboard 36

T

TERMINATE 命令 507
Text Extender 42
Thumbnail UDF 244
TicksPerQNote UDF 246
TicksPerSec UDF 247

U

UDF (使用者定義函數) 53
名稱 53
函數路徑 54
參照 187
超載 54
說明 53
標記 54
AlignValue 191
AspectRatio 193
BitsPerSample 194
BytesPerSec 195
Comment 196
CompressType 198
Content 199
DB2Audio 204
DB2Image 207
DB2Video 211
Duration 215
Filename 216
FindInstrument 217
FindTrackName 218
Format 219
FrameRate 220
GetInstruments 221
GetTrackNames 222
Height 223
Importer 224
ImportTime 225
MaxBytesPerSec 226
NumAudioTracks 227

UDF (使用者定義函數) (繼續)
NumChannels 228
NumColors 229
NumFrames 230
NumVideoTracks 231
QbScoreFromName 232
QbScoreFromStr 234
QbScoreTBFromName 235
QbScoreTBFromStr 237
Replace 239
SamplingRate 242
Size 243
Thumbnail 244
TicksPerQNote 246
TicksPerSec 247
Updater 248
UpdateTime 249
Width 250
UDF_MEM_SZ 參數 112
更新時 133
儲存時 112
擷取時 121
UDT (使用者定義類型) 52
名稱 53
說明 52
Unicode 支援 104
UPDATE DATABASE MANAGER
CONFIGURATION 命令 112
更新時 133
儲存時 112
擷取時 122
Updater UDF 248
UpdateTime UDF 249

V

Video Extender 42
概觀 42
DBvAdminGetInaccessibleFiles
API 332
DBvAdminGetReferencedFiles
API 334
DBvAdminIsFileReferenced
API 336
DBvAdminReorgMetadata
API 338

Video Extender (繼續)

- DBvBuildStoryboardFile API 340
- DBvBuildStoryboardTable
 - API 342
- DBvClose API 344
- DBvCreateIndex API 346
- DBvCreateIndexFromVideo
 - API 348
- DBvCreateShotCatalog API 350
- DBvDeleteShot API 352
- DBvDeleteShotCatalog API 354
- DBvDetectShot API 356
- DBvDisableColumn API 358
- DBvDisableDatabase API 360
- DBvDisableTable API 362
- DBvEnableColumn API 364
- DBvEnableDatabase API 366
- DBvEnableTable API 368
- DBvFrameDataTo24BitRGB
 - API 370
- DBvGetError API 372
- DBvGetFrame API 374
- DBvGetInaccessibleFiles API 376
- DBvGetReferencedFiles API 378
- DBvInitShotControl API 380
- DBvInitStoryboardCtrl API 382
- DBvInsertShot API 384
- DBvIsColumnEnabled API 386
- DBvIsDatabaseEnabled API 388
- DBvIsFileReferenced API 390
- DBvIsIndex API 392
- DBvIsTableEnabled API 394
- DBvMergeShots API 396
- DBvOpenFile API 398
- DBvOpenHandle API 400
- DBvPlay API 402
- DBvReorgMetadata API 406
- DBvSetFrameNumber API 408
- DBvSetShotComment API 410
- DBvUpdateShot API 412
- UDF 187
- UDT 187

W

- Width UDF 250

連絡 IBM

當您有技術上的問題時，請在洽詢「DB2 客戶支援中心」之前，仔細閱讀並執行疑難排解指南所建議的動作。該指南會告訴您必須預先準備的資訊，協助「DB2 客戶支援中心」提供更完善的服務。

若要取得 DB2 Universal Database 產品的相關資訊，或是訂購該系列產品，請洽詢當地 IBM 分公司的業務代表，或是 IBM 授權的軟體經銷商。

美國地區居民請撥下列電話：

- 1-800-237-5511，客戶支援中心
- 1-888-426-4343，取得可用服務選項的資訊

產品資訊

美國地區居民請撥下列電話：

- 1-800-IBM-CALL (1-800-426-2255) 或 1-800-3IBM-OS2 (1-800-342-6672)，訂購產品或取得一般資訊。
- 1-800-879-2755，訂購出版品。

<http://www.ibm.com/software/data/db2>

DB2 全球資訊網 (WWW) 網頁提供最新的 DB2 資訊，包括新聞、產品說明、教育課程等。

<http://www.ibm.com/software/data/support>

DB2 支援網頁提供常見問題、修正程式、書籍及最新的 DB2 技術資訊。

註：本資訊僅提供英文版。

<http://www.ibm.com/software/data/db2/extenders>

DB2 Extender 網頁提供目前可供使用的所有 DB2 Extender 之相關資訊。包括 DB2 XML Extender、DB2 Spatial Extender 及 DB2 AVI Extender。

<http://www.ibm.com/software/data/db2/extenders/support>

DB2 Extender 支援網頁提供常見問題、提示及要訣及文件。

註：本資訊僅提供英文版。

<http://www.elink.ibm.com/public/applications/publications/cgibin/pbi.cgi>

Publications Center 提供如何訂購或下載出版品的資訊。

<http://www.ibm.com/certify/index.html>

IBM 網站中的 Professional Certification Program 會提供包括 DB2 在內之各種 IBM 產品的認證測試資訊。

在 Compuserve 上：GO IBMDB2

輸入此項命令，即可存 IBM DB2 Family 論壇。這些論壇支援所有 DB2 產品。

關於美國以外地區如何連絡 IBM 的資訊，請參考 *IBM Software Support Handbook* 的附錄 A。

若要存取本文件，請造訪下列網頁：
<http://techsupport.services.ibm.com/guides/contacts.html>

註：在某些國家，IBM 授權經銷商應連絡其經銷商支援組織，而不是 IBM 支援中心。



SH40-0372-00

