

IBM® DB2® Universal Database



版次注意事項

版本 7.2 / 版本 7.1 *FixPak 3*

IBM® DB2® Universal Database



版次注意事項

版本 7.2 / 版本 7.1 *FixPak 3*

目錄

歡迎使用 DB2 Universal Database 版本 7! xiii

第1篇 特別需知 1

特別需知. 3

DB2 UDB 版本 7 的存取特性	3
鍵盤之輸入和導引.	3
方便顯示特性	3
選擇式警告提示	3
支援技術的相容性.	3
可存取的文件	4
附加的必要 Solaris 修補層次	4
Supported CPUs on DB2 Version 7 for Solaris	4
新增節點至分割資料庫的問題	4
移轉期間發生的錯誤	6
Red Flag Linux 的中文語言環境修正程式	6
如果可移除式磁碟機未連接, 則 DB2 安裝會中斷	6
日文和簡體中文 Linux 環境之 DB2 for Linux 的附加語言環境設定	6
Microsoft Internet Explorer 上的控制中心問題.	6
在 Windows 環境中「資訊型錄管理程式」及 Sybase 之間的不相容	7
控制中心功能流失.	7
Netscape CD 不隨附於 DB2 UDB	7
XML Readme 檔的錯誤.	7
Linux for S/390 上可能的資料流失	8
Windows 2000 上的 DB2 UDB	8

線上文件 (HTML、PDF 與「搜尋」) 9

Windows 2000 作業系統所支援的 Web 瀏覽器	9
在 Solaris 系統下搜尋 DB2 連線資訊	9
切換 NetQuestion for OS/2 以使用 TCP/IP	9
啟動 Netscape 時產生的錯誤訊息.	9
UNIX 系統上 Adobe Acrobat Reader 的架構需求	10
SQL Reference 僅提供一個 PDF 檔	11

第2篇 安裝與架構 13

一般安裝資訊 15

下載所有支援的 DB2 從屬站的安裝套件	15
在 Windows 2000 上安裝 DB2	15
關於概略表以特別暫存區定義的移轉問題.	15
Windows 2000 的 IPX/SPX 通信協定支援	15
升級 DB2 前一版之前先停止 DB2 程序	15
若已安裝其它 DB2 產品, 請在安裝 DB2 之後執行 db2iupdt.	16
設定 Linux 環境以執行 DB2 控制中心	16
DB2 Universal Database Enterprise Edition 及 DB2 Connect Enterprise Edition for Linux on S/390	16

DB2 Universal Database Enterprise - Extended Edition for UNIX 快速入門.	17
HP-UX 的 shmseg 核心參數	17
移轉 IBM Visual 倉儲控制資料庫	17
存取倉儲控制資料庫	17

DB2 Data Links Manager 快速入門 . . . 19

Dlfm 的啟動失敗訊息: 取得 afsfid 字首時發生錯誤	19
設定備份保存檔的 Tivoli Storage Manager 類別	19
DFS 從屬站啟動程式的磁碟空間需求	19
監督在 AIX 的資料鏈結檔案管理程式後端程序.	19
安裝和架構 DB2 Data Links Manager for AIX: DCE-DFS 環境裡的額外安裝注意事項.	19
失敗的 "dlfm add_prefix" 指令	20
安裝和架構 DB2 Data Links Manager for AIX: 使用 db2setup 公用程式在 AIX 上安裝 DB2 Data Links Manager.	20
安裝和架構 DB2 Data Links Manager for AIX: DCE-DFS 後置安裝作業	20
安裝和架構 DB2 Data Links Manager for AIX: 使用 Smit 來手動安裝 DB2 Data Links Manager	21
安裝和架構 DB2 Data Links 的 DFS 從屬站啟動程式	21
安裝和架構 DB2 Data Links Manager for Solaris	21
針對 AIX 上的 DB2 Data Links Manager 選擇備份方法.	22
針對 Solaris 作業環境上的 DB2 Data Links Manager 選擇備份方法.	23
針對 Windows NT 上的 DB2 Data Links Manager 選擇備份方法	24
備份 AIX 上的記錄檔案系統.	25
在 Windows NT 上資料鏈結中的 Administrator 群組專用權	26
將 Data Links File System Filter (DLFF) 安裝的記載減到最少	26
在安裝後記載訊息	27
使用 SMIT 解除安裝 DLFM 元件可能會除去額外的檔案集	27
開始之前/決定主電腦名稱	28
使用 DLFM: 從 DB2 資料庫捨棄 DB2 Data Links Manager 後的清除動作.	29
DLFM1001E (新的錯誤訊息).	29
DLFM 設定架構檔選項	30
在 AIX 上執行 Data Links/DFS Script dmapp_prestart 時發生錯誤	30
Tivoli Space Manager 與 Data Links 整合	30
功能限制	31
第 4 章 安裝和架構 DB2 Data Links Manager for AIX	31
一般安裝注意事項	31

安裝與架構補充資料 33

第 5 章 在 UNIX 作業系統上安裝 DB2 從屬站	33
HP-UX 核心程式架構參數	33
第 12 章 執行您自己的應用程式	33
使用 Run-Time Client 連結資料庫公用程式	33
UNIX 從屬站使用 ODBC 存取 DB2	33
第 24 章 設置聯合系統來存取多重資料來源	33
聯合系統	33
安裝 DB2 Relational Connect	34
第 26 章 存取 Oracle 資料來源	35
文件錯誤	36
存取 Sybase 資料來源 (新增章節)	36
新增 Sybase 資料來源至聯合伺服器	37
指定 Sybase 字碼頁	42
使用 ODBC 存取 Microsoft SQL Server 資料來源 (新增章節)	42
新增 Microsoft SQL Server 資料來源至聯合伺服器	43
複查 Microsoft SQL Server 字碼頁	47

第3篇 管理 49

Administration Guide: Planning (無中文版). 53

Chapter 8. Physical Database Design	53
Partitioning Keys	53
Designing Nodegroups	53
Chapter 9. Designing Distributed Databases	53
Updating Multiple Databases	53
Chapter 13. High Availability in the Windows NT Environment	54
Need to Reboot the Machine Before Running DB2MSCS Utility	54
Chapter 14. DB2 and High Availability on Sun Cluster 2.2	54
Veritas Support on Solaris	54
Appendix B. Naming Rules	55
Notes on Greater Than 8-Character User IDs and Schema Names	55
User IDs and Passwords	56
Appendix D. Incompatibilities Between Releases	56
Windows NT DLFS Incompatible with Norton's Utilities	56
SET CONSTRAINTS Replaced by SET INTEGRITY	57
Appendix E. National Language Support	57
National Language Versions of DB2 Version 7	57
Locale Setting for the DB2 Administration Server	60
DB2 UDB Supports the Baltic Rim Code Page (MS-1257) on Windows Platforms	60
Deriving Code Page Values	60
Country Code and Code Page Support	60
Character Sets	61

Administration Guide: Implementation(無中文版). 63

Adding or Extending DMS Containers (New Process)	63
Chapter 1. Administering DB2 using GUI Tools	63
Chapter 3. Creating a Database	64
Creating a Table Space	64
Creating a Sequence	65
Comparing IDENTITY Columns and Sequences	67
Creating an Index, Index Extension, or an Index Specification	67
Chapter 4. Altering a Database	67
Adding a Container to an SMS Table Space on a Partition	67
Altering an Identity Column	68
Altering a Sequence	68
Dropping a Sequence	69
Switching the State of a Table Space	69
Modifying Containers in a DMS Table Space	69
Chapter 5. Controlling Database Access	70
Sequence Privileges	70
Data Encryption	70
Chapter 8. Recovering a Database	71
How to Use Suspended I/O	71
Incremental Backup and Recovery	73
Parallel Recovery	75
Backing Up to Named Pipes	75
Backup from Split Image	76
On Demand Log Archive	78
Log Mirroring	78
Cross Platform Backup and Restore Support on Sun Solaris and HP	79
DB2 Data Links Manager Considerations/Backup Utility Considerations	79
DB2 Data Links Manager Considerations/Restore and Rollforward Utility Considerations	80
Restoring Databases from an Offline Backup without Rolling Forward	81
Restoring Databases and Table Spaces, and Rolling Forward to the End of the Logs	82
DB2 Data Links Manager and Recovery Interactions	82
Detection of Situations that Require Reconciliation	86
Appendix C. User Exit for Database Recovery	86
Appendix D. Issuing Commands to Multiple Database Partition Servers	87
Appendix I. High Speed Inter-node Communications	87
Enabling DB2 to Run Using VI	87

Administration Guide: Performance (無中文版) 89

Chapter 3. Application Considerations	89
Specifying the Isolation Level	89
Adjusting the Optimization Class	89
Dynamic Compound Statements	90
Chapter 4. Environmental Considerations	90
Using Larger Index Keys	90
Chapter 5. System Catalog Statistics	91
Collecting and Using Distribution Statistics	91

Rules for Updating Catalog Statistics	91	db2look - DB2 Statistics Extraction Tool.	127
Sub-element Statistics	92	db2updv7 - Update Database to Version 7 Current	
Chapter 6. Understanding the SQL Compiler	94	Fix Level	128
Replicated Summary Tables	94	New Command Line Processor Option (-x, Suppress	
Data Access Concepts and Optimization	96	printing of column headings).	129
Chapter 8. Operational Performance	96	True Type Font Requirement for DB2 CLP.	129
Managing the Database Buffer Pool.	96	ADD DATALINKS MANAGER	129
Managing Multiple Database Buffer Pools	97	ARCHIVE LOG (new command)	130
Chapter 9. Using the Governor	98	Archive Log	131
Chapter 13. Configuring DB2.	98	BACKUP DATABASE	132
Sort Heap Size (sorheap)	98	Syntax Diagram	132
Sort Heap Threshold (sheapthres).	98	DB2 Data Links Manager Considerations.	133
Maximum Percent of Lock List Before Escalation		BIND	133
(maxlocks)	99	CALL	134
Configuring DB2/DB2 Data Links Manager/Data		DROP DATALINKS MANAGER (new command)	135
Links Access Token Expiry Interval (dl_expint) .	100	DROP DATALINKS MANAGER	136
MIN_DEC_DIV_3 Database Configuration		EXPORT	139
Parameter	100	GET DATABASE CONFIGURATION	140
Application Control Heap Size (app_ctl_heap_sz)	101	GET ROUTINE (new command)	140
Database System Monitor Heap Size		GET ROUTINE	141
(mon_heap_sz)	101	GET SNAPSHOT	141
Maximum Number of Active Applications		IMPORT	141
(maxappls)	101	LIST HISTORY.	142
Recovery Range and Soft Checkpoint Interval		LOAD	143
(softmax)	101	PING (new command).	144
Track Modified Pages Enable (trackmod).	101	PING	145
Change the Database Log Path (newlogpath) . . .	102	PUT ROUTINE (new command)	146
Location of Log Files (logpath).	103	PUT ROUTINE	147
Maximum Storage for Lock List (locklist) . . .	103	RECONCILE.	147
Appendix A. DB2 Registry and Environment		REORGANIZE TABLE	148
Variables	103	RESTORE DATABASE	148
Table of New and Changed Registry Variables	104	Syntax	148
Appendix C. SQL Explain Tools	108	DB2 Data Links Manager Considerations.	149
Administering Satellites Guide and		ROLLFORWARD DATABASE	149
Reference(無中文版)	109	Documentation Error in CLP Return Codes	149
Setting up Version 7.2 DB2 Personal Edition and		Data Movement Utilities Guide and	
DB2 Workgroup Edition as Satellites	109	Reference(無中文版)	151
Prerequisites	109	Chapter 2. Import	151
Configuring the Version 7.2 System for		Using Import with Buffered Inserts	151
Synchronization	111	Chapter 3. Load.	151
Installing FixPak 2 or Higher on a Version 6		Pending States After a Load Operation	151
Enterprise Edition System.	112	Load Restrictions and Limitations	151
Upgrading a Version 6 DB2 Control Center and		totalreespace File Type Modifier	151
Satellite Administration Center	120	Chapter 4. AutoLoader.	151
Command Reference(無中文版).	121	rexecd Required to Run Autoloader When	
db2batch - Benchmark Tool	121	Authentication Set to YES	151
db2cap (new command)	121	Replication 指南與參考手冊	153
db2cap - CLI/ODBC Static Package Binding Tool	122	抄寫及非 IBM 伺服器	153
db2ckrst (new command)	123	在 Windows 2000 上的抄寫.	153
db2ckrst - Check Incremental Restore Image		儲存 SQL 檔案時的已知錯誤	153
Sequence	124	DB2 維護.	153
db2gncol (new command).	125	Web 上的 Data Difference Utility	153
db2gncol - Update Generated Column Values . .	126	第 3 章 資料抄寫實務範例	153
db2inidb - Initialize a Mirrored Database.	127	抄寫實務	153

第 5 章 抄寫規劃	154
表格及直欄名稱	154
DATALINK 抄寫	154
LOB 限制	155
規劃抄寫	155
第 6 章 設定抄寫環境	155
隨處更新 (Update-anywhere) 先決條件	155
設定抄寫環境	155
第 8 章 問題判斷	156
第 9 章 AS/400 的擷取程式和引用程式	156
第 10 章 OS/390 的擷取程式和引用程式	157
DB2 DataPropagator for OS/390 的先決條件	157
OS/390 上的 UNICODE 和 ASCII 編碼方法	157
第 11 章 UNIX 平台的擷取程式和引用程式	159
在 UNIX 和 Windows 上設定「擷取程式」和 「引用程式」的環境變數	159
第 14 章 表格結構	159
第 15 章 擷取程式和引用程式訊息	159
附錄 A. 從應用程式內啟動擷取程式和引用程式	160

System Monitor Guide and Reference(無中文版)	161
db2ConvMonStream	161

Troubleshooting Guide(無中文版)	163
Starting DB2 on Windows 95, Windows 98, and Windows ME When the User Is Not Logged On	163
Chapter 2. Troubleshooting the DB2 Universal Database Server	166

在 64 位元的平台上使用 DB2 Universal Database.	169
第 5 章 架構	169
LOCKLIST	169
shmsys:shminfo_shmmax	169
第 6 章 限制	169

XML Extender Administration and Programming(無中文版)	171
--	------------

MQSeries	173
安裝與架構 DB2 MQSeries Function	173
安裝 MQSeries	173
安裝 MQSeries AMI	173
啟用 DB2 MQSeries Function	174
MQSeries 傳訊樣式	174
訊息結構	175
MQSeries 功能概觀	175
限制	176
錯誤碼	176
用法實務手冊	176
基本傳訊	176
傳送訊息	177
擷取訊息	178
應用程式對應用程式連接	179

enable_MQFunctions	182
enable_MQFunctions	183
disable_MQFunctions	184
disable_MQFunctions	185

第4篇 管理工具 187

控制中心	189
管理 DB2 Server for VSE & VM 伺服器能力	189
Java 1.2 支援控制中心	189
在 Windows 作業系統下使用線上說明的「無效捷 徑」錯誤	189
OS/2 系統下的 Java 控制中心	190
在 Windows 作業系統之異動日誌檢視已完成工作的 「檔案拒絕存取」錯誤	190
多位置更新測試連接	190
DB2 for OS/390 的控制中心	190
OS/390 控制中心必要的修正程式	191
「建立空間層」對話框的變更	192
DB2 控制中心的疑難排解	192
UNIX 系統控制中心的疑難排解	192
OS/2 可能的 Infopop 問題	192
jdk11_path 架構參數的說明	192
使用「Script 中心」或「異動日誌」時發生的 Solaris 系統錯誤 (SQL10012N)	192
DPREPL.DFT 檔案的說明	193
啟動多個控制中心 applet	193
控制中心以 Applet 執行時的線上說明	193
在 Applet 模式 (Windows 95) 下執行控制中心	193
使用大的查詢結果	193

資訊中心	195
在 Windows 作業系統下的「無效捷徑」錯誤	195
當 Netscape 已開啓時，在 Netscape Navigator 中開 啓外部 Web 鏈結(UNIX 系統)	195
啓動「資訊中心」的問題	195
精靈	197
在建立資料庫精靈設定擴充大小	197
MQSeries 輔助精靈	197
OLE DB 輔助精靈	197

第5篇 Business Intelligence 199

Business Intelligence 教學指導	203
經過修訂的 Business Intelligence 教學指導	203
資料倉儲中心管理手冊	205
疑難排解	205
設置 Excel 為倉儲來源	205
定義及執行程序	205
匯出描述資料對話	205
為 Submit OS/390 JCL 工作串流程式 (VWPMVS) 定義值	205
變更資料倉儲範例附錄	205

資料倉儲中心訊息	206	定義 OS/2 資料庫中的倉儲來源或目標	231
在 DB2 OLAP Integration Server 中建立框架並載入資料	206	監督倉儲控制資料庫的狀態	231
將 Classic Connect 與資料倉儲中心搭配使用	206	使用 SQL 輔助程式與 TBC_MD 範例資料庫	232
資料倉儲中心環境結構	206	使用 FormatDate 函數	232
使用 Invert 轉換程式	207	變更語言設定	232
以 DB2 版本 7 倉儲代理程式存取 DB2 版本 5 資料	207	使用「產生鍵值表格」轉換程式	232
移轉 DB2 版本 5 伺服器	207	維護資料庫連接	232
變更代理程式架構	207	設定遠端「資料倉儲中心」從屬站	233
IBM ERwin 描述資料取出程式	208	定義 DB2 for VM 倉儲來源	233
內容	208	定義 DB2 for VM 或 DB2 for VSE 目標表格	233
軟體需求	209	啟用有定界符號的識別字支援	233
程式檔	209	Data Joiner 錯誤指出連結問題	233
建立標示語言檔	209	資料倉儲中心設定和執行抄寫	234
匯入標示語言檔到「資料倉儲中心」	211	疑難排解秘訣	236
匯入標示語言檔到「資訊型錄管理程式」	212	存取來源和目標檔	236
疑難排解	212	支援的非 IBM 資料庫來源的新增事項	237
ERwin 與 DB2 資料倉儲中心的對映	213	在資料倉儲中心中手動建立資料來源	239
篩選「資料倉儲中心」的名稱與位址	216	使用 Common Warehouse Metadata Interchange (CWM) 匯入及匯出描述資料	239
撰寫 Trillium 批次系統 JCL 檔	219	簡介	239
在 UNIX 和 Windows 上撰寫 Trillium 批次系統 Script 檔	220	匯入描述資料	239
定義 Trillium 批次系統步驟	220	在執行匯入公用程式之後更新描述資料	241
使用「Trillium 批次系統」使用者定義程式	220	匯出描述資料	241
錯誤處理常式	221	OS/390 Runstats 公用程式步驟	242
MQ Series 與「資料倉儲中心」的整合	223	OS/390 載入公用程式步驟	243
建立 MQSeries 訊息的概略表	223	Common Warehouse Metamodel (CWM) XML 支援	243
匯入 MQSeries 訊息和 XML 描述資料	224	處理模型產生器	243
Microsoft OLE DB 與「資料異動服務」支援	226	綱目模型產生器	244
建立 OLE DB 表格函數的概略表	227	必要輸入欄位	244
建立 DTS 資料包的概略表	227	「資料倉儲中心」發射台加強功能	244
增量 COMMIT 搭配 replace 使用	227	列印步驟資訊到檔案	244
元件追蹤資料檔名	227	Data Warehouse Center Application Integration Guide(無中文版) 245	
AIX 及 Solaris 作業環境上 Sybase 來源需要的 OPEN CLIENT	227	Additional metadata templates	245
範例登錄更正	228	Commit.tag	245
第 3 章 設定倉儲來源	228	ForeignKey.tag	246
對映 Microsoft Access 中的備忘錄欄位至倉儲來源	228	ForeignKeyAdditional.tag	248
第 10 章 維護倉儲資料庫	228	PrimaryKey.tag	250
鏈結表格至 DB2 UDB RUNSTATS 程式的步驟次類型	228	PrimaryKeyAdditional.tag	252
預設倉儲控制資料庫	228	資料倉儲中心線上說明 255	
倉儲控制資料庫管理視窗	229	定義抄寫的表格或概略表	255
變更作用中倉儲控制資料庫	229	執行具有 AS/400 代理程式的 Essbase VWRPs	255
建立和起始設定倉儲控制資料庫	229	使用公佈資料倉儲中心描述資料視窗及已結合的內容視窗	255
建立版本的 SQL 步驟	230	外來鍵	255
在「程序模型」視窗中，變更來源或目標	230	抄寫筆記本	255
新增說明至「資料倉儲中心」物件	230	匯入標示語言	255
執行「範例內容」	230	新增資料的鏈結	255
編輯 Create DDL SQL 陳述式	230	匯入表格	255
移轉 Visual Warehouse 商業檢視	231	RUNSTATS 和 REORGANIZE TABLE 線上說明集合	256
建立目標表格及主要鍵	231	「通知」頁面，「倉儲內容」與「排程」筆記本	256
使用 Merant ODBC 驅動程式	231	「代理程式端」筆記本的代理程式模組欄位	256
新建 ODBC 驅動程式	231		

DB2 OLAP Starter Kit	257	DB2 Query Patroller Client is a Separate Component	281
OLAP Server 網站	257	Migrating from Version 6 of DB2 Query Patroller	
支援的作業系統服務程式等級	257	Using dqpmigrate	281
完成 DB2 OLAP Starter Kit 在 UNIX 上的安裝	257	Enabling Query Management.	281
架構 OLAP Starter Kit 的 ODBC	258	Location of Table Space for Control Tables.	281
在 UNIX 系統架構資料來源	258	New Parameters for dqpstart Command	282
在 UNIX 系統上架構 OLAP 描述資料型錄	259	New Parameter for iwm_cmd Command	282
在 Windows 系統上架構資料來源	260	New Registry Variable:	
在 Windows 系統上架構 OLAP 描述資料型錄	261	DQP_RECOVERY_INTERVAL	283
在您架構資料來源之後	262	Starting Query Administrator.	283
從 OLAP Starter Kit 桌上管理程式登入	262	User Administration	283
Starter Kit 登入範例	263	Creating a Job Queue	283
自行建立和架構 OLAP Starter Kit 的範例資料庫	263	Using the Command Line Interface	283
移轉應用程式至 OLAP Starter Kit 版本 7.2	264	Query Enabler Notes	284
已知問題與限制.	264	DB2 Query Patroller Tracker may Return a Blank	
OLAP 試算表增益集 EQD 檔案遺失	266	Column Page.	284
資訊型錄管理程式管理手冊	267	Query Patroller and Replication Tools.	284
資訊型錄管理程式起始設定公用程式	267	Appendix B. Troubleshooting DB2 Query Patroller	
版權問題	267	Clients	284
安裝問題	267		
以 DB2 版本 7 資訊型錄管理程式存取 DB2 版本 5		第6篇 應用程式開發	285
資訊型錄	268	Administrative API Reference (無中文	版)
設置資訊型錄	269	db2ArchiveLog (new API)	289
與其它產品交換描述資料.	269	db2ArchiveLog	290
使用 flgnxoln 指令交換描述資料	269	db2ConvMonStream.	292
使用 MDISDGC 指令交換描述資料	269	db2DatabasePing (new API)	292
呼叫程式	269	db2DatabasePing - Ping Database	293
Information Catalog Manager		db2HistData	294
Programming Guide and		db2HistoryOpenScan	295
Reference(無中文版)	271	db2XaGetInfo (new API)	296
Information Catalog Manager Reason Codes.	271	db2XaGetInfo - Get Information for Resource	
資訊型錄管理程式使用手冊	273	Manager	297
資訊型錄管理程式：線上訊息	275	db2XaListIndTrans (new API that supercedes	
訊息 FLG0260E.	275	sqlxphqr)	297
訊息 FLG0051E.	275	db2XaListIndTrans - List Indoubt Transactions	298
訊息 FLG0003E.	275	db2GetSnapshot - Get Snapshot.	301
訊息 FLG0372E.	275	Forget Log Record	301
訊息 FLG0615E.	275	sqlaintp - Get Error Message	302
資訊型錄管理程式：線上說明	277	sqlbctcq - Close Tablespace Container Query	302
Web 資訊型錄管理程式	277	sqlubkp - Backup Database	302
DB2 Warehouse Manager 安裝手冊	279	sqlureot - Reorganize Table	302
倉儲轉換程式的軟體需求.	279	sqlurestore - Restore Database	302
SAP R/3 Connector	279	Documentation Error Regarding AIX Extended Shared	
安裝先決條件	279	Memory Support (EXTSHM).	302
Web Connector	279	SQLFUPD	303
安裝先決條件	279	locklist.	303
Query Patroller Administration		SQLEDBDESC	303
Guide(無中文版)	281	SQLFUPD Documentation Error.	303
viii 版次注意事項		應用程式開發手冊	305
		第 1 章 簡介	305
		支援軟體	305
		範例程式	307

第 3 章 開發 DB2 應用程式的一般資訊	307
開發檔案、make 檔和錯誤檢查的公用程式	307
第 4 章 開發 Java Applets 和應用程式	307
設定環境	307
第 5 章 開發 SQL 程序	309
設定 SQL 程序環境	309
設定編譯器環境變數	310
自訂編譯指令	310
保留中間檔	314
備份和復置	314
建立 SQL 程序	315
呼叫儲存程序	315
分送已編譯的 SQL 程序	315
第 7 章 開發 HP-UX 應用程式	316
HP-UX C	316
HP-UX C++	316
第 9 章 開發 OS/2 應用程式	316
VisualAge C++ for OS/2 版本 4.0	316
第 10 章 開發 PTX 應用程式	316
ptx/C++	316
第 12 章 開發 Solaris 應用程式	317
SPARCompiler C++	317
第 13 章 建置 Windows 32 位元作業系統的應用程式	317
VisualAge C++ 版本 4.0	317

Application Development Guide(無中文版) 319

Chapter 2. Coding a DB2 Application.	319
Activating the IBM DB2 Universal Database Project and Tool Add-ins for Microsoft Visual C++	319
Chapter 6. Common DB2 Application Techniques	319
Generating Sequential Values	319
Chapter 7. Stored Procedures	322
DECIMAL Type Fails in Linux Java Routines	322
Using Cursors in Recursive Stored Procedures	322
Writing OLE Automation Stored Procedures	323
Chapter 12. Working with Complex Objects: User-Defined Structured Types	323
Inserting Structured Type Attributes Into Columns	323
Chapter 13. Using Large Objects (LOBs).	323
Large object (LOBs) support in federated database systems	323
Tuning the system	325
Part 5. DB2 Programming Considerations.	325
IBM DB2 OLE DB Provider	325
Chapter 20. Programming in C and C++.	325
C/C++ Types for Stored Procedures, Functions, and Methods	326
Chapter 21. Programming in Java	327
Java Method Signature in PARAMETER STYLE JAVA Procedures and Functions	327
Connecting to the JDBC Applet Server	327
Appendix B. Sample Programs	328

CLI Guide and Reference(無中文版) 329

Binding Database Utilities Using the Run-Time Client	329
Using Static SQL in CLI Applications	329
Limitations of JDBC/ODBC/CLI Static Profiling	329
ADT Transforms	330
Chapter 3. Using Advanced Features	330
Writing Multi-Threaded Applications	330
Scrollable Cursors	331
Using Compound SQL.	332
Using Stored Procedures	332
Chapter 4. Configuring CLI/ODBC and Running Sample Applications	333
Configuration Keywords	333
Chapter 5. DB2 CLI Functions	333
SQLBindFileToParam - Bind LOB File Reference to LOB Parameter	333
SQLNextResult - Associate Next Result Set with Another Statement Handle	333
Appendix D. Extended Scalar Functions	336
Date and Time Functions	336
Appendix K. Using the DB2 CLI/ODBC/JDBC Trace Facility.	336

訊息參考手冊 337

取得訊息與 SQLSTATE 說明	337
在 DB2 Connect 變更的 SQLCODE 重新對映	337
新的與變更的訊息	337
呼叫層次介面 (CLI) 訊息	337
DB2 訊息	337
DBI 訊息	338
資料倉儲中心 (DWC) 訊息	338
SQL 訊息	339
更正 SQLSTATES	341

SQL Reference(無中文版) 343

SQL Reference is Provided in One PDF File	343
Chapter 3. Language Elements	343
Naming Conventions and Implicit Object Name Qualifications.	343
DATALINK Assignments	343
Expressions	343
Chapter 4. Functions	349
Enabling the New Functions and Procedures.	349
Scalar Functions	350
Table Functions	374
Procedures	378
Chapter 5. Queries	381
select-statement/syntax diagram	381
select-statement/fetch-first-clause	381
Chapter 6. SQL Statements	382
Update of the Partitioning Key Now Supported	382
Larger Index Keys for Unicode Databases	382
ALTER SEQUENCE	383
ALTER TABLE.	387
Compound SQL (Embedded).	390
Compound Statement (Dynamic)	390

CREATE FUNCTION (Source or Template)	393	Chapter 3. Using Advanced Features	435
CREATE FUNCTION (SQL Scalar, Table or Row)	394	Appendix C. DB2 CLI and ODBC.	440
CREATE METHOD	395	Data Movement Utilities Guide and Reference(無中文版)	442
CREATE SEQUENCE.	396	Appendix C. Export/Import/Load Utility File Formats	442
CREATE TRIGGER	400		
CREATE WRAPPER	403	第7篇 連接到主電腦系統 443	
DECLARE CURSOR	403	連接環境補充資料 445	
DELETE	403	在 VM 環境下安裝應用程式伺服器	445
DROP	404	CLI/ODBC/JDBC 架構 PATCH1 和 PATCH2 設定值	445
GRANT (Sequence Privileges)	405		
INSERT	406	第8篇 一般資訊 449	
SELECT INTO	407	一般資訊. 451	
SET ENCRYPTION PASSWORD	408	DB2 Universal Database Business Intelligence 快速導覽	451
SET transition-variable.	409	DB2 Everywhere 已改為 DB2 Everyplace	451
UPDATE	412	滑鼠需求	451
Chapter 7. SQL Procedures now called Chapter 7.		試圖從 DB2 Run-time Client 連結導致「找不到連結檔案」錯誤	452
SQL Control Statements	413	Search 探索 (Search Discovery).	452
SQL Procedure Statement.	413	HP-UX 11 的記憶體視窗.	452
FOR	415	dlfm client_conf 失效時使用者之動作.	453
Compound Statement changes to Compound Statement (Procedure)	417	很少見的事件：複製常駐程式在 dlfm 停止時沒有停止	453
RETURN	418	解除安裝 DB2 DFS 從屬站啟動程式.	453
SIGNAL	420	Windows NT 從屬站身分驗證	453
Appendix A. SQL Limits.	423	在建立子程序時 AutoLoader 可能會當掉	454
Appendix D. Catalog Views	423	DATALINK 復置	454
SYSCAT.SEQUENCES	423	Windows NT (CS/NT) 的 IBM Communications 伺服器定義使用者 ID 和通行碼	454
		節點定義	454
IBM DB2 儲存程序建置器 425		聯合系統的限制.	455
Java 1.2 支援儲存程序建置器	425	DataJoiner 限制.	455
DB2 儲存程序的遠端除錯	425	Windows NT 的希伯來文資訊型錄管理程式	455
在 Windows、OS/2 或 UNIX 平台上開發 SQL 程序	426	Microsoft SNA Server 和 SNA 多站更新 (兩階段確定) 支援	456
在 Solaris 平台使用 DB2 儲存程序建置器	426	啟動 Windows 之後無法啟動 DB2 的 SNA SPM	456
已知問題與限制.	426	DB2 管理伺服器的語言環境設定	456
在繁體中文語言環境下使用 DB2 儲存程序建置器	426	捷徑無效	457
UNIX (AIX、Sun Solaris、Linux) 安裝與儲存程序建置器	427	Windows NT 及 Windows 2000 上的 DB2 服務帳戶需求.	457
建置 OS/390 的 SQL 儲存程序	427	在版本 6 中建立的 Query Patroller 使用者失去了 EXECUTE 專用權	457
除錯 SQL 儲存程序	427	Query Patroller 限制	457
匯出 Java 儲存程序	427	需要確定所有會在資料倉儲中心(DWC)中使用的使用者定義程式	458
在 OS/390 上插入儲存程序	427	資料倉儲中心命令行匯出的新選項.	458
在工作站伺服器中設定 SQL 儲存程序的建置選項	428	Backup Services APIs (XBSA)	458
自動復新 OS/390 上儲存程序建置的 WLM 位址空間	428	OS/390 代理程式	459
在 OS/390 開發 Java 儲存程序	428	安裝概觀	459
為 MQ Series 與 OLE DB 建置 DB2 表格使用者定義功能 (UDF)	428	安裝明細	460
Unicode 更新 429			
簡介	429		
DB2 Unicode 資料庫及應用程式	429		
文件更新	431		
SQL Reference(無中文版)	431		
Chapter 3 Language Elements	431		
Chapter 4 Functions	435		
CLI Guide and Reference(無中文版)	435		

設置額外的代理程式函數	462	在 Windows 2000 Terminal Server 管理模式下執行	
用觸發程式排程倉儲步驟 (XTClient)	464	DB2	475
轉換程式	464	備份與復置指令的線上說明	475
存取 DB2 系列之外的資料庫	468	"Warehouse Manager" 應為 "DB2 Warehouse	
執行 DB2 for OS/390 公用程式	469	Manager"	475
抄寫	470		
代理程式記載	471	第9篇 附加資訊	477
Windows NT 上的從屬站端快取	472	附加資訊.	479
Enterprise Edition UNIX CD-ROM 的試用產品	472	DB2 Universal Database 和 DB2 Connect 線上支援	479
DB2 Connect Enterprise Edition UNIX CD-ROM 的		DB2 Magazine	479
試用產品	473		
捨棄 Data Links Manager	473	第10篇 附錄與後記	481
使用 SMIT 解除安裝 DLFM 元件可能會除去額外的		附錄. 注意事項	483
檔案集.	474	商標	485
在 Windows 2000 上使用 CLP 時發生錯誤		索引	487
SQL1035N	474		
SQL 輔助程式的加強功能	474		
Linux 上的 DB2 的 Gnome 和 KDE 桌面整合	474		

歡迎使用 DB2 Universal Database 版本 7 !

註: 設定字型為單距以方便檢視「版次注意事項」。

DB2 Universal Database 與 DB2 Connect 支援網站會定期更新。請跳至：
<http://www.ibm.com/software/data/db2/udb/winos2unix/support>，以取得最新的資訊。

本檔案包含下列的產品資訊；在 DB2 的手冊印製時，尚無法取得這部份的資訊：

- IBM DB2 Universal Database Personal Edition, 版本 7.2
- IBM DB2 Universal Database Workgroup Edition, 版本 7.2
- IBM DB2 Universal Database Enterprise Edition, 版本 7.2
- IBM DB2 Data Links Manager, 版本 7.2
- IBM DB2 Universal Database Enterprise - Extended Edition, 版本 7.2
- IBM DB2 Query Patroller, 版本 7.2
- IBM DB2 Personal Developer's Edition, 版本 7.2
- IBM DB2 Universal Developer's Edition, 版本 7.2
- IBM DB2 Data Warehouse Manager, 版本 7.2
- IBM DB2 Relational Connect, 版本 7.2

另有一個「版次注意事項」檔案為 READCON.TXT，這個檔案提供下列產品的資訊：

- IBM DB2 Connect Personal Edition, 版本 7.2
- IBM DB2 Connect Enterprise Edition, 版本 7.2

「新特性介紹」一書中包含部份的 DB2 版本 7.2 主要增加的概觀。若您沒有「新特性介紹」的版本 7.2，請至 <http://www.ibm.com/software/data/db2/udb/winos2unix/support> 檢視或下載。

註: 在頁面左側的修訂列 (i) 是表示該行自「版本注意事項」第一版之後在此層次中有被新增或修改。

第1篇 特別需知

特別需知

DB2 UDB 版本 7 的存取特性

DB2 UDB 系列產品包括了一些讓行動不便人士容易使用的特性。這些特性是：

- 易於輸入和導引的鍵盤特性
- 增強內容顯示的特性
- 音效和視覺警示之提示選項
- 支援技術的相容性
- 作業系統相容的存取特性
- 可存取文件之格式

鍵盤之輸入和導引

鍵盤輸入

只要使用鍵盤便可以操作 DB2 控制中心。功能表項目和控制項提供存取鍵，讓使用者可以直接從鍵盤啟動控制項或選取某一功能表項目。在控制項或是功能表內出現的存取鍵畫有底線，以方便我們的辨識。

鍵盤焦點

在 UNIX 作業系統內，鍵盤焦點的位置以高亮度顯示，顯示視窗內該區處於作用中，使用者的按鍵會產生影響。

方便顯示特性

DB2 控制中心有一些特性，可以增強使用者介面，以及改善弱視使用者對電腦使用的能力。這些增強的功能亦支援明顯對照的設定值，以及為客戶量身定做的字型內容。

明顯對照模式

控制中心介面支援由作業系統所提供的明顯對照模式選項。使用者可以設定背景顏色和前景顏色，使它形成強烈的對比。

字型設定值

控制中心介面允許使用者自功能表和對話視窗內選取文字的顏色、大小和字型。

無關顏色

使用者不需要去分辨顏色以使用此產品的各項功能。

選擇式警告提示

使用者可以選擇聲音警示或視覺警示。

支援技術的相容性

DB2 控制中心介面和螢幕讀取器應用程式相容，例如透過聲音存取。在應用程式模式內，控制中心介面有一些特性，可以幫助視障使用者得到螢光幕上的資訊。

可存取的文件

DB2 系列產品之文件為 HTML 格式。這可讓使用者以瀏覽器中的喜好設定來檢視文件。它也容許螢幕讀取器和其它協助技術的使用。

附加的必要 Solaris 修補層次

除了列示在 *DB2 for UNIX 快速入門* 手冊內的修補程式之外，還列示了 DB2 Universal Database 版本 7 for Solaris 版本 2.6 所需要的 106285-02 或更新的修補程式。

Supported CPUs on DB2 Version 7 for Solaris

UltraSparc 之前的 CPU 版本並不支援。

新增節點至分割資料庫的問題

在新增節點至分割資料庫時，若此資料庫有一個或多個頁大小非預設值 (4 KB) 的系統暫時表格空間，您可能會得到錯誤訊息：『SQL6073N 新增節點作業失敗』和一個 SQLCODE。會發生這種情形是因為在建立節點時，只有 IBMDEFAULTBP 緩衝池是以 4 KB 的頁大小存在。

例如，您可以使用 **db2start** 指令，新增節點至目前的分割資料庫中。

```
DB2START NODENUM 2 ADDNODE HOSTNAME newhost PORT 2
```

如果分割資料庫的暫時表格空間之頁大小為預設值，結果會傳回下列的訊息：

```
SQL6075W 開始資料庫管理程式作業已順利完成新增節點。  
此節點要等到所有的節點都停止，然後又再啟動時才有作用。
```

但是，如果分割資料庫的系統暫時表格空間之頁大小非預設值，則傳回的訊息是：

```
SQL6073N 新增節點作業失敗。SQLCODE = "<-902>"
```

另一個相似的範例，用新節點說明手動更新 `db2nodes.cfg` 檔案之後，您可以使用 **ADD NODE** 指令。在編輯該檔並對分割資料庫 (系統暫時表格空間為預設頁面大小) 執行 **ADD NODE** 指令之後，會傳回下列訊息：

```
DB20000I ADD NODE 指令順利完成。
```

但是，如果分割資料庫的系統暫時表格空間之頁大小非預設值，則傳回的訊息是：

```
SQL6073N 新增節點作業失敗。SQLCODE = "<-902>"
```

為了避免發生上述的問題，請執行：

```
DB2SET DB2_HIDDENBP=16
```

在發出 **db2start** 或 **ADD NODE** 指令之前。此登錄變數會使 DB2 配置 16 頁的隱藏緩衝池，而每一頁所使用的頁大小並非預設值。這可使 **ADD NODE** 作業順利完成。

另一個可避免這類問題發生的方法，是對 **ADD NODE** 或 **db2start** 指令指定 **WITHOUT TABLESPACES** 子句。在此之後，您必須利用 **CREATE BUFFERPOOL** 陳述式來建立緩衝池，並且利用 **ALTER TABLESPACE** 陳述式將系統暫時表格空間和緩衝池相關聯。

在新增節點至現存節點群組時，若此節點群組有一個或多個頁大小非預設值 (4 KB) 的表格空間，您可能得到錯誤訊息：『SQL0647N 緩衝池 "" 目前不在作用中。』。會發生這種情形是因為建立在新節點且頁大小非預設值的緩衝池並未為表格空間啟動。

例如，您可以使用 ALTER NODEGROUP 陳述式新增節點至節點群組：

```
DB2START
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2)
```

如果節點群組有頁大小為預設值的表格空間，結果會傳回下列的訊息：

```
SQL1759W 重新分配節點群組必須變更節點群組 "<ng1>" 內物件的資料定位，
以併入新增節點，
或排除捨棄節點。
```

但是，如果節點群組的表格空間之頁大小非預設值，則傳回的訊息是：

```
SQL0647N 緩衝池 "" 目前不在作用中。
```

一個可避免此問題的方法是為每一個頁大小建立緩衝池，並在發出 ALTER NODEGROUP 陳述式之前，重新連接至資料庫：

```
DB2START
CONNECT TO mpp1
CREATE BUFFERPOOL bp1 SIZE 1000 PAGESIZE 8192
CONNECT RESET
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2)
```

第二個避免問題的方法，請執行：

```
DB2SET DB2_HIDDENBP=16
```

發出 **db2start** 指令以及 CONNECT 和 ALTER NODEGROUP 陳述式之前。

使用 ALTER TABLESPACE 陳述式新增表格空間至節點，另一個問題會發生。例如：

```
DB2START
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2) WITHOUT TABLESPACES
ALTER TABLESPACE ts1 ADD ('ts1') ON NODE (2)
```

這一系列指令和陳述式會產生錯誤訊息 SQL0647N（而非預期訊息 SQL1759W）。

欲正確地完成此變更，必須在 ALTER NODEGROUP... WITHOUT TABLESPACES 陳述式之後，重新連接至資料庫。

```
DB2START
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2) WITHOUT TABLESPACES
CONNECT RESET
CONNECT TO mpp1
ALTER TABLESPACE ts1 ADD ('ts1') ON NODE (2)
```

另一個避免問題的方法，請執行：

```
DB2SET DB2_HIDDENBP=16
```

發出 **db2start** 指令以及 CONNECT、ALTER NODEGROUP 和 ALTER TABLESPACE 陳述式之前。

移轉期間發生的錯誤

移轉期間，即使移轉成功，錯誤項目仍會出現在 db2diag.log 檔案內（資料庫未移轉）。此時，可予以忽略。

Red Flag Linux 的中文語言環境修正程式

如果您使用「簡體中文紅旗 Linux 伺服器版本 1.1」，請洽詢「紅旗」以取得簡體中文語言環境修正程式。沒有版本 1.1 的簡體中文語言環境修正程式，DB2 就無法辨識簡體中文的字碼頁是 1386。

如果可移除式磁碟機未連接，則 DB2 安裝會中斷

在安裝 DB2 的期間，如果電腦的可移除式磁碟機未連接，則在選取安裝類型之後，安裝程式可能會中斷。若要解決此問題，在執行安裝程式時指定 -a 選項：

```
setup.exe -a
```

日文和簡體中文 Linux 環境之 DB2 for Linux 的附加語言環境設定

當您要在日文或簡體中文 Linux 系統上使用 Java GUI 工具（例如「控制中心」）時，則需要額外的語言環境設定。若沒有這項設定則無法正確顯示日文或中文字元。請在您的使用者設定檔裡加入下列設定，或者在每個「控制中心」呼叫之前從指令行執行此設定。

```
針對日文系統：  
export LC_ALL=ja_JP
```

```
針對簡體中文系統：  
export LC_ALL=zh_CN
```

Microsoft Internet Explorer 上的控制中心問題

這是 Internet Explorer 安全性選項設定所造成的問題。「控制中心」使用未簽署的 jar，因此存取系統資訊被安全管理程式停用了。

若要消除這個問題，請重新配置 IE 的安全性選項，如下所示：

1. 選取**檢視功能表 (IE4)** 或**工具功能表 (IE5)** 上的 **Internet** 選項。
2. 在「安全性」頁面，選取**信任的網站區域**。
3. 按一下**新增網站...**。
4. 將「控制中心 Web 伺服器」新增到信任的網站清單。如果控制中心 Web 伺服器在相同的網域中，可能只須要將 Web 伺服器名稱加入即可（不含網域名稱）。例如：

```
http://ccWebServer.ccWebServerDomain  
http://ccWebServer
```
5. 按一下**確定**。
6. 按一下**設定...**。
7. 捲動到下方的 **Java -> Java** 權限，再選取**自訂**。
8. 按一下 **Java 自訂設定...**。
9. 選取「**編輯許可權**」頁面。

10. 捲動到下方的「未簽名的內容 --> 執行未簽名的內容 --> 其它未簽名的許可權 --> 系統資訊」，再選取**啓動**。
11. 按一下每個開啓視窗的**確定**。

在 Windows 環境中「資訊型錄管理程式」及 Sybase 之間的不相容

「資訊型錄管理程式」版本 7 和 Sybase Open Client 安裝在同一台 Windows NT 或 Windows 2000 機器會造成錯誤，且 Sybase 公用程式會停止運作。會出現類似下列的錯誤訊息：

無法起始 LIBTCL.DLL。請確定 SYBASE 環境變數已正確設定。

避免這種情境的方法是從 Windows 環境參數中將環境參數 LC_ALL 除去。LC_ALL 是一個語言環境種類參數。語言環境種類是本土化常式所使用的明顯的常數，以指定程式要使用的語言環境資訊的部份。語言環境指的是地區(或國家)，程式的某些方面可針對地區自訂。語言環境相關區域包括，例如日期格式或貨幣符號的顯示格式。LC_ALL 會影響所有的語言環境特定行為 (所有種類)。

如果您將 LC_ALL 環境參數除去，這樣 ICM 就可和 Sybase 同時存在 Windows NT 平台上，下列機能會無法運作：

- 資訊型錄使用者
- 資訊型錄管理者
- 資訊型錄管理程式

控制中心功能流失

可將 FixPak 2 引用到 DB2 伺服器，則對於舊版的控制中心從屬站，應該不會有此處所描述的問題。然而，在 DB2 版本 7.2，舊版控制中心從屬站幾乎失去了所有的功能。這裡所說的舊版是指任何版本 6 FixPak 6 以前的從屬站，以及任何版本 7 FixPak 2 以前的從屬站。版本 5 的從屬站並不受影響。

建議的修正方式是升級任何受到影響的從屬站。版本 6 從屬站必須升級到 FixPak 6 或更新版本，版本 7 從屬站必須升級到 FixPak 2 或更新版本。

Netscape CD 不隨附於 DB2 UDB

Netscape CD 不再隨附於 DB2 UDB。請從 <http://www.netscape.com>，獲取 Netscape 產品。

XML Readme 檔的錯誤

在 DB2 XML Extender 版本 7.1 README.TXT 檔的「注意事項」中提到：

3. DB2 UDB 的預設版本是 DB2 UDB 版本 7.1。若您要在 AIX 與 Solaris 使用 DB2 UDB 版本 6.1，請確定您應該與 DB2 UDB 版本 6.1 案例及 DB2 UDB 版本 6.1 檔案庫一起執行。

這是不正確的。**DB2 XML Extender 僅在 DB2 版本 7.1 與 7.2 被支援。**

檔案 readme.aix、readme.nt 與 readme.sun 所列示的軟體需求為：

- DB2 UDB 6.1 (FP1_U465423) 或以上 (AIX)
- DB2 Universal Database 版本 6.1 或以上 (安裝 FixPak 3) (NT)
- DB2 UDB 版本 6.1 (FixPak FP1_U465424) 或以上 (Sun)

這是不正確的。DB2 XML Extender 需要 DB2 版本 7.1 或 7.2。

Linux for S/390 上可能的資料流失

若在 Linux for S/390 (已安裝 2.2 系列核心程式) 使用 DB2，該 Linux 電腦上的可用 RAM 數量限制將會小於 1 GB。限制 RAM 到 1 GB 能避免因 Linux 核心程式錯誤造成的可能性 DB2 資料流失。

這只發生於 DB2 在 Linux for S/390，不發生於 Linux on Intel。

您可從 http://www10.software.ibm.com/developerworks/opensource/linux390/alpha_src.html 取得核心修補程式以讓 RAM 可使用超過 1 GB。

Windows 2000 上的 DB2 UDB

在整個「版本注意事項」中，凡適用於 Windows NT 的參考資料 (除非另有指定) 尚適用於 Windows 2000。

線上文件 (HTML、PDF 與「搜尋」)

Windows 2000 作業系統所支援的 Web 瀏覽器

我們建議您在 Windows 2000 上使用 Microsoft Internet Explorer。

若您使用 Netscape，請注意下列事項：

- 使用 Netscape 會讓 Windows 2000 上的 DB2 連線資訊搜尋耗時較長。Netscape 將使用所有可用的 CPU 資源，並且似乎是不明確地執行。當最後傳回搜尋結果時，我們建議您在提出搜尋之後，在另一個視窗上按一下以變更焦點。接著會在合理的時間量裡傳回搜尋結果。
- 您可能會注意到：當您要求的說明正確地顯示在一個 Netscape 瀏覽器視窗內，然而，如果瀏覽器視窗一直開啓著，且稍後又從不同的「控制中心」部份要求說明，則瀏覽器裡不會有任何改變。若關閉瀏覽器視窗並再次要求說明，將會出現正確的說明。您可透過『啓動 Netscape 時產生的錯誤訊息』裡的下列步驟來修正這個問題。您也可以要求在要求「控制中心」說明之前關閉瀏覽器視窗，以避免這個問題。
- 要求「控制中心」的說明，或者「資訊中心」的主題時，您可能會收到錯誤訊息。要修正這個問題，請遵循『啓動 Netscape 時產生的錯誤訊息』裡的步驟。

在 Solaris 系統下搜尋 DB2 連線資訊

如果您有在 Solaris 系統下搜尋 DB2 連線資訊的問題，請檢查系統在 /etc/system 內的核心程式參數。以下是 DB2 搜尋系統所需核心程式參數的最小值，NetQuestion：

```
semsys:seminfo_semni 256
semsys:seminfo_semmap 258
semsys:seminfo_semms 512
semsys:seminfo_semmsl 512
semsys:seminfo_semmsl 50
shmsys:shminfo_shmmax 6291456
shmsys:shminfo_shmseg 16
shmsys:shminfo_shmmni 300
```

欲設定核心程式參數，在 /etc/system 之後新增下列指令：

```
set <semaphore_name> = value
```

欲使任何新建或變更的值生效，您必須重新啓動系統。

切換 NetQuestion for OS/2 以使用 TCP/IP

切換 NetQuestion 以在 OS/2 系統上使用 TCP/IP 的指示不完整。這些指示的 *.cfg 檔位置是 NetQuestion 安裝目錄的資料次目錄。您可以輸入下列其中一項指令來決定 NetQuestion 安裝目錄：

```
echo %IMNINSTSRV% //適用 SBCS 安裝
echo %IMQINSTSRV% //適用 DBCS 安裝
```

啓動 Netscape 時產生的錯誤訊息

當您啓動 Netscape 時，如果您發現下列的錯誤訊息：

找不到<檔案路徑>之檔案 (或是其中的元件)。
檢查並確定路徑和檔名的正確性，以及所有必要的檔案庫為有效性。
無法開啓 "D:\Program Files\SQLLIB\CC\..\doc\html\db2help\XXXXX.htm"

您應該執行下列步驟，以更正在 Windows NT、95 或 98 上的這個問題 (請參閱以下步驟，以了解如何在 Windows 2000 上執行)：

1. 從「開始」功能表，選取「程式集」—>「Windows 檔案總管」。即可開啓「Windows 檔案總管」。
2. 從「Windows 檔案總管」選取「檢視」—>「資料夾選項」。即可開啓「資料夾選項」筆記本。
3. 按一下**檔案類型**標籤。開啓「檔案類型」頁。
4. 在**登錄檔案類型欄位**中選取 Netscape Hypertext Document，並按一下「編輯」。即可開啓「編輯」檔案類型視窗。
5. 選取**動作**欄位中的開啓。
6. 按一下**編輯**按鈕。類型視窗的編輯動作開啓。
7. 取消使用 **DDE** 的勾選框。
8. 在**用來執行動作的應用程式**欄位中，請確定 "%1" 出現在字串的最後面 (包括引號以及第一個引號之前的空格)。

若在 Windows 2000 上發現錯誤訊息，您應該執行下列步驟：

1. 從**開始**功能表，選取 **Windows 檔案總管**。即可開啓「Windows 檔案總管」。
2. 從「Windows 檔案總管」，選取**工具 --> 資料夾選項**。即可開啓「資料夾選項」筆記本。
3. 按一下**檔案類型**標籤。
4. 在「檔案類型」頁上、**已登錄的檔案類型** 欄位中，強調顯示：HTM Netscape Hypertext Document，再按一下**進階**。即可開啓「編輯檔案類型」視窗。
5. 強調顯示**動作**欄位中的「開啓」。
6. 按一下**編輯**按鈕。即可開啓「編輯類型的動作」視窗。
7. 取消使用 **DDE** 的勾選框。
8. 在**用來執行動作的應用程式**欄位中，請確定 "%1" 出現在字串的最後面 (包括引號以及第一個引號之前的空格)。
9. 按一下**確定**。
10. 針對 HTML Netscape Hypertext Document 和 SHTML Netscape Hypertext Document 檔案類型重複步驟 4 到 8。

UNIX 系統上 Adobe Acrobat Reader 的架構需求

在 UNIX 型平台，Acrobat Reader 只提供英文版。欲以英文以外的語言環境開啓 PDF 檔案，系統會傳回錯誤訊息。訊息指出 PDF 檔案的字型存取或解壓縮產生錯誤，但是，實際上是因為英文版的 Acrobat Reader 無法在非英文語言環境的 UNIX 中執行。

欲檢視此 PDF 檔案，啓動英文版的 Acrobat Reader 前，先執行下列其中一項步驟，切換至英文的語言環境：

- 編輯 Acrobat Reader 的啓動 script，在啓動 script 內，`#!/bin/sh` 陳述式之後新增下列指令：

```
LANG=C;export LANG
```

這樣的話，當 Acrobat Reader 被其它的應用程式啓動，如被 Netscape 導引器或應用程式說明功能表啓動時，會產生正確的行為。

- 在命令提示內輸入 LANG=C，將 Acrobat Reader 的應用程式環境設定為英文。

進一步資訊，請連線 Adobe 系統 (<http://www.Adobe.com>)。

SQL Reference 僅提供一個 PDF 檔

每本書籍的「使用 DB2 檔案庫」附錄中，指出 SQL Reference 為 PDF 格式且有兩冊。
This is incorrect.

雖然列印的書籍有兩冊，而且兩冊相對應的書號都正確，但是只有一個 PDF 檔案，它包含了兩冊。PDF 檔名是 db2s0x70。

第2篇 安裝與架構

一般安裝資訊	15	使用 DLFM：從 DB2 資料庫捨棄 DB2 Data Links Manager 後的清除動作	29
下載所有支援的 DB2 從屬站的安裝套件	15	DLFM1001E (新的錯誤訊息)	29
在 Windows 2000 上安裝 DB2	15	DLFM 設定架構檔選項	30
關於概略表以特別暫存區定義的移轉問題	15	在 AIX 上執行 Data Links/DFS Script dmapp_prestart 時發生錯誤	30
Windows 2000 的 IPX/SPX 通信協定支援	15	Tivoli Space Manager 與 Data Links 整合	30
升級 DB2 前一版之前先停止 DB2 程序	15	功能限制	31
若已安裝其它 DB2 產品，請在安裝 DB2 之後執行 db2iupdt	16	第 4 章 安裝和架構 DB2 Data Links Manager for AIX	31
設定 Linux 環境以執行 DB2 控制中心	16	一般安裝注意事項	31
DB2 Universal Database Enterprise Edition 及 DB2 Connect Enterprise Edition for Linux on S/390	16	從 DB2 File Manager 版本 5.2 移轉到 DB2 Data Links Manager 版本 7	31
DB2 Universal Database Enterprise - Extended Edition for UNIX 快速入門	17	安裝與架構補充資料	33
HP-UX 的 shmseg 核心參數	17	第 5 章 在 UNIX 作業系統上安裝 DB2 從屬站	33
移轉 IBM Visual 倉儲控制資料庫	17	HP-UX 核心程式架構參數	33
存取倉儲控制資料庫	17	第 12 章 執行您自己的應用程式	33
DB2 Data Links Manager 快速入門	19	使用 Run-Time Client 連結資料庫公用程式	33
Dlfm 的啟動失敗訊息：取得 afsfid 字首時發生錯誤	19	UNIX 從屬站使用 ODBC 存取 DB2	33
設定備份保存檔的 Tivoli Storage Manager 類別	19	第 24 章 設置聯合系統來存取多重資料來源	33
DFS 從屬站啟動程式的磁碟空間需求	19	聯合系統	33
監督在 AIX 的資料鏈結檔案管理程式後端程序	19	限制	34
安裝和架構 DB2 Data Links Manager for AIX：DCE-DFS 環境裡的額外安裝注意事項	19	安裝 DB2 Relational Connect	34
失敗的 "dlfm add_prefix" 指令	20	在 Windows NT 伺服器上安裝 DB2 Relational Connect	34
安裝和架構 DB2 Data Links Manager for AIX：使用 db2setup 公用程式在 AIX 上安裝 DB2 Data Links Manager	20	在 AIX、Linux 及 Solaris 作業環境伺服器上安裝 DB2 Relational Connect	35
安裝和架構 DB2 Data Links Manager for AIX：DCE-DFS 後置安裝作業	20	第 26 章 存取 Oracle 資料來源	35
安裝和架構 DB2 Data Links Manager for AIX：使用 Smit 來手動安裝 DB2 Data Links Manager	21	文件錯誤	36
安裝和架構 DB2 Data Links 的 DFS 從屬站啟動程式	21	存取 Sybase 資料來源 (新增章節)	36
安裝和架構 DB2 Data Links Manager for Solaris	21	新增 Sybase 資料來源至聯合同伺服器	37
針對 AIX 上的 DB2 Data Links Manager 選擇備份方法	22	步驟 1：設定環境變數並更新設定檔登錄	37
針對 Solaris 作業環境上的 DB2 Data Links Manager 選擇備份方法	23	步驟 2：將 DB2 鏈結至 Sybase 從屬站軟體 (僅 AIX 與 Solaris)	38
針對 Windows NT 上的 DB2 Data Links Manager 選擇備份方法	24	步驟 3：重新使用 DB2 案例	38
備份 AIX 上的記錄檔案系統	25	步驟 4：建立並設定介面檔案	38
在 Windows NT 上資料鏈結中的 Administrator 群組專用權	26	步驟 5：建立外層	39
將 Data Links File System Filter (DLFF) 安裝的記載減到最少	26	步驟 6：選用的：設定 DB2_DJ_COMM 環境變數	39
在安裝後記載訊息	27	步驟 7：建立伺服器	39
使用 SMIT 解除安裝 DLFM 元件可能會除去額外的檔案集	27	選用的：步驟 8：設定 CONNECTSTRING 伺服器選項	40
開始之前/決定主電腦名稱	28	步驟 9：建立使用者對映	41
		步驟 10：建立表格與概略表的暱名	41
		指定 Sybase 字碼頁	42
		使用 ODBC 存取 Microsoft SQL Server 資料來源 (新增章節)	42
		新增 Microsoft SQL Server 資料來源至聯合同伺服器	43
		步驟 1：設定環境變數 (僅 AIX)	43
		步驟 2：執行 Shell Script (僅 AIX)	44

	步驟 3：選用的：設定 DB2_DJ_COMM 環境變數	
	數	44
	步驟 4：重新使用 DB2 案例 (僅 AIX)	44
	步驟 5：建立外層	44
	步驟 6：建立伺服器	45
	步驟 7：建立使用者對映	46
	步驟 8：建立表格與概略表的暱名	46
	步驟 9：選用的：獲得 ODBC 追蹤	47
	複查 Microsoft SQL Server 字碼頁	47

一般安裝資訊

下載所有支援的 DB2 從屬站的安裝套件

若要下載所有支援的 DB2 從屬站的安裝套件，包括所有版本 7 之前的從屬站，請連線到 IBM DB2 Client Application Enabler Pack 的網頁，其網址為：
<http://www.ibm.com/software/data/db2/db2tech/clientpak.html>

在 Windows 2000 上安裝 DB2

在 Windows 2000 上，安裝在前版 DB2 之上時，或是重新安裝現行版本時，請確定將所有 DB2 服務的回復選項設定為「Take No Action」。

關於概略表以特別暫存區定義的移轉問題

資料庫移轉之後，如果特別暫存區 USER 或 CURRENT SCHEMA 被用來定義概略表直欄，則概略表會變成無法使用。例如：

```
create view v1 (c1) as values user
```

在版本 5，USER 和 CURRENT SCHEMA 的資料類型是 CHAR(8)。自版本 6 起，它們被定義為 VARCHAR(128)。本範例中，如果概略表是由版本 5 所建立，直欄 c1 的資料類型則為 CHAR。移轉之後使用概略表，它會在執行時間編譯，但是，它會因為資料類型不符而失敗。

解決方案是放棄並重新建立該概略表。在捨棄概略表之前，查詢 SYSCAT.VIEWS 型錄概略表來擷取用來建立概略表的語法。例如：

```
select text from syscat.views where viewname='<>'
```

Windows 2000 的 IPX/SPX 通信協定支援

此資訊是參照「快速入門」一書的「安裝」這一章的「可能的主從連接實務」這一節。

已公佈的通信協定支援圖表並不完全正確。不支援使用 IPX/SPX 連接 OS/2 或 UNIX 型伺服器的 Windows 2000 從屬站。也不支援使用 IPX/SPX 連接 Windows 2000 伺服器的 OS/2 或 UNIX 型從屬站。

升級 DB2 前一版之前先停止 DB2 程序

此資訊是參照「DB2 for Windows 快速入門」一書中的移轉資訊。

若您要升級在 Windows 機器上執行的前一版 DB2，安裝程式所提供的警告包含了將 DB2 DLL 保留在記憶體的程序列示。在此時，您可以將列示裡的程序以手動方式停止，或者讓安裝程式自動關掉這些程序。建議您在安裝之前手動停止所有 DB2 程序，以避免流失資料。要確定 DB2 程序不在執行中的最佳方式是透過「Windows 服務」畫面來檢視系統的程序。在「Windows 服務」畫面中，請確定沒有執行 DB2 服務、OLAP 服務、或資料倉儲服務。

註: Windows 平台上一次只能執行一種版本的 DB2。例如，您無法在同樣的 Windows 機器上同時執行 DB2 版本 7 和 DB2 版本 6。若您要將 DB2 版本 7 安裝在已安裝 DB2 版本 6 的機器上，安裝程式會在安裝期間刪除 DB2 版本 6。請參照適當的快速入門手冊，以取得從 DB2 的前一版來移轉的相關資訊。

若已安裝其它 DB2 產品，請在安裝 DB2 之後執行 db2iupdt

下列資訊在您的「快速入門」安裝文件中已提供。

在 UNIX 型系統上安裝 DB2 UDB 版本 7 時，若已安裝 DB2 產品，您需要執行 **db2iupdt** 指令，以更新您要使用本產品新特性的案例。有些特性要在執行這個指令之後才能使用。

設定 Linux 環境以執行 DB2 控制中心

此資訊內含於「快速入門」一書的「安裝 DB2 控制中心」這一章。

離開 Linux 的 DB2 安裝程式並回到終端機視窗之後，請鍵入下列指令、設定正確的環境，以執行 DB2「控制中心」：

```
su -l <案例名稱>
export JAVA_HOME=/usr/jdk118
export DISPLAY=<機器名稱>:0
```

然後，開啓另一個終端機視窗並鍵入：

```
su root
xhost +<機器名稱>
```

關閉該終端機視窗，返回您登入為案例擁有者 ID 的終端機，並鍵入指令：

```
db2cc
```

以啓動「控制中心」。

DB2 Universal Database Enterprise Edition 及 DB2 Connect Enterprise Edition for Linux on S/390

DB2 Universal Database Enterprise Edition 及 DB2 Connect Enterprise Edition 現在可在 Linux/390 上使用了。在安裝 Linux 到 S/390 機器上之前，您應該要意識到軟體及硬體的需求：

硬體

S/390 9672 Generation 5 或以上，Multiprise 3000。

軟體

- SuSE Linux v7.0 for S/390 或 Turbolinux Server 6 for zSeries 及 S/390
- 核心層次 2.2.16，含 S/390 修補程式 (請見下述)
- glibc 2.1.3
- libstdc++ 6.1

Linux/390 需要下列修補程式：

- 此時不需要修補程式。

若需最新的更新資訊，請至 <http://www.software.ibm.com/data/db2/linux> 網站。

註:

1. 只支援 32 位元的 Intel-based Linux 及 Linux/390。
2. Linux/390 的 DB2 版本 7 未提供下列項目：
 - DB2 UDB Enterprise - Extended Edition
 - DB2 Extenders
 - Data Links Manager
 - DB2 Administrative Client
 - 變更通行碼支援
 - LDAP 支援

DB2 Universal Database Enterprise - Extended Edition for UNIX 快速入門

第 5 章「在 Linux 上安裝和架構 DB2 Universal Database」指出 Linux EEE 叢集中的每一個實體節點必須具備相同的 kernel、glibc 及 libstdc++ 層次。

DB2 EEE for Linux 試用版可從下列網站下載：

<http://www6.software.ibm.com/dl/db2udbd1/db2udbd1-p>

HP-UX 的 shmseg 核心參數

您的「快速入門」一書中提供有關更新 HP-UX 核心架構參數的資訊不正確。應該忽略 HP-UX 的 shmseg 核心參數的建議值。

應改用預設的 HP-UX 值 (120)。

移轉 IBM Visual 倉儲控制資料庫

DB2 Universal Database for Windows 快速入門 提供有關 Windows NT 和 Windows 2000 進行 DB2 Universal Database 版本 7 典型安裝期間，如何移轉作用中倉儲控制資料庫的資訊。如果要移轉一個以上的倉儲控制資料庫，則必須使用倉儲控制資料庫管理視窗進行附加資料庫的移轉。一次只能有一個倉儲控制資料庫作用。重新登入資料倉儲中心時，如果上一個資料庫不是您要使用的資料庫，則必須使用倉儲控制資料庫管理視窗來進行登記您所要使用的資料庫。

存取倉儲控制資料庫

於 Windows NT 安裝 DB2 版本 7，建立 DB2 版本 7 倉儲控制資料庫和倉儲伺服器。如果您有名為 Visual Warehouse 的倉儲控制資料庫，請將含有倉儲控制資料庫的 DB2 伺服器升級到 DB2 版本 7，倉儲控制資料庫的描述資料會移轉，以供 DB2 版本 7 資料倉儲中心所使用。欲繼續使用版本 7，請移轉所有的倉儲控制資料庫。DB2 版本 7 安裝期間，作用中的倉儲控制資料庫的描述資料會移轉到版本 7。欲移轉附加倉儲控制資料庫的描述資料，請使用倉儲控制資料庫移轉公用程式。於 Windows NT，選取**開始 --> 程式集 --> IBM DB2 --> 倉儲控制資料庫管理**，進行啓動。有關倉儲控制資料庫的移轉，請參閱 *Windows DB2 Universal Database 快速入門*

DB2 Data Links Manager 快速入門

Dlfm 的啟動失敗訊息：取得 afsfid 字首時發生錯誤

針對在 DCE-DFS 環境裡執行的 Data Links Manager，如果 dlfm 因下列錯誤啟動失敗，請洽詢 IBM 服務中心：

取得 afsfid 字首時發生錯誤

當使用 "dlfm add_prefix" 登記到 Data Links Manager 的 DFS 檔案集被刪除時，將會發生錯誤。

設定備份保存檔的 Tivoli Storage Manager 類別

欲指定備份保存檔使用那一個 TSM 管理類別，請設定 DLFM_TSM_MGMTCLASS DB2 登錄項目一個適當的管理類別名稱。

DFS 從屬站啟動程式的磁碟空間需求

DFS 從屬站啟動程式為選用的元件，您可以在 DB2 Universal Database 從屬站或伺服器安裝期間作選擇。雖然 DFS 從屬站啟動程式執行時並不需要 DB2 UDB 從屬站或伺服器，但是，安裝 DFS 從屬站啟動程式之前您必須安裝 DB2 Universal Database 從屬站。除了 DFS 從屬站啟動程式字碼需要 2 MB 的磁碟空間外，在安裝 DFS 從屬站啟動程式為 DB2 執行時間從屬站安裝的一部份時，您應該另外保留 40 MB 的額外磁碟空間。安裝 DFS 從屬站啟動程式為 DB2 管理從屬站或 DB2 伺服器安裝之一部份時，您需要更多的磁碟空間。有關 DB2 Universal Database 產品所需要的磁碟空間，詳細資訊請參閱 *DB2 for UNIX 快速入門手冊*。

監督在 AIX 的資料鏈結檔案管理程式後端程序

`dlfm see` 指令的輸出有變更。發出此指令來監督在 AIX 的資料鏈結檔案管理程式後端程序，結果與下列所示類似：

PID	PPID	PGID	RUNAME	UNAME	ETIME	DAEMON NAME
17500	60182	40838	dlfm	root	12:18	dlfm_copyd_(dlfm)
41228	60182	40838	dlfm	root	12:18	dlfm_chownd_(dlfm)
49006	60182	40838	dlfm	root	12:18	dlfm_upcall_(dlfm)
51972	60182	40838	dlfm	root	12:18	dlfm_gcd_(dlfm)
66850	60182	40838	dlfm	root	12:18	dlfm_retrieved_(dlfm)
67216	60182	40838	dlfm	dlfm	12:18	dlfm_deigrpd_(dlfm)
60182	1	40838	dlfm	dlfm	12:18	dlfmd_(dlfm)

DLFM SEE 要求成功。

括弧內的名稱為 dlfm 案例的名稱，本例名稱為 "dlfm"。

安裝和架構 DB2 Data Links Manager for AIX：DCE-DFS 環境裡的額外安裝注意事項

在「安裝先決條件」區段裡，應該新增的新資訊：

您也必須安裝 e-fix for DFS 3.1，或者 PTF 集 1 (變成可用時)。
可在以下網站找到 e-fix 的資訊：
http://www.transarc.com/Support/dfs/datalinks/efix_dfs31_main_page.html

以及：

必須在安裝 Data Links Manager 之前執行 dfs 從屬站。
使用 db2setup 或 smitty。

在「Keytab 檔案」區段裡，有個錯誤應該更正為：

含有主體和通行碼資訊的 keytab 檔案，應該稱為 datalink.ktb 和

正確的名稱：datalink.ktb 是用在下面的範例中。「Keytab 檔案」區段應該移到「DCE-DFS 後置安裝作業」之下，因為 DLMADMIN 案例建立後，這個檔案才能建立。

在「Data Links File Manager 伺服器與從屬站」中，請注意，必須在 Data Links Manager 從屬站之前先安裝 Data Links Manager 伺服器。

應該新增「備份目錄」章節：

若備份方法是針對本端檔案系統，
這必須是 DFS 檔案系統裡的目錄。
請確定 DFS 管理者已建立此 DFS 檔案集。
此檔案集應該不是 DMLFS 檔案集。

失敗的 "dlfm add_prefix" 指令

針對在 DCE/DFS 環境裡執行的 Data Links Manager，**dlfm add_prefix** 指令可能會失敗，而回覆碼為 -2061 (備份失敗)。若發生此狀況，請執行下列步驟：

1. 發出 **dlfm stop** 指令來停止 Data Links Manager 的常駐程式程序。
2. 發出 **dlfm stopdbm** 指令來停止 DB2 程序。
3. 發出 **dce_login root** 指令來取得 dce root 認證。
4. 發出 **dlfm startdbm** 指令來啟動 DB2 程序。
5. 發出 **dlfm add_prefix** 指令來登記 Data Links Manager 的檔案集。
6. 發出 **dlfm start** 指令來啟動 Data Links Manager 常駐程式程序。

安裝和架構 DB2 Data Links Manager for AIX：使用 db2setup 公用程式在 AIX 上安裝 DB2 Data Links Manager

在「已建立 DB2 資料庫 DLFM_DB」區段中，DLFM_DB 並不是建立在 DCE_DFS 環境中。必須以後置安裝步驟來執行。

在「DMAPP 的 DCE-DFS 預先啟動登錄」區段中，步驟 2 應該變更如下：

2. 新增指令到 /opt/dcelocal/tcl/user_cmd.tcl，以確定 DFS 啟動時也啟動了 DMAPP。

安裝和架構 DB2 Data Links Manager for AIX：DCE-DFS 後置安裝作業

應該新增下列「完成 Data Links Manager 安裝」章節：

在 Data Links Manager 伺服器上，必須執行下列步驟，以完成安裝：

1. 在「安裝和架構 DB2 Data Links Manager for AIX」這章的「DCE-DFS 環境的額外安裝注意事項」這節裡，將 keytab 檔建立為「Keytab 檔案」之下的概要。
2. 以 root 身分，輸入下列指令以啟動 DMAPP：


```
stop.dfs all
start.dfs all
```
3. 如下列步驟，使用 dce root 認證來執行 "dlfm setup"。
 - a. 以 Data Links Manager 的 DLMADMIN 管理者登入。
 - b. 以 root 身分，發出 dce_login。
 - c. 輸入指令：dlfm setup。

在 Data Links Manager 從屬站上，必須執行下列步驟，以完成安裝：

1. 在「安裝和架構 DB2 Data Links Manager for AIX」這章的「DCE-DFS 環境的額外安裝注意事項」這節裡，將 keytab 檔建立為「Keytab 檔案」之下的概要。
2. 以 root 身分，輸入下列指令以啟動 DMAPP：


```
stop.dfs all
start.dfs all
```

安裝和架構 DB2 Data Links Manager for AIX：使用 Smit 來手動安裝 DB2 Data Links Manager

根據「SMIT 後置安裝作業」章節，修改步驟 7 以指出 "dce_login root" 指令必須在 "dlfm setup" 之前發出。不需要步驟 11。在步驟 6 (dlfm server_conf) 和步驟 8 (dlfm client_conf) 完成時，會自動執行此步驟。也請移除步驟 12 (dlfm start)。若要完成安裝，請執行下列步驟：

1. 在「安裝和架構 DB2 Data Links Manager for AIX」這章的「DCE-DFS 環境的額外安裝注意事項」這節裡，將 keytab 檔建立為「Keytab 檔案」之下的概要。
2. 以 root 身分，輸入下列指令以啟動 DMAPP：


```
stop.dfs all
start.dfs all
```

安裝和架構 DB2 Data Links 的 DFS 從屬站啟動程式

在「架構 DFS 從屬站啟動程式」章節裡，將下列資訊加到步驟 2：

執行 "secval" 指令通常會完成架構。
不過，也可能需要重新啟動機器。
若在存取 READ PERMISSION DB 檔案時發現問題，
請重新啟動安裝 DB2 DFS 從屬站啟動程式的機器。

安裝和架構 DB2 Data Links Manager for Solaris

在安裝 DB2 Data Links Manager for Solaris 之後必須執行下列動作：

1. 將下列三行新增到 /etc/system 檔案：


```
set dlfsdrv:glob_mod_pri=0x100800
set dlfsdrv:glob_mesg_pri=0xff
set dlfsdrv:ConfigDlfsUid=UID
```

其中 UID 代表 id dlfm 的使用者 ID。
2. 重新開機，以使變更生效。

針對 AIX 上的 DB2 Data Links Manager 選擇備份方法

除了 Disk Copy 和 XBSA 之外，您也可以使用 Tivoli Storage Manager (TSM) 來備份位於 Data Links 伺服器裡的檔案。

若要將 Tivoli Storage Manager 作為保存伺服器：

1. 在 Data Links 伺服器上安裝 Tivoli Storage Manager。詳細資訊，請參考 Tivoli Storage Manager 的產品文件。
2. 以 Tivoli Storage Manager 伺服器來登錄 Data Links 伺服器從屬站應用程式。詳細資訊，請參考 Tivoli Storage Manager 的產品文件。
3. 將下列環境變數新增到 Data Links Manager 管理者的 db2profile 或 db2cshrc script 檔：

```
(針對 Bash、 Bourne 或 Korn shell)
export DSMI_DIR=/usr/tivoli/tsm/client/api/bin
export DSMI_CONFIG=$HOME/tsm/dsm.opt
export DSMI_LOG=$HOME/dldump
export PATH=$PATH:$DSMI_DIR
```

```
(針對 C shell)
setenv DSMI_DIR /usr/tivoli/tsm/client/api/bin
setenv DSMI_CONFIG ${HOME}/tsm/dsm.opt
setenv DSMI_LOG ${HOME}/dldump
setenv PATH=${PATH}:$DSMI_DIR
```

4. 請確定 dsm.sys TSM 系統選項檔位於 \$DSMI_DIR 目錄裡。
5. 請確定 dsm.opt TSM 使用者選項檔位於 *INSTHOME*/tsm 目錄裡，在此 *INSTHOME* 是 Data Links Manager 管理者的起始目錄。
6. 將 *PASSWORDACCESS* 選項設定為產生，在 */usr/tivoli/tsm/client/api/bin/dsm.sys* Tivoli Storage Manager 系統選項檔裡。
7. 使用產生選項在第一次啟動 Data Links 檔案管理程式之前，登錄 TSM 通行碼。使用這個方法，當 Data Links 檔案管理程式開始 TSM 伺服器的連線時，您不需要提供通行碼。詳細資訊，請參考 TSM 的產品文件。
8. 將 *DLFM_BACKUP_TARGET* 登錄變數設定到 TSM。在此例裡將忽略 *DLFM_BACKUP_DIR_NAME* 登錄變數的值。這樣會啟動 Tivoli Storage Manager 的備份選項。

註：

- a. 若您在執行時間裡，變更 TSM 和磁碟之間的 *DLFM_BACKUP_TARGET* 登錄變數設定，您應該察覺備份保存檔不會移到最近指定的保存位置。例如，若您以設定到 TSM 的 *DLFM_BACKUP_TARGET* 登錄值來啟動 Data Links 檔案管理程式，並將登錄值變更到磁碟位置，則所有最近保存的檔案將儲存在磁碟的新位置裡。先前保存到 TSM 的檔案將不會移到新的磁碟位置。
 - b. 若要置換預設的 TSM 管理類別，可以使用一個稱為 *DLFM_TSM_MGMTCLASS* 的新登錄變數。若此登錄變數維持在未設定的狀態，接著將會使用預設的 TSM 管理類別。
9. 輸入 **dlfm stop** 指令來停止 Data Links 檔案管理程式。
 10. 輸入 **dlfm start** 指令來啟動 Data Links 檔案管理程式。

針對 Solaris 作業環境上的 DB2 Data Links Manager 選擇備份方法

除了 Disk Copy 和 XBSA 之外，您也可以使用 Tivoli Storage Manager (TSM) 來備份位於 Data Links 伺服器裡的檔案。

若要將 Tivoli Storage Manager 作為保存伺服器：

1. 在 Data Links 伺服器上安裝 Tivoli Storage Manager。詳細資訊，請參考 Tivoli Storage Manager 的產品文件。
2. 以 Tivoli Storage Manager 伺服器來登錄 Data Links 伺服器從屬站應用程式。詳細資訊，請參考 Tivoli Storage Manager 的產品文件。
3. 將下列環境變數新增到 Data Links Manager 管理者的 db2profile 或 db2cshrc script 檔：

```
(針對 Bash、 Bourne 或 Korn shell)
export DSMI_DIR=/opt/tivoli/tsm/client/api/bin
export DSMI_CONFIG=$HOME/tsm/dsm.opt
export DSMI_LOG=$HOME/dldump
export PATH=$PATH:/opt/tivoli/tsm/client/api/bin
```

```
(針對 C shell)
setenv DSMI_DIR /opt/tivoli/tsm/client/api/bin
setenv DSMI_CONFIG ${HOME}/tsm/dsm.opt
setenv DSMI_LOG ${HOME}/dldump
setenv PATH=${PATH}:/opt/tivoli/tsm/client/api/bin
```

4. 請確定 dsm.sys TSM 系統選項檔位於 /opt/tivoli/tsm/client/api/bin 目錄裡。
5. 請確定 dsm.opt TSM 使用者選項檔位於 *INSTHOME*/tsm 目錄裡，在此 *INSTHOME* 是 Data Links Manager 管理者的起始目錄。
6. 將 *PASSWORDACCESS* 選項設定為產生，在 /opt/tivoli/tsm/client/api/bin/dsm.sys Tivoli Storage Manager 系統選項檔裡。
7. 使用產生選項在第一次啟動 Data Links 檔案管理程式之前，登錄 TSM 通行碼。使用這個方法，當 Data Links 檔案管理程式開始 TSM 伺服器的連線時，您不需要提供通行碼。詳細資訊，請參考 TSM 的產品文件。
8. 將 *DLFM_BACKUP_TARGET* 登錄變數設定到 TSM。在此例裡將忽略 *DLFM_BACKUP_DIR_NAME* 登錄變數的值。這樣會啟動 Tivoli Storage Manager 的備份選項。

註：

- a. 若您在執行時間裡，變更 TSM 和磁碟之間的 *DLFM_BACKUP_TARGET* 登錄變數設定，您應該察覺備份保存檔不會移到最近指定的保存位置。例如，若您以設定到 TSM 的 *DLFM_BACKUP_TARGET* 登錄值來啟動 Data Links 檔案管理程式，並將登錄值變更到磁碟位置，則所有最近保存的檔案將儲存在磁碟的新位置裡。先前保存到 TSM 的檔案將不會移到新的磁碟位置。
 - b. 若要置換預設的 TSM 管理類別，可以使用一個稱為 *DLFM_TSM_MGMTCLASS* 的新登錄變數。若此登錄變數維持在未設定的狀態，接著將會使用預設的 TSM 管理類別。
9. 輸入 **dlfm stop** 指令來停止 Data Links 檔案管理程式。
 10. 輸入 **dlfm start** 指令來啟動 Data Links 檔案管理程式。

針對 Windows NT 上的 DB2 Data Links Manager 選擇備份方法

每當 DATALINK 值插入具有 DATALINK 直欄的表格時 (定義為回復)，會安排 Data Links 伺服器的相對應 DATALINK 檔案，以備份到保存伺服器。目前來說，Disk Copy (預設方法) 和 Tivoli Storage Manager 是支援檔案備份到保存伺服器的兩種選項。未來的 DB2 Data Links Manager for Windows NT 版次將支援其它供應商的備份媒體和軟體。

Disk Copy (預設方法)

在 DB2 伺服器上輸入 **backup** 指令時，這個指令可確定資料庫裡的鏈結檔案會在 Data Links 伺服器上備份到 DLFM_BACKUP_DIR_NAME 環境變數所指定的目錄。這個變數的預設值是 c:\dlfmbackup，在此 c:\ 表示 Data Links Manager 的備份安裝磁碟機。

若要將此變數設定到 c:\dlfmbackup，請輸入下列指令：

```
db2set -g DLFM_BACKUP_DIR_NAME=c:\dlfmbackup
```

DLFM_BACKUP_DIR_NAME 環境變數指定的位置不能位於使用 Data Links 「檔案系統過濾器」的檔案系統上，而需要的空間可用於您指定給備份檔的目錄裡。

同時，請確定 DLFM_BACKUP_TARGET 變數經由輸入下列指令已設定為 LOCAL：

```
db2set -g DLFM_BACKUP_TARGET=LOCAL
```

設定或變更這些變數之後，請使用 **dlfm stop** 和 **dlfm start** 指令來停止並重新啟動 Data Links 檔案管理程式。

Tivoli Storage Manager

若要將 Tivoli Storage Manager 作為保存伺服器：

1. 在 Data Links 伺服器上安裝 Tivoli Storage Manager。詳細資訊，請參考 Tivoli Storage Manager 的產品文件。
2. 以 Tivoli Storage Manager 伺服器來登錄 Data Links 伺服器從屬站應用程式。詳細資訊，請參考 Tivoli Storage Manager 的產品文件。
3. 按一下**開始**，然後選取**設定 --> 控制台 --> 系統**。即會開啓「系統內容」視窗。選取**環境**標籤，並輸入下列環境變數和對應值：

變數	值
DSMI_DIR	c:\tsm\baclient
DSMI_CONFIG	c:\tsm\baclient\dsm.opt
DSMI_LOG	c:\tsm\dldump

4. 請確定 dsm.sys TSM 系統選項檔位於 c:\tsm\baclient 目錄裡。
5. 請確定 dsm.opt TSM 使用者選項檔位於 c:\tsm\baclient 目錄裡。
6. 將 **PASSWORDACCESS** 選項設定為產生，在 c:\tsm\baclient\dsm.sys Tivoli Storage Manager 系統選項檔裡。
7. 使用產生選項在第一次啟動 Data Links 檔案管理程式之前，登錄 TSM 通行碼。使用這個方法，當 Data Links 檔案管理程式開始 TSM 伺服器的連線時，您不需要提供通行碼。詳細資訊，請參考 TSM 的產品文件。

8. 使用下列指令將 `DLFM_BACKUP_TARGET` 環境變數設定到 TSM：

```
db2set -g DLFM_BACKUP_TARGET=TSM
```

在此例裡將忽略 `DLFM_BACKUP_DIR_NAME` 環境變數的值。這樣會啟動 Tivoli Storage Manager 的備份選項。

註：

- a. 若您在執行時間裡，變更 TSM 和 LOCAL 之間的 `DLFM_BACKUP_TARGET` 環境變數設定，您應該察覺備份保存檔不會移到最近指定的保存位置。例如，若您以設定到 TSM 的 `DLFM_BACKUP_TARGET` 環境變數來啟動 Data Links 檔案管理程式，並將其值變更到 LOCAL，則所有最近保存的檔案將儲存在磁碟的新位置裡。先前保存到 TSM 的檔案將不會移到新的磁碟位置。
 - b. 若要置換預設的 TSM 管理類別，可以使用一個稱為 `DLFM_TSM_MGMTCLASS` 的新環境變數。若此變數維持在未設定的狀態，接著將會使用預設的 TSM 管理類別。
9. 輸入 `dlfm stop` 指令來停止 Data Links 檔案管理程式。
 10. 輸入 `dlfm start` 指令來啟動 Data Links 檔案管理程式。

備份 AIX 上的記錄檔案系統

書中說明必須停止 Data Links Manager，而離線備份應該依據檔案系統來製作。下列方法除去停止 Data Links Manager 的需求，主要針對需要較高可用性的使用者。

1. 存取 CLI 來源檔 `quiesce.c` 及 Shell Script `online.sh`。這些檔案位於 `/samples/dlfm` 目錄。
2. 編譯 `quiesce.c`：

```
xlc -o quiesce -L$HOME/sql/lib/lib -I$HOME/sql/lib/include -c quiesce.c
```
3. 以 root 身分，在具有 DLFS 檔案系統的節點上執行 Script。

shell script `online.sh` 假設您在 Data Link Manager 節點上具有型錄登錄 (針對以 Data Link Manager 來登錄的每個資料庫)。也假設 `/etc/filesystems` 具有 DLFS 檔案系統的完整登錄。shell script 執行下列動作：

- 靜止資料庫裡的所有表格，這些資料庫以 Data Links Manager 來登錄。這樣會停止所有新的活動。
- 將檔案系統以唯讀檔案系統來解除裝載或重新裝載。
- 執行檔案系統備份。
- 將檔案系統以讀取/寫入檔案系統來解除裝載或重新裝載。
- 重設 DB2 表格；也就是，解除靜止狀態。

必須修改 Script 以適合您的環境，如下所示：

1. 選取備份指令，並放置於 script 的 `do_backup` 函數。
2. 設定 Script 內的下列環境變數：
 - `DLFM_INST`：設定這個到 DLFM 案例名稱。
 - `PATH_OF_EXEC`：設定這個到 "quiesce" 可執行檔所在的路徑。

以下列指令呼叫 script：

```
online.sh <filesystem_name>
```

在 Windows NT 上資料鏈結中的 Administrator 群組專用權

在 Windows NT 上，對於大部分的函數，一個隸屬於 Administrator 群組的使用者關於使用 DataLink 來鏈結的檔案和 UNIX 上的 root 使用者有相同的專用權。下面的表格將兩者做了比較。

作業	Unix (root)	Windows NT (Administrator)
更名	是	是
不需符記可存取檔案	是	是
刪除	是	否 (請看下面的說明)
更新	是	否 (請看下面的說明)

註: NTFS 不允許這些作業在唯讀檔案上執行。Administrator 使用者可以順利完成這些作業，方法為啓用檔案的寫入許可權。

將 Data Links File System Filter (DLFF) 安裝的記載減到最少

您可以變更 `d1fs_cfg` 檔案，將 Data Links File System Filter (DLFF) 安裝的記載減到最少。`d1fs_cfg` 檔案會傳到 `strload` 常式來載入驅動程式及架構參數。檔案的位置在 `/usr/lpp/db2_07_01/cfg/` 目錄。透過符號鏈結，也可在 `/etc` 目錄找到這個檔案。`d1fs_cfg` 檔案的格式如下：

```
d 'driver-name' 'vfs number' 'dlfm id' 'global message priority'  
  'global module priority' - 0 1
```

其中：

d **d** 參數指定要載入驅動程式。

driver-name

`driver-name` 是要載入的驅動程式的完整路徑。例如，DB2 版本 7 的完整路徑是 `/usr/lpp/db2_07_01/bin/d1fsdrv`。驅動程式名稱是 `d1fsdrv`。

vfs number

這是 DLFS 在 `/etc/vfs` 中的 `vfs` 項目。

dlfm id

這是 DataLinks Manager 管理者的使用者 ID。

global message priority

這是廣域訊息優先順序

global module priority

這是廣域模組優先順序

0 1 **0 1** 是用來建立此驅動程式非複製節點的較小數字。節點名稱是將較小數字附加到被複製的驅動程式節點名稱來建立的。不可給超過 5 個以上的較小數字 (0-4)。

現實中的範例可能如下：

```
d /usr/lpp/db2_07_01/bin/d1fsdrv 14,208,255,-1 - 0 1
```

會記載的訊息是依據 global message priority 及 global module priority 的設定值而定。若要將記載減到最少，您可以變更 global message priority 的值。

有 4 個訊息優先順序可以使用：

```
#define LOG_EMERGENCY    0x01
#define LOG_TRACING      0x02
#define LOG_ERROR        0x04
#define LOG_TROUBLESHOOT 0x08
```

大部份 DLFF 中的訊息使用 LOG_TROUBLESHOOT 作為訊息優先順序。這裡是一些替代架構範例：

如果您真的需要緊急(emergency)訊息及錯誤訊息，請將 dlfs_cfg 配置檔中的廣域訊息優先順序設定為 5 (1+4)：

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,5,-1 - 0 1
```

如果只需要錯誤訊息，請將廣域訊息優先順序設定為 4：

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,4,-1 - 0 1
```

如果您不需要 DLFS 記載，則將廣域訊息優先順序設定為 0：

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,0,-1 - 0 1
```

在安裝後記載訊息

如果您在安裝之後需要記載緊急、錯誤及疑難排解訊息，您必須修改 dlfs_cfg 檔案。dlfs_cfg 檔案的位置在 /usr/lpp/db2_07_01/cfg 目錄。廣域訊息優先順序必須設定為 255 (最大優先順序) 會 13 (8+4+1)。將優先順序設定為 13 (8+4+1) 會記載緊急、錯誤及疑難排解資訊。

在設定廣域訊息優先順序之後，解除裝載 DLFS 過濾檔案系統，並重新載入 dlfsdrv 驅動程式以在載入時使用新的優先順序值。在重新載入 dlfsdrv 驅動程式之後，必須重新裝載 DLFS 過濾檔案系統。

註：直到再度變更 dlfs_cfg 檔案，任何後續的 dlfsdrv 驅動程式載入會使用相同 dlfs_cfg 設定值。

使用 SMIT 解除安裝 DLFM 元件可能會除去額外的檔案集

在從安裝 Data Links Manager 的 AIX 機器上解除安裝 DB2 (版本 5、6 或 7) 之前，請遵循下列步驟：

1. 以 root 的身分，使用下列指令備份 /etc/vfs：

```
cp -p /etc/vfs /etc/vfs.bak
```
2. 解除安裝 DB2。
3. 以 root 的身分，用步驟 1 所做的備份取代 /etc/vfs：

```
cp -p /etc/vfs.bak /etc/vfs
```

開始之前/決定主電腦名稱

您必須決定每一台 DB2 伺服器及資料鏈結伺服器的名稱。您必須要知道這些主電腦名稱，才能驗證安裝作業。在連接到「DB2 Data Links 檔案管理程式」時，DB2 UDB 伺服器會內部傳送下列資訊到 DLFM：

- 資料庫名稱
- 案例名稱
- 主電腦名稱

然後 DLFM 會將此資訊與它的內部表格作比對，以決定是否容許連線成立。只有資料庫名稱、案例名稱和主電腦名稱的組合都已使用 `dlfm add_db` 指令註冊到 DLFM 時，連線才會成立。在 `dlfm add_db` 指令中使用的主電腦名稱必須和 DB2 UDB 伺服器所內部傳送的主電腦名稱完全相符。

使用以下列方式所得到的主電腦名稱：

1. 在 DB2 伺服器上輸入 `hostname` 指令。例如，此指令可能會傳回 `db2server`。
2. 根據您的平台，執行下列其中一項：

- 在 AIX 上，輸入 `host db2server` 指令，其中 `db2server` 是先前步驟中所得到的名稱。這個指令應該會傳回類似下列的輸出：

```
db2server.services.com is 9.11.302.341, Aliases: db2server
```

- 在 Windows NT 上，輸入 `nslookup db2server` 指令，其中 `db2server` 是先前步驟中所得到的名稱。這個指令應該會傳回類似下列的輸出：

```
伺服器：dnsserv.services.com  
位址：9.21.14.135  
名稱：db2server.services.com  
位址：9.21.51.178
```

- 在 Solaris 上，輸入 `cat /etc/hosts | grep 'hostname'`。如果主電腦名稱在 `/etc/hosts` 中指定時不含領域名稱，則這個指令應該會傳回類似下列的輸出：

```
9.112.98.167 db2server loghost
```

如果主電腦名稱指定時含領域名稱，則指令會傳回類似下列的輸出：

```
9.112.98.167 db2server.services.com loghost
```

在使用 `dlfm add_db` 指令註冊 DB2 UDB 資料庫時請用 `db2server.services.com` 作為主電腦名稱。如果在 `dlfm add_db` 指令中使用任何其它的別名，則 DB2 伺服器內部到 DLFM 的連線會失敗。

若要將 Data Links 伺服器註冊到 DB2 資料庫，可使用 DB2 `"add datalinks manager for database database_alias using node hostname port port_number"` 指令。

`hostname` 是 Data Links 伺服器的名稱。在此指令中可使用任何 Data Links 伺服器的有效別名。參照此 Data Links 伺服器的 DATALINK 值必須用 URL 值來指定 `hostname`；亦即在指派 URL 值給 DATALINK 直欄時，必須使用 `"add datalinks manager"` 指令中所使用的名稱。使用不同的別名將會造成 SQL 陳述式失敗。

使用 DLFM：從 DB2 資料庫捨棄 DB2 Data Links Manager 後的清除動作

當使用 DROP DATALINKS MANAGER 指令從資料庫將 DB2 Data Links Manager 捨棄時，指令本身不會清除關於 DB2 Data Links Manager 的相關資訊。使用者可以明確地起始解除鏈結任何鏈結到資料庫的檔案，以及備份資訊的垃圾收集。這可以使用 dlfm drop_dlm 指令來完成。這個指令會將特定資料庫的所有資訊非同步的刪除。DB2 Data Links Manager 必須在執行中，此指令才會成功。非常重要：此指令只在捨棄 DB2 Data Links Manager 之後才能使用；否則，將會失去關於 DB2 Data Links Manager 的重要資訊，且無法回復。

若要對特定資料庫起始解除鏈結處理程序及備份資訊的垃圾收集：

1. 以「Data Links 檔案管理程式」管理者身分登入系統。
2. 發出下列指令：

```
dlfm drop_dlm database instance hostname
```

其中：

database 是遠端 DB2 UDB 資料庫的名稱；
instance 是資料庫所在的案例；
hostname 是資料庫所在的 DB2 UDB 伺服器的主電腦名稱。

3. 登出。

若需完整的用法實務範例，顯示應該如何使用此指令的上下文，請參閱 *Command Reference*。

已為此指令建立了一個新的錯誤碼（請參閱『DLFM1001E (新的錯誤訊息)』）。

DLFM1001E (新的錯誤訊息)

DLFM1001E: 在 drop_dlm 處理程序時發生錯誤。

原因：

Data Links Manager 無法對指定的資料庫起始解除鏈結及垃圾收集處理程序。這可能因為下列任一原因而發生：

- Data Links Manager 不在執行中。
- 指令中指定的資料庫、案例及主電腦名稱的組合無效。
- 在 Data Links Manager 的任一元件服務失敗。

動作：

請執行下列步驟：

1. 確定 Data Links Manager 在執行中。啟動 Data Links Manager，如果它不在執行中。
2. 確定資料庫、案例及主電腦名稱的組合定義一個已註冊的資料庫。您可在 Data Links Manager 上使用 "dlfm list registered databases" 指令來達到目的。
3. 如果錯誤仍然發生，請參照 db2diag.log 檔案中的資訊，檢查是否有任何元件服務 (例如，Connection Management Service、Transaction Management Service 等等) 失敗。注意 db2diag.log 中的錯誤碼，並採取錯誤碼底下所建議的適當動作。

DLFM 設定架構檔選項

dlfm setup dlfm.cfg 選項已被除去。在文件中若有提及，請予以忽略。

在 AIX 上執行 Data Links/DFS Script dmapp_prestart 時發生錯誤

如果指令

```
/usr/sbin/cfgdmepl -a "/usr/lib/drivers/dmlfs.ext"
```

失敗，而回覆碼為 1 (在您執行 Data Links/DFS Script dmapp_prestart)，請安裝 DFS 3.1 ptfset1 來修正 cfgdmepl。

Tivoli Space Manager 與 Data Links 整合

DB2 Data Links Manager 現在已具有 Tivoli Space Manager 功能上的優點。Tivoli Space Manager Hierarchical Storage Manager (HSM) 從屬站程式會自動將合格檔案移轉至儲存體，以維護本端檔案系統上特定層次的可用空間。當存取移轉的檔案時，會自動重新呼叫這些檔案，而且容許使用者移轉及重新呼叫特定的檔案。

對於所使用檔案系統內含需要定期移至第三層儲存體 (其中的檔案系統空間需要在一般基礎上管理) 的大型檔案之客戶而言，這個新特性很有幫助。Tivoli Space Manager 目前提供一些方法來管理許多客戶的第三層儲存體。Tivoli Space Manager 的 DB2 Data Links Manager 新功能提供更大的彈性來管理 DATALINK 檔案空間。Tivoli Space Manager 容許定期調整 Data Links 管理的檔案系統配置，避免正常使用期間因疏乎而導致檔案系統填滿的風險，而不需要在所有檔案可能儲存於其中的 DB2 Data Links Manager 檔案系統中預先配置足夠的空間。

新增 Data Links 與 HSM 支援到檔案系統

以 Hierarchical Storage Management (HSM) 登記檔案系統時，請先以 HSM 登記再以 DataLinks File Manager 登記。

1. 登記 HSM 時，請使用指令 **"dsmmigfs add /fs"**。
2. 登記 DLM 時，請使用指令 **"dlfmfsmd/fs"**。

檔案系統的 Data Links 支援透過下列登錄，反映在 HSM 檔案系統的 /etc/filesystems 之段落中：

```
vfs = dlfs
mount = false
options = rw,Basefs=fsm
nodename = -
```

新增 Data Links 支援到現存的 HSM 檔案系統

登記 DLM 時，請使用指令 **"dlfmfsmd/fs"**。

新增 HSM 支援到現存的 Data Links 檔案系統

1. 登記 HSM 時，請使用指令 **"dsmmigfs add /fs"**。
2. 登記 DLM 時，請使用指令 **"dlfmfsmd/fs"**。

從 Data Links-HSM 檔案系統移除 Data Links 支援

移除 Data Links 支援時，請使用指令 **"dlfmfsmd -j /fs"**。

從 Data Links-HSM 檔案系統移除 HSM 支援

1. 移除 HSM 支援，請使用指令 **"dsmmigfs remove /fs"**。

2. 移除 Data Links 支援是 `"dlfmfsmd -j /fs"`。
3. 登記 DLM 時，請使用指令 `"dlfmfsmd/fs"`。

從 Data Links-HSM 檔案系統移除 Data Links 與 HSM 支援

1. 移除 HSM 支援，請使用指令 `"dsmmigfs remove /fs"`。
2. 移除 Data Links 支援時，請使用指令 `"dlfmfsmd -j/fs"`。

功能限制

本功能至目前僅支援在 AIX 上使用。

FC (Read permission DB) 鏈結檔的選擇性移轉 (dsmmigrate) 及取回只應由 root 使用者完成。

在 Read Permission DB 檔案為 DataLink Manager Administrator (dlfm) 的情況下，選擇性移轉只能由檔案擁有者執行。若要存取這類型檔案，則需要主電腦資料庫端的符記。唯一不需要符記的使用者為 root 使用者。root 使用者對 Read Permission DB 檔案執行選擇性移轉及取回比較容易。僅在第一次移轉時，dlfm 使用者才能使用有效的符記移轉 FC 檔案。若在取回之後嘗試第二次移轉，此作業會失敗，傳回錯誤訊息 "ANS1028S 內部程式錯誤。請聯絡客戶服務代表。" 非 root 使用者無法對 FC 檔案執行 **dsmmigrate**。一般而言，此限制對存取檔案伺服器上檔案的管理者影響不大。

stat 和 **statfs** 系統呼叫會將 Vfs 類型顯示為 **fsm**，而非顯示為 **dlfs**，雖然 **dlfs** 裝載在 **fsm** 之上。

上述行為代表 **dsmrecalld** 常駐程式的正常功能，其在檔案系統上執行 **statfs** 來檢查其 Vfs 類型是否為 **fsm**。

若檔案的最小 **inode** 號碼為 **FC (Read permission DB) 鏈結**的號碼，則指令 `"dsmis"` 不顯示任何輸出

dsmis 指令類似 **ls** 指令，其列示由 TSM 管理的檔案。不需要使用者動作

第 4 章 安裝和架構 DB2 Data Links Manager for AIX

一般安裝注意事項

從 **DB2 File Manager 版本 5.2** 移轉到 **DB2 Data Links Manager 版本 7**

步驟 3 中的資訊不正確。步驟 3 應如下所示：

"3. 以 DLFM 管理者身份，執行 `/usr/lpp/db2_07_01/adm/db2dlmmg` 指令。

安裝與架構補充資料

第 5 章 在 UNIX 作業系統上安裝 DB2 從屬站

HP-UX 核心程式架構參數

建議設定 HP-UX 核心程式參數錯誤指出 `msgmbn` 及 `msgmax` 應該設定為 65535 或更高。兩個參數都應該設定為 65535。

第 12 章 執行您自己的應用程式

使用 Run-Time Client 連結資料庫公用程式

Run-Time Client 不能用來連結資料庫公用程式 (匯入、匯出、重組、命令行處理器)，且 DB2 CLI 會在資料庫可以使用檔案之前將檔案連結到每一個資料庫。您必須改用 DB2 Administration Client 或 DB2 Application Development Client。

在資料庫可以使用這些資料庫公用程式和 DB2 CLI 連結檔案之前，必須將該資料庫和它們之間進行連接。網路環境下，如果您使用不同作業系統的多重從屬站，或是不同的 DB2 版本或服務程式等級，您必須馬上連結各個作業系統和 DB2 版本組合的公用程式。

UNIX 從屬站使用 ODBC 存取 DB2

在第 12 章 (「執行您自己的應用程式」) 中提到，以 ODBC 從屬站應用程式或 ODBC SDK 來安裝 ODBC 驅動程式管理程式，則需要更新 `odbcinst.ini`。這樣的說法不太正確。因為，當您安裝 Merant ODBC 驅動程式管理程式產品時，則不需要更新 `odbcinst.ini`。

第 24 章 設置聯合系統來存取多重資料來源

聯合系統

DB2 聯合系統是一種 DBMS 的特殊類型。聯合系統可讓您查詢並擷取位在其他 DBMS 上的資料，例如 Oracle、Sybase 及 Microsoft SQL Server。SQL 陳述式可以參照多重 DBMS 或單一陳述式中的個別資料庫。例如，您可以結合位在 DB2 Universal Database 表格、Oracle 表格及 Sybase 概略表中的資料。支援的 DBMS 包括 Oracle、Sybase、Microsoft SQL Server (for Windows NT) 及 DB2 Universal Database 系列的成員 (如 DB2 for OS/390、DB2 for AS/4000 及 DB2 for Windows)。

DB2 聯合系統包含具有 DB2 案例的伺服器 (資料庫將以聯合資料庫伺服器) 以及一個或多個資料來源。聯合資料庫包含識別資料來源及其性質的目錄登錄。資料來源包含 DBMS 及資料。DB2 Universal Database 有通信協定，稱為外層，您可以使用來存取這些資料來源。外層是一種機制，聯合伺服器用此機制來通信及擷取資料來源的資料。暱名是用來參照位在資料來源中的表格與概略表。應用程式連接到聯合資料庫，就像任何其他 DB2 資料庫。您使用的外層是以 DB2 Universal Database 正在執行的平台為依據。

在聯合系統設定之後，可以存取資料來源中的資訊，即使在一個大型資料庫中。使用者及應用程式傳送查詢至一個聯合資料庫，它從資料來源擷取資料。

DB2 聯合系統在部份限制下操作。分散式要求限制在 DB2 版本 7 中進行唯讀作業。此外，您不能執行與暱名相反的公用程式作業（像 LOAD、REORG、REORGCHK、IMPORT、RUNSTATS 等等）。但是，您可以使用透過機能，使用與該資料來源相關的 SQL 用語，將 DDL 及 DML 陳述式直接提出至 DBMS。

限制

版本 7.2 的新外層 (如 Linux 及 Solaris 上的 Oracle、AIX 及 Solaris 上的 Sybase 及 NT 及 AIX 上的 Microsoft SQL Server) 不可用於 FixPak 3；您必須購買 DB2 Relational Connect 版本 7.2。

安裝 DB2 Relational Connect

本節提供在伺服器上安裝 DB2 Relational Connect 的指令，您將以該伺服器作為聯合系統伺服器。

在 Windows NT 伺服器上安裝 DB2 Relational Connect

在您將 DB2 Relational Connect 安裝在您的 Windows NT 聯合伺服器之前：

- 請確定聯合伺服器上有安裝 DB2 Universal Database Enterprise Edition 或 DB2 Universal Database Enterprise -- Extended Edition。若您打算將 DB2 系列資料庫併入您的分散式要求中，則當您安裝 DB2 Universal Database 時，您必須選取 **Distributed Join for DB2** 資料來源選項。若要驗證已實行此選項，請檢查 FEDERATED 參數是設定為 YES。您可以發出 **GET DATABASE MANAGER CONFIGURATION** 指令檢查此設定，它會顯示所有參數及其現行設定值。

- 請確定您已在聯合伺服器上安裝資料來源的從屬站軟體 (如 Sybase Open Client)。

- 以您為了執行安裝所建立的使用者帳戶登入系統。
- 關閉任何執行中的程式，安裝程式才能依照要求來更新檔案。
- 呼叫安裝程式。您可以自動方式或以人工方式來呼叫安裝程式。若安裝程式無法自動啟動，或您想要以不同的語言執行安裝，請以人工方式呼叫安裝程式。

- 若要自動呼叫安裝程式，請將 DB2 Relational Connect CD 插入磁碟機中。自動執行特性會自動啟動安裝程式。決定系統語言，並啟動該語言的安裝程式。

- 自行呼叫安裝程式：

- 按一下**開始**，選取**執行**選項。
- 在**開啓欄位**中，輸入下列指令：

```
x:\setup /i language
```

其中：

x: 代表您的 CD-ROM 光碟機。

language

代表您語言的國碼 (如 EN 是英文)。

- 按一下**確定**。

即會開啓安裝發射台。

- 按一下**安裝**，開始安裝程序。

5. 遵循安裝程式中的提示。

當程式完成時，DB2 Relational Connect 與其它 DB2 產品將會安裝在您的安裝目錄中。

在 AIX、Linux 及 Solaris 作業環境伺服器上安裝 DB2 Relational Connect

在您將 DB2 Relational Connect 安裝在您的 AIX、Linux 及 Solaris 作業環境聯合伺服器之前：

- 請確定聯合伺服器上有安裝 DB2 Universal Database Enterprise Edition 或 DB2 Universal Database -- Extended Edition。若您打算將 DB2 系列資料庫併入您的分散式要求中，則當您安裝 DB2 Universal Database 時，您必須選取 **Distributed Join for DB2** 資料來源選項。若要驗證已實行此選項，請檢查 FEDERATED 參數是設定為 YES。您可以發出 **GET DATABASE MANAGER CONFIGURATION** 指令檢查此設定，它會顯示所有參數及其現行設定值。
- 請確定您已在聯合伺服器上安裝資料來源的從屬站軟體 (如 Sybase Open Client)。

若要将 DB2 Relational Connect 安裝在您的 AIX、Linux 及 Solaris 作業環境伺服器，請使用 db2setup 公用程式：

1. 以根權限使用者登入。
2. 插入並裝載您的 DB2 產品 CD-ROM。有關裝載 CD-ROM 的資訊，請參閱 AIX 快速入門。
3. 輸入 **cd lcdrom** 指令 (其中 *cdrom* 是您的產品 CD-ROM 的裝載點)，以變更裝載 CD-ROM 的目錄。
4. 輸入 **./db2setup** 指令。在幾分鐘後，即會開啓「DB2 安裝公用程式」視窗。
5. 選取**安裝**。即會開啓「安裝 DB2 V7」視窗。
6. 導引至您從屬站的 DB2 Relational Connect 產品，例如，Relational Connect for Sybase，並按空格鍵以選取。當您選取選項時，其旁邊即會出現星號。
7. 選取**確定**。即會開啓「建立 DB2 服務」視窗。
8. 您可以選擇建立 DB2 案例。選取**確定**。即會開啓「摘要報告安裝」視窗。會自動安裝兩個項目：Oracle 的分散式結合及 DB2 Relational Connect 的「產品標記」。「產品標記」是必要的，可讓您連接到 Sybase 資料來源。
9. 選擇**繼續執行**。即會出現一視窗，指出這是您停止 Relational Connect 的最後機會。選擇**確定**以繼續執行安裝。可能要花幾分鐘的時間才能完成安裝。
10. 當通知出現時，指出安裝已順利完成，請選取**確定**。「摘要報告」出現時，指出哪一個安裝選項的成功或失敗；請再次選取**確定**。

當安裝完成時，DB2 Relational Connect 與其它 DB2 產品將安裝在目錄中。在 AIX 上，此為 /usr/lpp/db2_07_01 目錄。在 Solaris 上，此為 /opt/IBMdb2/V7.1 目錄。在 Linux 上，此為 /usr/IBMdb2/V7.1 目錄。

第 26 章 存取 Oracle 資料來源

除了支援 AIX 及 Windows NT 上的外層，DB2 Universal Database 也支援 Linux 及 Solaris 作業環境上的 Oracle 外層。此支援限於 Oracle 版本 8。若要存取這些平台的外層，您需要插入 V7.2 DB2 Relational Connect CD，並選取 DB2 Relational Connect for Oracle 資料來源。

一旦您安裝了 DB2 Relational Connect，您可以將 Oracle 資料來源新增到聯合伺服器中：

1. 在 DB2 聯合伺服器上安裝及架構 Oracle 從屬站軟體。
2. 修改 db2dj.ini 檔案並發出 **db2set** 指令，以設定資料來源環境變數。
3. 對於 UNIX 平台上執行的 DB2 聯合伺服器，請執行 djxlink Script，將 Oracle SQL*Net 或 Net8 檔案庫鏈結編輯到您的 DB2 聯合伺服器。
4. 確定已更新 SQL*Net 或 Net8 tnsnames.ora 檔案。
5. 重新使用 DB2 案例。
6. 建立外層。
7. 選用的：設定 DB2_DJ_COMM 環境變數。
8. 建立伺服器。
9. 建立使用者對映。
10. 建立表格與概略表的暱名。

有關這些步驟的指示，包括設定環境變數，可見於「第 26 章」。設置聯合系統來存取 DB2 安裝與架構補充資料中的 Oracle 資料來源。

文件錯誤

在『新增 Oracle 資料來源至聯合系統』段落有下列錯誤：

- 程序中遺漏了一個步驟。正確的步驟為：

1. 參考 Oracle 提供的文件，在 DB2 聯合伺服器上安裝及架構 Oracle 從屬站軟體。
- 2.

修改 db2dj.ini 檔案並發出 **db2set** 指令，以設定資料來源環境變數。 **db2set** 指令會用您的設定值更新 DB2 設定檔登錄。

有關設定環境變數的詳細指示可見於「第 26 章」。設置離合系統來存取 DB2 安裝與架構補充資料中的 Oracle 資料來源。

3. 對於 UNIX 平台上執行的 DB2 聯合伺服器，請執行 djxlink Script，將 Oracle SQL*Net 或 Net8 檔案庫鏈結編輯到您的 DB2 聯合伺服器。根據您的平台，djxlink Script 的位置在：

```
/usr/lpp/db2_07_01/bin on AIX
```

```
/opt/IBMdb2/V7.1/bin Solaris
```

```
/usr/IBMdb2/V7.1/bin Linux
```

只有在 DB2 聯合伺服器上安裝 Oracle 的從屬站軟體之後，才執行此 Script。

- 文件指出要設定：

```
DB2_DJ_INI = sqllib/cfg/db2dj.ini
```

這是不正確的，應該設定如下：

```
DB2_DJ_INI = $INSTHOME/sqllib/cfg/db2dj.ini
```

存取 Sybase 資料來源 (新增章節)

在您將 Sybase 資料來源新增至聯合伺服器之前，您必須將 Sybase Open Client 軟體安裝並架構在 DB2 聯合伺服器上。請參閱文件中的安裝程序，其隨附在 Sybase 資料庫軟體中，以取得如何安裝 Open Client 軟體的特定明細。因為是安裝的一部份，請確定包括了型錄儲存程序及 Sybase Open Client 檔案庫。

若要設定聯合伺服器，以存取儲存在 Sybase 資料來源的資料，您必須：

1. 安裝 DB2 Relational Connect 版本 7.2。請參閱第34頁的『安裝 DB2 Relational Connect』。
2. 新增 Sybase 資料來源至您的聯合伺服器中。
3. 指定 Sybase 字碼頁。

本章討論步驟 2 及 3。

本章的指示可引用在 Windows NT、AIX 及 Solaris 作業環境。特定平台的不同處備註於其發生的地方。

新增 Sybase 資料來源至聯合伺服器

若要新增 Sybase 資料來源至聯合伺服器，您必須：

1. 設定環境變數並更新設定檔登錄。
2. 將 DB2 鏈結至 Sybase 從屬站軟體 (僅 AIX 與 Solaris)
3. 重新使用 DB2 案例。
4. 建立並設定介面檔案。
5. 建立外層。
6. 選用的：設定 DB2_DJ_COMM 環境變數。
7. 建立伺服器。
8. 選用的：設定 CONNECTSTRING 伺服器選項。
9. 建立使用者對映。
10. 建立表格與概略表的暱名。

本節將對這些步驟作詳細的解釋。

步驟 1：設定環境變數並更新設定檔登錄

修改 db2dj.ini 檔案並發出 **db2set** 指令，以設定資料來源環境變數。db2dj.ini 檔案包含了安裝在聯合伺服器上的 Sybase 從屬站軟體之架構資訊。**db2set** 指令會用您的設定值更新 DB2 設定檔登錄。

在分割的資料庫系統中，您可以使用特殊案例中的所有節點之單一 db2dj.ini 檔案，或使用特殊案例中的一或多個節點之唯一的 db2dj.ini 檔案。未分割的資料庫系統中，每一案例僅可有一個 db2dj.ini 檔案。

若要設定環境變數：

1. 編輯位在 sqllib/cfg 的檔案 db2dj.ini，並設定下列環境變數：

```
SYBASE="<sybase home directory>"
```

其中 *<sybase home directory>* 是 Sybase 從屬站安裝的目錄。

2. 以 Sybase 環境變數來更新 DB2 案例的 .profile 檔案。您可以發出下列指令來執行此動作：

```
export PATH=『$SYBASE/bin:$PATH』  
export SYBASE="<sybase home directory>"
```

其中 *<sybase home directory>* 是 Sybase 從屬站安裝的目錄。

3. 要執行 DB2 案例 .profile，請輸入：

. .profile

4. 發出 **db2set** 指令，用您的變更來更新 DB2 設定檔登錄。此指令的語法 **db2set**，是以您的資料庫系統結構為依據。若您正在下列任何資料庫系統結構中使用 **db2dj.ini** 檔案，才需要此步驟：

若您正在未分割資料庫系統中使用 **db2dj.ini** 檔案，或您要 **db2dj.ini** 檔案僅引用於目前的節點，請發出：

```
db2set DB2_DJ_INI = sqllib/cfg/db2dj.ini
```

若您正在分割資料庫系統中使用 **db2dj.ini** 檔案，或您要 **db2dj.ini** 檔案中的值引用於此案例的所有節點，請發出：

```
db2set -g DB2_DJ_INI = sqllib/cfg/db2dj.ini
```

若您正在分割資料庫系統中使用 **db2dj.ini** 檔案，或您要 **db2dj.ini** 檔案中的值引用於特定的節點，請發出：

```
db2set -i INSTANCEX 3 DB2_DJ_INI = sqllib/cfg/node3.ini
```

其中：

INSTANCEX

代表案例名稱。

3 代表列示在 **db2nodes.cfg** 檔案的節點號碼。

node3.ini

代表 **db2dj.ini** 檔案的修改及更名版本。

步驟 2：將 DB2 鏈結至 Sybase 從屬站軟體 (僅 AIX 與 Solaris)

若要啓用存取 Sybase 資料來源，DB2 聯合伺服器必須鏈結編輯至從屬站檔案庫。鏈結編輯程序以聯合伺服器將通信的資料來源來建立每一個資料來源的外層。當您執行 **djxlink Script** 時，建立外層檔案庫。若要發出 **djxlink Script** 類型：

```
djxlink
```

步驟 3：重新使用 DB2 案例

若要確定環境變數設定於程式中，請重新使用 DB2 案例。當您重新使用案例時，復新 DB2 案例以接受您所作的變更。

發出下列指令以重新使用 DB2 案例：

在 **DB2 for Windows NT** 伺服器上：

```
NET STOP instance_name  
NET START instance_name
```

在 **DB2 for AIX 及 Solaris** 伺服器上：

```
db2stop  
db2start
```

步驟 4：建立並設定介面檔案

若要建立並設定介面檔案，您必須建立檔案並使檔案可存取。

1. 使用 Sybase 提供的公用程式來建立介面檔案，其包括想存取的所有 Sybase Open Server 的資料。請參閱 Sybase 的安裝文件，以取得有關使用此公用程式的詳細資訊。

Windows NT 通常將此檔案命名為 `sql.ini`。將您剛建立的檔案，從 `sql.ini` 重新命名為 `interfaces`，為廣用於所有平台的檔案命名。若您選擇不要將 `sql.ini` 重新命名為 `interfaces`，您必須使用 `IFILE` 參數或 `CONNECTSTRING` 選項，其在步驟 8 中有說明。

在 AIX 及 Solaris 系統上，此檔案是命名為 `<instance home>/sqllib/interfaces`。

2. 讓介面檔案可存取 DB2。

在 **DB2 for Windows NT 伺服器上**：

將檔案放置在 DB2 案例的 `%DB2PATH%` 目錄。

在 **DB2 for AIX 及 Solaris 伺服器上**：

將檔案放置在 DB2 案例的 `$HOME/sqllib` 目錄。使用 `ln` 指令，鏈結到從 DB2 案例的 `$HOME/sqllib` 目錄之檔案。例如：

```
ln -s -f /home/sybase/interfaces /home/db2djinst1/sqllib
```

步驟 5：建立外層

使用 `CREATE WRAPPER` 陳述式來指定外層，其將用來存取 Sybase 資料來源。外層是一種機制，聯合伺服器用此機制來通信及擷取資料來源的資料。DB2 包括兩個 Sybase 外層：`CTLIB` 及 `DBLIB`。下列範例顯示 `CREATE WRAPPER` 陳述式：

```
CREATE WRAPPER CTLIB
```

其中 `CTLIB` 是 Sybase Open Client 軟體使用的預設外層名稱。`CTLIB` 外層可使用於 Windows NT、AIX 及 Solaris 伺服器。

您可以用您選擇的名稱來取代預設外層名稱。但是，若您這麼做，您也必須包含 `CREATE WRAPPER` 陳述式的聯合伺服器之 `LIBRARY` 參數及外層檔案庫的名稱。請參閱 *DB2 SQL Reference* 中的 `CREATE WRAPPER` 陳述式，以取得有關外層常式庫名稱的詳細資訊。

步驟 6：選用的：設定 `DB2_DJ_COMM` 環境變數

若要在存取 Sybase 資料來源時增進效能，請設定 `DB2_DJ_COMM` 環境變數。此變數會判斷是否要在聯合伺服器起始設定時，載入外層。設定 `DB2_DJ_COMM` 環境變數以包括外層檔案庫，與您在前一步驟指定的外層相符；例如：

在 **DB2 for Windows NT 伺服器上**：

```
db2set DB2_DJ_COMM='ctlib.dll'
```

在 **DB2 for AIX 伺服器上**：

```
db2set DB2_DJ_COMM='libctlib.a'
```

在 **DB2 for Solaris 伺服器上**：

```
db2set DB2_DJ_COMM='libctlib.so'
```

請確定等號 (=) 的兩邊都沒有空格。

請參閱 *DB2 SQL Reference*，以取得有關外層常式庫名稱的詳細資訊。請參閱 *Administration Guide*，以取得有關 `DB2_DJ_COMM` 環境變數的資訊。

步驟 7：建立伺服器

使用 `CREATE SERVER` 陳述式來定義每一個想要從其中存取資料來源的 Sybase 伺服器，例如：

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB
OPTIONS (NODE 'sybnode', DBNAME 'sybdb')
```

其中：

SYBSERVER

代表您指定給 Sybase 伺服器的名稱。此名稱必須是唯一的。

SYBASE

代表您要架構存取的資料來源類型。Sybase 是唯一有支援的資料來源。

12.0

代表您要存取的 Sybase 版本。支援的版本有 10.0、11.0、11.1、11.5、11.9 及 12.0。

CTLIB

代表您在 CREATE WRAPPER 陳述式中指定的外層名稱。

'sybnode'

代表 SYBSERVER 常駐的節點名稱。在介面檔案中取得節點值。此值有區分大小寫。

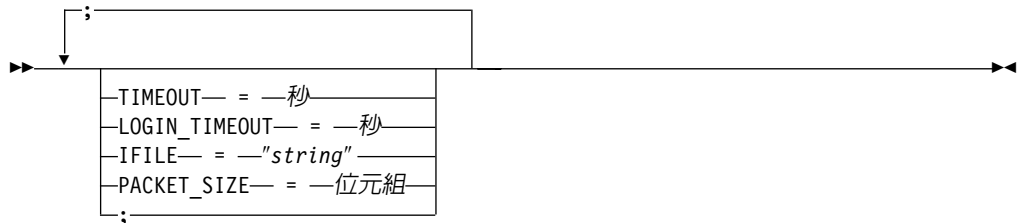
雖然節點名稱是指定為選項，但此為 Sybase 資料來源所需的。請參閱 *DB2 SQL Reference*，以取得其他選項的資訊。

'sybdb' 代表您想要存取的 Sybase 資料庫名稱。

選用的：步驟 8：設定 CONNECTSTRING 伺服器選項

指定逾時臨界值、介面檔案的路徑與名稱及介面檔案的分封大小。 Sybase Open Client 使用逾時臨界值來岔斷執行時間太長的查詢與回應。您可以使用 CREATE SERVER OPTION DDL 陳述式的 CONNECTSTRING 選項，設定 DB2 的臨界值。使用 CONNECTSTRING 來指定：

- SQL 查詢的逾時期間。
- 登入回應的逾時期間。
- 名稱介面的路徑名稱。
- 分封大小。



TIMEOUT

指定 DB2 Universal Database 等待任何 SQL 陳述式的 Sybase Open Client 之回應的秒數。seconds 值在 DB2 Universal Database 整數範圍中，是一個正整數。您指定的逾時值是依據您使用的外層。Windows NT、AIX 及 Solaris 伺服器都可以利用 DBLIB 外層。DBLIB 外層的預設值是 0。在 Windows NT、AIX 及 Solaris 伺服器上，DBLIB 的預設值會使 DB2 Universal Database 無限期地等待回應。

LOGIN_TIMEOUT

指定 DB2 Universal Database 等待 Sybase Open Client 之回應登入要求的秒數。預設值與 TIMEOUT 相同。

IFILE

指定 Sybase Open Client 介面檔案的路徑名稱。在 *string* 中識別的路徑必須以雙引號 (") 含括。在 Windows NT 伺服器上，預設值是 %DB2PATH%。在 AIX 及 Solaris 伺服器上，預設值是 sqllib/interfaces，在您的 DB2 Universal Database 案例的起始目錄中。

PACKET_SIZE

指定介面檔案的分封大小 (位元組)。若資料來源不支援指定的分封大小，連線將會失敗。當每一個記錄都很大時 (例如，將列數插入大型表格中)，增加分封大小可明顯地增加效能。位元組大小是一個數值。請參閱 Sybase 參考手冊以取得詳細資訊。

範例：

在 Windows NT 伺服器上，使用下列指令，將逾時值設定為 60 秒，並將介面檔案設定至 C:\etc\interfaces 中：

```
CREATE SERVER OPTION connectstring FOR SERVER sybase1  
SETTING 'TIMEOUT=60;LOGIN_TIMEOUT=5;IFILE="C:\etc\interfaces";'
```

在 AIX 與 Solaris 伺服器上，使用下列指令，將逾時值設定為 60 秒，並將介面檔案設定至 /etc/interfaces 中：

```
CREATE SERVER OPTION connectstring FOR SERVER sybase1  
SETTING 'TIMEOUT=60;PACKET_SIZE=4096;IFILE="/etc/interfaces";'
```

步驟 9：建立使用者對映

若聯合伺服器中的使用者 ID 與通行碼，與 Sybase 中的使用者 ID 與通行碼不同，請使用 CREATE USER MAPPING 陳述式，使本端使用者 ID 與 Sybase 資料來源中定義的使用者 ID 與通行碼相互對映；例如：

```
CREATE USER MAPPING FOR DB2USER SERVER SYBSERVER  
OPTIONS ( REMOTE_AUTHID 'sybuser', REMOTE_PASSWORD 'dayl1te')
```

其中：

DB2USER

代表您用來對映使用者 ID (定義於 Sybase 資料來源中) 的本端使用者 ID。

SYBSERVER

代表您在 CREATE SERVER 陳述式中定義的 Sybase 資料來源名稱。

'sybuser'

代表您要對映 DB2USER 的 Sybase 資料來源的使用者 ID。此值有區分大小寫。

'dayl1te'

代表與 'sybuser' 相關的通行碼。此值有區分大小寫。

請參閱 *DB2 SQL Reference*，以取得其他選項的詳細資訊。

步驟 10：建立表格與概略表的暱名

指定在您的 Sybase 資料來源中的每一個概略表或表格的暱名。當您查詢 Sybase 資料來源時，將使用這些暱名。Sybase 暱名有區分大小寫。將綱目及表格名稱以雙引號 (") 含括。下列範例顯示 CREATE NICKNAME 陳述式：

```
CREATE NICKNAME SYBSALES FOR SYBSERVER."salesdata"."europe"
```

其中：

SYBSALES

代表 Sybase 表格或概略表的專用暱名。

SYBSERVER."salesdata"."europe"

代表遵循下列格式的三部分識別字：

data_source_name."remote_schema_name"."remote_table_name"

針對您想要建立暱名的每一個表格或概略表，重複此步驟。當您建立暱名時，DB2 將使用連線來查詢資料來源型錄。此查詢會測試連接到資料來源的狀態。若連線無法運作，您會接收到一個錯誤訊息。

請參閱 *DB2 SQL Reference*，以取得有關 `CREATE NICKNAME` 陳述式的詳細資訊。有關一般暱名及驗證資料類型對映的詳細資訊，請參閱 *DB2 Administration Guide*。

指定 Sybase 字碼頁

此步驟僅在 DB2 聯合同服器及 Sybase 伺服器執行不同的字碼頁時才需要。使用與 DB2 相同的字碼集之資料來源不需要轉換。下列表格提供一般「國家語言支援 (NLS)」字碼頁的同等 Sybase 選項。您的 Sybase 資料來源架構必須與其語意相符，或者從屬站程式碼必須能夠偵測不符的語意，並將其標示為錯誤、或使用其本身的語意來對映資料。若從原始字碼頁到目標字碼頁，都找不到轉換表，則 DB2 會發出錯誤訊息。請參閱您的 Sybase 文件以取得詳細資訊。

表 1. 「Sybase 字碼頁」選項

字碼頁	同等 Sybase 選項
850	cp850
897	sjis
819	iso_1
912	iso_2
1089	iso_6
813	iso_7
916	iso_8
920	iso_9

使用 ODBC 存取 Microsoft SQL Server 資料來源 (新增章節)

在您將 Microsoft SQL Server 資料來源新增至 DB2 聯合同服器之前，您必須將 ODBC 驅動程式安裝並架構在聯合同服器上。請參閱文件中的安裝程序，其隨附在 ODBC 驅動程式，以取得如何安裝 ODBC 驅動程式的特定明細。

若要設定聯合同服器以存取儲存在 Microsoft SQL Server 資料來源的資料，您必須：

1. 在聯合同服器上安裝並架構 ODBC 驅動程式。請參閱文件中的安裝程序，其隨附在 ODBC 驅動程式，以取得如何安裝 ODBC 驅動程式的特定明細。

在 **DB2 for Windows NT 伺服器**上：

使用 ODBC 裝置管理程式架構系統 DSN。

在 DB2 for AIX 伺服器上：

安裝 MERANT 提供的執行緒版本，指定 MERANT 檔案庫目錄為 LIBPATH 中的第一個登錄，並設定 .odbc.ini 檔案。在起始目錄建立 .odbc.ini 起始目錄。

2. 安裝 DB2 Relational Connect 版本 7.2。請參閱第34頁的『安裝 DB2 Relational Connect』。
3. 新增 Microsoft SQL Server 資料來源至您的聯合同伺服器中。
4. 指定 Microsoft SQL Server 字碼頁。

本章討論步驟 3 及 4。

本章的指示可引用在 Windows NT 及 AIX 平台。特定平台的不同處備註於其發生的地方。

新增 Microsoft SQL Server 資料來源至聯合同伺服器

在您安裝 ODBC 驅動程式及 DB2 Relational Connect 之後，使用下列步驟，將 Microsoft SQL Server 資料來源新增到您的中：

1. 設定環境變數 (僅 AIX)。
2. 執行 Shell Script (僅 AIX)。
3. 選用的：設定 DB2_DJ_COMM 環境變數。
4. 重新使用 DB2 案例 (僅 AIX)。
5. 建立外層。
6. 建立伺服器。
7. 建立使用者對映。
8. 建立表格與概略表的暱名。
9. 選用的：獲得 ODBC 追蹤。

下列各節將對這些步驟作詳細的解釋。

步驟 1：設定環境變數 (僅 AIX)

修改 db2dj.ini 檔案並發出 **db2set** 指令，以設定資料來源環境變數。db2dj.ini 檔案包含資訊目的，以連接 Microsoft SQL Server 資料來源。**db2set** 指令會用您的設定值更新 DB2 設定檔登錄。

在分割的資料庫系統中，您可以使用特殊案例中的所有節點之單一 db2dj.ini 檔案，或使用特殊案例中的一或多個節點之唯一的 db2dj.ini 檔案。未分割的資料庫系統中，每一案例僅可有一個 db2dj.ini 檔案。

若要設定環境變數：

1. 編輯位在 sqllib/cf\$HOME/sqllib/cfg/g 的檔案 db2dj.ini，並設定下列環境變數：

```
ODBCINI=$HOME/.odbc.ini
DJX_ODBC_LIBRARY_PATH=<path to the Merant driver>/lib
LIBPATH=<path to the Merant driver>/lib
DB2ENVLIST=LIBPATH
```

發出 **db2set** 指令，用您的變更來更新 DB2 設定檔登錄。**db2set** 的語法，是以您的資料庫系統結構為依據。

- 若您正在未分割資料庫系統中使用 `db2dj.ini` 檔案，或若您在分割的資料庫系統中使用 `db2dj.ini` 檔案，且您要 `db2dj.ini` 檔案中的值僅引用於目前的節點，請發出下列指令：

```
db2set DB2_DJ_INI=<path to ini file>/db2dj.ini
```

- 若您正在分割資料庫系統中使用 `db2dj.ini` 檔案，或您要 `db2dj.ini` 檔案中的值引用於此案例的所有節點，請發出下列指令：

```
db2set -g DB2_DJ_INI=<path to ini file>/db2dj.ini
```

- 若您正在分割資料庫系統中使用 `db2dj.ini` 檔案，或您要 `db2dj.ini` 檔案中的值引用於特定的節點，請發出下列指令：

```
db2set -i INSTANCEX 3 DB2_DJ_INI=$HOME/sql1lib/cfg/node3.ini
```

其中：

INSTANCEX

代表案例名稱。

3 代表列示在 `db2nodes.cfg` 檔案的節點號碼。

node3.ini

代表 `db2dj.ini` 檔案的修改及更名版本。

步驟 2：執行 Shell Script (僅 AIX)

`djxlink.sh` Shell Script 鏈結從屬站檔案庫至外層檔案庫。若要執行 Shell Script：

```
djxlink
```

步驟 3：選用的：設定 DB2_DJ_COMM 環境變數

若您發現其花費過多的時間來存取 Microsoft SQL Server 資料來源，您可以在聯合伺服器起始設定，而非在您嘗試存取資料來源時，設定 `DB2_DJ_COMM` 環境變數來下載外層，如此便可增進效能。設定 `DB2_DJ_COMM` 環境變數以包括外層檔案庫，與您在第 5 步驟指定的外層相符。例如：

在 **DB2 for Windows NT** 伺服器上：

```
db2set DB2_DJ_COMM=djxmssql3.dll
```

在 **DB2 for AIX** 伺服器上：

```
db2set DB2_DJ_COMM=libmssql3.a
```

請確定等號 (=) 的兩邊都沒有空格。

請參閱 *DB2 SQL Reference*，以取得有關外層常式庫名稱的詳細資訊。

步驟 4：重新使用 DB2 案例 (僅 AIX)

若要確定環境變數設定於程式中，請重新使用 DB2 案例。當您重新使用案例時，復新 DB2 案例以接受您所作的變更。發出下列指令，重新使用 DB2 案例：

```
db2stop
db2start
```

步驟 5：建立外層

DB2 Universal Database 有兩種通信協定，稱為外層，您可以使用來存取 Microsoft SQL Server 資料來源。外層是一種機制，聯合伺服器用此機制來通信及擷取資料來源的資料。您使用的外層是以 DB2 Universal Database 正在執行的平台為依據。使用第45頁的表2 作為指南，以選取適當的外層。

表 2. ODBC 驅動程式

ODBC 驅動程式	平台	外層名稱
ODBC 3.0 (或更新版本) 驅動程式	Windows NT	DJXMSSQL3
MERANT DataDirect Connect ODBC 3.6 驅動程式	AIX	MSSQLODBC3

使用 CREATE WRAPPER 陳述式來指定外層，其將用來存取 Microsoft SQL Server 資料來源。下列範例顯示 CREATE WRAPPER 陳述式：

```
CREATE WRAPPER DJXMSSQL3
```

其中 DJXMSSQL3 是 DB2 for Windows NT 伺服器 (使用 ODBC 3.0 驅動程式) 上使用的預設外層名稱。若您有 DB2 for AIX 伺服器，您可以指定 MSSQLODBC3 外層名稱。

您可以用您選擇的名稱來取代預設外層名稱。但是，若您這麼做，您必須包含 CREATE WRAPPER 陳述式的聯合伺服器平台之 LIBRARY 參數及外層檔案庫的名稱。例如：

在 **DB2 for Windows NT 伺服器**上：

```
CREATE WRAPPER wrapper_name LIBRARY 'djmssql3.dll'
```

其中 *wrapper_name* 是您想要提供給外層的名稱，而 'djmssql3.dll' 是常式庫名稱。

在 **DB2 for AIX 伺服器**上：

```
CREATE WRAPPER wrapper_name LIBRARY 'libmssql3.a'
```

其中 *wrapper_name* 是您想要提供給外層的名稱，而 'libdjmssql.a' 是常式庫名稱。

請參閱 *DB2 SQL Reference* 中的 CREATE WRAPPER 陳述式，以取得有關外層常式庫名稱的詳細資訊。

步驟 6：建立伺服器

使用 CREATE SERVER 陳述式來定義您想要連接的每一個 Microsoft SQL Server 資料來源。例如：

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 7.0 WRAPPER djmssql3
OPTIONS (NODE 'sqlnode', DBNAME 'database_name')
```

其中：

sqlserver

代表您指定給 Microsoft SQL Server 伺服器的名稱。此名稱必須是唯一的。

MSSQLSERVER

代表您要架構存取的資料來源類型。

7.0

代表您要存取的 Microsoft SQL Server 版本。DB2 Universal Database 支援 Microsoft SQL Server 版本 6.5 與 7.0。

DJXMSSQL3

代表您在 CREATE WRAPPER 陳述式中定義的外層名稱。

'sqlnode'

代表參照 Microsoft SQL Server 的系統 DSN 名稱。此值有區分大小寫。您要存取的 Microsoft SQL Server 版本。DB2 Universal Database 支援 Microsoft SQL Server 版本 6.5 與 7.0。

雖然節點名稱 (系統 DSN 名稱) 是指定為 CREATE SERVER 陳述式中的選項，但此為 Microsoft SQL Server 資料來源所需的。

請參閱 *DB2 SQL Reference*，以取得您可以使用 CREATE WRAPPER 陳述式的其他選項。

'database_name'

代表您要連接的資料庫之名稱。

雖然資料庫名稱是指定為 CREATE SERVER 陳述式中的選項，但此為 Microsoft SQL Server 資料來源所需的。

步驟 7：建立使用者對映

若聯合伺服器中的使用者 ID 與通行碼，與 Microsoft SQL Server 中的使用者 ID 與通行碼不同，請使用 CREATE USER MAPPING 陳述式，使本端使用者 ID 與 Microsoft SQL Server 資料來源中定義的使用者 ID 與通行碼相互對映；例如：

```
CREATE USER MAPPING FOR db2user SERVER server_name  
OPTIONS (REMOTE_AUTHID 'mssqluser', REMOTE_PASSWORD 'dayllite')
```

其中：

db2user

代表您用來對映使用者 ID (定義於 Microsoft SQL Server 資料來源中) 的本端使用者 ID。

server_name

代表您在 CREATE SERVER 陳述式中定義的伺服器名稱。

'mssqluser'

代表 Microsoft SQL Server 資料來源的使用者 ID，您要用此來對映 *db2user*。此值有區分大小寫。

'dayllite'

代表與 *'mssqluser'* 相關的通行碼。此值有區分大小寫。

請參閱 *DB2 SQL Reference*，以取得您可以使用 CREATE USER MAPPING 陳述式的其他選項。

步驟 8：建立表格與概略表的暱名

指定您想要存取的 Microsoft SQL Server 資料來源中，每一個概略表及表格之暱名。當您查詢 Microsoft SQL Server 資料來源時，將使用這些暱名。使用 CREATE NICKNAME 陳述式來指定暱名。暱名有區分大小寫。下列範例顯示 CREATE NICKNAME 陳述式：

```
CREATE NICKNAME mssqlsales FOR server_name.salesdata.europe
```

其中：

mssqlsales

代表 Microsoft SQL Server 表格或概略表的專用暱名。

server_name.salesdata.europe

代表遵循下列格式的三部分識別字：

data_source_server_name.remote_schema_name.remote_table_name

建議您在的 *remote_schema_name* 及 *remote_table_name* 部份使用雙引號。

當您建立暱名時，DB2 會嘗試存取資料來源型錄表格 (Microsoft SQL Server 參照這些表格作為系統表格)。這會測試連接到資料來源的狀態。若連線失敗，您會接收到一個錯誤訊息。

針對您想要建立暱名的所有資料庫表格及概略表，重複此步驟。

有關 CREATE NICKNAME 陳述式的詳細資訊，請參閱 *DB2 SQL Reference*。有關一般暱名及驗證資料類型對映的詳細資訊，請參閱 *DB2 Administration Guide*。

步驟 9：選用的：獲得 ODBC 追蹤

若您存取資料來源時遇到問題，您可以取得 ODBC 追蹤資訊，以分析並解決這些問題。若要確定 ODBC 追蹤的運作正常，請使用「ODBC 資料來源管理者」提供的追蹤工具。啓動追蹤會影響系統效能，因此當您已解決問題時，應該關閉追蹤。

複查 Microsoft SQL Server 字碼頁

Microsoft SQL Server 支援許多 DB2 UDB 支援的一般「國家語言支援 (NLS)」字碼頁選項。使用與 DB2 相同的字碼集之資料來源不需要轉換。表格 3 列示 DB2 Universal Database 及 Microsoft SQL Server 均有支援的字碼頁。

表 3. DB2 UDB 及 Microsoft SQL Server 字碼頁選項

字碼頁	支援的語言
1252	ISO 字集
850	多種語言
437	英文
874	泰語
932	日文
936	簡體中文
949	韓語
950	繁體中文
1250	主要歐洲語
1251	斯拉夫語
1253	希臘語
1254	土耳其語
1255	希伯來文
1256	阿拉伯語

當 DB2 聯合伺服器與 Microsoft SQL Server 執行不同的「國家語言支援 (NLS)」字碼頁時，您的 Microsoft SQL Server 資料來源架構必須與其語意相符，或者從屬站程式碼必須能夠偵測不符的語意，並將其標示為錯誤、或使用其本身的語意來對映資料。若從原始字碼頁到目標字碼頁，都找不到轉換表，則 DB2 會發出錯誤訊息。請參閱您的 Microsoft SQL Server 文件，以取得詳細資訊。

第3篇 管理

Administration Guide: Planning (無中文版)	53	Sequence Privileges	70
Chapter 8. Physical Database Design	53	Data Encryption	70
Partitioning Keys	53	Chapter 8. Recovering a Database	71
Designing Nodegroups	53	How to Use Suspended I/O	71
Chapter 9. Designing Distributed Databases	53	Incremental Backup and Recovery	73
Updating Multiple Databases	53	Restoring from Incremental Backup Images	74
Chapter 13. High Availability in the Windows NT		Parallel Recovery	75
Environment	54	Backing Up to Named Pipes	75
Need to Reboot the Machine Before Running		Backup from Split Image	76
DB2MSPCS Utility	54	On Demand Log Archive	78
Chapter 14. DB2 and High Availability on Sun		Log Mirroring	78
Cluster 2.2	54	Cross Platform Backup and Restore Support on	
Veritas Support on Solaris	54	Sun Solaris and HP	79
Appendix B. Naming Rules	55	DB2 Data Links Manager Considerations/Backup	
Notes on Greater Than 8-Character User IDs and		Utility Considerations	79
Schema Names	55	DB2 Data Links Manager Considerations/Restore	
User IDs and Passwords	56	and Rollforward Utility Considerations	80
Appendix D. Incompatibilities Between Releases	56	Restoring Databases from an Offline Backup	
Windows NT DLFS Incompatible with Norton's		without Rolling Forward	81
Utilities	56	Restoring Databases and Table Spaces, and Rolling	
SET CONSTRAINTS Replaced by SET		Forward to the End of the Logs	82
INTEGRITY	57	DB2 Data Links Manager and Recovery	
Appendix E. National Language Support	57	Interactions	82
National Language Versions of DB2 Version 7	57	Detection of Situations that Require Reconciliation	86
Control Center and Documentation Filesets	59	Appendix C. User Exit for Database Recovery	86
Locale Setting for the DB2 Administration Server	60	Appendix D. Issuing Commands to Multiple Database	
DB2 UDB Supports the Baltic Rim Code Page		Partition Servers	87
(MS-1257) on Windows Platforms	60	Appendix I. High Speed Inter-node Communications	87
Deriving Code Page Values	60	Enabling DB2 to Run Using VI	87
Country Code and Code Page Support	60		
Character Sets	61		
Administration Guide: Implementation(無中文版)	63	Administration Guide: Performance (無中文版)	89
Adding or Extending DMS Containers (New Process)	63	Chapter 3. Application Considerations	89
Chapter 1. Administering DB2 using GUI Tools	63	Specifying the Isolation Level	89
Chapter 3. Creating a Database	64	Adjusting the Optimization Class	89
Creating a Table Space	64	Dynamic Compound Statements	90
Using Raw I/O on Linux	64	Chapter 4. Environmental Considerations	90
Creating a Sequence	65	Using Larger Index Keys	90
Comparing IDENTITY Columns and Sequences	67	Chapter 5. System Catalog Statistics	91
Creating an Index, Index Extension, or an Index		Collecting and Using Distribution Statistics	91
Specification	67	Rules for Updating Catalog Statistics	91
Chapter 4. Altering a Database	67	Sub-element Statistics	92
Adding a Container to an SMS Table Space on a		Chapter 6. Understanding the SQL Compiler	94
Partition	67	Replicated Summary Tables	94
Altering an Identity Column	68	Data Access Concepts and Optimization	96
Altering a Sequence	68	Chapter 8. Operational Performance	96
Dropping a Sequence	69	Managing the Database Buffer Pool	96
Switching the State of a Table Space	69	Managing Multiple Database Buffer Pools	97
Modifying Containers in a DMS Table Space	69	Chapter 9. Using the Governor	98
Chapter 5. Controlling Database Access	70	Chapter 13. Configuring DB2	98
		Sort Heap Size (sorheap)	98
		Sort Heap Threshold (sheapthres)	98

Maximum Percent of Lock List Before Escalation (maxlocks)	99	BACKUP DATABASE	132
Configuring DB2/DB2 Data Links Manager/Data Links Access Token Expiry Interval (dl_expint)	100	Syntax Diagram	132
MIN_DEC_DIV_3 Database Configuration Parameter	100	DB2 Data Links Manager Considerations.	133
Application Control Heap Size (app_ctl_heap_sz) Database System Monitor Heap Size (mon_heap_sz)	101	BIND	133
Maximum Number of Active Applications (maxappls)	101	CALL	134
Recovery Range and Soft Checkpoint Interval (softmax)	101	DROP DATALINKS MANAGER (new command) DROP DATALINKS MANAGER	135 136
Track Modified Pages Enable (trackmod).	101	EXPORT	139
Change the Database Log Path (newlogpath)	102	GET DATABASE CONFIGURATION	140
Location of Log Files (logpath).	103	GET ROUTINE (new command)	140
Maximum Storage for Lock List (locklist)	103	GET ROUTINE	141
Appendix A. DB2 Registry and Environment Variables	103	GET SNAPSHOT	141
Table of New and Changed Registry Variables	104	IMPORT	141
Appendix C. SQL Explain Tools	108	LIST HISTORY	142
Administering Satellites Guide and Reference(無中文版).	109	LOAD	143
Setting up Version 7.2 DB2 Personal Edition and DB2 Workgroup Edition as Satellites	109	PING (new command).	144
Prerequisites	109	PING	145
Installation Considerations.	110	PUT ROUTINE (new command)	146
Configuring the Version 7.2 System for Synchronization	111	PUT ROUTINE	147
Installing FixPak 2 or Higher on a Version 6 Enterprise Edition System.	112	RECONCILE.	147
Upgrading Version 6 DB2 Enterprise Edition for Use as the DB2 Control Server	113	REORGANIZE TABLE	148
Upgrading a Version 6 DB2 Control Center and Satellite Administration Center	120	RESTORE DATABASE	148
Command Reference(無中文版)	121	Syntax	148
db2batch - Benchmark Tool	121	DB2 Data Links Manager Considerations.	149
db2cap (new command)	121	ROLLFORWARD DATABASE	149
db2cap - CLI/ODBC Static Package Binding Tool	122	Documentation Error in CLP Return Codes	149
db2ckrst (new command)	123	Data Movement Utilities Guide and Reference(無中文版).	151
db2ckrst - Check Incremental Restore Image Sequence	124	Chapter 2. Import	151
db2gncol (new command).	125	Using Import with Buffered Inserts	151
db2gncol - Update Generated Column Values	126	Chapter 3. Load.	151
db2inidb - Initialize a Mirrored Database.	127	Pending States After a Load Operation	151
db2look - DB2 Statistics Extraction Tool.	127	Load Restrictions and Limitations	151
db2updv7 - Update Database to Version 7 Current Fix Level	128	totalreespace File Type Modifier	151
New Command Line Processor Option (-x, Suppress printing of column headings).	129	Chapter 4. AutoLoader.	151
True Type Font Requirement for DB2 CLP.	129	rexecd Required to Run Autoloader When Authentication Set to YES	151
ADD DATALINKS MANAGER	129	Replication 指南與參考手冊	153
ARCHIVE LOG (new command)	130	抄寫及非 IBM 伺服器	153
Archive Log	131	在 Windows 2000 上的抄寫.	153
		儲存 SQL 檔案時的已知錯誤	153
		DB2 維護.	153
		Web 上的 Data Difference Utility	153
		第 3 章 資料抄寫實務範例	153
		抄寫實務	153
		第 5 章 抄寫規劃	154
		表格及直欄名稱.	154
		DATALINK 抄寫	154
		LOB 限制	155
		規劃抄寫	155
		第 6 章 設定抄寫環境	155
		隨處更新 (Update-anywhere) 先決條件	155
		設定抄寫環境	155
		第 8 章 問題判斷	156

第 9 章 AS/400 的擷取程式和引用程式	156
第 10 章 OS/390 的擷取程式和引用程式	157
DB2 DataPropagator for OS/390 的先決條件	157
OS/390 上的 UNICODE 和 ASCII 編碼方法	157
選擇編碼方法	157
設定編碼方法	158
第 11 章 UNIX 平台的擷取程式和引用程式	159
在 UNIX 和 Windows 上設定「擷取程式」和 「引用程式」的環境變數.	159
第 14 章 表格結構	159
第 15 章 擷取程式和引用程式訊息	159
附錄 A. 從應用程式內啟動擷取程式和引用程式	160
System Monitor Guide and Reference(無中文 版).	161
db2ConvMonStream.	161
Troubleshooting Guide(無中文版).	163
Starting DB2 on Windows 95, Windows 98, and Windows ME When the User Is Not Logged On	163
Chapter 2. Troubleshooting the DB2 Universal Database Server	166
在 64 位元的平台上使用 DB2 Universal Database	169
第 5 章 架構	169
LOCKLIST	169
shmsys:shminfo_shmmax	169
第 6 章 限制	169
XML Extender Administration and Programming(無中文版)	171
MQSeries	173
安裝與架構 DB2 MQSeries Function	173
安裝 MQSeries	173
安裝 MQSeries AMI	173
啟用 DB2 MQSeries Function	174
MQSeries 傳訊樣式	174
訊息結構	175
MQSeries 功能概觀	175
限制	176
錯誤碼.	176
用法實務手冊	176
基本傳訊	176
傳送訊息	177
擷取訊息	178
應用程式對應用程式連接.	179
Request/Reply 通信.	179
發佈/訂閱.	180
enable_MQFunctions	182
enable_MQFunctions	183
disable_MQFunctions	184
disable_MQFunctions	185

Administration Guide: Planning (無中文版)

Chapter 8. Physical Database Design

Partitioning Keys

In the 『Nodegroup Design Considerations』 subsection of the 『Designing Nodegroups』 section, the following text from the 『Partitioning Keys』 sub-subsection stating the points to be considered when defining partitioning keys should be **deleted** only if `DB2_UPDATE_PART_KEY=ON`:

註: If `DB2_UPDATE_PART_KEY=OFF`, then the restrictions still apply.

註: In FixPak 3 and later, the default value will be ON.

- You cannot update the partitioning key column value for a row in the table.
- You can only delete or insert partitioning key column values.

Designing Nodegroups

Within the section titled 『Designing Nodegroups,』 the subsection titled 『Nodegroup Design Considerations,』 and the subsection titled 『Replicated Summary Tables,』 disregard the last sentence of the second paragraph:

The `REPLICATED` keyword can only be specified for a summary table that is defined with the `REFRESH DEFERRED` option.

Chapter 9. Designing Distributed Databases

Updating Multiple Databases

In the section "Updating Multiple Databases", the list of setup steps has an inaccuracy. Step 4, which now reads as follows:

Precompile your application program to specify a type 2 connection (that is, specify `CONNECT 2` on the `PRECOMPILE PROGRAM` command), and one-phase commit (that is, specify `SYNCPOINT ONEPHASE` on the `PRECOMPILE PROGRAM` command), as described in the Application Development Guide.

should be changed to:

Precompile your application program to specify a type 2 connection (that is, specify `CONNECT 2` on the `PRECOMPILE PROGRAM` command), and two-phase commit (that is, specify `SYNCPOINT TWOPHASE` on the `PRECOMPILE PROGRAM` command), as described in the Application Development Guide.

Chapter 13. High Availability in the Windows NT Environment

Need to Reboot the Machine Before Running DB2MSCS Utility

The DB2MSCS utility is used to perform the required setup to enable DB2 for failover support under the Microsoft Cluster Service environment. For the DB2MSCS utility to run successfully, the Cluster Service must be able to locate the resource DLL, db2wolf.dll, which resides under the %ProgramFiles%\SQLLIB\bin directory.

The DB2 UDB Version 7 Installation Program sets the PATH system environment variable to point to the %ProgramFiles%\SQLLIB\bin directory. However, it is not required that you reboot the machine after installation if you are running on the Windows 2000 operating system.

If you want to run the DB2MSCS utility, you must reboot the machine so that the PATH environment variable is updated for the Cluster Service.

Chapter 14. DB2 and High Availability on Sun Cluster 2.2

DB2 Connect is supported on Sun Cluster 2.2 if:

- The protocol to the host is TCP/IP (not SNA)
- Two-phase commit is not used. This restriction is relaxed if the user configures the SPM log to be on a shared disk (this can be done through the *spm_log_path* database manager configuration parameter), and the failover machine has an identical TCP/IP configuration (the same host name, IP address, and so on).

Veritas Support on Solaris

DB2 now supports Veritas, which provides cluster support for DB2 High Availability on Solaris.

Description

Brings online, takes offline, and monitors a DB2 UDB instance.

Entry Points

Online

Use db2start to bring up instance.

Offline

Use db2stop to bring down instance.

Monitor

Determines if specified DB2 instance is up. Uses appropriate process monitoring and (optional) database monitoring.

Clean Removes DB2 instance resources.

Attribute	Type	Definition
probeDatabase	string	Database to be monitored
instanceOwner	string	Instance owner name
instanceHome	string	Home directory of instance owner

probeTable	string	Table in probeDatabase to monitor
monitorLevel	integer	1 implies process monitoring, 2 implies db mon
nodeNumber	integer	Nodenumber of instance to start (unset is EE)

Type Definition

```

type DB2UDB (
    static int CleanTimeout = 240
    static int MonitorTimeout = 30
    static int OfflineTimeout = 240
    static int OnlineRetryLimit = 2
    static int OnlineTimeout = 120
    static int OnlineWaitLimit = 1
    static int RestartLimit = 3
    static int ToleranceLimit = 1
    static str ArgList[] =
{ probeDatabase, instanceOwner, instanceHome, probeTable, monitorLevel,
nodeNumber }
    NameRule = resource.db2udb
    str probeDatabase
    str instanceOwner
    str instanceHome
    str probeTable
    int monitorLevel
    int nodeNumber
)

```

Sample Configuration

```

DB2UDB db2_resource_n0 (
    probeDatabase = sample
    probeTable = vcstable
    instanceOwner = stevera
    instanceHome = "/export/home/stevera"
    monitorLevel = 2
)

```

Installation

- Create the directory /opt/VRTSvcs/bin/DB2UDB.
- Copy the files online, offline, monitor, clean, DB2UDBAgent into /opt/VRTSvcs/bin/DB2UDB and ensure that they are marked executable.
- Copy the file db2udb.type.cf into /etc/VRTSvcs/conf/config.
- Stop the cluster (for example, hastop -all).
- Add the line include db2udb.type.cf into the file main.cf after the line include types.cf
- Verify the cluster configuration is valid with /opt/VRTSvcs/bin/hacf -verify /etc/VRTSvcs/conf/config

You are now ready to create the DB2 resources necessary to control DB2 instances.

Appendix B. Naming Rules

Notes on Greater Than 8-Character User IDs and Schema Names

- DB2 Version 7 products on Windows 32-bit platforms support user IDs that are up to 30 characters long. However, because of native support of Windows NT and Windows 2000, the practical limit for user ID is 20 characters.

- DB2 Version 7 supports non-Windows 32-bit clients connecting to Windows NT and Windows 2000 with user IDs longer than 8 characters when user ID and password are being specified explicitly. This excludes connections using Client or DCE authentication.
- DCE authentication on all platforms continues to have the 8-character user ID limit.
- The authid returned in the SQLCA from a successful CONNECT or ATTACH is truncated to 8 characters. The SQLWARN fields contain warnings when truncation occurs. For more information, refer to the description of the CONNECT statement in the *SQL Reference*.
- The authid returned by the command line processor (CLP) from a successful CONNECT or ATTACH is truncated to 8 characters. An ellipsis (...) is appended to the authid to indicate truncation.
- DB2 Version 7 supports schema names with length up to 30 bytes, with the following exceptions:
 - Tables with schema names longer than 18 bytes cannot be replicated.
 - User defined types (UDTs) cannot have schema names longer than 8 bytes.

User IDs and Passwords

Within the section titled 『User IDs and Passwords,』 change the reference to 『A through Z』 to:

Single-byte uppercase and lowercase Latin letters (A...Z, a...z). Support for other letters and characters depends on the code page being used. See the appendix titled 『National Language Support (NLS)』 for more information on code page support.

Appendix D. Incompatibilities Between Releases

Windows NT DLFS Incompatible with Norton's Utilities

The Windows NT Data Links File System is incompatible with Norton Utilities. When a file is deleted from a drive controlled by DLFS, a kernel exception results: error 0x1E (Kernel Mode Exception Not Handled). The exception being 0xC0000005 (Access Violation). This access violation happens because the Norton Utilities driver gets loaded after the DLFS filter driver gets loaded.

A temporary work-around is to load the DLFS driver, after the Norton Utilities driver is loaded. This work-around can be done by changing the DLFS driver startup to manual. Click on **Start** and select **Settings—> Control Panel—> Devices—>DLFS** and set it to manual.

A batch file, that can be added to the startup folder, can be created which loads the DLFS driver and the DLFM Service on system startup. The contents of the batch file are as follows:

```
net start dlfsd
net start "dlfm service"
```

Name this batch file start_dlfs.bat, and copy it into the C:\WINNT\Profiles\Administrator\Start Menu\Programs\Startup directory.

Only the administrator has the privilege to load the DLFS filter driver and the DLFM service.

SET CONSTRAINTS Replaced by SET INTEGRITY

The SET CONSTRAINTS statement has been replaced by the SET INTEGRITY statement. For backwards compatibility, both statements are accepted in DB2 UDB V7.

Appendix E. National Language Support

National Language Versions of DB2 Version 7

DB2 Version 7 is available in English, French, German, Italian, Spanish, Brazilian Portuguese, Japanese, Korean, Simplified Chinese, Traditional Chinese, Danish, Finnish, Norwegian, Swedish, Czech, Dutch, Hungarian, Polish, Turkish, Russian, Bulgarian, and Slovenian.

On UNIX-based platforms, the DB2 product messages and library can be installed in several different languages. The DB2 installation utility lays down the message catalog file sets into the most commonly used locale directory for a given platform as shown in the following tables. Table 4 provides information for AIX, HP-UX, and Solaris. Table 5 provides information for Linux, Linux/390, SGI, and Dynix.

Table 4. AIX, HP-UX, Solaris

Operating System	AIX		HP-UX		Solaris	
Language	Locale	Cde Pg	Locale	Cde Pg	Locale	Cde Pg
French	fr_FR	819	fr_FR.iso88591	819	fr	819
	Fr_FR	850	fr_FR.roman8	1051		
German	de_DE	819	de_DE.iso88591	819	de	819
	De_DE	850	de_DE.roman8	1051		
Italian	it_IT	819	it_IT.iso88591	819	it	819
	It_IT	850	it_IT.roman8	1051		
Spanish	es_ES	819	es_ES.iso88591	819	es	819
	Es_ES	850	es_ES.roman8	1051		
Brazilian Portuguese	pt_BR	819			pt_BR	819
Japanese	ja_JP	954	ja_JP.eucJP	954	ja	954
	Ja_JP	932				
Korean	ko_KR	970	ko_KR.eucKR	970	ko	970
Simplified Chinese	zh_CN	1383	zh_CN.hp15CN	1383	zh	1383
	Zh_CN.GBK	1386				
Traditional Chinese	zh_TW	964	zh_TW.eucTW	964	zh_TW	964
	Zh_TW	950	zh_TW.big5	950		

Table 4. AIX, HP-UX, Solaris (continue)

Danish	da_DK	819	da_DK.iso88591	819	da	819
	Da_DK	850	da_DK.roman8	1051		
Finnish	fi_FI	819	fi_FI.iso88591	819	fi	819
	Fi_FI	850	fi_FI.roman8	1051		
Norwegian	no_NO	819	no_NO.iso88591	819	no	819
	No_NO	850	no_NO.roman8	1051		
Sweden	sv_SE	819	sv_SE.iso88591	819	sv	819
	Sv_SE	850	sv_SE.roman8	1051		
Czech	cs_CZ	912				
Hungarian	hu_HU	912				
Polish	pl_PL	912				
Dutch	nl_NL	819				
	Nl_NL	850				
Turkish	tr_TR	920				
Russian	ru_RU	915				
Bulgarian	bg_BG	915	bg_BG.iso88595	915		
Slovenian	sl_SI	912	sl_SI.iso88592	912	sl_SI	912

Table 5. Linux, Linux/390, SGI, Dynix

Operating System	Linux		Linux/390		SGI		Dynix	
	Locale	C d e Pg	Locale	C d e Pg	Locale	C d e Pg	Locale	C d e Pg
French	fr	819	fr	819			fr	819
German	de	819	de	819			de	819
Italian							es	819
Spanish								
Brazilian Portuguese								
Japanese	ja_JP.ujis	954	ja_JP.ujis	954			ja_JP.EUC	954
Korean	ko	970	ko	970	ko_KO.euc	970		
Simplified Chinese	zh_zh_CN.GBK	1386	zh_zh_CN.GBK	1386				
Traditional Chinese	zh_TW.Big5	950	zh_TW.Big5	950				
Danish								
Finnish								
Norwegian								
Sweden								
Czech								
Hungarian								
Polish								
Dutch							nl	819

Table 5. Linux, Linux/390, SGI, Dynix (continue)

Turkish								
Russian								
Bulgarian								
Slovenian								

If your system uses the same code pages but different locale names than those provided above, you can still see the translated messages by creating a link to the appropriate message directory.

For example, if your AIX machine default locale is ja_JP.IBM-eucJP and the code page of ja_JP.IBM-eucJP is 954, you can create a link from /usr/lpp/db2_07_01/msg/ja_JP.IBM-eucJP to /usr/lpp/db2_07_01/msg/ja_JP by issuing the following command:

```
ln -s /usr/lpp/db2_07_01/msg/ja_JP /usr/lpp/db2_07_01/msg/ja_JP.IBM-eucJP
```

After the execution of this command, all DB2 messages come up in Japanese.

Control Center and Documentation Filesets

The Control Center, Control Center Help and documentation filesets are placed in the following directories on the target workstation:

- DB2 for AIX:
 - /usr/lpp/db2_07_01/cc/%L
 - /usr/lpp/db2_07_01/java/%L
 - /usr/lpp/db2_07_01/doc/%L
 - /usr/lpp/db2_07_01/qp/\$L
 - /usr/lpp/db2_07_01/spb/%L
- DB2 for HP-UX:
 - /opt/IBMDB2/V7.1/cc/%L
 - /opt/IBMDB2/V7.1/java/%L
 - /opt/IBMDB2/V7.1/doc/%L
- DB2 for Linux:
 - /usr/IBMDB2/V7.1/cc/%L
 - /usr/IBMDB2/V7.1/java/%L
 - /usr/IBMDB2/V7.1/doc/%L
- DB2 for Solaris:
 - /opt/IBMDB2/V7.1/cc/%L
 - /usr/IBMDB2/V7.1/java/%L
 - /opt/IBMDB2/V7.1/doc/%L

Control Center file sets are in Unicode code page. Documentation and Control Center help file sets are in a browser-recognized code set. If your system uses a different locale

name than the one provided, you can still run the translated version of the Control Center and see the translated version of help by creating links to the appropriate language directories.

For example, if your AIX machine default locale is ja_JP.IBM-eucJP, you can create links from /usr/lpp/db2_07_01/cc/ja_JP.IBM-eucJP to /usr/lpp/db2_07_01/cc/ja_JP and from /usr/lpp/db2_07_01/doc/ja_JP.IBM-eucJP to /usr/lpp/db2_07_01/doc/ja_JP by issuing the following commands:

- **In -s /usr/lpp/db2_07_01/cc/ja_JP /usr/lpp/db2_07_01/cc/ja_JP.IBM-eucJP**
- **In -s /usr/lpp/db2_07_01/doc/ja_JP /usr/lpp/db2_07_01/doc/ja_JP.IBM-eucJP**

After the execution of these commands, the Control Center and help text come up in Japanese.

Note: The Web Control Center is not supported running natively on Linux/390 or NUMA-Q. It can be used from a client workstation to manage databases on these platforms.

Locale Setting for the DB2 Administration Server

Please ensure that the locale of the DB2 Administration Server instance is compatible to the locale of the DB2 instance. Otherwise, the DB2 instance cannot communicate with the DB2 Administration Server.

If the LANG environment variable is not set in the user profile of the DB2 Administration Server, the DB2 Administration Server will be started with the default system locale. If the default system locale is not defined, the DB2 Administration Server will be started with code page 819. If the DB2 instance uses one of the DBCS locales, and the DB2 Administration Server is started with code page 819, the instance will not be able to communicate with the DB2 Administration Server. The locale of the DB2 Administration Server and the locale of the DB2 instance must be compatible. For example, on a Simplified Chinese Linux system, "LANG=zh_CN" should be set in the DB2 Administration Server's user profile.

DB2 UDB Supports the Baltic Rim Code Page (MS-1257) on Windows Platforms

DB2 UDB supports the Baltic Rim code page, MS-1257, on Windows 32-bit operating systems. This code page is used for Latvian, Lithuanian, and Estonian.

Deriving Code Page Values

Within the section titled 『Deriving Code Page Values,』 change the first paragraph, from:

However, it is not necessary to set the DB2CODEPAGE registry variable, because DB2 will determine the appropriate code page value from the operating system.

to: Normally, you do not need to set the DB2CODEPAGE registry variable because DB2 automatically derives the code page information from the operating system.

Country Code and Code Page Support

Within the section titled 『Country Code and Code Page Support,』 add the following information to the table:

Code Page	Group	Code-Set	Tr.	Country Code	Locale	OS	Country Name
943	D-1	IBM-943	JP	81	ja_JP.PCK	Sun	Japan

Character Sets

Within the section titled 『Character Sets』 and the subsection 『Character Set for Identifiers,』 replace the last two sentences in the first paragraph with the following:

Use special characters #, @, and \$ with care in an NLS environment because they are not included in the NLS host (EBCDIC) invariant character set. Characters from the extended character set can also be used, depending on the code page that is being used. If you are using the database in a multiple code page environment, you must ensure that all code pages support any elements from the extended character set you plan to use.

Administration Guide: Implementation(無中文版)

Adding or Extending DMS Containers (New Process)

DMS containers (both file containers and raw device containers) which are added (during tablespace creation or after) or extended are now done so in parallel through the prefetchers. To achieve an increase in parallelism of these create / resize container operations, one can increase the number of prefetchers running in the system. The only process which is not done in parallel is the logging of these actions and, in the case of creating containers, the tagging of the containers.

註: Parallelism of the CREATE TABLESPACE / ALTER TABLESPACE (with respect to adding new containers to an existing tablespace) will no longer increase when the number of prefetchers equals the number of containers being added.

Chapter 1. Administering DB2 using GUI Tools

Within the section titled 『The Alert Center』, remove the last two sentences in the section.

Within the section titled 『Performance Monitor』, remove the second bullet item from the 『Define performance variables』 list in the subsection 『Monitoring Performance at a Point in Time.』

Also later in this same subsection, the last few paragraphs in the section should be rewritten as follows:

For each, a variety of performance variables can be monitored. The Performance Variable Reference Help, available from the **Help** menu of any Snapshot Monitor window, provides a description of all the performance variables. These variables are organized into categories. The following categories exist:

- Instance: Agents, Connections, Sort
- Database: Lock and Deadlock, Buffer Pool and I/O, Connections, Sort, SQL Statement Activity
- Table: Table
- Table space: Buffer Pool and I/O
- Database Connections: Buffer Pool and I/O, Lock and Deadlock, Sort, SQL Statement Activity

For detailed information on how to generate snapshots, see the online help.

In this same section, remove the last sentence in the subsection titled 『Action Required When an Object Appears in the Alert Center.』

Chapter 3. Creating a Database

Creating a Table Space

Using Raw I/O on Linux

Linux has a pool of raw device nodes that must be bound to a block device before raw I/O can be performed on it. There is a raw device controller that acts as the central repository of raw to block device binding information. Binding is performed using a utility named `raw`, which is normally supplied by the Linux distributor.

Before you set up raw I/O on Linux, you require the following:

- one or more free IDE or SCSI disk partitions
- Linux kernel 2.4.0 or later (However, some Linux distributions offer raw I/O on 2.2 kernels.)
- a raw device controller named `/dev/rawctl` or `/dev/raw`. If not, create a symbolic link:

```
# ln -s /dev/your_raw_dev_ctrl /dev/rawctl
```
- the `raw` utility, which is usually provided with the Linux distribution
- DB2 Version 7.1 FixPak 3 or later

註: Of the distributions currently supporting raw I/O, the naming of raw device nodes is different:

Distribution	Raw device nodes	Raw device controller
RedHat 6.2	<code>/dev/raw/raw1</code> to 255	<code>/dev/rawctl</code>
SuSE 7.0	<code>/dev/raw1</code> to 63	<code>/dev/raw</code>

DB2 supports either of the above raw device controllers, and most other names for raw device nodes. Raw devices are not supported by DB2 on Linux/390.

To configure raw I/O on Linux:

In this example, the raw partition to be used is `/dev/sda5`. It should not contain any valuable data.

Step 1. Calculate the number of 4 096-byte pages in this partition, rounding down if necessary. For example:

```
# fdisk /dev/sda
Command (m for help): p

Disk /dev/sda: 255 heads, 63 sectors, 1106 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot   Start    End  Blocks  Id System
  /dev/sda1             1    523   420097  83  Linux
  /dev/sda2             524   1106  4682947+  5  Extended
  /dev/sda5             524   1106  4682947  83  Linux

Command (m for help): q
#
```

The number of pages in `/dev/sda5` is


```
num_pages = floor( ((1106-524+1)*16065*512)/4096 )
num_pages = 11170736
```

Step 2. Bind an unused raw device node to this partition. This needs to be done every time the machine is rebooted, and requires root access. Use `raw -a` to see which raw device nodes are already in use:

```
# raw /dev/raw/raw1 /dev/sda5
/dev/raw/raw1: bound to major 8, minor 5
```

Step 3. Set global read permissions on the raw device controller and the disk partition. Set global read and write permissions on the raw device:

```
# chmod a+r /dev/rawctl
# chmod a+r /dev/sdb1
# chmod a+r /dev/raw/raw1
```

Step 4. Create the table space in DB2, specifying the raw device, not the disk partition. For example:

```
CREATE TABLESPACE dms1
MANAGED BY DATABASE
USING (DEVICE '/dev/raw/raw1' 11170736)
```

Table spaces on raw devices are also supported for all other page sizes supported by DB2.

Creating a Sequence

Following the section titled 『Defining an Identity Column on a New Table,』 add the following section, "Creating a Sequence":

A *sequence* is a database object that allows the automatic generation of values. Sequences are ideally suited to the task of generating unique key values. Applications can use sequences to avoid possible concurrency and performance problems resulting from the generation of a unique counter outside the database.

Unlike an identity column attribute, a sequence is not tied to a particular table column nor is it bound to a unique table column and only accessible through that table column.

A sequence can be created, or altered, so that it generates values in one of these ways:

- Increment or decrement monotonically without bound
- Increment or decrement monotonically to a user-defined limit and stop
- Increment or decrement monotonically to a user-defined limit and cycle back to the beginning and start again

The following is an example of creating a sequence object:

```
CREATE SEQUENCE order_seq
START WITH 1
INCREMENT BY 1
NOMAXVALUE
NOCYCLE
CACHE 24
```

In this example, the sequence is called `order_seq`. It will start at 1 and increase by 1 with no upper limit. There is no reason to cycle back to the beginning and restart from

1 because there is no assigned upper limit. The number associated with the CACHE parameter specifies the maximum number of sequence values that the database manager preallocates and keeps in memory.

The sequence numbers generated have the following properties:

- Values can be any exact numeric data type with a scale of zero. Such data types include: SMALLINT, BIGINT, INTEGER, and DECIMAL.
- Consecutive values can differ by any specified integer increment. The default increment value is 1.
- Counter value is recoverable. The counter value is reconstructed from logs when recovery is required.
- Values can be cached to improve performance. Preallocating and storing values in the cache reduces synchronous I/O to the log when values are generated for the sequence. In the event of a system failure, all cached values that have not been committed are never used and considered lost. The value specified for CACHE is the maximum number of sequence values that could be lost.

If a database that contains one or more sequences is recovered to a prior point in time, then this could cause the generation of duplicate values for some sequences. To avoid possible duplicate values, a database with sequences should not be recovered to a prior point in time.

Sequences are only supported in a single node database.

There are two expressions used with a sequence.

The PREVVAL expression returns the most recently generated value for the specified sequence for a previous statement within the current session.

The NEXTVAL expression returns the next value for the specified sequence. A new sequence number is generated when a NEXTVAL expression specifies the name of the sequence. However, if there are multiple instances of a NEXTVAL expression specifying the same sequence name within a query, the counter for the sequence is incremented only once for each row of the result.

The same sequence number can be used as a unique key value in two separate tables by referencing the sequence number with a NEXTVAL expression for the first table, and a PREVVAL expression for any additional tables.

For example:

```
INSERT INTO order (orderno, custno)
VALUES (NEXTVAL FOR order_seq, 123456);
INSERT INTO line_item (orderno, partno, quantity)
VALUES (PREVVAL FOR order_seq, 987654, 1)
```

The NEXTVAL or PREVVAL expressions can be used in the following locations:

- INSERT statement, VALUES clause
- SELECT statement, SELECT list
- SET assignment statement
- UPDATE statement, SET clause

- VALUES or VALUES INTO statement

Comparing IDENTITY Columns and Sequences

Following the new section titled 『Creating a Sequence』, add the following section:

While there are similarities between IDENTITY columns and sequences, there are also differences. The characteristics of each can be used when designing your database and applications.

An identity column has the following characteristics:

- An identity column can be defined as part of a table only when the table is created. Once a table is created, you cannot alter it to add an identity column. (However, existing identity column characteristics may be altered.)
- An identity column automatically generates values for a single table.
- When an identity column is defined as GENERATED ALWAYS, the values used are always generated by the database manager. Applications are not allowed to provide their own values during the modification of the contents of the table.

A sequence object has the following characteristics:

- A sequence object is a database object that is not tied to any one table.
- A sequence object generates sequential values that can be used in any SQL statement.
- Since a sequence object can be used by any application, there are two expressions used to control the retrieval of the next value in the specified sequence and the value generated previous to the statement being executed. The PREVVAL expression returns the most recently generated value for the specified sequence for a previous statement within the current session. The NEXTVAL expression returns the next value for the specified sequence. The use of these expressions allows the same value to be used across several SQL statements within several tables.

While these are not all of the characteristics of these two items, these characteristics will assist you in determining which to use depending on your database design and the applications using the database.

Creating an Index, Index Extension, or an Index Specification

Within the section titled 『Creating an Index, Index Extension, or an Index Specification』, add the following note in the paragraph beginning with the sentence: 『Any column that is part of an index key is limited to 255 bytes.』

註: The DB2_INDEX_2BYTEVARLEN registry variable can be used to allow columns with a length greater than 255 bytes to be specified as part of an index key.

Chapter 4. Altering a Database

Under the section 『Altering a Table Space』, the following new sections are to be added:

Adding a Container to an SMS Table Space on a Partition

You can add a container to an SMS table space on a partition (or node) that currently has no containers.

The contents of the table space are rebalanced across all containers. Access to the table space is not restricted during the rebalancing. If you need to add more than one container, you should add them all at the same time.

To add a container to an SMS table space using the command line, enter the following:

```
ALTER TABLESPACE <name>
  ADD ('<path>')
  ON NODE (<partition_number>)
```

The partition specified by number, and every partition (or node) in the range of partitions, must exist in the nodegroup on which the table space is defined. A `partition_number` may only appear explicitly or within a range in exactly one *on-nodes-clause* for the statement.

The following example shows how to add a new container to partition number 3 of the nodegroup used by table space 『plans』 on a UNIX based operating system:

```
ALTER TABLESPACE plans
  ADD ('/dev/rhdisk0')
  ON NODE (3)
```

Following the section titled 『Changing Table Attributes,』 add the following sections:

Altering an Identity Column

Modify the attributes of an existing identity column with the ALTER TABLE statement. For more information on this statement, including its syntax, refer to the *SQL Reference*.

There are several ways to modify an identity column so that it has some of the characteristics of sequences.

There are some tasks that are unique to the ALTER TABLE and the identity column:

- RESTART resets the sequence associated with the identity column to the value specified implicitly or explicitly as the starting value when the identity column was originally created.
- RESTART WITH <numeric-constant> resets the sequence associated with the identity column to the exact numeric constant value. The numeric constant could be any positive or negative value with no non-zero digits to the right of any decimal point that could be assigned to the identity column.

Altering a Sequence

Modify the attributes of an existing sequence with the ALTER SEQUENCE statement. For more information on this statement, including its syntax, refer to the *SQL Reference*.

The attributes of the sequence that can be modified include:

- Changing the increment between future values
- Establishing new minimum or maximum values
- Changing the number of cached sequence numbers
- Changing whether the sequence will cycle or not
- Changing whether sequence numbers must be generated in order of request
- Restarting the sequence

There are two tasks that are not found as part of the creation of the sequence. They are:

- **RESTART.** Resets the sequence to the value specified implicitly or explicitly as the starting value when the sequence was created.
- **RESTART WITH numeric-constant.** Resets the sequence to the exact numeric constant value. The numeric constant can be any positive or negative value with no non-zero digits to the right of any decimal point.

After restarting a sequence or changing to **CYCLE**, it is possible to generate duplicate sequence numbers. Only future sequence numbers are affected by the **ALTER SEQUENCE** statement.

The data type of a sequence cannot be changed. Instead, you must drop the current sequence and then create a new sequence specifying the new data type.

All cached sequence values not used by DB2 are lost when a sequence is altered.

Dropping a Sequence

To delete a sequence, use the **DROP** statement. For more information on this statement, including its syntax, refer to the *SQL Reference*.

A specific sequence can be dropped by using:

```
DROP SEQUENCE sequence_name
```

where the `sequence_name` is the name of the sequence to be dropped and includes the implicit or explicit schema name to exactly identify an existing sequence.

Sequences that are system-created for **IDENTITY** columns cannot be dropped using the **DROP SEQUENCE** statement.

Once a sequence is dropped, all privileges on the sequence are also dropped.

Switching the State of a Table Space

The **SWITCH ONLINE** clause of the **ALTER TABLESPACE** statement can be used to move table spaces in an **OFFLINE** state to an **ONLINE** state if the containers associated with that table space have become accessible. The table space is moved to an **ONLINE** state while the rest of the database is still up and being used.

An alternative to the use of this clause is to disconnect all applications from the database and then to have the applications connect to the database again. This moves the table space from an **OFFLINE** state to an **ONLINE** state.

To switch the table space to an **ONLINE** state using the command line, enter:

```
ALTER TABLESPACE <name>  
SWITCH ONLINE
```

Modifying Containers in a DMS Table Space

DMS table spaces are now created and resized in parallel, which offers a performance benefit. The degree of parallelism is equal to the number of prefetchers plus 1.

Chapter 5. Controlling Database Access

Following the section titled 『Index Privileges,』 add the following section:

Sequence Privileges

The creator of a sequence automatically receives the USAGE privilege. The USAGE privilege is needed to use NEXTVAL and PREVVVAL expressions for the sequence. To allow other users to use the NEXTVAL and PREVVVAL expressions, sequence privileges must be granted to PUBLIC. This allows all users to use the expressions with the specified sequence.

Following the section titled 『Monitoring Access to Data Using the Audit Facility,』 add the following section:

Data Encryption

One part of your security plan may involve encrypting your data. To do this, you can use encryption and decryption built-in functions: ENCRYPT, DECRYPT_BIN, DECRYPT_CHAR, and GETHINT. For more information on these functions, including their syntax, refer to the SQL Reference section of the Release Notes.

The ENCRYPT function encrypts data using a password-based encryption method. These functions also allow you to encapsulate a password hint. The password hint is embedded in the encrypted data. Once encrypted, the only way to decrypt the data is by using the correct password. Developers that choose to use these functions should plan for the management of forgotten passwords and unusable data.

The result of the ENCRYPT functions is the same data type as the first argument.

Only VARCHARs can be encrypted.

The declared length of the result is one of the following:

- The length of the data argument plus 42 when the optional hint parameter is specified.
- The length of the data argument plus 10 when the optional hint parameter is not specified.

The DECRYPT_BIN and DECRYPT_CHAR functions decrypt data using password-based decryption.

The result of the DECRYPT_BIN and DECRYPT_CHAR functions is the same data type as the first argument.

The declared length of the result is the length of the original data.

The GETHINT function returns an encapsulated password hint. A password hint is a phrase that will help data owners remember passwords. For example, the word 『Ocean』 can be used as a hint to remember the password "Pacific".

The password that is used to encrypt the data is determined in one of two ways:

- Password Argument. The password is a string that is explicitly passed when the ENCRYPT function is invoked. The data is encrypted and decrypted with the given password.

- **Special Register Password.** The SET ENCRYPTION PASSWORD statement encrypts the password value and sends the encrypted password to the database manager to store in a special register. ENCRYPT, DECRYPT_BIN and DECRYPT_CHAR functions invoked without a password parameter use the value in the ENCRYPTION PASSWORD special register.

The initial or default value for the special register is an empty string.

Valid lengths for passwords are between 6 and 127 inclusive. Valid lengths for hints are between 0 and 32 inclusive.

When the ENCRYPTION PASSWORD special register is set from the client, the password is encrypted at the client, sent to the database server, and then decrypted. To ensure that the password is not left readable, it is also re-encrypted at the database server. DECRYPT_BIN and DECRYPT_CHAR functions must decrypt the special register before use. The value found in the ENCRYPTION PASSWORD is also not left readable. Gateway security is not supported.

Chapter 8. Recovering a Database

How to Use Suspended I/O

In Chapter 8. 『Recovering a Database』, the following new section on using the suspended I/O function is to be added:

Note: The information below about the db2inidb utility supercedes the information in the Version 7.2 What's New book.

db2inidb is a new tool shipped with DB2 that can perform crash recovery and put a database in rollforward pending state.

Suspended I/O supports continuous system availability by providing a full implementation for online split mirror handling, that is, splitting a mirror without shutting down the database. If a customer cannot afford doing offline or online backups on a large database, backups or system copies can be done from a mirror image by using suspended I/O and split mirror.

Depending on how the storage devices are being mirrored, the uses of db2inidb will vary. The following uses assume that the entire database is mirrored consistently through the storage system.

In a multi-node environment, the db2inidb tool must be run on every partition before the split image can be used from any of the partitions. The db2inidb tool can be run on all partitions simultaneously.

1. Making a Clone Database

The objective here is to have a clone of the primary database to be used for read-only purposes. The following procedure describes how a clone database may be made:

- a. Suspend I/O on the primary system by entering the following command:

```
db2 set write suspend for database
```

b. Use the operating system level command to split the mirror from the primary database.

c. Resume I/O on primary system by entering the following command:

```
db2 set write resume for database
```

After running the command, the database on the primary system should be back to a normal state.

d. Attach to the mirrored database from another machine.

e. Start the database instance by entering the following command:

```
db2start
```

f. Start the DB2 crash recovery by entering the following command:

```
db2inidb database_name AS SNAPSHOT
```

Note: This command will rollback the changes made by transactions that are inflight at the time of the split.

You can also use this process for an offline backup, but if restored on the primary system, this backup cannot be used to roll forward, because the log chain will not match.

2. Using the Split Mirror as a Standby Database

As the mirrored (standby) database is continually rolling forward through the logs, new logs that are being created by the primary database, are constantly fetched from the primary system. The following procedure describes how the split mirror can be used as a standby database:

a. Suspend I/O writes on the primary database.

b. Split the mirror from the primary system.

c. Resume the I/O writes on the primary database so that the primary database goes back to normal processing.

d. Attach the mirrored database to another instance.

e. Place the mirror in roll-forward pending and roll forward the mirror. Run the db2inidb tool (db2inidb <db_alias> as standby) to remove the suspend write state and to place the mirrored database in a roll-forward pending state.

f. Copy logs by setting up a user exit program to retrieve log files from the primary system to ensure that the latest logs will be available for this mirrored database.

g. Roll forward the database to the end of the logs.

h. Go back to step f and repeat this process until the primary database is down.

3. Using the Split Mirror as a Backup Image

The following procedure describes how to use the mirrored system as a backup image to restore over the primary system:

a. Use operating system commands to copy the mirrored data and logs on top of the primary system.

b. Start the database instance by entering the following command:

```
db2start
```


- c. Run the following command to place the mirrored database in a roll-forward pending state and to remove the suspend write state.

```
db2inidb database_alias AS MIRROR
```
- d. Roll forward the database to the end of the logs.

Incremental Backup and Recovery

In Chapter 8, 『Recovering a Database,』 the following is a new section about incremental backup and recovery:

As the size of databases, and particularly warehouses, continues to expand into the terabyte and petabyte range, the time and hardware resources required to back up and recover these databases are also growing substantially. Full database and table space backups are not always the best approach when dealing with large databases, because the storage requirements for multiple copies of such databases are enormous. Consider the following issues:

- When a small percentage of the data in a warehouse changes, it should not be necessary to back up the entire database.
- Appending table spaces to existing databases and then taking only table space backups is risky, because data outside of the backed up table spaces may change.

DB2 now supports incremental backup and recovery (but not of long field or large object data). An *incremental backup* is a backup image that contains only pages that have been updated since the previous backup was taken. In addition to updated data and index pages, each incremental backup image also contains all of the initial database meta-data (such as database configuration, table space definitions, database history, and so on) that is normally stored in full backup images.

Two types of incremental backup are supported:

- *Incremental*. An incremental backup image is a copy of all database data that has changed since the most recent successful full backup operation. This is also known as a cumulative backup image, because a series of incremental backups taken over time will each have the contents of the previous incremental backup image. The predecessor of an incremental backup image is always the most recent successful full backup of the same object.
- *Delta*. A delta, or incremental delta, backup image is a copy of all database data that has changed since the last successful backup (full, incremental, or delta) of the table space in question. This is also known as a differential, or non-cumulative, backup image. The predecessor of a delta backup image is the most recent successful backup containing a copy of each of the table spaces in the delta backup image.

The key difference between incremental and delta backup images is their behavior when successive backups are taken of an object that is continually changing over time. Each successive incremental image contains the entire contents of the previous incremental image, plus any data that has changed, or is new, since the previous backup was produced. Delta backup images contain only the pages that have changed since the previous image was produced.

Combinations of database and table space incremental backups are permitted, in both online and offline modes of operation. Be careful when planning your backup strategy, because

combining database and table space incremental backups implies that the predecessor of a database backup (or a table space backup of multiple table spaces) is not necessarily a single image, but could be a unique set of previous database and table space backups taken at different times.

To rebuild the database or the table space to a consistent state, the recovery process must begin with a consistent image of the entire object (database or table space) to be restored, and must then apply each of the appropriate incremental backup images in the order described below (see the "Restore Method" section).

To enable the tracking of database updates, DB2 supports a new database configuration parameter, TRACKMOD, which can have one of two accepted values:

- NO. Incremental backup is not permitted with this configuration. Database page updates are not tracked or recorded in any way.
- YES. Incremental backup is permitted with this configuration. When update tracking is enabled, the change becomes effective at the first successful connection to any database in the instance. A full database backup is necessary before an incremental backup can be taken.

The default TRACKMOD setting for existing databases is NO; for new databases, it is YES.

The granularity of the tracking is at the table space level for both SMS and DMS table spaces.

Although minimal, the tracking of updates to the database can have an impact on the run-time performance of transactions that update or insert data.

Restoring from Incremental Backup Images

A restore operation from incremental backup images always consists of the following steps:

1. Identifying the incremental target image. The DBA must first determine the final image to be restored, and request an incremental restore operation from the DB2 restore utility. This image is known as the target image of the incremental restore, because it will be the last image to be restored. An incremental restore command against this image may initiate the creation of a new database with the configuration and table space definitions from this target image. The incremental target image is specified using the TAKEN AT parameter in the RESTORE DATABASE command.
2. Restoring the most recent full database or table space image to establish a baseline against which each of the subsequent incremental backup images can be applied.
3. Restoring each of the required full or table space incremental backup images, in the order in which they were produced, on top of the baseline image restored in Step 2.
4. Repeating Step 3 until the target image from Step 1 is read a second time. The target image is accessed twice during a complete incremental restore operation. During the first access, only initial data is read from the image; none of the user data is read. The complete image is read and processed only during the second access.

The target image of the incremental restore operation must be accessed twice to ensure that the database is initially configured with the correct history, database configuration, and table space definitions for the database that will be created during the restore

operation. In cases where a table space has been dropped since the initial full database backup image was taken, the table space data for that image will be read from the backup images but ignored during incremental restore processing.

For example:

1. db2 restore database sample incremental taken at <ts>

where:

<ts> points to the last incremental backup image to be restored

2. db2 restore database sample incremental taken at <ts1>

where:

<ts1> points to the initial full database (or table space) image

3. db2 restore database sample incremental taken at <tsX>

where:

<tsX> points to each incremental backup image in creation sequence

4. Repeat Step 3, restoring each incremental backup image up to and including image <ts>

In cases where a database restore operation is being attempted, and table space incremental backup images have been produced, the table space images must be restored in the chronological order of their backup time stamps.

Parallel Recovery

DB2 now uses multiple agents to perform both crash recovery and database rollforward recovery. You can expect better performance during these operations, particularly on symmetric multi-processor (SMP) machines; using multiple agents during database recovery takes advantage of the extra CPUs that are available on SMP machines.

The new agent type introduced by this enhancement is db2agnsc. DB2 chooses the number of agents to be used for database recovery based on the number of CPUs on the machine. For SMP machines, the number of agents used is (number of CPUs + 1). On a machine with a single CPU, three agents are used for more efficient reading of logs, processing of log records, and prefetching of data pages.

DB2 distributes log records to these agents so that they can be reapplied concurrently, where appropriate. The processing of log records is parallelized at the page level (log records on the same data page are processed by the same agent); therefore, performance is enhanced, even if all the work was done on one table.

Backing Up to Named Pipes

Support is now available for database backup to (and database restore from) local named pipes on UNIX based systems. Both the writer and the reader of the named pipe must be on the same machine. The pipe must exist and be located on a local file system. Because the named pipe is treated as a local device, there is no need to specify that the target is a named pipe. Following is an AIX example:

1. Create a named pipe:
mkfifo /u/dbuser/mypipe
2. Use this pipe as the target for a database backup operation:
db2 backup db sample to /u/dbuser/mypipe

3. Restore the database:

```
db2 restore db sample into mynewdb from /u/dbuser/mypipe
```

Backup from Split Image

DB2 now supports a full offline database backup on the split mirrored copy of a database. Online backup is not supported and is not necessary because the database, which is in rollforward pending state, is unavailable. When a split mirrored backup image is restored, it must be rolled forward because there may have been active transactions when the split occurred.

Note: For DB2 Version 7.1 FixPak 3 and DB2 Version 7.2 this support will be limited to databases that contain DMS table spaces only. If an attempt is made to back up a database after a split and the database contains any SMS table spaces, the backup will fail.

Once a database has been split, the **db2inidb** utility must be used to specify one of the following options:

- Snapshot. This initiates crash recovery, making the database consistent. A new log chain starts, and the database will not be able to roll forward through any of the logs from the original database. The database is available for any operation, including backup.
- Standby. This places the database in rollforward pending state. Crash recovery is *not* performed, and the database remains inconsistent.
- Mirror. This causes a mirrored copy of the database to replace the original database. The database is placed in rollforward pending state, and the WRITE SUSPEND state is turned off. Crash recovery is *not* performed, and the database remains inconsistent.

Following are some usage scenarios:

- Making a database clone.

The objective here is to have a read-only clone of the primary database that can be used, for example, to create reports. To do this, follow these steps:

1. Suspend I/O on the primary system:

```
db2 set write suspend for database
```

2. Split the mirror. Use operating system level commands to split the mirror from the primary database.

3. Resume I/O on the primary system:

```
db2 set write resume for database
```

The database on the primary system should now be back to a normal state.

4. Mount the split mirrors of the database to another host.

5. Start the instance:

```
db2start
```

6. Start DB2 crash recovery:

```
db2inidb <db_name> as snapshot
```

You can also use this process for an offline backup, but if restored on the primary system, this backup cannot be used to roll forward, because the log chain will not match.

- Using the split mirror as a standby database.

The idea here is that the mirrored (standby) database is continually rolling forward through the logs, and even new logs that are being created by the primary database are continually fetched from the primary system. To use the split mirror as a standby database, follow these steps:

1. Suspend I/O on the primary system:

```
db2 set write suspend for database
```

2. Split the mirror. Use operating system level commands to split the mirror from the primary database.

3. Resume I/O on the primary system:

```
db2 set write resume for database
```

The database on the primary system should now be back to a normal state.

4. Mount the split mirrors of the database to another host.

5. Remove the suspend write state and put the mirrored database in rollforward pending state.

```
db2inidb <db_name> as standby
```

6. Copy logs. Set up a user exit program to retrieve log files from the primary system's archive location, so that the latest logs will be available for this mirrored database.

7. Rollforward the mirror to the end of the logs.

```
db2 rollforward db <db_name> to end of logs
```

8. Repeat the process from step 6 until the primary database is down.

- Using the split mirror to recover the primary system.

The following procedure describes how to use the mirrored system as a backup image to restore the primary system:

1. Copy over. Use operating system commands to copy the mirrored data and logs on top of the primary system.

2. Start the instance:

```
db2start
```

3. Put the restored mirror in rollforward pending state and roll the mirror forward to the end of the logs:

```
db2inidb <dbname> as mirror
```

- Taking a backup without performing crash recovery.

Performing an offline backup on the split mirror without performing crash recovery means that you can restore this backup image on top of the primary system. To do this, follow these steps:

1. Suspend I/O on the primary system:

```
db2 set write suspend for database
```

2. Split the mirror. Use operating system level commands to split the mirror from the primary database.

3. Resume I/O on the primary system:

```
db2 set write resume for database
```

The database on the primary system should now be back to a normal state.

4. Mount the split mirrors of the database to another host.

5. Start the instance:

```
db2start
```

6. Put the mirrored database in rollforward pending state:

```
db2inidb <db_name> as standby
```

7. Invoke a database backup operation:

```
db2 backup database <db_name>
```

This results in an implicit database connection, but does not initiate DB2 crash recovery.

On Demand Log Archive

DB2 now supports the closing (and, if the user exit option is enabled, the archiving) of the active log for a recoverable database at any time. This allows you to collect a complete set of log files up to a known point, and then to use these log files to update a standby database.

註: On demand log archiving does not guarantee the log files will be archived immediately; it truncates the log file and issues an archive request, but it is still subject to delays associated with the user exit program

You can initiate on demand log archiving by invoking the new DB2 ARCHIVE LOG command, or by calling the new **db2ArchiveLog** API.

Log Mirroring

In Chapter 8, 『Recovering a Database,』 the following new section on using the suspended I/O function is to be added:

DB2 now supports log mirroring at the database level. Mirroring log files helps protect a database from:

- Accidental deletion of an active log
- Data corruption caused by hardware failure

If you are concerned that your active logs may be damaged (as a result of a disk crash), you should consider using a new DB2 registry variable, DB2_NEWLOGPATH2, to specify a secondary path for the database to manage copies of the active log, mirroring the volumes on which the logs are stored.

The DB2_NEWLOGPATH2 registry variable allows the database to write an identical second copy of log files to a different path. It is recommended that you place the secondary log path on a physically separate disk (preferably one that is also on a different disk controller). That way, the disk controller cannot be a single point of failure.

Note: Because Windows NT and OS/2 do not allow "mounting" a device under an arbitrary path name, it is not possible (on these platforms) to specify a secondary path on a separate device.

DB2_NEWLOGPATH2 can be enabled (set to 1) or disabled (set to 0). The default value is zero. If this variable is set to 1, the secondary path name is the current value of the LOGPATH variable concatenated with the character 2. For example, in an SMP environment, if LOGPATH is /u/dbuser/sqllogdir/logpath, the secondary log path will be /u/dbuser/sqllogdir/logpath2. In an MPP environment, if LOGPATH is /u/dbuser/sqllogdir/logpath, DB2 will append the node indicator to the path and use /u/dbuser/sqllogdir/logpath/NODE0000 as the primary log path. In this case, the secondary log path will be /u/dbuser/sqllogdir/logpath2/NODE0000.

When DB2_NEWLOGPATH2 is first enabled, it will not actually be used until the current log file is completed on the next database startup. This is similar to how NEWLOGPATH is currently used.

If there is an error writing to either the primary or secondary log path, the database will mark the failing path as "bad", write a message to the db2diag.log file, and write subsequent log records to the remaining "good" log path only. DB2 will not attempt to use the "bad" path again until the current log file is completed. When DB2 needs to open the next log file, it will verify that this path is valid, and if so, will begin to use it. If not, DB2 will not attempt to use the path again until the next log file is accessed for the first time. There is no attempt to synchronize the log paths, but DB2 keeps information about access errors that occur, so that the correct paths are used when log files are archived. If a failure occurs while writing to the remaining "good" path, the database abends.

Cross Platform Backup and Restore Support on Sun Solaris and HP

Support is now available for cross platform backup and restore support between Sun Solaris and HP. When you transfer the backup image between systems, you must transfer it in binary mode. On the target system, the database must be created with the same code page/territory as the system on which the original database was created.

DB2 Data Links Manager Considerations/Backup Utility Considerations

Replace the second paragraph in this section with:

When files are linked, the Data Links servers schedule them to be copied asynchronously to an archive server such as ADSM, or to disk. When the backup utility runs, DB2 ensures that all files scheduled for copying have been copied. At the beginning of backup processing, DB2 contacts all Data Links servers that are specified in the DB2 configuration file. If a Data Links server has one or more linked files and is not running, or stops running during the backup operation, the backup will not contain complete DATALINK information. The backup operation will complete successfully. Before the Data Links server can be marked as available to the database again, backup processing for all outstanding backups must complete successfully. If a backup is initiated when there are already twice the value of num_db_backups (see below) outstanding backups waiting to be completed on the Data Links server, the backup operation will fail. That Data Links server must be restarted and the outstanding backups completed before additional backups are allowed.

DB2 Data Links Manager Considerations/Restore and Rollforward Utility Considerations

Replace paragraphs beginning with:

When you restore a database or table space and do not specify the WITHOUT DATALINK...

and

When you restore a database or table space and you do specify the WITHOUT DATALINK option...

with:

When you restore a database or table space, the following conditions must be satisfied for the restore operation to succeed:

- o If any Data Links Server recorded in the backup file is not running, the restore operation will still complete successfully.

Tables with DATALINK column information that are affected by the missing Data Links server will be put into datalink reconcile pending state after the restore operation (or the rollforward operation, if used) completes. Before the Data Links servers can be marked as available to the database again, this restore processing must complete successfully.

- o If any Data Links Server recorded in the backup file stops running during the restore operation, the restore operation will fail. The restore can be restarted with the Data Links Server down (see above).
- o If a previous database restore operation is still incomplete on any Data Links server, subsequent database or table space restore operations will fail until those Data Links servers are restarted, and the incomplete restore is completed.
- o Information about all DATALINK columns that are recorded in the backup file must exist in the appropriate Data Links servers' registration tables.

If all the information about the DATALINK columns is not recorded in the registration tables, the table with the missing DATALINK column information is put into datalink reconcile not possible state after the restore operation (or the roll-forward operation, if used) completes.

If the backup is not recorded in the registration tables, it may mean that the backup file that is provided is earlier than the value for num_db_backups and has already been "garbage collected". This means that the archived files from this earlier backup have been removed and cannot be restored. All tables that have DATALINK columns are put into datalink reconcile pending state.

If the backup is not recorded in the registration tables, it may mean that backup processing has not yet been completed because the Data Links server is not running. All tables that have DATALINK columns are put into datalink reconcile pending state. When the Data Links server is restarted, backup processing will be completed before restore processing.

The table remains available to users, but the values in the DATALINK columns may not reference the files accurately (for example, a file may not be found that matches a value for the DATALINK column). If you do not want this behavior, you can put the table into check pending state by issuing the "SET CONSTRAINTS for tablename TO DATALINK RECONCILE PENDING" statement.

If, after a restore operation, you have a table in datalink reconcile not possible state, you can fix the DATALINK column data in one of the ways suggested under "Removing a Table from the Datalink_Reconcile_Not_Possible State".

The note at the bottom of the first paragraph remains the same.

Add the following at the end of this section:

It is strongly recommended that the datalink.cfg file be archived to cover certain unusual recovery cases, since the datalink.cfg file in the database backup image only reflects the datalink.cfg as of the backup time. Having the latest datalink.cfg file is required to cover all recovery cases. Therefore, the datalink.cfg file must be backed up after every ADD DATALINKS MANAGER or DROP DATALINKS MANAGER command invocation. This would help to retrieve the latest datalink.cfg file, if the latest datalink.cfg file is not available on disk.

If the latest datalink.cfg file is not available on disk, replace the existing datalink.cfg file (restored from a backup image) with the latest datalink.cfg file that was archived before running a rollforward operation. Do this after the database is restored.

Restoring Databases from an Offline Backup without Rolling Forward

You can only restore without rolling forward at the database level, not the table space level. To restore a database without rolling forward, you can either restore a nonrecoverable database (that is, a database that uses circular logging), or specify the WITHOUT ROLLING FORWARD parameter on the RESTORE DATABASE command.

If you use the restore utility with the WITHOUT DATALINK option, all tables with DATALINK columns are placed in datalink reconcile pending (DRP) state, and no reconciliation is performed with the Data Links servers during the restore operation.

If you do not use the WITHOUT DATALINK option, and a Data Links server recorded in the backup file is no longer defined to the database (that is, it has been dropped using the DROP DATALINKS MANAGER command), tables that contain DATALINK data referencing the dropped Data Links server are put in DRP state by the restore utility.

If you do not use the WITHOUT DATALINK option, all the Data Links servers are available, and all information about the DATALINK columns is fully recorded in the registration tables, the following occurs for each Data Links server recorded in the backup file:

- All files linked after the backup image that was used for the database restore operation are marked as unlinked (because they are not recorded in the backup image as being linked).
- All files that were unlinked after the backup image, but that were linked before the backup image was taken, are marked as linked (because they are recorded in the backup image as being linked). If the file was subsequently linked to another table in another database, the restored table is put into the datalink reconcile pending state.

Note: The above cannot be done if the backup image that was used for the database restore operation was taken when at least one Data Links server was not running, since the DATALINK information in the backup is incomplete. The above is also not done if the backup image that was used for the database restore operation was taken after

a database restore with or without rollforward. In both cases, all tables with DATALINK columns are placed in datalink reconcile pending state, and no reconciliation is performed with the Data Links servers during the restore operation.

Restoring Databases and Table Spaces, and Rolling Forward to the End of the Logs

If you restore, then roll forward the database or table space to the end of the logs (meaning that all logs are provided), a reconciliation check is not required unless at least one of the Data Links servers recorded in the backup file is not running during the restore operation. If you are not sure whether all the logs were provided for the roll-forward operation, or think that you may need to reconcile DATALINK values, do the following:

1. Issue the SQL statement for the table (or tables) involved:

```
SET CONSTRAINTS FOR tablename TO DATALINK RECONCILE PENDING
```

This puts the table into datalink reconcile pending state and check pending state.

2. If you do not want a table to be in check pending state, issue the following SQL statement:

```
SET CONSTRAINTS FOR tablename IMMEDIATE CHECKED
```

This takes the table out of check pending state, but leaves it in datalink reconcile pending state. You must use the reconcile utility to take the table out of this state.

It may happen that the backup file contains DATALINK data that refers to a DB2 Data Links Manager (that is, a DB2 Data Links Manager was registered to the database when the backup was taken) that has been dropped from the database. For each table space being rolled forward that contains at least one table with DATALINK data referencing the dropped DB2 Data Links Manager, all tables are put in DRP state by the rollforward utility.

DB2 Data Links Manager and Recovery Interactions

The following table shows the different types of recovery that you can perform, the DB2 Data Links Manager processing that occurs during restore and roll-forward processing, and whether you need to run the Reconcile utility after the recovery is complete:

Type of Recovery	DB2 Data Links Manager Processing during Restore	DB2 Data Links Manager Processing during Rollforward	Reconcile
Non-recoverable database (<i>logretain=NO</i>)			
Database restore of a complete backup, all Data Links Servers up	Fast reconcile is performed	N/A	Can be optionally run if problem with file links is suspected
Database restore using WITHOUT DATALINK option	Tables put in <i>Datalink_Reconcile _Pending</i> state	N/A	Required

Type of Recovery	DB2 Data Links Manager Processing during Restore	DB2 Data Links Manager Processing during Rollforward	Reconcile
Database restore of a complete backup, at least one Data Links server down	Fast reconcile is performed only on those tables in table spaces that do not have links to a Data Links server that is down, other tables put in Datalink_Reconcile_Pending state	NA	Required for tables in table spaces with links to the Data Links server that is down
Database restore of an incomplete backup, all Data Links servers up	Fast reconcile is not performed, all tables with DATALINK columns put in Datalink_Reconcile_Pending state	NA	Required
Recoverable database (<i>logretain=YES</i>)			
Database restore using WITHOUT ROLLING FORWARD option, using a complete backup, all Data Links servers up	Fast reconcile is performed	N/A	Optional
Database restore using WITHOUT ROLLING FORWARD and WITHOUT DATALINK options, using a complete or incomplete backup, Data Links servers up or down	Tables put in <i>Datalink_Reconcile _Pending</i> state	N/A	Required
Database restore using WITHOUT ROLLING FORWARD option, using a complete backup, at least one Data Links server down	Fast reconcile is performed only on those tables in table spaces that do not have links to the Data Links servers that are down, other tables put in Datalink_Reconcile_Pending state	N/A	Required on tables in table spaces with links to the Data Links servers that are down
Database restore using WITHOUT ROLLING FORWARD option, using an incomplete backup, Data Links servers up or down	Fast reconcile is not performed, all tables with DATALINK columns put into Datalink_Reconcile_Pending state	N/A	Required
Database restore and roll forward to end of logs, using a complete backup, all Data Links servers up	No action	No action	Optional

Type of Recovery	DB2 Data Links Manager Processing during Restore	DB2 Data Links Manager Processing during Rollforward	Reconcile
Database restore and roll forward to end of logs, using a complete backup, at least one Data Links server down during roll forward processing	No action	No action	Optional
Database restore and roll forward to end of logs, using a complete or an incomplete backup, any Data Links server down during restore	No action	All tables with DATALINK columns put into Datalink_Reconcile_Pending state	Required for all tables with DATALINK columns
Database restore and roll forward to end of logs, using an incomplete backup, all Data Links servers up during restore	No action	No action	Optional
Database restore and roll forward to end of logs, using a complete or an incomplete backup, all Data Links servers up, backup unknown at any Data Links server	No action	All tables in table spaces with links to a Data Links server where the backup is unknown put in Datalink_Reconcile_Pending state	Required
Table space restore and roll forward to end of logs, using a complete backup, all Data Links servers up	No action	No action	Optional
Table space restore and roll forward to end of logs, using a complete backup, at least one Data Links server down during roll forward processing	No action	No action	Optional
Table space restore and roll forward to end of logs, using a complete or an incomplete backup, any Data Links server down during restore processing	No action	All tables in table spaces with links to any Data Links server that is down put into Datalink_Reconcile_Pending state	Required for tables in table spaces with links to any Data Links server that is down
Table space restore and roll forward to end of logs, using an incomplete backup, all Data Links servers up	No action	No action	Optional

Type of Recovery	DB2 Data Links Manager Processing during Restore	DB2 Data Links Manager Processing during Rollforward	Reconcile
Database restore and roll forward to a point in time, using a complete or an incomplete backup, Data Links servers up or down during restore and/or roll forward processing	No action	Tables put in <i>Datalink_Reconcile _Pending</i> state	Required
Table space restore and roll forward to a point in time, using a complete or an incomplete backup, Data Links servers up or down during restore and/or rollforward processing	No action	Tables put in <i>Datalink_Reconcile _Pending</i> state	Required
Database restore to a different database name, alias, hostname, or instance with no roll forward 1	Tables put in <i>Datalink_Reconcile _Not_Possible</i> state	N/A	Optional, but tables in <i>Datalink_Reconcile _Not_Possible</i> state must be manually fixed
Database restore to a different database name, alias, hostname or instance, and roll forward	No action	Tables put in <i>Datalink_Reconcile _Not_Possible</i> state	Optional, but tables in <i>Datalink_Reconcile _Not_Possible</i> state must be manually fixed
Database restore from an unusable backup (image has been garbage-collected on the Data Links server) with no roll forward 1, with or without WITHOUT DATALINK option	Tables put in <i>Datalink_Reconcile _Pending</i> state	No action	Required
Database restore from an unusable backup (image has been garbage-collected on the Data Links server), and roll forward, with or without WITHOUT DATALINK option	No action	Tables put in <i>Datalink_Reconcile _Pending</i> state	Required
Table space restore from an unusable backup (image has been garbage-collected on the Data Links server), and roll forward	No action	Tables put in <i>Datalink_Reconcile _Pending</i> state	Required

註:

1. A restore using an offline backup and the WITHOUT ROLLING FORWARD option (*logretain* is on), or a restore using an offline backup (*logretain* is off).

2. A *complete* backup is a backup taken when all required Data Links servers were running. An *incomplete* backup is a backup taken when at least one required Data Links server was not running.
3. Fast reconcile processing will not be performed if the backup image that was used for the database restore operation was taken after a database restore, with or without rollforward. In this case, all tables with DATALINK columns are put in *Datalink_Reconcile_Pending* state.

Detection of Situations that Require Reconciliation

Following are some situations in which you may need to run the reconcile utility:

- The entire database is restored and rolled forward to a point in time. Because the entire database is rolled forward to a committed transaction, no tables will be in check pending state (due to referential constraints or check constraints). All data in the database is brought to a consistent state. The DATALINK columns, however, may not be synchronized with the metadata in the DB2 Data Links Manager, and reconciliation is required.

In this situation, tables with DATALINK data will already be in DRP state. You should invoke the reconcile utility for each of these tables.

- A particular Data Links server running the DB2 Data Links Manager loses track of its metadata. This can occur for different reasons. For example:
 - The Data Links server was cold started.
 - The Data Links server metadata was restored to a back-level state.

In some situations, such as during SQL UPDATES and DELETES, DB2 may be able to detect a problem with the metadata in a Data Links server. In these situations, the SQL statement would fail. You would put the table in DRP state by using the SET CONSTRAINTS statement, then run the reconcile utility on that table.

- A file system is not available (for example, because of a disk crash) and is not restored to the current state. In this situation, files may be missing.
- A DB2 Data Links Manager is dropped from a database, and there are DATALINK FILE LINK CONTROL values referencing that DB2 Data Links Manager. You should run the reconcile utility on such tables.

Appendix C. User Exit for Database Recovery

Under the section 『Archive and Retrieve Considerations』, the following paragraph is no longer true and should be removed from the list:

A user exit may be interrupted if a remote client loses its connection to the DB2 server. That is, while handling the archiving of logs through a user exit, one of the other SNA-connected clients dies or powers off resulting in a signal (SIGUSR1) being sent to the server. The server passes the signal to the user exit causing an interrupt. The user exit program can be modified to check for an interrupt and then continue.

The Error Handling section has a Notes list that should replace the contents of Note 3 with the following information:

- User exit program requests are suspended for five minutes. During this time, all requests are ignored including the log file request that caused the return code.

Following the five minute suspension in processing requests, the next request is processed. If no error occurs with the processing of this request, then processing of new user exit program requests continues and DB2 will reissue the archive request for the log files that either failed to archive previously, or were suspended. If a return code of greater than 8 is generated during the retry, requests are suspended for an additional five minutes. The five minute suspensions continue until the problem is corrected or the database is stopped and restarted.

Once all applications disconnect from the database and the database is reopened, DB2 will issue the archive request for any log file that might not have been successfully archived in the previous use of the database.

If the user exit program fails to archive log files, your disk can be filled with log files and performance may be degraded because of extra work to format these log files. Once the disk becomes full, the database manager will not accept further application requests for database changes.

If the user exit program was called to retrieve log files, roll-forward recovery is suspended but not stopped unless a stop was specified in the ROLLFORWARD DATABASE utility. If a stop was not specified, you can correct the problem and resume recovery.

Appendix D. Issuing Commands to Multiple Database Partition Servers

At the bottom of the section 『Specifying the Command to Run』, add the following:

When you run any korn-shell shell-script which contains logic to read from stdin in the background, you should explicitly redirect stdin to a source where the process can read without getting stopped on the terminal (SIGTTIN message). To redirect stdin, you can run a script with the following form:

```
shell_script </dev/null &
```

if there is no input to be supplied.

In a similar way, you should always specify </dev/null when running db2_all in the background. For example:

```
db2_all ";run_this_command" </dev/null &
```

By doing this you can redirect stdin and avoid getting stopped on the terminal.

An alternative to this method, when you are not concerned about output from the remote command, is to use the 『daemonize』 option in the db2_all prefix:

```
db2_all ";daemonize_this_command" &
```

Appendix I. High Speed Inter-node Communications

The following section has been updated:

Enabling DB2 to Run Using VI

Detailed installation information is found in *DB2 Enterprise - Extended Edition for Windows 快速入門*.

After completing the installation of DB2 as documented in *DB2 Enterprise - Extended Edition for Windows 快速入門*, set the following DB2 registry variables and carry out the following tasks on each database partition server in the instance:

- Set DB2_VI_ENABLE=ON

Use the **db2set** command to modify the value for the registry variable. Use the **db2_all** command to run the **db2set** command on all database partition servers in the instance. You must be logged on with a user account that is a member of the Administrators group to run the **db2_all** command.

In the following example, the ; character is placed inside the double quotation marks to allow the request to run concurrently on all the database partition servers in the instance:

```
db2_all ";db2set DB2_VI_ENABLE=ON"
```

For more information about the **db2_all** command, see "Issuing Commands to Multiple Database Partition Servers" in the *Administration Guide: Implementation*.

- Set DB2_VI_DEVICE=nic0

For example:

```
db2_all ";db2set DB2_VI_DEVICE=nic0"
```

註: With Synfinity Interconnect, this variable should be set DB2_VI_DEVICE=VINIC. The device name (VINIC) must be in upper case.

- Set DB2_VI_VIPL=vipl.dll

For example:

```
db2_all ";db2set DB2_VI_VIPL=vipl.dll"
```

Note: The value used in the example is the default for the registry variable. For more information on the registry variables, see *Administration Guide: Performance*.

- Enter db2start on the MPP instance.
- Review the db2diag.log file. There should be one message for each partition stating that 『VI is enabled.』
- Fast Communications Manager (FCM) configuration parameters may need to be updated. Should you encounter a problem as a result of resource constraints involving FCM, you should raise the values of the FCM configuration parameters. If you are moving from another high speed interconnect environment where you have increased the values for the FCM configuration parameters, you may need to lower these values. Also, on Windows NT, you may be required to set the DB2NTMEMSIZE registry variable to override the DB2 defaults. Refer to *Administration Guide: Performance* for more information on the registry variables.

Administration Guide: Performance (無中文版)

Chapter 3. Application Considerations

Specifying the Isolation Level

Within the section titled 『Specifying the Isolation Level』, add the following paragraphs:

In addition to setting the isolation level at the package-level when preparing or binding an application, you can set an isolation level at the statement-level. A statement-level isolation level is specified using the WITH-clause.

The following SQL statements support statement-level isolation:

- SELECT statement
- SELECT INTO
- Searched DELETE
- INSERT
- Searched UPDATE
- DECLARE CURSOR

There are some conditions associated with the use of statement-level isolation:

- The WITH-clause cannot be used on subqueries
- The WITH UR option only applies to read-only operations. If it is used in other situations, the statement is automatically changed from 『UR』 to 『CS』.
- The default isolation level for a statement is the isolation level of the package in which the statement is bound.
- The statement-level isolation level overrides the isolation level specified for the package in which the statement appears.

Adjusting the Optimization Class

Within the section titled 『Adjusting the Optimization Class』, replace the current level 2 information with the following:

- 2 -** This class directs the optimizer to use a degree of optimization which significantly improves upon that of class 1, while keeping the compilation cost significantly lower than classes 3 and above for complex queries. In particular:
- All available statistics, including both frequency and quantile non-uniform distribution statistics, are utilized.
 - All of the query rewrite rules are applied including the routing of queries to summary tables, except computationally intensive rules which are applicable only in very rare cases.
 - Greedy join enumeration is used.
 - A wide range of access methods are considered, including list prefetch and summary table routing.
 - The star join strategy is considered, if applicable.

Optimization class 2 is quite similar to class 5 except that it uses Greedy join enumeration rather than Dynamic Programming. This class has the most optimization of all the optimization classes that use the Greedy join enumeration algorithm, which considers fewer alternatives for complex queries, and therefore consumes less compilation time than classes 3 and above. It is therefore recommended for very complex queries in a decision support or online analytic processing (OLAP) environment. In such cases, there is a good chance the same query is executed infrequently, so that its access plan is unlikely to remain in the cache until the next occurrence of the query.

Dynamic Compound Statements

This new section should be added after the section titled 『Compound SQL』 :

A dynamic compound statement groups other SQL statements together into an executable block. Within the dynamic compound statement you can declare SQL variables, declare conditions associated with SQLSTATEs, and have one or more SQL procedural statements. If an error occurs in the dynamic compound statement, all prior SQL statements are rolled back and the remaining SQL statements in the dynamic compound statement are not processed.

The dynamic compound statement can be embedded in a trigger, SQL function, SQL method, or issued through the use of dynamic SQL statements. This executable statement can be dynamically prepared. No privileges are required to invoke the statement but the authorization ID associated with the statement must have the necessary privileges to invoke the embedded SQL statements within the compound statement.

Variables are present in the sub-statements in the variable declaration. Conditions are present in the sub-statements based on the SQLSTATE values of the condition declaration. Dynamic compound statements are compiled by DB2 as a single statement. This statement can be used effectively for short scripts involving little control flow logic but significant data flow. For larger constructs with nested complex control flow, you should consider using SQL procedures.

There are several control flow logic statements that can be used within the dynamic compound statement. These include: the FOR statement, the IF statement, the ITERATE statement, and the WHILE statement. Details about these statements, and the other supported statements, are found in the *SQL Reference*.

Chapter 4. Environmental Considerations

Following the section titled 『Using the Index Advisor』 , add the following section:

Using Larger Index Keys

It is possible to allow columns with a length greater than 255 bytes to be specified as part of an index key. The DB2_INDEX_2BYTEVARLEN registry variable allows for the use of 2 bytes instead of 1 to store the length of an index key.

There are several SQL statements affected by changes to the registry variable. They are:

- CREATE TABLE. Primary, foreign, and unique keys with variable key parts can have a size greater than 255 bytes.
 - CREATE INDEX. All indexes, including unique indexes and include columns, having variable key parts can have a size greater than 255 bytes.
 - ALTER TABLE. Primary, foreign, and unique keys with variable key parts can have a size greater than 255 bytes. All indexes, including unique indexes and include columns, having variable key parts can have a size greater than 255 bytes.
- The foreign key restriction to 255 bytes is removed no matter what the value of the registry variable. The truth-condition for the primary key corresponding to the foreign key enforces any restriction or limit.

To convert existing indexes to use larger index keys: drop the indexes, set the DB2_INDEX_2BYTEVARLEN registry variable to ON, and then recreate the indexes (using the larger columns).

For more information on the SQL statements, including syntax descriptions, refer to the *SQL Reference*.

Chapter 5. System Catalog Statistics

Collecting and Using Distribution Statistics

In the subsection called 『Example of Impact on Equality Predicates』, there is a discussion of a predicate $C \leq 10$. The error is stated as being -86%. This is incorrect. The sentence at the end of the paragraph should read:

Assuming a uniform data distribution and using formula (1), the number of rows that satisfy the predicate is estimated as 1, an error of -87.5%.

In the subsection called 『Example of Impact on Equality Predicates』, there is a discussion of a predicate $C > 8.5$ AND $C \leq 10$. The estimate of the r_2 value using linear interpolation must be changed to the following:

$$r_2 = \frac{10 - 8.5}{100 - 8.5} \times (\text{number of rows with value } > 8.5 \text{ and } \leq 100.0)$$

$$r_2 = \frac{10 - 8.5}{100 - 8.5} \times (10 - 7)$$

$$r_2 = \frac{1.5}{91.5} \times (3)$$

$$r_2 = 0$$

The paragraph following this new example must also be modified to read as follows:

The final estimate is $r_1 + r_2 = 7$, and the error is only -12.5%.

Rules for Updating Catalog Statistics

Within the section titled 『Rules for Updating Column Statistics』, the last bulleted-list item in the first list item should be replaced by the following:

HIGH2KEY must be greater than LOW2KEY whenever there are more than 3 distinct values in the corresponding column. In the case of 3 or less distinct values in the column, HIGH2KEY can be equal to LOW2KEY.

Sub-element Statistics

In FixPak 1, an option is provided to collect and use sub-element statistics. These are statistics about the content of data in columns when the data has a structure in the form of a series of sub-fields or sub-elements delimited by blanks.

For example, suppose a database contains a table DOCUMENTS in which each row describes a document, and suppose that in DOCUMENTS there is a column called KEYWORDS containing a list of relevant keywords relating to this document for text retrieval purposes. The values in KEYWORDS might be as follows:

```
'database simulation analytical business intelligence'  
'simulation model fruitfly reproduction temperature'  
'forestry spruce soil erosion rainfall'  
'forest temperature soil precipitation fire'
```

In this example, each column value consists of 5 sub-elements, each of which is a word (the keyword), separated from the others by one blank.

For queries that specify LIKE predicates on such columns using the % match_all character:

```
SELECT .... FROM DOCUMENTS WHERE KEYWORDS LIKE '%simulation%'
```

it is often beneficial for the optimizer to know some basic statistics about the sub-element structure of the column, namely:

SUB_COUNT

The average number of sub-elements.

SUB_DELIM_LENGTH

The average length of each delimiter separating each sub-element, where a delimiter, in this context, is one or more consecutive blank characters.

In the KEYWORDS column example, SUB_COUNT is 5, and SUB_DELIM_LENGTH is 1, because each delimiter is a single blank character.

In FixPak 1, the system administrator controls the collection and use of these statistics by means of an extension to the DB2_LIKE_VARCHAR registry variable. This registry variable affects how the DB2 UDB optimizer deals with a predicate of the form:

```
COLUMN LIKE '%xxxxxx'
```

where xxxxxx is any string of characters; that is, any LIKE predicate whose search value starts with a % character. (It may or may not end with a % character). These are referred to as "wildcard LIKE predicates" below. For all predicates, the optimizer has to estimate how many rows match the predicate. For wildcard LIKE predicates, the optimizer assumes that the COLUMN being matched has a structure of a series of elements concatenated together to form the entire column, and estimates the length of each element based on the length of the string, excluding leading and trailing % characters. The new syntax is:

```
db2set DB2_LIKE_VARCHAR=[Y|N|S|num1][,Y|N|num2]
```

where

- the first term (preceding the comma) means the following, but only for columns that do not have positive sub-element statistics

S	Use the algorithm as used in DB2 Version 2.
N	Use a fixed-length sub-element algorithm.
Y (default)	Use a variable-length sub-element algorithm with a default value for the algorithm parameter.
num1	Use a variable-length sub-element algorithm, and use num1 as the algorithm parameter.

- the second term (following the comma) means:

N (default)	Do not collect or use sub-element statistics.
Y	Collect sub-element statistics. Use a variable-length sub-element algorithm that uses those statistics, together with a default value for the algorithm parameter in the case of columns with positive sub-element statistics.
num2	Collect sub-element statistics. Use a variable-length sub-element algorithm that uses those statistics, together with num2 as the algorithm parameter in the case of columns with positive sub-element statistics.

If the value of DB2_LIKE_VARCHAR contains only the first term, no sub-element statistics are collected, and any that have previously been collected are ignored. The value specified affects how the optimizer calculates the selectivity of wildcard LIKE predicates in the same way as before; that is:

- If the value is S, the optimizer uses the same algorithm as was used in DB2 Version 2, which does not presume the sub-element model.
- If the value is N, the optimizer uses an algorithm that presumes the sub-element model, and assumes that the COLUMN is of a fixed length, even if it is defined as variable length.
- If the value is Y (the default) or a floating point constant, the optimizer uses an algorithm that presumes the sub-element model and recognizes that the COLUMN is of variable length, if so defined. It also infers sub-element statistics from the query itself, rather than from the data. This algorithm involves a parameter (the "algorithm parameter") that specifies how much longer the element is than the string enclosed by the % characters.
- If the value is Y, the optimizer uses a default value of 1.9 for the algorithm parameter.
- If the value is a floating point constant, the optimizer uses the specified value for the algorithm parameter. This constant must lie within the range of 0 to 6.2.

If the value of DB2_LIKE_VARCHAR contains two terms, and the second is Y or a floating point constant, sub-element statistics on single-byte character set string columns of type CHAR, VARCHAR, GRAPHIC, or VARGRAPHIC are collected during a RUNSTATS operation and used during compilation of queries involving wildcard LIKE predicates. The optimizer uses an algorithm that presumes the sub-element model and uses the SUB_COUNT and SUB_DELIM_LENGTH statistics, as well as an algorithm parameter, to calculate the selectivity of the predicate. The algorithm parameter is specified in the same way that the inferential algorithm is specified, that is:

- If the value is Y, the optimizer uses a default value of 1.9 for the algorithm parameter.
- If the value is a floating point constant, the optimizer uses the specified value for the algorithm parameter. This constant must lie within the range of 0 to 6.2.

If, during compilation, the optimizer finds that sub-element statistics have not been collected on the column involved in the query, it will use the "inferential" sub-element algorithm; that is, the one used when only the first term of DB2_LIKE_VARCHAR is specified. Thus, in order for the sub-element statistics to be used by the optimizer, the second term of DB2_LIKE_VARCHAR must be set both during RUNSTATS and compilation.

The values of the sub-element statistics can be viewed by querying SYSIBM.SYSCOLUMNS. For example:

```
select substr(NAME,1,16), SUB_COUNT, SUB_DELIM_LENGTH
       from sysibm.syscolumns where tname = 'DOCUMENTS'
```

The SUB_COUNT and SUB_DELIM_LENGTH columns are not present in the SYSSTAT.COLUMNS statistics view, and therefore cannot be updated.

註: RUNSTATS may take longer if this option is used. For example, RUNSTATS may take between 15 and 40% longer on a table with five character columns, if the DETAILED and DISTRIBUTION options are not used. If either the DETAILED or the DISTRIBUTION option is specified, the percentage overhead is less, even though the absolute amount of overhead is the same. If you are considering using this option, you should assess this overhead against improvements in query performance.

Chapter 6. Understanding the SQL Compiler

The following sections require changes:

Replicated Summary Tables

The following information will replace or be added to the existing information already in this section:

Replicated summary tables can be used to assist in the collocation of joins. For example, if you had a star schema where there is a large fact table spread across twenty nodes, then the joins between the fact table and the dimension tables are most efficient if these tables are collocated.

By placing all of the tables in the same nodegroup, at most there would one dimension table partitioned correctly for a collocated join. All other dimension tables would not be able to be used in a collocated join because the join column(s) on the fact table would not correspond to the fact table's partitioning key.

For example, you could have a table called FACT (C1, C2, C3, ...) partitioned on C1; and a table called DIM1 (C1, dim1a, dim1b, ...) partitioned on C1; and a table called DIM2 (C2, dim2a, dim2b, ...) partitioned on C2; and so on.

From this example, you could see that the join between FACT and DIM1 is perfect because the predicate DIM1.C1 = FACT.C1 would be collocated. Both of these tables are partitioned on column C1.

The join between DIM2 with the predicate WHERE DIM2.C2 = FACT.C2 cannot be collocated because FACT is partitioned on column C1 and not on column C2.

In this case, it would be good to replicate DIM2 in the fact table's nodegroup. In this way we can do the join locally on each partition.

Note: The replicated summary tables discussion here has to do with intra-database replication. Inter-database replication has to do with subscriptions, control tables, and data located in different databases and on different operating systems. If you are interested in inter-database replication refer to the *Replication Guide and Reference* for more information.

When creating a replicated summary table, the source table could be a single-node nodegroup table or a multi-node nodegroup table. In most cases, the table is small and can be placed in a single-node nodegroup. You may place a limit on the data to be replicated by specifying only a subset of the columns from the table, or by limiting the number of rows through the predicates used, or by using both methods when creating the replicated summary table.

Note: The data capture option is not required for replicated summary tables to function.

The replicated summary table could also be created in a multi-node nodegroup. The nodegroup is the same as the nodegroup in which you have placed your large tables. In this case, copies of the source table are created on all of the partitions of the nodegroup. Joins between a large fact table and the dimension tables have a better chance of being done locally in this environment rather than having to broadcast the source table to all partitions.

Indexes on replicated tables are not created automatically. Indexes are created and may be different from those identified in the source table.

Note: You cannot create unique indexes (or put on any constraints) on the replicated tables. This will prevent constraint violations that are not present on the source tables. These constraints are disallowed even if there is the same constraint on the source table.

After using the REFRESH statement, you should run RUNSTATS on the replicated table as you would any other table.

The replicated tables can be referenced directly within a query. However, you cannot use the NODENUMBER() predicate with a replicated table to see the table data on a particular partition.

To see if a created replicated summary table was used (given a query that referenced the source table), you can use the EXPLAIN facility. First, you would ensure the EXPLAIN tables existed. Then, you would create an explain plan for the SELECT statement you are interested in. Finally, you would use db2exfmt utility to format the EXPLAIN output.

The access plan chosen by the optimizer may or may not use the replicated summary table depending on the information that needs to be joined. Not using the replicated summary table could occur if the optimizer determined that it would be cheaper to broadcast the original source table to the other partitions in the nodegroup.

Data Access Concepts and Optimization

The section 『Multiple Index Access』 under 『Index Scan Concepts』 has changed.

Add the following information before the note at the end of the section:

To realize the performance benefits of dynamic bitmaps when scanning multiple indexes, it may be necessary to change the value of the sort heap size (*sortheap*) database configuration parameter, and the sort heap threshold (*sheapthres*) database manager configuration parameter.

Additional sort heap space is required when dynamic bitmaps are used in access plans. When *sheapthres* is set to be relatively close to *sortheap* (that is, less than a factor of two or three times per concurrent query), dynamic bitmaps with multiple index access must work with much less memory than the optimizer anticipated.

The solution is to increase the value of *sheapthres* relative to *sortheap*.

The section 『Search Strategies for Star Join』 under 『Predicate Terminology』 has changed.

Add the following information at the end of the section:

The dynamic bitmaps created and used as part of the Star Join technique uses sort heap memory. See Chapter 13, "Configuring DB2" in the *Administration Guide: Performance* manual for more information on the Sort Heap Size (*sortheap*) database configuration parameter.

Chapter 8. Operational Performance

Managing the Database Buffer Pool

Within the section titled 『Managing the Database Buffer Pool』, add the following information after the paragraph that begins 『When creating the buffer pool, by default the page size is 4 KB.』 :

When working with Windows 2000, buffer pool sizes up to 64 GB in size are supported less the size of DB2 and the operating system. (This assumes that DB2 is the primary product on the system.) This support is available through Microsoft Address Windowing Extensions (AWE).

Although AWE can be used with buffer pools of any size, if you require AWE use on larger buffer pools there are other recommended Windows products. Windows 2000 Advanced Server provides support for up to 8 GB of memory. Windows 2000 Data Center Server provides support for up to 64 GB of memory.

DB2 and Windows 2000 must be configured correctly to support AWE buffer pools. The buffer pool that will take advantage of AWE must exist in the database.

To have a 3 GB user space allocated, use the /3GB Windows 2000 boot option. This allows a larger AWE window size to be used. To enable access to more than 4 GB of memory via the AWE memory interface, use the /PAE Windows 2000 boot option. To

verify that you have the correct boot option selected, under Control, select System, then select 『Startup and Recovery』. From the drop-down list you can see the available boot options. If the boot option (/3GB or /PAE) you want is selected, then you are ready to proceed to the next task in setting up AWE support. If the option you want is not available for selection, you must add the option to the boot.ini file on the system drive. The boot.ini file contains a list of actions to be done when the operating system is started. Add /3GB, or /PAE, or both (separated by blanks) at the end of the list of existing parameters. Once you have saved this changed file, you can verify and select the correct boot option as mentioned above.

Windows 2000 also has to be modified to associate the 『lock pages in memory』 -right with the user under which DB2 is installed. To set the 『lock pages in memory』 -right, once you have logged on to Windows 2000 as the user who installed DB2, under the Start menu on Windows 2000 select the 『Administrative Tools』 folder, and then the 『Local Security Policy』 program. Under the local policies, you can select the user rights assignment for the 『lock pages in memory』 -right.

DB2 requires the setting of the DB2_AWE registry variable. To set this registry variable correctly, you will need to know the buffer pool ID of the buffer pool you wish to allow support of AWE. You also need to know the number of physical pages and the address window pages to allocate. The number of physical pages to allocate should be some value less than the total available physical pages. The actual number chosen will depend on your working environment. For example, if you have an environment where only DB2 and database applications are used on your system, then you can choose to have from one-half to one GB less than the total size of the physical pages as the value used with the DB2_AWE variable. If you have an environment where other non-database applications are using the system, then you will have to increase the value you subtract from the total to allow more physical pages for those other applications. The number used in the DB2_AWE registry variable is the number of physical pages to be used in support of AWE and for use by DB2. The upper limit on the address window pages is 1.5 GB, or 2.5 GB when the /3GB Windows 2000 boot option is in effect.

For information on setting the DB2 registry variable DB2_AWE, see the table of new and changed registry variables in 『Appendix A. DB2 Registry and Environment Variables』 later in this section.

Managing Multiple Database Buffer Pools

Within the section titled 『Managing Multiple Database Buffer Pools』, add the following paragraph after the paragraph that begins 『When working with your database design, you may have determined that tables with 8 KB page sizes are best.』 :

When working with Windows 2000, the DB2_AWE registry variable can be used to override the buffer pool size settings in the catalog and configuration files. Use of this registry variable allows buffer pool sizes of up to approximately 64 GB.

Within the same section, replace the paragraph just before the note with the following:

The reason for allowing the database manager to start with minimal-sized values is to allow you to connect to the database. You can then reconfigure the buffer pool sizes,

or perform other critical tasks, with the goal of restarting the database with correct buffer pool sizes. Do not consider operating the database for an extended time in such a state.

Within the section titled 『Reorganizing Catalogs and User Tables』, the last sentence (with a short list) in the paragraph that begins 『The REORG utility allows you to specify a temporary table space...』 can be replaced by:

Using the same table space to reorganize tables is faster but greater logging occurs and there must be enough space for the reorganized table. If you specify a temporary table space, it is generally recommended that you specify an SMS temporary table space. A DMS temporary table space is not recommended since you can only have one REORG in progress using this type of table space.

Within the section titled 『Extending Memory』, add the following paragraph after the third paragraph in this section:

When allocating Windows 2000 Address Windowing Extensions (AWE) buffer pools using the DB2_AWE registry variable, the extended storage cache cannot be used.

Chapter 9. Using the Governor

Within the section titled 『Creating the Governor Configuration File』, the first sentence in the first paragraph following the schedule action discussion should be replaced with:

If more than one rule applies to an application, all of the rules are applied. Depending on the rule and the limits being set, the action associated with the rule limit encountered first is the action that is first to be applied.

Chapter 13. Configuring DB2

The following parameters require changes:

Sort Heap Size (sortheap)

The 『Recommendation』 section has changed. The information here should now read:

When working with the sort heap, you should consider the following:

- Appropriate indexes can minimize the use of the sort heap.
- Hash join buffers and dynamic bitmaps (used for index ANDing and Star Joins) use sort heap memory. Increase the size of this parameter when these techniques are used.
- Increase the size of this parameter when frequent large sorts are required.
- ... (the rest of the items are unchanged)

Sort Heap Threshold (sheapthres)

The second last paragraph in the description of this parameter has changed. The paragraph should now read:

Examples of those operations that use the sort heap include: sorts, dynamic bitmaps (used for index ANDing and Star Joins), and operations where the table is in memory.

The following information is to be added to the description of this parameter:

There is no reason to increase the value of this parameter when moving from a single-node to a multi-node environment. Once you have tuned the database and database manager configuration parameters on a single node (in a DB2 EE) environment, the same values will in most cases work well in a multi-node (in a DB2 EEE) environment.

The Sort Heap Threshold parameter, as a database manager configuration parameter, applies across the entire DB2 instance. The only way to set this parameter to different values on different nodes or partitions, is to create more than one DB2 instance. This will require managing different DB2 databases over different nodegroups. Such an arrangement defeats the purpose of many of the advantages of a partitioned database environment.

Maximum Percent of Lock List Before Escalation (maxlocks)

The following change pertains to the Recommendation section of the "Maximum Percent of Lock List Before Escalation (maxlocks)" database configuration parameter.

Recommendation: The following formula allows you to set *maxlocks* to allow an application to hold twice the average number of locks:

$$\text{maxlocks} = 2 * 100 / \text{maxappls}$$

Where 2 is used to achieve twice the average and 100 represents the largest percentage value allowed. If you have only a few applications that run concurrently, you could use the following formula as an alternative to the first formula:

$$\text{maxlocks} = 2 * 100 / (\text{average number of applications running concurrently})$$

One of the considerations when setting *maxlocks* is to use it in conjunction with the size of the lock list (*locklist*). The actual limit of the number of locks held by an application before lock escalation occurs is:

$$\text{maxlocks} * \text{locklist} * 4096 / (100 * 36)$$

Where 4096 is the number of bytes in a page, 100 is the largest percentage value allowed for *maxlocks*, and 36 is the number of bytes per lock. If you know that one of your applications requires 1000 locks, and you do not want lock escalation to occur, then you should choose values for *maxlocks* and *locklist* in this formula so that the result is greater than 1000. (Using 10 for *maxlocks* and 100 for *locklist*, this formula results in greater than the 1000 locks needed.)

If *maxlocks* is set too low, lock escalation happens when there is still enough lock space for other concurrent applications. If *maxlocks* is set too high, a few applications can consume most of the lock space, and other applications will have to perform lock escalation. The need for lock escalation in this case results in poor concurrency.

You may use the Database System Monitor to help you track and tune this configuration parameter.

Configuring DB2/DB2 Data Links Manager/Data Links Access Token Expiry Interval (dl_expint)

Contrary to the documentation, if dl_expint is set to "-1", the access control token expires. The workaround for this is to set dl_expint to its maximum value, 31536000 (seconds). This corresponds to an expiration time of one year, which should be adequate for all applications.

MIN_DEC_DIV_3 Database Configuration Parameter

The addition of the MIN_DEC_DIV_3 database configuration parameter is provided as a quick way to enable a change to computation of the scale for decimal division in SQL. MIN_DEC_DIV_3 can be set to YES or NO. The default value for MIN_DEC_DIV_3 is NO.

The MIN_DEC_DIV_3 database configuration parameter changes the resulting scale of a decimal arithmetic operation involving division. If the value is NO, the scale is calculated as $31-p+s-s'$. Refer to the *SQL Reference*, Chapter 3, "Decimal Arithmetic in SQL" for more information. If set to YES, the scale is calculated as $\text{MAX}(3, 31-p+s-s')$. This causes the result of decimal division to always have a scale of at least 3. Precision is always 31.

Changing this database configuration parameter may cause changes to applications for existing databases. This can occur when the resulting scale for decimal division would be impacted by changing this database configuration parameter. Listed below are some possible scenarios that may impact applications. These scenarios should be considered before changing the MIN_DEC_DIV_3 on a database server with existing databases.

- If the resulting scale of one of the view columns is changed, a view that is defined in an environment with one setting could fail with SQLCODE -344 when referenced after the database configuration parameter is changed. The message SQL0344N refers to recursive common table expressions, however, if the object name (first token) is a view, then you will need to drop the view and create it again to avoid this error.
- A static package will not change behavior until the package is rebound, either implicitly or explicitly. For example, after changing the value from NO to YES, the additional scale digits may not be included in the results until rebind occurs. For any changed static packages, an explicit rebind command can be used to force a rebind.
- A check constraint involving decimal division may restrict some values that were previously accepted. Such rows now violate the constraint but will not be detected until the one of the columns involved in the check constraint row is updated or the **SET INTEGRITY** command with the **IMMEDIATE CHECKED** option is processed. To force checking of such a constraint, perform an **ALTER TABLE** command in order to drop the check constraint and then perform an **ALTER TABLE** command to add the constraint again.

註: DB2 Version 7 also has the following limitations:

1. The command **GET DB CFG FOR DBNAME** will not display the MIN_DEC_DIV_3 setting. The best way to determine the current setting is to observe the side-effect of a decimal division result. For example, consider the following statement:

VALUES (DEC(1,31,0)/DEC(1,31,5))

If this statement returns sqlcode SQL0419N, then the database does not have MIN_DEC_DIV_3 support or it is set to OFF. If the statement returns 1.000, then MIN_DEC_DIV_3 is set to ON.

2. MIN_DEC_DIV_3 does not appear in the list of configuration keywords when you run the following command: **? UPDATE DB CFG**

Application Control Heap Size (app_ctl_heap_sz)

The text for this parameter should now read:

For partitioned databases and non-partitioned databases with intra-parallelism enabled (intra_parallel=ON), this is the size of the shared memory area allocated for the application control heap. For non-partitioned databases where intra-parallelism is disabled (intra_parallel=OFF), this is the maximum private memory that will be allocated for the heap. There is one application control heap per connection per partition.

The application control heap is required primarily for sharing information between agents working on behalf of the same request, and, in a partitioned database environment, for storing executable sections representing SQL statements. Usage of this heap is minimal for non-partitioned databases when running queries with a degree of parallelism less than or equal to 1.

This heap is also used to store descriptor information for declared temporary tables. The descriptor information for all declared temporary tables that have not been explicitly dropped is kept in this heap's memory and cannot be dropped until the declared temporary table is dropped.

The 『Recommendation』 portion remains unchanged.

Database System Monitor Heap Size (mon_heap_sz)

The default for the OS/2 and Windows NT Database server with local and remote clients and Satellite database server with local clients has changed from 24 to 32. The range is unchanged.

Maximum Number of Active Applications (maxappls)

The upper range limit for all platforms has changed from 64 000 to 60 000. The default value is unchanged.

Recovery Range and Soft Checkpoint Interval (softmax)

The unit of measure is changed to the percentage of the size of one primary log file.

Track Modified Pages Enable (trackmod)

Configuration Type: Database

Parameter Type: Configurable

Default [Range]: Off [On; Off]

When this parameter is set to ON, the database manager will track which pages in the database have changed since the most recent full backup was taken. This allows the backup utility to determine which pages should be included in an incremental backup without having to examine every page individually. For SMS tablespaces, the granularity of this tracking is at the tables pace level. For DMS table spaces, the granularity is at the extent level for data and index pages and at the table space level for other page types. After setting this parameter to ON, you must take a full database backup in order to have a baseline against which incremental backups can be taken.

Change the Database Log Path (newlogpath)

Configuration Type: Database

Parameter Type: Configurable

Default [Range]: Null [any valid path or device]

Related Parameters: Location of Log Files (logpath); Database is Consistent (database_consistent)

This parameter allows you to specify a string of up to 242 bytes to change the location where the log files are stored. The string can point to either a path name, or to a raw device. If the string points to a path name, it must be a fully qualified path name, not a relative path name.

Note: In a partitioned database environment, the node number is automatically appended to the path. This is done to maintain the uniqueness of the path in multiple logical node configurations.

To specify a device, specify a string that the operating system identifies as a device. For example, on Windows NT,

\\.\d: or \\.\PhysicalDisk5

Note: You must have Windows NT Version 4.0 with Service Pack 3 installed to be able to write logs to a device.

On UNIX-based platforms,

/dev/rdblog8

Note: You can only specify a device on AIX, Windows 2000, Windows NT, Solaris, HP-UX, NUMA-Q, and Linux platforms.

The new setting does not become the value of logpath until both of the following occur:

- The database is in a consistent state, as indicated by the database_consistent parameter.
- All users are disconnected from the database.

When the first new connection is made to the database, the database manager will move the logs to the new location specified by logpath.

There might be log files in the old log path. These log files might not have been archived. You might need to archive these log files manually. Also, if you are running replication on this database, replication might still need the log files from before the log path change.

If the database is configured with the User Exit Enable (userexit) database configuration parameter set to "Yes", and if all the log files have been archived either by DB2 automatically or by yourself manually, then DB2 will be able to retrieve the log files to complete the replication process. Otherwise, you can copy the files from the old log path to the new log path.

Recommendation: Ideally, the log files will be on a physical disk which does not have high I/O. For instance, avoid putting the logs on the same disk as the operating system or high volume databases. This will allow for efficient logging activity with a minimum of overhead such as waiting for I/O.

You can use the database system monitor to track the number of I/Os related to database logging.

For more information, refer to the following monitor element descriptions in the System Monitor Guide and Reference:

- log_reads (number of log pages read)
- log_writes (number of log pages written)

The preceding data elements return the amount of I/O activity related to database logging. You can use an operating system monitor tool to collect information about other disk I/O activity, then compare the two types of I/O activity.

Location of Log Files (logpath)

Configuration Type: Database

Parameter Type: Informational

Related Parameters: Change the Database Log Path (newlogpath)

This parameter contains the current path being used for logging purposes. You cannot change this parameter directly as it is set by the database manager after a change to the newlogpath parameter becomes effective. When a database is created, the recovery log file for it is created in a subdirectory of the directory containing the database. The default is a subdirectory named SQLOGDIR under the directory created for the database.

Maximum Storage for Lock List (locklist)

The maximum value is increased from 60 000 to 524 288.

Appendix A. DB2 Registry and Environment Variables

The following registry variables are new or require changes:

Table of New and Changed Registry Variables

Table 6. Registry Variables

Variable Name	Operating System	Values
Description		
DB2MAXFSCRSEARCH	All	Default=5 Values: -1, 1 to 33554
Specifies the number of free space control records to search when adding a record to a table. The default is to search five free space control records. Modifying this value allows you to balance insert speed with space reuse. Use large values to optimize for space reuse. Use small values to optimize for insert speed. Setting the value to -1 forces the database manager to search all free space control records.		
DLFM_TSM_MGMTCLASS	AIX, Windows NT, Solaris	Default: the default TSM management class Values: any valid TSM management class
Specifies which TSM management class to use to archive and retrieve linked files. If there is no value set for this variable, the default TSM management class is used.		
DB2_CORRELATED_PREDICATES	All	Default=YES Values: YES or NO
The default for this variable is YES. When there are unique indexes on correlated columns in a join, and this registry variable is YES, the optimizer attempts to detect and compensate for correlation of join predicates. When this registry variable is YES, the optimizer uses the KEYCARD information of unique index statistics to detect cases of correlation, and dynamically adjusts the combined selectivities of the correlated predicates, thus obtaining a more accurate estimate of the join size and cost.		
DB2_VI_DEVICE	Windows NT	Default=null Values: nic0 or VINIC
Specifies the symbolic name of the device or Virtual Interface Provider Instance associated with the Network Interface Card (NIC). Independent hardware vendors (IHVs) each produce their own NIC. Only one (1) NIC is allowed per Windows NT machine; Multiple logical nodes on the same physical machine will share the same NIC. The symbolic device name 『VINIC』 must be in upper case and can only be used with Synfinity Interconnect. All other currently supported implementations use 『nic0』 as the symbolic device name.		
DB2_SELECTIVITY	ALL	Default=NO Values: YES or NO
This registry variable controls where the SELECTIVITY clause can be used. See the <i>SQL Reference</i> , Language Elements, Search Conditions for complete details on the SELECTIVITY clause.		
When this registry variable is set to YES, the SELECTIVITY clause can be specified when the predicate is a basic predicate where at least one expression contains host variables.		
DB2_UPDATE_PART_KEY	ALL	Default=YES Values: YES or NO
For FixPak 3 and later, the default value is YES. This registry variable specifies whether or not update of the partitioning key is permitted.		
DB2_BLOCK_ON_LOG_DISK_FULL	ALL	Default=NO Values: YES or NO

Table 6. Registry Variables (continue)

Variable Name	Operating System	Values
Description		
<p>This DB2 registry variable can be set to prevent "disk full" errors from being generated when DB2 cannot create a new log file in the active log path.</p> <p>Instead, DB2 attempts to create the log file every 5 minutes until it succeeds. After each attempt, DB2 writes a message to the db2diag.log file. The only way that you can confirm that your application is hanging because of a log disk full condition is to monitor the db2diag.log file.</p> <p>Until the log file is successfully created, any user application that attempts to update table data will not be able to commit transactions. Read-only queries may not be directly affected; however, if a query needs to access data that is locked by an update request, or a data page that is fixed in the buffer pool by the updating application, read-only queries will also appear to hang.</p>		
DB2_INDEX_2BYTEVARLEN	All	Default=NO Values: YES or NO
<p>This registry variable allows columns with a length greater than 255 bytes to be specified as part of an index key. Indexes already created before turning this registry variable YES will continue to have the 255 key limit restriction. Indexes created after turning this registry variable YES will behave as a two-byte index even when the registry variable is turned NO again.</p> <p>Several SQL statements are affected by changes to this registry variable including CREATE TABLE, CREATE INDEX, and ALTER TABLE. For more information on these statements, refer to the changes documented for the <i>SQL Reference</i>.</p>		
DB2_FORCE_FCM_BP	AIX	Default=NO Values: YES or NO
<p>Specifies from where the fast communications manager (FCM) resources are allocated. The resources may be allocated from either the database manager shared memory segment or a separate one. With multiple logical nodes on the same machine, this registry variable should be used. On a partitioned database system with symmetric multi-processing (SMP) enabled, the setting of this registry variable has no effect on how communication takes place. In this case, communication is always through shared memory. However, it does affect the number of shared memory segments DB2 will use.</p>		
DB2_AWE	Windows 2000	Default=NULL Values: <entry>[; <entry>;...] where <entry>=<buffer pool ID>, <number of physical pages>, <number of address windows>
<p>Allows DB2 UDB on Windows 2000 to allocate buffer pools that use up to 64 GB of memory. Windows 2000 must be configured correctly to support Address Windowing Extensions (AWE) buffer pools. This includes associating the 『lock pages in memory』 -right with the user on Windows 2000 and setting this registry variable on DB2. In setting this variable you need to know the buffer pool ID that is to be used for AWE support. You also need to determine the number of physical pages to allocate and the number of address windows.</p> <p>For information on determining the number of physical pages to allocate and the number of address windows, see the section on 『Managing the Database Buffer Pool』 found in 『Chapter 8. Operational Performance』 earlier in this section.</p> <p>Note: If AWE support is enabled, extended storage (ESTORE) cannot be used for any of the buffer pools in the database. The buffer pools referenced by this variable must already exist in SYSIBM.SYSBUFFERPOOLS.</p>		
DB2_STPROC_LOCKUP_FIRST	All	Default=NO Values: YES or NO
<p>This registry variable has been renamed from DB2_DARI_LOOKUP_ALL.</p>		

Table 6. Registry Variables (continue)

Variable Name	Operating System	Values
Description		
DB2MEMDISCLAIM	AIX	Default=YES Values: YES or NO
<p>On AIX, memory used by DB2 processes may have some associated paging space. This paging space may remain reserved, even when the associated memory has been freed. The reservation of the paging space depends on the AIX system's tunable virtual memory management allocation policy. This registry variable controls whether DB2 agents explicitly request that AIX disassociate the reserved paging space from the freed memory.</p> <p>A setting of 『YES』 results in smaller paging space requirements, and possibly less disk activity from paging. A setting of 『NO』 results in greater paging space requirements, and possibly more disk activity from paging. In some situations, such as if paging space is plentiful, and if real memory is so plentiful that paging never occurs, then a setting of NO will provide a small performance improvement.</p>		
DB2MEMMAXFREE	All	Default=8 388 608 bytes Values: 0 to 2 ³² -1 bytes
<p>This registry variable controls the maximum amount of unused memory in bytes retained by DB2 processes.</p>		
DB2_ANTIJOIN	All	Default=NO in a EEE environment Default=YES in a non-EEE environment Values: YES or NO
<p>For DB2 Universal Database EEE environments: When YES is specified, the optimizer will search for opportunities to transform NOT EXISTS subqueries into anti-joins which can be processed more efficiently by DB2. For non-EEE environments: When NO is specified, the optimizer will limit the opportunities to transform NOT EXISTS subqueries into anti-joins.</p>		
NEWLOGPATH2	UNIX	Default=NO Values: YES or NO
<p>This parameter allows you to specify whether a secondary path should be used to implement dual logging. The path that will be used is generated by appending the character '2' to the current value of 'LOGPATH'.</p>		
DB2DOMAINLIST	Windows NT	Default=NULL Values: one or more valid Windows NT domains (comma separating each)
<p>Defines one or more Windows NT domains. Only users belonging to these domains will have their connection or attachment requests accepted.</p> <p>This registry variable should only be used under a pure Windows NT domain environment with DB2 servers and clients running DB2 Universal Database Version 7.1 (or later).</p>		
DB2_LIKE_VARCHAR	All	Default=Y,N Values: Y, N, S, floating point constant between 0 and 6.2

Controls the collection and use of sub-element statistics. These are statistics about the content of data in columns when the data has a structure in the form of a series of sub-fields or sub-elements delimited by blanks.

This registry variable affects how the optimizer deals with a predicate of the form:

```
COLUMN LIKE '%xxxxxx%'
```

where the xxxxxx is any string of characters.

The syntax showing how this registry variable is used is:

```
db2set DB2_LIKE_VARCHAR=[Y|N|S|num1] [,Y|N|S|num2]
```

where

- The term preceding the comma, or the only term to the right of the predicate, means the following but only for columns that do not have positive sub-element statistics:
 - S - The optimizer estimates the length of each element in a series of elements concatenated together to form a column based on the length of the string enclosed in the % characters.
 - Y - The default. Use a default value of 1.9 for the algorithm parameter. Use a variable-length sub-element algorithm with the algorithm parameter.
 - N - Use a fix-length sub-element algorithm.
 - num1 - Use the value of num1 as the algorithm parameter with the variable length sub-element algorithm.
- The term following the comma means the following:
 - N - The default. Do not collect or use sub-element statistics.
 - Y - Collect sub-element statistics. Use a variable-length sub-element algorithm that uses the collected statistics together with the 1.9 default value for the algorithm parameter in the case of columns with positive sub-element statistics.
 - num2 - Collect sub-element statistics. Use a variable-length sub-element algorithm that uses the collected statistics together with the value of num2 as the algorithm parameter in the case of columns with positive sub-element statistics.

DB2_PINNED_BP	AIX, HP-UX	Default=NO
Values: YES or NO		
This variable is used to hold the database global memory (including buffer pools) associated with the database in the main memory on some AIX operating systems. Keeping this database global memory in the system main memory allows database performance to be more consistent.		
If, for example, the buffer pool was swapped out of the system main memory then database performance would deteriorate. The reduction of disk I/O by having the buffer pools in system memory improves database performance. If you have other applications that require more of the main memory, you will want to allow the database global memory, depending on the system main memory requirements, to be swapped out of main memory.		
When working with HP-UX in a 64-bit environment, in addition to modifying this registry variable, the DB2 instance group must be given the MLOCK privilege. This is done by having a user with root access rights do the following:		
1. Add the DB2 instance group to the /etc/privgroup file. For example, if the DB2 instance group belongs to db2iadm1 group then the following line must be added to the /etc/privgroup file:		
db2iadm1 MLOCK		
2. Issue the following command:		
setprivgrp -f /etc/privgroup		
DB2_RR_TO_RS	All	Default=NO
Values: YES or NO		

Next key locking guarantees Repeatable Read (RR) isolation level by automatically locking the next key for all INSERT and DELETE statements and the next higher key value above the result set for SELECT statements. For UPDATE statements that alter key parts of an index, the original index key is deleted and the new key value is inserted. Next key locking is done on both the key insertion and key deletion. Next key locking is required to guarantee ANSI and SQL92 standard RR, and is the DB2 default.

If your application appears to stop or hang, you should examine snapshot information for your application. If the problem appears to be with next key locking, you can set the DB2_RR_TO_RS registry variable on based on two conditions. You can turn DB2_RR_TO_RS on if none of your applications rely on Repeatable Read (RR) behavior and if it is acceptable for scans to skip over uncommitted deletes. The skipping behavior affects the RR, Read Stability (RS), and Cursor Stability (CS) isolation levels. (There is no row locking for Uncommitted Read (UR) isolation level.)

When DB2_RR_TO_RS is on, RR behavior cannot be guaranteed for scans on user tables because next key locking is not done during index key insertion and deletion. Catalog tables are not affected by this option.

The other change in behavior is that with DB2_RR_TO_RS on, scans will skip over rows that have been deleted but not committed, even though the row may have qualified for the scan.

Appendix C. SQL Explain Tools

The section titled 『Running db2expln and dynexpln』 should have the last paragraph replaced with the following:

To run db2expln, you must have SELECT privilege to the system catalog views as well as EXECUTE authority for the db2expln package. To run dynexpln, you must have BINDADD authority for the database, the schema you are using to connect to the database must exist or you must have the EXPLICIT_SCHEMA authority for the database, and you must have any privileges needed for the SQL statements being explained. (Note that if you have SYSADM or DBADM authority, you will automatically have all these authorization levels.)

Administering Satellites Guide and Reference(無中文版)

Setting up Version 7.2 DB2 Personal Edition and DB2 Workgroup Edition as Satellites

The sections that follow describe how to set up Windows-based Version 7.2 DB2 Personal Edition and DB2 Workgroup Edition systems so that they can be used as fully functional satellites in a satellite environment. For information about the terms and concepts used in the information that follows, refer to the *Administering Satellites Guide and Reference*. You can find this book at the following URL:

http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v6pubs.d2w/en_main

For Technotes that supplement the information in the *Administering Satellites Guide and Reference*, refer to the following URL:

<http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/browse.d2w/report?type=tech5udb&tech5udb=Y>

Prerequisites

To set up either DB2 Personal Edition or DB2 Workgroup Edition as satellites, you require the following:

1. A DB2 control server

The DB2 control server is a DB2 Enterprise Edition system that runs on Windows NT or AIX, and has the Control Server component installed. The DB2 Enterprise Edition system that you use must be at Version 6 with FixPak 2 or higher, or Version 7 at any FixPak level.

- If you have a Version 6 Enterprise Edition system that you want to use as the DB2 control server, see 第112頁的『Installing FixPak 2 or Higher on a Version 6 Enterprise Edition System』.
- If you are using Version 7 and do not have the Control Server component installed, install this component, re-install any FixPaks that you have already installed, then create the DB2 control server instance and satellite control database. Refer to the *Administering Satellites Guide and Reference* for instructions on creating these objects.

Note: If you are installing a Version 7.2 Enterprise Edition system on Windows NT for use as the DB2 control server, and you want to perform a response file installation, see the Technote entitled *DB2 Control Server Response File Keywords* for information about the keywords to specify in the response file.

2. The DB2 control server instance and the satellite control database

The DB2 control server instance is typically called DB2CTLSV, and the satellite control database is called SATCTLDDB. The DB2 control server instance and the satellite control database are on the Enterprise Edition system, and, on Windows NT, are automatically created when you install DB2 with the Control Server component. If you install DB2 on AIX, see the *Administering Satellites Guide and Reference* for information about creating the DB2 control server instance and the satellite control database.

3. The Satellite Administration Center

The Satellite Administration Center is the set of GUI tools that you use to set up and administer the satellite environment. You access this set of tools from the DB2 Control Center. For more information about Satellite Administration Center and the satellite environment, see the *Administering Satellites Guide and Reference*, and the online help that is available from the Satellite Administration Center. If you are running a Version 6 DB2 Control Center, see page 120 『Upgrading a Version 6 DB2 Control Center and Satellite Administration Center』.

If you have not already used the Satellite Administration Center to set up the satellite environment and to create the object that represents the new satellite in the Satellite Administration Center, you should do so before installing the satellite. For more information, see the description of how to set up and test a satellite environment in the *Administering Satellites Guide and Reference*.

4. A Version 7.2 Personal Edition or Workgroup Edition system that you want to use as a satellite.

Installation Considerations

When you install either DB2 Personal Edition or DB2 Workgroup Edition, you do not have to select any special component to enable either system to synchronize. If you intend to perform a response file installation, see 『Performing a Response File Installation』 for the keywords that you should specify when installing the Version 7.2 system. If you are performing an interactive installation of your Version 7.2 system, see 第111頁的 『Configuring the Version 7.2 System for Synchronization』 after you finish installing DB2 for values that you must set at the Version 7.2 system to enable it to synchronize.

Performing a Response File Installation: If you are performing a response file installation of Version 7.2 DB2 Personal Edition or DB2 Workgroup Edition, you can set the following keywords in the response file.

If you decide to not specify one or more of these keywords during the response file installation, see page 111 『Configuring the Version 7.2 System for Synchronization』 for additional steps that you must perform after installing DB2 to enable the Version 7.2 system to synchronize. You can also use the instructions in this section if you want to change any values that were specified during the response file installation.

db2.db2satelliteid

Sets the satellite ID on the system.

Note: If you do not specify this keyword, the satellite ID is automatically set to the user ID that was used to install DB2. If you want to use this user ID as the satellite ID, you do not have to specify a value for this keyword.

db2.db2satelliteappver

Sets the application version on the system.

Note: If you do not specify this keyword, the application version on the satellite is automatically set to V1R0M00. If you want to use this value as the application version, you do not have to specify a value for this keyword.

db2.satctldb_username

Sets the user name to be used for the system to connect to the satellite control database.

db2.satctldb_password

Sets the password that the user name passes to the DB2 control server when the user name connects to the satellite control database.

After you complete the response file installation, the Version 7.2 system is ready to synchronize. You should issue the **db2sync -t** command on the satellite to verify that the values specified on the satellite are correct, and that the satellite can connect to the satellite control database.

For additional information about performing a response file installation, refer to the *Administering Satellites Guide and Reference*.

Note:

1. In Version 7, user IDs and passwords are required for the creation of all services on Windows NT and Windows 2000. These user IDs and passwords are specified in the response file by keyword pairs. The first keyword pair found in the response file becomes the default user ID and password for all services, unless you provide an override for a service by specifying the specific keyword pair for that service.

In Version 6, the **admin.userid** and the **admin.password** keywords could be specified during a response file installation of DB2 Satellite Edition to specify the user ID and password that would be used by the Remote Command Service. For Version 7.2 Personal Edition and Workgroup Edition, if you specify these keywords, they are used for the DB2DAS00 instance on the Version 7.2 system. For a DB2 Version 7.2 system, the Remote Command Service will use the user ID and password that is used by the DB2 instance on the system. If you do not specify values for **db2.userid** and **db2.password**, the defaulting rule described above applies.

2. In Version 6, you could create a database when installing DB2 Satellite Edition using a response file installation. You cannot create a database during a response file installation on the Version 7.2 Personal Edition or Workgroup Edition system that you intend to use as a satellite. The following keywords (which are described in the *Administering Satellites Guide and Reference*), are not supported:

- **db2.userdb_name**
- **db2.userdb_recoverable**
- **db2.userdb_rep_src**

Configuring the Version 7.2 System for Synchronization

If you install the Version 7.2 system interactively, several values must be set on the DB2 Personal Edition or DB2 Workgroup Edition system after installing DB2 before the system can synchronize.

Note: You can use execute an operating system script on the system to set all values at the satellite except for the user ID and password that the satellite uses to connect to the satellite control database (see step 4).

1. Set the satellite ID by using the **db2set** command.

If you install DB2 Personal Edition or DB2 Workgroup Edition interactively, the satellite ID is automatically set to the user ID that was used to install DB2. If you want to use this user ID as the satellite ID, you do not have to perform this step. For information about setting the satellite ID, see the *Administering Satellites Guide and Reference*.

2. Set the application version on the satellite by using the **db2sync -s** command.

If you install DB2 Personal Edition or DB2 Workgroup Edition interactively, the application version on the satellite is automatically set to V1R0M00. If you want to use this value as the application version, you do not have to perform this step.

You can use the **db2sync -g** command on the satellite to view the current setting of the application version. If you want to change this value, issue the **db2sync -s** command. You are prompted to provide a new value for the application version. For more information about setting the application version, see the *Administering Satellites Guide and Reference*.

3. Issue the **catalog node** and **catalog database** commands on the satellite to catalog the DB2 control server instance and the satellite control database, SATCTLDB, at the satellite.

You can also use the **db2sync -t** command on the satellite to open the DB2 Synchronizer application in test mode. If the SATCTLDB database is not cataloged at the satellite when you issue the command, the Catalog Control Database window opens. You can either use the DB2 discovery feature that is available from the Catalog Control Database window to catalog the DB2 control server and the SATCTLDB database, or you can type the hostname and server name in this window. You will also be prompted to specify the user ID and password that the satellite will use to connect to the satellite control database, as described in step 4.

Note: After you install Version 7.2 DB2 Personal Edition or DB2 Workgroup Edition interactively, the DB2 Synchronizer does not start automatically in test mode (as was the case for Version 6 DB2 Satellite Edition).

4. Issue the **db2sync -t** command on the satellite to:

- Specify the user ID and the password that the satellite will use to connect to the satellite control database

If synchronization credentials are not already stored at the satellite, the Connect to Control Database window opens. You must use this window to specify the user ID and password the satellite will use to connect to the satellite control database.

- Verify the values that are set on the satellite are correct
- Verify that the satellite can connect to the satellite control database

After you complete these configuration tasks, the Version 7.2 system is ready to synchronize.

Installing FixPak 2 or Higher on a Version 6 Enterprise Edition System

The sections that follow describe the tasks that you must perform to upgrade a Version 6 Enterprise Edition system on Windows NT or AIX for use as a DB2 control server. If you are using a Version 6 DB2 Control Center, also perform the steps in page 120 『Upgrading a Version 6 DB2 Control Center and Satellite Administration Center』 to

verify that you have the correct level of the DB2 Control Center and the Satellite Administration Center to administer the satellite environment.

Upgrading Version 6 DB2 Enterprise Edition for Use as the DB2 Control Server

For a Version 6 DB2 Enterprise Edition system to be used as the DB2 control server, it must be installed with the Control Server component, and DB2 Enterprise Edition should be at the FixPak 2 service level, or higher. Depending on whether the DB2 control server component is installed, and the service level of DB2 Enterprise Edition, you will have to perform one of the following tasks:

- Install the DB2 control server component to an existing DB2 Enterprise Edition V6.1 system and install FixPak 2 or higher. Then update the satellite control database (SATCTLDB) on the system.
- Upgrade an already installed DB2 control server to the FixPak 2 level or higher.

Use the information that follows to identify which of the two preceding tasks you need to perform, and the steps that apply to your situation. The following is a summary of the steps that you will perform.

1. First, assess the current state of your DB2 Enterprise Edition installation. You will determine whether the Control Server component is installed, and the service level of DB2.
2. Second, based on the state information that you obtain, you will determine what needs to be done.
3. Third, you will perform the necessary steps to upgrade DB2 Enterprise Edition.

The DB2 control server can only run on DB2 Enterprise Edition for Windows NT and AIX. Continue with the instructions that are appropriate for your platform:

- 『Upgrading DB2 Enterprise Edition on Windows NT』
- Page 117 『Upgrading DB2 Enterprise Edition on AIX』

Upgrading DB2 Enterprise Edition on Windows NT: Use the information in the sections that follow to determine the current service level of your Version 6 DB2 Enterprise Edition system, and the steps that you need to perform to update the system to the FixPak 2 service level or higher. You will need to perform the steps of one or more of the following sections:

- 『Assessing DB2 Enterprise Edition on Windows NT』
- Page 114 『Determining What Needs to Be Done』
- Page 114 『Installing the Control Server Component on Windows NT』
- Page 115 『Installing FixPak 2 or Higher on Windows NT』
- Page 116 『Upgrading the SATCTLDB on Windows NT』

Assessing DB2 Enterprise Edition on Windows NT: If you have DB2 Enterprise Edition installed on Windows NT, perform the following steps:

1. Check whether the Control Server component is installed. Use the Registry Editor to display the list of installed components:
 - a. Enter `regedit` at a command prompt.

- b. Under the HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2\Components registry key, check whether the Control Server is listed. If it is not listed, the control server is not installed.
2. Determine the service level of DB2 Enterprise Edition. Issue the `db2level` command from a command prompt. Use the table that follows to interpret the output:

Values of Key Fields in the db2level output			Your DB2 system is at:
Release	Level	Informational Tokens	
SQL06010	01010104	db2_v6, n990616	Version 6.1 base
SQL06010	01020104	DB2 V6.1.0.1, n990824, WR21136	Version 6.1 plus FixPak 1
SQL06010	01030104	DB2 V6.1.0.6, s991030, WR21163 <i>or</i> DB2 V6.1.0.9, s000101, WR21173	Version 6.1 plus FixPak 2

Note: If the level is greater than 01030104, your system is at a higher FixPak than FixPak 2.

3. Record the information that you find, and continue at 『Determining What Needs to Be Done』 .

Determining What Needs to Be Done: Using the information that you have gathered, find the row in the following table that applies to your situation, and follow the steps that are required to prepare your DB2 Enterprise Edition system to support the DB2 control server at the FixPak 2 level or higher.

Sections that follow the table provide instructions for performing the required steps. Consider checking off each step as you perform it. Only perform the steps that apply to your situation.

Control Server Component Installed	Service Level of DB2 Enterprise Edition System	Steps required to prepare your DB2 Enterprise Edition system
No	Version 6.1 base, or Version 6.1 plus FixPak 1, or Version 6.1 plus FixPak 2 or higher	Perform the following steps: 1. 『Installing the Control Server Component on Windows NT』 2. 第115頁的『Installing FixPak 2 or Higher on Windows NT』 3. 第116頁的『Upgrading the SATCTLDB on Windows NT』
Yes	Version 6.1 base, or Version 6.1 plus FixPak 1	Perform the following steps: 1. 第115頁的『Installing FixPak 2 or Higher on Windows NT』 2. 第116頁的『Upgrading the SATCTLDB on Windows NT』
Yes	Version 6.1, plus FixPak 2 or higher	Perform the following step: 1. 第116頁的『Upgrading the SATCTLDB on Windows NT』

Installing the Control Server Component on Windows NT: To install the Control Server component on Windows NT:

1. Ensure that all database activity on the system is complete before proceeding.
2. Insert the DB2 Universal Database Enterprise Edition Version 6.1 CD in the CD drive.
If the installation program does not start automatically, run the setup command in the root of the CD to start the installation process.
3. When prompted, shut down all the processes that are using DB2.
4. On the Welcome window, select **Next**.
5. On the Select Products window, ensure that DB2 Enterprise Edition is selected.
6. On the Select Installation Type panel, click **Custom**.
7. On the Select Components panel, ensure that the Control Server component is selected, and click **Next**.

Note: If you select other components that are not already installed on your system, these components will be installed too. You cannot alter the drive or directory in which DB2 is installed.

8. On the Configure DB2 Services panels, you can modify the protocol values and the start-up options for the Control Server instance, or take the default values. Either modify the defaults and click **Next**, or click **Next** to use the defaults.
9. Click **Next** on the Start Copy files window to begin the installation process.
10. When the file copying process is complete, you have the option of rebooting your system. You should reboot now. The changes made to the system for the Control Server do not take effect until the system is rebooted.

When the installation process is complete and you have rebooted the system, the satellite control database (SATCTLDB) that was created as part of the Control Server installation must be cataloged in the DB2 instance if you want to use the DB2 Control Center and Satellite Administration Center locally on the system. To catalog the SATCTLDB database:

1. Open a DB2 Command Window by selecting **Start>Programs>DB2 for Windows NT>Command Window**
2. Ensure that you are in the db2 instance.
Issue the `set` command and check the value of `db2instance`. If the value is not `db2`, issue the following command:

```
set db2instance=db2
```
3. Catalog the `db2ctlsv` instance by entering the following command:

```
db2 catalog local node db2ctlsv instance db2ctlsv
```
4. Catalog the SATCTLDB database by entering the following command

```
db2 catalog database satctldb at node db2ctlsv
```
5. Commit the cataloging actions by entering the following command:

```
db2 terminate
```
6. Close the DB2 Command Window.

Installing FixPak 2 or Higher on Windows NT: To upgrade an existing Version 6 DB2 Enterprise Edition system on Windows NT to FixPak 2 or higher, either:

- Download the latest FixPak for DB2 Enterprise Edition for Windows NT V6.1 from the Web, along with its accompanying readme. The FixPak can be downloaded by following the instructions at URL:

<http://www-4.ibm.com/software/data/db2/db2tech/version61.html>

Install the FixPak following the instructions in the readme.txt file.

- Use a DB2 Universal Database, Version 6.1 FixPak for Windows NT CD that is at FixPak 2 level or higher, and follow the instructions in the readme.txt file in the WINNT95 directory on the CD to complete the installation.

Upgrading the SATCTLDB on Windows NT: To upgrade the SATCTLDB database on Windows NT

1. Determine the level of the SATCTLDB database:
 - a. Log on with a user ID that has local administrative authority on the Windows NT system.
 - b. Open a DB2 Command Window by selecting **Start>Programs>DB2 for Windows NT>Command Window**.
 - c. Connect to the SATCTLDB by entering the following command

```
db2 connect to satctldb
```
 - d. Determine if the trigger I_BATCHSTEP_TRGSCR exists in the database by issuing the following query:

```
db2 select name from sysibm.systriggers where name='I_BATCHSTEP_TRGSCR'
```

Record the number of rows that are returned.

- e. Enter the following command to close the connection to the database:

```
db2 connect reset
```

If step 1d returned one row, the database is at the correct level. In this situation, skip step 2, and continue at step 3. If zero (0) rows are returned, the database is not at the correct level, and must be upgraded, as described in step 2, before you can perform step 3.

2. To upgrade the SATCTLDB database, perform the following steps. Enter all commands in the DB2 Command Window:
 - a. Switch to the directory <db2path>\misc, where <db2path> is the install drive and path, for example c:\sql1lib.
 - b. Ensure that you are in the db2ct1sv instance.
Issue the set command and check the value of db2instance. If the value is not db2ct1sv, issue the following command:

```
set db2instance=db2ct1sv
```
 - c. Drop the SATCTLDB database by entering the following command:

```
db2 drop database satctldb
```
 - d. Create the new SATCTLDB database by entering the following command:

```
db2 -tf satctldb.dd1 -z satctldb.log
```
 - e. Issue the following command:

```
db2 terminate
```
3. Bind the db2satcs.dll stored procedure to the SATCTLDB database. Perform the following steps:
 - a. Connect to the SATCTLDB database by entering the following command

db2 connect to satctl1db

b. Switch to the directory <db2path>\bnd, where <db2path> is the install drive and path, for example c:\sql1ib.

c. Issue the bind command, as follows:

db2 bind db2satcs.bnd

4. Enter the following command to close the connection to the database:

db2 connect reset

5. Close the DB2 Command Window.

Upgrading DB2 Enterprise Edition on AIX: Use the information in the sections that follow to determine the current service level of your Version 6 DB2 Enterprise Edition system, and the steps that you need to perform to update the system to the FixPak 2 service level, or higher. You will need to perform the steps of one or more of the following sections:

- 『Assessing DB2 Enterprise Edition on AIX』
- 『Determining What Needs to Be Done』
- Page 118 『Installing the Control Server Component on AIX』
- Page 118 『Installing FixPak 2 or Higher on AIX』
- Page 119 『Upgrading the SATCTLDB Database on AIX』

Assessing DB2 Enterprise Edition on AIX: If you have Version 6 DB2 Enterprise Edition installed on AIX, perform the following steps:

1. Check whether the Control Server component is installed. Enter the following command:

```
ls1pp -l | grep db2_06_01.ctsr
```

If no data is returned, the Control Server component is not installed.

2. Determine the service level of the DB2 Enterprise Edition. Log on as a DB2 instance owner, and issue the db2level command. Use the table that follows to interpret the output:

Values of Key Fields in the db2level output			Your DB2 system is at:
Release	Level	Informational Tokens	
SQL06010	01010104	db2_v6, n990616	Version 6.1 base
SQL06010	01020104	DB2 V6.1.0.1, n990824, U465423	Version 6.1 plus FixPak 1
SQL06010	01030104	DB2 V6.1.0.6, s991030, U468276 or DB2 V6.1.0.9, s000101, U469453	Version 6.1 plus FixPak 2

Note: If the level is greater than 01030104, your system is at a higher FixPak than FixPak 2.

3. Record the information that you find, and continue at 『Determining What Needs to Be Done』.

Determining What Needs to Be Done: Using the information that you have gathered, find the row in the following table that applies to your situation, and follow the steps that are required to prepare your Version 6 DB2 Enterprise Edition system to support the DB2 control server at the FixPak 2 level.

Sections that follow the table provide instructions for performing the required steps. Consider checking off each step as you perform it. Only perform the steps that apply to your situation.

Control Server Component Installed	Service Level of DB2 Enterprise Edition System	Steps required to prepare your DB2 Enterprise Edition system
No	Version 6.1 base, or Version 6.1 plus FixPak 1, or Version 6.1 plus FixPak 2 or higher	Perform the following steps: 1. 『Installing the Control Server Component on AIX』 2. 『Installing FixPak 2 or Higher on AIX』 3. 第119頁的『Upgrading the SATCTLDB Database on AIX』
Yes	Version 6.1 base, or Version 6.1 plus FixPak 1	Perform the following steps: 1. 『Installing FixPak 2 or Higher on AIX』 2. 第119頁的『Upgrading the SATCTLDB Database on AIX』
Yes	Version 6.1, plus FixPak 2 or higher	Perform the following step: 1. 第119頁的『Upgrading the SATCTLDB Database on AIX』

Installing the Control Server Component on AIX: To install the Control Server component on AIX

1. Log on as a user with root authority.
2. Insert the DB2 Universal Database Enterprise Edition Version 6.1 CD in the CD drive.
3. Change to the directory where the CD is mounted, for example, `cd /cdrom`.
4. Type the following command to start the DB2 installer:

```
./db2setup
```
5. When the DB2 Installer window opens, use the tab key to select the **Install** option, and press Enter.
6. Locate the Enterprise Edition line and use the tab key to select the **Customize** option beside it. Press Enter.
7. Select the DB2 Control Server component, tab to OK, and press Enter.
8. Follow the instructions on the remaining windows to complete the installation of the DB2 Control Server component.

When the installation process is complete, create the DB2CTLSV instance and the SATCTLDB database. To perform these tasks, follow the detailed instructions in "Setting up the DB2 Control Server on AIX" in Chapter 13 of the *Administering Satellites Guide and Reference*.

Installing FixPak 2 or Higher on AIX: To upgrade an existing DB2 Enterprise Edition system AIX to FixPak 2 or higher, either:

- Download the latest FixPak for DB2 Enterprise Edition for AIX V6.1 from the Web, along with its accompanying Fixpak.Readme. The FixPak can be downloaded by following the instructions at URL:

`http://www-4.ibm.com/software/data/db2/db2tech/version61.html`

Install the FixPak following the instructions in the Fixpak.Readme file.

- Use a DB2 Universal Database, Version 6.1 FixPak for AIX CD that is at FixPak 2 level or higher, and follow the instructions in the readme directory on the CD to complete the installation.

Ensure that you have updated the DB2CTLSV instance by running the `db2iupdt` command as instructed in the FixPak.Readme file.

Upgrading the SATCTLDB Database on AIX: To upgrade the SATCTLDB database on AIX:

1. Determine the level of the SATCTLDB database:
 - a. Log in as `db2ctlsv`.
 - b. Ensure that the database server has been started. If the server is not started, issue the `db2start` command.
 - c. Connect to the SATCTLDB database by entering the following command:

```
db2 connect to satctl db
```

- d. Determine if the trigger `I_BATCHSTEP_TRGSCR` exists in the database by issuing the following query:

```
db2 "select name from sysibm.systriggers where name='I_BATCHSTEP_TRGSCR'"
```

Record the number of rows that are returned.

- e. Enter the following command to close the connection to the database:

```
db2 connect reset
```

If step 1d returned one row, the database is at the correct level. In this situation, skip step 2, and continue at step 3. If zero (0) rows are returned, the database is not at the correct level, and must be upgraded, as described in step 2, before you can perform step 3.

2. To upgrade the SATCTLDB database to the FixPak 2 level, perform the following steps. Enter all commands in the DB2 Command Window:
 - a. Switch to the `$HOME/sqllib/misc` directory.
 - b. Drop the SATCTLDB database by entering the following command:


```
db2 drop database satctl db
```
 - c. Create the new SATCTLDB database by entering the following command:


```
db2 -tf satctl db.dd1 -z $HOME/satctl db.log
```
 - d. Issue the following command:


```
db2 terminate
```
3. Bind the `db2satcs.dll` stored procedure to the SATCTLDB database. Perform the following steps:
 - a. Connect to the SATCTLDB database by entering the following command

| db2 connect to satctlldb

| b. Switch to the directory \$HOME/sqlllib/bnd.

| c. Issue the bind command, as follows:

| db2 bind db2satcs.bnd

| 4. Enter the following command to close the connection to the database:

| db2 connect reset

| **Upgrading a Version 6 DB2 Control Center and Satellite Administration Center**

| To use a Version 6 DB2 Control Center and Satellite Administration Center with a Version 6 DB2 control server and satellite control database (SATCTLDB) that have been upgraded to FixPak 2 or higher, the tools must also be upgraded to FixPak 2 or higher.

| If the DB2 Control Center and Satellite Administration Center are running on the same system as the DB2 control server, they were upgraded when the DB2 Enterprise Edition system was upgraded to FixPak 2. However, if you run these tools on another system, you must upgrade this system to the FixPak 2 level or higher.

| To upgrade this system to FixPak 2 or higher:

- | • Download the latest FixPak for your product at the V6.1 level from the Web, along with its accompanying readme. FixPaks can be downloaded by following the instructions at URL:

| <http://www-4.ibm.com/software/data/db2/db2tech/version61.html>

| Install the FixPak following the instructions in the readme file.

- | • Use a DB2 Universal Database, Version 6.1 FixPak CD for the operating system that you are running that is at FixPak 2 level or higher, and follow the instructions in the readme to complete the installation.

Command Reference(無中文版)

db2batch - Benchmark Tool

The last sentence in the description of the PERF_DETAIL parameter should read:

A value greater than 1 is only valid on DB2 Version 2 and DB2 UDB servers,
and is not currently supported on host machines.

db2cap (new command)

db2cap - CLI/ODBC Static Package Binding Tool

Binds a capture file to generate one or more static packages. A capture file is generated during a static profiling session of a CLI/ODBC/JDBC application, and contains SQL statements that were captured during the application run. This utility processes the capture file so that it can be used by the CLI/ODBC/JDBC driver to execute static SQL for the application.

For more information on how to use static SQL in CLI/ODBC/JDBC applications, see the Static Profiling feature in the *CLI Guide and Reference*.

Authorization

- Access privileges to any database objects referenced by SQL statements recorded in the capture file.
- Sufficient authority to set bind options such as OWNER and QUALIFIER if they are different from the connect ID used to invoke the **db2cap** command.
- BINDADD authority if the package is being bound for the first time; otherwise, BIND authority is required.

Command Syntax

```
db2cap -h -? bind capture-file -d database_alias  
-u userid -p password
```

Command Parameters

-h/-? Displays help text for the command syntax.

bind *capture-file*

Binds the statements from the capture file and creates one or more packages.

-d *database_alias*

Specifies the database alias for the database that will contain one or more packages.

-u *userid*

Specifies the user ID to be used to connect to the data source.

註: If a user ID is not specified, a trusted authorization ID is obtained from the system.

-p *password*

Specifies the password to be used to connect to the data source.

Usage Notes

The command must be entered in lowercase on UNIX platforms, but can be entered in either lowercase or uppercase on Windows operating systems and OS/2.

This utility supports a number of user-specified bind options that can be found in the capture file. For performance and security reasons, the file can be examined and edited with a text editor to change these options.

The `SQLERROR(CONTINUE)` and the `VALIDATE(RUN)` bind options can be used to create a package.

When using this utility to create a package, static profiling must be disabled.

The number of packages created depends on the isolation levels used for the SQL statements that are recorded in the capture file. The package name consists of up to a maximum of the first seven characters of the package keyword from the capture file, and one of the following single-character suffixes:

- 0 - Uncommitted Read (UR)
- 1 - Cursor Stability (CS)
- 2 - Read Stability (RS)
- 3 - Repeatable Read (RR)
- 4 - No Commit (NC)

To obtain specific information about packages, the user can:

- Query the appropriate SYSIBM catalog tables using the `COLLECTION` and `PACKAGE` keywords found in the capture file.
- View the capture file.

db2ckrst (new command)

db2ckrst - Check Incremental Restore Image Sequence

Queries the database history and generates a list of timestamps for the backup images required for an incremental restore. A simplified restore syntax for a manual incremental restore is also generated.

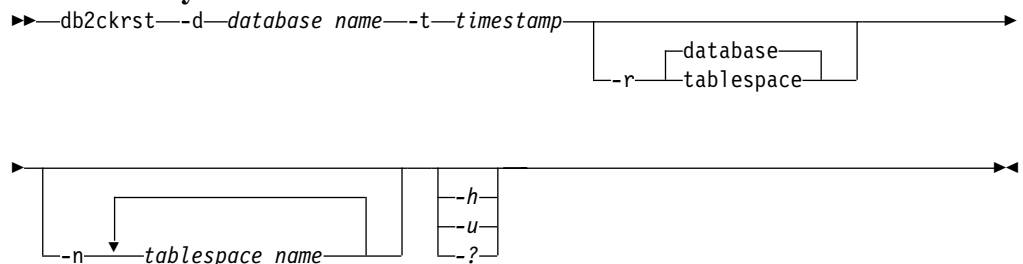
Authorization

None

Required Connection

None

Command Syntax



Command Parameters

-d database namefile-name

Specifies the alias name for the database that will be restored.

-t timestamp

Specifies the timestamp for a backup image that will be incrementally restored.

-r Specifies the type of restore that will be executed. The default is database.

註: If tablespace is chosen and no table space names are given, the utility looks into the history entry of the specified image and uses the table space names listed to do the restore.

-n tablespace name

Specifies the name of one or more table spaces that will be restored.

註: If a database restore type is selected and a list of table space names is specified, the utility will continue as a tablespace restore using the table space names given.

-h/-u/-?

Displays help information. When this option is specified, all other options are ignored, and only the help information is displayed.

Examples

```
db2ckrst -d mr -t 20001015193455 -r database
db2ckrst -d mr -t 20001015193455 -r tablespace
db2ckrst -d mr -t 20001015193455 -r tablespace -n tbsp1 tbsp2
```

```
> db2 backup db mr
```

```

Backup successful. The timestamp for this backup image is : 20001016001426
> db2 backup db mr incremental

Backup successful. The timestamp for this backup image is : 20001016001445
> db2ckrst -d mr -t 20001016001445

Suggested restore order of images using timestamp 20001016001445 for database mr.
=====
db2 restore db mr incremental taken at 20001016001445
db2 restore db mr incremental taken at 20001016001426
db2 restore db mr incremental taken at 20001016001445
=====

> db2ckrst -d mr -t 20001016001445 -r tablespace -n userspace1
Suggested restore order of images using timestamp 20001016001445 for database mr.
=====
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at 20001016001445
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at 20001016001426
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at 20001016001445
=====

```

Usage Notes

The database history must exist in order for this utility to be used. If the database history does not exist, specify the HISTORY FILE option in the RESTORE command before using this utility.

If the FORCE option of the PRUNE HISTORY command is used, it will be possible to delete entries that are required for recovery from the most recent, full database backup image. The default operation of the PRUNE HISTORY command prevents required entries from being deleted. It is recommended that the FORCE option of the PRUNE HISTORY command not be used.

It is recommended that you keep good records of your backups and use this utility as a guide.

db2gncol (new command)

db2gncol - Update Generated Column Values

Updates generated columns in tables that are in check pending mode and have limited log space. This tool is used to prepare for a SET INTEGRITY statement on a table that has columns which are generated by expressions.

Authorization

One of the following

- *sysadm*
- *dbadm*

Command Syntax

```
▶▶ db2gncol -d database -s schema_name -t table_name -c commit_count ▶▶  
  
┌──────────────────┬────────┐  
└-u userid -p password┘ └-h┘
```

Command Parameters

-d *database*

Specifies an alias name for the database in which the table is located.

-s *schema_name*

Specifies the schema name for the table. The schema name is case sensitive.

-t *table_name*

Specifies the table for which new column values generated by expressions are to be computed. The table name is case sensitive.

-c *commit_count*

Specifies the number of rows updated between commits. This parameter influences the size of the log space required to generate the column values.

-u *userid*

Specifies a user ID with system administrator or database administrator privileges. If this option is omitted, the current user is assumed.

-p *password*

Specifies the password for the specified user ID.

-h

Displays help information. When this option is specified, all other options are ignored, and only the help information is displayed.

Usage Notes

Using this tool instead of the FORCE GENERATED option on the SET INTEGRITY statement may be necessary if a table is large and the following conditions exist:

- All column values must be regenerated after altering the generation expression of a generated column.
- An external UDF used in a generated column was changed, causing many column values to change.
- A generated column was added to the table.

- A large load or load append was performed that did not provide values for the generated columns.
- The log space is too small due to long-running concurrent transactions or the size of the table.

This tool will regenerate all column values that were created based on expressions. While the table is being updated, intermittent commits are performed to avoid using up all of the log space. Once **db2gncol** has been run, the table can be taken out of check pending mode using the SET INTEGRITY statement.

db2inidb - Initialize a Mirrored Database

In a split mirror environment, this command is used to initialize a mirrored database for different purposes.

Authorization

Must be one of the following:

- *sysadm*
- *sysctrl*
- *sysmaint*

Required Connection

None

Command Syntax

```
▶▶ db2inidb database_alias AS 

|          |
|----------|
| SNAPSHOT |
| STANDBY  |
| MIRROR   |

 ▶▶
```

Command Parameters

database_alias

Specifies the alias of the database to be initialized.

SNAPSHOT

Specifies that the mirrored database will be initialized as a clone of the primary database. This database is read only.

STANDBY

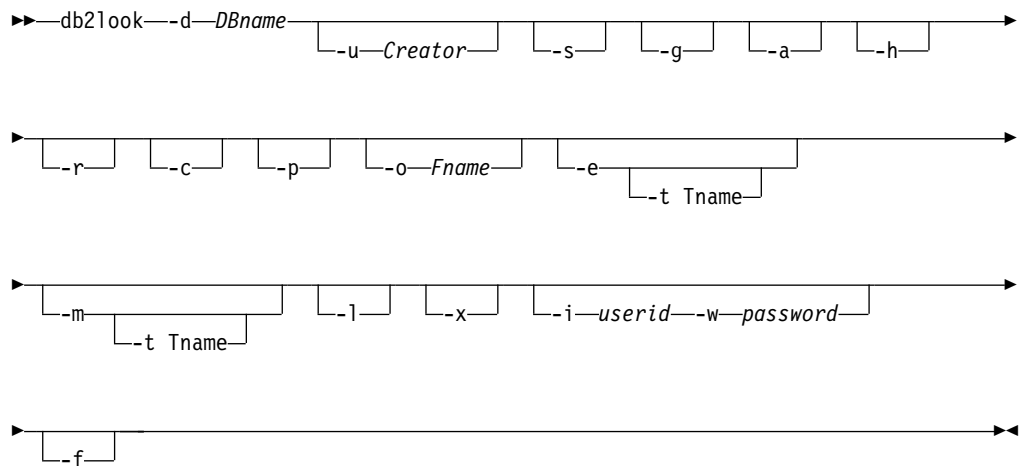
Specifies that the database will be placed in roll forward pending state. New logs from the primary database can be fetched and applied to the standby database. The standby database can then be used in place of the primary database if it goes down.

MIRROR

Specifies that the mirrored database is to be used as a backup image which can be used to restore the primary database.

db2look - DB2 Statistics Extraction Tool

The syntax diagram should appear as follows:



The -td x parameter has been added following the -c parameter. Its definition is as follows:

Specifies the statement delimiter for SQL statements generated by db2look. If this option is not specified, the default is the semicolon ';'. It is recommended that this option be used if the -e option is specified. In this case, the extracted objects may contain triggers or SQL routines.

The following example will also be added:

Generate the DDL statements for objects created by all users in the database DEPARTMENT. The db2look output is sent to file db2look.sql:

```
db2look -d department -a -e -td % -o db2look.sql
db2 -td% -f db2look.sql
```

db2updv7 - Update Database to Version 7 Current Fix Level

This command updates the system catalogs in a database to support the current FixPak in the following ways:

- Enables the use of the new built-in functions (ABS, DECRYPT_BIN, DECRYPT_CHAR, ENCRYPT, GETHINT, MULTIPLY_ALT, and ROUND).
- Enables the use of the new built-in procedures (GET_ROUTINE_SAR and PUT_ROUTINE_SAR).
- Adds or applies corrections to WEEK_ISO and DAYOFWEEK_ISO functions on Windows and OS/2 databases.
- Applies a correction to table packed descriptors for tables migrated from Version 2 to Version 6.
- Creates the view SYSCAT.SEQUENCES.

Authorization

sysadm

Required Connection

Database. This command automatically establishes a connection to the specified database.

Command Syntax

```
db2updv7 -d database_name [-u userid -p password] [-h]
```

Command Parameters

-d database-name

Specifies the name of the database to be updated.

-u userid

Specifies the user ID.

-p password

Specifies the password for the user.

-h

Displays help information. When this option is specified, all other options are ignored, and only the help information is displayed.

Example

After installing the FixPak, update the system catalog in the sample database by issuing the following command:

```
db2updv7 -d sample
```

Usage Notes

This tool can only be used on a database running DB2 Version 7.1 or Version 7.2 with at least FixPak 2 installed. If the command is issued more than once, no errors are reported and each of the catalog updates is applied only once.

To enable the new built-in functions, all applications must disconnect from this database and the database must be deactivated if it has been activated.

New Command Line Processor Option (-x, Suppress printing of column headings)

A new option, -x, tells the command line processor to return data without any headers, including column names. The default setting for this command option is OFF.

True Type Font Requirement for DB2 CLP

To correctly display the national characters for single byte (SBCS) languages correctly from the DB2 command line processor (CLP) window, change the font to True Type.

ADD DATALINKS MANAGER

The required authorization level for this command is one of the following:

- *sysadm*
- *sysctrl*
- *sysmaint*

The following usage note should be added:

This command is effective only after all applications have been disconnected from the database. The DB2 Data Links Manager being added must be completely set up and running for this command to be successful. The database must also be registered on the DB2 Data Links Manager using the `dlfm add_db` command. The maximum number of DB2 Data Links Managers that can be added to a database is 16.

ARCHIVE LOG (new command)

Archive Log

Closes and truncates the active log file for a recoverable database. If user exit is enabled, issues an archive request.

Authorization

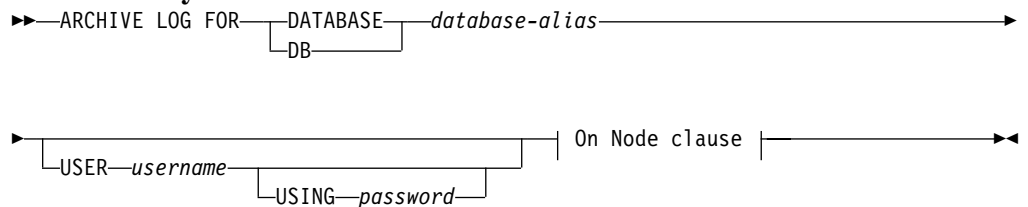
One of the following:

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

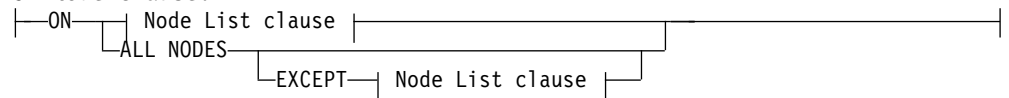
Required Connection

This command automatically establishes a connection to the specified database. If a connection already exists, an error is returned.

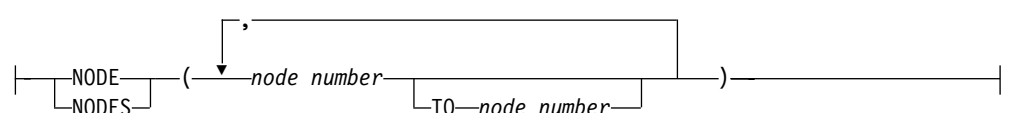
Command Syntax



On Node clause:



Node List clause:



Command Parameters

DATABASE database-alias

Specifies the alias of the database whose active log is to be archived.

USER username

Identifies the user name under which a connection will be attempted.

USING password

Specifies the password to authenticate the user name.

ON ALL NODES

Specifies that the command should be issued on all nodes in the db2nodes.cfg file. This is the default if a node clause is not specified.

EXCEPT

Specifies that the command should be issued on all nodes in the `db2nodes.cfg` file, except those specified in the node list.

ON NODE/ON NODES

Specifies that the logs should be archived for the specified database on a set of nodes.

node number

Specifies a node number in the node list.

TO node number

Used when specifying a range of nodes for which the logs should be archived. All nodes from the first node number specified up to and including the second node number specified are included in the node list.

Usage Notes

This command can be used to collect a complete set of log files up to a known point. The log files can then be used to update a standby database.

This function can only be executed when there is no database connection to the specified database. This prevents a user from executing the command with uncommitted transactions. As such, the ARCHIVE LOG command will not forcibly commit the user's incomplete transactions. If a database connection to the specified database already exists and this command is executed, the command will terminate and return an error. If another application has transactions in progress with the specified database when this command is executed, there will be a slight performance degradation since the command flushes the log buffer to disk. Any other transactions attempting to write log records to the buffer will have to wait until the flush is complete.

If used in an MPP environment, a subset of nodes may be specified by using a node clause. If the node clause is not specified, the default behaviour for this command is to close and archive the active log on all nodes.

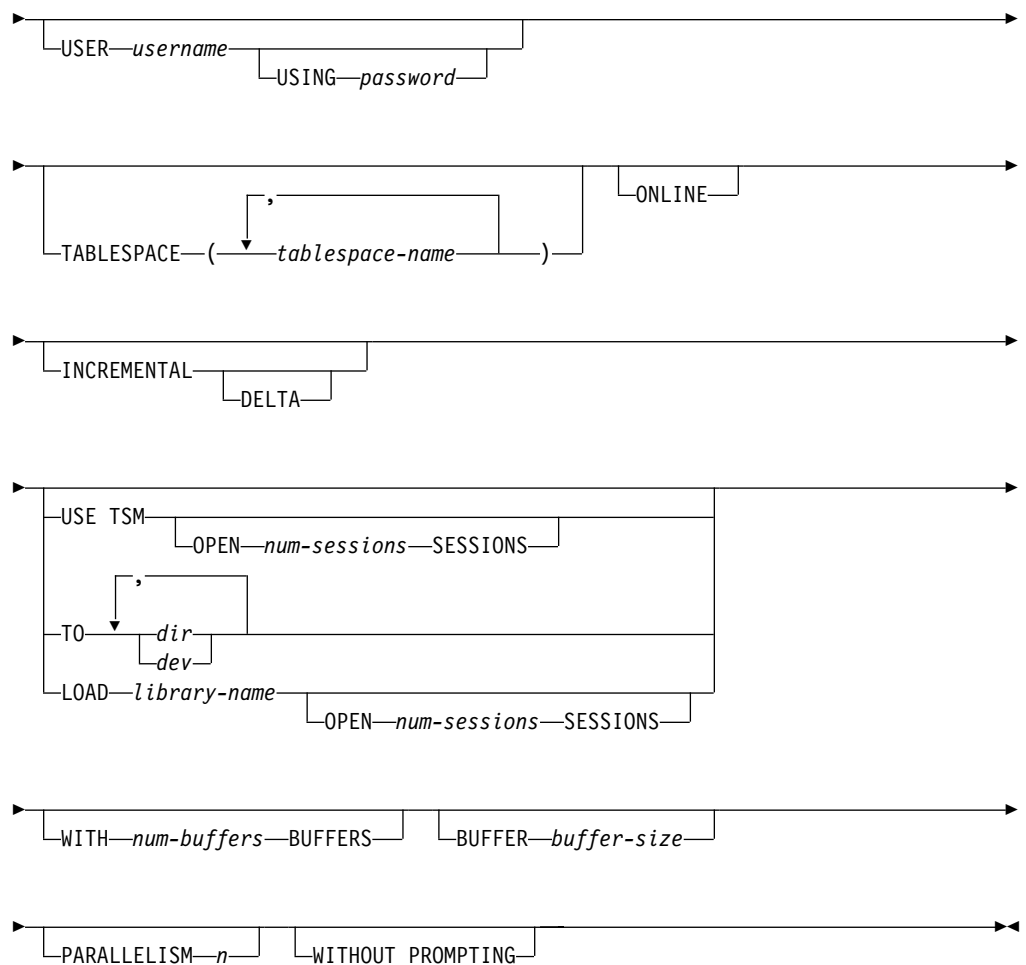
Using this command will cause a database to lose a portion of its LSN space, and thereby hasten the exhaustion of valid LSNs.

BACKUP DATABASE

Syntax Diagram

The syntax diagram for BACKUP DATABASE will be updated to reflect the new INCREMENTAL parameter and the optional DELTA argument. Specifying the INCREMENTAL option alone will result in a cumulative backup image being produced. The optional DELTA argument can be used to specify the production of a non-cumulative backup image.

►►—BACKUP—DATABASE
DB—*database-alias*—►



DB2 Data Links Manager Considerations

If one or more Data Links servers are configured for the database, the backup operation will succeed, even if a Data Links server is not available. When the Data Links server restarts, backup processing will be completed on that Data Links server before it becomes available to the database again.

註: If there are twice as many backups still waiting for an unavailable Data Links server as are retained in the history file for the database (database configuration parameter `num_db_backups`), the backup operation will fail.

BIND

The command syntax for DB2 should be modified to show the federated parameter as follows:

```
FEDERATED--+-NO--+-
          '-YES--'
```

FEDERATED

Specifies whether a static SQL statement in a package references a nickname or

a federated view. If this option is not specified and a static SQL statement in the package references a nickname or a federated view, a warning is returned and the package is created.

NO A nickname or federated view is not referenced in the static SQL statements of the package. If a nickname or federated view is encountered in a static SQL statement during the prepare or bind of this package, an error is returned and the package is *not* created.

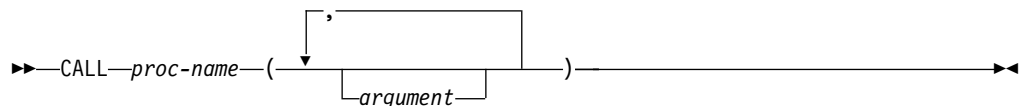
YES A nickname or federated view can be referenced in the static SQL statements of the package. If no nicknames or federated views are encountered in static SQL statements during the prepare or bind of the package, no errors or warnings are returned and the package is created.

Note: In Version 7 FixPak 2, an SQL1179W warning message is generated by the server when precompiling a source file or binding a bind file without specifying a value for the FEDERATED option. The same message is generated when the source file or bind file includes static SQL references to a nickname. There are two exceptions:

- For clients that are at an earlier FixPak than Version 7 FixPak 2 or for downlevel clients, the sqlprep() API does not report this SQL1179W warning in the message file. The Command Line Processor PRECOMPILE command also does not output the warning in this case.
- For clients that are at an earlier FixPak than Version 7 FixPak 2 or for downlevel clients, the sqlabndx API does report this SQL1179W warning in the message file. However, the message file also incorrectly includes an SQL0092N message indicating that no package was created. This is not correct as the package is indeed created. The Command Line Processor BIND command returns the same erroneous warning.

CALL

The syntax for the CALL command should appear as follows:



The description of the *argument* parameter has been changed to:

Specifies one or more arguments for the stored procedure. All input and output arguments must be specified in the order defined by the procedure. Output arguments are specified using the "?" character. For example, a stored procedure foo with one integer input parameter and one output parameter would be invoked as "call foo (4, ?)".

Note:

1. When invoking this utility from an operating system prompt, it may be necessary to delimit the command as follows:

```
"call DEPT_MEDIAN (51)"
```

A single quotation mark (') can also be used.

2. The stored procedure being called must be uniquely named in the database.

3. The stored procedure must be cataloged. If an uncataloged procedure is called, a DB21036 error message is returned.
4. A DB21101E message is returned if not enough parameters are specified on the command line, or the command line parameters are not in the correct order (input, output), according to the stored procedure definition.
5. There is a maximum of 1023 characters for a result column.
6. LOBS and binary data (FOR BIT DATA, VARBINARY, LONGVARBINARY, GRAPHIC, VARGAPHIC, or LONGVARGRAPHIC) are not supported.
7. CALL supports result sets.
8. If an SP with an OUTPUT variable of an unsupported type is called, the CALL fails, and message DB21036 is returned.
9. The maximum length for an INPUT parameter to CALL is 1024.

DROP DATALINKS MANAGER (new command)

DROP DATALINKS MANAGER

Drops a DB2 Data Links Manager from the list of registered DB2 Data Links Managers for a specified database.

Authorization

One of the following:

- *sysadm*
- *sysctrl*
- *sysmaint*

Command Syntax

```
▶ DROP DATALINKS MANAGER FOR DATABASE dbname USING name ▶
```

DB

Command Parameters

DATABASE dbname

Specifies a database name.

USING name

Specifies the name of the DB2 Data Links Manager server as shown by the LIST DATALINKS MANAGER command.

Examples

Example 1

Dropping a DB2 Data Links Manager micky.almaden.ibm.com from database TEST under instance validate residing on host bramha.almaden.ibm.com when some database tables have links to micky.almaden.ibm.com. *It is extremely important that the following steps be taken when dropping a DB2 Data Links Manager.*

1. Take a database backup for database TEST.
2. If there are any links to micky.almaden.ibm.com, unlink them:
 - a. Log on with a user ID belonging to SYSADM_GROUP and obtain an exclusive mode connection to the database TEST.

```
connect to test in exclusive mode
```

Ensure that this is the only connection to test using that user ID. This will prevent any new links from being created.

- b. Obtain a list of all FILE LINK CONTROL DATALINK columns and the tables containing them in the database.

```
select tabname, colname from syscat.columns where substr(dl_features,2,1)='F'
```

- c. For each FILE LINK CONTROL DATALINK column in the list, issue SQL SELECT to determine if links to micky.almaden.ibm.com exist. For example, for a DATALINK column c in table t, the SELECT statement would be:

```
select count(*) from t where dlurlserver(t.c) = 'MICKY.ALMADEN.IBM.COM'
```


- d. For each FILE LINK CONTROL DATALINK column containing such links, issue SQL UPDATE to unlink values which are links to micky.almaden.ibm.com. For example, for a DATALINK column c in table t, the UPDATE statement would be:

```
update t set t.c = null where dlurlserver(t.c)='MICKY.ALMADEN.IBM.COM'
```

If t.c is not nullable, the following can be used instead:

```
update t set t.c = dlvalue('') where dlurlserver(t.c)='MICKY.ALMADEN.IBM.COM'
```

- e. Commit this SQL UPDATE:

```
commit
```

3. Issue the DROP DATALINKS MANAGER command:

```
drop datalinks manager for db test using node micky.almaden.ibm.com
```

4. Terminate the exclusive mode connection to make the changes effective and to allow other connections to the database:

```
terminate
```

5. Initiate unlink processing and garbage collection of backup information for TEST on micky.almaden.ibm.com. As DB2 Data Links Manager Administrator, issue the following command on micky.almaden.ibm.com:

```
dlfm drop_dlm test validate bramha.almaden.ibm.com
```

This will unlink any files that are still linked to database TEST, just in case the user has missed unlinking them before invoking step 3. If micky.almaden.ibm.com has backup information (for example, archive files, metadata) for files previously linked to database TEST, this command will initiate garbage collection of that information. The actual unlinking and garbage collection will be performed asynchronously.

Example 2

Deleting DATALINK values that are links to files on a DB2 Data Links Manager called micky.almaden.ibm.com, when the Manager has already been dropped from database TEST. This may be required if steps in Example 1 were not followed while dropping micky.almaden.ibm.com. SQL DELETE, SELECT, and UPDATE statements will not be successful for such DATALINK values (SQL0368). The user must run a reconcile operation for each table that contains such DATALINK values. Each DATALINK value that was a link to micky.almaden.ibm.com will be updated to NULL or a zero-length DATALINK value. Any row containing such a value will be inserted into the exception table (if one was specified). However, the DATALINK value will not include the prefix name. The prefix name in the original DATALINK value is no longer obtainable by the system, because micky.almaden.ibm.com has been dropped. For example, if the original DATALINK value was 'http://host.com/dlfs/x/y/a.b' and '/dlfs' is the prefix name, the DATALINK value in the exception table will contain 'http://host.com/x/y/a.b'. The files referenced by these DATALINK values will continue to remain in linked state on the DB2 Data Links Manager. The dlfm drop_dlm command can be issued on micky.almaden.ibm.com to initiate unlink processing for these files. If micky.almaden.ibm.com has backup information (for example, archive files, metadata) for files previously linked to database TEST, this command will initiate garbage collection of that information. The actual unlinking and garbage collection will be performed asynchronously.

Example 3

Multiple incarnations of a DB2 Data Links Manager micky.almaden.ibm.com for a database TEST. This scenario demonstrates that a DB2 Data Links Manager can be re-registered after being dropped, and that it is then treated as a completely new DB2 Data Links Manager. The following steps are only illustrative of a scenario that is possible. If, as recommended, the steps in Example 1 are followed for dropping micky.almaden.ibm.com, links to the older incarnation of micky.almaden.ibm.com will not exist; that is, one will not see error SQL0368 in step 7 below.

1. Register micky.almaden.ibm.com to database TEST:

```
add datalinks manager for db test using node micky.almaden.ibm.com port 14578
```

2. Create links to files on micky.almaden.ibm.com:

```
connect to test
create table t(c1 int, c2 datalink linktype url file link control mode
db2options)
insert into t values(1, dlvalue('file://micky.almaden.ibm.com/pictures/
yosemite.jpg'))
commit
terminate
```

3. Drop micky.almaden.ibm.com from database TEST:

```
drop datalinks manager for db test using micky.almaden.ibm.com
```

4. Select DATALINK values:

```
connect to test
select * from t
terminate
```

The user will see:

```
SQL0368 The DB2 Data Links Manager "MICKY.ALMA DEN.IBM.COM"
is not registered to the database. SQLSTATE=55022.
```

5. Register micky.almaden.ibm.com to database TEST again:

```
add datalinks manager for db test using node micky.almaden.ibm.com port 14578
```

6. Insert more DATALINK values:

```
connect to test
insert into t values(2, dlvalue('file://micky.almaden.ibm.com/pictures/
tahoe.jpg'))
commit
```

7. Select DATALINK values:

```
select c2 from t where c1 = 2
```

is successful because the value being selected is a link to the currently registered incarnation of micky.almaden.ibm.com.

```
select c2 from t where c1 = 1
```

returns:

```
SQL0368 The DB2 Data Links Manager "MICKY.ALMA DEN.IBM.COM"
is not registered to the database. SQLSTATE=55022.
```

because the value being selected is a link to the incarnation of micky.almaden.ibm.com which was dropped in step 3 above.

Usage Notes

The effects of the DROP DATALINKS MANAGER command cannot be rolled back. It is extremely important to follow the steps outlined in Example 1 when using the DROP DATALINKS MANAGER command.

This command is effective only after all applications have been disconnected from the database.

Upon successful completion of the command, the user is informed (DB210201I) that no processing has been done on the DB2 Data Links Manager. Before dropping a DB2 Data Links Manager, the user must ensure that the database does not have any links to files on that DB2 Data Links Manager. If links do exist in the database after a DB2 Data Links Manager has been dropped, the user must run the reconcile utility to get rid of them. The reconcile utility will set these links to NULL (if the DATALINK column is nullable), or to a zero-length DATALINK value.

Files corresponding to links between a database and a dropped DB2 Data Links Manager remain in linked state. That is, they are inaccessible to operations like read, write, rename, delete, change of permissions, or change of ownership.

Archived copies of unlinked files on the DB2 Data Links Manager will not be garbage collected by this command. However, users can explicitly initiate unlink processing and garbage collection using the dlfm drop_dlm command on the DB2 Data Links Manager.

It is recommended that a database backup be taken before dropping a DB2 Data Links Manager. In addition, ensure that all replication subscriptions have replicated all changes involving this DB2 Data Links Manager.

If a backup was taken before the DB2 Data Links Manager was dropped from a database, and that backup image is used to restore after that DB2 Data Links Manager was dropped, restore or rollforward processing may put certain tables in datalink reconcile pending (DRP) state.

EXPORT

In the section "DB2 Data Links Manager Considerations", Step 3 of the procedure to ensure that a consistent copy of the table and the corresponding files referenced by DATALINK columns are copied for export should read:

3. Run the dlfm_export utility at each Data Links server. Input to the dlfm_export utility is the control file name, which is generated by the export utility. This produces a tar (or equivalent) archive of the files listed within the control file. For Distributed File Systems (DFS), the dlfm_export utility will get the DCE network root credentials before archiving the files listed in the control file. dlfm_export does not capture the ACLs information of the files that are archived.

In the same section, the bullets following "Successful execution of EXPORT results in the generation of the following files" should be modified as follows:

The second sentence in the first bullet should read:

A DATALINK column value in this file has the same format as that used by the import and load utilities.

The first sentence in the second bullet should read:

Control files `server_name`, which are generated for each Data Links server. (On the Windows NT operating system, a single control file, `ctrlfile.lst`, is used by all Data Links servers. For DFS, there is one control file for each cell.)

The following sentence should be added to the paragraph before Table 5:

For more information about `difm_export`, refer to the "Data Movement Utilities Guide and Reference" under "Using Export to move DB2 Data Links Manager Data".

GET DATABASE CONFIGURATION

The description of the `DL_TIME_DROP` configuration parameter should be changed to the following:

Applies to DB2 Data Links Manager only. This parameter specifies the number of days files would be retained on an archive server (such as a TSM server) after a `DROP DATABASE` command is issued.

The new parameter `TRACKMOD` will be added to the `GET DATABASE CONFIGURATION` command. The syntax will appear as follows:

```
▶▶ GET DATABASE CONFIGURATION FOR database-alias
    |   |
    |   +--- DB
    |   +--- CONFIG
    |   +--- CFG
```

```
▶ TRACKMOD NO
           |
           +--- YES
```

The parameter description will be added as follows:

TRACKMOD

Indicates whether DB2 should track modified pages in the database in order to allow incremental backups to be taken.

OFF Specifies that changed pages should not be tracked. This is the default for databases created prior to Version 7.1, FixPak 3.

ON Specifies that changed pages should be tracked. When this parameter is set, incremental backups of the database can be made. This is the default for databases created with Version 7.1, FixPak 3 or later.

GET ROUTINE (new command)

GET ROUTINE

Retrieves a routine SQL Archive (SAR) file for a specified SQL routine.

Authorization

dbadm

Required Connection

Database. If implicit connect is enabled, a connection to the default database is established.

Command Syntax

```
► GET ROUTINE INTO file_name FROM SPECIFIC PROCEDURE routine_name ◄
```

Command Parameters

INTO file-name

Names the file where routine SQL archive (SAR) is stored.

FROM Indicates that start of the specification of the routine to be retrieved.

SPECIFIC

The specified *routine-name* is given as a specific name.

PROCEDURE

The routine is an SQL procedure.

routine-name

The name of the procedure. If SPECIFIC is specified then it is the specific name of the procedure. If the name is not qualified with a schema name, the CURRENT SCHEMA is used as the schema name of the routine. The *routine-name* must be an existing procedure that is defined as an SQL procedure.

Examples

```
GET ROUTINE INTO procs/proc1.sar FROM PROCEDURE myappl.proc1;
```

GET SNAPSHOT

The description for the FCM FOR ALL NODES parameter should appear as follows:

Provides Fast Communication Manager (FCM) statistics between the node against which the GET SNAPSHOT command was issued and the other nodes in the EEE instance.

IMPORT

In the section "DB2 Data Links Manager Considerations", the following sentence should be added to Step 3:

For Distributed File Systems (DFS), update the cell name information in the URLs (of the DATALINK columns) from the exported data for the SQL table, if required.

The following sentence should be added to Step 4:

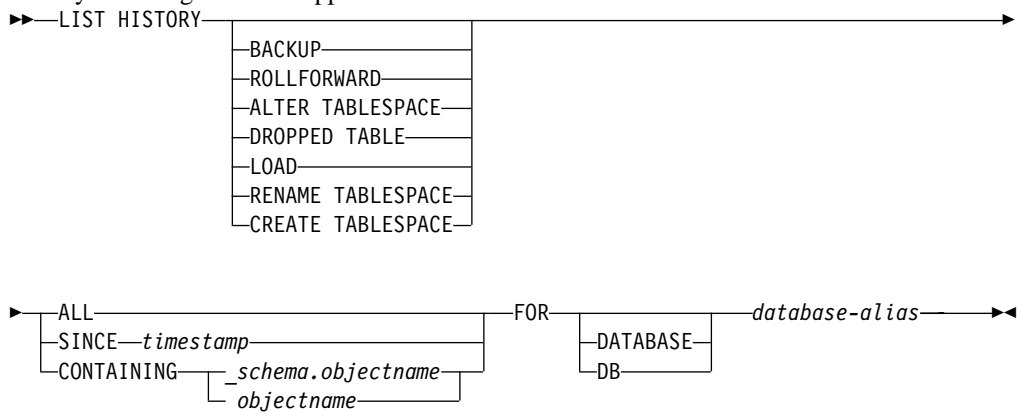
For DFS, define the cells at the target configuration in the DB2 Data Links Manager configuration file.

The paragraph following Step 4 should read:

When the import utility runs against the target database, files referred to by DATALINK column data are linked on the appropriate Data Links servers.

LIST HISTORY

The CREATE TABLESPACE parameter will be added to the LIST HISTORY command. The syntax diagram will appear as follows:



The parameter description will be added as follows:

CREATE TABLESPACE

Lists all CREATE TABLESPACE and DROP TABLESPACE operations.

The Usage Notes will be updated as follows:

The following symbols will be added to the Operation section of the report:

- A - Create tablespace
- O - Drop tablespace
- U - Unload

The symbols under the Type section of the report will be reorganized as follows:

- Backup Types
 - F - Offline
 - N - Online
 - I - Incremental Offline
 - O - Incremental Online
 - D - Delta Offline
 - E - Delta Online
- Rollforward Types
 - E - End of log
 - P - Point in time

- Load Types
 - I - Insert
 - R - Replace
- Alter tablespace Types
 - C - Add containers
 - R - Rebalance
- Quiesce Types
 - S - Quiesce Share
 - U - Quiesce Update
 - X - Quiesce Exclusive
 - Z - Quiesce Reset

LOAD

In the section "DB2 Data Links Manager Considerations", add the following sentence to Step 1 of the procedure that is to be performed before invoking the load utility, if data is being loaded into a table with a DATALINK column that is defined with FILE LINK CONTROL:

For Distributed File Systems (DFS), ensure that the DB2 Data Links Managers within the target cell are registered.

The following sentence should be added to Step 5:

For DFS, register the cells at the target configuration referred to by DATALINK data (to be loaded) in the DB2 Data Links Manager configuration file.

In the section "Representation of DATALINK Information in an Input File", the first note following the parameter description for `urlname` should read:

Currently "http", "file", "unc", and "dfs" are permitted as a schema name.

The first sentence of the second note should read:

The prefix (schema, host, and port) of the URL name is optional. For DFS, the prefix refers to the schema cellname filespace-junction portion.

In the DATALINK data examples for both the delimited ASCII (DEL) file format and the non-delimited ASCII (ASC) file format, the third example should be removed.

The DATALINK data examples in which the load or import specification for the column is assumed to be `DL_URL_DEFAULT_PREFIX` should be removed and replaced with the following:

Following are DATALINK data examples in which the load or import specification for the column is assumed to be `DL_URL_REPLACE_PREFIX` ("http://qso"):

- `http://www.almaden.ibm.com/mrep/intro.mpeg`

This sample URL is stored with the following parts:

- schema = http
- server = qso

- path = /mrep/intro.mpeg
- comment = NULL string
- /u/me/myfile.ps
 - This is stored with the following parts:
 - schema = http
 - server = qso
 - path = /u/me/myfile.ps
 - comment = NULL string

PING (new command)

PING

Tests the network response time of the underlying connectivity between a client and a database server where DB2 Connect is used to establish the connection.

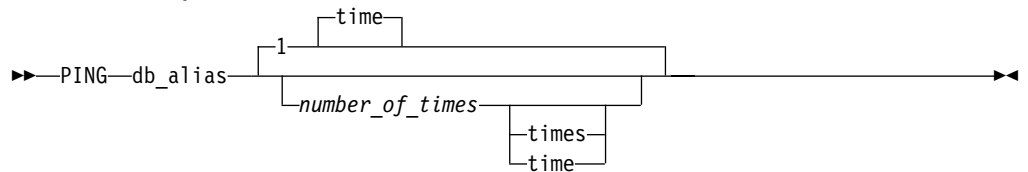
Authorization

None

Required Connection

Database

Command Syntax



Command Parameters

db_alias

Specifies the database alias for the database on a DRDA server that the ping is being sent to.

註: This parameter, although mandatory, is not currently used. It is reserved for future use. Any valid database alias name can be specified.

number of times

Specifies the number of iterations for this test. The value must be between 1 and 32767 inclusive. The default is 1. One timing will be returned for each iteration.

Examples

To test the network response time for the connection to the host database server hostdb once:

```
db2 ping hostdb 1  
or:  
db2 ping hostdb
```

The command will display output that looks like this:

```
Elapsed time: 7221 microseconds
```

To test the network response time for the connection to the host database server hostdb 5 times:

```
db2 ping hostdb 5  
or:  
db2 ping hostdb 5 times
```

The command will display output that looks like this:

Elapsed time: 8412 microseconds
Elapsed time: 11876 microseconds
Elapsed time: 7789 microseconds
Elapsed time: 10124 microseconds
Elapsed time: 10988 microseconds

Usage Notes

A database connection must exist before invoking this command, otherwise an error will result.

The elapsed time returned is for the connection between the client and a DRDA server database via DB2 Connect.

PUT ROUTINE (new command)

PUT ROUTINE

Uses the specified routine SQL Archive (SAR) file to define a routine in the database.

Authorization

dbadm

Required Connection

Database. If implicit connect is enabled, a connection to the default database is established.

Command Syntax

```
►► PUT ROUTINE FROM file-name [OWNER new-owner] [USE REGISTERS] ►►
```

Command Parameters

FROM *file-name*

Names the file where routine SQL archive (SAR) is stored.

OWNER *new-owner*

Specifies a new authorization-name for the routine that will be used for authorization checking of the routine. The new-owner must have the necessary privileges for the routine to be defined. If the OWNER clause is not specified, the authorization-name that originally defined the routine is used.

USE REGISTERS

Indicates that the CURRENT SCHEMA and CURRENT PATH special registers are used to define the routine. If this clause is not specified, the settings for the default schema and SQL path are the settings used when the routine was originally defined. CURRENT SCHEMA is used as the schema name for unqualified object names in the routine definition (including the name of the routine) and CURRENT PATH is used to resolve unqualified routines and data types in the routine definition.

Syntax

```
PUT ROUTINE FROM procs/proc1.sar;
```

Usage Notes

No more than one procedure can be concurrently installed under a given schema.

RECONCILE

The following usage note should be added to the command description:

During reconciliation, attempts are made to link files which exist according to table data, but which do not exist according to Data Links Manager metadata, if no other conflict exists. A required DB2 Data Links Manager is one which has a referenced DATALINK value in the table. Reconcile tolerates unavailability of a required DB2 Data Links Manager as well as those that are configured to the database but are not part of the table data.

If an exception table is not specified, the exception report file (*filename.exp*) will have the host name, file name, column ID, and reason code for each of the DATALINK column values for which file references could not be re-established. If the file reference could not be re-established because the DB2 Data Links Manager itself was dropped from the database using the DROP DATALINKS MANAGER command, the file name reported in the exception report file is not the full file name; that is, the prefix part is missing. In the exception table for the DATALINK values whose DB2 Data Links Manager is dropped or is not available, the file name in the DATALINK value is not the full file name. The prefix part is missing. For example, if the original DATALINK value was 'http://host.com/dlfs/x/y/a.b', the value reported in the exception table will be 'http://host.com/x/y/a.b'; that is, the prefix name 'dlfs' will not be included. The exception report file in this case will have 'x/y/a.b'; that is, the prefix name 'dlfs' will not be included.

At the end of the reconciliation process, the table is taken out of datalink reconcile pending (DRP) state only if reconcile processing is complete on all the required DB2 Data Links Managers. If reconcile processing is pending on any of the required DB2 Data Links Managers (because they were unavailable), the table will remain, or be placed, in DRP state.

The following should be added to the list of possible violations:

```
00010-DB2 Data Links Manager referenced by the DATALINK value
has been dropped from the database using the DROP DATALINKS
MANAGER command. In this case, the corresponding DATALINK value
in the exception table will not contain the prefix name.
For example, if the original DATALINK value was
'http://host.com/dlfs/prfx/x/y/a.b', and '/dlfs/prfx' is
the prefix name, the exception table will contain
'http://host.com/x/y/a.b'.
```

REORGANIZE TABLE

The following sentence will be added to the Usage Notes:

REORGANIZE TABLE cannot use an index that is based on an index extension.

RESTORE DATABASE

Syntax

The following option will be added to the syntax of the RESTORE DATABASE command after the TABLESPACE/TABLESPACE ONLINE/HISTORY FILE options:



The parameter descriptions will be added as follows:

INCREMENTAL

Results in a manual cumulative restore of the database. The user will issue each of the restore commands.

ABORT

This parameter should be used to terminate an incremental restore before successful completion.

The following examples will also be added:

The following is a sample weekly incremental backup strategy with a recoverable database. A full backup is scheduled once per week, a delta every day, plus an incremental mid-week:

```
(Sun) backup db kdr use adsm
(Mon) backup db kdr online incremental delta use adsm
(Tue) backup db kdr online incremental delta use adsm
(Wed) backup db kdr online incremental use adsm
(Thu) backup db kdr online incremental delta use adsm
(Fri) backup db kdr online incremental delta use adsm
(Sat) backup db kdr online incremental use adsm
```

For a manual database restore of images created above on Friday morning, issue the following commands:

```
restore db kdr incremental taken at (Thu)
restore db kdr incremental taken at (Sun)
restore db kdr incremental taken at (Wed)
restore db kdr incremental taken at (Thu)
```

Notes: Any RESTORE command of the form `db2 restore db <name>` will perform a full database restore, regardless of whether the image being restored is a database image or a table space image. Any RESTORE command of the form `db2 restore db <name> tablespace` will perform a table space restore of the table spaces found in the image. Any RESTORE command in which a list of table spaces is provided will perform a restore of whatever table spaces were explicitly listed.

DB2 Data Links Manager Considerations

The second paragraph in the section entitled "DB2 Data Links Manager Considerations" should be replaced with:

```
If one or more Data Links servers are configured for the database,
the restore operation will succeed, even if a Data Links server is not available.
When the Data Links server restarts, restore processing will be completed
on that Data Links server before it becomes available to the database again.
NOTE: If a database restore operation is still waiting for an unavailable
Data Links server, any subsequent database or table space restore operations
will fail.
```

ROLLFORWARD DATABASE

The second paragraph in the section entitled "DB2 Data Links Manager Considerations" should be replaced with:

```
If one or more Data Links servers are configured for the database,
the rollforward operation will succeed, even if a Data Links server is not available.
When the Data Links server restarts, rollforward processing will be completed
on that Data Links server before it becomes available to the database again.
```

Documentation Error in CLP Return Codes

In the Command Line Processor Return Codes section of Chapter 2, the second paragraph should appear as follows:

|
| For example, the following Bourne shell script executes the GET DATABASE MANAGER
| CONFIGURATION command, then inspects the CLP return code:

```
| db2 get database manager configuration  
| if [ "$?" = "0" ]  
| then echo "OK!"  
| fi
```

Data Movement Utilities Guide and Reference(無中文版)

Chapter 2. Import

Using Import with Buffered Inserts

The note at the end of this section should read:

註: In all environments except EEE, the buffered inserts feature is disabled during import operations in which the INSERT_UPDATE parameter is specified.

Chapter 3. Load

Pending States After a Load Operation

The first two sentences in the last paragraph in this section have been changed to the following:

The fourth possible state associated with the load process (check pending state) pertains to referential and check constraints, DATALINKS constraints, AST constraints, or generated column constraints. For example, if an existing table is a parent table containing a primary key referenced by a foreign key in a dependent table, replacing data in the parent table places both tables (not the table space) in check pending state.

Load Restrictions and Limitations

The following restrictions apply to generated columns and the load utility:

- It is not possible to load a table having a generated column in a unique index unless the generated column is an "include column" of the index or the generatedoverride file type modifier is used. If this modifier is used, it is expected that all values for the column will be supplied in the input data file.
- It is not possible to load a table having a generated column in the partitioning key unless the generatedoverride file type modifier is used. If this modifier is used, it is expected that all values for the column will be supplied in the input data file.

totalreespace File Type Modifier

The totalreespace file type modifier (LOAD) has been modified to accept a value between 0 and 2 147 483 647.

Chapter 4. AutoLoader

rexecd Required to Run Autoloader When Authentication Set to YES

In the Autoloader Options section the following note will be added to the AUTHENTICATION and PASSWORD Parameters description:

| In a Linux environment, if you are running the autoloader with the authentication option
| set to YES, rexecd must be enabled on all machines. If rexecd is not enabled the following
| error message will be generated:

| openbreeze.torolab.ibm.com: Connection refused
| SQL6554N An error occurred when attempting to remotely execute a process.

| The following error messages will be generated in the db2diag.log file:

| 2000-10-11-13.04.16.832852 Instance:svtdbm Node:000
| PID:19612(db2atld) Appid:
| oper_system_services sqloRemoteExec Probe:31

Replication 指南與參考手冊

抄寫及非 IBM 伺服器

使用 DataJoiner 版本 2 或更新版來抄寫資料至非 IBM 伺服器，或自非 IBM 伺服器抄寫資料，如 Informix、Microsoft SQL Server、Oracle、Sybase、Sybase SQL Anywhere。您不可以使用此類型抄寫的關聯式連接函數，因為 DB2 Relational Connect 版本 7 沒有更新的能力。對現存的 DB2 和 DataJoiner 版本所有的平台（AS/400、OS/2、OS/390、UNIX 及 Windows），您必須使用 DJRA（DataJoiner 抄寫管理）來管理特殊的抄寫。

在 Windows 2000 上的抄寫

DB2 DataPropagator 版本 7 與 Windows 2000 作業系統相容。

儲存 SQL 檔案時的已知錯誤

若在 DB2 Connect Personal Edition 中使用「控制中心」，您無法儲存 SQL 檔案。若試圖儲存 SQL 檔案，您將收到一則指示 DAS 不在作用中的錯誤訊息，事實上 DAS 沒有作用，因為它並沒有隨 DB2 Connect PE 出貨。

DB2 維護

建議您針對在抄寫環境中使用的各種 DB2 產品安裝最新的 DB2 維護。

Web 上的 Data Difference Utility

您可以從 Web 下載 Data Difference utility (DDU)，網址如下：
<ftp://ftp.software.ibm.com/ps/products/datapropagator/fixes/>。DDU 是一個範例公用程式，供您比較相同檔案的兩個不同版本，以及產生輸出檔來顯示版本之間的差異。詳細資訊，請參閱此範例公用程式所附的 README 檔。

第 3 章 資料抄寫實務範例

抄寫實務

請參閱 DataPropagator 網站的 Library 頁面 (<http://www.ibm.com/software/data/dpropr/>) 以取得不同的新資料抄寫實務範例。遵照實務範例中的步驟，從 AIX 上的 Oracle 資料庫抄寫來源表格將變更複製到 DB2 for Windows NT 上資料庫中的目標表格。那個實務範例使用 DB2 DataJoiner Replication Administration (DJRA) 工具、擷取觸發函式、引用程式及 DB2 DataJoiner。

在書的第 44 頁，步驟 6 中有關建立通行碼檔案的指示應該如下：

步驟 6：建立通行碼檔案

因為引用程式需要連接到來源伺服器，您必須建立一個通行碼檔案來做使用者身分驗證。請確定用來執行引用程式的使用者 ID 可以讀取通行碼檔案。

若要建立通行碼檔案：

1. 從 Windows NT 指令提示視窗，變更目錄到 C:\scripts。
2. 在此目錄中建立一個叫做 DEPTQUAL.PWD 的新檔案。您可以使用任何文字編輯程式來建立這個檔案，比如 Notepad。通行碼檔案的命名慣例為 *applyqual.pwd*；其中 *applyqual* 字串是區分大小寫的，必須符合您在建立定期抄寫設定時所用的引用限定元的字體和值。在此實務範例中，引用限定元是 DEPTQUAL。

註：也支援 DB2 DataPropagator 版本 5 以來的檔案命名慣例。

3. 通行碼檔案的內容格式如下：

```
SERVER=server USER=userid PWD=password
```

其中：

伺服器

指來源、目標或控制伺服器的名稱，必須與定期抄寫設定表格中所顯示的完全一致。在此實務範例中，這些名稱為 SAMPLE 及 COPYDB。

userid

您計劃用來管理該特定資料庫的使用者 ID。若是 Windows NT 及 UNIX 作業系統，這個值是區分大小寫的。

password

與使用者 ID 相關的通行碼。若是 Windows NT 及 UNIX 作業系統，則這個值有區分大小寫。

請不要在這個檔案中加入空白行或註解行。僅新增伺服器名稱、使用者 ID 和通行碼資訊。

4. 通行碼檔案的內容看起來應該類似：

```
SERVER=SAMPLE USER=subina PWD=subpw  
SERVER=COPYDB USER=subina PWD=subpw
```

關於 DB2 身分驗證與機密保護的詳細資訊，請參閱 *IBM DB2 Administration Guide*。

第 5 章 抄寫規劃

表格及直欄名稱

抄寫不支援表格及直欄名稱中的空白。

DATALINK 抄寫

在 Solaris 上，DATALINK 抄寫可作為版本 7.1 FixPak 1 的一部份。它需要一個 FTP 常駐程式，在來源及目標 DATALINK 檔案系統中執行，並支援 MDTM (modtime) 指令，來顯示給定檔案的前次修改時間。如果您用的是版本 2.6 的 Solaris 作業系統，或任何不包括 MDTM 的 FTP 支援，則您需要附加的軟體，比如 WU-FTPD。

您不能在 AS/400 和其他平台之間的 DB2 資料庫抄寫 DATALINK 直欄。

在 AS/400 平台沒有支援 DATALINK 值「註解」屬性的抄寫。

執行 AIX 4.2 時，在您執行預設的使用者跳出程式 (ASNDLCOPY) 之前，必須安裝 APAR IY03101 (AIX 4210-06 建議維護 AIX 4.2.1) 的 PTF。這個 PTF 包括了 FTP 常駐程式內 "modtime/MDTM" 指令的 Y2K修正程式。欲驗證該修正程式，請檢查由 "modtime <file>" 指令所傳回來的前次修改時間，其中 <file> 為 2000 年 1 月 1 日之後所修改的檔案。

如果目標表格是外部 CCD 表格，DB2 DataPropagator 會呼叫 ASNDLCOPY 常式抄寫 DATALINK 檔案。關於 ASNDLCOPY 和 ASNDLCOPYDF 的使用，請參閱各程式原始碼的序言。The following restrictions apply:

- 內部 CCD 表格含有 DATALINK 指示符，但沒有 DATALINK 值。
- 壓縮外部 CCD 表格含有 DATALINK 值。
- 非壓縮 CCD 目標表格不能有 DATALINK 直欄。
- 來源和目標伺服器相同時，定期抄寫設定不能有 DATALINK 直欄的成員。

LOB 限制

壓縮內部 CCD 表格不能和 LOB 直欄和 LOB 指示符有關聯。

規劃抄寫

在第 65 頁，「連通性」應該包括下列情形：

如果引用程式和控制伺服器不能連接，引用程式會終止。

使用 AS/400 的資料區塊，請確定間隔期間所抄寫的資料總量不超出 4 百萬列數，不能如本書內 69 頁中所提的 4 MB。

第 6 章 設定抄寫環境

隨處更新 (Update-anywhere) 先決條件

如果您要使用衝突偵測和定期抄寫設定內多達 150 個的定期抄寫設定成員來設定隨處更新抄寫，則必須執行下列的 DDL，以建立控制伺服器的 ASN.IBMSNAP_COMPENSATE 表格。

```
CREATE TABLE ASN.IBMSNAP_COMPENSATE (  
    APPLY_QUAL char(18) NOT NULL,  
    MEMBER SMALLINT,  
    INTENTSEQ CHAR(10) FOR BIT DATA,  
    OPERATION CHAR(1));
```

設定抄寫環境

第 95 頁，「自行設定 CD 表格、索引和表格空間名稱」說明：DPREPL.DFT 檔案不是位於 \sql1lib\bin 目錄，就是位於 \sql1lib\java 目錄。這是不正確的，DPREPL.DFT 位於 \sql1lib\cc 目錄。

第 128 頁，保留限制說明：陳述當「擷取程式」暖開機或使用「擷取程式」刪改指令時，才使用保留限制來刪改列數。若以自動刪改選項啟動「擷取程式」，則它不會使用保留限制來刪改列數。

第 8 章 問題判斷

抄寫分析程式在 Windows 32 位元系統和 AIX 上執行。欲在 AIX 上執行分析程式，請確定 PATH 環境變數內 sqllib/bin 目錄出現在 /usr/local/bin 前，以避免和 /usr/local/bin/analyze 發生衝突。

抄寫分析程式有兩個附加選用的關鍵字：CT 和 ATT。

CT=*n* 只顯示擷取追蹤表格內 *n* 天之後的登錄。此關鍵字為選用的。若您沒有指定此關鍵字，其預設值為 7 天。

AT=*n* 只顯示引用追蹤表格內 *n* 天之後的登錄。此關鍵字為選用的。若您沒有指定此關鍵字，其預設值為 7 天。

範例：

```
analyze mydb1 mydb2 f=mydirectory ct=4 at=2 deepcheck q=applyqual1
```

下列為「抄寫分析程式」之最新關鍵字資訊：

deepcheck

指定分析程式執行較完整的分析包括下列資訊：CD 和 UOW 刪改資訊、DB2 for OS/390 表格空間分割和壓縮細節、關於定期抄寫設定的目標索引之分析、定期抄寫時間，以及定期抄寫設定 SQL 陳述式錯誤。分析包括所有的伺服器。此關鍵字為選用的。

lightcheck

指定自報告中除去下列資訊：ASN.IBMSNAP_SUBS_COLS 表格所有直欄明細、定期抄寫錯誤、不規則、遺漏，不正確或無效的索引。此資訊中的縮減儲存資源和產生一個較小的 HTML 輸出檔。此關鍵字和 deepcheck 關鍵字相互排斥。

AS/400 平台的 PTF 抄寫有分析程式工具。這些工具收集有關抄寫環境的資訊，同時，產生一個傳送到 IBM 服務中心代表的 HTML 檔案，以輔助問題的決定。欲得到 AS/400 工具，下載適當的 PTF (例如，產品 5769DP2 就須下載 PTF SF61798 或更新的版本)。

新增下列問題和解決方案至「疑難排解」一節中：

問題：沒有抄寫變更的情況下，引用程式形成迴圈；
引用追蹤表顯示 STATUS=2。
定期抄寫設定包括多個來源表格。若要改進設定中單一來源表格的熱點處理，就會為該來源表格定義一個內部 CCD 表格 (但是在不同的定期抄寫設定中)。會對來源表格做出變更，但移入內部 CCD 表格的「引用」程序會非同步的執行 (例如，引用程式可能尚未啟動或事件尚未觸發等)。從來源表格抄寫更新到目標表格的引用程式會進入迴圈，因為它在等待內部 CCD 表格被更新。
若要停止迴圈，啟動內部 CCD 表格的引用程式 (或觸發會導致抄寫發生的事件)。引用程式會移入內部 CCD 表格，並容許循環中的引用程式處理所有來源表格來的變更。
含有來源表格的定期抄寫設定會發生類似的狀況，其中內部 CCD 表格由多重的引用程式移入。

第 9 章 AS/400 的擷取程式和引用程式

在 178 頁，「工作管理的備註」應如下所示：

您可以變更預設定義或提供個人定義。
建立個人子系統說明，必須替子系統 QZSNDPR 命名，並且在 QDPR 以外的檔案庫建立。
有關定義的變更，詳細資料請參閱「OS/400 工作管理 V4R3」。

新增下列至「驗證和自行設定 AS/400 下 DB2 DataPropagator 的安裝」(第 178 頁)：

如果因為高量的異動而有鎖定競爭的問題，則您可將預設的等待逾時值從 30 增加到 120。您可在每次擷取工作開始時變更工作，或您可使用下列程序來變更所有在子系統中執行的工作的預設等待逾時值：

1. 發出下列指令，重複 QGPL/QBATCHI 以建立新的類別物件：

```
CRTDUPOBJ OBJ(QBATCH) FROMLIB(QGPL) OBJTYPE(*CLS) TOLIB(QDPR) NEWOBJ(QZSNDPR)
```

2. 變更最新建類別的等待逾時值 (例如，變更至 300 秒)：

```
CHGCLS CLS(QDPR/QZSNDPR) DFTWAIT(300)
```

3. 利用新建類別更新子系統說明 QDPR/QZSNDPR 內的路徑登錄：

```
CHGRTGE SBS(D(QDPR/QZSNDPR) SEQNBR(9999) CLS(QDPR/QZSNDPR)
```

第 194 頁的「使用刪除異動日誌接收器結束常式」應併入這個句子：若您移除刪除異動日誌接收器結束常式，請確定用於來源表格的所有異動日誌皆有 DLTRCV(*NO)。

在 195 頁，ADDEXITPGM 指令參數應讀取下列所示：

```
ADDEXITPGM EXITPNT(QIBM_QJO_DLT_JRNRCV)
            FORMAT(DRCV0100)
            PGM(QDPR/QZSNDREP)
            PGMNBR(*LOW)
            CRTEXITPNT(*NO)
            PGM(DTA(65535 10 QSYS)
```

第 10 章 OS/390 的擷取程式和引用程式

在第 10 章中更新下列段落：

DB2 DataPropagator for OS/390 的先決條件

您必須具備 DB2 for OS/390 版本 5、DB2 for OS/390 版本 6 或 DB2 for OS/390 版本 7，才能執行 DB2 DataPropagator for OS/390 版本 7 (V7)。

OS/390 上的 UNICODE 和 ASCII 編碼方法

DB2 DataPropagator for OS/390 V7 支援 UNICODE 和 ASCII 編碼方法。若要開發新的編碼方法，您必須具備 DB2 for OS/390 V7，而且必須手動建立或轉換 DB2 DataPropagator 來源、目標及控制表格，我們將會在下列各節中說明。不過，即使您沒有修改任何編碼方法，您現存的抄寫環境將使用 DB2 DataPropagator for OS/390 V7。

選擇編碼方法

若您的來源、CD 及目標表格使用相同的編碼方法，您可以將抄寫環境中的資料轉換需求降至最低。對表格選擇編碼方法時，請遵循單一 CCSID 規則：表格空間中的字元資料可以用 ASCII、UNICODE 或 EBCDIC 來編碼。表格空間內的所有表格必須使用相同的編碼方法。SQL 陳述式中所有表格的編碼方法必須相同。同時，您在檢視及結合中所用的全部表格必須相同的編碼方法。

若未遵循單一 CCSID 規則，則在連結或執行期間，DB2 將會偵測違規並傳回 SQLCODE -873。哪些表格應該是 ASCII 或 UNICODE，視您的主/從架構而定。特別地，當選擇表格的編碼方法時，請遵循這些規則：

- DB2 for OS/390 上的來源或目標表格可以是 EBCDIC、ASCII 或 UNICODE。它們可以在使用任何支援的 DBMS (DB2 系列，或具有 DataJoiner 的非 DB2 系列) 中相同或不同編碼方法的表格之間複製。

- 在 DB2 for OS/390 來源伺服器上，相同伺服器上所有 CD、UOW、登記及刪改控制表格必須使用相同的編碼方法。為確保一致性，請務必明確指定編碼方法。
- 相同控制伺服器上的所有控制表格 (ASN.IBMSNAP_SUBS_xxxx) 必須使用相同的編碼方法。
- 其它控制表格可使用任何編碼方法；不過，建議 ASN.IBMSNAP_CRITSEC 表格保留 EBCDIC。

設定編碼方法

若要對表格指定適當的編碼方法，請修改用來產生表格的 SQL：

- 以適當的編碼方法建立新的來源和目標表格，或變更現存目標及來源表格的編碼方法。建議您變更現存表格的編碼方法之前停止「擷取程式」和「引用程式」，然後以冷開機方式啟動「擷取程式」並重新啟動「引用程式」。變更現存表格的編碼方法：

1. 使用 Reorg 公用程式複製現存的表格。
2. 捨棄現存的表格。
3. 指定新編碼方法來重建表格。
4. 使用 Load 公用程式將原始資料載入新表格中。

有關 Load 和 Reorg 公用程式的詳細資訊，請參閱 *DB2 Universal Database for OS/390 Utility Guide and Reference*。

- 以適當的編碼方法建立新控制表格或修改現存表格的編碼方法。

DPCNTL.MVS 檢附於 DB2 for OS/390 的 sqllib\samples\repl 中，它包含建立控制表格的數個 CREATE TABLE 陳述式。對於必須是 ASCII 或 UNICODE 的表格而言（例如 ASN.IBMSNAP_REGISTER 和 ASN.IBMSNAP_PRUNCNTL），請加入 CCSID ASCII 或 CCSID UNICODE 關鍵字，如下列範例所示。

```
CREATE TABLE ASN.IBMSNAP_PRUNCNTL (
  TARGET_SERVER      CHAR( 18)          NOT NULL,
  TARGET_OWNER       CHAR( 18)          NOT NULL,
  TARGET_TABLE       CHAR( 18)          NOT NULL,
  SYNCHTIME          TIMESTAMP,
  SYNCHPOINT         CHAR( 10)          FOR BIT DATA,
  SOURCE_OWNER       CHAR( 18)          NOT NULL,
  SOURCE_TABLE       CHAR( 18)          NOT NULL,
  SOURCE_VIEW_QUAL   SMALLINT          NOT NULL,
  APPLY_QUAL         CHAR( 18)          NOT NULL,
  SET_NAME           CHAR( 18)          NOT NULL,
  CNTL_SERVER        CHAR( 18)          NOT NULL,
  TARGET_STRUCTURE   SMALLINT          NOT NULL,
  CNTL_ALIAS         CHAR( 8)
) CCSID UNICODE
DATA CAPTURE CHANGES
  IN TSSNAP02;
```

若要修改現存的控制表格和 CD 表格，請使用 Reorg 和 Load 公用程式。

- 當建立新抄寫來源或定期抄寫設定時，請修改管理工具所產生的 SQL 檔案，以指定適當的編碼方法。此 SQL 檔案具有數個 CREATE TABLE 陳述式，可用來分別針對抄寫來源及定期抄寫設定建立 CD 和目標表格。請在適當位置加入關鍵字 CCSID ASCII 或 CCSID UNICODE。例如：

```
CREATE TABLE user1.cdtable1 (
  employee_name      varchar,
  employee_age       decimal
) CCSID UNICODE;
```

第 11 章 UNIX 平台的擷取程式和引用程式

在 UNIX 和 Windows 上設定「擷取程式」和「引用程式」的環境變數

若您使用預設字碼頁值以外的字碼頁建立來源資料庫，請將 DB2CODEPAGE 環境變數設為該字碼頁。有關設定 DB2CODEPAGE 之前取得字碼頁值的詳細資訊，請參閱 *DB2 Administration Guide*。「擷取程式」必須在與資料庫 (擷取程式自其中擷取資料) 相同的字碼頁中執行。DB2 從「擷取程式」執行所在的作用中環境取得「擷取程式」字碼頁。若未設定 DB2CODEPAGE，DB2 會從作業系統取得字碼頁值。若您建立資料庫時是使用預設字碼頁，則衍生自作業系統的值對「擷取程式」而言是正確值。

第 14 章 表格結構

在 339 頁，添加下列句子至值為 2 的 STATUS 直欄說明：

如果您使用內部 CCD 表格，且持續在引用追蹤表的狀態直欄中得到 "2" 值，請至「第 8 章：問題判定」參照「問題：沒有抄寫變更的情況下，引用程式形成迴圈；引用追蹤表顯示 STATUS=2」。

第 15 章 擷取程式和引用程式訊息

訊息 ASN0017E 應改為：

ASN0017E

「擷取程式」發現嚴重的內部錯誤，無法發出正確的錯誤訊息。常式名稱為 routine。回覆碼為 return_code。

新增下列資料至訊息 ASN1027S：

ASN1027S

指定太多大型物件 (LOB) 直欄。錯誤碼為 <error_code>。

說明：對定期抄寫設定成員指定了太多大型物件 (BLOB、CLOB 或 DBCLOB) 直欄。可容許的最大直欄數為 10。

使用者回應：自定期抄寫設定成員除去過多的大型物件直欄。

訊息 ASN1048E 應如下所示：

ASN1048E

「引用程式」循環的執行失敗。詳細資料，請參閱引用追蹤表：<text>

說明：「引用程式」循環失敗。在訊息中，<text> 定義 <target_server>、<target_owner>、<target_table>、<stmt_number> 及 <cntl_server>。

使用者回應：檢查審核追蹤表格的 APPERRM 欄位，以確定「引用程式」循環的失敗原因。

附錄 A. 從應用程式內啓動擷取程式和引用程式

在 399 頁，錯誤出現在範例常式的說明內，該範例常式啓動擷取程式和引用程式，但是，範例的字碼是正確的。範例的後半部與引用程式參數有關，雖然說明表示它和擷取程式參數有關。

您可以自以下目錄獲得引用程式和擷取程式的範例及各自的 `make` 檔：

```
For NT - sqllib\samples\repl  
For UNIX - sqllib/samples/repl
```

System Monitor Guide and Reference(無中文版)

db2ConvMonStream

In the Usage Notes, the structure for the snapshot variable datastream type SQLM_ELM_SUBSECTION should be sqlm_subsection.

Troubleshooting Guide(無中文版)

Starting DB2 on Windows 95, Windows 98, and Windows ME When the User Is Not Logged On

For a **db2start** command to be successful in a Windows 95, Windows 98, or Windows Millennium Edition (ME) environment, you must either:

- Log on using the Windows logon window or the Microsoft Networking logon window
- Issue the **db2logon** command (see note 1 in page 164 for information about the **db2logon** command).

In addition, the user ID that is specified either during the logon or for the **db2logon** command must meet DB2's requirements (see note 2 in page 165).

When the **db2start** command starts, it first checks to see if a user is logged on. If a user is logged on, the **db2start** command uses that user's ID. If a user is not logged on, the **db2start** command checks whether a **db2logon** command has been run, and, if so, the **db2start** command uses the user ID that was specified for the **db2logon** command. If the **db2start** command cannot find a valid user ID, the command terminates.

During the installation of DB2 Universal Database Version 7 on Windows 95, Windows 98, and Windows ME, the installation software, by default, adds a shortcut to the Startup folder that runs the **db2start** command when the system is booted (see note 第164頁的1 for more information). If the user of the system has neither logged on nor issued the **db2logon** command, the **db2start** command will terminate.

If you or your users do not normally log on to Windows or to a network, you can hide the requirement to issue the **db2logon** command before a **db2start** command by running commands from a batch file as follows:

1. Create a batch file that issues the **db2logon** command followed by the **db2start.exe** command. For example:

```
@echo off
db2logon db2local /p:password
db2start
cls
exit
```

2. Name the batch file **db2start.bat**, and store it in the **/bin** directory that is under the drive and path where you installed DB2. You store the batch file in this location to ensure that the operating system can find the path to the batch file.

The drive and path where DB2 is installed is stored in the DB2 registry variable **DB2PATH**. To find the drive and path where you installed DB2, issue the following command:

```
db2set -g db2path
```

Assume that the **db2set** command returns the value **c:\sql1lib**. In this situation, you would store the batch file as follows:

c:\sql1lib\bin\db2start.bat

3. To start DB2 when the system is booted, you should run the batch file from a shortcut in the Startup folder. You have two options:

- Modify the shortcut that is created by the DB2 installation program to run the batch file instead of **db2start.exe**. In the preceding example, the shortcut would now run the db2start.bat batch file. The shortcut that is created by DB2 installation program is called **DB2 - DB2.lnk**, and is located in c:\WINDOWS\Start Menu\Programs\Start\DB2 - DB2.lnk on most systems.
- Add your own shortcut to run the batch file, and delete the shortcut that is added by the DB2 installation program. Use the following command to delete the DB2 shortcut:

```
del "C:\WINDOWS\Start Menu\Programs\Startup\DB2 - DB2.lnk"
```

If you decide to use your own shortcut, you should set the *close on exit* attribute for the shortcut. If you do not set this attribute, the DOS command prompt is left in the task bar even after the **db2start** command has successfully completed. To prevent the DOS window from being opened during the **db2start** process, you can create this shortcut (and the DOS window it runs in) set to run minimized.

Note: As an alternative to starting DB2 during the boot of the system, DB2 can be started prior to the running of any application that uses DB2. See note 5 in page 165 for details.

If you use a batch file to issue the **db2logon** command before the **db2start** command is run, and your users occasionally log on, the **db2start** command will continue to work, the only difference being that DB2 will use the user ID of the logged on user. See note 1 for additional details.

Note:

1. The **db2logon** command simulates a user logon. The format of the db2logon command is:

```
db2logon userid /p:password
```

The user ID that is specified for the command must meet the DB2 naming requirements (see note 12 in page 165 for more information). If the command is issued without a user ID and password, a window opens to prompt the user for the user ID and password. If the only parameter provided is a user ID, the user is not prompted for a password; under certain conditions a password is required, as described below.

The user ID and password values that are set by the **db2logon** command are only used if the user did not log on using either the Windows logon window or the Microsoft Networking logon window. If the user has logged on, and a **db2logon** command has been issued, the user ID from the **db2logon** command is used for all DB2 actions, but the password specified on the **db2logon** command is ignored.

When the user has not logged on using the Windows logon window or the Microsoft Networking logon window, the user ID and password that are provided through the **db2logon** command are used as follows:

- The **db2start** command uses the user ID when it starts, and does not require a password.
- In the absence of a high-level qualifier for actions like creating a table, the user ID is used as the high-level qualifier. For example:
 - a. If you issue the following: `db2logon db2local`
 - b. Then issue the following: `create table tabl`
 The table is created with a high-level qualifier as `db2local.tab1`.

You should use a user ID that is equal to the schema name of your tables and other objects.

- When the system acts as client to a server, and the user issues a `CONNECT` statement without a user ID and password (for example, `CONNECT TO TEST`) and authentication is set to server, the user ID and password from the **db2logon** command are used to validate the user at the remote server. If the user connects with an explicit user ID and password (for example, `CONNECT TO TEST USER userID USING password`), the values that are specified for the `CONNECT` statement are used.
2. In Version 7, the user ID that is either used to log on or specified for the **db2logon** command must conform to the following DB2 requirements:
 - It cannot be any of the following: `USERS`, `ADMINS`, `GUESTS`, `PUBLIC`, `LOCAL`, or any SQL reserved word that is listed in the *SQL Reference*.
 - It cannot begin with: `SQL`, `SYS` or `IBM`
 - Characters can include:
 - A through Z (Windows 95, Windows 98, and Windows ME support case-sensitive user IDs)
 - 0 through 9
 - @, #, or \$
 3. You can prevent the creation of the **db2start** shortcut in the Startup folder during a customized interactive installation, or if you are performing a response file installation and specify the `DB2.AUTOSTART=NO` option. If you use these options, there is no `db2start` shortcut in the Startup folder, and you must add your own shortcut to run the `db2start.bat` file.
 4. On **Windows 98** and **Windows ME** an option is available that you can use to specify a user ID that is always logged on when Windows 98 or Windows ME is started. In this situation, the Windows logon window will not appear. If you use this option, a user is logged on and the **db2start** command will succeed if the user ID meets DB2 requirements (see note 2 for details). If you do not use this option, the user will always be presented with a logon window. If the user cancels out of this window without logging on, the **db2start** command will fail unless the **db2logon** command was previously issued, or invoked from the batch file, as described above.
 5. If you do not start DB2 during a system boot, DB2 can be started by an application. You can run the `db2start.bat` file as part of the initialization of applications that use DB2. Using this method, DB2 will only be started when the application that will

use it is started. When the user exits the application, a **db2stop** command can be issued to stop DB2. Your business applications can start DB2 in this way, if DB2 is not started during the system boot.

To use the DB2 Synchronizer application or call the synchronization APIs from your application, DB2 must be started if the scripts that are download for execution contain commands that operate either against a local instance or a local database. These commands can be in database scripts, instance scripts, or embedded in operating system (OS) scripts. If an OS script does not contain Command Line Processor commands or DB2 APIs that use an instance or a database, it can be run without DB2 being started. Because it may be difficult to tell in advance what commands will be run from your scripts during the synchronization process, DB2 should normally be started before synchronization begins.

If you are calling either the **db2sync** command or the synchronization APIs from your application, you would start DB2 during the initialization of your application. If your users will be using the DB2 Synchronizer shortcut in the DB2 for Windows folder to start synchronization, the DB2 Synchronization shortcut must be modified to run a db2sync.bat file. The batch file should contain the following commands to ensure that DB2 is running before synchronization begins:

```
@echo off
db2start.bat
db2sync.exe
db2stop.exe
cls
exit
```

In this example, it is assumed that the db2start.bat file invokes the **db2logon** and **db2start** commands as described above.

If you decide to start DB2 when the application starts, ensure that the installation of DB2 does not add a shortcut to the Startup folder to start DB2. See note 3 in page 165 for details.

Chapter 2. Troubleshooting the DB2 Universal Database Server

Under the 『Locking and Deadlocks』 section, under the 『Applications Slow or Appear to Hang』 subsection, change the description under 『Lock waits or deadlocks are not caused by next key locking』 to :

Next key locking guarantees Repeatable Read (RR) isolation level by automatically locking the next key for all INSERT and DELETE statements and the next higher key value above the result set for SELECT statements. For UPDATE statements that alter key parts of an index, the original index key is deleted and the new key value is inserted. Next key locking is done on both the key insertion and key deletion. It is required to guarantee ANSI and SQL92 standard RR, and is the DB2 default.

Examine snapshot information for the application. If the problem appears to be with next key locking, you can set the DB2_RR_TO_RS option on if none of your applications rely on Repeatable Read (RR) behavior and it is acceptable for scans to skip over uncommitted deletes.

| When DB2_RR_TO_RS is on, RR behavior cannot be guaranteed for scans on user tables
| because next key locking is not done during index key insertion and deletion. Catalog
| tables are not affected by this option.

| The other change in behavior is that with DB2_RR_TO_RS on, scans will skip over rows
| that have been deleted but not committed, even though the row may have qualified for
| the scan.

| For example, consider the scenario where transaction A deletes the row with column1=10
| and transaction B does a scan where column1>8 and column1<12.

| With DB2_RR_TO_RS off, transaction B will wait for transaction A to commit or rollback.
| If it rolls back, the row with column1=10 will be included in the result set of transaction
| B's query.

| With DB2_RR_TO_RS on, transaction B will not wait for transaction A to commit or
| rollback. It will immediately receive query results that do not include the deleted row.

| Do not use this option if you require ANSI and SQL92 standard RR or if you do not
| want scans to skip uncommitted deletes.

在 64 位元的平台上使用 DB2 Universal Database

第 5 章 架構

LOCKLIST

下列資訊應該加到「表 2」。

參數	之前的上限	現行的上限
LOCKLIST	60000	524288

shmsys:shminfo_shmmax

在 64 位元 Solaris 作業系統上的 DB2 使用者應視需要增加 `/etc/system` 中 `"shmsys:shminfo_shmmax"` 的值，以配置大的資料庫共用記憶體組。*DB2 for UNIX 快速入門* 書中建議參數可設為 "機器中實體 RAM 的 90%，單位為位元組"。這項建議對於 64 位元施行一樣有效。

然而，在 *DB2 for UNIX 快速入門* 書中的下列建議有問題：對於擁有超過 4 GB RAM 的 32 位元系統而言（在 Solaris 作業系統上總共最多可到 64 GB），如果使用者將 `shmmax` 值設定為一個大於 4 GB 的數字，並且是使用 32 位元的核心，而核心只會看到數字的後 32 位元，有時會造成 `shmmax` 的值非常的小。

第 6 章 限制

目前在 64 位元的作業系統上沒有 LDAP 支援。

32 位元及 64 位元的資料庫無法建立在相同的路徑上。例如，如果有一個 32 位元的資料庫位於 `<somepath>`，則：

```
db2 create db <somedb> on <somepath>
```

如果是從 64 位元的案例發出指令，會得到這樣的錯誤訊息 "SQL10004C 存取資料庫目錄時發生 I/O 錯誤。"

XML Extender Administration and Programming(無中文版)

Release Notes for the IBM DB2 XML Extender can be found on the DB2 XML Web site: <http://www-4.ibm.com/software/data/db2/extenders/xmlext/library.html>

MQSeries

本節說明如何使用 DB2 及 MQSeries 來建構結合傳訊與資料庫存取的應用程式。本節的焦點在一系列的功能，與「使用者定義的功能 (UDF)」類似，它可能已在 DB2 Universal Database 版本 7.2 中選用性地啓用。使用這些基本的功能，可能可以支援大範圍的應用程式，從簡式事件通知到資料倉儲。

有關資料倉儲應用程式的詳細資訊，請參閱第223頁的『MQ Series 與「資料倉儲中心」的整合』。

安裝與架構 DB2 MQSeries Function

本節說明如何架構 DB2 環境以使用 DB2 MQSeries Function。只要您成功完成下列程序，就可以從 SQL 使用 DB2 MQSeries Function。關於這些功能的說明，請至「版本注意事項」的 SQL Reference 一節。關於附加資訊，包括最新的文件、提示及秘訣，請至 <http://www.ibm.com/software/data/integration/MQSeries>。

架構與啓用 DB2 MQSeries Function 的基本程序為：

1. 安裝 MQSeries。
2. 安裝 MQSeries AMI。
3. 啓用與架構 DB2 MQSeries Function

此外，若要使用 DB2 MQSeries Function 提供的發佈/訂閱功能，您也必須安裝 MQSeries Integrator 或 MQSeries Publish/Subscribe Function。關於 MQSeries Integrator 的資訊，請至 <http://www.ibm.com/software/ts/mqseries/integrator>。關於 MQSeries Publish/Subscribe 功能的資訊，請至種類 3 下的 <http://www.ibm.com/software/ts/mqseries/txppacs>。

安裝 MQSeries

第一個步驟請確定 MQSeries 版本 5.2 已安裝於您的 DB2 伺服器上。若您已安裝此版本的 MQSeries，請跳至下一步驟「安裝 MQSeries AMI」。DB2 版本 7.2 包括 DB2 使用的 MQSeries 伺服器之副本。安裝 MQSeries 或升級現存的 MQSeries 安裝之特定平台指令，可在特定的平台「快速入門」書籍中找到，其位在 <http://www.ibm.com/software/ts/mqseries/library/manuals>。請確定當您進行安裝程序時，要設定預設佇列管理程式。

安裝 MQSeries AMI

下個步驟是安裝 MQSeries 應用程式傳訊介面 (AMI)。這是 MQSeries 程式設計介面的副檔名，它提供管理及程式設計一個清楚的隔離。DB2 MQSeries Function 需要此介面的安裝。若 MQSeries AMI 已安裝在您的 DB2 伺服器上，請跳至下一步驟「啓用 DB2 MQSeries Function」。若尚未安裝 MQSeries AMI，則您可以從 DB2 7.2 提供的安裝套裝軟體來安裝，或從 MQSeries 支援套裝軟體網站 <http://www.ibm.com/software/ts/mqseries/txppacs>，下載 AMI 副本。您可以在「種類 3 - 產品副檔名」下找到 AMI。爲了方便起見，我們提供您 DB2 的 MQSeries AMI 之副本。此檔案位在 `sqllib/cfg` 目錄中。此檔案的名稱是作業系統相依項：

AIX 版本 4.3 及更新版本	ma0f_ax.tar.Z
HP-UX	ma0f_hp.tar.Z
Solaris 作業環境	ma0f_sol7.tar.Z
Windows 32 位元	ma0f_nt.zip

遵循 AMI Readme 檔中簡述的正常 AMI 安裝程序，該檔在壓縮安裝映像檔中。

啓用 DB2 MQSeries Function

在此步驟期間，您將架構並啓用 DB2 MQSeries Function 的資料庫。

enable_MQFunctions 公用程式是一個彈性的指令，會先檢查已設定適當的 MQSeries 環境，然後安裝並建立 DB2 MQSeries 功能的預設架構，以這些功能啓用指定的資料庫，並確認架構運作正常。

1. 針對 Windows NT 或 Windows 2000，請至步驟 5。
2. 在 UNIX 設定群組：若您在 UNIX 上啓用這些功能，您必須先將 DB2 案例擁有者 (通常是 de2inst1) 及與隔離 UDF 相關的使用者 ID (通常是 db2fenc1)，新增到 MQSeries 群組 mqm 中。DB2 功能需要此設定以存取 MQSeries。
3. 在 UNIX 上設定 DB2 環境變數：將 AMT_DATA_PATH 環境變數新增到 DB2 瞭解的列示中。您可以編輯檔案 \$INSTHOME/sqllib/profile.env，將 AMT_DATA_PATH 新增至 DB2ENVLIST。也可使用 **db2set** 指令。
4. 在 UNIX 上，重新啓動資料庫案例：若要讓環境變數變生效，必須重新啓動資料庫案例。
5. 在 UNIX 上將目錄變更為 \$INSTHOME/sqllib/cfg，在 Windows 上將目錄變更為 %DB2PATH%/cfg。
6. 執行指令 **enable_MQFunctions** 以架構並啓用 DB2 MQSeries 功能的資料庫。請參閱第182頁的『enable_MQFunctions』，以取得此指令的完整說明。下列提供一些一般範例。在成功完成之後，將啓用指定的資料庫並測試架構。
7. 若要使用「命令行處理器」測試這些功能，請在您連接到已啓用的資料庫時，發出下列指令：

```
values DB2MQ.MQSEND('a test')
values DB2MQ.MQRECEIVE()
```

第一個陳述式將傳送訊息 "a test" 至 DB2MQ_DEFAULT_Q 佇列，而第二個則將其接收回來。

註：執行 **enable_MQFunctions** 的結果，將建立預設 MQSeries 環境。將建立 MQSeries 佇列管理程式 DB2MQ_DEFAULT_MQM 及預設佇列 DB2MQ_DEFAULT_Q。若檔案 amt.xml、amthost.xml 及 amt.dtd 已不存在於 AMT_DATA_PATH 指向的目錄時，將建立這些檔案。若 amthost.xml 檔案不存在，且不包含 connectionDB2MQ 的定義，則會將行新增到有適當資訊的檔案中。原來的檔案之副本將被儲存為 DB2MQSAVE.amthost.xml。

MQSeries 傳訊樣式

DB2 MQSeries 功能支援三種傳訊模型：資料圖、發佈/訂閱 (p/s) 及要求/回覆 (r/r)。

以資料圖來傳送的訊息會傳送到單一目的地，且不預期回覆。在 p/s 模型中，一或多個公布者將訊息傳送至公佈服務程式，分送訊息至一或多個用戶。要求/回覆與資料圖相似，但傳送者預期接收回應。

訊息結構

MQSeries 本身不會要求或支援任何其他傳送的訊息之特殊結構。

其它產品如 MQSeries Integrator (MQSI)，就會提供以 C、COBOL 或 XML 字串形成的訊息之支援。在 MQSI 中的結構訊息是由訊息儲存庫定義。XML 訊息通常有一個自行說明的訊息結構，且可能也透過儲存庫來管理。訊息也可能是沒有結構的，需要使用者程式碼來剖析或建構訊息內容。這類訊息往往是不完全的結構，也就是說，它們使用位元組位置或固定的定界符號來分隔訊息中的欄位。「MQSeries 輔助精靈」提供這種不完全結構的訊息之支援。透過某些新功能，將 XML 訊息支援提供給 DB2 XML 擴充元。

MQSeries 功能概觀

MQSeries 功能集由 DB2 UDB 版本 7.2 提供，可讓 SQL 陳述式包括傳訊作業。這表示此支援可使用於以任何支援語言撰寫的應用程式，例如，使用任何資料庫介面的 C、Java、SQL。下列顯示的所有範例均在 SQL 中。可以用所有標準方式，從其他程式設計語言使用此 SQL。上述的所有 MQSeries 傳訊樣式都已支援。有關 MQSeries 功能的詳細資訊，請參閱「版本注意事項」的 SQL Reference 一節。

在基本架構中，MQSeries 伺服器是位在與 DB2 一起的資料庫伺服器機器中。MQSeries 功能安裝在 DB2 上，並提供對 MQSeries 伺服器的存取權限。DB2 從屬站可位在可存取 DB2 Server 的任何機器上。多個從屬站可同時透過資料庫存取 MQSeries 功能。透過提供的功能，DB2 從屬站可在 SQL 陳述式內執行傳訊作業。這些傳訊作業可讓 DB2 應用程式在它們之間通信，或與其他 MQSeries 應用程式通信。

enable_MQFunctions 指令是用來啟用 MQSeries 功能的 DB2 資料庫。它將自動建立簡式預設架構，從屬站應用程式會使用該架構，不需要管理動作。若要取得說明，請參閱 `enable_MQFunctions` 及 `disable_MQFunctions`。預設架構提供應用程式設計師一個快速啟動的方法，及一個更簡易的開發介面。若有需要，可增加架構其他的功能。

範例 1：若要使用預設架構來傳送簡式訊息，SQL 陳述式如下：

```
VALUES DB2MQ.MQSEND('simple message')
```

這會將訊息 `simple message` 傳送至 MQSeries 佇列管理程式及預設架構指定的佇列。

MQSeries 的「應用程式傳訊介面 (AMI)」，在傳訊活動及指定那些活動的傳送方式之定義之間，提供清楚的隔離。這些定義保留在外部儲存庫檔案中，並使用 AMI 管理工具來管理。這可讓 AMI 應用程式容易發展並維護。DB2 提供的 MQSeries 功能是以 AMI MQSeries 介面為依據。AMI 支援外部架構檔的使用，該檔稱為「AMI 儲存庫」，以儲存架構資訊。預設架構包括 MQSeries AMI 儲存庫架構，以使用 DB2。

MQSeries AMI 的兩個主要概念，服務程式點及原則，被轉遞至 DB2 MQSeries 功能中。服務程式點是邏輯終點，即訊息可能傳送或接收的點。在 AMI 儲存庫中，每一個服務程式點是以 MQSeries 佇列名稱及佇列管理程式來定義。原則定義服務程式選項的品質，其應使用於給定的傳訊作業。主要的服務程式品質包括訊息優先順序及持續性。提供

預設服務程式點及原則定義，且可能由發展者使用，以進一步簡化應用程式。範例 1 可以下列方式重寫，以明確地指定預設服務程式點及原則名稱：

範例 2：

```
VALUES DB2MQ.MQSEND('DB2.DEFAULT.SERVICE', 'DB2.DEFAULT.POLICY', 'simple message')
```

佇列可能由伺服器的一或多個應用程式服務，而佇列及應用程式即常駐於該伺服器中。在許多架構中，多重佇列將被定義為支援不同的應用程式及目的。因為這個理由，在進行 MQSeries 要求時，經常定義不同的服務程式點是很重要的。在下列範例中有說明：

範例 3：

```
VALUES DB2MQ.MQSEND('ODS_Input', 'simple message')
```

註：在此範例中，未指定原則，但將使用預設原則。

限制

MQSeries 提供訊息作業及資料庫作業的能力，以在單一工作單元中結合，作為被保護交易。Unix 及 Windows 上的 MQSeries 功能原本不支援此功能。

當使用傳送或接收功能時，一個訊息的最大長度是 4000 個字元。這也是使用 MQPublish 來公布訊息的最大訊息大小。

錯誤碼

您可在「MQSeries 應用程式傳訊介面手冊」的附錄 B 中，找到 MQSeries 功能傳回的回覆碼。

用法實務手冊

MQSeries Function 可使用於大範圍的實務手冊。本節將討論一些最普遍的實務，包括「基本傳訊」、「應用程式連接」及「資料公布」。

基本傳訊

當所有資料庫應用程式連接到相同的 DB2 伺服器時，會發生 MQSeries DB2 功能的訊息之最基本形式。從屬站可能是資料庫伺服器的本端或分散在網路環境中。

在簡式實務範例中，從屬站 A 呼叫 MQSEND 功能，傳送使用者定義至預設服務程式位置。MQSeries 功能隨後會在資料庫伺服器的 DB2 內執行。稍後，從屬站 B 呼叫 MQRECEIVE 功能，移除預設服務程式定義的佇列中之最早記載記錄的訊息，並將其傳回從屬站。同樣的，執行此作業的 MQSeries 功能由 DB2 執行。

資料庫從屬站可以用許多方式來使用簡式傳訊。傳訊的某些一般用法如下：

- 資料收集 -- 以來自一或多個可能不同的資訊來源之形式接收資訊。資訊來源可能是商業應用程式如 SAP，或組織內部發展的應用程式。這類資料可從佇列接收，並儲存在資料庫表格中，以供進一步的處理或分析。
- 工作負荷分送 -- 工作要求公布於佇列，由相同應用程式的多個案例共用。當案例備妥執行一些工作時，它會從佇列頂端接收到訊息，包含要執行的工作要求。使用此技術，多個案例可以共用儲存要求之單一佇列所代表的工作負荷。

- 應用程式信號 -- 在幾個程序合作的情況中，訊息通常用來協調其成果。這些訊息可能包含工作要執行的指令或要求。通常，這種信號是單向的；也就是說，起始訊息的一方並不預期回應。請參閱第179頁的『Request/Reply 通信』，以取得詳細資訊。
- 應用程式通知 -- 通知與信號類似，資料從初始程式傳送出，且不預期回應。但是，通知通常包含關於已發生之業務事件的資料。第180頁的『發佈/訂閱』是通知的更進階之形式。

下列實務範例擴充了上述說明的簡式實務範例，以納入遠端傳訊。也就是說，訊息在機器 A 與機器 B 之間傳送。步驟順序如下：

1. DB2 從屬站執行 MQSEND 呼叫，指定已定義的目標服務程式，以代表機器 B 的遠端佇列。
2. MQSeries DB2 功能執行實際的 MQSeries 作業以傳送訊息。在機器 A 上的 MQSeries 伺服器接受訊息及保證，將其傳送至服務程式點定義及機器 A 的現行 MQSeries 架構所定義的目的地。伺服器決定這是機器 B 上的佇列。然後它會嘗試將訊息傳送至機器 B 的 MQSeries 伺服器，並在需要時透過地重試。
3. 機器 B 上的 MQSeries 伺服器接收來自機器 A 之伺服器的訊息，並將其放置在機器 B 的目的地佇列。
4. 機器 B 上的 MQSeries 從屬站要求佇列的最早記載記錄之訊息。

傳送訊息

使用 MQSEND，DB2 使用者或發展者可選擇要傳送的資料，要傳送的位置，及要傳送的時間。在業界中，這通常叫做 "Send and Forget"，表示傳送者只要傳送訊息，而依據 MQSeries 的遞送通信協定保證，確保訊息可到達目的地。下列範例會對此作說明。

範例 4：若要以原則 highPriority 來將使用者定義字串傳送至服務程式點 myplace：

```
VALUES DB2MQ.MQSEND('myplace','highPriority','test')
```

在這裡，原則 highPriority 參照 AMI 儲存庫所定義的原則，將 MQSeries 優先順序設定為最高層次，也可能調整其他服務程式的品質，例如持續性。

訊息內容可能由 SQL 的任何標準尺寸組合及使用者指定的資料組合而成。這包括巢狀功能、運算子及強制轉型。例如，提供一表格 EMPLOYEE，有 VARCHAR 直欄 LASTNAME、FIRSTNAME 及 DEPARTMENT，針對每一個 DEPARTMENT 5LGA 中的員工傳送包含此資訊之訊息，您可執行下列：

範例 5：

```
SELECT DB2MQ.MQSEND(LASTNAME || ' ' || FIRSTNAME || ' ' || DEPARTMENT)
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

若此表格也有一個完整的 AGE 直欄，它可能也會被併入，如下所示：

範例 6：

```
SELECT DB2MQ.MQSEND(LASTNAME || ' ' || FIRSTNAME || ' ' || DEPARTMENT || ' ' || char(AGE))
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

最後，下列範例顯示如何使用任何有效的 SQL 表示式來衍生訊息內容。提供第二個表格 DEPT，其包含 VARCHAR 直欄 DEPT_NO 及 DEPT_NAME，可傳送包含員工 LASTNAME 及 DEPT_NAME 的訊息：

範例 7：

```
SELECT DB2MQ.MQSEND(e.LASTNAME || ' ' || d.DEPTNAME) FROM EMPLOYEE e, DEPT d
WHERE e.DEPARTMENT = d.DEPTNAME
```

擷取訊息

MQSeries DB2 功能可接收或讀取訊息。讀取與接收之間的不同處，在於讀取傳回佇列之最早記載記錄的訊息，不需從佇列移除，而接收作業會使訊息從佇列移除。使用接收作業來擷取的訊息，僅可以擷取一次，而使用讀取作業來擷取的訊息，可多次擷取相同訊息。下列範例會對此作說明：

範例 8：

```
VALUES DB2MQ.MQREAD()
```

此範例使用服務程式原則的預設品質，傳回 VARCHAR 字串，該字串包含預設服務程式定義的佇列之最早記載記錄的訊息。重要事項：請注意若沒有可供讀取的訊息，將傳回一個 NULL 值。此作業不會變更佇列。

範例 9：

```
VALUES DB2MQ.MQRECEIVE('Employee_Changes')
```

上述範例顯示如何使用預設原則，將訊息從 Employee_Changes 服務程式定義的佇列之最早記載記錄移除。

DB2 的一個非常有力的功能，是從使用者定義 (或 DB2 提供的) 功能，產生表格的能力。您可以開發此表格功能，讓佇列內容可作為 DB2 表格。下列範例說明這個最簡易的形式：

範例 10：

```
SELECT t.* FROM table ( DB2MQ.MQREADALL() ) t
```

此查詢傳回一表格，由所有預設服務程式定義之佇列的所有訊息及有關這些訊息的描述資料所組成。當傳回的表格結構之完整定義是定義於附錄中，則第一直欄會反映訊息內容，其餘直欄則包含描述資料。若只要傳回訊息，可重新撰寫範例：

範例 11：

```
SELECT t.MSG FROM table (DB2MQ.MQREADALL()) t
```

表格功能傳回的表格與資料庫直接擷取的表格相同。這表示您可以廣泛的方法使用此表格。例如，您可以結合一表格與另一表格的內容，或計算佇列中的訊息數量：

範例 12：

```
SELECT t.MSG, e.LASTNAME
FROM table (DB2MQ.MQREADALL() ) t, EMPLOYEE e
WHERE t.MSG = e.LASTNAME
```

範例 13：

```
SELECT COUNT(*) FROM table (DB2MQ.MQREADALL()) t
```

您可以透過表格功能來建立概略表，以隱藏表格來源是佇列的事實。例如，下列範例透過服務程式 NEW_EMPLOYEES 所參照的佇列，來建立名為 NEW_EMP 的概略表：

範例 14：

```
CREATE VIEW NEW_EMP (msg) AS
  SELECT t.msg FROM table (DB2MQ.MQREADALL()) t
```

在此案例中，概略表僅以包含整個訊息的單一直欄來定義。若訊息是簡式的架構，例如包含固定長度的兩個欄位，則它會直接向前以使用 DB2 內建功能，將訊息剖析至兩直欄中。例如，若您知道傳送至特殊佇列的訊息都包含一個 18 位元的姓氏，其後則是一個 18 位元的名字，則您可以定義一個概略表，包含每一個個別直欄的欄位，例如：

範例 15：

```
CREATE VIEW NEW_EMP2 AS
  SELECT left(t.msg,18) AS LNAME, right(t.msg,18) AS FNAME
  FROM table(DB2MQ.MQREADALL()) t
```

「DB2 儲存程序建置器」的新特性「MQSeries 輔助精靈」，可用來建立新的 DB2 表格功能及概略表，其將有定界符號的訊息結構對映至直欄。

最後，通常要將一或多個訊息內容儲存在資料庫中。這可使用 SQL 的完整功能來完成，以操作並儲存訊息內容。最簡易的範例可能是：

範例 16：

```
INSERT INTO MESSAGES
  SELECT t.msg FROM table (DB2MQ.MQRECEIVEALL()) t
```

提供包含 VARCHAR(2000) 單一直欄的表格 MESSAGES，上述的陳述式會從預設服務程式佇列，將訊息插入表格中。可修飾此技術，以涵蓋情況的廣泛種類。

應用程式對應用程式連接

應用程式整合是許多解決方案的一般元素。不論將購買的應用程式整合到現存的基礎架構中，或只將新開發的應用程式整合到現存環境中，我們通常會面對集中子系統之不同集合的作業，以構成完整的工作。MQSeries 通常被視為整合應用程式的基本工具。可在大部份硬體、軟體及語言環境中存取，MQSeries 提供交互連接應用程式的不同集合之方法。

此節將討論部份應用程式整合實務及如何以 DB2 來使用它們。因為主題相當廣泛，「應用程式整合」的廣泛討論超越了此工作的範圍。因此，重點僅在兩個簡單的主題：Request/Reply 通信，及 MQSeries Integrator 及 Publish/Subscribe。

Request/Reply 通信

「Request/Reply (R/R)」通信方法是最普遍的技術，一個應用程式可向另一個應用程式要求服務。執行此動作的其中一種方法，是讓要求程式將訊息傳送至要求執行某些工作的服務提供程式。一旦工作已完成，供應程式可決定將結果 (或完成確認) 傳回要求器。但使用上述的基本傳訊技術，就不能連接傳送者要求與服務提供程式回應。除非要求程式在繼續執行之前，先等待回覆，否則某些機制必須用來結合每一個回覆及其要求。MQSeries 不強制發展程式建立這類機制，而是提供一個相關 ID，使交換處的訊息有相關性。

當有許多方法可供此機制使用時，最簡易的方法是讓要求器用已知的相關 ID 來標示訊息，例如下列所示：

範例 17：

```
DB2MQ.MQSEND ('myRequester','myPolicy','SendStatus:cust1','Req1')
```

此陳述式將最終參數 Req1 新增至上述的 MQSEND 陳述式中，以指出要求的相關 ID。

若要接收此特定要求的回覆，可使用相對應的 MQRECEIVE 陳述式，選擇性地接收指示的服務程式所定義的第一個訊息，其與下列相關 ID 相符：

範例 18：

```
DB2MQ.MQRECEIVE ('myReceiver','myPolicy','Req1')
```

若服務要求的應用程式忙線中，且要求器在回覆傳送之前，發出上述的 MQRECEIVE，則將找不到與相關 ID 符合的訊息。

若要接收服務程式要求及相關 ID，使用下列的陳述式：

範例 19：

```
SELECT msg, correlid FROM table (DB2MQ.MQRECEIVEALL('aServiceProvider','myPolicy',1)) t
```

這會傳回服務程式 aServiceProvider 的第一個要求之訊息及相關 ID。

一旦執行了服務程式，它會將回覆訊息傳送給 aReauester 說明的佇列。在這期間，服務要求程式仍會執行其他作業。事實上，並不保證起始服務程式要求會在設定時間內回應。如同這類的應用程式層次逾時，必須由發展程式來管理；要求程式必須輪詢偵測回覆的情況。

這種時間獨立非同步處理的優點，是要求程式及服務供應程式是完全獨立地執行。這可用於適合的環境，應用程式僅會斷續地連接，也可用於更批次導向的環境，多重要求或回覆會在處理程序前先聚集起來。這種集合體通常用於資料倉儲環境，以定期地更新資料倉儲或作業資料儲存。

發佈/訂閱

簡式資料出版品： 應用程式整合中的另一個一般實務範例，是一應用程式通知其他應用程式有關興趣的事件。您可容易地將訊息傳送到佇列，該佇列由另一個應用程式監督。訊息內容可以是使用者定義字串，或可以從資料庫直欄來撰寫。簡式訊息往往是需要使用 MQSEND 功能來傳送的訊息。當這類訊息需要同時傳送到多個接收者時，可以使用 MQSeries AMI 的「分送清單」機能。

分送清單是使用 AMI 管理工具來定義。分送清單包含個別服務程式的清單。傳送到分送清單的訊息，會轉遞到此清單中所定義的每一個服務程式。當已知一些服務程式一直對每個訊息有興趣時，這是特別有用的功能。下列範例顯示傳送訊息至分送清單 interestedParties：

範例 20：

```
DB2MQ.MQSEND('interestedParties','information of general interest');
```

當特殊服務程式應接收的訊息需要更多控制時，則需要 Publish/Subscribe 能力。Publish/Subscribe 系統通常提供一個可調整的、安全的環境，許多訂戶可在其中登記，以

從多個公布者接收訊息。若要支援此能力，可使用 MQPublish 介面，與 MQSeries Integrator 或 MQ Series Publish/Subscribe 機能連接。

MQPublish 可讓使用者選用性地指定與訊息相關的主題。主題可讓用戶更清楚地指定要接受的訊息。步驟順序如下：

1. MQSeries 管理者架構 MQSeries Integrator 發佈/訂閱功能。
2. 感興趣的應用程式在 MQSI 架構所定義的訂閱點訂閱，選用性地指定其有興趣的主題。每一個用戶選取相關的主題，並可以利用 MQSeries Integrator V2 之以內容為基礎的訂閱技術。重要事項：請注意，服務程式名稱代表的佇列有定義用戶。
3. DB2 應用程式公布服務程式點 Weather 的訊息。訊息指出天氣是 Sleet，且其主題為 Austin，因此通知有興趣的用戶，告知 Austin 的天氣是 Sleet。
4. 實際公布訊息的機制是由 DB2 提供的 MQSeries 功能來處理。使用服務程式名稱 Weather，將訊息傳送至 MQSeries Integrator。
5. MQSI 從 Weather 服務程式接受訊息，執行 MQSI 架構定義的任何處理程序，並決定其滿意的訂購。然後 MQSI 會將訊息轉遞至其符合的準則之用戶佇列。
6. 已訂閱 Weather 服務程式，並在 Austin 登記喜好的應用程式，將在接收服務程式中接收到訊息 Sleet。

若要使用所有預設值及 NULL 主題來公布此資料，您可使用下列陳述式：

範例 21：

```
SELECT DB2MQ.MQPUBLISH(LASTNAME || ' ' || FIRSTNAME || ' ' || DEPARTMENT || ' ' || character(AGE))
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

完整的指定所有參數並簡化訊息，使其僅包含 LASTNAME，陳述式可能如下所示：

範例 22：

```
SELECT DB2MQ.MQPUBLISH('HR_INFO_PUB', 'SPECIAL_POLICY', LASTNAME,
'ALL_EMP:5LGA', 'MANAGER')
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

此陳述式使用 SPECIAL_POLICY 服務程式，將訊息公布至 HR_INFO_PUB 公佈服務程式中。訊息指出傳送者是 MANAGER 主題。主題字串說明可指定使用「：」連接的多重主題。在此範例中，使用這兩個主題可讓用戶登記 ALL_EMP 或只登記 5LGA，以接收這些訊息。

若要接收公布的訊息，您必須先將您的喜好登記在訊息中，訊息包含給定的主題並指出訊息應該傳送的用戶服務程式之名稱。重要事項：請注意 AMI 用戶服務程式有定義分配管理系統服務程式及接收者服務程式。分配管理系統服務程式是用戶與發布/訂閱分配管理系統通信的方式，而接收者服務程式是符合訂閱要求將傳送的地方。下列陳述式在主題 ALL_EMP 中登記喜好。

範例 23：

```
DB2MQ.MQSUBSCRIBE('aSubscriber', 'ALL_EMP')
```

一旦訂閱了應用程式，以主題 ALL_EMP 公布的訊息，將轉遞到用戶服務程式所定義的接收者服務程式中。一個應用程式可以有多重的並行訂閱。若要取得符合您的訂閱的

訊息，您可以使用任何標準的訊息擷取功能。例如，若用戶服務程式 aSubscriber 將接收者服務程式定義為 aSubscriberReceiver，而非下列陳述式，則將會非破壞性地讀取第一個訊息：

範例 24：

```
DB2MQ.MQREAD('aSubscriberReceiver')
```

若要決定以下要公布的訊息與主題，您可以使用其中一種表格功能。下列陳述式會接收 aSubscriberReceiver 的前五個訊息，並顯示訊息與主題：

範例 25：

```
SELECT t.msg, t.topic FROM table (DB2MQ.MQRECEIVEALL('aSubscriberReceiver',5)) t
```

若要以主題 ALL_EMP 來讀取所有訊息，您可以影響 SQL 的能力，請發出：

範例 26：

```
SELECT t.msg FROM table (DB2MQ.MQREADALL('aSubscriberReceiver')) t
WHERE t.topic = 'ALL_EMP'
```

註：重要的是您必須瞭解若 MQRECEIVEALL 使用限制，而非使用整個佇列，則並非只有那些訊息會公布主題 ALL_EMP。這是因為執行表格功能後，才引用限制。

當您不再有興趣訂閱特殊主題時，您必須使用陳述式，明確地取消訂閱，例如：

範例 27：

```
DB2MQ.MQUNSUBSCRIBE('aSubscriber', 'ALL_EMP')
```

一旦發出此陳述式，發佈/訂閱分配管理系統將不再傳送符合訂閱的訊息。

自動化公佈： 資料庫傳訊的另一個重要技術是自動化公佈。使用 DB2 中的觸發函式機能，您可以自動公布訊息，作為觸發函式呼叫的一部份。當其他的自動資料公布技術存在時，觸發程式基礎的方法可讓管理者或開發者自由地建構訊息內容，並彈性定義觸發動作。請注意，任何觸發函式的使用，都必須給付執行頻率與成本的費用。下列範例說明觸發函式如何使用 MQSeries DB2 功能。

下列範例顯示每次雇用新員工時，公布訊息是很容易的。任何訂閱 HR_INFO_PUB 服務程式並在 NEW_EMP 中登記喜好的使用者或應用程式，將會接收到一訊息，包括每一個新員工的日期、名稱及部門。

範例 28：

```
CREATE TRIGGER new_employee AFTER INSERT ON employee REFERENCING NEW AS n
FOR EACH ROW MODE DB2SQL
VALUES DB2MQ.MQPUBLISH('HR_INFO_PUB&', 'NEW_EMP',
current date || ' ' || LASTNAME || ' ' || DEPARTMENT)
```

enable_MQFunctions

enable_MQFunctions

啓用指定的資料庫之 DB2 MQSeries 功能，並驗證 DB2 MQSeries 功能可以適當地執行。若未安裝與架構 MQSeries 和 MQSeries AMI，指令將會失敗。

授權

下列其中一項：

- *sysadm*
- *dbadm*
- 若功能的隱含或明確的綱目名稱不存在，則 `IMPLICIT_SCHEMA` 在資料庫上。
- 若綱目名稱 `DB2MQ` 存在，則 `CREATEIN` 專用權在綱目上。

指令語法

```
▶▶enable_MQFunctions--n--database--u--userid--p--password--force--noValidate▶▶
```

指令參數

-n database

指定要啓用的資料庫名稱。

-u userid

指定使用者 ID 以連接到資料庫。

-p password

指定使用者 ID 的通行碼。

-force 指定重新安裝期間發現的警告應被忽略。

-noValidate

指定 DB2 MQSeries 功能的驗證將不執行。

範例

在下列範例中，已建立 DB2MQ 功能。使用者連線到資料庫 `SAMPLE`。已使用預設綱目 `DB2MQ`。

```
enable_MQFunctions -n sample -u user1 -p password1
```

用法注意事項

DB2 MQ 功能在此指令自動建立的綱目 `DB2MQ` 下執行。

執行指令之前：

- 確定已安裝 MQ 與 AMI，且 MQSeries 版本是 5.2 或更新版本。
- 確定已定義環境變數 `$AMT_DATA_PATH`。
- 將目錄變更為 `DB2PATH` 的 `cfg` 次目錄。

在 UNIX 上：

- 使用 `db2set`，將 `AMT_DATA_PATH` 新增至 `DB2ENVLIST`。

- 確定與 UDF 執行相關的使用者帳戶是 mqm 群組的成員。
- 確定要呼叫此指令的使用者是 mqm 群組的成員。

註: MQSeries 5.2 不支援 ATX 4.2。

disable_MQFunctions

disable_MQFunctions

停用指定的資料庫之 DB2 MQSeries 功能。

授權

下列其中一項：

- *sysadm*
- *dbadm*
- 若功能的隱含或明確的綱目名稱不存在，則 `IMPLICIT_SCHEMA` 在資料庫上。
- 若綱目名稱 `DB2MQ` 存在，則 `CREATEIN` 專用權在綱目上。

指令語法

```
▶▶ disable_MQFunctions -n database -u userid -p password ▶▶
```

指令參數

-n database

指定資料庫名稱。

-u userid

指定用來連接資料庫的使用者 ID。

-p password

指定使用者 ID 的通行碼。

範例

在下列範例中，已停用資料庫 `SAMPLE` 的 `DB2MQ` 功能。

```
disable_MQFunctions -n sample -u user1 -p password1
```


第4篇 管理工具

控制中心	189
管理 DB2 Server for VSE & VM 伺服器能力	189
Java 1.2 支援控制中心	189
在 Windows 作業系統下使用線上說明的「無效捷徑」錯誤	189
OS/2 系統下的 Java 控制中心	190
在 Windows 作業系統之異動日誌檢視已完成工作的「檔案拒絕存取」錯誤	190
多位置更新測試連接	190
DB2 for OS/390 的控制中心	190
OS/390 控制中心必要的修正程式	191
「建立空間層」對話框的變更	192
DB2 控制中心的疑難排解	192
UNIX 系統控制中心的疑難排解	192
OS/2 可能的 Infopop 問題	192
jdk11_path 架構參數的說明	192
使用「Script 中心」或「異動日誌」時發生的 Solaris 系統錯誤 (SQL10012N)	192
DPREPL.DFT 檔案的說明	193
啓動多個控制中心 applet	193
控制中心以 Applet 執行時的線上說明	193
在 Applet 模式 (Windows 95) 下執行控制中心	193
使用大的查詢結果	193
資訊中心	195
在 Windows 作業系統下的「無效捷徑」錯誤	195
當 Netscape 已開啓時，在 Netscape Navigator 中開啓外部 Web 鏈結(UNIX 系統)	195
啓動「資訊中心」的問題	195
精靈	197
在建立資料庫精靈設定擴充大小	197
MQSeries 輔助精靈	197
OLE DB 輔助精靈	197

控制中心

管理 DB2 Server for VSE & VM 伺服器能力

DB2 Universal Database 版本 7 「控制中心」增強對 DB2 伺服器 for VSE 和 VM 資料庫的支援。「控制中心」檢視所有 VSE 和 VM 資料庫物件的 DB2 伺服器。同時也支援建立索引、重組索引、更新統計值和重新連結指令。重組索引和重新連結需要儲存程序，而該儲存程序執行於 VSE 和 VM 主電腦上的 DB2 伺服器。此儲存程序由控制中心所提供，適用於 VSE 和 VM 下 DB2 伺服器的 VSE 和 VM 特性。

整合控制中心容許使用者管理 DB2，無論 DB2 伺服器執行於那一個平台。VSE 和 VM 物件的 DB2 伺服器顯示在控制中心主視窗，除了 DB2 Universal Database 物件。選取物件時，會呼叫管理這些物件的相對應動作和公用程式。範例，使用者可以列示特別資料庫的索引，選取其中的一個索引並將之重組。使用者也可以列示資料庫的表格，並執行更新統計值，或定義表格為抄寫來源。

有關架構控制中心，並在 VSE 和 VM 物件的 DB2 伺服器執行管理作業的資訊，請參照 *DB2 Connect User's Guide* 或安裝與架構補充。

Java 1.2 支援控制中心

使用 Java 1.2 中的雙向支援，控制中心可以支援雙向語言，如阿拉伯語和希伯來語。此支援只可用於 Windows NT 平台。

欲辨識和使用 Java 1.2，控制中心要進行安裝：

1. JDK 1.2.2 在 DB2 UDB CD 的 DB2\bid\NT 目錄下。
ibm-inst-n122p-win32-x86.exe 是安裝程式，ibm-jdk-n122p-win32-x86.exe 是 JDK 分送。將兩個檔案複製到硬碟上的暫時目錄中，然後從該目錄執行安裝程式。
2. 將它安裝在 <DB2PATH>\java\Java12 下，<DB2PATH> 是 DB2 的安裝路徑。
3. JDK/JRE 安裝提示時，不要選取 JDK/JRE 為系統 VM。

Java 1.2 安裝成功後，以正常方式啟動控制中心時會使用 Java 1.2。

欲停止使用 Java 1.2，請在 <DB2PATH>\java\Java12 下解除安裝 JDK/JRE，或更改 <DB2PATH>\java\Java12 次目錄的名字。

註：不要混淆 <DB2PATH>\java\Java12 和 <DB2PATH>\Java12。<DB2PATH>\Java12 是 DB2 安裝的一部份，並且包括 Java 1.2 的 JDBC 支援。

在 Windows 作業系統下使用線上說明的「無效捷徑」錯誤

使用控制中心線上說明，您可能會遇到這樣的錯誤：「無效的捷徑」。如果您最近安裝了新的 Web 瀏覽器或 Web 瀏覽器新版，請確定 HTML 和 HTM 文件和更改過的瀏覽器相結合。請參閱視窗說明主題「開啓檔案時變更程式的啟動」。

OS/2 系統下的 Java 控制中心

控制中心必須安裝在 HPFS 格式化的磁碟機上。

在 Windows 作業系統之異動日誌檢視已完成工作的「檔案拒絕存取」錯誤

在 DB2 Universal Database for Windows NT 上，在試圖開啓異動日誌來檢視在「Script 中心」所建立的工作明細時，發生「檔案拒絕存取」的錯誤。工作狀態顯示完成。若「Script 中心」中建立的工作含有 START 指令，便會發生這樣的行爲。爲了避免這樣的行爲，請在批次檔和工作內，使用 START/WAIT 指令，而不要使用 START 指令。

多位置更新測試連接

多位置更新測試連接在版本 7 「控制中心」的功能，爲目標案例的版本所限制。「遠端」測試連接功能欲執行的目標案例，至少需要版本 7。欲在版本 6 執行多位置更新測試連接功能，請在目標案例中提到控制中心，並在那兒執行它。

DB2 for OS/390 的控制中心

利用 OS/390 的 DB2 UDB 控制中心來管理授權公用程式的使用。對於 OS/390 的 DB2 UDB，公用程式函數是獨立有順序特性的元素，在它被 DB2 控制中心管理之前，必須授權和安裝在環境內。

資料庫 DB2 for OS/390 子系統，控制中心會定義 "CC390" 資料庫，作爲控制中心的內部支援。請不要修改此資料庫。

雖然在控制中心目錄或資訊中心作業資訊中，沒有特別提到 DB2 for OS/390 版本 7.1，但是，該文件支援 OS/390 版本 7.1 DB2 的函數。很多 DB2 for OS/390 版本 6 的特殊函數，和 DB2 for OS/390 版本 7.1 相關。有些 DB2 for OS/390 版本 7.1 的特殊函數，在目錄中沒有版本名稱。在控制中心架構 DB2 for OS/390 版本 7.1 子系統之後，對所有該版本的文件都有存取權限。

欲自控制中心存取和使用 Generate DDL 函數，請安裝 Generate DDL 函數：

- 關於版本 5，使用 DB2 for OS/390 版本 5 來安裝 DB2Admin。
- 關於版本 6，安裝小的程式設計加強功能，對於 DB2 for OS/390 版本 7.1 管理特性而言就如 PTF 般。
- 關於版本 7.1，Generate DDL 函數爲獨立定價之 DB2 for OS/390 版本 7.1 管理特性之一部份。

欲從控制中心存取儲存程序建置器，在啓動 DB2 UDB 控制中心時，必須進行安裝。它是 DB2 應用程式開發從屬站的一部份。

欲在工作站爲 OS/390 子系統編目 DB2，請選擇使用從屬站架構輔助程式工具。

1. 在「來源」頁面，指定**自行架構與資料庫的連線**圓鈕。
2. 在「通信協定」頁面，完成適當的通信資訊。
3. 在「資料庫」頁面，於**資料庫名稱**欄位指定子系統名稱。
4. 在「節點選項」頁面，選取**架構節點選項 (選用的)**勾選框。
5. 在**作業系統**欄位選取「MVS/ESA, OS/390」。
6. 按一下**完成**，以完成架構。

若要透過閘道機器來編目 DB2 for OS/390 子系統，請遵循上述閘道機器的步驟 1-6。然後：

1. 在從屬站機器，啟動控制中心。
2. 在「系統」資料夾上按一下滑鼠右鍵，再選取**新增**。
3. 在「新增系統」對話框內，在**系統名稱**欄位鍵入閘道機器名稱。
4. 在**遠端案例**欄位鍵入 DB2DAS00。
5. 針對 TCP/IP 通信協定 (在「通信協定」參數裡)，請在**主電腦名稱**欄位指定閘道機器的主電腦名稱。
6. 在**服務程式名稱**鍵入 523。
7. 按一下**確定**來新增系統。您應該可以在**系統**資料夾下面看到新增的閘道機器。
8. 展開閘道機器名稱。
9. 在「案例」資料夾上按一下滑鼠右鍵，再選取**新增**。
10. 在「新增案例」對話框中，按一下**復新**，列出閘道機器上可用的案例。如果閘道機器為 Windows NT 系統，則會在案例 DB2 下編目 DB2 for OS/390 子系統。
11. 選取案例。此案例會自動填寫通信協定參數。
12. 按一下**確定**來新增案例。
13. 開啓**案例**資料夾，可以看到剛才新增的案例。
14. 展開案例。
15. 在**資料庫**資料夾上按一下滑鼠右鍵，再選取**新增**。
16. 按一下**復新**，顯示閘道機器上的本端資料庫。如果您要在「新增資料庫」對話框中新增 DB2 子系統，請於「資料庫名稱」欄位鍵入子系統名稱。選項：輸入子系統 (或是資料庫) 的本端別名。
17. 按一下**確定**。

現在，您已經成功地在控制中心新增子系統。開啓資料庫時，會顯示 DB2 for OS/390 子系統。

『控制中心 390』 章節的第一個段落指出：

OS/390 的 DB2 UDB 控制中心可讓您管理已授權 IBM DB2 公用程式的使用。若為 DB2 UDB for OS/390 個別訂購特性的元件中的公用程式函數，在由 DB2 控制中心管理之前，必須經過授權且安裝在環境中。

此章節應改為：

OS/390 的 DB2 控制中心可讓您管理已授權 IBM DB2 公用程式的使用。若為個別訂購產品的元件中的公用程式函數必須經過授權且安裝在環境中，才能由 DB2 控制中心來管理。

OS/390 控制中心必要的修正程式

將 APAR PQ36382 引用到 DB2 for OS/390 版本 5 和 DB2 for OS/390 版本 6 的 390 賦能特性，使用 DB2 UDB 控制中心版本 7 來執行這些子系統的公用程式。

引用 APAR 至下列的 FMID：

DB2 for OS/390 Version 5 390 Enablement: FMID JDB551D
DB2 for OS/390 Version 6 390 Enablement: FMID JDB661D

「建立空間層」對話框的變更

"<<" 及 ">>" 按鈕已自「建立空間層」對話框中除去。

DB2 控制中心的疑難排解

在「快速入門」一書的「控制中心的安裝與架構」一章中，其「疑難排解資訊」一節會告訴您在將「控制中心」用為 applet 時若發生問題，應如何從命令視窗中重設您從屬站瀏覽器的 CLASSPATH。本節亦會說明如何從前述的命令視窗啟動瀏覽器，但卻未提供啟動瀏覽器的指令。若要啟動 Internet Explorer，請鍵入 `start iexplore`，然後按 **Enter** 鍵。若要啟動 Netscape，請鍵入 `start netscape`，然後按 **Enter** 鍵。這些指令係假設瀏覽器已設在您的 PATH 中。如若不然，請將其加入您的 PATH 中，或是切換到瀏覽器的安裝目錄，重新發出 **start** 指令。

UNIX 系統控制中心的疑難排解

如果您無法在 UNIX 型系統啟動控制中心，則設定 JAVA_HOME 環境變數指向 Java 分送：

- Java 安裝在 /usr/jdk118，請設定 JAVA_HOME 到 /usr/jdk118。
 - 對於 sh、ksh、bash shell：

```
匯出 JAVA_HOME=/usr/jdk118.
```
 - 對於 csh 或 tcsh shell：

```
setenv JAVA_HOME /usr/jdk118
```
-

OS/2 可能的 Infopop 問題

如果您是在 OS/2 上執行控制中心，使用解析度 1024x768 及 256 色，並啓用了 Workplace Shell Palette Awareness，則超出現行視窗邊框的 Infopop 會以黑色文字顯示在黑色背景下。欲修正此問題，變更顯示設定為 256 色以上，或停止使用 Workplace Shell Palette Awareness。

jdk11_path 架構參數的說明

在控制中心說明內，JDK 1.1 安裝路徑 (jdk11_path) 架構參數的說明，於子標題引用至下少了一行。「引用至」下完整的列示為：

- 本端和遠端從屬站的資料庫伺服器
 - 從屬站
 - 具有本端從屬站的資料庫伺服器
 - 具有本端及遠端從屬站的分割資料庫伺服器
 - 具有本端從屬站的衛星資料庫伺服器
-

使用「Script 中心」或「異動日誌」時發生的 Solaris 系統錯誤 (SQL10012N)

在選取「Script 中心」或「異動日誌」中的 Solaris 系統時，可能會發生下列錯誤：

```
SQL10012N - 當載入指定的檔案庫 "/udbprod/db2as/sql1lib/function/unfenced/  
db2scdar!ScheduleInfoOpenScan"時，接收到非預期的作業系統錯誤。  
SQLSTATE=42724。
```


這是由於 Solaris 執行鏈結器中的錯誤所導致。您只需引用下列的修補程式，即可更正此問題：

105490-06 (107733 淘汰了 105490) for Solaris 2.6

DPREPL.DFT 檔案的說明

在「控制中心」中，「工具設定」筆記本之「抄寫」頁的說明步驟 5d 寫道：

將檔案儲存到「控制中心」的工作目錄（例如，SQLLIB\BIN），
如此一來，系統可以使用其作為預設檔案。

應改寫為：

將檔案儲存到「控制中心」的工作目錄（SQLLIB\CC），
如此一來，系統可以使用其作為預設檔案。

啓動多個控制中心 applet

您不能在一台機器上同時啓動一個以上的控制中心 applet。這項限制適用於所有支援瀏覽器所執行的控制中心 applet。

控制中心以 Applet 執行時的線上說明

當「控制中心」是以 applet 方式執行時，F1 鍵只能夠在具有 Infopop 功能的視窗或筆記本中運作。

您可以在下列元件中按 F1 鍵，叫出 Infopop：

- DB2 Universal Database for OS/390
- 精靈

除上述兩項以外的「控制中心」元件皆無法使用 F1 鍵叫出說明。此時若要顯示其它元件的說明，請使用「說明」按鈕或「說明」下拉功能表。

在 Applet 模式 (Windows 95) 下執行控制中心

若指定的使用者 ID 和密碼無效，便無法開啓「Script 中心」。登入「控制中心」時，請確定使用者 ID 和密碼是否有效。

使用大的查詢結果

其方便使用者產生傳回大量列數的查詢。不太適合使用者用來預測實際傳回的列數。關於可能傳回數以千計或數以萬計列數的查詢，有下列兩個問題：

1. 需要花費很多時間才能取得結果。
2. 需要大量的從屬站記憶體才能保存結果。

爲了使這個處理更加便利，DB2 將大量結果集分割成許多片段。以一次一個片段的方式擷取及顯示查詢結果。

結果如下：

1. 當查詢的第一個片段可以檢視，同時已擷取剩餘片段時，會減少顯示時間。
2. 當在任何時指定時間上從屬站僅儲存查詢結果的一個片段，會減少從屬站的記憶體需求。

控制記憶體中的查詢結果列數：

1. 開啓「工具設定」筆記本的「一般」頁面。
2. 在**最大值**區段中，選取：
 - **範例內容**，可限制「範例內容」視窗中顯示的結果列數。在輸入欄位中指定結果集的片段大小 (列數)。
 - **命令中心**，可限制「命令中心」的「查詢結果」頁面上顯示的結果列數。在輸入欄位中指定結果集的片段大小 (列數)。

當使用「範例內容」視窗或「命令中心」的「查詢結果」頁面上的查詢結果時，**記憶體中列數**欄位指示保留在記憶體中的查詢列數。此數目絕不會大於**最大值**設定。按一下**下一步**，擷取至結果集的下一個片段。當**下一步**非作用中時，表示您已到達結果集尾端。

資訊中心

在 Windows 作業系統下的「無效捷徑」錯誤

使用資訊中心，您會發現如下的錯誤：「無效捷徑」。如果您最近安裝了新的 Web 瀏覽器或 Web 瀏覽器新版，請確定 HTML 和 HTM 文件和更改過的瀏覽器相結合。請參閱視窗說明主題「開啓檔案時變更程式的啓動」。

當 Netscape 已開啓時，在 Netscape Navigator 中開啓外部 Web 鏈結(UNIX 系統)

如果已經開啓 Netscape Navigator，並顯示本端 DB2 HTML 文件或外部網站，試圖從「資訊中心」開啓外部網站將導致 Netscape 發生錯誤。錯誤會提到 "Netscape 找不到名稱爲 <external site> 的檔案或目錄"。

若要解決這個問題，請在開啓外部網站之前，關閉開啓的 Netscape 瀏覽器。Netscape 將重新啓動並起始外部網站。

請注意，利用已開啓的 Netscape 來開啓本端 DB2 HTML 文件不會發生這個錯誤。

啓動「資訊中心」的問題

在某些系統上，若您使用「開始功能表」、「第一個步驟」或 **db2ic** 指令來呼叫「資訊中心」，啓動速度會非常慢。若遇到這個問題，請啓動「控制中心」，然後選取說明 --> 資訊中心。

精靈

在建立資料庫精靈設定擴充大小

可以使用建立資料庫精靈來設定新建資料庫之使用者表格空間的 (不是目錄或暫時表格的表格空間) 擴充大小和預先提取大小參數。只有當使用者表格空間在「使用者表格」精靈指定至少一個配置區時，才會啓用此特性。

MQSeries 輔助精靈

DB2 版本 7.2 提供新的 MQSeries 輔助精靈。此精靈建立一個表格函數，使用 DB2 MQSeries Functions 來讀取 MQSeries 佇列，DB2 MQSeries Functions 亦為版本 7.2 中的新功能。根據您的設定，此精靈可將每一則 MQSeries 訊息視為一個有定界符號的字串，或一個固定長度的直欄字串。建立的表格函數可根據您的設定來剖析字串，將每一則 MQSeries 訊息傳回成一列表格函數。此精靈亦可讓您在表格函數之上建立概略表，預覽 MQSeries 訊息和表格函數結果。這個精靈可從「儲存程序建置器」或「資料倉儲中心」來啓動。

本精靈的需求為：

- MQSeries 版本 5.2
- MQSeries 應用程式傳訊介面 (AMI)
- DB2 MQSeries Function

關於這些需求的詳細資訊，請參閱第173頁的『MQSeries』。

關於範例與 MQSeries 輔助精靈的教學指導，請跳至 <http://www.ibm.com/software/data/db2/udb/ide> 的教學指導段落。

OLE DB 輔助精靈

此精靈協助您建立一個表格函數，從支援 Microsoft OLE DB 標準的另一個資料庫提供者來讀取資料。另外，您可建立一個 DB2 表格，由 OLE DB 表格函數來提供備妥的資料，您亦可建立 OLE DB 表格函數的概略表。這個精靈可從「儲存程序建置器」或「資料倉儲中心」來啓動。

本精靈的需求為：

- OLE DB 提供者 (如 Oracle 或 Microsoft SQL Server)
- OLE DB 支援功能

關於範例與 OLE DB 輔助精靈的教學指導，請跳至 <http://www.ibm.com/software/data/db2/udb/ide> 的教學指導段落。

第5篇 Business Intelligence

Business Intelligence 教學指導	203	建立 MQSeries 訊息的概略表	223
經過修訂的 Business Intelligence 教學指導	203	基本要求	223
資料倉儲中心管理手冊	205	限制	223
疑難排解	205	建立 MQSeries 訊息的概略表	223
設置 Excel 為倉儲來源	205	匯入 MQSeries 訊息和 XML 描述資料	224
定義及執行程序	205	基本要求	224
匯出描述資料對話	205	限制	224
為 Submit OS/390 JCL 工作串流程式 (VWPMVS)		匯入 MQSeries 訊息與 XML 描述資料	224
定義值	205	使用 MQSeries 使用者定義程式	225
變更資料倉儲範例附錄	205	錯誤回覆碼	225
資料倉儲中心訊息	206	錯誤日誌檔	225
在 DB2 OLAP Integration Server 中建立框架並載入		Microsoft OLE DB 與「資料異動服務」支援	226
資料	206	建立 OLE DB 表格函數的概略表	227
將 Classic Connect 與資料倉儲中心搭配使用	206	建立 DTS 資料包的概略表	227
資料倉儲中心環境結構	206	增量 COMMIT 搭配 replace 使用	227
使用 Invert 轉換程式	207	元件追蹤資料檔名	227
以 DB2 版本 7 倉儲代理程式存取 DB2 版本 5 資		AIX 及 Solaris 作業環境上 Sybase 來源需要的	
料	207	OPEN CLIENT	227
移轉 DB2 版本 5 伺服器	207	範例登錄更正	228
變更代理程式架構	207	第 3 章 設定倉儲來源	228
UNIX 倉儲代理程式	207	對映 Microsoft Access 中的備忘錄欄位至倉儲來	
Microsoft Windows NT、Windows 2000 和		源	228
OS/2 倉儲代理程式	207	第 10 章 維護倉儲資料庫	228
IBM ERwin 描述資料取出程式	208	鏈結表格至 DB2 UDB RUNSTATS 程式的步驟	
內容	208	次類型	228
軟體需求	209	預設倉儲控制資料庫	228
程式檔	209	倉儲控制資料庫管理視窗	229
建立標示語言檔	209	變更作用中倉儲控制資料庫	229
匯入標示語言檔到「資料倉儲中心」	211	建立和起始設定倉儲控制資料庫	229
匯入標示語言檔到「資訊型錄管理程式」	212	建立版本的 SQL 步驟	230
疑難排解	212	在「程序模型」視窗中，變更來源或目標	230
ERwin 與 DB2 資料倉儲中心的對映	213	新增說明至「資料倉儲中心」物件	230
ERwin 與資訊型錄管理程式的對映	214	執行「範例內容」	230
篩選「資料倉儲中心」的名稱與位址	216	編輯 Create DDL SQL 陳述式	230
基本要求	216	移轉 Visual Warehouse 商業檢視	231
Trillium 軟體系統元件	217	建立目標表格及主要鍵	231
將「Trillium 批次系統」與「資料倉儲中心」		使用 Merant ODBC 驅動程式	231
搭配使用	217	新建 ODBC 驅動程式	231
匯入 Trillium 描述資料	218	定義 OS/2 資料庫中的倉儲來源或目標	231
對映描述資料	219	監督倉儲控制資料庫的狀態	231
限制	219	使用 SQL 輔助程式與 TBC_MD 範例資料庫	232
撰寫 Trillium 批次系統 JCL 檔	219	使用 FormatDate 函數	232
在 UNIX 和 Windows 上撰寫 Trillium 批次系統		變更語言設定	232
Script 檔	220	使用「產生鍵值表格」轉換程式	232
定義 Trillium 批次系統步驟	220	維護資料庫連接	232
使用「Trillium 批次系統」使用者定義程式	220	設定遠端「資料倉儲中心」從屬站	233
錯誤處理常式	221	定義 DB2 for VM 倉儲來源	233
錯誤回覆碼	222	定義 DB2 for VM 或 DB2 for VSE 目標表格	233
日誌檔	223	啟用有定界符號的識別字支援	233
MQ Series 與「資料倉儲中心」的整合	223	Data Joiner 錯誤指出連結問題	233
		資料倉儲中心設定和執行抄寫	234

疑難排解秘訣	236	架構 OLAP Starter Kit 的 ODBC	258
存取來源和目標檔	236	在 UNIX 系統架構資料來源	258
支援的非 IBM 資料庫來源的新增事項	237	架構 ODBC 環境變數	258
在資料倉儲中心中手動建立資料來源	239	編輯 odbc.ini 檔案	258
使用 Common Warehouse Metadata Interchange (CWM) 匯入及匯出描述資料	239	新增資料來源到 odbc.ini 檔	259
簡介	239	ODBC 的 DB2 設定值範例	259
匯入描述資料	239	ODBC 的 Oracle 設定值範例	259
在執行匯入公用程式之後更新描述資料	241	在 UNIX 系統上架構 OLAP 描述資料型錄	259
匯出描述資料	241	在 Windows 系統上架構資料來源	260
OS/390 Runstats 公用程式步驟	242	在 Windows 系統上架構 OLAP 描述資料型錄	261
OS/390 載入公用程式步驟	243	在您的架構資料來源之後	262
Common Warehouse Metamodel (CWM) XML 支援	243	從 OLAP Starter Kit 桌上管理程式登入	262
處理模型產生器	243	Starter Kit 登入範例	263
綱目模型產生器	244	自行建立和架構 OLAP Starter Kit 的範例資料庫	263
必要輸入欄位	244	移轉應用程式至 OLAP Starter Kit 版本 7.2	264
「資料倉儲中心」發射台加強功能	244	已知問題與限制	264
列印步驟資訊到檔案	244	OLAP 試算表增益集 EQD 檔案遺失	266
Data Warehouse Center Application Integration Guide(無中文版)	245	資訊型錄管理程式管理手冊	267
Additional metadata templates	245	資訊型錄管理程式起始設定公用程式	267
Commit.tag	245	版權問題	267
Tokens	246	安裝問題	267
Examples of values	246	以 DB2 版本 7 資訊型錄管理程式存取 DB2 版本 5	
ForeignKey.tag	246	資訊型錄	268
Tokens	246	設置資訊型錄	269
Examples of values	248	與其它產品交換描述資料	269
ForeignKeyAdditional.tag	248	使用 flgnxoln 指令交換描述資料	269
Tokens	249	使用 MDISDGC 指令交換描述資料	269
Examples of values	250	呼叫程式	269
PrimaryKey.tag	250	Information Catalog Manager Programming Guide and Reference(無中文版)	271
Tokens	250	Information Catalog Manager Reason Codes	271
Examples of values	251	資訊型錄管理程式使用手冊	273
PrimaryKeyAdditional.tag	252	資訊型錄管理程式：線上訊息	275
Tokens	252	訊息 FLG0260E	275
Examples of values	253	訊息 FLG0051E	275
資料倉儲中心線上說明	255	訊息 FLG0003E	275
定義抄寫的表格或概略表	255	訊息 FLG0372E	275
執行具有 AS/400 代理程式的 Essbase VWPs	255	訊息 FLG0615E	275
使用公佈資料倉儲中心描述資料視窗及已結合的內容視窗	255	資訊型錄管理程式：線上說明	277
外來鍵	255	Web 資訊型錄管理程式	277
抄寫筆記本	255	DB2 Warehouse Manager 安裝手冊	279
匯入標示語言	255	倉儲轉換程式的軟體需求	279
新增資料的鏈結	255	SAP R/3 Connector	279
匯入表格	255	安裝先決條件	279
RUNSTATS 和 REORGANIZE TABLE 線上說明集合	256	Web Connector	279
「通知」頁面，「倉儲內容」與「排程」筆記本	256	安裝先決條件	279
「代理程式端」筆記本的代理程式模組欄位	256	Query Patroller Administration Guide(無中文版)	281
DB2 OLAP Starter Kit	257	DB2 Query Patroller Client is a Separate Component	281
OLAP Server 網站	257	Migrating from Version 6 of DB2 Query Patroller	
支援的作業系統服務程式等級	257	Using dqpmigrate	281
完成 DB2 OLAP Starter Kit 在 UNIX 上的安裝	257		

Enabling Query Management.	281
Location of Table Space for Control Tables.	281
New Parameters for dqpstart Command	282
New Parameter for iwm_cmd Command	282
New Registry Variable:	
DQP_RECOVERY_INTERVAL	283
Starting Query Administrator.	283
User Administration	283
Creating a Job Queue	283
Using the Command Line Interface	283
Query Enabler Notes	284
DB2 Query Patroller Tracker may Return a Blank	
Column Page.	284
Query Patroller and Replication Tools.	284
Appendix B. Troubleshooting DB2 Query Patroller	
Clients	284

Business Intelligence 教學指導

經過修訂的 Business Intelligence 教學指導

FixPak 2 包括了修訂過的 Business Intelligence 教學指導及資料倉儲中心範例資料庫 (更正了許多版本 7.1 中所存在的問題)。爲了引用修訂過的資料倉儲中心範例資料庫，請執行下列事項：

如果您尚未安裝範例資料庫，請使用「第一個步驟」發射台來建立新的範例資料庫。按一下 **開始**，選取 **程式集** → **IBM DB2** → **第一個步驟**。

如果您在先前已安裝了範例資料庫，請將範例資料庫 DWCTBC、TBC_MD 及 TBC 捨棄。如果您對範例資料庫新增了任何要保存的資料，請在捨棄資料庫之前將資料備份。若要捨棄這三個範例資料庫：

1. 開啓 DB2 命令視窗，按一下**開始**，再選取 **程式集** → **IBM DB2** → **命令視窗**。
2. 在 DB2 命令視窗中，鍵入下列三行命令 (鍵入每一行後按 Enter)：

```
db2 drop database dwctbc
db2 drop database tbc_md
db2 drop database tbc
```
3. 關閉 DB2 命令視窗。
4. 使用「第一個步驟」發射台來建立新的範例資料庫。按一下 **開始**，選取 **程式集** → **IBM DB2** → **第一個步驟**。

資料倉儲中心管理手冊

疑難排解

資料倉儲中心疑難排解資訊移至 DB2 Troubleshooting Guide。

設置 Excel 為倉儲來源

在「第 3 章 設定倉儲來源」的「在 Windows NT 上設置非 DB2 資料庫倉儲來源」中，有關 Microsoft Excel 的段落缺少一個步驟。新的步驟顯示在以下的步驟 3 之中。

如果您正在使用 Microsoft Excel 95/97 ODBC 驅動程式存取 Excel 試算表，您必須為試算表中的每個工作表建立一個已命名的表格。若要為每一個工作表建立已命名的表格：

1. 選取您要的直欄和橫列。
2. 按一下 Excel → 插入 → 名稱 → 定義。
3. 請確認「定義名稱」視窗的「參照」欄位包含了您在步驟 1 中所選取的資料格。如果沒有，請按一下「參照」欄位最右邊的圖示，來包含所有您已選取的資料格。
4. 為已標記的資料輸入名稱 (或使用預設名稱)。
5. 按一下「確定」。

定義及執行程序

在「第 5 章 定義及執行程序」的「從資料倉儲中心外啟動步驟」中，請注意到在倉儲伺服器工作站及代理程式端上需要 JDK 1.1.8 或更新的版本，如果您啟動的步驟有雙位元組的名稱。

匯出描述資料對話

在第 12 章「匯出及匯入資料倉儲中心描述資料」的「將描述資料匯出至標示語言檔」這一節中，步驟 5 應如下所示：

如果您不想匯出與您正匯出之程序相關的時程表資訊，請清除**併入時程表**勾選框。

為 Submit OS/390 JCL 工作串程式 (VWPMVS) 定義值

在 180 頁，「定義 Submit OS/390 JCL 工作串程式」(VWPMVS) 章節內步驟 8，說明在 JES 檔案同一個目錄下定義 .netrc 檔案。該程式卻建立了 .netrc 檔案。如果這個檔案不存在，該程式會在起始目錄下建立它。如果 .netrc 檔案已經存在，則該程式會更改現存檔的檔名並建立新檔案。程式完成處理程序時，新建立的 .netrc 檔案會被刪除，再把原來的檔案的檔名更改為 .netrc。

變更資料倉儲範例附錄

- 在資料倉儲範例附錄內，「檢視和修改範例描述資料」一節中，來源表格的列示內含 GEOGRAPHIES 表格。

- 在資料倉儲範例附錄的「提昇步驟」一節中，提昇產品模式程序內，下列陳述式是不正確的，因為目標表格是在您提昇測試模式步驟時所建立：
資料倉儲中心開始建立目標表格，並顯示進度視窗。

資料倉儲中心訊息

在 Microsoft Windows NT 及 Windows 2000 上，「資料倉儲中心」會將事件記載在系統的事件日誌中。其事件 ID 會對應到「資料倉儲中心」的訊息碼上。有關「資料倉儲中心」訊息的資訊，請參閱 *訊息參考手冊*。

在 DB2 OLAP Integration Server 中建立框架並載入資料

第 315 頁的圖 20 有誤。下列指令才是正確：

```
"C:\IS\bin\olapicmd" < "C:\IS\Batch\my_script.script" >  
"C:\IS\Batch\my_script.log"
```

若路徑中的目錄名稱中含有空白（如：Program Files），您必須使用雙引號括住 "C:\IS\bin\olapicmd"。

將 Classic Connect 與資料倉儲中心搭配使用

- 在「附錄 F 將 Classic Connect 與資料倉儲中心搭配使用」中，第 359 頁的「安裝 CROSS ACCESS ODBC 驅動程式」章節的內容已改為下列敘述：

執行 DB2 Warehouse Manager 版本 7 的自訂安裝，以安裝 CROSS ACCESS ODBC 驅動程式，並選取「Classic Connect 驅動程式」元件。此驅動程式不屬於 DB2 Warehouse Manager 標準安裝的一部份。

CROSS ACCESS ODBC 驅動程式會被安裝到 SQLLIB 目錄的 ODBC32 子目錄中。

在安裝完成後，您必須自行將驅動程式的路徑（例如，C:\Program Files\SQLLIB\ODBC32）加到 PATH 系統環境變數。如果您已安裝其它版本的 CROSS ACCESS ODBC 驅動程式，把 ...\SQLLIB\ODBC32\ 路徑放到其它版本路徑的前面。作業系統會使用包括 CROSS ACCESS ODBC 驅動程式的路徑中的第一個目錄。

- 下列程序應該要加到「附錄 F 將 Classic Connect 與資料倉儲中心搭配使用」中：
安裝 Classic Connect ODBC 驅動程式：
 1. 插入 Warehouse Manager CD-ROM 到光碟機中。發射台就會開啓。
 2. 按一下發射台的**安裝**。
 3. 在「選取產品」視窗中，請確定選取了 **DB2 Warehouse Manager** 勾選框，然後按一下**下一步**。
 4. 在「選取安裝類型」視窗中，請選取**自訂安裝**，然後按一下**下一步**。
 5. 在「選取元件」視窗中，選取 **Classic Connect 驅動程式**及**倉儲代理程式**，清除所有其它的勾選框，然後按一下**下一步**。
 6. 在「開始複製檔案」視窗，複查您的選擇。如果您要變更任何選項，請按**上一步**回到要變更選項的視窗。按一下**下一步**，開始安裝。

資料倉儲中心環境結構

在第 371 頁的「附錄 G 資料倉儲中心環境結構」中，表格中有一個不正確的項目。C:\Program Files\SQLLIB\ODBC32並沒有加到 PATH 環境變數。PATH 環境變數所做的唯一更新為 C:\Program Files\SQLLIB\BIN。

使用 Invert 轉換程式

書中提到 Invert 轉換程式可以根據參數來建立目標表格，但是它沒有提到產生的目標表格不會有期望的輸出直欄，必須在目標表格中明白的建立這些直欄。

以 DB2 版本 7 倉儲代理程式存取 DB2 版本 5 資料

DB2 版本 7 倉儲代理程式由 DB2 版本 7 安裝程序架構，支援對 DB2 版本 6 和 DB2 版本 7 資料的存取。欲存取 DB2 版本 5 的資料，您必須採取下列其中一項方法：

- 移轉 DB2 版本 5 伺服器至 DB2 版本 6 或 DB2 版本 7。
- 於適當的作業系統中修改代理程式架構，可以存取 DB2 版本 5 的資料。

DB2 版本 7 倉儲代理程式不支援 DB2 版本 2 或之前版本資料的存取。

移轉 DB2 版本 5 伺服器

有關移轉移 DB2 版本 5 伺服器的資訊，請參閱作業系統的 *DB2 Universal Database 快速入門*。

變更代理程式架構

下列資訊說明如何變更各作業系統之代理程式架構。移轉 DB2 伺服器至 DB2 版本 6 或更新版時，除去架構的變更。

UNIX 倉儲代理程式

設定 UNIX 倉儲代理程式，CLI 或 ODBC 存取指令對 DB2 版本 5 之資料的存取：

1. 安裝 DB2 版本 6 執行時間從屬站。自下列 URL 選取從屬站下載，可以得到執行時間從屬站：
`http://www.ibm.com/software/data/db2/udb/support`
2. 更新倉儲代理程式架構檔，DB2INSTANCE 環境變數指向 DB2 版本 6 案例。
3. 編目 DB2 版本案例的所有資料庫，存取倉儲代理程式。
4. 發出代理常駐程式程序 ID 的刪除指令，停止代理常駐程式程序。代理常駐程式將自動重新啟動。刪除程序需要有 root 權限。

Microsoft Windows NT、Windows 2000 和 OS/2 倉儲代理程式

設定 Microsoft NT、Windows 2000 或 OS/2 倉儲代理程式，存取 DB2 版本 5 的資料。

1. 在沒有安裝 DB2 版本 7.2 倉儲代理程式的工作站，安裝 DB2 Connect Enterprise Edition 版本 6。

DB2 Connect Enterprise Edition 是 DB2 Universal Database Enterprise Edition 和 DB2 Universal Database Enterprise - Extended Edition 的一部分。如果安裝這些 DB2 產品的版本 6，無須另外安裝 DB2 Connect。

限制: 在相同的 Windows NT 或 OS/2 工作站，不能安裝 DB2 多重版本。您可以在其它 Windows NT 工作站、OS/2 工作站或 UNIX 工作站上安裝 DB2 Connect。

2. 架構倉儲代理程式和 DB2 Connect 版本 6，存取 DB2 版本 5 的資料。詳細資訊，請參閱 *DB2 Connect 使用手冊*。以下為必要步驟的概觀：

- a. 在 DB2 版本 5 系統，使用 DB2 命令行處理器編目版本 5 的資料庫，存取倉儲代理程式。
- b. 在 DB2 Connect 系統，使用 DB2 命令行處理器進行編目：
 - DB2 版本 5 系統的 TCP/IP 節點
 - DB2 版本 5 系統的資料庫
 - DB2 版本 5 系統的 DCS 登錄項目
- c. 於倉儲代理程式工作站，使用 DB2 命令行處理器進行編目：
 - DB2 Connect 系統的 TCP/IP 節點
 - DB2 Connect 系統的資料庫

有關編目資料庫的資訊，請參閱 *DB2 Universal Database 安裝與架構補充*

3. 於倉儲代理程式工作站，將 DB2 CLI 套裝軟體和透過 DB2 Connect 所存取的各個資料庫進行連結。

下列 DB2 指令提供連結到 v5database 的範例，一個假設性的 DB2 版本 5 的資料庫。使用 DB2 命令行處理器發出下列指令。db2cli.lst 及 db2ajgrt 的位置在 \sqllib\bnd 目錄中。

```
db2 connect to v5database user userid using password
db2 bind db2ajgrt.bnd
db2 bind @db2cli.lst blocking all grant public
```

其中 *userid* 是 v5database 的使用者 ID，而 *password* 是使用者 ID 的通行碼。

db2cli.list 被連結到 DB2 版本 5 資料庫時，會發生錯誤。發生錯誤，是因為此架構並不支援大型物件。這項錯誤不會影響倉儲代理程式對 DB2 版本 5 資料庫的存取。

需要有 DB2 Universal Database 版本 5 的 FixPak 14 (於 2000 年 6 月發行)，才能透過 DB2 Connect 來存取 DB2 版本 5 的資料。請參照 FixPak 內的 APAR 編號 JR14507。

IBM ERwin 描述資料取出程式

內容

軟體需求

程式檔

建立標示語言檔

匯入標示語言檔到「資料倉儲中心」

匯入標示語言檔到「資訊型錄管理程式」

疑難排解

ERwin 到「資料倉儲中心」對映

ERwin 到「資訊型錄管理程式」對映

本節說明如何使用 IBM ERwin Metadata Extract Program 來從 ER1 檔取出描述資料，以及建立「DB2 資料倉儲中心」或「資訊型錄管理程式 (DataGuide)」標示語言檔。

描述資料取出程式會取出所有實體物件 (如儲存於輸入 ER1 檔中的資料庫、表格及直欄)，並且將描述資料模型寫入「資料倉儲中心」或「資訊型錄管理程式」標示語言檔中。建立物件之間 (如資料庫和表格之間，以及表格和實體之間) 的所有對應關係標籤時，也會擷取及建立由實體及屬性組成的「資訊型錄管理程式」之邏輯模型。對不含資料庫的表格而言，會建立一個名為 DATABASE 的預設資料庫。對不含綱目的表格而言，會使用預設綱目 USERID。對模型名稱而言，會使用 ER1 檔名。有關 ER1 屬性對映至「資料倉儲中心」或「資訊型錄管理程式」的詳細資訊，請參閱「ERwin 與 DB2 資料倉儲中心的對映」和「ERwin 與資訊型錄管理程式的對映」。

描述資料取出程式支援具有關聯式資料庫的所有 ER1 模型，包括 DB2、Informix、Oracle、Sybase、ODBC 資料來源及 Microsoft SQL Server。

軟體需求

系統需符合下列的軟體需求以執行描述資料取出程式：

- Windows NT 4.0 或以上
- ERwin 3.5.2 (裝有 Service Pack 3 Build 466)

系統需符合下列的軟體需求以匯入 ERwin 標示語言檔：

資料倉儲中心：

IBM DB2 Universal Database 版本 7.2

資訊型錄管理程式：

IBM DB2 Warehouse Manager 版本 7.2

模板標示語言檔 (.tag) 必須位於 VWS_TEMPLATES 環境變數所指示的目錄中。

類型標示語言檔 (.typ) 必須位於 DGWPATH 環境變數所指示的目錄中。

程式檔

描述資料取出程式安裝在 IBM DB2 目錄的 sqllib\bin 次目錄中。此程式將下列檔案安裝到您的目錄中：

flgerwin.exe

主要的移轉程式

erwext.dll

標示語言檔產生器 DLL

cdmerwsn.dll

ERwin API 外層類別 DLL

若要啟動取出程式，請從指令提示上發出 flgerwin 指令。

建立標示語言檔

若要建立「資料倉儲中心」或「資訊型錄管理程式」標示語言檔，請執行 flgerwin.exe 程式並提供兩個主要參數。第一個參數為描述資料自其中取出的 ER1 檔。第二個參數為輸出標示語言檔的名稱。依預設，取出程式將 MERGE 參數加入「資料倉儲中心」標示語言檔中。

指令語法為：

```
flgerwin inputFile.er1 outputFile.tag [-dwc] [-icm] [-m] [-u] [-a] [-d]
```

建立星狀綱目的指令語法為：

```
flgerwin inputFile.er1 outputFile.tag [-dwc] [-starschema]
```

-dwc 請建立「資料倉儲中心」標示語言檔。您可使用 **-dwc** 的選用性參數 **-m** 與 **-starschema**。

-icm 請建立「資訊型錄管理程式」標示語言檔。**-icm** 可用的選用性參數為 **-m**、**-u**、**-a** 及 **-d**。

-starschema

請建立 ERwin 模型星狀綱目標示語言檔。

-m 指定物件上的動作為 MERGE。

-u 指定物件上的動作為 UPDATE。

-a 指定物件上的動作為 ADD。

-d 指定物件上的動作為 DELETE。

描述資料取出程式使用的是描述資料而非資料。完成 ERwin 標示語言檔匯入後及使用目標表格之前，您必須符合通行碼及使用者 ID。將描述資料與現存的資料庫資料合併：

在內容 --> 資料庫 --> 使用者 ID 下變更「資料倉儲中心」使用者 ID 和通行碼，使符合已合併的資料庫使用者 ID 和通行碼。

透過描述資料取出程式，您可以匯入標示語言檔作為目標。在最新匯入的描述資料中，尚未移入一些表格。您可以用邏輯或實體表示法檢視這些表格，然後建立一個倉儲步驟來移入從 ERwin 匯入的表格定義。

輸入 ER1 檔案必須處於可寫入狀態。執行描述資料取出程式後，ER1 檔案即變成唯讀狀態。若要變更為讀取/寫入模式，請使用如下所範例所示的指令：

```
attrib -r erwinsimplemode.er1
```

其中，erwinsimplemode.er1 是 ERwin 純文字檔的名稱。

若 ER1 檔案已使用於現行 ERwin 階段作業中，或偵測到某個錯誤狀況，則描述資料取出程式以唯讀狀態儲存 ER1 檔案。若 ER1 檔案處於唯讀狀態，您會收到一個異常程式終止錯誤訊息。描述資料取出程式會顯示目前處理的表格名稱。當描述資料取出程式完成處理時，您會收到一則參考訊息。

當您藉由將維度表格自動結合成事實表格來建立星狀綱目時，需要一段時間才能完成處理程序，而所需要的時間視您使用的表格數目而定。在處理程序期間，自動結合同行為綠色。儲存之後，自動結合同行會變成黑色。

使用自動產生的限制名稱來確保限制名稱是唯一的。

在處理程序期間，您會收到訊息“發現重複的直欄。直欄將無法被取出。”這是一個參考訊息，並不影響取出程式的順利完成。當外來鍵的實體名稱與目前所處理表格中的直欄之實體名稱相同時，會顯示此訊息。

匯入標示語言檔到「資料倉儲中心」

您可以匯入標示語言檔到「資料倉儲中心」。您可以使用「資料倉儲中心」或命令行。

若要使用「資料倉儲中心」匯入標示語言檔，請：

1. 按一下 **開始 -> 程式集 -> IBM DB2 -> 控制中心**。即會開啓 DB2「控制中心」。
2. 開啓「資料倉儲中心」並登入。
3. 在**倉儲**上按一下右鍵。即會開啓「匯入」視窗。
4. 按一下 **匯入描述資料 -> ERwin**。即可開啓**匯入描述資料**視窗。
5. 在**輸入檔**欄位中，鍵入輸入標示語言檔的名稱，並按一下**確定**。
6. 勾選**取出星狀綱目**勾選框，以定義 ERwin 星狀綱目描述資料模型爲倉儲綱目。

完成匯入之後，您可以按一下**檢視 -> 復新**以檢視新的步驟。

若要使用命令行匯入標示語言檔，請：

```
iwh2imp2 tag-filename log-pathname  
target-control-db userid password
```

tag-filename

標示語言檔的完整路徑和檔案名稱。

log-pathname

日誌檔的完整路徑名稱。

target-control-db

匯入的目標資料庫名稱。

userid 用來存取控制資料庫的使用者 ID。

password

用來存取控制資料庫的通行碼。

若要變更 DB2 資料庫定義使其成爲「資料倉儲中心」中的一個來源，您可以變更標示語言檔：

- 針對您要作爲來源的每一個資料庫，將 ISWH 標示從 ISWH(Y) 變更爲 ISWH(N)。
- 針對您要作爲來源的每一個資料庫，將關係標示從 **:RELTYPE.TYPE(LINK) SOURCETYPE(SCGTARIR) TARGETYPE(DATABASE)** 變更爲 **:RELTYPE.TYPE(LINK) SOURCETYPE(SCGSR CIR) TARGETYPE(DATABASE)**。

匯入標示語言檔後，您會收到下列訊息：

```
Message: DWC13238E The object of type "COLUMN" identified by "DBNAME(____)  
OWNER(____) TABLE(____) COLUMNS(____)" is defined twice in the tag language  
file.
```

這是一個參考訊息，您的匯入已順利完成。若您擁有的實體具有同名的外來鍵，或擁有的實體具有受截斷影響的類似已命名直欄，或其它類似的情況，您會收到此訊息。檢查您的模型是否有重複的直欄名稱，然後進行適當的調整。

匯入標示語言檔到「資訊型錄管理程式」

匯入標示語言檔到「資訊型錄管理程式」有兩個方法。您可以使用「資訊型錄管理者」或使用指令行。

使用「資訊型錄管理者」來匯入標示語言檔：

1. 按一下**開始** --> **程式集** --> **DB2** --> **資訊型錄管理程式**。
2. 按一下**型錄** --> **匯入**。即會開啓「匯入」視窗。
3. 按一下**尋找**來搜尋標示語言檔，然後按一下**匯入**。

完成匯入之後，您可以按兩下**主旨**圖示，這會開啓一個視窗，顯示所有匯入的模型及資料庫。

若要使用命令介面匯入標示語言檔，請輸入下列指令：

```
DGUIDE /USERID userid /PASSWORD password  
/DGNAME dgname /IMPORT filename /LOGFILE  
filename /ADMIN /RESTART (B|C)
```

/USERID

用來存取控制資料庫的使用者 ID。

/PASSWORD

使用者 ID 的通行碼。

/DGNAME

「資訊型錄管理程式」名稱。

/IMPORT

標示語言檔的完整路徑和檔案名稱。

/LOGFILE

日誌檔的完整路徑名稱。

/ADMIN

指示您以管理者身份登入。

/RESTART

指示匯入將從標示語言檔開頭處啓動 (選項 B)，或從上一個確定點啓動 (選項 C，預設值)。

疑難排解

若您收到錯誤訊息，請在此處的訊息中找到解決錯誤可採取的動作。

遺失的 ER1 輸入檔或標示輸出檔。

描述資料取出程式需要特定次序的兩個參數。第一個參數為 ER1 檔名，第二個參數為標示輸出檔名稱。若指定現存的標示語言檔名稱，則檔案會被改寫。

Windows 系統異常程式終止。

輸入 ER1 檔可能處於唯讀狀態。若儲存 ER1 檔時發生問題，而且描述資料取出程式將該檔案置於唯讀模式中，則會發生此情況。請在指令 Shell 中發出下列指令：

```
attrib -r inputFile.er1
```

藉以將 ER1 檔案狀態變更為讀取/寫入。

無法開啓標示語言檔。

檢查系統中有無造成檔案無法在現行磁碟機開啓或建立的任何問題。

找不到模板檔的路徑。

未設定環境變數 VWS_TEMPLATES。檢查「資料倉儲中心」是否已安裝。

找不到類型檔的路徑。

未設定環境變數 DGWPATH。檢查「資料倉儲中心」是否已安裝。

未支援的伺服器版本：...

您嘗試自其中取出的輸入 ER1 檔儲存在不受程式支援的目標伺服器上。請啓動 ERwin、開啓 ER1 檔，然後按一下**伺服器 --> 目標伺服器**，以及適當的版本 [請參閱軟體需求]。儲存 ER1 檔。

不明的 ERwAPI 錯誤。

發生 ERwin API 錯誤，程式無法取得錯誤的相關資訊。請確定 ERwin 3.5.2 已安裝。您必須登記 ERwin API。

若要登記 ERwin API，請從安裝 ERwin 程式檔所在的目錄執行下列指令：**regsvr32 er2api32.dll**。您會看到訊息 "DllRegisterServer in er2api32.dll succeeded"。您可以從「資料倉儲中心」啓動取出程式，或從指令 Shell 發出 flgerwin 指令來啓動取出程式。

取出程式錯誤：...

檢查錯誤訊息並採取適當動作。這最有可能是內部取出程式錯誤，請向 IBM 業務代表報告問題。

不明的取出程式錯誤。

發生不明的錯誤。這最有可能是內部錯誤，請向 IBM 業務代表報告問題。

取出程式因錯誤而終止。

發生無法順利完成取出程式的錯誤。請參閱附加的錯誤訊息以解決問題，或聯絡 IBM 業務代表。

ERwin 與 DB2 資料倉儲中心的對映

本節說明主要 ERwin 物件屬性如何對應到「資料倉儲中心」標示：

資料庫 - WarehouseDatabase.tag 或 SourceDatabase.tag

ERwin	命令行標示	資料倉儲中心
圖解名稱	NAME	倉儲來源或倉儲目標的名稱
圖解作者	RESPNSBL	聯絡
資料庫名稱	DBNAME	資料庫名稱

ERwin	命令行標示	資料倉儲中心
資料庫版本	DBTYPE	資料庫類型
圖解說明	SHRTDESC	說明

表格 - Table.tag

ERwin	命令行標示	資料倉儲中心
表格名稱	NAME	表格名稱
表格名稱	TABLES	表格名稱
資料庫名稱	DBNAME	無
表格擁有者	OWNER	表格綱目
表格註解	SHRTDESC	說明

直欄 - Column.tag

ERwin	命令行標示	資料倉儲中心
直欄名稱	NAME	直欄名稱
資料類型	NATIVEDT	資料類型
長度	LENGTH	長度
小數位數	SCALE	小數位數
Null 選項	NULLABLE	接受空值 (勾選框)
位置	POSNO	無
主要鍵	KEYPOSNO	無
資料庫名稱	DBNAME	無
表格擁有者	OWNER	無
表格名稱	TABLES	無
直欄註解	SHRTDESC	說明

ERwin 與資訊型錄管理程式的對映

本節說明主要 ERwin 物件屬性如何對應到「資訊型錄管理程式」標示：

資料庫 - Database.tag

ERwin	命令行標示	「資訊型錄管理程式」介面
圖解名稱	NAME	資料庫名稱
圖解作者	RESPNSBL	資料庫擁有者
資料庫名稱	DBNAME	資料庫名稱
資料庫版本	DBTYPE	資料庫類型
圖解說明	SHRTDESC	簡短的說明

表格 - TableOrView.tag

ERwin	命令行標示	「資訊型錄管理程式」介面
表格名稱	NAME	表格名稱

ERwin	命令行標示	「資訊型錄管理程式」介面
表格名稱	TABLES	表格名稱
資料庫名稱	DBNAME	資料庫名稱
表格擁有者	OWNER	表格擁有者
表格註解	SHRTDESC	簡短的說明
ERwin API	TABVIEW	定義代表一個概略表

直欄 - ColumnOrField.tag

ERwin	命令行標示	「資訊型錄管理程式」介面
直欄名稱	NAME	直欄名稱
資料類型	DATATYPE	直欄的資料類型
長度	LENGTH	直欄的長度
小數位數	SCALE	直欄的小數位數
Null 選項	NULLS	直欄可否為空值 (?)
位置	POSNO	直欄位置
主要鍵	KEYPOSNO	主要鍵中直欄的位置
ERwin API	ISKEY	直欄是否為鍵的一部份 (?)
ERwin API	UNIQKEY	直欄是否為唯一鍵 (?)
資料庫名稱	DBNAME	資料庫名稱
表格擁有者	OWNER	表格擁有者
表格名稱	TABLES	表格名稱
直欄註解	SHRTDESC	簡短的說明
ERwin	ISTEXT	資料是否為文字 (?)
ERwin API	IDSRES	資料的解析度

模型 - Model.tag

ERwin	命令行標示	「資訊型錄管理程式」介面
ER1 檔名	NAME	模型名稱
圖解作者	RESPNSBL	進一步資訊...
圖解說明	SHRTDESC	簡短的說明

實體 - Entity.tag

ERwin	命令行標示	「資訊型錄管理程式」介面
實體名稱	NAME	實體名稱
附註	SHRTDESC	簡短的說明
定義	LONGDESC	詳細說明
實體擁有者	RESPNSBL	進一步資訊...

屬性 - Attribute.tag

ERwin	命令行標示	「資訊型錄管理程式」介面
物件名稱	NAME	物件名稱
附註	SHRTDESC	簡短的說明
定義	LONGDESC	詳細說明
資料類型	DATATYPE	成員的資料類型
長度	LENGTH	成員的長度

篩選「資料倉儲中心」的名稱與位址

使用「資料倉儲中心」和「Trillium 軟體系統」來篩選名稱與位址資料。「Trillium 軟體系統」是一個名稱與位址篩選產品，可以重新格式化、標準化及驗證名稱和位址資料。藉由從使用者定義程式啟動「Trillium 批次系統」程式，您可以在「資料倉儲中心」中使用「Trillium 軟體系統」。當您從「Trillium 批次系統」Script 或 JCL 匯入描述資料時，會將使用者定義程式新增至「倉儲」樹狀結構中。

「資料倉儲中心」已提供與 Vality and Evolutionary Technologies, Inc 的工具之整合。

基本要求

- 您必須在倉儲代理程式端或遠端主電腦上安裝「Trillium 軟體系統」。
- 在 UNIX 和 Windows 平台上，必須將「Trillium 軟體系統」的 bin 目錄路徑新增至系統環境變數 PATH 中，才能啟用代理程式程序來執行「Trillium 批次系統」程式。在 UNIX 上，啟動 vwdaemon 程序之前，必須將 PATH 變數加入 IWH.environment 檔中，以完成執行。
- 使用者必須具備 Trillium 軟體的使用常識。

下表顯示軟體需求。

作業系統	必要的軟體
UNIX	Trillium 軟體系統 版本 4.0 「資料倉儲管理程式」版本 7.2 倉儲代理程式
Windows NT 與 Windows 2000	Trillium 軟體系統 版本 4.0 「資料倉儲管理程式」版本 7.2 倉儲代理程式 以遠端存取而言，主電腦必須安裝 ftpd 和 rexecd 常駐程式。
OS/390	安裝在遠端 OS/390 主電腦的「Trillium 軟體系統」版本 4.0 安裝在 UNIX、Windows NT 上的「資料倉儲管理程式」版本 7.2 倉儲代理程式 必須安裝 TCP/IP 3.2 (或以上) 僅支援為遠端主電腦的 OS/390 作業系統

Trillium 軟體系統元件

「Trillium 軟體系統」由四個主要元件組作：轉換器、剖析器、geocoder 及匹配器。使用這些元件為一組功能，以執行名稱與位址篩選作業。您可以從「Trillium 批次系統」（一個使用者定義程式）執行這些元件。

轉換器 使用轉換器來標準化及將來源資料轉換成指定的輸出格式。

剖析器 使用剖析器來解譯名稱和位址來源資料，以及建立來源資料相關的描述資料。

地理編碼程式 (Geocoder)

使用 geocoder 將來源資料與郵遞服務程式資料做比較，以提供任何遺漏資訊，例如寄件者或 ZIP+4 字碼。geocoder 也可透過「美國統計調查」資料來執行比對作業。

匹配器 使用匹配器來比較類似名稱和位址以識別重複記錄。您可以使用匹配器執行參照比對，以比較一個記錄與一組記錄。

將「Trillium 批次系統」與「資料倉儲中心」搭配使用

在「資料倉儲中心」中，您可以匯入「Trillium 批次系統」描述資料及建立使用者定義程式步驟。此步驟會呼叫本端倉儲代理程式端或遠端倉儲代理程式端上的「Trillium 批次系統」Script。在「資料倉儲中心」中，「Trillium 批次系統」Script 是一個具有來源及目標檔的步驟。來源檔是用於第一個「Trillium 批次系統」命令的輸入資料檔。目標檔是 Script 中最後一個 Trillium 命令所建立的輸出資料檔。然後，可複製此步驟至另一個處理程序，搭配其它步驟使用。

下圖顯示「Trillium 批次系統」輸入及輸出資料檔與「資料倉儲中心」中來源及目標檔之間的關係。

```
REM Running the converter
pfcondrv -parmfile c:\tril40\us_proj\parms\pfcondrv.par
REM Running the parser
pfprsdrv -parmfile c:\tril40\us_proj\parms\pfprsdrv.par
REM Running the Matcher
cfmatdrv -parmfile c:\tril40\us_proj\parms\pfmatdrv.par
```

圖 1. 範例 Trillium Script 檔

```
INP_FNAME01 c:\tril40\us_proj\data\convinp
INP_DDL01 c:\tril40\us_proj\dict\input.dd1
```

圖 2. pfcondrv.par 檔的內容

```
OUT_DDNAME c:\tril40\us_proj\data\maout
DDL_OUT_FNAME c:\tril40\us_proj\dict\parseout.dd1
```

圖 3. pfmatdrv.par 檔的內容

```
c:\Tril40\us_proj\data\convinp (source file) ->
Trillium Batch System Step -> c:\tril40\us_proj\data\maout (target file)
```

圖 4. 「Trillium 批次系統」步驟定義

匯入 Trillium 描述資料

若要匯入 Trillium 描述資料到「資料倉儲中心」：

1. 建立「Trillium 批次系統 Script」或 JCL。您可以使用任何 Script 或 JCL 撰寫工具來建立 Script 或 JCL 檔。
2. 以滑鼠右鍵按一下倉儲，然後按一下匯入描述資料 -> Trillium，開啓「Trillium 批次系統」視窗。
3. 在 **Script 或 JCL** 欄位中，鍵入您要執行的「Trillium 批次系統」Script 或 JCL 檔之名稱。
4. 在**輸入檔**欄位中，鍵入第一次在指定的 Script 或 JCL 檔中執行之「Trillium 批次系統」程式的輸入資料檔名稱。
5. 在**輸入 DDL** 欄位中，鍵入說明輸入資料檔的輸入 DDL 檔之名稱。此檔案必須可在倉儲代理程式端使用。
6. 在**輸出檔**欄位中，鍵入 Script 或 JCL 檔中最後一個「Trillium 批次系統」程式的輸出資料檔名稱。
7. 在**輸出 DDL** 欄位中，鍵入說明輸出資料檔的輸出 DDL 檔之名稱。此檔案必須可在倉儲代理程式端使用。
8. 選用的：在**輸出錯誤檔**欄位中，鍵入您要使用的輸出錯誤檔名稱。此錯誤檔會從「Trillium 批次系統」程式擷取執行期錯誤。這些錯誤記錄於 stderr 日誌中。以本端主電腦而言，若您未在此指定名稱，則會建立預設輸出錯誤檔。有關輸出錯誤檔的詳細資訊，請參閱主題「錯誤處理常式」。
9. 按一下**連線**標籤。
10. 若您匯入的 Trillium 描述資料位於倉儲代理程式端上，請按一下**本端主電腦**。若您匯入的 Trillium 描述資料不是位於倉儲代理程式端上，請按一下**遠端主電腦**，然後指定遠端主電腦。請參閱本節稍後提到的主題「指定遠端主電腦」。
11. 按一下**確定**，以匯入 Trillium 描述資料並關閉該筆記本。
12. 若 Script 或 JCL 不是從預設代理程式端執行，請在「內容」筆記本中指定用於「Trillium 批次系統」步驟的倉儲代理程式端。

匯入作業完成後，下列倉儲物件會新增至「倉儲」樹狀結構中。

- Trillium 批次系統 *scriptName* 模板，其中 *scriptName* 是 Script 或 JCL 檔的名稱。
- 「Trillium 批次系統」處理程序。
- 執行使用者定義程式的「Trillium 批次系統」步驟。
- 匯入描述資料時指定的倉儲檔案來源及目標。檔案來源及檔案目標為固定檔案。
- 「Trillium 批次系統」程式群組。

指定遠端主電腦

若要指定遠端主電腦，請：

1. 按一下**遠端主電腦**，然後鍵入含有您匯入的描述資料之遠端系統的 TCP/IP 主電腦名稱。若選取「遠端主電腦」，則目標檔被建立為本端檔案，因為遠端目標檔不受支援。您可以新增 FTP 步驟來使遠端檔案成為指定的本端目標檔。
2. 在**遠端作業系統**清單中，按一下您存取的遠端主電腦之作業系統。
3. 在**遠端使用者 ID** 欄位中，鍵入您存取的遠端主電腦之使用者 ID。

4. 在**通行碼選項**清單中，選取用於所要存取的遠端主電腦之通行碼選項：

不需要通行碼

指定存取遠端主電腦的描述資料不需要通行碼。

擷取通行碼

指定從使用者定義程式擷取通行碼。

在**通行碼程式**欄位中，鍵入將擷取通行碼的通行碼程式名稱。此程式必須位於倉儲代理程式端，並且將通行碼寫入輸出檔的第一行中。

在**程式參數**欄位中，鍵入通行碼程式的參數。第一個參數必須是寫入通行碼的輸出檔。

稍後輸入通行碼

指定稍後再輸入通行碼。

在執行「Trillium 批次系統」程式的步驟之「內容」筆記本中，輸入通行碼。

對映描述資料

為了建立來源和目標檔的描述資料，Trillium 會讀取 Trillium DDL 檔。此 DDL 檔會轉換成下列「資料倉儲中心」資料類型：

UNIX、Windows NT 及作業系統的 DDL 資料類型	倉儲資料類型
ASCII CHARACTER	CHARACTER(<i>n</i>)
ASCII NUMERIC	
EBCDIC CHARACTER	
EBCDIC NUMERIC	
其它類型	NUMERIC
註： 唯有「Trillium 軟體系統」在 OS/390 作業系統上執行，EBCDIC CHARACTER 和 EBCDIC NUMERIC 資料類型才受到支援。	

變數 *n* 代表字串中的字元數。

限制

您可以透過「資料倉儲中心」中的匯入描述資料作業及 Trillium DDL，在輸入及輸出 DDL 檔中指定一些重疊欄位。不過，相對應的倉儲來源及倉儲目標檔案不能在「資料倉儲中心」中搭配 SQL 步驟或範例內容使用。因為匯入描述資料作業會忽略越過整個記錄的重疊欄位，所以您仍可指定這些欄位，但它們在結果來源及目標檔中不能當作直欄使用。

若指定了錯誤檔，則 Script 的名稱不能包含任何空格。

撰寫 Trillium 批次系統 JCL 檔

若要撰寫 Trillium 批次系統 JCL 檔，下列需求必須符合。

- 工作名稱必須是使用者 ID 加上一個字元。
- 工作必須遞送至已保留輸出類別中。

- 執行「Trillium 批次系統」程式的每一個工作步驟，必須併入定義永久資料集的 `SYSTEM DD` 陳述式。此資料集包含「Trillium 批次系統」程式的錯誤。提出 JCL 之前，此資料集會自動被刪除。有關錯誤處理常式及報告的詳細資訊，請參閱主題「錯誤處理常式」。

當 Script 或 JCL 在遠端主電腦上執行時，必須指定輸出錯誤檔；否則，錯誤訊息不會被記錄及傳送至「資料倉儲中心」。在 UNIX 或 Windows 上，擷取錯誤訊息最簡單的方法是撰寫另一個呼叫「Trillium 批次系統」Script 的 Script，並將標準錯誤傳送至輸出檔。

```
//SYSTEM DD UNIT=&UNIT,
//          DISP=(MOD,CATLG,KEEP),
//          SPACE=(400,(20,20),,ROUND),
//          DSN=&PROJPREF.&TRILVER.&PROJECT.STDERR;
```

圖 5. 併入 `SYSTEM DD` 陳述式的工作步驟範例

在 UNIX 和 Windows 上撰寫 Trillium 批次系統 Script 檔

若「Trillium 批次系統」Script 或參數檔包含輸入檔的相對路徑，使用者必須將 Script 檔開頭的 `cd` 陳述式放置到 Script 檔的目錄中。

定義 Trillium 批次系統步驟

在定義「Trillium 批次系統」步驟之前，您必須將所要使用的 Trillium 描述資料匯入處理程序中。新增「Trillium 批次系統」步驟至處理程序中：

1. 開啓處理程序模型產生器中的處理程序。
2. 按一下選用區上的 **Trillium 批次系統** 圖示。
3. 按一下 **Trillium 批次系統程式** -> *programName*，其中 *programName* 是您要使用的「Trillium 批次系統」程式名稱。
4. 按一下畫布上您要顯示步驟所在的位置。
5. 完成 *DB2 Universal Database* 說明中主題「定義一個執行使用者定義程式的步驟」內的步驟。

使用「Trillium 批次系統」使用者定義程式

「Trillium 批次系統」使用者定義程式隨附於 Windows NT 和 UNIX 的「DB2 資料倉儲中心」版本 7.2 中。您匯入 Trillium 描述資料時所建立的「Trillium 批次系統」步驟，將執行「Trillium 批次系統」使用者定義程式。使用者定義程式會呼叫「Trillium 批次系統」Script 或 JCL。下表包含用於「Trillium 批次系統」Script 或 JCL 的參數：

參數	值
遠端主電腦	<ul style="list-style-type: none"> • localhost 為預設值。若「Trillium 批次系統」安裝在倉儲代理程式端，則使用此值。 • 若「Trillium 批次系統」安裝在遠端作業系統上，則使遠端主電腦名稱。
Script 或 JCL	Script 或 JCL 的名稱

遠端作業系統	遠端主電腦上作業系統的名稱。若遠端主電腦參數為 localhost，則系統不處理此參數。有效值為： <ul style="list-style-type: none"> • MVS for OS/390 作業系統 • UNIX for AIX、SUN Solaris、HP-UX 及 NUMA/Q 作業系統 • WIN for Windows NT 或 2000 作業系統
遠端使用者 ID	擁有執行遠端命令的權限之使用者 ID。若 RemoteHostName 的值為 localhost，則系統不處理此參數。
通行碼選項	取得通行碼的方法。有效值為： ENTERPASSWORD 若通行碼在下一個參數傳遞，則使用此值。 PASSWORDNOTREQUIRED 若不需要通行碼，則使用此值。 GETPASSWORD 若程式名稱在下一個參數傳遞，則使用此值。 限制： <ul style="list-style-type: none"> • 此程式必須位於代理程式端、將通行碼寫入輸出檔的第一行中，以及在順利執行後傳回 0。 • 通行碼參數值必須是通行碼程式的名稱。 • 程式參數值必須是以雙引號括住的字串。 • 字串中第一個參數必須是寫入通行碼所在的輸出檔名稱。
通行碼	有效值為通行碼或通行碼程式名稱。通行碼程式必須位於倉儲代理程式端。
程式參數	通行碼程式的參數。
輸出錯誤檔	輸出錯誤檔的名稱。
註： 此表格中所有參數的資料類型為 CHARACTER。	

錯誤處理常式

在 Windows NT 及 UNIX 作業系統上，「Trillium 批次系統」程式將錯誤訊息寫入標準錯誤 (stderr) 檔中，而在 OS/390 作業系統上，是寫入 SYSTEMM 資料集中。

若要擷取 Windows NT 或 UNIX 作業系統的「Trillium 批次系統」程式錯誤，必須將標準錯誤重新導向至輸出錯誤檔。

若要擷取 OS/390 作業系統的「Trillium 批次系統」程式錯誤，JCL 必須併入 SYSTEMM DD 陳述式。

若在「匯入描述資料」視窗中指定輸出錯誤檔名，您必須重新導向或將標準錯誤輸出儲存至錯誤檔中。「資料倉儲中心」會讀取此檔案並傳回內含 ERROR 字串的所有行為錯誤訊息。所有「Trillium 批次系統」程式錯誤訊息都包含 ERROR 字串。

若倉儲代理程式端上執行的 Script 或 JCL 中未指定輸出錯誤檔，則「資料倉儲中心」會自動建立一個檔名並將標準錯誤輸出重新導向至該檔案。若發現任何錯誤，則不會刪除錯誤檔。錯誤檔儲存於環境變數 VWS_LOGGING 所指定的目錄中。該檔名為 tbsudp-date- time.err，其中 date 是建立檔案的系統日期，而 time 是建立檔案的系統時間。下列檔名顯示輸出錯誤檔名的格式：

tbsudp-021501-155606.err

錯誤回覆碼

錯誤碼	說明
0	成功時
4	警告：通行碼檔案可能無法消除，或「Trillium 批次系統」使用者定義程式存取暫存檔時發生內部錯誤。 請檢查通行碼檔案狀態，或環境變數 VWS_LOGGING 所指定目錄下建立的所有暫存檔狀態。
8	參數值或數目不正確。請參考日誌檔或文件來取得正確語法。
12	「Trillium 批次系統」使用者定義程式透過 FTP 連接遠端主電腦時發生問題。檢查 FTP 連線或主電腦名稱、使用者 ID 及通行碼。
16	「Trillium 批次系統」使用者定義程式無法建立日誌或內部檔案。 檢查使用者有無正確授權及磁碟空間是否已滿。
20	無法執行 OS/390 JCL，或「Trillium 批次系統」使用者定義程式透過 FTP 從 OS/390 刪除或取得檔案時發生錯誤。 檢查 JESLogFile 來找出錯誤原因。
48	找不到環境變數 VWS_LOGGING 或無法建立日誌檔。請參考日誌檔以取得詳細資訊。
56	無法執行 Windows NT 或 UNIX Script，或「Trillium 批次系統」使用者定義程式連接遠端主電腦時發生錯誤。檢查連線或主電腦名稱、使用者 ID 及通行碼。
500	Script 或 JCL 檔傳回錯誤，或未傳回錯誤，但錯誤檔包含資料。請參考日誌檔以取得詳細資訊。在 OS/390 上，亦請參閱 JESLogFile。

日誌檔

當「Trillium 批次系統」使用者定義程式執行時，「資料倉儲中心」將所有診斷資訊儲存在日誌檔中。該日誌檔名稱爲 `tbsudp-date-time.log`，其中 `date` 是建立檔案的系統日期，而 `time` 是建立檔案的系統時間。該日誌檔建立於代理程式端上由環境變數 `VWS_LOGGING` 所指定的目錄中。若「Trillium 批次系統」使用者定義程式順利執行，則會刪除該日誌檔。

MQ Series 與「資料倉儲中心」的整合

現在，「資料倉儲中心」可讓您存取 MQSeries 訊息佇列的資料，作為一個 DB2 資料庫概略表。提供精靈來建立 DB2 表格函數及供您在其中存取資料的 DB2 概略表。每一個 MQSeries 訊息被視為一個有定界符號的字串，其根據您的規格來剖析並傳回為一個結果列。此外，屬於 XML 文件的 MQSeries 訊息可存取為倉儲來源。使用「資料倉儲中心」，您可以從 MQSeries 訊息佇列及「DB2 XML 擴充元文件存取定義 (DAD)」檔匯入描述資料。

建立 MQSeries 訊息的概略表

基本要求

DB2 Universal Database 版本 7.2.

DB2 Warehouse Manager 版本 7.2

MQSeries 支援。有關 MQSeries 需求的詳細資訊，請參閱第173頁的『MQSeries』。有關設定倉儲來源的資訊，請參閱使用者定義的函數之安裝區段。

限制

- 當編目倉儲來源資料庫時，會在代理程式機器上編目資料庫別名。不過，當建立 MQSeries 和 XML 概略表時，「資料倉儲中心」假設資料庫別名也定義於從屬站機器上，並將嘗試使用倉儲來源資料庫使用者 ID 和通行碼來連接它。若順利完成，則會呼叫精靈供您建立概略表。若失敗的話，則會顯示一則警告訊息，而您必須在精靈中編目或選擇另一個資料庫別名。
- 請參閱「版本注意事項」的 SQL Reference 區段，以取得 MQ 訊息的最大長度。

建立 MQSeries 訊息的概略表

建立 MQSeries 訊息的概略表：

1. 從「資料倉儲中心」視窗中，展開倉儲來源樹狀結構。
2. 展開包含概略表的倉儲來源。
3. 以滑鼠右鍵按一下概略表資料夾，然後按一下對 MQSeries 訊息建立...

即會開啓 MQSeries 精靈。完成此精靈後，會在「資料倉儲中心」中建立一個新概略表。當選定概略表後，便會存取 MQSeries 佇列，並且在精靈中根據您的規格，將每一個訊息剖析為有定界符號的字串。

匯入 MQSeries 訊息和 XML 描述資料

基本要求

DB2 Universal Database 版本 7.2.

DB2 XML Extender 版本 7.2.

MQSeries 支援。有關 MQSeries 需求的詳細資訊，請參閱第173頁的『MQSeries』。有關設定倉儲來源的資訊，請參閱使用者定義的函數之安裝區段。

限制

若目標表格內存在主要或外來鍵，則匯入會失敗。在匯入之前，您必須手動刪除「資料倉儲」中這些鍵值的定義。

匯入 MQSeries 訊息與 XML 描述資料

若要匯入 MQSeries 描述資料到「資料倉儲中心」：

1. 準備倉儲目標資料庫：
 - 您必須定義倉儲目標端以及登記和啓用轉換程式。
 - 您必須對 DB2 XML Extender 啓用倉儲目標端。詳細資訊，請參閱 DB2 XML Extender 版本 7.2 版本注意事項。
 - 建立 XML Extender 資料存取定義 (DAD) 檔，指示「資料倉儲中心」如何將 XML 文件內容對映到倉儲表格。使用 DAD 檔對資料庫啓用 XML 集合。詳細資訊，請參閱 DB2 XML Extender 版本 7.2 版本注意事項。
2. 以滑鼠右鍵按一下**倉儲**，然後按一下**匯入描述資料 -> MQSeries**，開啓**匯入描述資料**視窗。
3. 在 **AMI 服務程式**欄位中，鍵入傳送或擷取訊息所在的服務程式點。
4. 在 **AMI 原則**欄位中，鍵入傳訊系統用來執行作業的原則。
5. 在 **DAD 檔案**欄位中，鍵入 DB2 XML Extender DAD 檔的名稱，或按一下省略符號 (...) 來搜尋檔案以選取。此檔案必須是本端檔案。
6. 在**倉儲目標端**欄位中，從組合框清單中選取執行步驟所在的倉儲目標端名稱。倉儲目標端必須已完成定義。
7. 在**綱目**欄位中，鍵入用來定義 DAD 檔 (不含 qualifier) 中表格名稱的綱目名稱。預設綱目採用您先前所選取倉儲目標端的登入使用者 ID。
8. 選擇**目標選項**：
 - 若要此步驟在執行期取代目標表格內容，請按一下**取代表格內容**圓鈕。
 - 若要此步驟在執行期添加到目標表格內容，請按一下**添加表格內容**圓鈕。
9. 按一下**確定**。
 - 將關閉「匯入描述資料」視窗。

匯入作業完成後，下列倉儲物件會新增至「倉儲」樹狀結構中。

- 名為 MQSeries 和 XML 的主旨區域。
- 名為 MQSeries 和 XML 的處理程序。
- 名為 MQSeries 和 XML 的使用者定義程式群組。
- 說明於 DAD 檔中所有倉儲目標端表格的定義。
- <ServiceName>.<DAD file base name>.<Warehouse target Name> step.

- <ServiceName>.<DAD file base name> program template.

若倉儲目標代理程式端與本端機器不同，您必須變更步驟參數：

1. 以滑鼠右鍵按一下步驟，然後選取**內容**。按一下內容筆記本中的**參數**標籤。
2. 將 DAD 檔參數的名稱變更爲遠端倉儲目標代理程式端上相同 DAD 檔的名稱。
3. 請確定「處理程序選項」標籤中的「代理程式端」包含預期的代理程式端。

使用 MQSeries 使用者定義程式

MQSeries 和 XML 儲存程序稱爲 **MQXMLXF**，它隨附於 Windows NT 和 UNIX 的「DB2 資料倉儲中心」版本 7.2 中。您匯入 MQSeries 和 XML 描述資料時所建立的步驟會執行此儲存程序。下表描述其參數：

參數	值
MQSeries ServiceName	訊息傳送至或從其中擷取出來的服務程式點名稱。
MQSeries PolicyName	傳訊系統用來執行作業的原則名稱。
DAD 檔名	DB2 XML Extender DAD 檔的名稱
TargetTableList	以逗點區隔的步驟之目標表格清單
選項	REPLACE 或 APPEND
RUN ID	步驟版本號碼 (適用記載)
註： 此表格中所有參數的資料類型爲 CHARACTER。	

若**選項**具有值 REPLACE，則儲存程序會刪除目標表格中所有列。該儲存程序也會呼叫 DB2 XML Extender 儲存程序，針對所有現存的 MQSeries 訊息在目標表格中輸入資料。

錯誤回覆碼

當執行此步驟時，儲存程序會傳回錯誤碼 SQLCODE -443 和 SQLSTATE 38600。若要診斷錯誤，請參閱下表中可能的診斷文字。

錯誤碼	說明
AMIRC=xxxx;<日誌檔名稱>	xxxx 是來自 AMI 層的回覆碼。詳細資訊，請參閱 MQSeries 文件。<日誌檔名稱> 指示日誌檔的位置。
XMLRC=xxxx;<日誌檔名稱>	xxxx 是來自 DB2 XML Extender 的回覆碼。有關回覆碼的說明，請參閱 DB2 XML Extender 文件。 <日誌檔名稱> 指示日誌檔的位置。
SQLCODE=xxxx;<日誌檔名稱>	xxxx 是執行 SQL 要求時傳回的非零 SQLCODE。 <日誌檔名稱> 指示日誌檔的位置。

若是所有錯誤，請參閱日誌檔以取得詳細資訊。

錯誤日誌檔

當 MQXMLXF 執行時，「資料倉儲中心」將所有診斷資訊儲存在日誌檔中。日誌檔名稱爲 mqxf<nnnnnnnn>.log，其中 <nnnnnnnn> 是傳遞至儲存程序的 RunID。「資料倉儲中心」會在 VWS_LOGGING 環境變數所指示的目錄中建立該檔案。若未定義此環境變數，則會在暫時目錄中建立日誌檔。

若要使 VWS_LOGGING 環境變數出現在 Unix 平台的儲存程序中，您應該在 **db2start** 命令之前，使用 **db2set** 命令來將 VWS_LOGGING 新增至 DB2ENVLIST 環境變數中。下圖為環境命令範例。

```
db2set DB2ENVLIST="AMT_DATA_PATH VWS_LOGGING"
```

圖 6. 環境變數命令範例

若步驟順利執行，將刪除日誌檔。

Microsoft OLE DB 與「資料異動服務」支援

現在，「資料倉儲中心」可讓您存取 OLE DB 提供者的資料，作為一個 DB2 資料庫概略表。您可以使用「資料倉儲中心」提供的 OLE DB 輔助精靈，建立 DB2 OLE DB 表格函數及供您在其中存取資料的 DB2 概略表。

Microsoft Data Transformation Services (DTS) 可讓您在 OLE DB 來源和目標之間輸入、匯出及轉換資料，以建置資料倉儲和資料中心。DTS 隨 Microsoft SQL Server 安裝。所有 DTS 作業儲存在 DTS 資料包中，您可以使用 Microsoft OLE DB Provider for DTS Packages 來執行及存取 DTS 資料包。因為您可以存取 DTS 資料包作為 OLE DB 來源，所以也可以透過 OLE DB 輔助精靈，以對 OLE DB 資料來源所用的相同方式，對 DTS 資料包建立概略表。當您在執行期存取概略表時，會執行 DTS 資料包，而且 DTS 資料包中作業的目標表格變成建立的概略表。

在「資料倉儲中心」中建立概略表之後，您可以像使用其它任何概略表一樣地使用它。例如，您可以在 SQL 步驟中將 DB2 表格與 OLE DB 來源結合在一起。當您使用 SQL 步驟中建立的概略表時，會呼叫 DTS 提供者並執行 DTS 資料包。

軟體需求：

- DB2 Universal Database for Windows NT 版本 7.2，如倉儲目標端資料庫
- DB2 Warehouse Manager 版本 7.2
- 若倉儲目標端資料庫已在版本 7.2 之前建立，您必須在安裝 DB2 UDB for Windows NT 版本 7.2 之後，執行 db2upd7 命令
- 當編目倉儲來源資料庫時，會在倉儲代理程式端上編目資料庫別名。不過，當啟動精靈時，「資料倉儲中心」會假設資料庫別名也定義在從屬工作站上，並將嘗試使用倉儲來源資料庫使用者 ID 和通行碼來連接它。若連接成功，則精靈會啟動供您建立概略表。若連接失敗，則會顯示一則警告訊息，而您必須在精靈中編目或選擇另一個資料庫別名。
- 若要識別 DTS 資料包中特定的表格，您必須在建立目標表格的 DataPumpTask 之「工作流程內容」視窗的「選項」標籤中，選取 **DSO 列集提供者** 勾選框。若開啓多重 DSO 列集提供者屬性，則僅會使用第一個選取的步驟結果。當選取某概略表後，會傳回其目標表格的列集，但忽略您在後續步驟中建立的其它所有列集。
- 當對精靈輸入表格名稱時，請使用步驟名稱，其顯示於該作業的「工作流程內容」筆記本之「選項」頁面中。
- DTS 資料包連接字串與 dtsrun 命令的語法相同。

建立 OLE DB 表格函數的概略表

建立 OLE DB 表格函數的概略表：

1. 從「資料倉儲中心」視窗中，展開倉儲來源樹狀結構。
2. 展開包含概略表的倉儲來源。
3. 以滑鼠右鍵按一下**概略表**資料夾，然後按一下對 **OLE DB 表格函數**建立。

即會開啓 OLE DB 輔助精靈。此精靈會逐步引導您完成在倉儲來源資料庫中建立新概略表的作業。

建立 DTS 資料包的概略表

若要為 DTS 資料包建立概略表：

1. 從「資料倉儲中心」視窗中，展開倉儲來源樹狀結構。
2. 展開包含概略表的倉儲來源。
3. 以滑鼠右鍵按一下**概略表**資料夾，然後按一下 **Microsoft OLE DB Provider for DTS Packages**。

即會開啓 OLE DB 輔助精靈。此精靈會逐步引導您完成在倉儲來源資料庫中建立新概略表的作業。

有關 DTS 的詳細資訊，請參閱 Microsoft Platform SDK 2000 文件，其內容為如何建置連接 DTS 提供者時精靈需要的提供者字串之詳細說明。

增量 COMMIT 搭配 replace 使用

在母體類型為 Replace 的步驟中，唯有插入新資料時，才會使用增量確定。在單一確定範圍內，原始資料會被刪除。若您需要刪除資料而不產生日誌記錄，請在執行具有附加母體類型的 SQL 步驟之前，執行載入空白檔案的步驟。

元件追蹤資料檔名

「資料倉儲中心」在 Windows NT 上撰寫這些檔案：

AGNTnnnn.Log 包含追蹤資訊。*nnnn* 為倉儲代理程式的數字程序 ID，根據作業系統，它可以是 4 或 5 位數。

AGNTnnnn.Set 包含代理程式的環境設定。*nnnn* 為倉儲代理程式的數字程序 ID，根據作業系統，它可以是 4 或 5 位數。

預設目錄為 *x:\program files\sql\logging*，其中 *x* 是安裝 DB2 所在的磁碟機。

AIX 及 Solaris 作業環境上 Sybase 來源需要的 OPEN CLIENT

在第 3 章「設置倉儲來源」，在 AIX 方面的表格 3「AIX 上支援的資料來源的連接需求」以及 Solaris 作業環境方面的表格 4「Solaris 作業環境上支援的資料來源的連接需求」中，「如何連接：」直欄中的 Sybase 登錄應該包含另一個額外的步驟。其它步驟顯示在以下的步驟 3 中。

3. 安裝 OPEN CLIENT

請注意：在 Windows NT 或 Windows 2000 平台上連接 Sybase 來源，需要 OPEN CLIENT。

範例登錄更正

在 *Data Warehouse Center 管理手冊* 一書的「第 3 章 設定倉儲來源」中關於 Driver 屬性的路徑並不正確。下列路徑才是正確的：

圖 6

Driver=/home/db2_07_01/3.6/odbc/lib/ivinf12.so

圖 8

Driver=/home/db2_07_01/3.6/odbc/lib/ivsyb1112.so

圖 10

Driver=/home/db2_07_01/3.6/lib/ivor814.so

圖 11

Driver=/home/db2_07_01/3.6/odbc/lib/ivmsss14.so

第 3 章 設定倉儲來源

對映 Microsoft Access 中的備忘錄欄位至倉儲來源

Microsoft Access 資料庫的備忘錄欄位在資料倉儲中心中是以超過 1GB 直欄大小的 LONG VARCHAR 資料類型來表示。若要支援實際的系統架構，資料倉儲中心會截斷超過 128 KB 的值。若要避免在倉儲來源中截斷備忘錄欄位值，請您在步驟中使用表格之前，先將接收備忘錄欄位資料的直欄資料類型從 LONG VARCHAR 變更為 CLOB。如果您沒有變更直欄的資料類型，任何大於 128 KB 的值都會被截斷。

OS/390 及 OS/400 需要支援 CLOB 資料類型的 DRDA。OS/390 從 DB2 版本 6 開始支援 CLOB 資料類型。OS/400 從版本 4、具有 DB FixPak 4 的版次 4 或更新版本 (PTF SF99104) 開始支援 CLOB 資料類型。若是 OS/400，安裝磁碟版本 4，1999 年 2 月的版次 4，也支援 CLOB 資料類型。

第 10 章 維護倉儲資料庫

鏈結表格至 DB2 UDB RUNSTATS 程式的步驟次類型

RUNSTATS 程式的步驟次類型自倉儲目標讀取並寫入倉儲目標。在定義此步驟的值之前，請先將目標鏈結到「程序模型」視窗的步驟次類型。

預設倉儲控制資料庫

在 Windows NT 或 Windows 2000 安裝 DB2 的期間，如果在 Windows NT 系統登錄中沒有識別作用中的倉儲控制資料庫，DB2 會在資料倉儲中心建立和起始設定預設倉儲控制資料庫。利用起始設定程序，資料倉儲中心建立控制表格，以儲存資料倉儲中心描述資料。

預設倉儲控制資料庫的名稱為 DWCTRLDB。登入時，資料倉儲中心指定 DWCTRLDB 為倉儲控制資料庫預設值。欲查看倉儲控制資料庫的名稱，按一下資料倉儲中心「登入」視窗的進階按鈕。

倉儲控制資料庫管理視窗

倉儲控制資料庫管理視窗在 Windows NT 或 Windows 2000 安裝典型 DB2 期間進行安裝。利用此視窗變更作用中的倉儲控制資料庫、建立和起始設定新建倉儲控制資料庫、移轉 IBM Visual 倉儲所使用的倉儲控制資料庫。下列討論上述的活動。

使用倉儲控制資料庫管理視窗前停止倉儲伺服器。

變更作用中倉儲控制資料庫

如果您要使用作用中的倉儲控制資料庫以外的倉儲控制資料庫，請使用倉儲控制資料庫管理視窗將該資料庫登記為作用中的控制資料庫。登入資料倉儲中心時，指定作用中倉儲控制資料庫以外的名稱，您會得到錯誤的訊息，說明您所指定的資料庫和倉儲伺服器所指定的資料庫不相符合。

登記資料庫：

1. 按一下**開始 --> 程式集 --> IBM DB2 --> 倉儲控制資料庫**
2. 在**新建控制資料庫**欄位中，輸入您要使用的控制資料庫之名稱。
3. 在**綱目**欄位中，輸入資料庫需要的綱目名稱。
4. 在**使用者 ID** 欄位中，輸入存取資料庫所需要的使用者 ID 名稱。
5. 在**通行碼**欄位中，輸入使用者 ID 的通行碼。
6. 在**驗證密碼**欄位中，再一次輸入通行碼。
7. 按一下**確定**。
視窗保持開啓。**訊息**欄位顯示登錄程序的狀態訊息。
8. 程序完成後，關閉視窗。

建立和起始設定倉儲控制資料庫

如果您要建立預設值以外的倉儲控制資料庫，請在安裝程序期間建立，或是在倉儲控制資料庫管理視窗進行安裝之後建立。使用安裝程序在與倉儲伺服器相同和不同的工作站上建立資料庫。

欲變更安裝期間所建立倉儲控制資料庫名稱，請執行自訂安裝，並變更定義本端倉儲控制資料庫視窗內的名稱。安裝程序以您所指定的名稱建立資料庫，起始設定資料倉儲中心所使用的資料庫，登記資料庫為作用中倉儲控制資料庫。

欲在已安裝倉儲伺服器以外的工作站建立倉儲控制資料庫，請在自訂安裝期間，選取倉儲本端控制資料庫。安裝程序將建立該資料庫。安裝完成後，請按照『變更作用中倉儲控制資料庫』內的步驟，在倉儲伺服器工作站使用倉儲控制資料庫管理視窗。安裝期間，請指定資料庫名稱。起始設定資料倉儲中心所使用的資料庫，登記該資料庫為作用中倉儲控制資料庫。

欲在安裝程序後建立和起始設定倉儲控制資料庫，請於倉儲伺服器工作站使用倉儲控制資料庫管理視窗。如果新建的倉儲控制資料庫不在倉儲伺服器工作站，則必須先在

倉儲伺服器工作站建立該資料庫並將它編入目錄。遵循 第229頁的『變更作用中倉儲控制資料庫』內的步驟。安裝期間，請指定資料庫名稱。

登入資料倉儲中心時，按一下**進階**按鈕，並輸入作用中倉儲控制資料庫名稱。

建立版本的 SQL 步驟

在建立版本的 SQL 步驟時，基於使用情況，您可能會考慮在版本直欄建立非唯一的索引，以增進刪除版本的效能。對於大量倉儲表格請考慮到這一點，因為插入少量的列數會影響到插入的效能。

在「程序模型」視窗中，變更來源或目標

在「程序模型」視窗中，如果變更了來源或目標，將會自動儲存所做的變更。如果您做了任何其它的變更(比如新增步驟)，您必須明確地儲存變更，以達到永久變更的目的。若要儲存變更，請按一下“程序 → 儲存”。

新增說明至「資料倉儲中心」物件

在資料倉儲中心中，於筆記本的說明欄位，您可以指定多達 254 個字元。此最大值取代了線上說明中所指定的最大長度。

執行「範例內容」

- 您無法成功地執行在純文字檔來源使用 AS/400 代理程式的「範例內容」要求。雖然您可以建立純文字檔來源，並企圖使用 AS/400 代理程式發出範例內容要求，但是這項要求會失敗。
- 在對程序模型產生器中的倉儲目標執行「範例內容」時，可能會收到錯誤訊息。這個錯誤與倉儲來源、倉儲目標、程序步驟的代理程式端之可用性有關。一個步驟可用的代理程式端之列表，是倉儲來源 IR 代理程式端、倉儲目標 IR 代理程式端、此特別步驟可用的代理程式端交集後所得到的結果 (所指的步驟為代理程式端內容筆記本的最後一頁所選的步驟)。例如，檢視執行 FTP Put 程式 (VWPRCPY) 的程序的「範例內容」。代理程式端定義的代理程式端選取程序中所使用的步驟。當您對目標檔執行「範例內容」，通常使用選取列表中的第一個代理程式端。但是，資料庫維護作業會影響代理程式端列表的次序。如果選取的代理程式端與來源或目標檔的系統不同，範例內容將失敗。

編輯 Create DDL SQL 陳述式

於開發模式下，編輯目標表格的 Create DDL SQL 陳述式時，會看到下列令人迷惑的訊息：『Create DDL SQL 陳述式的任何變更將不反映在表格定義或實際的表格上。要繼續嗎？』

變更會反映在實際實體的表格上。忽略該訊息，繼續變更 Create DDL 陳述式。

在開發模式步驟中，此訊息的修正版應如下所示：『Create DDL SQL 陳述式的任何變更將不反映在表格定義中。要繼續嗎？』

對於測試或生產模式中的步驟，此訊息是正確的。將步驟升級到測試模式時，會建立實體的目標表格。資料倉儲中心不會變更該表格。

移轉 Visual Warehouse 商業檢視

如果要將 Visual Warehouse 描述資料同步化之商業檢視移轉至資料倉儲中心時，您可在移轉倉儲控制資料庫之前將商業檢視提昇至「生產」狀態。如果商業檢視在「生產」狀態，它的排程就會移轉至資料倉儲中心。如果商業檢視不在「生產」狀態，它就會移轉至「測試」狀態，而沒有排程。不可將移轉步驟提昇至「生產」狀態。您必須先在資料倉儲中心再次建立同步化步驟，然後刪除移轉步驟。

建立目標表格及主要鍵

資料倉儲中心建立此步驟的目標表格後，並沒有產生此目標表格的主要鍵。某些轉換程式，如 Moving Average，會使用所產生的表格來做為來源表格，其來源表格也須要主要鍵。在您使用以轉換程式產生的表格之前，請先在 DB2 控制中心的表格上按滑鼠右鍵來定義此表格的主要鍵，然後按一下變更。

使用 Merant ODBC 驅動程式

如果要存取 Windows NT 上的 Microsoft SQL Server，請使用 Merant ODBC 驅動程式來確認其系統路徑是否包含 sqllib\odbc32 目錄。

新建 ODBC 驅動程式

如果您將使用已鏈結至存取 Merant ODBC 來源的資料倉儲中心 AIX 或 Sun 代理程式，並且將存取 DB2 資料庫，請變更 .odbc.ini 檔案中 DB2 來源區段的「Driver=」屬性值，如下所示：

AIX：驅動程式名稱爲 /usr/lpp/db2_07_01/lib/db2_36.o

AIX 的範例 ODBC 來源登錄：

```
[SAMPLE] Driver=/usr/lpp/db2_07_01/lib/db2_36.o
Description=DB2 ODBC Database
Database=SAMPLE
```

Sun：驅動程式名稱爲 /opt/IBMDB2/V7.1/lib/libdb2_36.so

Sun 的範例 ODBC 來源登錄：

```
[SAMPLE] Driver=/opt/IBMDB2/V7.1/lib/libdb2_36.so
Description=DB2 ODBC Database
Database=SAMPLE
```

定義 OS/2 資料庫中的倉儲來源或目標

定義 OS/2 資料庫的倉儲來源或倉儲目標後，請用大寫字母鍵入資料庫名稱。

監督倉儲控制資料庫的狀態

DB2 控制中心或命令行處理器可能會指出倉儲控制資料庫是否有不一致的狀態。此狀態指出倉儲伺服器沒有發出它起始的啟動到倉儲日誌器。

使用 SQL 輔助程式與 TBC_MD 範例資料庫

在 TBC_MD 資料庫中所含的資料倉儲範例中，您不可以使用 SQL 輔助程式來變更 Select Scenario SQL 步驟中的 SQL，因為 SQL 是在 SQL 輔助程式產生之後編輯的。

使用 FormatDate 函數

若要使用 FormatDate 函數，請在「內容」筆記本的「SQL 陳述式」頁上，按一下**建置 SQL**，來進行 SQL 步驟。

FormatDate 函數的輸出是資料類型 varchar(255)。您不可從 Function Parameters-FormatData 頁面中的**種類**列示，選取日期、時間、或日期/時間來變更資料。

變更語言設定

在 AIX 和 Solaris 作業環境中，安裝程序會為資訊型錄設定要公佈的語言，並匯出到 OLAP 整合伺服器。如果您想在安裝時使用這些語言函數，請輸入下列指令 (一整行) 以便建立軟鏈結。

在 AIX

```
/usr/bin/ln -sf /usr/lpp/db2_07_01/msg/locale/flgnxolv.str  
/usr/lpp/db2_07_01/bin/flgnxolv.str
```

語言環境

語言的語言環境 (locale) 名稱的格式為 xx_yy

在 Solaris 作業環境

```
/usr/bin/ln -sf /opt/IBMDB2/V7.1/msg/locale/flgnxolv.str  
/opt/IBMDB2/V7.1/bin/flgnxolv.str
```

語言環境

語言的語言環境 (locale) 名稱的格式為 xx_yy

使用「產生鍵值表格」轉換程式

當您使用「產生鍵值表格」轉換程式的**更新鍵值直欄的值**選項時，轉換程式僅更新表格中沒有值的列。(也就是，值為 NULL)。當您在表格中插入其它橫列時，除非您重新執行轉換程式，否則鍵值為空值。

為了避免此問題，請執行下列動作：

- 在起始執行轉換程式之後，請使用**取代全部值**選項以建立所有橫列的值。

維護資料庫連接

當 DB2 伺服器管理的資料庫已停止或重新啟動時，倉儲伺服器並不會維護本端或遠端資料庫的連接。若您要停止或重新啟動 DB2，請一併停止或重新啟動倉儲伺服器。

設定遠端「資料倉儲中心」從屬站

當您在具有倉儲伺服器的其它工作站上，安裝 DB2 管理從屬站與資料倉儲工具來設定資料倉儲中心管理從屬站時，您必須新增倉儲中心所列的 TCP/IP 埠號到服務程式檔案給從屬工作站。請在服務程式檔案新增下列項目：

```
vwkernel      11000/tcp
```

定義 DB2 for VM 倉儲來源

當您為 DB2 for VM 資料庫 (從 DRDA 闢道存取) 定義倉儲來源時，CLOB 與 BLOB 資料類型有使用限制：

- 您不能使用範例內容函數來檢視 CLOB 與 BLOB 資料的資料類型。
- 您不能使用具 SQL 步驟的 CLOB 與 BLOB 資料類型的直欄。

此限制是 DB2 for VM 版本 5.2 伺服器的已知限制，其無法使用 DRDA 傳輸 LOB 物件至 DB2 版本 7 從屬站。

定義 DB2 for VM 或 DB2 for VSE 目標表格

當您在資料倉儲中心定義 DB2 for VM 或 DB2 for VSE 目標表格時，請勿選取**授與公用權限**勾選框。DB2 for VM 與 DB2 for VSE 並未支援資料倉儲中心產生的 GRANT 指令語法。

啓用有定界符號的識別字支援

啓用可支援 Sybase 與 Microsoft Windows NT 的 SQL Server 有定界符號的識別字時：請選取 ODBC Driver Setup 筆記本進階頁面的 **Enable Quoted Identifiers** 勾選。

啓用可支援 UNIX 的 Sybase 的有定界符號的識別字時，請編輯 .odbc.ini 檔中的 Sybase 資料來源包括連接屬性 EQI=1。

Data Joiner 錯誤指出連結問題

使用 DataJoiner 與 DB2 7.1 版 + FixPak 2 或更新版本的客戶可能會得到一個錯誤訊息，指出連結有問題。

例如，在使用 DataJoiner 來源與資料倉儲中心 7.2 版代理程式時，您可能會發生鏈結錯誤：

```
DWC07356E 針對步驟 "?" 的版本 "0"，類型 importTableNames 的指令之代理程式處理程序失敗。  
SQL0001N 連結或前置編譯未能順利完成。  
SQL0001N 找不到 NULLID.SQLL6D05 套裝軟體。SQLSTATE=51002 RC = 7356 RC2 = 8600
```

若要更正該問題，請新增下列行至 db2cli.ini 檔案：

```
[COMMON]  
DYNAMIC=1
```

在 UNIX 系統上，db2cli.ini 檔案是位在 .../sqllib/cfg 目錄。在 Windows NT 上，db2cli.ini 檔案是位在 .../sqllib 目錄。

資料倉儲中心設定和執行抄寫

1. 進行資料倉儲中心設定和執行抄寫，抄寫控制表格必須在倉儲控制資料庫和倉儲目標資料庫內。

抄寫需要抄寫控制表格存在於控制和目標資料庫內。在 ASN 綱目中可以找到抄寫控制表格，它們開始於 IBMSNAP。如果控制表格並不存在，當您透過控制中心來定義抄寫來源時，抄寫控制表格會在資料庫內自動建立。備註：控制表格也一定存在於目標 DB 中。欲得到在目標 DB 所建立控制表格，請使用控制中心建立抄寫來源，然後再移除抄寫來源，只留下控制表格。或者，使用 DJRA (Data Joiner Replication Administration) 產品來定義控制表格。

2. 安裝和使用 DJRA

欲使用 DJRA 來定義控制表格，則必須先進行安裝。DJRA 傳送為 DB2 的一部份。欲安裝 DJRA，請跳至 d:\sqlib\djra 目錄，並按一下 djra.exe 套裝軟體。將 DJRA 安裝到您的系統。之後，欲在 Windows NT 的開始功能表存取 DJRA，請按一下 Windows NT 選擇的 DB2，選取抄寫，選取抄寫管理工具。DJRA 介面和一般的 NT 應用程式不太相同。所執行的每一個函數，它建立一組要執行的 SQL，但是卻沒有去執行。使用者必須自行儲存所產生的 SQL，然後，選取執行 SQL 函數進行執行。

3. 設定執行擷取程式和引用程式

正在測試中的系統，有關如何架構系統來執行擷取程式和引用程式的指令，請參閱 Replication 指南和參考手冊。連結擷取和引用程式中將被使用的各個資料庫。注意，您不需要建立通行碼檔案。資料倉儲中心會為定期抄寫方式自動建立通行碼檔案。

4. 定義控制中心內抄寫來源

使用控制中心定義抄寫來源。資料倉儲中心支援五種抄寫型類：使用者副本，時間點，基本聚集，變更聚集和暫置表格 (CCD 表格)。使用者副本、時間點和壓縮暫置表格的型類，需要抄寫來源表格有一個主要鍵。其它抄寫型類則不需要。選擇被定義為抄寫來源的輸入表格時，請記得這一點。實際上，抄寫來源是原始來源表格和建立之 CD 表格 (變更資料) 的定義，資料變更移動至目標表格前保留在 CD 表格內。在控制中心內定義抄寫來源，ASN.IBMSNAP_REGISTER 會寫入一項記錄，以定義該來源和它的 CD 表格。CD 表格在同一時間內建立，但是，開始時它裡頭沒有資料。定義抄寫來源，您可以選擇只併入後像直欄，或選擇併入前像和後像直欄。透過控制中心抄寫來源介面的勾選框來選項。您所選擇的前像和後像直欄會轉換至新建 CD 表格內所建立的直欄。CD 表格內，後像直欄和原始來源表格直欄名稱同名。後像直欄名稱的第一個字元為 X。

5. 匯入抄寫來源至資料倉儲中心

一旦在控制中心建立抄寫來源，您可以將它匯入資料倉儲中心。匯入來源時，請確定按一下「可以抄寫的表格」勾選框。這告訴了資料倉儲中心看看 ASN.IBMSNAP_REGISTER 表格內的記錄，有那些表格被定義為抄寫來源。

6. 在資料倉儲中心內定義抄寫步驟

在程序模型產生器中，選取五種抄寫型類中的一種：基本聚集、變更聚集、時間點、暫置表格、使用者副本。欲定義基本聚集或變更聚集抄寫類型，請參閱以下有關如何在資料倉儲中心安裝基本聚集或變更聚集抄寫。為抄寫類型選取適當的抄寫來源。如上所說，使用者副本、時間點和壓縮暫置表格的抄寫型類需要輸入

來源有一個主要鍵。連接抄寫來源至抄寫步驟。開啓抄寫步驟的內容。跳至參數標籤。選取所要的直欄。選取勾選框來建立目標表格。選取倉儲目標。跳至處理程序選項並填寫參數。按「確定」。

7. 啓動擷取程式

DOS 視窗內，請輸入：ASNCCP source-database COLD PRUNE

COLD 參數指示 COLD 的啓動，並刪除存在於 CD 表格的任何資料。PRUNE 參數指示擷取程式維護 IBMSNAP_PRUNCNTL 表格。讓擷取程式繼續執行。要退出時，請利用 DOS 視窗內的 Ctrl-Break 下令停止。注意，啓動引用程式前請啓動擷取程式。

8. 抄寫步驟升級到測試 (Promote-To-Test)

回到資料倉儲中心，對於定義的抄寫步驟，將步驟升級到「測試」模式。這些會造成定期抄寫方式資訊寫入抄寫控制表格。您將看到新增至

IBMSNAP_SUBS_SET、IBMSNAP_SUBS_MEMBR、IBMSNAP_SUBS_COLS、IBMSNAP_SUBS_EVENT的記錄，以支援定期抄寫。目標表格也會在目標資料庫中建立。若抄寫類型為使用者副本、時間點、壓縮暫置表格，目標表格需要一個主要鍵。跳至控制中心來建立主要鍵。注意，在不同的直欄裡，有些抄寫目標表格也需要唯一索引。字碼存在於資料倉儲中心中，在表格建立時，它會建立這些唯一索引，所以，您不需要自己建立這些索引。注意，如果在控制中心內定義主要鍵，同時，直欄中已有一個唯一索引，當您在建立主要鍵時回傳回警告訊息。忽略此警告訊息。

9. 抄寫步驟升級到生產 (Promote-To-Production)

在升級到生產期間，不會做任何定期抄寫方式變更。這完全和其它步驟的「資料倉儲中心」作業相同。

10. 執行抄寫步驟

抄寫步驟被提昇為測試模式後就可以執行。來源表格進行任何變更前，請先執行一次。跳至「工作進行中」(Work-in-Progress;WIP) 章節，選取抄寫步驟。執行它。步驟執行時，IBMSNAP_SUBS_EVENT 表格內的事件記錄會更新，IBMSNAP_SUBS_SET 的定期抄寫記錄為後置作用中。定期抄寫馬上執行。定期抄寫執行時，代理程式呼叫引用程式處理作用中的定期抄寫。之後，更新原始來源表格，所變更的資料將移至 CD 表格。如果執行下列抄寫步驟，引用程式再一次執行，變更資料將從 CD 表格移動到目標表格。

11. 抄寫步驟降級到測試 (Demote-To-Test)

在降級到測試期間，不會做任何定期抄寫方式變更。這完全和其它步驟的「資料倉儲中心」作業相同。

12. 抄寫步驟降級到開發 (Demote-to-Development)

當您將抄寫步驟降級到開發時，定期抄寫資訊會自抄寫控制表格內移除。降級到生產完成後，特定定期抄寫不會在抄寫控制表格內留下記錄。目標表格也會在此被除去。CD 表格會適當地保留下來，因為它屬於抄寫來源的定義。

13. 如何在資料倉儲中心設定基本聚集或變更聚集。

- 輸入表格。選擇可以被 GROUP BY 陳述式使用的輸入表格。例如，使用含有這些直欄的輸入表格：SALES，REGION，DISTRICT。
- 抄寫步驟。選擇基本或變更聚集。開啓步驟內容。
 - 在引用程式執行時，它必須執行類似 SELECT SUM(SALES), REGION, DISTRICT GROUP BY REGION, DISTRICT 的 SELECT 陳述式。因此，在

選取的輸出直欄中，您必須選擇 REGION、DISTRICT 及一個計算直行 SUM(SALES)。使用新增計算直行按鈕。範例中請輸入 SUM(SALES) 於「表示式」欄位。儲存它。

- WHERE 子句。有一項抄寫需求：在您設定只需 GROUP BY 子句的抄寫步驟時，必須提供 DUMMY WHERE 子句 (比如 1=1)。不要將 "WHERE" 這個字加到 WHERE 子句中。因此，基本聚集的資料倉儲中心 GUI 內，只有一個 WHERE 子句輸入欄位。在此欄位中，例如：輸入： 1=1 GROUP BY REGION, DISTRICT 到「變更聚集」，有「WHERE 子句」和「GROUP BY」兩個輸入欄位：「WHERE 子句」欄位輸入： 1=1，「GROUP BY」欄位輸入： GROUP BY REGION, DISTRICT
- 設定剩下的步驟內容，就如您設定其它抄寫類型一般。按「確定」以儲存步驟，並建立目標表格物件。
 - 開啓目標表格物件。更改計算直行表示式的輸出直欄名稱成爲有效的直欄名稱，並且指定直欄的有效的資料類型。儲存目標表格物件。
 - 在抄寫步驟執行提昇測試 (Promote-to-Test)。建立目標表格。不需要主要鍵。
 - 執行步驟如其它抄寫步驟般。

疑難排解秘訣

- 開啓引用程式的追蹤，在倉儲內容畫面設定代理程式值 = 4。代理程式開啓對引用程式完整的追蹤，當代理程式追蹤 = 4 時。
如果在 CD 表格中沒有任何資料，則很可能是擷取程式沒有啓動，或建立變更資料的原始來源表格沒有更新。
- 排程筆記本之通知頁面的郵件伺服器欄位自線上說明中遺漏了。
- 郵件伺服器需要支援 ESMTP，資料倉儲中心通知才能工作。在「開啓工作進行中」視窗的說明中，按一下 "倉儲 → 工作進行中"，而不是 "倉儲中心 → 工作進行中"。

存取來源和目標檔

下列表格列示，資料倉儲中心所支援的來源和目標檔之版本及版次。

表 7. IBM 倉儲來源所支援的版本和版次

來源	版本/版次
IMS	5.1
DB2 Universal Database for Windows NT	5.2 - 7.1
DB2 Universal Database Enterprise-Extended Edition	5.2 - 7.1
DB2 Universal Database for OS/2	5.2 - 7.1
DB2 Universal Database for AS/400	3.7 - 4.5
DB2 Universal Database for AIX	5.2 - 7.1
DB2 Universal Database for Solaris Operating Environment	5.2 - 7.1
DB2 Universal Database for OS/390	4.1 - 7.1
DB2 DataJoiner	2.1.1

表 7. IBM 倉儲來源所支援的版本和版次 (繼續)

來源	版本/版次
DB2 for VM	5.3.4 或更新版
DB2 for VSE	7.1

來源	Windows NT	AIX
Informix	7.2.2 - 8.2.1	7.2.4 - 9.2.0
Oracle	7.3.2 - 8.1.5	8.1.5
Microsoft SQL Server	7.0	
Microsoft Excel	97	
Microsoft Access	97	
Sybase	11.5	11.9.2

表 8. IBM 倉儲目標檔所支援的版本和版次

目標	版本/版次
DB2 Universal Database for Windows NT	6 - 7
DB2 Universal Database Enterprise-Extended Edition	6 - 7
DB2 Universal Database for OS/2	6 - 7
DB2 Universal Database for AS/400	3.1-4.5
DB2 Universal Database for AIX	6 -7
DB2 Universal Database for Solaris Operating Environment	6 -7
DB2 Universal Database for OS/390	4.1 - 7
DB2 DataJoiner	2.1.1
DB2 DataJoiner/Oracle	8
DB2 for VM	3.4 - 5.3.4
DB2 for VSE	3.2, 7.1
CA/400	3.1.2

支援的非 IBM 資料庫來源的新增事項

下表所列，是對所支援之非 IBM 資料庫來源的新增項目：

資料庫	作業系統	資料庫從屬站需求
Informix	AIX	Informix-Connect 與 ESQ/C 版本 9.1.4 或更新版
Informix	Solaris 作業環境	Informix-Connect 與 ESQ/C 版本 9.1.3 或更新版
Informix	Windows NT	Informix-Connect for Windows Platforms 2.x 或 Informix-Client Software Developer's Kit for Windows Platforms 2.x

資料庫	作業系統	資料庫從屬站需求
Oracle 7	AIX	Oracle7 SQL*Net 及 Oracle7 SQL*Net 共用檔案庫 (利用 genclntsh script 所建立)
Oracle 7	Solaris 作業環境	Oracle7 SQL*Net 及 Oracle7 SQL*Net 共用檔案庫 (利用 genclntsh script 所建立)
Oracle 7	Windows NT	SQL*Net 現行版本的正確 DLL，外加 OCIW32.DLL。例如，SQL*Net 2.3 需使用 ORA73.DLL、CORE35.DLL、NLSRTL32.DLL、CORE350.DLL 及 OCIW32.DLL。
Oracle 8	AIX	Oracle8 Net8 及 Oracle8 SQL*Net 共用檔案庫 (利用 genclntsh8 script 所建立)
Oracle 8	Solaris 作業環境	Oracle8 Net8 及 Oracle8 SQL*Net 共用檔案庫 (利用 genclntsh8 script 所建立)
Oracle 8	Windows NT	若要存取遠端等級為 8.0.3 版以上的 Oracle8 資料庫伺服器，您便須安裝 Oracle Net8 Client 版本 7.3.4.x、8.0.4 或更新版。 在 Intel 系統上，您必須在您的路徑中安裝 Oracle Net8 Client 的正確 DLL (如：Ora804.DLL、PLS804.DLL 及 OCI.DLL)。
Sybase	AIX	在非 DCE 環境中 (ibsyb15 ODBC 驅動程式)： libct 檔案庫 在 DCE 環境中 (ibsyb1115 ODBC 驅動程式)： Sybase 11.1 從屬站檔案庫 libct_r
Sybase	Solaris 作業環境	在非 DCE 環境中 (ibsyb15 ODBC 驅動程式)： libct 檔案庫 在 DCE 環境中 (ibsyb1115 ODBC 驅動程式)： Sybase 11.1 從屬站檔案庫 libct_r
Sybase	Windows NT	Sybase Open Client-Library 10.0.4 或更新版及適當的 Sybase Net-Library。

在資料倉儲中心中手動建立資料來源

當使用 Relational Connect 及 『Create Nickname』 陳述式來建立資料來源時，對於在資料倉儲中心中匯入表格的相關函數，該資料來源將無法使用。若要將資料來源用作來源表格或目標表格，請執行下列步驟：

1. 定義來源-目標，而不匯入任何表格。
2. 展開資料倉儲中心主視窗的倉儲來源/目標樹狀結構，並在所欲來源-目標的『表格』上按一下滑鼠右鍵。
3. 按一下**定義**。
4. 使用開啓的筆記本定義資料來源，並確定為每一個資料來源定義了直欄。

若需詳細資訊，請參閱「資訊中心」中的『定義倉儲來源表格』或『定義倉儲目標表格』。

使用 Common Warehouse Metadata Interchange (CWMI) 匯入及匯出描述資料

簡介

除了目前支援的標示語言檔案，資料倉儲中心現在也可以匯入及匯出描述資料到符合 Common Warehouse Metamodel (CWM) 標準的 XML 檔案，以及從 XML 檔案匯入及匯出描述資料。匯入及匯出這些依照 CWM 的 XML 檔案就是所指的 *Common Warehouse Metadata Interchange (CWMI)*。

您可以從下列資料倉儲中心物件匯入及匯出描述資料：

- 倉儲來源
- 倉儲目標
- 主旨區域，包括程序、來源、目標及步驟
- 使用者定義程式

CWMI 匯入及匯出公用程式目前不支援特定類型的描述資料，包括：排程、倉儲綱目、捷徑步驟、重疊關係、使用者及群組。

資料倉儲中心會建立一個包含匯入及匯出程序結果的日誌檔。通常日誌檔會建立在 `x:\program files\sqllib\logging` 目錄 (其中 `x:` 是安裝 DB2 的磁碟機)，或是您指定在 `VWS_LOGGING` 環境變數的目錄。日誌檔是純文字的；您可以用任何文字編輯程式來檢視它。

匯入描述資料

您可以在資料倉儲中心中匯入描述資料，或者從命令行匯入描述資料。

透過匯入程序所建立的新物件會指派給預設的資料倉儲中心安全群組。若需詳細資訊，請參閱版本注意事項中的「在匯入後更新機密保護」。

如果是匯入關於某步驟的描述資料，則可以將多個檔案與步驟相關聯。關於步驟的描述資料會儲存在 XML 檔案中，但是有時步驟會有相關聯的資料是儲存為 BLOB。BLOB 描述資料的檔名與 XML 檔案相同，但是它是在個別的檔案中，其副檔名為數字。所有相關的步驟檔案在匯入時必須在相同的目錄中。

當步驟在測試或生產模式時更新步驟

步驟必須在開發模式中，資料倉儲中心才能更新步驟的描述資料。如果步驟是在測試或生產模式，則在匯入描述資料之前將步驟降級為開發模式：

1. 登入資料倉儲中心。
2. 在要降級的步驟上按一下滑鼠右鍵，再按一下**模式**。
3. 按一下**開發**。

現在步驟就在開發模式了。在匯入描述資料之後，再將步驟變更回測試或生產模式。

從資料倉儲中心匯入資料

您可以在資料倉儲中心之中匯入描述資料：

1. 登入資料倉儲中心。
2. 按一下左窗格中的**倉儲**。
3. 按一下 **選取的 -> 匯入描述資料 -> 交換檔案...**。
4. 在「匯入描述資料」視窗中，指定包含要匯入的描述資料的檔名。您可以鍵入檔名或用瀏覽來找檔案。
 - 如果您知道位置，則請鍵入完整的路徑和要匯入的檔案名稱。請確定加上 **.xml** 副檔名，以指定要以 **XML** 格式來匯入描述資料。
 - 若要用瀏覽來找檔案：
 - a. 按一下 **(...)** 按鈕。
 - b. 在「檔案」視窗中，將**檔案類型**變更為 **XML**。
 - c. 移至正確的目錄，再選取要匯入的檔案。

註： 檔案必須為 **.xml** 副檔名。
 - d. 按一下**確定**。
5. 在**匯入描述資料**視窗中，按一下**確定**來完成。在資料倉儲中心匯入檔案時會顯示**進度**視窗。

使用命令行來匯入描述資料

您也可以使用命令行來匯入描述資料。下列為匯入的指令語法：

```
CWMImport XML_file dwcControlDB dwcUserId dwcPW [PREFIX = DWCtbschema]
```

<i>XML_file</i>	要匯入的 XML 檔案的完整路徑和檔案名稱 (包括磁碟機及目錄)。此參數是必要的。
<i>dwcControlDB</i>	描述資料所要匯入的倉儲控制資料庫名稱。此參數是必要的。
<i>dwcUserId</i>	您用來登入倉儲控制資料庫的使用者 ID 。此參數是必要的。
<i>dwcPW</i>	您用來登入倉儲控制資料庫的使用者通行碼。此參數是必要的。
<i>[PREFIX=DWCtbschema]</i>	資料倉儲中心系統表格的資料庫綱目名稱，有時是指表格字首。如果沒有指定值給 PREFIX= ，則預設綱目名稱為 IWH 。此參數是可選用的。

在執行匯入公用程式之後更新描述資料

在匯入後更新機密保護

作為安全上的考量，資料倉儲中心不會匯入或匯出通行碼。您必須視需要在新物件上更新通行碼。若需匯入注意事項的詳細資訊，請參閱資料倉儲中心管理手冊的第 12 章「匯出和匯入資料倉儲中心描述資料」。

在匯入描述資料時，所有物件都會指派給預設安全群組。您可以變更可存取物件的群組：

1. 登入資料倉儲中心。
2. 在包含要變更的物件的資料夾上按一下滑鼠右鍵。
3. 按一下**內容**，再按一下**機密保護**標籤。
4. 從**選取的倉儲群組**列示中將群組除去，或者從**可用的倉儲群組**列示中新增群組。
5. 按一下**確定**。

匯出描述資料

您可以在資料倉儲中心中匯出描述資料，或者從命令行匯出描述資料。

部份步驟的描述資料是儲存為 BLOB。BLOB 描述資料會匯出到個別檔案中，其檔名與步驟的 XML 檔案相同，但是副檔名是數字 (.1、.2 等等)。

從資料倉儲中心匯出資料

您可以在資料倉儲中心之中匯出描述資料：

1. 登入資料倉儲中心。
 2. 按一下左窗格中的**倉儲**。
 3. 按一下 **選取的** -> **匯出描述資料** -> **交換檔案**。
 4. 在「匯出描述資料」視窗中，指定將包含匯出的描述資料的檔名。您可以輸入檔名或用瀏覽來找檔案：
 - 如果您知道完整的路徑及要使用的檔名，請輸入到**檔名**輸入欄位。請確定加上 .xml 副檔名，以指定要以 XML 格式來匯出描述資料。
 - 若要用瀏覽來找檔案：
 - a. 按一下 (...) 按鈕。
 - b. 在**檔案**視窗中，將**檔案類型**變更為 XML。
 - c. 移至正確的目錄，再選取要匯出的檔案。
- 註：**任何您所選取的現存檔案會被匯出的描述資料覆寫。
- d. 按一下**確定**。
 5. 當「匯出描述資料」視窗顯示了正確的檔名，在**可用的物件**列示中按一下要匯出其描述資料的物件。
 6. 按一下 > 符號將選取的物件從**可用的物件**列示中移動到**選取的物件**列示。重複以上步驟，直到所有要匯出的物件都列示在**選取的物件**列示中。
 7. 按一下**確定**。

資料倉儲中心會建立一個輸入檔案 (它包含與要匯出的資料倉儲中心物件的相關資訊)，然後匯出有關那些物件的描述資料。在資料倉儲中心匯出描述資料時會顯示進度視窗。

當匯出程序完成時，將會收到有關匯出程序的參考訊息。回覆碼 0 表示匯出成功。您也可以檢視日誌檔，以取得詳細資訊。

使用命令行來匯出描述資料

在從命令行匯出描述資料之前，您必須先建立一個輸入檔案。輸入檔是一個副檔名為 .INP 的文字檔，它會依照要匯出的物件類型列出所有物件。當您在資料倉儲中心之中匯出時，輸入檔是自動建立的，但是從命令行匯出，您必須先建立輸入檔。您可以使用任何文字編輯程式來建立輸入檔。鍵入所有的物件名稱 (與顯示在資料倉儲中心中的名稱相同)。請確定檔案是建立在一個可讀寫的目錄中。當您執行匯出公用程式時，資料倉儲中心會將 XML 檔案寫入到輸入檔所在的相同目錄中。

下列為一個範例輸入檔：

```
<PROC>
教學指導事實表格程序
<IR>
教學指導檔案來源
教學指導目標
<UDP>
新建程式群組
```

在 <PROC> (程序) 區段，列出所有要匯出的程序。在 <IR> (資訊資源) 區段，列出所有要匯出的倉儲來源及目標。資料倉儲中心會自動併入與這些來源及目標相關聯的表格和直欄。在 <UDP> (使用者定義程式) 區段，列出所有要匯出的程式群組。

若要匯出描述資料，在 DOS 提示符號下輸入下列指令：

```
CWMEexport INPcontrol_file dwcControlDB dwcUserID dwcPW [PREFIX=DWCtbschema]
```

<i>INPcontrol_file</i>	包含要匯出物件的 .INP 檔案的完整路徑和檔案名稱 (包括磁碟機及目錄)。此參數是必要的。
<i>dwcControlDB</i>	要從其匯出的倉儲控制資料庫名稱。此參數是必要的。
<i>dwcUserID</i>	您用來登入倉儲控制資料庫的使用者 ID。此參數是必要的。
<i>dwcPW</i>	您用來登入倉儲控制資料庫的通行碼。此參數是必要的。
<i>[PREFIX=DWCtbschema]</i>	資料倉儲中心系統表格的資料庫綱目名稱，有時是指表格字首。如果沒有指定值給 PREFIX=，則預設值為 IWH。此參數是可選用的。

OS/390 Runstats 公用程式步驟

在 DWC 中定義 OS/390 Runstats 公用程式步驟時，使用者應該知道下列位於「步驟內容對話框」的「參數」標籤中的內容。若是表格空間欄位，使用者應該以大寫輸入名稱。如果表格空間不在資料庫 DSNDB04 中，則需要以資料庫包含它來定義表格空間名稱。例如，輸入 SAMPLE.EMPLOYEE。說明目前不涉及此欄位。

OS/390 載入公用程式步驟

在 DWC 中定義 OS/390 載入公用程式步驟時，使用者應該知道下列位於「步驟內容對話框」的「參數」標籤中的內容。爲了要讓載入能夠執行，使用者一定要選取「進階」按鈕。否則將無法建立載入陳述式的 INTO 子句，在執行載入時將會失敗。

此外，FixPak 3 包含的修正程式可除去載入資料集名稱前後的雙引號。少了此修正程式，將無法執行載入。

Common Warehouse Metamodel (CWM) XML 支援

版本 7.2 CWM 工具集使用 Java Development Kit (JDK) 1.2.2 或 1.3。

現在，您可以匯入及匯出下列 CWM XML 物件：

其它處理程序的捷徑步驟

當您匯出含有某步驟的處理程序，而該步驟與另一處理程序中的步驟有關係（一個捷徑），這兩個處理程序都會被匯出且其關係會受到維護。

有條件地階式排列關係

現在，您可以匯入及匯出步驟之間不同的階式排列關係，包括 CHILD、SUCCESS、FAILURE 及 UNCONDITIONAL。

倉儲來源作為概略表物件

現在，當您匯出時，您可以將倉儲來源定義爲概略表物件。概略表物件的處理方式與表格物件相同。

SQLDataType 用於直欄和欄位

現在，您可以將 SQLDataType 用於直欄和欄位。

相同表格的多重相關名稱

在匯入或匯出期間，您可以有多重相關名稱，相同表格的每一個名稱都有自己的直欄對映。

新的 SAP 和 WebSphere Site Analyzer (WSA) 來源支援

新增新的來源支援標籤之後，現在您可以匯出儲存於倉儲中的 SAP 和 WSA 資訊。

處理模型產生器

您可以調整處理模型產生器選用區大小，使它符合您的螢幕大小。選用區上的圖示會自動重新定位至多直欄選用區。

當您按一下某個選用區圖示時，您會在選用區物件上看到一個標題。

現在，您可以藉由選取概略表功能表中的顯示業務名稱選項，依照表格和檔案物件的業務名稱來檢視它們。您亦可藉由選取縮放選項來調整百分比設定，以及縮小或放大處理程序概略表。

若物件在選用區內重疊，您可以按一下物件將它們拉至螢幕頂端。此外，物件名稱會往下折成許多列以節省選用區空間。

您現在可以使用刪除鍵來移除物件。當儲存某個處理程序時，也會儲存表格變更。選擇行為不是自動的。若要從倉儲來源或目標及處理程序中移除某個表格、檔案或概略表，您可以按一下滑鼠右鍵，然後選取**從來源移除**動作（若物件位於倉儲來源中），或選取**從目標移除**動作（若物件位於倉儲目標中）。

現在，游標顯示選用區選擇狀態。此外，狀態行顯示游標所在的物件名稱。

綱目模型產生器

現在，您可以將綱目模型產生器內表格縮至最小及放至最大。當您將表格縮至最小時，它變成圖示。爲了更精確顯示，您可以建立一個星狀綱目佈置。

必要輸入欄位

現在，「資料倉儲中心」對必要的欄位顯示紅色邊框。紅色邊框提醒您定義「資料倉儲中心」物件所需的必要資訊，例如資料庫名稱、使用者 ID 或通行碼。當您輸入必要的資訊後，邊框便會消失。

「資料倉儲中心」發射台加強功能

當您從發射台建立「資料倉儲中心」物件時，會展開導引樹狀結構，顯示新物件的位置。

列印步驟資訊到檔案

現在，您可以將步驟的相關資訊（例如主旨區域、來源表格名稱及目標表格名稱）列印至文字檔。若要列印步驟資訊至檔案，請以滑鼠右鍵按一下處理模型產生器中的步驟圖示、按一下**列印** → **列印至檔案**，然後指定您要列印資訊的檔案名稱。

Data Warehouse Center Application Integration Guide(無中文版)

In Chapter 5. Metadata Templates, Table 16 describes Column tag tokens. The information in the manual should state that `*ColumnPositionNumber` should start with "1". The manual incorrectly gives "0" as the starting character.

Later in Chapter 5, in Table 42, the `TableTypeOfFile` token is required if the type specified for the `DatabaseType` token in the corresponding `SourceDataBase.tag` is `ISV_IR_FF`. If it is not specified, an error will be detected.

In Chapter 6. Data Warehouse Center metadata, the description of the `POSNO` column object property should be changed to:

An index, starting with 1, of the column
or field in the row of the table or file.

In Chapter 8. Information Catalog Manager object types, the directory where you can find the `.TYP` files, which include the tag language for defining an object type, has been changed to `\SQLLIB\DGWIN\TYPES`.

Additional metadata templates

In Chapter 5, Metadata Templates, the following metadata templates should be included.

表 9. New Metadata templates supplied with the Data Warehouse Center

Template	Description	See:
<code>commit.tag</code>		『Commit.tag』
<code>Foreignkey.tag</code>		第246頁的『ForeignKey.tag』
<code>Foreignkeyadditional.tag</code>		第248頁的『ForeignKeyAdditional.tag』
<code>Primarykey.tag</code>		第250頁的『PrimaryKey.tag』
<code>Primarykeyadditional.tag</code>		第252頁的『PrimaryKeyAdditional.tag』

Commit.tag

Use this template to improve performance when you are using large tag language files. A commit template can be inserted between any of the groups of templates described here. A commit template cannot be inserted between templates within a group. For example, it is valid to insert a commit template between `AgentSite.tag` and `VWPGGroup.tag` but invalid to insert a commit tag between `VWPPProgramTemplate.tag` and `VWPPProgramTemplateParameter.tag`. If commit templates are used incorrectly, import may report an error.

- `AgentSite.tag`
- `VWPGGroup.tag`
- `VWPPProgramTemplate.tag`, `VWPPProgramTemplateParameter.tag`
- `SourceDatabase.tag`
- `WarehouseDatabase.tag`

- Table.tag, Column.tag
- SubjectArea.tag
- Process.tag
- Step.tag, StepInputTable.tag, StepOutputTable.tag, StepVWPOutputTable.tag, StepVWPProgramInstance.tag, VWPProgramInstanceParameter.tag
- StepCascade.tag
- StarSchema.tag, StarSchemaInputTable.tag
- PrimaryKey.tag, PrimaryKeyAdditional.tag
- ForeignKey.tag, ForeignKeyAdditional.tag

The use of the commit template is optional.

Tokens

Table 10 provides information about each token in the template.

Table 10. Commit.tag tokens

Token	Description	Allowed values
Relationship parameters		
*CurrentCheckPointID++	An index, starting with 0, that increases each time it is substituted in a token.	A numeric value.
This token is required.		

Examples of values

Table 11 provides example values for each token to illustrate the kind of metadata you might provide for each token.

Table 11. Example values for Commit.tag tokens

Token	Example value
*CurrentCheckPointID++	1

ForeignKey.tag

Use this template to define foreign key constraints on tables. The ForeignKey.tag template defines the relationships to the table and the column on which the constraint is being defined. This template also defines the relationships to the table and column of the primary key that is being referred to. Before you use the ForeignKey.tag template, you must define the primary key constraint (using the PrimaryKey.tag template) and the tables and columns (using the Table.tag and Column.tag templates) on which you want to define the foreign key constraint.

Tokens

Table 12 provides information about each token in the template.

Table12. ForeignKey.tag tokens

Token	Description	Allowed values
Entity parameters		
*ConstraintName	The name of the constraint. The name must be unique within a table or field. This token is required.	A text string, up to 80 bytes in length.
*ForeignColumnName	Name of the column on which the foreign key constraint is being defined.	A text string, up to 254 bytes in length.
*ForeignKeyID	The key that uniquely identifies the foreign key. The key must be unique from all other keys in the tag language file. Tip: Finish processing the ForeignKey.tag template before increasing the value of the key. This token is required.	A numeric value.
*MapID	An arbitrary number that is unique from all other keys in the interchange file. Tip: Finish processing the ForeignKey.tag template before increasing the value of this token. This token is required.	A numeric value.
*PrimaryColumnName	The column name of the referenced column.	A text string, up to 80 bytes in length.
*ReferencedPrimaryKeyID	The key that uniquely identifies the primary key. The key must be unique from all other keys in the tag language file. Tip: Finish processing the ForeignKey.tag template before increasing the value of the key. This token is required.	A numeric value.
Relationship parameters		
*DatabaseName	The business name of the warehouse source or warehouse target. This token is required.	A text string, up to 40 bytes in length.

Table 12. *ForeignKey.tag* tokens (continue)

Token	Description	Allowed values
<i>*ForeignKeyPhysicalName</i>	The database-defined name of the physical table containing the foreign keys that reference the keys in other tables.	A text string, up to 254 bytes in length.
<i>*PrimaryTablePhysicalName</i>	The database-defined name of the physical table containing the keys that are referenced by the foreign keys.	A text string, up to 80 bytes in length.
<i>*PrimaryTableOwner</i>	The owner, high-level qualifier, collection, or schema of the table that contains the primary key column that is being referenced. This token is required.	A text string, up to 128 bytes in length.
<i>*ForeignKeyOwner</i>	The owner, high-level qualifier, collection, or schema of the table that contains the foreign key constraint column. This token is required.	A text string, up to 128 bytes in length.

Examples of values

Table 13 provides example values for each token to illustrate the kind of metadata that you might provide for each token.

表 13. Example values for *ForeignKey.tag* tokens

Token	Example value
<i>*ConstraintName</i>	Department
<i>*DatabaseName</i>	Finance Warehouse
<i>*ForeignKeyColumnName</i>	Country_code
<i>*ForeignKeyID</i>	07011
<i>*ForeignKeyPhysicalName</i>	Geography
<i>*MapID</i>	02568
<i>*PrimaryColumnName</i>	State_code
<i>*Referenced PrimaryKeyID</i>	Name
<i>*PrimaryTablePhysicalName</i>	City
<i>*PrimaryTableOwner</i>	DB2ADMIN
<i>*ForeignKeyOwner</i>	IWH

ForeignKeyAdditional.tag

Use this template to define a composite foreign key. Before you use the *ForeignKeyAdditional.tag* template, you must define a constraint (using the *ForeignKey.tag* template) on the first column. You can then add columns by using this template for each column that you want to add.

Tokens

Table 14 provides information about each token in the template.

Table 14. *ForeignKeyAdditional.tag* tokens

Token	Description	Allowed values
Entity parameters		
<i>*ForeignKeyColumnName</i>	The name of the column on which the foreign key constraint is being defined.	A text string, up to 80 bytes in length.
<i>*ForeignKeyID</i>	The key that uniquely identifies the foreign key. The key must be unique from all other keys in the tag language file. Tip: Finish processing the <i>ForeignKey.tag</i> template before increasing the value of the key. This token is required.	A numeric value.
<i>*MapID</i>	An arbitrary number that is unique from all other keys in the interchange file. Tip: Finish processing the <i>ForeignKeyAdditional.tag</i> template before increasing the value of this token. This token is required.	A numeric value.
<i>*MapSeqNo</i>	A number signifying each additional column added as part of a composite key to the foreign key constraint.	Unique, increasing, consecutive number starting at 2.
<i>*PrimaryColumnName</i>	The column name of the referenced column.	A text string, up to 80 bytes in length.
Relationship parameters		
<i>*DatabaseName</i>	The business name of the warehouse source or warehouse target. This token is required.	A text string, up to 40 bytes in length.
<i>*ForeignTablePhysicalName</i>	The database-defined name of the physical table containing the foreign keys that reference the keys in other tables.	A text string, up to 80 bytes in length.
<i>*PrimaryTablePhysicalName</i>	The database-defined name of the physical table containing the keys that are referenced by the foreign keys.	A text string, up to 80 bytes in length.

Table 14. *ForeignKeyAdditional.tag* tokens (continue)

Token	Description	Allowed values
<i>*PrimaryTableOwner</i>	The owner, high-level qualifier, collection, or schema of the table that contains the primary key column that is being referenced. This token is required.	A text string, up to 128 bytes in length.
<i>*ForeignTableOwner</i>	The owner, high-level qualifier, collection, or schema of the table that contains the foreign key constraint column. This token is required.	A text string, up to 128 bytes in length.

Examples of values

表15 provides example values for each token to illustrate the kind of metadata that you might provide for each token.

表 15. Example values for *ForeignKeyAdditional.tag* tokens

Token	Example value
<i>*DatabaseName</i>	Finance Warehouse
<i>*ForeignColumnName</i>	Country_code
<i>*ForeignKeyID</i>	07011
<i>*ForeignTablePhysicalName</i>	Geography
<i>*MapID</i>	22578
<i>*MapSeqNo</i>	2
<i>*PrimaryColumnName</i>	State_code
<i>*PrimaryTablePhysicalName</i>	City
<i>*PrimaryTableOwner</i>	DB2ADMIN
<i>*ForeignTableOwner</i>	IWH

PrimaryKey.tag

Use this template to define primary key constraints on tables. The template also defines the relationships to the table and the column on which the constraint is being defined. Before you use the *PrimaryKey.tag* template, you must define the tables and columns (using the *Table.tag* and *Column.tag* templates) on which you want to define the primary key constraint.

Tokens

Table 16 provides information about each token in the template.

Table 16. *PrimaryKey.tag* tokens

Token	Description	Allowed values
Entity parameters		

Table 16. PrimaryKey.tag tokens (continue)

Token	Description	Allowed values
*ColumnName	The name of the column or field. A text string, up to 80 bytes in length. The name must be unique within a table or field. This token is required.	
*MapID	An arbitrary number that is unique from all other keys in the interchange file. Tip: Finish processing the PrimaryKey.tag template before increasing the value of this token. This token is required.	A numeric value.
*PrimaryKeyID	The key that uniquely identifies the primary key. The key must be unique from all other keys in the tag language file. Tip: Finish processing the ForeignKey.tag template before increasing the value of the key. This token is required.	A numeric value.
Relationship parameters		
*DatabaseName	The business name of the warehouse source or warehouse target. This token is required.	A text string, up to 40 bytes in length.
*TableOwner	The owner, high-level qualifier, collection, or schema of the table that contains the column. This token is required.	A text string, up to 128 bytes in length.
*TablePhysicalName	The physical name of the table or file that contains the column as defined to the database manager or file system. This token is required.	A text string, up to 80 bytes in length.

Examples of values

Table 17 provides example values for each token to illustrate the kind of metadata that you might provide for each token.

Table 17. Example values for PrimaryKey.tag tokens

Token	Example value
*ColumnName	Country_code
*DatabaseName	Finance Warehouse
*MapID	54627
*PrimaryKeyID	74622
*TableOwner	DB2ADMIN
*TablePhysicalName	GEOGRAPHY

PrimaryKeyAdditional.tag

Use this template to define a composite primary key. Before you use the PrimaryKeyAdditional.tag template, you must define a constraint on the first column by using the PrimaryKey.tag template. Any additional columns can then be added using this template. The template also relates the additional primary keys to the first primary key which is defined using PrimaryKey.tag.

Tokens

Table 18 provides information about each token in the template.

Table 18. PrimaryKeyAdditional.tag tokens

Token	Description	Allowed values
Entity parameters		
*ColumnName	The name of the column or field. A text string, up to 80 bytes in length. The name must be unique within a table or field. This token is required.	
*FirstPrimaryKeyID	The key that uniquely identifies the primary key. The key must be unique from all other keys in the tag language file. Tip: Finish processing the ForeignKey.tag template before increasing the value of the key. This token is required.	A numeric value.
*MapID	An arbitrary number that is unique from all other keys in the interchange file. Tip: Finish processing the PrimaryKeyAdditional.tag template before increasing the value of this token. This token is required.	A numeric value.

Table 18. PrimaryKeyAdditional.tag tokens (continue)

Token	Description	Allowed values
*MapSeqNo	A number signifying each additional column added as part of a composite key to the primary key constraint.	Unique, increasing, consecutive number starting at 2.
Relationship parameters		
*DatabaseName	The business name of the warehouse source or warehouse target.	A text string, up to 40 bytes in length.
	This token is required.	
*TableOwner	The owner, high-level qualifier, collection, or schema of the table that contains the column.	A text string, up to 15 bytes in length.
	This token is required.	
*TablePhysicalName	The physical name of the table or file that contains the column as defined to the database manager or file system.	A text string, up to 80 bytes in length.
	This token is required.	

Examples of values

Table 19 provides example values for each token to illustrate the kind of metadata that you might provide for each token.

Table 19. Example values for PrimaryKeyAdditional.tag tokens

Token	Example value
*ColumnName	Country_code
*DatabaseName	Finance Warehouse
*MapID	99542
*MapSeqNo	2
*FirstPrimaryKeyID	07801
*TableOwner	DB2ADMIN
*TablePhysicalName	GEOGRAPHY

資料倉儲中心線上說明

定義抄寫的表格或概略表

使用 DB2 控制中心定義抄寫的表格或概略表之後，才能被資料倉儲中心當作抄寫來源使用。

執行具有 AS/400 代理程式的 Essbase VWP

執行具有 AS/400 代理程式的 Essbase VWP，ARBORLIB 和 ARBORPATHB 必須被設定為 *sys 環境變數。使用者 ID 必須有 *jobctl 權限才能進行設定。這些環境變數需要指向已安裝 Essbase 的檔案庫。

使用公佈資料倉儲中心描述資料視窗及已結合的內容視窗

在作業說明的步驟 10 中，有一個範例狀態是這樣的：如果您指定限制值 1 (限制樹狀結構層次的物件) 並公佈處理程序，則只公佈與顯示該處理程序的 1 步驟。在所有狀況下，此範例是不正確的。步驟 8 第二個項目符號的項目中，第一項陳述式是不正確的。它應該為「按一下直欄層次以產生資訊型錄來源直欄和目標直欄之間的轉換物件」。

外來鍵

「外來鍵」在線上說明的資料應為「倉儲外來鍵」。

抄寫筆記本

「定義抄寫筆記本」在線上說明的資料應為「抄寫步驟筆記本」。

匯入標示語言

匯入標示語言線上說明：項目符號列示顯示常發生的重要錯誤，其中一項為「匯入的標示語言檔案沒有適當地匯出」。此項目不適用於一般輸入錯誤的列示。

新增資料的鏈結

在線上說明的『新增資料』主題中，「新增來源表格到程序」及「新增目標表格到程序」主題的鏈結中斷。您可以在說明索引中找到這些主題。

匯入表格

說明主題「匯入來源表格及概略表到倉儲來源中」及「匯入目標表格到倉儲目標中」對於萬用字元有著不正確的描述。句子為：

例如，XYZ* 會傳回綱目是以這些字元開頭的表格及概略表。

應改為：

例如，XYZ% 會傳回綱目是以這些字元開頭的表格及概略表。

RUNSTATS 和 REORGANIZE TABLE 線上說明集合

這些公用程式的線上說明，指示要執行統計的表格，或是需要重組的表格，必須和來源及目標鏈結。由於步驟寫入來源，您只需要鏈結步驟和來源。

「通知」頁面，「倉儲內容」與「排程」筆記本

在「倉儲內容」筆記本的「通知」頁中寫道：

「傳送者」輸入欄位的起始設定值為 <current user's logon ID> 字串。

應變更為：

「傳送者」輸入欄位的起始設定值為 <current logon user email address> 字串。

在「排程」筆記本的「通知」頁中，傳送者會起始設定成「倉儲內容」筆記本中的設定值。若無設定值，便會將該值設定成目前登入之使用者的電子郵件地址。若找不到該登入使用者的電子郵件地址，便會將其設定為登入使用者的使用者 ID。

「代理程式端」筆記本的代理程式模組欄位

倉儲代理常駐程式產生倉儲代理程式時，「代理程式端」筆記本的代理程式模組欄位提供執行程式的名稱。不要變更欄位的名稱，除非 IBM 要求您要這麼做。

DB2 OLAP Starter Kit

IBM DB2 OLAP Starter Kit 7.2 在一些作業系統上增加 Oracle、MS-SQL、Sybase 及 Informix 關聯式資料庫管理系統 (RDBMS) 的支援。版本 7.2 包含所有支援的 RDBMS 的 Script 和工具，包括 DB2。它們有一些限制，詳細資訊請參閱第264頁的『已知問題與限制』。

DB2 OLAP Starter Kit for DB2 Universal Database 版本 7.2 的服務程式等級相當於 Hyperion Essbase 6.1 修補程式 2 加上 Hyperion Integration Server 2.0 修補程式 2。

OLAP Server 網站

若需 DB2 OLAP Starter Kit 最新的安裝及使用秘訣，請造訪 DB2 OLAP Server 網站的 Library 網頁：

<http://www.ibm.com/software/data/db2/db2olap/library.html>

支援的作業系統服務程式等級

OLAP Starter Kit for Version 7.2 的伺服器元件支援下列作業系統和服務程式等級：

- 安裝 SP 5 的 Windows NT 4.0 與 Windows 2000
- AIX 版本 4.3.3 或以上
- Solaris 作業系統版本 2.6、7 與 8 (Sun OS 5.6、5.7 或 5.8)

從屬站元件執行於 Windows 95、Windows 98、Windows NT 4.0 SP5 與 Windows 2000。

完成 DB2 OLAP Starter Kit 在 UNIX 上的安裝

安裝 DB2 OLAP Starter Kit 請依照 DB2 Universal Database for UNIX 的安裝基本程序進行。產品檔案會由安裝程式放到系統目錄：(AIX: /usr/lpp/db2_07_01；Solaris: /opt/IBMDB2/V7.1)。

建立案例的階段，會建立兩個 DB2 OLAP 目錄 (essbase 與 is) 位於案例使用者起始目錄下 sqllib 的目錄下。同一時間機器僅允許執行一個 OLAP 伺服器案例。若要完成安裝，使用者必須手動設定 is/bin 目錄，因為它無法鏈結到系統的 is/bin 目錄。它必須鏈結到案例的起始目錄下的可寫入目錄。

若要完成在 Solaris 上的安裝，請以案例 ID 登入切換到 sqllib/is 目錄，然後輸入下列指令：

```
rm bin
mkdir bin
cd bin
ln -s /opt/IBMDB2/V7.1/is/bin/ismesg.mdb ismesg.mdb
ln -s /opt/IBMDB2/V7.1/is/bin/olapicmd olapicmd
ln -s /opt/IBMDB2/V7.1/is/bin/olapisvr olapisvr
ln -s /opt/IBMDB2/V7.1/is/bin/essbase.mdb essbase.mdb
ln -s /opt/IBMDB2/V7.1/is/bin/libolapams.so libolapams.so
```

架構 OLAP Starter Kit 的 ODBC

IBM DB2 OLAP Starter Kit 7.2 需要一個 ODBC.ini 檔案，讓 ODBC 作業從 OLAP Integration Server 連接至關聯式資料來源和「OLAP 描述資料型錄」。

- 在 Windows 系統上，此檔案位於系統登錄的 HKEY_LOCAL_MACHINE/SOFTWARE/ODBC 內。請使用 ODBC Data Source Administrator 來儲存有關如何連接至關聯式資料來源的資訊。
- 在 UNIX 系統上，安裝程式建立一個模型檔 odbc.ini。欲儲存有關如何連接至關聯式資料來源的資訊，請使用您喜好的編輯器來編輯此檔案。

此 ODBC.ini 檔案由 ODBC 軟體套件提供，亦內含於 Microsoft Office 軟體內。關於可安裝 ODBC 驅動程式或 ODBC 管理程式的應用程式的詳細資訊，請造訪下列網站：
<http://support.microsoft.com/support/kb/articles/Q113/1/08.asp>。

AIX 機器上的 Oracle 使用者：欲架構 ODBC for Oracle，您必須更新 ODBC.ini 檔案來指向 MERANT 3.6 驅動程式。

在版本 7.2 中，OLAP Starter Kit 管理 ODBC 連線來連接至關聯式資料來源和「OLAP 描述資料型錄」。為了支援這些 ODBC 連接，OLAP Starter Kit 在 Windows NT 4.0、Windows 2000、AIX 及 Solaris 上使用 ODBC 驅動程式。

- DB2 Universal Database 版本 6 資料庫從屬站：在 Windows NT 4.0 SP5 或 Windows 2000、AIX 4.3.3 及 Solaris 作業系統 2.6、7 或 8 (Sun OS 5.6、5.7 或 5.8) 上是 DB2 版本 6 ODBC 驅動程式。
- DB2 Universal Database 7.1 資料庫從屬站：在 Windows NT 4.0 SP5 或 Windows 2000、AIX 4.3.3 及 Solaris 作業系統 2.6、7 或 8 (Sun OS 5.6、5.7 或 5.8) 上是 DB2 版本 7 ODBC 驅動程式。
- Oracle 8.04 和 8i SQL*Net 8.0 資料庫從屬站：在 Windows NT 4.0 SP5 或 Windows 2000、AIX 4.3.3、Solaris 作業系統 2.6、7 或 8 (Sun OS 5.6、5.7 或 5.8) 上是 MERANT 3.6 ODBC 驅動程式。
- MS SQL Server 6.5.201 (不需要「資料庫從屬站»)：在 Windows NT 4.0 SP5 或 Windows 2000 上是 MS SQL Server 6.5 ODBC 驅動程式。
- MS SQL Server 7.0 (不需要「資料庫從屬站»)：在 Windows NT 4.0 SP5 或 Windows 2000 上是 MS SQL Server 7.0 ODBC 驅動程式。

在 UNIX 系統架構資料來源

在 AIX 和 Solaris 上，您必須手動設定 ODBC 的環境變數，並編輯 odbc.ini 檔案來架構關聯式資料來源和「OLAP 描述資料型錄」。若您新增或變更驅動程式或資料來源，請務必編輯 odbc.ini 檔案。

架構 ODBC 環境變數

在 UNIX 系統上，您必須設定環境變數才能存取 ODBC 核心元件。Starter Kit 起始目錄中提供 is.sh 和 is.csh Shell Script 來設定必要的變數。在使用 ODBC 來連接資料來源之前，您必須執行其中一個 Script。對於您用來執行 OLAP Starter Kit 的使用者名稱，您應該在登入 Script 中包含這些 Script。

編輯 odbc.ini 檔案

欲於 odbc.ini 檔案中架構資料來源，您必須新增 ODBC 資料來源的名稱和說明，在您針對此資料來源名稱所建立的區段中提供 ODBC 驅動程式路徑、檔名及其他驅動程式

設定值。安裝程式在 ISHOME 目錄中安裝一個範例檔 `odbc.ini`。此檔案包含通用 ODBC 連接和所支援的 ODBC 驅動程式的架構資訊。請使用此檔案做為起點，將您使用的 ODBC 驅動程式對映至關聯式資料來源和「OLAP 描述資料型錄」。

若您使用的檔案不是 `odbc.ini`，請將 `ODBCINI` 環境變數設定成您使用的檔名。

新增資料來源到 `odbc.ini` 檔

1. 在執行 OLAP Starter Kit 伺服器的系統上，請使用諸如 `vi` 的文字編輯程式來開啓 `odbc.ini` 檔案。
2. 找到 `[ODBC Data Sources]` 為首的區段，新增一行資料來源名稱和說明，例如：`mydata=data source for analysis`。為了避免混淆，資料來源的名稱應該符合 RDBMS 中的資料庫名稱。
3. 新增一行以方括弧包含新資料來源的名稱，例如 `[mydata]`，在檔案中新增一個新的區段。
4. 在資料來源名稱接下來的幾行中，新增此資料來源名稱及任何必要的 ODBC 驅動程式資訊所需的 ODBC 驅動程式之完整路徑和檔案名稱。根據下列幾節所示的範例來對映至您 RDBMS 上的資料來源。請確定 ODBC 驅動程式檔案確實存在於您在 `Driver=` 設定中指定的位置。
5. 當您編輯 `odbc.ini` 完成後，請儲存檔案，結束文字編輯程式。

ODBC 的 DB2 設定值範例

下列範例顯示如何編輯 `odbc.ini`，使用 IBM DB2 原始 ODBC 驅動程式來連接至 AIX 上的 DB2 Universal Database 版本 6.1 的關聯式資料來源 `db2data`。在 `vi` 編輯器中，使用 `$ODBCINI` 指令來編輯 `odbc.ini`，並插入下列陳述式：

```
[ODBC Data Sources]
db2data=DB2 Source Data on AIX
...
[db2data]
Driver=/home/db2inst1/sqllib/lib/db2.o
Description=DB2 Data Source - AIX, native
```

ODBC 的 Oracle 設定值範例

以下範例說明如何編輯 `odbc.ini`，使用 MERANT Version 3.6 ODBC 驅動程式來連接至 Oracle 版本 8 (在 Solaris) 的關聯式資料來源 `oradata`。此範例中，`LogonID` 和 `Password` 以 OLAP Starter Kit 使用者名稱和通行碼的實際值來填入。

```
[ODBC Data Sources]
oradata=Oracle8 Source Data on Solaris
...
[myoracle] Driver=
/export/home/users/dkendric/is200/odbc/lib/ARor815.so
Description=my oracle source
```

在 UNIX 系統上架構 OLAP 描述資料型錄

在 AIX 和 Solaris 上架構 OLAP Metadata Catalog 類似架構一個資料來源。若要 OLAP Metadata Catalog 資料庫，請在 `odbc.ini` 檔案中新增一個資料來源名稱和區段，如第 258 頁的『編輯 `odbc.ini` 檔案』所述。不需要其它變更。

您必須先在支援的 RDBMS 中建立一個「OLAP 描述資料型錄」資料庫，才能將它架構成 ODBC 資料來源。

以下範例說明如何編輯 `odbc.ini`，使用原始 ODBC 驅動程式來連接至 DB2 版本 6.1 (在 Solaris 上) 的「OLAP 描述資料型錄」TBC_MD。

```
[ODBC Data Sources]
odc6a5a=db2 v6
...
[odc6a5a]
Driver=/home/db2inst1/sql1lib/lib/db2.0
Description=db2
```

在 Windows 系統上架構資料來源

欲於 Windows NT 或 Windows 2000 系統上架構關聯式資料來源，您必須啟動「ODBC 管理程式」，然後建立連線來連接至您要用來建立 OLAP 模型和 metaoutline 的資料來源。從 Windows 控制台執行「ODBC 管理程式」公用程式。下列範例建立一個 DB2 資料來源；其他 RDBMS 的對話框會不同。

欲使用「ODBC 管理程式」來架構關聯式資料來源，請完成下列步驟：

1. 在 Windows 桌上管理程式，開啓「控制台」視窗。
2. 在「控制台」視窗中，執行下列其中一項步驟：
 - a. 在 Windows NT 上，按兩下 **ODBC** 圖示，開啓「ODBC 資料來源管理程式」對話框。
 - b. 在 Windows 2000，按兩下**系統管理工具**圖示，再按兩下**資料來源 (ODBC)** 圖示，開啓「ODBC 資料來源管理員」對話框。
3. 在「ODBC 資料來源管理程式」對話框中，按一下**系統資源來源名稱**標籤。
4. 按一下**新增**來開啓「建立新資料來源」對話框。
5. 在「ODBC 管理程式」的「建立新資料來源」對話框的驅動程式列示框中，選取適當的驅動程式，例如 IBM DB2 ODBC 驅動程式，按一下**完成**來開啓「ODBC IBMDB2 驅動程式 - 新增」對話框。
6. 在「ODBC IBM DB2 驅動程式 - 新增」對話框中，在**資料庫別名**下拉列示中，選取您的關聯式來源 (例如，此範例應用程式中的 TBC) 的資料庫名稱。
7. 在**說明**文字框中，輸入選用的說明來表示您要如何使用此驅動程式，按一下**新增**。例如，輸入下列文字來說明 My Business 資料庫：

Customers, products, markets

您可輸入下列文字來說明範例應用程式資料庫：

Sample relational data source

這些說明可在您從 OLAP Starter Kit Desktop 連線時，協助您識別可選擇的資料來源。

8. 按一下「確定」來返回「ODBC 資料來源管理員」對話框。您輸入的資料來源名稱及您對映的驅動程式顯示在**系統資源來源名稱**標籤的**系統資料來源**列示框中。

欲編輯資料來源的架構資訊：

1. 選取資料來源名稱，按一下**架構**來開啓「ODBC IBM DB2 - 新增」對話框。
2. 修正您要變更的任何資訊。
3. 按兩次**確定**以結束。

在 Windows 系統上架構 OLAP 描述資料型錄

欲於 Windows NT 或 Windows 2000 上架構「OLAP 描述資料型錄」，請啓動「ODBC 管理程式」，然後建立連線來連接至含有「OLAP 描述資料型錄」資料庫的資料來源。

下列範例建立一個 DB2 資料來源；其他 RDBMS 的對話框會不同。欲建立「OLAP 描述資料型錄」的資料來源，請完成下列步驟：

1. 在桌上管理程式，開啓「控制台」視窗。
2. 在「控制台」視窗中，執行下列其中一項步驟：
 - a. 在 Windows NT 上，按兩下 **ODBC** 圖示，開啓「ODBC 資料來源管理程式」對話框。
 - b. 在 Windows 2000，按兩下**系統管理工具**圖示，再按兩下**資料來源 (ODBC)** 圖示，開啓「ODBC 資料來源管理員」對話框。
3. 在「ODBC 資料來源管理程式」對話框中，按一下**系統資源來源名稱**標籤。
4. 按一下**新增**來開啓「建立新資料來源」對話框。
5. 在「ODBC 管理程式」的「建立新資料來源」對話框的驅動程式列示框中，選取適當的驅動程式，例如 IBM DB2 ODBC 驅動程式，按一下**完成**來開啓「ODBC IBMDB2 驅動程式 - 新增」對話框。
6. 在「ODBC IBM DB2 驅動程式 - 新增」對話框中，在**資料庫別名**下拉列示中，選取您的「OLAP 描述資料型錄」(例如，此範例應用程式中的 TBC_MD) 的資料庫名稱。選取的資料庫名稱會自動顯示在**資料來源名稱**文字框中。
7. 若您要變更資料來源的名稱，請選取**資料來源名稱**文字框中顯示的名稱，輸入新的名稱來表示您要如何使用此驅動程式，按一下**新增**。例如，您可輸入下列名稱來表示您要使用此驅動程式來連接至第一個「OLAP 描述資料型錄」：

OLAP Catalog first

您可輸入下列名稱來表示您要連接至範例應用程式「OLAP 描述資料型錄」資料庫：

TBC_MD

8. 在**說明**文字框中，輸入說明來表示您要如何使用此驅動程式。例如，您可輸入下列文字來說明「OLAP 描述資料型錄」：

My first models and metaoutlines

您可輸入下列文字來說明範例應用程式「OLAP 描述資料型錄」資料庫：

Sample models and metaoutlines

這些說明可在您從 OLAP Starter Kit Desktop 連接至「OLAP 描述資料型錄」時，協助您識別想要選擇的型錄。

9. 按一下「確定」來返回「ODBC 資料來源管理員」對話框。您輸入的資料來源名稱及您對映的驅動程式顯示在**系統資源來源名稱**標籤的**系統資料來源**列示框中。

欲編輯資料來源的架構資訊：

1. 選取資料來源名稱，按一下**架構**來開啓「ODBC IBM DB2 - 新增」對話框。
2. 修正您要變更的任何資訊。
3. 按兩次**確定**以結束。

在您的架構資料來源之後

在架構關聯式資料來源和「OLAP 描述資料型錄」之後，即可從 OLAP Starter Kit 來連接。然後可以建立、修改及儲存 OLAP 模型和 metaoutline。

在呼叫 SQL Server 資料庫期間，SQL Server ODBC 驅動程式可能會逾時。請於資料庫非忙線時重試。增加驅動程式逾時期間可避免此問題。詳細資訊，請參閱您所使用的驅動程式的 ODBC 文件。

關於 ODBC 連接問題與解決方案，請參閱 *OLAP Integration Server System Administrator's Guide*。

從 OLAP Starter Kit 桌上管理程式登入

使用 OLAP Starter Kit 桌上管理程式來建立 OLAP 模型和 Metaoutline，請將從屬站軟體連接到兩個伺服器：DB2 OLAP 整合伺服器和 DB2 OLAP 伺服器。登入對話提示桌上管理程式連接到這兩個伺服器的必要資訊。在對話的左邊，請輸入有關 DB2 OLAP 整合伺服器的資訊。在對話的右邊，請輸入有關 DB2 OLAP 伺服器的資訊。

欲連接到 DB2 OLAP 整合伺服器：

- 伺服器：輸入整合伺服器的主電腦名稱或 IP 位址。如果您已安裝整合伺服器和桌上管理程式於同一個工作站，典型的值為 "localhost" 或 "127.0.0.1"。
- OLAP 描述資料目錄：連接 OLAP 整合伺服器，必須指定描述資料目錄。OLAP 整合伺服器儲存有關 OLAP 模型和名為 Metadata Catalog 關聯式資料庫所建立的 metaoutline 的資訊。此關聯式資料庫必須登記給 ODBC。目錄資料庫有一組 OLAP 整合伺服器能辨識的關聯式表格。在登入對話中，指定整合伺服器，並展開 OLAP 描述資料目錄欄位的下拉功能表，您可以看到 OLAP 整合伺服器所知道的 ODBC 資料來源名稱之列表。選擇含有描述資料型錄表格的 ODBC 資料庫。
- 使用者名稱和通行碼：使用您在畫面所指定的使用者名稱和通行碼，OLAP 整合伺服器將與描述資料目錄連結。這是登入帳戶，存在於伺服器（不是從屬站，除非伺服器和從屬站在同一台機器執行）。使用者名稱必須為建立 OLAP 描述資料目錄的使用者。否則，OLAP 整合伺服器將找不到目錄資料庫內的關聯式表格，因為表格綱目的名稱不一樣。

DB2 OLAP 伺服器資訊為選用的，所以，在登入對話的右邊之輸入欄位可能是空白的。但是，有些桌上管理程式和管理程式的作業需要和 DB2 OLAP 伺服器連接。如果您把這些欄位空下來，當整合伺服器需要和 DB2 OLAP 伺服器連接以完成您所要求的作業時，桌上管理程式會再一次顯示登入對話。建議您在登入對話的 DB2 OLAP 伺服器欄位填入資料。

欲連接到 DB2 OLAP 伺服器：

- 伺服器：輸入 DB2 OLAP 伺服器的主電腦名稱或 IP 位址。如果您在執行 OLAP Starter Kit，則 OLAP 伺服器和整合伺服器相同。如果整合伺服器和 OLAP 伺服器安裝在不同的主電腦，請輸入定義於 OLAP 整合伺服器內的主電腦名稱或 IP 位址。
- 使用者名稱和通行碼：使用您在畫面所指定的使用者名稱和通行碼，OLAP 整合伺服器將與 DB2 OLAP 伺服器連結。此使用者名稱和通行碼必須已經定義在 DB2 OLAP 伺服器內。OLAP 伺服器自作業系統另外地管理它所擁有的使用者名稱和通行碼。

Starter Kit 登入範例

下列範例假設，在 OLAP Starter Kit 安裝期間，您建立了 OLAP 範例，並選取了 *db2admin* 作為管理者使用者 ID，*password* 為管理者通行碼。

- OLAP 整合伺服器：伺服器是 *localhost*，OLAP 描述資料目錄是 *TBC_MD*，使用者名稱是 *db2admin*，通行碼是 *password*
- DB2 OLAP 伺服器：伺服器是 *localhost*，使用者名稱是 *db2admin*

自行建立和架構 OLAP Starter Kit 的範例資料庫

安裝 OLAP Starter Kit 時，範例資料庫會自動建立。下列指示解譯，必要時，如何自行安裝目錄和範例資料庫。

1. 在視窗下，按一下 **開始** → **程式集** → **DB2 for Windows NT** → **命令視窗**，以開啓「命令中心」視窗。
2. 建立目錄資料庫：
 - a. 輸入 `db2 create db OLAP_CAT`
 - b. 輸入 `db2 connect to OLAP_CAT`
3. 建立資料庫內的表格：
 - a. 導覽 `\SQLLIB\IS\ocscript\ocdb2.sql`
 - b. 輸入 `db2 -tf ocdb2.sql`
4. 建立範例來源資料庫：
 - a. 輸入 `db2 connect reset`
 - b. 輸入 `db2 create db TBC`
 - c. 輸入 `db2 connect to TBC`
5. 建立資料庫內的表格：
 - a. 導覽 `\SQLLIB\IS\samples\`
 - b. 複製 `tbcdb2.sql` 到 `\SQLLIB\samples\db2sampl\tbc`
 - c. 複製 `lddb2.sql` 到 `\SQLLIB\samples\db2sampl\tbc`
 - d. 導覽 `\SQLLIB\samples\db2sampl\tbc`
 - e. 輸入 `db2 -tf tbcdb2.sql`
 - f. 輸入 `db2 -vf lddb2.sql`，以載入範例來源資料到表格內。
6. 建立範例目錄資料庫：
 - a. 輸入 `db2 connect reset`
 - b. 輸入 `db2 create db TBC_MD`
 - c. 輸入 `db2 connect to TBC_MD`
7. 建立資料庫內的表格：
 - a. 導覽 `\SQLLIB\IS\samples\tbc_md`
 - b. 複製 `ocdb2.sql` 到 `\SQLLIB\samples\db2sampl\tbcmd`
 - c. 複製 `lcdb2.sql` 到 `\SQLLIB\samples\db2sampl\tbcmd`
 - d. 導覽 `\SQLLIB\samples\db2sampl\tbcmd`
 - e. 輸入 `db2 -tf ocdb2.sql`
 - f. 輸入 `db2 -vf lcdb2.sql`，以載入範例描述資料到表格。

8. 架構 ODBC for TBC_MD, TBC, AND OLAP_CAT :
 - a. 按一下**開始**→**設定**→**控制台**，以開啓 NT 控制台
 - b. 從列示中選取 ODBC (或 ODBC 資料來源)。
 - c. 選取「系統 DSN」標籤。
 - d. 按一下「**新增**」。 **建立新的資料來源**視窗開啓。
 - e. 從列示中選取 IBM DB2 ODBC DRIVER。
 - f. 按一下「**完成**」。 ODBC IBM D2 驅動程式 - 新增視窗開啓。
 - g. 在**資料來源名稱**欄位中，輸入資料來源 (OLAP_CAT) 的名稱。
 - h. 在**資料庫別名**欄位中，輸入別名，或是按一下↓，並在列示中選取 OLAP_CAT。
 - i. 按一下「**確定**」。
 - j. 對於 TBC_MD 和 TBC 資料庫，請重複以上步驟。

移轉應用程式至 OLAP Starter Kit 版本 7.2

安裝程式不重新安裝 OLAP Starter Kit 範例應用程式、資料庫及資料檔。您的現存應用程式和資料庫不受影響。然而，在安裝之前，最好先備份您的應用程式和資料庫。

您的應用程式在開啓時會自動移轉至版本 7.2。

已知問題與限制

本段落列示 DB2 OLAP Starter Kit 的已知限制。

Windows 平台的 Informix RDBMS 和 Merant 驅動程式的相容性

爲了讓 Windows 平台的 Merant 驅動程式能夠和 Informix RDBMS 一起運作，PATH 陳述式中必須新增下列兩個項目：

- C:\Informix
- C:\Informix\bin

這兩個項目必須位於 PATH 的開頭。

OLAP 模型中的維度和相關 Metaoutline 之間可能的不一致情形

在某些情況下，您建立的維度在 OLAP 模型中無相對應的維度。在下列實務中會發生這些情形：

1. 建立並儲存一個新的 OLAP 模型。
2. 根據此模型來建立一個 metaoutline，但並未儲存 metaoutline。
3. 返回 OLAP 模型，刪除其中一個 metaoutline 維度所依賴的維度。
4. 返回 metaoutline，開啓、關閉，再重新開啓。metaoutline 將包含一個維度，此維度在 OLAP 模型中無相對應的維度。

OLAP Starter Kit 無法區分這種情況下建立的不一致維度和 metaoutlin 中的使用者定義維度。因此，不一致維度會顯示在 metaoutline 中，但 metaoutline 視它爲一個使用者定義維度，因爲 OLAP 模型中無相對應的維度存在。

在 Windows 2000 平台上，TMP 的環境變數設定導致成員和資料載入失敗

因爲 Windows 2000 和 Windows NT 之間，TMP 的預設系統和使用者環境變數設定有所差異，當 OLAP Starter Kit 在 Windows 2000 平台上執行時，成

員和資料載入會失敗。造成的錯誤訊息告訴使用者無法建立暫存檔。您可採取下列步驟來解決此 Windows 2000 的限制：

1. 建立一個目錄 C:\TEMP
2. 將系統和使用者的環境變數 TMP 設定為 TMP=C:\TEMP

安裝 ODBC 不取代現存的 Merant 驅動程式

現存的 3.6 Merant ODBC 驅動程式不隨此安裝而更新。若您從 OLAP Starter Kit 版本 7.1 修正套件 2 或更高版本來升級，您應該繼續使用先前安裝的 ODBC 驅動程式

在 UNIX 平台上使用 Merant Informix ODBC 驅動程式

欲於 UNIX 平台上使用 Merant Informix ODBC 驅動程式，您必須執行下列其中一項動作：

- 在啓動 Starter Kit 之前，將 LANG 環境變數設為 en_US。例如，若是 Korn shell，請鍵入：

```
export LANG='en_US'
```

每次啓動 OLAP Starter Kit 時皆設定此變數。

- 若您的 LANG 環境變數已設定成不同值，請於安裝之後建立下列符號鏈結：

```
ln -s $ISHOME/locale/en_US $ISHOME/locale/$LANG
```

混合 OLAP 從屬站和伺服器的服務程式等級

IBM 建議您將 DB2 OLAP Starter Kit 的從屬站和伺服器元件保持在相同版本和修正套件層次。但在某些狀況下，您可混合從屬站和伺服器元件的服務程式等級：

使用同一版本但不同服務程式等級的從屬站和伺服器

IBM 不支援和建議同時使用較新的從屬站和較舊的伺服器。不過，雖然 IBM 不支援，您仍可同時使用較舊的從屬站和較新的伺服器。您可能會遭遇一些問題。例如：

- 來自伺服器的訊息可能不正確。您可升級從屬站的 message.MDB 檔案來符合伺服器的層次，即可解決此問題。
- 新的伺服器特性沒有作用。當您嘗試使用新特性時，從屬站、伺服器或兩者可能失效。
- 從屬站可能無法適當地連接伺服器。

在同一個版本中使用多重伺服器和單一從屬站

若您需要將一個從屬站連接至不同機器或作業系統上的幾個 OLAP 伺服器，IBM 建議您這些伺服器全部使用相同版本和服務程式等級。您的從屬站至少應該與最低層次的伺服器相同。若您遭遇問題，則可能需要使用不同從屬站機器來配合適當的主電腦，或將所有從屬站和伺服器升級成相同的服務程式等級。

混合不同版本的從屬站和伺服器

IBM 不支援同時使用 OLAP Starter Kit 從屬站和伺服器版本 7.1 與從屬站和伺服器版本 7.2。當 IBM OLAP 產品升級為較新版本層次時，網路更新和資料格式變更通常要求從屬站和伺服器為相同的版本層次。

混合 IBM 產品 (DB2 OLAP Starter Kit) 和 Hyperion 產品 (Hyperion Essbase 和 Hyperion Integration Server)

IBM 不支援混合 IBM 的 OLAP 從屬站和伺服器與 Hyperion Solutions 的 OLAP 從屬站和伺服器。雖然混合這些元件在某些情況下可以運作，但特性上的部份差異可能會造成問題。

OLAP 試算表增益集 EQD 檔案遺失

在 DB2 OLAP Starter Kit 中的試算表增益集具有 Query Designer (EQD) 元件。EQD 線上說明功能表中所含的**教學指導**按鈕，不會顯示任何內容。其所應顯示的教材為 *OLAP Spreadsheet Add-in User's Guide for Excel* 的第 2 章，以及 *OLAP Spreadsheet Add-in User's Guide for 1-2-3*。EQD 教學指導中的所有資訊，可以從「資訊中心」這些書籍的 HTML 版及 PDF 版查到。

資訊型錄管理程式管理手冊

資訊型錄管理程式起始設定公用程式

使用起始設定資訊型錄管理程式 (ICM) 公用程式，您現在可以利用下列指令，添加 SQL 陳述式至 CREATE TABLE 陳述式的結尾：

```
CREATEIC \DBTYPE dbtype \DGNAME dname \USERID userid \PASSWORD password
\KA1 userid \TABOPT "directory:\tabopt.file"
```

您可以從安裝 DB2 的目錄，在 CREATEIC 公用程式中指定 TABOPT 關鍵字。TABOPT 關鍵字之後的值是 *tabopt.file*，檔案名稱是含有完整的路徑。如果目錄名稱有空白，請以引號括住該名稱。*tabopt.file* 檔案的內容必須含有資訊，才能添加至 CREATE TABLE 陳述式。您可以使用下列任一種 SQL 陳述式寫入此 *tabopt.file* 檔案。ICM 公用程式將讀取此檔案，然後將它添加到 CREATE TABLE 陳述式。

表 20. SQL 陳述式

IN MYTABLESPACE	以其 MYTABLESPACE 資料建立表格
DATA CAPTURE CHANGES	以擴充格式建立表格並記載 SQL 變更
IN ACCOUNTING INDEX IN ACCOUNT_IDX	以其 ACCOUNTING 資料及 ACCOUNT_IDX 索引建立表格

內容檔案的最大值大小是 1000 單一位元組字元。

這個新功能只能在 Windows 及 UNIX 系統上使用。

版權問題

如果您得到了下列訊息：

```
FLG0083E: 您沒有「IBM 資訊型錄管理程式起始設定」公用程式的有效授權。
請洽詢您當地的軟體經銷商或 IBM 業務代表。
```

您必須購買 DB2 Warehouse Manager 或 IBM DB2 OLAP Server，並安裝「資訊型錄管理程式」元件，當中包含了「資訊型錄起始設定」公用程式。

安裝問題

若您在安裝了 DB2 Warehouse Manager 或 IBM DB2 OLAP Server 之後，又在相同的工作站上安裝其它「資訊型錄管理程式管理者」元件 (利用 DB2 Universal Database CD-ROM)，便會覆寫原有的資訊型錄起始設定公用程式。此時，您可以從 \sqllib\bin 目錄下，找出 createic.bak 與 flgnmocr.bak 檔案，將其分別更名爲 createic.exe 及 flgnmocr.exe。

若您要另外安裝 DB2 Universal Database 上的「資訊型錄管理程式」元件，則這些元件不能安裝在 Data Warehouse Manager 所安裝的同一部工作站上。您可以參閱「DB2 Warehouse Manager 安裝手冊」的第 3 章「安裝資訊型錄管理程式元件」，查看進一步的資訊。

以 DB2 版本 7 資訊型錄管理程式存取 DB2 版本 5 資訊型錄

由 DB2 版本 7 安裝程序所架構的 DB2 版本 7 資訊型錄管理程式次元件，支援存取儲存在 DB2 版本 6 和 DB2 版本 7 資料庫的資訊型錄。修改次元件的架構，存取儲存在 DB2 版本 5 資料庫的資訊型錄。DB2 版本 7 資訊型錄管理程式次成份並不支援存取 DB2 版本 2 或之前版本的資料。

設定資訊型錄管理者，存資訊型錄的資訊型錄使用者和資訊型錄起始設定公用程式，儲存在 DB2 版本 5 資料庫：

1. 在沒有安裝 DB2 版本 7 資訊型錄管理程式的工作站，安裝 DB2 Connect Enterprise Edition 版本 6。

DB2 Connect Enterprise Edition 是 DB2 Universal Database Enterprise Edition 和 DB2 Universal Database Enterprise - Extended Edition 的一部分。如果安裝這些 DB2 產品的版本 6，無須另外安裝 DB2 Connect。

限制: 在相同的 Windows NT 或 OS/2 工作站，不能安裝 DB2 多重版本。您可以在其它 Windows NT 工作站、OS/2 工作站或 UNIX 工作站上安裝 DB2 Connect。

2. 架構資訊型錄管理程式和 DB2 Connect 版本 6，存取 DB2 版本 5 的資料。詳細資訊，請參閱 *DB2 Connect 使用手冊*。以下為必要步驟的概觀：
 - a. 在 DB2 版本 5 系統，使用 DB2 命令行處理器編目版本 5 的資料庫，存取資訊型錄管理程式。
 - b. 在 DB2 Connect 系統，使用 DB2 命令行處理器進行編目：
 - DB2 版本 5 系統的 TCP/IP 節點
 - DB2 版本 5 系統的資料庫
 - DB2 版本 5 系統的 DCS 登錄項目
 - c. 在資訊型錄管理程式的工作站，使用 DB2 命令行處理器進行編目：
 - DB2 Connect 系統的 TCP/IP 節點
 - DB2 Connect 系統的資料庫

有關編目資料庫的資訊，請參閱 *DB2 Universal Database 安裝與架構補充*

3. 於資訊型錄管理程式的倉儲，將 DB2 CLI 套裝軟體和透過 DB2 Connect 所存取的各個資料庫進行連結。

下列 DB2 指令提供連結到 v5database 的範例，一個假設性的 DB2 版本 5 的資料庫。使用 DB2 命令行處理器發出下列指令。db2cli.lst 及 db2ajgrt 的位置在 \sqllib\bnd 目錄中。

```
db2 connect to v5database user 使用者 ID using 通行碼
db2 bind db2ajgrt.bnd
db2 bind @db2cli.lst blocking all grant public
```

使用者 ID 是 v5database 的使用者，通行碼是指使用者 ID 的通行碼。

db2cli.list 被連結到 DB2 版本 5 資料庫時，會發生錯誤。發生錯誤，是因為此架構並不支援大型物件。這項錯誤不會影響倉儲代理程式對 DB2 版本 5 資料庫的存取。

需要有 DB2 Universal Database 版本 5 的 FixPak 14 (於 2000 年 6 月發行)，才能透過 DB2 Connect 來存取 DB2 版本 5 的資料。請參照 FixPak 內的 APAR 編號 JR14507。

設置資訊型錄

在第 1 章第 1 段的步驟 2「設置資訊型錄」提到：

當您安裝了 DB2 Warehouse Manager 或 DB2 OLAP Server，
即會在 DB2 Universal Database for Windows NT 中建立預設的資訊型錄。

這個陳述式是不正確的。您必須定義新的資訊型錄。詳細資訊，請參閱「建立資訊型錄」部份。

與其它產品交換描述資料

在第 6 章「與其它產品交換描述資料」的「識別要公佈的 OLAP 物件」區段，在第二段有一個句子提到：

當您公佈 DB2 OLAP 整合伺服器描述資料時，
會在資訊型錄的「多重維度資料庫間的維度」與 OLAP
整合伺服器的表格物件間建立鏈結關係。

這段說明應如下所示：

當您公佈 DB2 OLAP 整合伺服器描述資料時，
會在資訊型錄「多重維度資料庫物件及表格物件中的維度」
之間建立鏈結關係。

這一句也出現在附錄 C「描述資料對映」的「資訊型錄管理程式與 OLAP 伺服器間的描述資料對映」區段。

使用 flgnxoln 指令交換描述資料

在第 6 章「交換描述資料」中有一個段落的標題是「識別要公佈的 OLAP 物件」。在段落的最後面有使用 flgnxoln 指令來公佈 OLAP 伺服器描述資料到資訊型錄的範例。此範例顯示 db2olap.ctl 和 db2olap.ff 檔案的目錄為 x:\Program Files\sqllib\logging，這是不正確的。目錄應該如第 87 頁所說：x:\Program Files\sqllib\exchange。

使用 MDISDGC 指令交換描述資料

第 6 章「與其它產品交換描述資料」的「將符合 MDIS 的描述資料轉換到標示語言檔」(第 89 頁)。您不可以從 MS-DOS 命令提示下發出 MDISDGC 指令，而應該從 DB2 命令視窗中發出該指令。該章節的第一句寫道：「將 MDIS 格式的描述資料轉換成標示語言檔」中亦提到您必從 MS-DOS 命令提示發出 DGMDISC 指令。您必須從 DB2 命令視窗中發出 DGMDISC 指令。

呼叫程式

「資訊型錄管理手冊」中部分的範例中的指令包含了 Program Files 的目錄名稱。當您呼叫一個包含 Program Files 目錄名稱的指令時，您必須將程式呼叫包在雙引號內。例如在附錄 B「預先定義的資訊型錄管理程式物件類型」的「以預先定義的物件類型起始設定您的資訊型錄」段落中有一個範例。如果您使用這裡的範例，當您從 DOS 提示符號執行它時會得到錯誤訊息。下列是正確的範例：

```
"X:\Program Files\SQLLIB\SAMPLES\SAMPDATA\DGWDEMO"  
/T userid password dname
```

Information Catalog Manager Programming Guide and Reference(無 中文版)

Information Catalog Manager Reason Codes

In Appendix D: Information Catalog Manager reason codes, some text might be truncated at the far right column for the following reason codes: 31014, 32727, 32728, 32729, 32730, 32735, 32736, 32737, 33000, 37507, 37511, and 39206. If the text is truncated, please see the HTML version of the book to view the complete column.

資訊型錄管理程式使用手冊

在第 2 章，有一個段落稱為「登記伺服器節點與遠端資訊型錄」。此區段列出了您在使用「資訊型錄管理程式」登記遠端資訊型錄時可從「DB2 控制中心」完成的步驟。這個章節的最後一個段落提到在從「DB2 控制中心」完成這些步驟之後 (新增系統、新增案例及新增資料庫)，在開啓「資訊型錄管理程式」之前必須關閉「控制中心」。這項資訊並不正確。在開啓「資訊型錄管理程式」之前並不需要關閉「控制中心」。

同樣的修正也適用於「登記伺服器節點及遠端資訊型錄」作業的線上說明，以及「登記伺服器節點及資訊型錄」視窗的線上說明。

資訊型錄管理程式：線上訊息

訊息 FLG0260E

訊息說明的第二句應為：

錯誤造成資訊型錄失敗的 rollback。
資訊型錄的狀況不穩定，但是，沒有製造任何的變更。

訊息 FLG0051E

訊息說明內的第二項目符號應為：

資訊型錄含有太多物件或物件類型。

管理者回應為：

使用匯入函數從目前資訊型錄刪除部份物件或物件類型。

訊息 FLG0003E

訊息說明應為：

資訊型錄必須在使用前進行登記。
資訊型錄可能登記錯誤。

訊息 FLG0372E

訊息說明的第一句應為：

ATTACHMENT-IND 值為物件所忽略，因為該物件是 Attachment 物件。

訊息 FLG0615E

訊息的第二句應為：

「資訊型錄管理程式」遇到非預期的資料庫錯誤，
或者在目前的目錄或路徑裡找不到連結檔。

資訊型錄管理程式：線上說明

資訊型錄視窗：選取的功能表開啓項目的線上說明錯誤地說「開啓選取的物件」。它應該說「開啓定義搜尋視窗」。

Web 資訊型錄管理程式

使用位於 DB2 UDB for OS/390 system 的資訊型錄，沒有不區分大小寫搜尋的功能。簡式搜尋和進階搜尋的情況都是如此。線上說明並沒有解譯，對於簡式搜尋，DB2 UDB for OS/390 資訊型錄的所有搜尋為區分大小寫的。此外，所有分組種類物件都可以擴充，即使它們不是基礎物件。

DB2 Warehouse Manager 安裝手冊

倉儲轉換程式的軟體需求

在您計劃要使用倉儲轉換程式的資料庫上，必須先安裝 Java Developer's Kit (JDK) 版本 1.1.8 (或以上)。

SAP R/3 Connector

從 SAP R/3 商業物件欄位對映直欄到 DB2 表格時，有些產生的直欄名稱可能會超過 30 字元。在此情況下，產生的直欄名稱只會反映 SAP 欄位名稱的前 30 個字。若產生的名稱並非您所要的，您可以使用「內容」筆記本變更表格。

安裝先決條件

設定 RFC_INI 環境變數。例如，Set RFC_INI=c:\rfcapi.ini。在設定變數之後，請您務必重新啟動電腦。

Web Connector

若您在執行 Web Connector 發生問題，IBM 服務程式可能會要求您傳送 Connector 的追蹤。

若要啓用 Web Connector 的追蹤，請將 Warehouse Center 代理程式的追蹤設定為大於 0。追蹤檔的名稱為 WSApiid.log，其中程序 ID 是代理程式的 Windows 程序 ID。追蹤檔會建立於 \sqllib\logging 目錄下。

安裝先決條件

請安裝 Java 執行環境 (JRE) 或 Java Virtual Machine (JVM) 版本 1.2.2 (或以上)，並將它設成您的預設。若要建立 JRE 版本作為您的預設，請將 1.2.2 JRE 的路徑新增到您的系統 PATH 變數中 (如 C:\JDKs\IBM\java12\bin;)。在變更預設 JRE 之後，請您務必重新啟動電腦。若您未安裝 Java，您可以從安裝 Data Warehouse Connector 的 CD 來安裝。

Query Patroller Administration Guide(無中文版)

DB2 Query Patroller Client is a Separate Component

The DB2 Query Patroller client is a separate component that is not part of the DB2 Administration client. This means that it is not installed during the installation of the DB2 Administration Client, as indicated in the Query Patroller Installation Guide. Instead, the Query Patroller client must be installed separately.

The version and level of the Query Patroller client and the Query Patroller server must be the same.

Migrating from Version 6 of DB2 Query Patroller Using `dqpmigrate`

The `dqpmigrate` command must be used if the Version 7 Query Patroller Server was installed over the Version 6 Query Patroller Server. For FixPak 2 or later, you do not have to run `dqpmigrate` manually as the installation of the FixPak runs this command for you. Without using this command, the existing users defined in v6 have no EXECUTE privileges on several new stored procedures added in Version 7.

註: `dqpmigrate.bnd` is found in the `sqllib/bnd` directory and `dqpmigrate.exe` is found in the `sqllib/bin` directory.

To use **`dqpmigrate`** manually to grant the EXECUTE privileges, perform the following after installing the FixPak:

1. Bind the `/sqllib/bnd/dqpmigrate.bnd` package file to the database where the Query Patroller server has been installed by entering the following command:

```
db2 bind dqpmigrate.bnd
```
2. Execute **`dqpmigrate`** by entering the following:

```
dqpmigrate dbalias userid passwd
```

Enabling Query Management

In the "Getting Started" chapter under "Enabling Query Management", the text should read:

You must be the owner of the data base, or you must have SYSADM, SYSCTRL, or SYSMAINT authority to set database configuration parameters.

Location of Table Space for Control Tables

In Chapter 1, System Overview, under DB2 Query Patroller Control Tables, the following text is to be added at the end of the section's first paragraph:

The table space for the DB2 Query Patroller control tables must reside in a single-node nodegroup, or DB2 Query Patroller will not function properly.

New Parameters for `dqpstart` Command

In Chapter 2, Getting Started, under Starting and Stopping DB2 Query Patroller, the following text is to be added following the last paragraph:

New Parameters for the **dqpstart** command:

RESTART parameter:

Allows the user to replace the host name and/or the node type of the specified node in the `dqpnodes.cfg` file. DB2 Query Patroller will be started on this node.

Note: Before running the `DQPSTART` command with the `RESTART` parameter, ensure the following:

1. DB2 Query Patroller is already stopped on the host that is going to be replaced.
2. DB2 Query Patroller is not already running on the new host.

The syntax is as follows:

```
dqpstart nodenum node_num restart hostname server | agent | none
```

ADDNODE parameter:

Allows the user to add a new node to the `dqpnodes.cfg` file. DB2 Query Patroller will be started on this node after the new node entry is added to the `dqpnodes.cfg` file. The syntax is as follows:

```
dqpstart nodenum node_num addnode hostname server | agent | none
```

DROPNODE parameter:

Allows the user to drop a node from the `dqnodes.cfg` file. DB2 Query Patroller will be stopped on this node before the node entry is dropped from the `dqpnodes.cfg` file. The syntax is as follows:

```
dqpstop nodenum node_num dropnode
```

New Parameter for `iwm_cmd` Command

A new **-v** parameter has been added to the **iwm_cmd** command to allow the user to recover the status of the jobs that were running on the node specified. Only jobs on an inactive node are allowed to be recovered. This command should be issued when there is a node failure and there are some jobs running on that node or being cancelled at the time. Jobs that were in "Running" state will be resubmitted and set back to "Queued" state. Jobs that were in "Cancelling" state will be set to "Cancelled" state.

The partial syntax is as follows:

```
►► iwm_cmd [ -u user_id ] [ -p password ] [ -v node_id_to_recover ]
```

node_id_to_recover

Specifies the node on which the jobs are to be recovered.

New Registry Variable: DQP_RECOVERY_INTERVAL

There is a new registry variable called `DQP_RECOVERY_INTERVAL` which is used to set the interval of time in minutes that the `iwm_scheduler` searches for recovery files. The default is 60 minutes.

Starting Query Administrator

In the "Using QueryAdministrator to Administer DB2 Query Patroller" chapter, instructions are provided for starting QueryAdministrator from the Start menu on Windows. The first step provides the following text:

If you are using Windows, you can select **DB2 Query Patroller** --> **QueryAdministrator** from the **IBM DB2** program group.

The text should read:

DB2 Query Patroller --> **QueryAdmin.**

User Administration

In the "User Administration" section of the "Using QueryAdministrator to Administer DB2 Query Patroller" chapter, the definition for the Maximum Elapsed Time parameter indicates that if the value is set to 0 or -1, the query will always run to completion. This parameter cannot be set to a negative value. The text should indicate that if the value is set to 0, the query will always run to completion.

The Max Queries parameter specifies the maximum number of jobs that the DB2 Query Patroller will run simultaneously. Max Queries must be an integer within the range of 0 to 32767.

Creating a Job Queue

In the "Job Queue Administration" section of the "Using QueryAdministrator to Administer DB2 Query Patroller" chapter, the screen capture in the steps for "Creating a Job Queue" should be displayed after the second step. The Information about new Job Queue window opens once you click **New** on the Job Queue Administration page of the QueryAdministrator tool.

References to the Job Queues page or the Job Queues tab should read Job Queue Administration page and Job Queue Administration tab, respectively.

Using the Command Line Interface

For a user with User authority on the DB2 Query Patroller system to submit a query and have a result table created, the user may require `CREATETAB` authority on the database. The user does not require `CREATETAB` authority on the database if the `DQP_RES_TBLSPC` profile variable is left unset, or if the `DQP_RES_TBLSPC` profile variable is set to the name of the default table space. The creation of the result tables will succeed in this case because users have the authority to create tables in the default table space.

Query Enabler Notes

- When using third-party query tools that use a keyset cursor, queries will not be intercepted. In order for Query Enabler to intercept these queries, you must modify the `db2cli.ini` file to include:

```
[common]
DisableKeySetCursor=1
```

- For AIX clients, please ensure that the environment variable `LIBPATH` is not set. Library `libXext.a`, shipped with the JDK, is not compatible with the library in the `/usr/lib/X11` subdirectory. This will cause problems with the Query Enabler GUI.

DB2 Query Patroller Tracker may Return a Blank Column Page

FixPak 3 includes a fix for the DB2 Query Patroller Tracker. The Tracker will now correctly report queries which hit no columns. An example of such a query is "SELECT COUNT(*) FROM ...". Since this kind of query does not hit any column in the table, the Tracker will present a blank page for the column page. This blank column page is not a defect.

Query Patroller and Replication Tools

Query Patroller Version 7 will intercept the queries of the replication tools (`asnapply`, `asnccp`, `djra` and `analyze`) and cause these tools to malfunction. A workaround is to disable dynamic query management when running these tools.

Appendix B. Troubleshooting DB2 Query Patroller Clients

In Appendix B, Troubleshooting DB2 Query Patroller Clients, section: Common Query Enabler Problems, problem #2, the text of the first bullet is replaced with:

Ensure that the path setting includes `jre`.

第6篇 應用程式開發

Administrative API Reference (無中文版)	289
db2ArchiveLog (new API)	289
db2ArchiveLog	290
db2ConvMonStream.	292
db2DatabasePing (new API)	292
db2DatabasePing - Ping Database	293
db2HistData	294
db2HistoryOpenScan	295
db2XaGetInfo (new API)	296
db2XaGetInfo - Get Information for Resource Manager	297
db2XaListIndTrans (new API that supercedes sqlxphqr)	297
db2XaListIndTrans - List Indoubt Transactions	298
db2GetSnapshot - Get Snapshot	301
Forget Log Record	301
sqlaintp - Get Error Message	302
sqlbctcq - Close Tablespace Container Query	302
sqlubkp - Backup Database	302
sqlureot - Reorganize Table	302
sqlurestore - Restore Database	302
Documentation Error Regarding AIX Extended Shared Memory Support (EXTSHM).	302
SQLFUPD	303
locklist.	303
SQLEDBDESC	303
SQLFUPD Documentation Error.	303
應用程式開發手冊	305
第 1 章 簡介	305
支援軟體	305
範例程式	307
第 3 章 開發 DB2 應用程式的一般資訊	307
開發檔案、make 檔和錯誤檢查的公用程式	307
第 4 章 開發 Java Applets 和應用程式	307
設定環境	307
OS/2 上的 JDK 層次	309
HP-UX 上的 Java2.	309
第 5 章 開發 SQL 程序	309
設定 SQL 程序環境	309
設定編譯器環境變數	310
自訂編譯指令	310
保留中間檔	314
備份和復置	314
建立 SQL 程序	315
呼叫儲存程序	315
分送已編譯的 SQL 程序	315
第 7 章 開發 HP-UX 應用程式	316
HP-UX C.	316
HP-UX C++.	316
第 9 章 開發 OS/2 應用程式	316
VisualAge C++ for OS/2 版本 4.0.	316
第 10 章 開發 PTX 應用程式	316
ptx/C++.	316
第 12 章 開發 Solaris 應用程式	317
SPARCompiler C++.	317
第 13 章 建置 Windows 32 位元作業系統的應用程 式	317
VisualAge C++ 版本 4.0.	317
Application Development Guide(無中文版)	319
Chapter 2. Coding a DB2 Application.	319
Activating the IBM DB2 Universal Database Project and Tool Add-ins for Microsoft Visual C++.	319
Chapter 6. Common DB2 Application Techniques	319
Generating Sequential Values	319
Controlling Sequence Behavior	320
Improving Performance with Sequence Objects	321
Comparing Sequence Objects and Identity Columns	322
Chapter 7. Stored Procedures	322
DECIMAL Type Fails in Linux Java Routines	322
Using Cursors in Recursive Stored Procedures	322
Writing OLE Automation Stored Procedures.	323
Chapter 12. Working with Complex Objects: User-Defined Structured Types	323
Inserting Structured Type Attributes Into Columns	323
Chapter 13. Using Large Objects (LOBs).	323
Large object (LOBs) support in federated database systems	323
How DB2 retrieves LOBs	324
How applications can use LOB locators	324
Restrictions on LOBs	324
Mappings between LOB and non-LOB data types	325
Tuning the system	325
Part 5. DB2 Programming Considerations.	325
IBM DB2 OLE DB Provider	325
Chapter 20. Programming in C and C++.	325
C/C++ Types for Stored Procedures, Functions, and Methods	326
Chapter 21. Programming in Java	327
Java Method Signature in PARAMETER STYLE JAVA Procedures and Functions	327
Connecting to the JDBC Applet Server	327
Appendix B. Sample Programs	328
CLI Guide and Reference(無中文版)	329
Binding Database Utilities Using the Run-Time Client	329
Using Static SQL in CLI Applications	329
Limitations of JDBC/ODBC/CLI Static Profiling	329

ADT Transforms	330	DECRYPT_BIN and DECRYPT_CHAR	350
Chapter 3. Using Advanced Features	330	ENCRYPT	351
Writing Multi-Threaded Applications	330	GETHINT.	353
Scrollable Cursors	331	IDENTITY_VAL_LOCAL.	354
Server-side Scrollable Cursor Support for		LCASE and UCASE (Unicode).	359
OS/390.	331	MQPUBLISH	359
Using Compound SQL.	332	MQREAD.	360
Using Stored Procedures	332	MQRECEIVE	361
Writing a Stored Procedure in CLI	332	MQSEND.	363
CLI Stored Procedures and Autobinding	333	MQSUBSCRIBE.	364
Chapter 4. Configuring CLI/ODBC and Running		MQUNSUBSCRIBE	365
Sample Applications	333	MULTIPLY_ALT	367
Configuration Keywords	333	REC2XML	368
Chapter 5. DB2 CLI Functions	333	ROUND	372
SQLBindFileToParam - Bind LOB File Reference		WEEK_ISO	374
to LOB Parameter	333	Table Functions	374
SQLNextResult - Associate Next Result Set with		MQREADALL	374
Another Statement Handle	333	MQRECEIVEALL	376
Purpose	334	Procedures	378
Syntax	334	GET_ROUTINE_SAR	379
Function Arguments	334	PUT_ROUTINE_SAR	379
Usage	334	Chapter 5. Queries	381
Return Codes.	335	select-statement/syntax diagram	381
Diagnostics	335	select-statement/fetch-first-clause	381
Restrictions	335	Chapter 6. SQL Statements	382
References	335	Update of the Partitioning Key Now Supported	382
Appendix D. Extended Scalar Functions	336	Statement: ALTER TABLE	382
Date and Time Functions.	336	Statement: CREATE TABLE.	382
Appendix K. Using the DB2 CLI/ODBC/JDBC Trace		Statement: DECLARE GLOBAL TEMPORARY	
Facility.	336	TABLE PARTITIONING KEY	
訊息參考手冊	337	(column-name,...)	382
取得訊息與 SQLSTATE 說明	337	Statement: UPDATE	382
在 DB2 Connect 變更的 SQLCODE 重新對映.	337	Larger Index Keys for Unicode Databases	382
新的與變更的訊息	337	ALTER TABLE.	382
呼叫層次介面 (CLI) 訊息	337	CREATE INDEX	382
DB2 訊息.	337	CREATE TABLE	383
DBI 訊息.	338	ALTER SEQUENCE	383
資料倉儲中心 (DWC) 訊息	338	ALTER SEQUENCE	384
SQL 訊息.	339	ALTER TABLE.	387
更正 SQLSTATES	341	Compound SQL (Embedded).	390
SQL Reference(無中文版)	343	Compound Statement (Dynamic)	390
SQL Reference is Provided in One PDF File	343	Compound Statement (Dynamic)	391
Chapter 3. Language Elements	343	CREATE FUNCTION (Source or Template).	393
Naming Conventions and Implicit Object Name		CREATE FUNCTION (SQL Scalar, Table or	
Qualifications.	343	Row)	394
DATALINK Assignments	343	CREATE METHOD	395
Expressions	343	CREATE METHOD	396
Syntax Diagram	343	CREATE SEQUENCE.	396
OLAP Functions.	344	CREATE SEQUENCE.	397
Sequence Reference.	346	CREATE TRIGGER	400
Chapter 4. Functions	349	CREATE TRIGGER	401
Enabling the New Functions and Procedures.	349	CREATE WRAPPER	403
Scalar Functions.	350	DECLARE CURSOR	403
ABS or ABSVAL	350	DELETE	403
		DROP	404
		GRANT (Sequence Privileges)	405

GRANT (Sequence Privileges)	406	Predicates	435
INSERT	406	Chapter 4 Functions	435
SELECT INTO	407	Scalar Functions.	435
SET ENCRYPTION PASSWORD	408	CLI Guide and Reference(無中文版)	435
SET ENCRYPTION PASSWORD	409	Chapter 3. Using Advanced Features	435
SET transition-variable.	409	Writing a DB2 CLI Unicode Application.	435
SET Variable.	410	Appendix C. DB2 CLI and ODBC.	440
UPDATE	412	ODBC Unicode Applications.	440
Chapter 7. SQL Procedures now called Chapter 7.		Data Movement Utilities Guide and Reference(無中文 版)	442
SQL Control Statements	413	Appendix C. Export/Import/Load Utility File Formats	442
SQL Procedure Statement.	413		
SQL Procedure Statement.	414		
FOR	415		
FOR	416		
Compound Statement changes to Compound Statement (Procedure)	417		
RETURN	418		
RETURN	419		
SIGNAL	420		
SIGNAL	421		
Appendix A. SQL Limits.	423		
Appendix D. Catalog Views	423		
SYSCAT.SEQUENCES	423		
IBM DB2 儲存程序建置器	425		
Java 1.2 支援儲存程序建置器	425		
DB2 儲存程序的遠端除錯	425		
在 Windows、OS/2 或 UNIX 平台上開發 SQL 程序	426		
在 Solaris 平台使用 DB2 儲存程序建置器	426		
已知問題與限制.	426		
在繁體中文語言環境下使用 DB2 儲存程序建置器	426		
UNIX (AIX、Sun Solaris、Linux) 安裝與儲存程序建 置器	427		
建置 OS/390 的 SQL 儲存程序	427		
除錯 SQL 儲存程序	427		
匯出 Java 儲存程序	427		
在 OS/390 上插入儲存程序	427		
在工作站伺服器中設定 SQL 儲存程序的建置選項	428		
自動復新 OS/390 上儲存程序建置的 WLM 位址空 間	428		
在 OS/390 開發 Java 儲存程序	428		
為 MQ Series 與 OLE DB 建置 DB2 表格使用者定 義功能 (UDF)	428		
Unicode 更新	429		
簡介	429		
DB2 Unicode 資料庫及應用程式	429		
文件更新	431		
SQL Reference(無中文版)	431		
Chapter 3 Language Elements	431		
Promotion of Data Types.	431		
Casting Between Data Types.	432		
Assignments and Comparisons	433		
Rules for Result Data Types.	434		
Rules for String Conversions.	434		
Expressions	435		

Administrative API Reference (無中文版)

| **db2ArchiveLog (new API)**
|

db2ArchiveLog - Archive Active Log

db2ArchiveLog

Closes and truncates the active log file for a recoverable database. If user exit is enabled, issues an archive request.

Authorization

One of the following:

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

Required Connection

This API automatically establishes a connection to the specified database. If a connection to the specified database already exists, the API will return an error.

Version

db2ApiDf.h

C API Syntax

```
/* File: db2ApiDf.h */
/* API: Archive Active Log */
SQL_API_RC SQL_API_FN
db2ArchiveLog (
    db2UInt32 version,
    void *pDB2ArchiveLogStruct,
    struct sqlca * pSqlca);

typedef struct
{
    char          *piDatabaseAlias;
    char          *piUserName;
    char          *piPassword;
    db2UInt16     iAllNodeFlag;
    db2UInt16     iNumNodes;
    SQL_PDB_NODE_TYPE *piNodeList;
    db2UInt32     iOptions;
}
```

Generic API Syntax

```

/* File: db2ApiDf.h */
/* API: Archive Active Log */
SQL_API_RC SQL_API_FN
db2gArchiveLog (
    db2UInt32 version,
    void *pDB2ArchiveLogStruct,
    struct sqlca * pSqlca);

typedef struct
{
    db2UInt32          iAliasLen;
    db2UInt32          iUserNameLen;
    db2UInt32          iPasswordLen;
    char               *piDatabaseAlias;
    char               *piUserName;
    char               *piPassword;
    db2UInt16          iAllNodeFlag;
    db2UInt16          iNumNodes;
    SQL_PDB_NODE_TYPE *piNodeList;
    db2UInt32          iOptions;
}

```

API Parameters**version**

Input. Specifies the version and release level of the variable passed in as the second parameter, *pDB2ArchiveLogStruct*.

pDB2ArchiveLogStruct

Input. A pointer to the *db2ArchiveLogStruct* structure.

pSqlca

Output. A pointer to the *sqlca* structure.

iAliasLen

Input. A 4-byte unsigned integer representing the length in bytes of the database alias.

iUserNameLen

A 4-byte unsigned integer representing the length in bytes of the user name. Set to zero if no user name is used.

iPasswordLen

Input. A 4-byte unsigned integer representing the length in bytes of the password. Set to zero if no password is used.

piDatabaseAlias

Input. A string containing the database alias (as cataloged in the system database directory) of the database for which the active log is to be archived.

piUserName

Input. A string containing the user name to be used when attempting a connection.

piPassword

Input. A string containing the password to be user when attempting a connection.

db2ArchiveLog - Archive Active Log

iAllNodeFlag

MPP only. Input. Flag indicating whether the operation should apply to all nodes listed in the `db2nodes.cfg` file. Valid values are:

DB2ARCHIVELOG_NODE_LIST

Apply to nodes in a node list that is passed in *piNodeList*.

DB2ARCHIVELOG_ALL_NODES

Apply to all nodes. *piNodeList* should be NULL. This is the default value.

DB2ARCHIVELOG_ALL_EXCEPT

Apply to all nodes except those in the node list passed in *piNodeList*.

iNumNodes

MPP only. Input. Specifies the number of nodes in the *piNodeList* array.

piNodeList

MPP only. Input. A pointer to an array of node numbers against which to apply the archive log operation.

iOptions

Input. Reserved for future use.

db2ConvMonStream

In the Usage Notes, the structure for the snapshot variable datastream type `SQLM_ELM_SUBSECTION` should be `sqlm_subsection`.

db2DatabasePing (new API)

db2DatabasePing - Ping Database

Tests the network response time of the underlying connectivity between a client and a database server. This API can be used by an application when a host database server is accessed via DB2 Connect either directly or through a gateway.

Authorization

None

Required Connection

Database

Version

db2ApiDf.h

C API Syntax

```

/* File: db2ApiDf.h */
/* API: Ping Database */
/* ... */

SQL_API_RC SQL_API_FN
db2DatabasePing (
    db2UInt32    versionNumber,
    void        *pParmStruct,
    struct sqlca *pSqlca);
/* ... */

typedef SQL_STRUCTURE db2DatabasePingStruct
{
    char        iDbAlias[SQL_ALIAS_SZ + 1];
    db2UInt16   iNumIterations;
    db2UInt32   *poElapsedTime;
}

```

Generic API Syntax

```

/* File: db2ApiDf.h */
/* API: Ping Database */
/* ... */
SQL_API_RC SQL_API_FN
db2gDatabasePing (
    db2UInt32    versionNumber,
    void        *pParmStruct,
    struct sqlca *pSqlca);
/* ... */

typedef SQL_STRUCTURE db2gDatabasePingStruct
{
    db2UInt16    iDbAliasLength;
    char        iDbAlias[SQL_ALIAS_SZ];
    db2UInt16    iNumIterations;
    db2UInt32    *poElapsedTime;
}

```

API Parameters**versionNumber**

Input. Version and release of the DB2 Universal Database or DB2 Connect product that the application is using.

db2ArchiveLog - Archive Active Log

Note: Constant `db2Version710` or higher should be used for DB2 Version 7.1 or higher.

iDbAliasLength

Input. Length of the database alias name.

註: This parameter is not currently used. It is reserved for future use.

iDbAlias

Input. Database alias name.

註: This parameter is not currently used. It is reserved for future use.

iNumIterations

Input. Number of test request iterations. The value must be between 1 and 32767 inclusive.

poElapsedTime

Output. A pointer to an array of 32-bit integers where the number of elements is equal to `iNumIterations`. Each element in the array will contain the elapsed time in microseconds for one test request iteration.

Note: The application is responsible for allocating the memory for this array prior to calling this API.

pSqlca

Output. A pointer to the `sqlca` structure. For more information about this structure, see the *Administrative API Reference*.

Usage Notes

A database connection must exist before invoking this API, otherwise an error will result.

This function can also be invoked using the PING command. For a description of this command, see the *Command Reference*.

db2HistData

The following entries should be added to Table 11. Fields in the `db2HistData` Structure:

Field Name	Data Type	Description
<code>oOperation</code>	char	See Table 12.
<code>oOptype</code>	char	See Table 13.

The following table will be added following Table 11.

Table 12. Valid event values for `oOperation` in the `db2HistData` Structure

Value	Description	C Definition	COBOL/FORTRAN Definition
A	add tablespace	DB2HISTORY_OP_ADD_TABLESPACE	DB2HIST_OP_ADD_TABLESPACE
B	backup	DB2HISTORY_OP_BACKUP	DB2HIST_OP_BACKUP
C	load-copy	DB2HISTORY_OP_LOAD_COPY	DB2HIST_OP_LOAD_COPY
D	dropped table	DB2HISTORY_OP_DROPPED_TABLE	DB2HIST_OP_DROPPED_TABLE

db2ArchiveLog - Archive Active Log

Value	Description	C Definition	COBOL/FORTRAN Definition
F	roll forward	DB2HISTORY_OP_ROLLFWD	DB2HIST_OP_ROLLFWD
G	reorganize table	DB2HISTORY_OP_REORG	DB2HIST_OP_REORG
L	load	DB2HISTORY_OP_LOAD	DB2HIST_OP_LOAD
N	rename tablespace	DB2HISTORY_OP_REN_TABLESPACE	DB2HIST_OP_REN_TABLESPACE
O	drop tablespace	DB2HISTORY_OP_DROP_TABLESPACE	DB2HIST_OP_DROP_TABLESPACE
Q	quiesce	DB2HISTORY_OP_QUIESCE	DB2HIST_OP_QUIESCE
R	restore	DB2HISTORY_OP_RESTORE	DB2HIST_OP_RESTORE
S	run statistics	DB2HISTORY_OP_RUNSTATS	DB2HIST_OP_RUNSTATS
T	alter tablespace	DB2HISTORY_OP_ALT_TABLESPACE	DB2HIST_OP_ALT_TBS
U	unload	DB2HISTORY_OP_UNLOAD	DB2HIST_OP_UNLOAD

The following table will also be added.

Table 13. Valid oOtype values db2HistData Structure

oOperation	oOtype	Description	C/COBOL/FORTRAN Definition
B	F	Offline	DB2HISTORY_OPTYPE_OFFLINE
	N	Online	DB2HISTORY_OPTYPE_ONLINE
	I	Incremental offline	DB2HISTORY_OPTYPE_INCR_OFFLINE
	O	Incremental online	DB2HISTORY_OPTYPE_INCR_ONLINE
	D	Delta offline	DB2HISTORY_OPTYPE_DELTA_OFFLINE
	E	Delta online	DB2HISTORY_OPTYPE_DELTA_ONLIN
F	E	End of log	DB2HISTORY_OPTYPE_EOL
	P	Point in time	DB2HISTORY_OPTYPE_PIT
L	I	Insert	DB2HISTORY_OPTYPE_INSERT
	R	Replace	DB2HISTORY_OPTYPE_REPLACE
Q	S	Quiesce share	DB2HISTORY_OPTYPE_SHARE
	U	Quiesce update	DB2HISTORY_OPTYPE_UPDATE
	X	Quiesce exclusive	DB2HISTORY_OPTYPE_EXCL
	Z	Quiesce reset	DB2HISTORY_OPTYPE_RESET
R	F	Offline	DB2HISTORY_OPTYPE_OFFLINE
	N	Online	DB2HISTORY_OPTYPE_ONLINE
	I	Incremental offline	DB2HISTORY_OPTYPE_INCR_OFFLINE
	O	Incremental online	DB2HISTORY_OPTYPE_INCR_ONLINE
T	C	Add containers	DB2HISTORY_OPTYPE_ADD_CONT
	R	Rebalance	DB2HISTORY_OPTYPE_REB

db2HistoryOpenScan

The following value will be added to the **iCallerAction** parameter.

db2ArchiveLog - Archive Active Log

DB2HISTORY_LIST_CRT_TABLESPACE

Select only the CREATE TABLESPACE and DROP TABLESPACE records that pass the other filters.

db2XaGetInfo (new API)

db2XaGetInfo - Get Information for Resource Manager

Extracts information for a particular resource manager once an xa_open call has been made.

Authorization

None

Required Connection

Database

Version

sqlxa.h

C API Syntax

```

/* File: sqlxa.h */
/* API: Get Information for Resource Manager */
/* ... */
SQL_API_RC SQL_API_FN
db2XaGetInfo (
    db2Uint32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2XaGetInfoStruct
{
    db2int32 iRmid;
    struct sqlca oLastSqlca;
} db2XaGetInfoStruct;

```

API Parameters**versionNumber**

Input. Specifies the version and release level of the structure passed in as the second parameter, *pParmStruct*.

pParmStruct

Input. A pointer to the *db2XaGetInfoStruct* structure.

pSqlca

Output. A pointer to the *sqlca* structure. For more information about this structure, see the *Administrative API Reference*.

iRmid Input. Specifies the resource manager for which information is required.

oLastSqlca

Output. Contains the *sqlca* for the last XA API call.

註: Only the *sqlca* that resulted from the last failing XA API can be retrieved.

db2XaListIndTrans (new API that supercedes sqlxphqr)

db2ArchiveLog - Archive Active Log

db2XaListIndTrans - List Indoubt Transactions

Provides a list of all indoubt transactions for the currently connected database.

Scope

This API affects only the node on which it is issued.

Authorization

One of the following:

- *sysadm*
- *dbadm*

Required Connection

Database

Version

db2ApiDf.h

C API Syntax

```
/* File: db2ApiDf.h */
/* API: List Indoubt Transactions */
/* ... */
SQL_API_RC SQL_API_FN
db2XaListIndTrans (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2XaListIndTransStruct
{
    db2XaRecoverStruct * piIndoubtData;
    db2UInt32            iIndoubtDataLen;
    db2UInt32            oNumIndoubtsReturned;
    db2UInt32            oNumIndoubtsTotal;
    db2UInt32            oReqBufferLen;
} db2XaListIndTransStruct;

typedef SQL_STRUCTURE db2XaRecoverStruct
{
    sqluint32            timestamp;
    SQLXA_XID            xid;
    char                  dbalias[SQLXA_DBNAME_SZ];
    char                  applid[SQLXA_APPLID_SZ];
    char                  sequence_no[SQLXA_SEQ_SZ];
    char                  auth_id[SQL_USERID_SZ];
    char                  log_full;
    char                  connected;
    char                  indoubt_status;
    char                  originator;
    char                  reserved[8];
} db2XaRecoverStruct;
```

API Parameters**versionNumber**

Input. Specifies the version and release level of the structure passed in as the second parameter, *pParmStruct*.

pParmStruct

Input. A pointer to the *db2XaListIndTransStruct* structure.

pSqlca

Output. A pointer to the *sqlca* structure. For more information about this structure, see the *Administrative API Reference*.

piIndoubtData

Input. A pointer to the application supplied buffer where indoubt data will be returned. The indoubt data is in *db2XaRecoverStruct* format. The application can traverse the list of indoubt transactions by using the size of the *db2XaRecoverStruct* structure, starting at the address provided by this parameter.

If the value is NULL, DB2 will calculate the size of the buffer required and return this value in *oReqBufferLen*. *oNumIndoubtsTotal* will contain the total number of indoubt transactions. The application may allocate the required buffer size and issue the API again.

oNumIndoubtsReturned

Output. The number of indoubt transaction records returned in the buffer specified by *pIndoubtData*.

oNumIndoubtsTotal

Output. The Total number of indoubt transaction records available at the time of API invocation. If the *piIndoubtData* buffer is too small to contain all the records, *oNumIndoubtsTotal* will be greater than the total for *oNumIndoubtsReturned*. The application may reissue the API in order to obtain all records.

註: This number may change between API invocations as a result of automatic or heuristic indoubt transaction resynchronisation, or as a result of other transactions entering the indoubt state.

oReqBufferLen

Output. Required buffer length to hold all indoubt transaction records at the time of API invocation. The application can use this value to determine the required buffer size by calling the API with *pIndoubtData* set to NULL. This value can then be used to allocate the required buffer, and the API can be issued with *pIndoubtData* set to the address of the allocated buffer.

註: The required buffer size may change between API invocations as a result of automatic or heuristic indoubt transaction resynchronisation, or as a result of other transactions entering the indoubt state. The application may allocate a larger buffer to account for this.

timestamp

Output. Specifies the time when the transaction entered the indoubt state.

xid

Output. Specifies the XA identifier assigned by the transaction manager to uniquely identify a global transaction.

db2ArchiveLog - Archive Active Log

dbalias

Output. Specifies the alias of the database where the indoubt transaction is found.

applid Output. Specifies the application identifier assigned by the database manager for this transaction.

sequence_no

Output. Specifies the sequence number assigned by the database manager as an extension to the *applid*.

auth_id

Output. Specifies the authorization ID of the user who ran the transaction.

log_full

Output. Indicates whether or not this transaction caused a log full condition. Valid values are:

SQLXA_TRUE

This indoubt transaction caused a log full condition.

SQLXA_FALSE

This indoubt transaction did not cause a log full condition.

connected

Output. Indicates whether or not the application is connected. Valid values are:

SQLXA_TRUE

The transaction is undergoing normal syncpoint processing, and is waiting for the second phase of the two-phase commit.

SQLXA_FALSE

The transaction was left indoubt by an earlier failure, and is now waiting for resynchronisation from the transaction manager.

indoubt_status

Output. Indicates the status of this indoubt transaction. Valid values are:

SQLXA_TS_PREP

The transaction is prepared. The *connected* parameter can be used to determine whether the transaction is waiting for the second phase of normal commit processing or whether an error occurred and resynchronisation with the transaction manager is required.

SQLXA_TS_HCOM

The transaction has been heuristically committed.

SQLXA_TS_HROL

The transaction has been heuristically rolled back.

SQLXA_TS_MACK

The transaction is missing commit acknowledgement from a node in a partitioned database.

SQLXA_TS_END

The transaction has ended at this database. This transaction may be re-activated, committed, or rolled back at a later time. It is also possible that the transaction manager encountered an error and the transaction will

db2ArchiveLog - Archive Active Log

not be completed. If this is the case, this transaction requires heuristic actions, because it may be holding locks and preventing other applications from accessing data.

Usage Notes

A typical application will perform the following steps after setting the current connection to the database or to the partitioned database coordinator node:

1. Call **db2XaListIndTrans** with *piIndoubtData* set to NULL. This will return values in *oReqBufferLen* and *oNumIndoubtsTotal*.
2. Use the returned value in *oReqBufferLen* to allocate a buffer. This buffer may not be large enough if there are additional indoubt transactions because the initial invocation of this API to obtain *oReqBufferLen*. The application may provide a buffer larger than *oReqBufferLen*.
3. Determine if all indoubt transaction records have been obtained. This can be done by comparing *oNumIndoubtsReturned* to *oNumIndoubtTotal*. If *oNumIndoubtsTotal* is greater than *oNumIndoubtsReturned*, the application can repeat the above steps.

See Also

"sqlxhfrg - Forget Transaction Status", "sqlxphcm - Commit an Indoubt Transaction", and "sqlxphrl - Roll Back an Indoubt Transaction" in the *Administrative API Reference*.

db2GetSnapshot - Get Snapshot

The syntax for the db2GetSnapshot API should be as follows:

```
int db2GetSnapshot( unsigned char version;
                  db2GetSnapshotData *data,
                  struct sqlca *sqlca);
```

The parameters described in data are:

```
typedef struct db2GetSnapshotData{
    sqlma *piSqlmaData;
    sqlm_collected *poCollectedData
    void *poBuffer;
    db2uint32 iVersion;
    db2int32 iBufferSize;
    db2uint8 iStoreResult;
    db2uint16 iNodeNumber;
    db2uint32 *poOutputFormat;
}db2GetSnapshotData;
```

Forget Log Record

The following information will be added to Appendix F following the MPP Subordinator Prepare section.

This log record is written after a rollback of indoubt transactions or after a commit of two-phase commit. The log record is written to mark the end of the transaction and releases any log resources held. In order for the transaction to be forgotten, it must be in a heuristically completed state.

db2ArchiveLog - Archive Active Log

Table 21. Forget Log Record Structure

Description	Type	Offset (Bytes)
Log header	LogManagerLogRecordHeader	0(20)
time	sqluint64	20(8)
<i>Total Length: 28 bytes</i>		

sqlaintp - Get Error Message

The following usage note is to be added to the description of this API:

In a multi-threaded application, sqlaintp must be attached to a valid context; otherwise, the message text for SQLCODE -1445 cannot be obtained.

sqlbctcq - Close Tablespace Container Query

Load is not a valid Authorization level for this API.

sqlubkp - Backup Database

For the **BackupType** parameter the SQLUB_FULL value will be replaced by the SQLUB_DB. A backup of all tablespaces in the database will be taken.

To support the new incremental backup functionality the SQLUB_INCREMENTAL and SQLUB_DELTA parameters will also be added. An incremental backup image is a copy of all database data which has changed since the most recent successful, full backup. A delta backup image is a copy of all database data that has changed since the most recent successful backup of any type

sqlureot - Reorganize Table

The following sentence will be added to the Usage Notes:

REORGANIZE TABLE cannot use an index that is based on an index extension.

sqlurestore - Restore Database

For the **RestoreType** parameter the SQLUD_FULL value will be replaced by the SQLUD_DB. A restore of all table spaces in the database will be taken. This will be run offline.

To support the new incremental restore functionality the SQLUD_INCREMENTAL parameter will also be added.

An incremental backup image is a copy of all database data which has changed since the most recent successful full backup.

Documentation Error Regarding AIX Extended Shared Memory Support (EXTSHM)

In "Appendix E. Threaded Applications with Concurrent Access", Note 2 should now read:

db2ArchiveLog - Archive Active Log

2. By default, AIX does not permit 32-bit applications to attach to more than 11 shared memory segments per process, of which a maximum of 10 can be used for local DB2 connections.

To use EXTSHM with DB2, do the following:

In client sessions:

```
export EXTSHM=ON
```

When starting DB2:

```
export EXTSHM=ON
db2set DB2ENVLIST=EXTSHM
db2start
```

On EEE, also add the following lines to sqllib/db2profile:

```
EXTSHM=ON
export EXTSHM
```

SQLFUPD

locklist

The name of the token has changed from SQLF_DBTN_LOCKLIST to SQLF_DBTN_LOCK_LIST. The locklist parameter has been changed from a SMALLINT to a 64-bit unsigned INTEGER. The following addition should be made to the table of Updatable Database Configuration Parameters.

Parameter Name	Token	Token Value	Data Type
locklist	SQLF_DBTN_LOCK_LIST	704	Uint64

The new maximum for this parameter is 524 288.

SQLLEDBDESC

Two values will be added to the list of valid values for SQLDBCSS (defined in sqlenv). They are:

SQL_CS_SYSTEM-NLSCHAR

Collating sequence from system using the NLS version of compare routines for character types.

SQL_CS_USER-NLSCHAR

Collating sequence from user using the NLS version of compare routines for character types.

SQLFUPD Documentation Error

In 『Chapter 3. Data Structures』, Table 53. Updatable Database Configuration Parameters incorrectly lists the token value for dbheap as 701. The correct value is 58.

db2ArchiveLog - Archive Active Log

應用程式開發手冊

第 1 章 簡介

支援軟體

註: **PHP**。PHP 可使用為從 Web 型應用程式存取 DB2 的方法。PHP 是一種伺服器端、內嵌 HTML 的交互式平台 Script 語言。它支援使用 Unified-ODBC 存取方法的 DB2 存取，其中使用者層次 PHP 使用 ODBC 呼叫與 DB2 通信。與標準 ODBC 不同的是，Unified-ODBC 方法直接與 DB2 CLI 層通信，而不透過 ODBC 層。有關 PHP 搭配使用 DB2 的詳細資訊，請參考下列的 DB2 支援網站：
www.ibm.com/software/data/db2/udb/winos2unix/support。

AIX

列示的 C 及 C++ 編譯器的版本應為下列：

IBM C 及 C++ 編譯器，適用 AIX 版本 3.6.6 (適用於 64 位元的版本 3.6.6.3)
IBM C for AIX 4.4
IBM VisualAge C++ 版本 4.0

註: 請從下列網址下載這些編譯器版本的最新 FixPak

<http://www.ibm.com/software/ad/vacpp/service/csd.html>

Micro Focus COBOL 編譯器所列示的版本應為：

AIX 4.2.1

Micro Focus COBOL 版本 4.0.20 (PRN 12.03 或更新的版本)
Micro Focus COBOL 版本 4.1.10 (PRN 13.04 或更新的版本)

AIX 4.3

Micro Focus COBOL Server Express 版本 1.0

註: 有關 Micro Focus COBOL 儲存程序及 AIX 4.3 上的 UDF 的 DB2 支援資訊，請造訪 DB2 Application Development 網頁：

<http://www.ibm.com/software/data/db2/udb/ad>

若要使用 IBM XL Fortran for AIX 版本 5.1.0 編譯器來開發 64 位元應用程式，請在編譯及鏈結步驟中使用 "-q64" 選項。請注意此編譯器在先前版本時並不支援 64 位元應用程式。

HP-UX

C++ 編譯器所列示的版本為：

HP aC++，版本 A.03.25

註: HP 不支援以新、舊編譯器來編譯的物件之間的二進位相容性，因此這將迫使建立 C++ 應用程式的重新編譯，以存取 HP-UX 上的 DB2。也必須建立 C++ 應用程式以處理這個新編譯器的異常狀況。

這是 aCC 轉移指南的 URL：<http://www.hp.com/esy/lang/cpp/tguide>。C++ 的不相容部份可在下列網站中找到：

<http://www.hp.com/esy/lang/cpp/tguide/transcontent.html#RN.CVT.1.2>
<http://www.hp.com/esy/lang/cpp/tguide/transcontent.html#RN.CVT.3.3>

C 與 C++ 部份可在下列網站中找到：

<http://www.hp.com/esy/lang/cpp/tguide/transcontent.html#RN.CVT.3.3.1>

雖然 C 和 aCC 相容，當使用兩種不同的物件類型時，含有 "main" 的物件必須以 aCC 編譯，而最終可執行檔必須以 aCC 來鏈結。

Linux

DB2 for Linux 支援下列 REXX 版本：

Object REXX Interpreter for Linux 版本 2.1

Linux/390

DB2 for Linux/390 只支援 Java、C 及 C++。

OS/2

C/C++ 編譯器所列示的版本為：

IBM VisualAge C++ for OS/2 版本 3.6.5 及版本 4.0

註：請從下列網址下載這些編譯器版本的最新 FixPak

<http://www.ibm.com/software/ad/vacpp/service/csd.html>

有關這些 VisualAge C++ 編譯器的未來服務支援限制，請參閱下列網址的新聞區段：

<http://www-4.ibm.com/software/ad/vacpp/>

Solaris

Micro Focus COBOL 編譯器所列示的版本應為：

Micro Focus COBOL Server Express 版本 1.0

Windows 32 位元作業系統

IBM VisualAge C++ 編譯器所列示的版本為：

IBM VisualAge C++ 適用於 Windows 版本 3.6.5 和 4.0

註：請從下列網址下載這些編譯器版本的最新 FixPak

<http://www.ibm.com/software/ad/vacpp/service/csd.html>

有關這些 VisualAge C++ 編譯器的未來服務支援限制，請參閱下列網址的新聞區段：

<http://www-4.ibm.com/software/ad/vacpp/>

Micro Focus COBOL 編譯器所列示的版本應為：

Micro Focus COBOL 版本 4.0.20

Micro Focus COBOL Net Express 版本 3.0

範例程式

下列內容應該新增到「物件連結和內嵌範例」一節中：

salarycltvc

Visual C++ DB2 CLI 範例稱 Visual Basic 儲存程序為 salarysrv。

SALSVADO

範例 OLE 自動化儲存程序 (SALSVADO) 和 SALCLADO 從屬站，在 32 位元的 Visual Basic 和 ADO 下執行，計算表格 staff2 的平均薪資。

下列內容應該新增到「日誌管理使用者跳出程式範例」一節中：

於 AIX 使用 3.1.6 或更高層次的 ADSM API 從屬站之應用程式必須由 xlc_r 或 xIC_r 編譯器呼叫來建立，而非由 xlc 或 xIC 來建立，即使該應用程式為單一執行緒。此確保檔案庫為執行緒的安全。這適用於「日誌管理使用者跳出程式範例」，db2uext2.cadsm。

如果您的應用程式是由非執行緒安全檔案庫來編譯，您可以引用固定檢驗 IC21925E 或連接應用程式提供者。固定檢驗可用於 index.storsys.ibm.com 匿名 FTP 伺服器。這會使得 ADSM API 層次回到 3.1.3。

第 3 章 開發 DB2 應用程式的一般資訊

開發檔案、make 檔和錯誤檢查的公程式

表格 16 內 bldevm 的登錄應該被讀取為：

bldevm

事件監督程式範例程式，evm (只適用於 AIX、OS/2 和 Windows 32 位元作業系統)。

表格 17 應該包括下列登錄：

bldmevm

事件監督程式範例程式，evm，及 Microsoft Visual C++ 編譯器。

bldvevm

事件監督程式範例程式，evm，加上 VisualAge Visual C++ 編譯器。

第 4 章 開發 Java Applets 和應用程式

設定環境

若您在支援的平台上使用 IBM JDK 1.1.8 來建置 SQLJ 程式，則需要 1999 年 11 月 24 日 (或之後) 的 JDK 開發日期。否則您會在編譯期間收到 JNI 錯誤訊息。

若您在支援的平台上使用 IBM JDK 1.2.2 來建置 SQLJ 程式，則需要 2000 年 4 月 17 日 (或之後) 的 JDK 建置日期。否則，您會在編譯期間收到無效的 Java 類型。

針對次章節 **AIX**、**HP-UX**、**Linux** 及 **Solaris**，取代 JDBC 2.0 的資訊如下：

使用 Java 應用程式的 **JDBC 2.0 驅動程式**

JDBC 1.22 驅動程式是所有作業系統預設的驅動程式。欲使用 JDBC 2.0 的新特性，您必須安裝 JDK 1.2 支援。利用 JDBC 2.0 的新特性來執行應用程式之前，您必須自 `sql11ib/java12` 目錄發出 `usejdbc2` 指令來設定環境。希望應用程式可以一直使用 JDBC 2.0 驅動程式，您得考慮新增下列內容到登入設定檔，如 `.profile`，或至 Shell 起始設定 script，如 `.bashrc`、`.cshrc` 或 `.kshrc`：

```
. sql11ib/java12/usejdbc2
```

確定此指令放置在執行 `db2profile` 的指令之後，因為 `usejdbc2` 應該在 `db2profile` 之後執行。

欲往回切換至 JDBC 1.22 驅動程式，請在 `sql11ib/java12` 目錄下執行下列指令：

```
. usejdbc1
```

使用 Java 儲存程序和 UDF 的 JDBC 2.0 驅動程式

欲使用 Java 儲存程序和 UDF 的 JDBC 2.0 驅動程式，您必須設定案例隔離使用者的環境。預設的隔離使用者 ID 是 `db2fenc1`。欲設定隔離使用者 ID 的環境，請執行下列步驟：

1. 新增下列內容至隔離使用者 ID 設定檔，如 `.profile`，或至隔離使用者 ID Shell 起始設定 script，如 `.bashrc`、`.cshrc` 或 `.kshrc`：

```
. sql11ib/java12/usejdbc2
```

2. 自 CLP 發出下列指令：

```
db2set DB2_USE_JDK12=1
```

欲往回切換至支援 Java UDFs 和儲存程序的 JDBC 1.22 驅動程式，請執行下列步驟：

1. 在隔離使用者 ID 設定檔，如 `.profile`，或自隔離使用者 ID Shell 起始設定 script，如 `.bashrc`、`.cshrc` 或 `.kshrc`內將以下行列移除：

```
. sql11ib/java12/usejdbc2
```

2. 自 CLP 發出下列指令：

```
db2set DB2_USE_JDK12=
```

希望應用程式可以一直使用 JDBC 2.0 驅動程式，您可以新增以下行列至登入設定檔，如 `.profile`，或至 Shell 起始設定 script，如 `.bashrc`、`.cshrc` 或 `.kshrc`：

```
. sql11ib/java12/usejdbc2
```

確定此指令放置在執行 `db2profile` 的指令之後，因為 `usejdbc2` 應該在 `db2profile` 之後執行。

HP-UX

在 DB2 for HP-UX (含 JDK 1.1) 上，不支援 Java 儲存程序及使用者定義的函數。

Silicon Graphics IRIX

當使用 `-o32` 物件類型來建置 SQLJ 應用程式並使用具有 JDK 1.2.2 的 Java JIT 編譯器時，若 SQLJ 轉換程式因分區段錯誤而失敗，請嘗試以這個指令來關掉 JIT 編譯器：

```
export JAVA_COMPILER=NONE
```

在 Silicon Graphics IRIX 上開發 Java SQLJ 程式時需要 JDK 1.2.2。

Windows 32 位元作業系統

使用 Java 儲存程序和 UDF 的 JDBC 2.0 驅動程式

欲使用 Java 儲存程序和 UDF 的 JDBC 2.0 驅動程式，您必須執行下列步驟來設定環境：

1. 在 sqllib\java12 目錄下發出下列指令：

```
usejdbc2
```

2. 自 CLP 發出下列指令：

```
db2set DB2_USE_JDK12=1
```

欲往回切換至支援 Java UDFs 和儲存程序的 JDBC 1.22 驅動程式，請執行下列步驟：

1. 在 sqllib\java12 目錄下發出下列指令：

```
usejdbc2
```

2. 自 CLP 發出下列指令：

```
db2set DB2_USE_JDK12=
```

OS/2 上的 JDK 層次

執行 1999 年 9 月之前發表的 JDK 1.1.8 版的 OS/2 無法顯示某些訊息。請確定您具有最新的 JDK 1.1.8 版。

HP-UX 上的 Java2

若要執行 Java2 儲存程序，必須變更共用檔案庫路徑，如下所示：

```
export SHLIB_PATH=$JAVADIR/jre/lib/PA_RISC:$JAVADIR/  
jre/lib/PA_RISC/classic:$HOME/sqllib/lib/usr/lib:$SHLIB_PATH
```

\$JAVADIR 是 Java2 SDK 的位置。

第 5 章 開發 SQL 程序

設定 SQL 程序環境

下列是關於在「安裝」中設定 DB2 環境指示的附加說明。

若需 SQL 程序支援，您必須在伺服器上安裝「應用程式開發從屬站 (Application Development Client)」。若需安裝「應用程式開發從屬站」的相關資訊，請參照適用您的平台的 *快速入門* 書籍。若需您平台中 DB2 支援的 C 及 C++ 編譯器的相關資訊，請參閱「平台支援的軟體」。

註：在 OS/2 FAT 檔案系統上，「SQL 程序」的綱目名稱限制為八個字元以下。綱目名稱若長於八個字元，則必須使用 HPFS 檔案系統。

編譯器架構由兩部分組成：設定編譯器的環境變數及定義編譯指令。環境變數提供編譯器的二進位檔(bin)、檔案庫(lib)及併入檔的路徑。編譯指令是 DB2 將用來編譯 SQL 程序所產生的 C 檔案的完整指令。

設定編譯器環境變數

在 OS/2、Windows 及 UNIX 作業系統上配置環境有不同的規則，如下所示。在部分的情況下，不需要作架構；在其它情形時，DB2_SQLROUTINE_COMPILER_PATH DB2 登錄變數必須設定為指向一個適當的設定環境變數的可執行 Script。

註： 您可使用 **db2set** 指令或使用「儲存程序建置器」的「SQL 儲存程序建置選項」對話框來設定這個 DB2 登錄變數的值。使用「SQL 儲存程序建置選項」對話框不需要實際存取資料庫伺服器，資料庫伺服器也不用重新啓動以使變更生效。

在 OS/2 上：

針對 IBM VisualAge C++ for OS/2 版本 3.6：

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\ibmcxxo\bin\setenv.cmd"
```

針對 IBM VisualAge C++ for OS/2 版本 4：

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\ibmcpp40\bin\setenv.cmd"
```

註： 這些指令是假設 C++ 編譯器是安裝在 C: 磁碟機上。如果需要，變更磁碟機或路徑來反映您系統上 C++ 編譯器的位置。

在 Windows 32 位元作業系統上，如果編譯器的環境變數是設定為 SYSTEM 變數，則不需要作任何配置。否則，請將 DB2_SQLROUTINE_COMPILER_PATH DB2 登錄變數設定為：

針對 Microsoft Visual C++ 版本 5.0：

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\devstudio\vc\bin\vcvars32.bat"
```

針對 Microsoft Visual C++ 版本 6.0：

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\Micros~1\vc98\bin\vcvars32.bat"
```

針對 IBM VisualAge C++ for Windows 版本 3.6：

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\ibmcxxw\bin\setenv.bat"
```

針對 IBM VisualAge C++ for Windows 版本 4：

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\ibmcppw40\bin\setenv.bat"
```

註： 這些指令是假設 C++ 編譯器是安裝在 C: 磁碟機上。如果需要，變更磁碟機或路徑來反映您系統上 C++ 編譯器的位置。

在 UNIX 作業系統上，DB2 會在您第一次編譯儲存程序時產生可執行的 Script 檔 \$HOME/sql/lib/function/routine/sr_cpath (包含編譯器環境變數的預設值)。您可編輯此檔案，如果預設值不適用於您的編譯器。另外，您可將 DB2_SQLROUTINE_COMPILER_PATH DB2 登錄變數設為包含另一個可執行 Script 的完整路徑名稱，在 Script 內指定期望的值 (請看上面的範例)。

自訂編譯指令

Application Development Client 的安裝提供了一個預設的編譯指令，至少可用於每個平台上支援的編譯器之一：

```
AIX : IBM C Set++ for AIX 版本 3.6.6  
Solaris : SPARCompiler C++ 版本 4.2 及 5.0  
HP-UX : HP-UX C++ 版本 A.12.00
```

Linux : GNU/Linux g++ 版本 egcs-2.90.27 980315 (egcs-1.0.2 release)
PTX : ptx/C++ 版本 5.2
OS/2 : IBM VisualAge C++ for OS/2 版本 3
Windows NT 及 Windows 2000 : Microsoft Visual C++ 版本 5.0 及 6.0

若要使用其它編譯器，或要自訂預設指令，您必須將
DB2_SQLROUTINE_COMPILE_COMMAND DB2 登錄變數設為如下的指令：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=compilation_command
```

其中 *compilation_command* 是 C 或 C++ 編譯指令，包括建立儲存程序所需的選項及參數。

在編譯指令中，請使用關鍵字 `SQLROUTINE_FILENAME` 來取代所產生的 SQC、C、PDB、DEF、EXP、訊息日誌及共用檔案庫檔案的檔名。僅針對 AIX，使用 `SQLROUTINE_ENTRY` 關鍵字來取代登錄名稱。

註： 您可使用 `db2set` 指令或使用「儲存程序建置器」的「SQL 儲存程序建置選項」對話框來設定這個 DB2 登錄變數的值。使用「SQL 儲存程序建置選項」對話框不需要實際存取資料庫伺服器，資料庫伺服器也不用重新啟動以使變更生效。

下列是 C 或 C++ 編譯器在支援的伺服器平台的
DB2_SQLROUTINE_COMPILE_COMMAND 預設值。

AIX

使用 IBM C for AIX 版本 3.6.6：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=x1c -H512 -T512 \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c -bE:SQLROUTINE_FILENAME.exp \  
-e SQLROUTINE_ENTRY -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -lc -ldb2
```

使用 IBM C Set++ for AIX 版本 3.6.6：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=x1C -H512 -T512 \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c -bE:SQLROUTINE_FILENAME.exp \  
-e SQLROUTINE_ENTRY -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -lc -ldb2
```

若沒有設定 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登錄變數，則這是預設編譯指令。

註： 若要在 AIX 上編譯 64 位元 SQL 程序，將 `-q64` 選項新增到上面的指令。

使用 IBM VisualAge C++ for AIX 版本 4：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld"
```

若您在 `vacbld` 指令之後沒有指定架構檔，DB2 會在首次建立 SQL 程序時，建立下列預設的架構檔。

```
$HOME/sql1lib/function/routine/sqlproc.icc
```

若您要使用自己的架構檔，可在設定 DB2_SQLROUTINE_COMPILE_COMMAND 的 DB2 登錄值時，指定您要使用的架構檔：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld  
%DB2PATH%/function/sqlproc.icc"
```

HP-UX

使用 HP C 編譯器版本 A.11.00.03：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc +DAportable +ul -Aa +z \
-I$HOME/sql1ib/include -c SQLROUTINE_FILENAME.c; \
ld -b -o SQLROUTINE_FILENAME SQLROUTINE_FILENAME.o \
-L$HOME/sql1ib/lib -ldb2
```

使用 HP-UX C++ 版本 A.12.00 :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=CC +DAportable +a1 +z -ext \
-I$HOME/sql1ib/include -c SQLROUTINE_FILENAME.c; \
ld -b -o SQLROUTINE_FILENAME SQLROUTINE_FILENAME.o \
-L$HOME/sql1ib/lib -ldb2
```

若沒有設定 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登記變數，則這是預設編譯指令。

Linux

使用 GNU/Linux gcc 版本 2.7.2.3 :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc \
-I$HOME/sql1ib/include SQLROUTINE_FILENAME.c \
-shared -o SQLROUTINE_FILENAME -L$HOME/sql1ib/lib -ldb2
```

使用 GNU/Linux g++ 版本 egcs-2.90.27 980315 (egcs-1.0.2 版次) :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=g++ \
-I$HOME/sql1ib/include SQLROUTINE_FILENAME.c \
-shared -o SQLROUTINE_FILENAME -L$HOME/sql1ib/lib -ldb2
```

若沒有設定 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登記變數，則這是預設編譯指令。

PTX

使用 ptx/C 版本 4.5 :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc -KPIC \
-I$HOME/sql1ib/include SQLROUTINE_FILENAME.c \
-G -o SQLROUTINE_FILENAME.so -L$HOME/sql1ib/lib -ldb2 ; \
cp SQLROUTINE_FILENAME.so SQLROUTINE_FILENAME
```

使用 ptx/C++ 版本 5.2 :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=c++ -KPIC \
-D RWSTD_COMPILE_INSTANTIATE=0 \
-I$HOME/sql1ib/include SQLROUTINE_FILENAME.c \
-G -o SQLROUTINE_FILENAME.so -L$HOME/sql1ib/lib -ldb2 ; \
cp SQLROUTINE_FILENAME.so SQLROUTINE_FILENAME
```

若沒有設定 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登記變數，則這是預設編譯指令。

OS/2

使用 IBM VisualAge C++ for OS/2 版本 3 :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="icc -Ge- -Gm+ -W2 \
-I%DB2PATH%\include SQLROUTINE_FILENAME.c \
/B\%/NOFREE /NOI /ST:64000\% SQLROUTINE_FILENAME.def \
%DB2PATH%\lib\db2api.lib"
```

若沒有設定 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登記變數，則這是預設編譯指令。

使用 IBM VisualAge C++ for OS/2 版本 4 :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacblid"
```

若您在 vacblid 指令之後沒有指定架構檔，DB2 會在首次建立 SQL 程序時，建立下列預設的架構檔。

```
%DB2PATH%\function\routine\sqlproc.icc
```

若您要使用自己的架構檔，可在設定 DB2_SQLROUTINE_COMPILE_COMMAND 的 DB2 登記值時，指定您要使用的架構檔：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacblid  
%DB2PATH%\function\sqlproc.icc"
```

Solaris

使用 SPARCompiler C 版本 4.2 和 5.0 :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc -xarch=v8plusa -Kpic \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-G -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib \  
-R$HOME/sql1lib/lib -ldb2
```

使用 SPARCompiler C++ 版本 4.2 和 5.0 :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=CC -xarch=v8plusa -Kpic \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-G -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib \  
-R$HOME/sql1lib/lib -ldb2
```

若沒有設定 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登記變數，則這是預設編譯指令。

註:

1. -xarch=v8plusa 編譯器選項已新增到預設編譯器指令。有關為何新增此選項的細節，請參閱第317頁的『第 12 章 開發 Solaris 應用程式』。
2. 若要在 Solaris 上編譯 64 位元 SQL 程序，取消 -xarch=v8plusa 選項，並將 -xarch=v9 選項新增到上面的指令。

Windows NT 和 Windows 2000

註: Windows 98 或 Windows 95 不支援 SQL 程序。

使用 Microsoft Visual C++ 版本 5.0 和 6.0 :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=c1 -Od -w2 /TC -D_X86_=1  
-I%DB2PATH%\include SQLROUTINE_FILENAME.c /link -dll  
-def:SQLROUTINE_FILENAME.def /out:SQLROUTINE_FILENAME.dll  
%DB2PATH%\lib\db2api.lib
```

若沒有設定 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登記變數，則這是預設編譯指令。

使用 IBM VisualAge C++ for Windows 版本 3.6 :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="ilib /GI  
SQLROUTINE_FILENAME.def &icc -Ti -Ge- -Gm+ -W2  
-I%DB2PATH%\include SQLROUTINE_FILENAME.c  
/B\"/ST:64000 /PM:VIO /DLL\" SQLROUTINE_FILENAME.exp  
%DB2PATH%\lib\db2api.lib"
```

使用 IBM VisualAge C++ for Windows 版本 4 :

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld"
```

若您在 vacbld 指令之後沒有指定架構檔，DB2 會在首次建立 SQL 程序時，建立下列預設的架構檔。

```
%DB2PATH%\function\routine\sqlproc.icc
```

若您要使用自己的架構檔，可在設定 DB2_SQLROUTINE_COMPILE_COMMAND 的 DB2 登記值時，指定您要使用的架構檔：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld  
%DB2PATH%\function\sqlproc.icc"
```

若要傳回預設編譯器選項，以下列指令將 DB2_SQLROUTINE_COMPILE_COMMAND 的 DB2 登記值設定為 NULL：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=
```

保留中間檔

當 SQL 程序沒有成功建立時，您必須手動刪除被丟棄的中間檔。這些檔案位於下列目錄裡：

UNIX \$DB2PATH/function/routine/sqlproc/\$DATABASE/\$SCHEMA/tmp

在此 \$DB2PATH 是建立案例的目錄，\$DATABASE 是資料庫名稱，而 \$SCHEMA 是用來建立 SQL 程序的綱目名稱。

OS/2 和 Windows

```
%DB2PATH%\function\routine\sqlproc%\DATABASE%\%SCHEMA%\tmp
```

其中 %DB2PATH% 代表建立案例的目錄，%DATABASE% 代表資料庫名稱，而 %SCHEMA% 代表用來建立 SQL 程序的綱目名稱。

備份和復置

在建立 SQL 程序時，產生的共用檔案庫/DLL 也會保留在型錄表格中，如果產生的共用檔案庫/DLL 小於 2 MB。在備份及復置資料庫時，任何產生的共用檔案庫/DLL 小於 2 MB 的 SQL 程序會使用型錄表格中所保留的版本來做備份及復置。如果您有產生的共用檔案庫/DLL 大於 2 MB 的 SQL 程序，請確定您也用資料庫備份及復置來做檔案系統的備份及復置。如果沒有，您必須使用 syscat.procedures 型錄表格中的來源自行重建 SQL 程序的共用檔案庫/DLL。

註：在資料庫回復時，所有檔案系統上屬於被回復資料庫的 SQL 程序可執行檔會被移除。如果索引建立架構參數 (indexrec) 設為 RESTART，就會從型錄表格擷取所有 SQL 程序可執行檔，並於下次連接時放回檔案系統。否則，SQL 可執行檔會在 SQL 程序第一次執行時擷取。

可執行檔會被放回到下列目錄：

UNIX \$DB2PATH/function/routine/sqlproc/\$DATABASE

在此 \$DB2PATH 是建立案例的目錄，而 \$DATABASE 是用來建立 SQL 程序的資料庫名稱。

OS/2 和 Windows

```
%DB2PATH%\function\routine\sqlproc%\DATABASE%
```

其中 %DB2PATH% 代表建立案例的目錄，而 %DATABASE% 代表用來建立 SQL 程序的資料庫名稱。

建立 SQL 程序

將資料庫管理程式的架構參數 `KEEPDARI` 設定為 'NO'，以建置 SQL 程序。若 SQL 程序一旦執行時維持載入狀態，要以同名捨棄和重建儲存程序時可能會遭遇問題，如同檔案庫無法復新，而可執行檔無法從檔案系統捨棄。當您嘗試復原變更或捨棄資料庫時也會遭遇問題，因為無法刪除可執行檔。

請參閱「應用程式開發手冊，第 2 章 設定」的「更新資料庫管理程式架構檔」，以取得設定 `KEEPDARI` 參數的詳細資訊。

註: SQL 程序不支援下列參數的資料類型：

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

呼叫儲存程序

「使用呼叫指令」中的第一段應該讀取：

欲使用呼叫指令，請輸入儲存程序名稱，以及 `IN` 或 `INOUT` 的參數，還有被視為每一個輸出參數之位置保留符號 '?'。若要取得 `CALL` 指令的詳細語法，請參閱第134頁的『`CALL`』。

分送已編譯的 SQL 程序

註: 若要在 DB2 伺服器之間分送已編譯的 SQL 程序，您必須對於每一個會作為已編譯 SQL 程序的來源 (或目標) 的 DB2 伺服器執行下列步驟：

步驟 1. 安裝 FixPak 3

步驟 2. 發出 `db2updv7` 指令以啓用 DB2 來取出及安裝已編譯的 SQL 程序：

```
db2updv7 -d database_name
```

當您定義一個 SQL 程序時，它會轉換為一個 C 程式、預先編譯過、對目標資料庫連結、編譯並鏈結以建立一個共用檔案庫。編譯及鏈結的步驟需要在資料庫伺服器機器上有可用的 C 或 C++ 編譯器。然而，您在定義 SQL 程序時，您可將它以已編譯的形式分送到在相同平台上執行的 DB2 資料庫，而不需要可存取 C 或 C++ 編譯器。DB2 容許使用者從一個資料庫取出已編譯形式的 SQL 程序，以及安裝已編譯形式的 SQL 程序到另一個資料庫。

DB2 提供了指令行介面及程式設計介面來執行取出及安裝作業。指令行介面由兩個 `CLP` 指令組成：`GET ROUTINE` 及 `PUT ROUTINE`。程式設計介面由兩個內建儲存程序組成：`GET_ROUTINE_SAR` 及 `PUT_ROUTINE_SAR`。有關指令行介面的詳細資訊，請參照 `Command Reference`。有關程式設計介面的詳細資訊，請參照 `SQL Reference`。

若要從一個資料庫伺服器分送已編譯的 SQL 程序到另一個資料庫伺服器，請執行下列步驟：

步驟 1. 開發應用程式，包括定義為應用程式部分的 SQL 程序。

步驟 2. 在測試程序後，將每一個程序的已編譯版本取出到一個不同的檔案。若需詳細資訊，請參照 *Command Reference* 中的 *GET ROUTINE* 指令或 *SQL Reference* 中的 *GET_ROUTINE_SAR* 儲存程序。

步驟 3. 在每一台伺服器上安裝每一個程序的已編譯版本，您可發出 *PUT ROUTINE* 指令，或呼叫 *PUT_ROUTINE_SAR* 儲存程序 (使用 2 所建立的檔案)。每一個資料庫伺服器的作業系統及 DB2 層次必須相同。

第 7 章 開發 HP-UX 應用程式

HP-UX C

在「多重執行緒應用程式」內，*bldmt* Script 檔由不同的編譯選項修正。新版本在 *sqllib/samples/c* 目錄下。

HP-UX C++

在建置 Script 中，HP aC++ 編譯器的 C++ 編譯器變數 *CC* 被取代為 *aCC*。建置 Script 的修正版在 *sqllib/samples/cpp* 目錄下。

"*+u1*" 編譯選項應該和 *aCC* 編譯器一起用來建置儲存程序及 UDF。此選項容許未調整的資料存取。附在 *DB2 for HP-UX* 中的範例建置 Script、*bldsrv* 和 *bldudf*，以及範例 *makefile* 尚未用此選項更新。在使用前應該要修改它們來加入這個選項。這裡是 *bldsrv* 及 *bldudf* Script 的新編譯步驟：

```
aCC +DAportable +u1 -Aa +z -ext -I$DB2PATH/include -c $1.C
```

在「多重執行緒應用程式」內，*bldmt* Script 檔由不同的編譯選項修正。新版本在 *sqllib/samples/cpp* 目錄下。

第 9 章 開發 OS/2 應用程式

VisualAge C++ for OS/2 版本 4.0

針對 OS/2 和 Windows，使用 *set* 指令而不使用此節中的 *export* 指令。例如，*set CLI=tbinfo*。

在「DB2 CLI 應用程式」的次章節「建置並執行內含的 SQL 應用程式」裡，針對 OS/2 和 Windows，必須使用 *cliapi.icc* 檔而非 *cli.icc* 檔，因為內含的 SQL 應用程式需要由 *cliapi.icc* 鏈結的 *db2api.lib* 檔案庫。

第 10 章 開發 PTX 應用程式

ptx/C++

檔案庫需要以 *-shared* 選項鏈結，以建置儲存程序和使用者的定義函數。在 *sqllib/samples* 目錄裡，*makefile*、建置 *scripts* *bldsrv* 和 *bldudf* 已更新併入這個選項，如同下列 *bldsrv* 的鏈結步驟：

```
c++ -shared -G -o $1 $1.o -L$DB2PATH/lib -ldb2
```

第 12 章 開發 Solaris 應用程式

SPARCompiler C++

於 Solaris 執行 C/C++ 應用程式和 SQL 程序的問題

使用 Sun WorkShop Compiler C/C++，如果您碰到可執行檔接收到下列錯誤訊息時：

1. 第一行語法錯誤: '('預期的
2. ksh: <應用程式名稱>: 不能執行 (應用程式名稱爲可執行檔名稱)

您可能遇到的問題是，編譯器在鏈結 libdb2.so 時，沒有產生有效的可執行檔。建議修正方法，新增下列編譯器選項至編譯和鏈結指令：

```
-xarch=v8plusa
```

範例，當編譯範例應用程式 dynamic.sqc：

```
embprep dynamic sample
embprep utilemb sample
cc -c utilemb.c -xarch=v8plusa -I/export/home/db2inst1/sqllib/include
cc -o dynamic dynamic.c utilemb.o -xarch=v8plusa -I/export/home/db2inst1/sqllib/include \
-L/export/home/db2inst1/sqllib/lib -R/export/home/db2inst1/sqllib/lib -l db2
```

註：

1. 若您在 Solaris 上使用 SQL 程序，並透過 DB2_SQLROUTINE_COMPILE_COMMAND 設定檔變數來使用自己的編譯字串，請確定您已將上述的編譯器選項併入。預設的編譯器指令包括此選項：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="cc -# -Kpic -xarch=v8plusa -I$HOME/sqllib/inclu
SQLROUTINE_FILENAME.c -G -o SQLROUTINE_FILENAME -L$HOME/sqllib/lib -R$HOME/sqllib/lib
```

2. 若要在 Solaris 上編譯 64 位元 SQL 程序，取消 -xarch=v8plusa 選項，並將 -xarch=v9 選項新增到上面的指令。

第 13 章 建置 Windows 32 位元作業系統的應用程式

VisualAge C++ 版本 4.0

針對 OS/2 和 Windows，使用 set 指令而不使用此節中的 export 指令。例如，set CLI=tbinfo。

在「DB2 CLI 應用程式」的次章節「建置並執行內含的 SQL 應用程式」裡，針對 OS/2 和 Windows，必須使用 cliapi.icc 檔而非 cli.icc 檔，因爲內含的 SQL 應用程式需要由 cliapi.icc 鏈結的 db2api.lib 檔案庫。

Application Development Guide(無中文版)

Chapter 2. Coding a DB2 Application

Activating the IBM DB2 Universal Database Project and Tool Add-ins for Microsoft Visual C++

Before running the **db2vccmd** command (step 1), please ensure that you have started and stopped Visual C++ at least once with your current login ID. The first time you run Visual C++, a profile is created for your user ID, and that is what gets updated by the **db2vccmd** command. If you have not started it once, and you try to run **db2vccmd**, you may see errors like the following:

```
"Registering DB2 Project add-in ...Failed! (rc = 2)"
```

Chapter 6. Common DB2 Application Techniques

Generating Sequential Values

Generating sequential values is a common database application development problem. The best solution to that problem is to use sequence objects and sequence expressions in SQL. Each *sequence object* is a uniquely named database object that can be accessed only by sequence expressions. There are two *sequence expressions*: the PREVVAL expression and the NEXTVAL expression. The PREVVAL expression returns the most recently generated value for the specified sequence for a previous statement. The NEXTVAL sequence expression increments the value of the sequence object and returns the new value of the sequence object.

To create a sequence object, issue the CREATE SEQUENCE statement. For example, to create a sequence object called id_values using the default attributes, issue the following statement:

```
CREATE SEQUENCE id_values
```

To display the current value of the sequence object, issue a VALUES statement using the PREVVAL expression:

```
VALUES PREVVAL FOR id_values
```

```
1
-----
1
1 record(s) selected.
```

You can repeatedly retrieve the current value of the sequence object, and the value that the sequence object returns does not change until you issue a NEXTVAL expression. In the following example, the PREVVAL expression returns a value of 1, until the NEXTVAL expression increments the value of the sequence object:

```
VALUES PREVVAL FOR id_values
1
-----
1
1 record(s) selected.
```

```
VALUES PREVVAL FOR id_values
1
-----
1
1 record(s) selected.
```

```
VALUES NEXTVAL FOR id_values
1
-----
2
1 record(s) selected.
```

```
VALUES PREVVAL FOR id_values
1
-----
2
1 record(s) selected.
```

To update the value of a column with the next value of the sequence object, include the `NEXTVAL` expression in the `UPDATE` statement, as follows:

```
UPDATE staff
SET id = NEXTVAL FOR id_values
WHERE id = 350
```

To insert a new row into a table using the next value of the sequence object, include the `NEXTVAL` expression in the `INSERT` statement, as follows:

```
INSERT INTO staff (id, name, dept, job)
VALUES (NEXTVAL FOR id_values, 'Kandil', 51, 'Mgr')
```

For more information on the `PREVVAL` and `NEXTVAL` expressions, refer to the *SQL Reference*.

Controlling Sequence Behavior

You can tailor the behavior of sequence objects to meet the needs of your application. You change the attributes of a sequence object when you issue the `CREATE SEQUENCE` statement to create a new sequence object, and when you issue the `ALTER SEQUENCE` statement for an existing sequence object. Following are some of the attributes of a sequence object that you can specify:

Data type

The `AS` clause of the `CREATE SEQUENCE` statement specifies the numeric data type of the sequence object. The data type, as specified in the “SQL Limits” appendix of the *SQL Reference*, determines the possible minimum and maximum values of the sequence object. You cannot change the data type of a sequence

object; instead, you must drop the sequence object by issuing the DROP SEQUENCE statement and issuing a CREATE SEQUENCE statement with the new data type.

Start value

The START WITH clause of the CREATE SEQUENCE statement sets the initial value of the sequence object. The RESTART WITH clause of the ALTER SEQUENCE statement resets the value of the sequence object to a specified value.

Minimum value

The MINVALUE clause sets the minimum value of the sequence object.

Maximum value

The MAXVALUE clause sets the maximum value of the sequence object.

Increment value

The INCREMENT BY clause sets the value that each NEXTVAL expression adds to the sequence object. To decrement the value of the sequence object, specify a negative value.

Sequence cycling

The CYCLE clause causes the value of a sequence object that reaches its maximum or minimum value to return to its start value on the following NEXTVAL expression.

For example, to create a sequence object called id_values that starts with a value of 0, has a maximum value of 1000, increments by 2 with each NEXTVAL expression, and returns to its start value when the maximum value is reached, issue the following statement:

```
CREATE SEQUENCE id_values
  START WITH 0
  INCREMENT BY 2
  MAXVALUE 1000
  CYCLE
```

For more information on the CREATE SEQUENCE and ALTER SEQUENCE statements, refer to the *SQL Reference*.

Improving Performance with Sequence Objects

Like identity columns, using sequence objects to generate values generally improves the performance of your applications in comparison to alternative approaches. The alternative to sequence objects is to create a single-column table that stores the current value and incrementing that value with either a trigger or under the control of the application. In a distributed environment where applications concurrently access the single-column table, the locking required to force serialized access to the table can seriously affect performance.

Sequence objects avoid the locking issues that are associated with the single-column table approach and can cache sequence values in memory to improve DB2 response time. To maximize the performance of applications that use sequence objects, ensure that your sequence object caches an appropriate amount of sequence values. The CACHE clause of the CREATE SEQUENCE and ALTER SEQUENCE statements specifies the maximum number of sequence values that DB2 generates and stores in memory.

If your sequence object must generate values in order, without introducing gaps in that order due to a system failure or database deactivation, use the ORDER and NO CACHE

clauses in the CREATE SEQUENCE statement. The NO CACHE clause guarantees that no gaps appear in the generated values at the cost of some of your application's performance because it forces your sequence object to write to the database log every time it generates a new value.

Comparing Sequence Objects and Identity Columns

Although sequence objects and identity columns appear to serve similar purposes for DB2 applications, there are a number of important differences:

- An identity column automatically generates values for a column in a single table. A sequence object generates sequential values that can be used in any SQL statement.
- An identity column generates values that are guaranteed to be unique. Including the CYCLE clause in a CREATE SEQUENCE or ALTER SEQUENCE statement enables that sequence object to generate duplicate values.

Chapter 7. Stored Procedures

DECIMAL Type Fails in Linux Java Routines

This problem occurs because the IBM Developer Kit for Java does not create links for its libraries in the /usr/lib directory. The security model for DB2 routines does not allow them to access libraries outside of the standard system libraries. To enable DECIMAL support in Java routines on Linux, perform the following steps:

1. Create symbolic links from the IBM Developer Kit for Java libraries to /usr/lib/ by issuing the following command with root authority:

For IBM Developer Kit for Java 1.1.8:

```
ln -sf /usr/jdk118/lib/linux/native_threads/* /usr/lib/
```

For IBM Developer Kit for Java 1.3:

```
ln -sf /opt/IBMJava2-13/jre/bin/*.so /usr/lib/
```

2. Issue the **ldconfig** command to update the list of system-wide libraries.

Using Cursors in Recursive Stored Procedures

To avoid errors when using SQL Procedures or stored procedures written in embedded SQL, close all open cursors before issuing a recursive CALL statement.

For example, assume the stored procedure MYPROC contains the following code fragment:

```
OPEN c1;  
CALL MYPROC();  
CLOSE c1;
```

DB2 returns an error when MYPROC is called because cursor c1 is still open when MYPROC issues a recursive CALL statement. The specific error returned by DB2 depends on the actions MYPROC performs on the cursor.

To successfully call MYPROC, rewrite MYPROC to close any open cursors *before* the nested CALL statement as shown in the following example:

```
OPEN c1;  
CLOSE c1;  
CALL MYPROC();
```

Close all open cursors before issuing the nested CALL statement to avoid an error.

Writing OLE Automation Stored Procedures

The last sentence in the following paragraph is missing from the second paragraph under section "Writing OLE automation Stored Procedures":

After you code an OLE automation object, you must register the methods of the object as stored procedures using the CREATE PROCEDURE statement. To register an OLE automation stored procedure, issue a CREATE PROCEDURE statement with the LANGUAGE OLE clause. The external name consists of the OLE progID identifying the OLE automation object and the method name separated by ! (exclamation mark). The OLE automation object needs to be implemented as an in-process server (.DLL).

Chapter 12. Working with Complex Objects: User-Defined Structured Types

Inserting Structured Type Attributes Into Columns

The following rule applies to embedded static SQL statements: To insert an attribute of a user-defined structured type into a column that is of the same type as the attribute, enclose the host variable that represents the instance of the type in parentheses, and append the double-dot operator and attribute name to the closing parenthesis. For example, consider the following situation:

- PERSON_T is a structured type that includes the attribute NAME of type VARCHAR(30).
- T1 is a table that includes a column C1 of type VARCHAR(30).
- personhv is the host variable declared for type PERSON_T in the programming language.

The proper syntax for inserting the NAME attribute into column C1 is:

```
EXEC SQL INSERT INTO T1 (C1) VALUES ((:personhv)..NAME)
```

Chapter 13. Using Large Objects (LOBs)

Large object (LOBs) support in federated database systems

DB2 supports three types of large objects (LOBs): character large objects (CLOBs), double-byte character large objects (DBCLOBs) and binary large objects (BLOBs). For general information about DB2 LOB support, see the following DB2 books:

- *DB2 Application Development Guide*(無中文版)
- *DB2 SQL Reference*(無中文版)
- *DB2 Administration Guide: Planning*(無中文版)

In a federated database system, you can access and manipulate LOBs at remote data sources. Because LOBs can be very large, transferring LOBs from a remote data source can be time consuming. The DB2 federated database attempts to minimize transferring LOB data from the data sources, and also attempts to deliver requested LOB data directly from the data source to the requesting application without materializing the LOB at DB2.

This section discusses:

- How DB2 retrieves LOBs

- How applications can use LOB locators
- Restrictions on LOBs
- Mappings between LOB and non-LOB data types
- Tuning the system

How DB2 retrieves LOBs

DB2 federated systems use two mechanisms to retrieve LOBs: LOB streaming and LOB materialization.

LOB streaming: In LOB streaming, LOB data is retrieved in stages. DB2 uses LOB streaming for data in result sets of queries that are completely pushed down. For example, consider the following query:

```
SELECT empname, picture FROM orc_emp_table WHERE empno = '01192345'
```

where *picture* represents a LOB column and *orc_emp_table* represents a nickname referencing an Oracle table containing employee data. The DB2 query processor marks the *picture* column for streaming if it decides to run the entire query at the Oracle data source. At execution time, if DB2 notes that a LOB is marked for streaming, it retrieves the LOB in stages from the data source. DB2 then transfers the data to the application memory space.

LOB materialization: In LOB materialization, the remote LOB data is retrieved by DB2 and stored locally at the federated server. DB2 uses LOB materialization when:

- The LOB column cannot be deferred or streamed.
- A function must be applied to a LOB column locally, before the data is transferred. This happens when DB2 compensates for functions not available at a remote data source. For example, Microsoft SQL Server does not provide a SUBSTR function for LOB columns. To compensate, DB2 materializes the LOB column locally and applies the DB2 SUBSTR function to the retrieved LOB.

How applications can use LOB locators

Applications can request LOB locators for LOBs stored in remote data sources. A LOB locator is a 4-byte value stored in a host variable that a program can use to refer to a LOB value (or LOB expression) held in the database system. Using a LOB locator, a program can manipulate the LOB value as if the LOB value was stored in a regular host variable. The difference in using the LOB locator is that there is no need to transport the LOB value from the server to the application (and possibly back again). See the *DB2 Application Development Guide* for additional information about LOB locators.

DB2 can retrieve LOBs from remote data sources, store them at DB2, and then issue a LOB locator against the stored LOB. LOB locators are released when:

- Applications issue "FREE LOCATOR" SQL statements.
- Applications issue COMMIT statements.
- DB2 is restarted.

Restrictions on LOBs

When using and retrieving LOBs, consider that:

- DB2 is unable to bind remote LOBs to a file reference variable.

- LOBs are not supported in pass-through mode.

Mappings between LOB and non-LOB data types

There are a few cases in which you can map a DB2 LOB data type to a non-LOB data type at a data source. When you need to create a mapping between a column with a DB2 LOB type and its counterpart column at a data source, it is recommended that you use a LOB data type as a counterpart if at all possible.

To create a mapping, use the create type mapping DDL statement. For example:

```
CREATE TYPE MAPPING my_oracle_lob FROM sysibm.clob TO SERVER TYPE oracle TYPE long
```

where:

my_oracle_lob

Is the name of the type mapping.

sysibm.clob

Is the DB2 CLOB data type.

oracle Is the type of server you are connecting to.

long Is the Oracle data type counterpart.

Tuning the system

If an application that retrieves remote LOBs returns an error message indicating there is not enough system resources to process the statement, increase the value of the application heap size parameter, APPLHEAPSZ, in the database configuration file. For example:

```
DB2 UPDATE DB CFG FOR EMPLOYEE USING APPLHEAPSZ 512
```

where *EMPLOYEE* is the name of the database you are tuning and *512* is the value of the application heap size parameter.

Part 5. DB2 Programming Considerations

IBM DB2 OLE DB Provider

Installing IBM DB2 Version 7.1 FixPak 1 or later corrects the condition that caused DB2 to issue the following error:

```
Test connection failed because of an error in initializing provider.  
The IBM OLE DB Provider is not available at this time. Please refer to  
the readme file for more information.
```

For more information on using the IBM OLE DB Provider for DB2, please refer to <http://www.ibm.com/software/data/db2/udb/ad/v71/oledb.html>.

Chapter 20. Programming in C and C++

The following table supplements the information included in chapter 7, "Stored Procedures", chapter 15, "Writing User-Defined Functions and Methods", and chapter 20, 『Programming in C and C++』. The table lists the supported mappings between SQL data types and C data types for stored procedures, UDFs, and methods.

C/C++ Types for Stored Procedures, Functions, and Methods

Table 22. SQL Data Types Mapped to C/C++ Declarations

SQL Column Type	C/C++ Data Type	SQL Column Type Description
SMALLINT (500 or 501)	sqlint16	16-bit signed integer
INTEGER (496 or 497)	sqlint32	32-bit signed integer
BIGINT (492 or 493)	sqlint64	64-bit signed integer
REAL (480 or 481)	float	Single-precision floating point
DOUBLE (480 or 481)	double	Double-precision floating point
DECIMAL(<i>p,s</i>) (484 or 485)	Not supported.	To pass a decimal value, define the parameter to be of a data type castable from DECIMAL (for example CHAR or DOUBLE) and explicitly cast the argument to this type.
CHAR(<i>n</i>) (452 or 453)	char[<i>n+1</i>] where <i>n</i> is large enough to hold the data 1<= <i>n</i> <=254	Fixed-length, null-terminated character string
CHAR(<i>n</i>) FOR BIT DATA (452 or 453)	char[<i>n+1</i>] where <i>n</i> is large enough to hold the data 1<= <i>n</i> <=254	Fixed-length character string
VARCHAR(<i>n</i>) (448 or 449) (460 or 461)	char[<i>n+1</i>] where <i>n</i> is large enough to hold the data 1<= <i>n</i> <=32 672	Null-terminated varying length string
VARCHAR(<i>n</i>) FOR BIT DATA (448 or 449)	struct { sqluint16 length; char[<i>n</i>]; }	Not null-terminated varying length character string
	1<= <i>n</i> <=32 672	
LONG VARCHAR (456 or 457)	struct { sqluint16 length; char[<i>n</i>]; }	Not null-terminated varying length character string
	32 673<= <i>n</i> <=32 700	
CLOB(<i>n</i>) (408 or 409)	struct { sqluint32 length; char data[<i>n</i>]; }	Non null-terminated varying length character string with 4-byte string length indicator
	1<= <i>n</i> <=2 147 483 647	
BLOB(<i>n</i>) (404 or 405)	struct { sqluint32 length; char data[<i>n</i>]; }	Non null-terminated varying binary string with 4-byte string length indicator
	1<= <i>n</i> <=2 147 483 647	
DATE (384 or 385)	char[11]	null-terminated character form

Table 22. SQL Data Types Mapped to C/C++ Declarations (continue)

SQL Column Type	C/C++ Data Type	SQL Column Type Description
TIME (388 or 389)	char[9]	null-terminated character form
TIMESTAMP (392 or 393)	char[27]	null-terminated character form
註: The following data types are only available in the DBCS or EUC environment when precompiled with the WCHARTYPE NOCONVERT option.		
GRAPHIC(<i>n</i>) (468 or 469)	sqldbchar[<i>n+1</i>] where <i>n</i> is large enough to hold the data 1<= <i>n</i> <=127	Fixed-length, null-terminated double-byte character string
VARGRAPHIC(<i>n</i>) (400 or 401)	sqldbchar[<i>n+1</i>] where <i>n</i> is large enough to hold the data 1<= <i>n</i> <=16 336	Not null-terminated, variable-length double-byte character string
LONG VARGRAPHIC (472 or 473)	struct { sqluint16 length; sqldbchar[<i>n</i>] }	Not null-terminated, variable-length double-byte character string
	16 337<= <i>n</i> <=16 350	
DBCLOB(<i>n</i>) (412 or 413)	struct { sqluint32 length; sqldbchar data[<i>n</i>]; }	Non null-terminated varying length character string with 4-byte string length indicator
	1<= <i>n</i> <=1 073 741 823	

Chapter 21. Programming in Java

Java Method Signature in PARAMETER STYLE JAVA Procedures and Functions

If specified after the Java method name in the EXTERNAL NAME clause of the CREATE PROCEDURE or CREATE FUNCTION statement, the Java method signature must correspond to the default Java type mapping for the signature specified after the procedure or function name. For example, the default Java mapping of the SQL type INTEGER is "int", not "java.lang.Integer".

Connecting to the JDBC Applet Server

It is essential that the db2java.zip file used by the Java applet be at the same FixPak level as the JDBC applet server. Under normal circumstances, db2java.zip is loaded from the Web Server where the JDBC applet server is running, as shown in Figure 22 of the book. This ensures a match. If, however, your configuration has the Java applet loading db2java.zip from a different location, a mismatch can occur. Prior to FixPak 2, this could lead to unexpected failures. As of FixPak 2, matching FixPak levels between the two files is strictly enforced at connection time. If a mismatch is detected, the connection is rejected, and the client receives one of the following exceptions:

- If db2java.zip is at FixPak 2 or later:

```
COM.ibm.db2.jdbc.DB2Exception: [IBM][JDBC Driver]
CLI0621E Unsupported JDBC server configuration.
```

- If db2java.zip is prior to FixPak 2:

```
COM.ibm.db2.jdbc.DB2Exception: [IBM][JDBC Driver]
CLI0601E Invalid statement handle or statement is closed.
SQLSTATE=S1000
```

If a mismatch occurs, the JDBC applet server logs one of the following messages in the jdbcerr.log file:

- If the JDBC applet server is at FixPak 2 or later:

```
jdbcFSQLConnect: JDBC Applet Server and client (db2java.zip)
versions do not match. Unable to proceed with connection., einfo= -111
```

- If the JDBC applet server is prior to FixPak 2:

```
jdbcServiceConnection(): Invalid Request Received., einfo= 0
```

Appendix B. Sample Programs

The following should be added to the "Object Linking and Embedding Samples" section:

salarycltvc A Visual C++ DB2 CLI sample that calls the
Visual Basic stored procedure, salarysrv.

SALSVADO A sample OLE automation stored procedure (SALSVADO) and a
SALCLADO client (SALCLADO), implemented in 32-bit Visual Basic and ADO,
that calculates the median salary in table staff2.

CLI Guide and Reference(無中文版)

Binding Database Utilities Using the Run-Time Client

The Run-Time Client cannot be used to bind the database utilities (import, export, reorg, the command line processor) and DB2 CLI bind files to each database before they can be used with that database. You must use the DB2 Administration Client or the DB2 Application Development Client instead.

You must bind these database utilities and DB2 CLI bind files to each database before they can be used with that database. In a network environment, if you are using multiple clients that run on different operating systems, or are at different versions or service levels of DB2, you must bind the utilities once for each operating system and DB2-version combination.

Using Static SQL in CLI Applications

For more information on using static SQL in CLI applications, see the Web page at: <http://www.ibm.com/software/data/db2/udb/staticcli/>

Limitations of JDBC/ODBC/CLI Static Profiling

JDBC/ODBC/CLI static profiling currently targets straightforward applications. It is not meant for complex applications with many functional components and complex program logic during execution.

An SQL statement must have successfully executed for it to be captured in a profiling session. In a statement matching session, unmatched dynamic statements will continue to execute as dynamic JDBC/ODBC/CLI calls.

An SQL statement must be identical character-by-character to the one that was captured and bound to be a valid candidate for statement matching. Spaces are significant: for example, "COL = 1" is considered different than "COL=1". Use parameter markers in place of literals to improve match hits.

When executing an application with pre-bound static SQL statements, dynamic registers that control the dynamic statement behavior will have no effect on the statements that are converted to static.

If an application issues DDL statements for objects that are referenced in subsequent DML statements, you will find all of these statements in the capture file. The JDBC/ODBC/CLI Static Profiling Bind Tool will attempt to bind them. The bind attempt will be successful with DBMSs that support the VALIDATE(RUN) bind option, but it fail with ones that do not. In this case, the application should not use Static Profiling.

The Database Administrator may edit the capture file to add, change, or remove SQL statements, based on application-specific requirements.

ADT Transforms

The following supercedes existing information in the book.

- There is a new descriptor type (smallint) `SQL_DESC_USER_DEFINED_TYPE_CODE`, with values:

```
SQL_TYPE_BASE 0 (this is not a USER_DEFINED_TYPE)
SQL_TYPE_DISTINCT 1
SQL_TYPE_STRUCTURED 2
This value can be queried with either SQLColAttribute
or SQLGetDescField (IRD only).
```

The following attributes are added to obtain the actual type names:

```
SQL_DESC_REFERENCE_TYPE
SQL_DESC_STRUCTURED_TYPE
SQL_DESC_USER_TYPE
```

The above values can be queried using `SQLColAttribute` or `SQLGetDescField` (IRD only).

- Add `SQL_DESC_BASE_TYPE` in case the application needs it. For example, the application may not recognize the structured type, but intends to fetch or insert it, and let other code deal with the details.
- Add a new connection attribute called `SQL_ATTR_TRANSFORM_GROUP` to allow an application to set the transform group (rather than use the SQL "SET CURRENT DEFAULT TRANSFORM GROUP" statement).

- Add a new statement/connection attribute called `SQL_ATTR_RETURN_USER_DEFINED_TYPES` that can be set or queried using `SQLSetConnectAttr`, which causes CLI to return the value `SQL_DESC_USER_DEFINED_TYPE_CODE` as a valid SQL type. This attribute is required before using any of the transforms.

- By default, the attribute is off, and causes the base type information to be returned as the SQL type.

- When enabled, `SQL_DESC_USER_DEFINED_TYPE_CODE` will be returned as the `SQL_TYPE`. The application is expected to check for `SQL_DESC_USER_DEFINED_TYPE_CODE`, and then to retrieve the appropriate type name. This will be available to `SQLColAttribute`, `SQLDescribeCol`, and `SQLGetDescField`.

- The `SQLBindParameter` does not give an error when you bind `SQL_C_DEFAULT`, because there is no code to allow `SQLBindParameter` to specify the type `SQL_USER_DEFINED_TYPE`. The standard default C types will be used, based on the base SQL type flowed to the server. For example:

```
sqlrc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR, SQL_VARCHAR, 30,
                          0, &c2, 30, NULL);
```

Chapter 3. Using Advanced Features

Writing Multi-Threaded Applications

The following should be added to the end of the "Multi-Threaded Mixed Applications" section:

註: It is recommended that you do not use the default stack size, but instead increase the stack size to at least 256 000. DB2 requires a minimum stack size of 256 000

when calling a DB2 function. You must ensure therefore, that you allocate a total stack size that is large enough for both your application and the minimum requirements for a DB2 function call.

Scrollable Cursors

The following information should be added to the "Scrollable Cursors" section:

Server-side Scrollable Cursor Support for OS/390

The UDB client for the Unix, Windows, and OS/2 platforms supports updatable server-side scrollable cursors when run against OS/390 Version 7 databases. To access an OS/390 scrollable cursor on a three-tier environment, the client and the gateway must be running DB2 UDB Version 7.1, FixPak 3 or later.

There are two application enablement interfaces that can access scrollable cursors: ODBC and JDBC. The JDBC interface can only access static scrollable cursors, while the ODBC interface can access static and keyset-driven server-side scrollable cursors.

Cursor Attributes: The table below lists the default attributes for OS/390 Version 7 cursors in ODBC.

Table 23. Default attributes for OS/390 cursors in ODBC

Cursor Type	Cursor Sensitivity	Cursor Updatable	Cursor Concurrency	Cursor Scrollable
forward-only ^a	unspecified	non-updatable	read-only concurrency	non-scrollable
static	insensitive	non-updatable	read-only concurrency	scrollable
keyset-driven	sensitive	updatable	values concurrency	scrollable

^a Forward-only is the default behavior for a scrollable cursor without the FOR UPDATE clause. Specifying FOR UPDATE on a forward-only cursor creates an updatable, lock concurrency, non-scrollable cursor.

Supported Fetch Orientations: All ODBC fetch orientations are supported via the SQLFetchScroll or SQLExtendedFetch interfaces.

Updating the Keyset-Driven Cursor: A keyset-driven cursor is an updatable cursor. The CLI driver appends the FOR UPDATE clause to the query, except when the query is issued as a SELECT ... FOR READ ONLY query, or if the FOR UPDATE clause already exists. The keyset-driven cursor implemented in DB2 for OS/390 is a values concurrency cursor. A values concurrency cursor results in optimistic locking, where locks are not held until an update or delete is attempted. When an update or delete is attempted, the database server compares the previous values the application retrieved to the current values in the underlying table. If the values match, then the update or delete succeeds. If the values do not match, then the operation fails. If failure occurs, the application should query the values again and re-issue the update or delete if it is still applicable.

An application can update a keyset-driven cursor in two ways:

- Issue an UPDATE WHERE CURRENT OF "<cursor name>" or DELETE WHERE CURRENT OF "<cursor name>" using SQLPrepare() with SQLExecute() or SQLExecDirect().

- Use `SQLSetPos()` or `SQLBulkOperations()` to update, delete, or add a row to the result set.

Note: Rows added to a result set via `SQLSetPos()` or `SQLBulkOperations()` are inserted into the table on the server, but are not added to the server's result set. Therefore, these rows are not updatable nor are they sensitive to changes made by other transactions. The inserted rows will appear, however, to be part of the result set, since they are cached on the client. Any triggers that apply to the inserted rows will appear to the application as if they have not been applied. To make the inserted rows updatable, sensitive, and to see the result of applicable triggers, the application must issue the query again to regenerate the result set.

Troubleshooting for Applications Created Before Scrollable Cursor Support:

Since scrollable cursor support is new, some ODBC applications that were working with previous releases of UDB for OS/390 or UDB for Unix, Windows, and OS/2 may encounter behavioral or performance changes. This occurs because before scrollable cursors were supported, applications that requested a scrollable cursor would receive a forward-only cursor. To restore an application's previous behavior before scrollable cursor support, set the following configuration keywords in the `db2cli.ini` file:

Table 24. Configuration keyword values restoring application behavior before scrollable cursor support

Configuration Keyword Setting	Description
<code>PATCH2=6</code>	Returns a message that scrollable cursors (both keyset-driven and static) are not supported. CLI automatically downgrades any request for a scrollable cursor to a forward-only cursor.
<code>DisableKeysetCursor=1</code>	Disables both the server-side and client-side keyset-driven scrollable cursors. This can be used to force the CLI driver to give the application a static cursor when a keyset-driven cursor is requested.
<code>UseServerKeysetCursor=0</code>	Disables the server-side keyset-driven cursor for applications that are using the client-side keyset-driven cursor library to simulate a keyset-driven cursor. Only use this option when problems are encountered with the server-side keyset-driven cursor, since the client-side cursor incurs a large amount of overhead and will generally have poorer performance than a server-side cursor.

Using Compound SQL

The following note is missing from the book:

Any SQL statement that can be prepared dynamically, other than a query, can be executed as a statement inside a compound statement.

Note: Inside Atomic Compound SQL, `savepoint`, `release savepoint`, and `rollback to savepoint` SQL statements are also disallowed. Conversely, Atomic Compound SQL is disallowed in `savepoint`.

Using Stored Procedures

Writing a Stored Procedure in CLI

Following is an undocumented limitation on CLI stored procedures:

If you are making calls to multiple CLI stored procedures, the application must close the open cursors from one stored procedure before calling the next stored procedure. More specifically, the first set of open cursors must be closed before the next stored procedure tries to open a cursor.

CLI Stored Procedures and Autobinding

The following supplements information in the book:

The CLI/ODBC driver will normally autobind the CLI packages the first time a CLI/ODBC application executes SQL against the database, provided the user has the appropriate privilege or authorization. Autobinding of the CLI packages cannot be performed from within a stored procedure, and therefore will not take place if the very first thing an application does is call a CLI stored procedure. Before running a CLI application that calls a CLI stored procedure against a new DB2 database, you must bind the CLI packages once with this command:

UNIX

```
db2 bind <BNDPATH>/@db2cli.lst blocking all
```

Windows and OS/2

```
db2bind "%DB2PATH%\bnd\@db2cli.lst" blocking
```

The recommended approach is to always bind these packages at the time the database is created to avoid autobind at runtime. Autobind can fail if the user does not have privilege, or if another application tries to autobind at the same time.

Chapter 4. Configuring CLI/ODBC and Running Sample Applications

Configuration Keywords

Disregard the last paragraph in the CURRENTFUNCTIONPATH keyword. The correct information is as follows:

This keyword is used as part of the process for resolving unqualified function and stored procedure references that may have been defined in a schema name other than the current user's schema. The order of the schema names determines the order in which the function and procedure names will be resolved. For more information on function and procedure resolution, refer to the SQL Reference.

Chapter 5. DB2 CLI Functions

SQLBindFileToParam - Bind LOB File Reference to LOB Parameter

The last parameter - IndicatorValue - in the SQLBindFileToParam() CLI function is currently documented as "output (deferred)". It should be "input (deferred)".

SQLNextResult - Associate Next Result Set with Another Statement Handle

The following text should be added to Chapter 5, 『DB2 CLI Functions』 :

Purpose

Specification: DB2 CLI 7.x

Syntax

```
SQLRETURN SQLNextResult (SQLHSTMT StatementHandle1  
                        SQLHSTMT StatementHandle2);
```

Function Arguments

Table 25. *SQLNextResult* Arguments

Data Type	Argument	Use	Description
SQLHSTMT	<i>StatementHandle</i>	input	Statement handle.
SQLHSTMT	<i>StatementHandle</i>	input	Statement handle.

Usage

A stored procedure returns multiple result sets by leaving one or more cursors open after exiting. The first result set is always accessed by using the statement handle that called the stored procedure. If multiple result sets are returned, either `SQLMoreResults()` or `SQLNextResult()` can be used to describe and fetch the result set.

`SQLMoreResults()` is used to close the cursor for the first result set and allow the next result set to be processed, whereas `SQLNextResult()` moves the next result set to *StatementHandle2*, without closing the cursor on *StatementHandle1*. Both functions return `SQL_NO_DATA_FOUND` if there are no result sets to be fetched.

Using `SQLNextResult()` allows result sets to be processed in any order once they have been transferred to other statement handles. Mixed calls to `SQLMoreResults()` and `SQLNextResult()` are allowed until there are no more cursors (open result sets) on *StatementHandle1*.

When `SQLNextResult()` returns `SQL_SUCCESS`, the next result set is no longer associated with *StatementHandle1*. Instead, the next result set is associated with *StatementHandle2*, as if a call to `SQLExecDirect()` had just successfully executed a query on *StatementHandle2*. The cursor, therefore, can be described using `SQLNumResultSets()`, `SQLDescribeCol()`, or `SQLColAttribute()`.

After `SQLNextResult()` has been called, the result set now associated with *StatementHandle2* is removed from the chain of remaining result sets and cannot be used again in either `SQLNextResult()` or `SQLMoreResults()`. This means that for 'n' result sets, `SQLNextResult()` can be called successfully at most 'n-1' times.

If `SQLFreeStmt()` is called with the `SQL_CLOSE` option, or `SQLFreeHandle()` is called with *HandleType* set to `SQL_HANDLE_STMT`, all pending result sets on this statement handle are discarded.

`SQLNextResult()` returns `SQL_ERROR` if *StatementHandle2* has an open cursor or *StatementHandle1* and *StatementHandle2* are not on the same connection. If any errors or warnings are returned, `SQLError()` must always be called on *StatementHandle1*.

註: `SQLMoreResults()` also works with a parameterized query with an array of input parameter values specified with `SQLParamOptions()` and `SQLBindParameter()`. `SQLNextResult()`, however, does not support this.

Return Codes

- `SQL_SUCCESS`
- `SQL_SUCCESS_WITH_INFO`
- `SQL_STILL_EXECUTING`
- `SQL_ERROR`
- `SQL_INVALID_HANDLE`
- `SQL_NO_DATA_FOUND`

Diagnostics

Table 26. `SQLNextResult` `SQLSTATEs`

SQLSTATE	Description	Explanation
40003 08S01	Communication Link failure.	The communication link between the application and data source failed before the function completed.
58004	Unexpected system failure.	Unrecoverable system error.
HY001	Memory allocation failure.	DB2 CLI is unable to allocate the memory required to support execution or completion of the function.
HY010	Function sequence error.	The function was called while in a data-at-execute (<code>SQLParamData()</code> , <code>SQLPutData()</code>) operation. <i>StatementHandle2</i> has an open cursor associated with it. The function was called while within a <code>BEGIN COMPOUND</code> and <code>END COMPOUND</code> SQL operation.
HY013	Unexpected memory handling error.	DB2 CLI was unable to access the memory required to support execution or completion of the function.
HYT00	Time-out expired.	The time-out period expired before the data source returned the result set. Time-outs are only supported on non-multitasking systems such as Windows 3.1 and Macintosh System 7. The time-out period can be set using the <code>SQL_ATTR_QUERY_TIMEOUT</code> attribute for <code>SQLSetConnectAttr()</code> .

Restrictions

Only `SQLMoreResults()` can be used for parameterized queries.

References

- "SQLMoreResults - Determine If There Are More Result Sets" on page 535
- "Returning Result Sets from Stored Procedures" on page 120

Appendix D. Extended Scalar Functions

Date and Time Functions

The following functions are missing from the Date and Time Functions section of Appendix D "Extended Scalar Functions":

DAYOFWEEK_ISO(*date_exp*)

Returns the day of the week in *date_exp* as an integer value in the range 1-7, where 1 represents Monday. Note the difference between this function and the DAYOFWEEK() function, where 1 represents Sunday.

WEEK_ISO(*date_exp*)

Returns the week of the year in *date_exp* as an integer value in the range of 1-53. Week 1 is defined as the first week of the year to contain a Thursday. Therefore, Week1 is equivalent to the first week that contains Jan 4, since Monday is considered to be the first day of the week.

Note that WEEK_ISO() differs from the current definition of WEEK(), which returns a value up to 54. For the WEEK() function, Week 1 is the week containing the first Saturday. This is equivalent to the week containing Jan. 1, even if the week contains only one day.

DAYOFWEEK_ISO() and WEEK_ISO() are automatically available in a database created in Version 7. If a database was created prior to Version 7, these functions may not be available. To make DAYOFWEEK_ISO() and WEEK_ISO() functions available in such a database, use the **db2updb** system command. For more information about **db2updb**, see the "Command Reference" section in these Release Notes.

Appendix K. Using the DB2 CLI/ODBC/JDBC Trace Facility

The sections within this appendix have been updated. See the 『Traces』 chapter in the *Troubleshooting Guide* for the most up-to-date information on this trace facility.

訊息參考手冊

取得訊息與 SQLSTATE 說明

命令行處理器的可用說明是包含訊息與 SQLSTATE 值的新的 (或更新的) 說明，在訊息參考手冊中尚未提供。

若要從命令行處理器顯示訊息，請在作業系統的命令提示下輸入下列指令：

```
db2 "? XXXnnnnn"
```

其中 *XXX* 代表訊息的字首；而 *nnnnn* 代表訊息碼。

例如：db2 "? SQL30081" 會顯示關於 SQL30081 訊息的說明。

若要從命令行處理器顯示 SQLSTATE 文字，請在作業系統的命令提示下輸入下列指令：

```
db2 "? XXXXX"
```

其中 *XXXXX* 代表 SQLSTATE 值。

例如：db2 "? 428F1" 會顯示 SQLSTATE 428F1 的文字。

在 DB2 Connect 變更的 SQLCODE 重新對映

DB2 Connect 的預設 SQLCODE 重新對映在版本 7.2 中已變更。當主電腦資料庫傳回 SQLCODE 值 -567，DB2 Connect 會在傳回 SQLCODE 值到 DB2 從屬站之前重新對映 SQLCODE 值到 -551。

新的與變更的訊息

以下列示的是 DB2 訊息參考手冊 版本 7.1 以後變更的訊息碼。在使用 DB2 時若您接收到這些訊息的其中一則，您將會接收到更正的新訊息；但是，這訊息會無法對應到訊息參考手冊中的資訊。

呼叫層次介面 (CLI) 訊息

CLI0645E

CLI0646E

CLI0647E

DB2 訊息

DB21086I

DB210060E

DB210061E

DB210062E

DB210113E

DB210114E
DB210115E
DB210116E
DB210117E
DB210118E
DB210120E
DB210121E
DB210200I
DB210201I

DBI 訊息

DBI1172E
DBI1793W
DBI1794E
DBI1795E
DBI1796W
DBI1797I

資料倉儲中心 (DWC) 訊息

DWC0000I
DWC03504E
DWC08900E
DWC08901E
DWC08902E
DWC08903E
DWC08904E
DWC08907C
DWC08908C
DWC08909C
DWC08910E
DWC08911E
DWC08912E
DWC08913E
DWC08914E
DWC08915E
DWC08917E
DWC08919I
DWC08930E
DWC08931E
DWC08932E

DWC08933E
DWC08934E
DWC08935E
DWC08936W
DWC08937I
DWC08938I
DWC08939I
DWC08940I
DWC08941I
DWC08960I
DWC08961I
DWC08962I
DWC08963I
DWC08964I
DWC08965I
DWC08966E
DWC08967E
DWC08968E
DWC13239E
DWC13300E
DWC13301E
DWC13302E
DWC13304E
DWC13603E
DWC13700E
DWC13701E
DWC13702E
DWC13703E
DWC13705E
DWC13706E
DWC13707E

SQL 訊息

SQL0017N
SQL0056N
SQL0057N
SQL0058N
SQL0097N
SQL0224N
SQL0225N

SQL0227N
SQL0228N
SQL0231W
SQL0243N
SQL0244N
SQL0270N
SQL0301N
SQL0303N
SQL0336N
SQL0348N
SQL0349N
SQL0357N
SQL0358N
SQL0368N
SQL0408N
SQL0423N
SQL0590N
SQL0670N
SQL0845N
SQL0846N
SQL1179W
SQL1186N
SQL1550N
SQL1551N
SQL1552N
SQL1553N
SQL1704N
SQL2077W
SQL2078N
SQL2417N
SQL2426N
SQL2571N
SQL2572N
SQL2573N
SQL2574N
SQL2575N
SQL2576N
SQL4942N
SQL5012N
SQL6583N

SQL20005N
SQL20117N
SQL20121N
SQL20133N
SQL20134N
SQL20135N
SQL20143N
SQL20144N
SQL20145N
SQL20146N
SQL20147N
SQL20148N
SQL20153N
SQL21000N

更正 SQLSTATES

表 27.

42630	SQLSTATE 或 SQLCODE 變數在此環境定義是無效的。
42631	表示式必須指定於 SQL 函數的 RETURN 陳述式中。
42632	在 SQL 函數或方法中必須要有 RETURN 陳述式。
428F2	在 SQL 程序的 RETURN 陳述式中必須要有整數表示式。
560B7	對多列 INSERT 來說，每一列的 NEXTVAL 順序表示式用法必須相同。

SQL Reference(無中文版)

SQL Reference is Provided in One PDF File

The "Using the DB2 Library" appendix in each book indicates that the SQL Reference is available in PDF format as two separate volumes. This is incorrect.

Although the printed book appears in two volumes, and the two corresponding form numbers are correct, there is only one PDF file, and it contains both volumes. The PDF file name is db2s0x70.

Chapter 3. Language Elements

Naming Conventions and Implicit Object Name Qualifications

Add the following note to this section in Chapter 3:

The following names, when used in the context of SQL Procedures, are restricted to the characters allowed in an ordinary identifier, even if the names are delimited:

- condition-name
- label
- parameter-name
- procedure-name
- SQL-variable-name
- statement-name

DATALINK Assignments

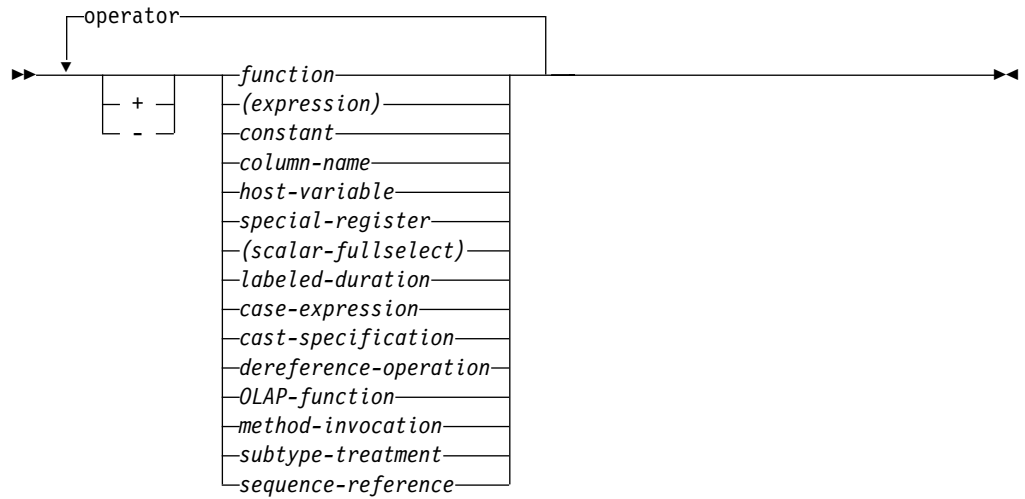
A paragraph in this section has been changed to the following:

Note that the size of a URL parameter or function result is the same on both input or output and is bound by the length of the DATALINK column. However, in some cases the URL value returned has an access token attached. In situations where this is possible, the output location must have sufficient storage space for the access token and the length of the DATALINK column. Hence, the actual length of the comment and URL in its fully expanded form provided on input should be restricted to accommodate the output storage space. If the restricted length is exceeded, this error is raised.

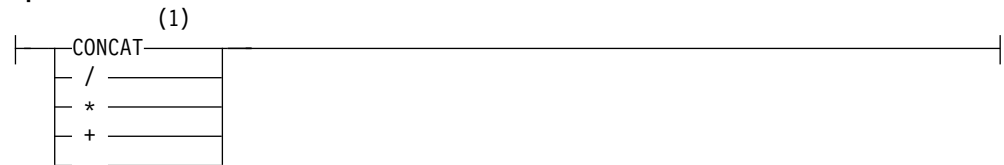
Expressions

Syntax Diagram

The syntax diagram has changed:



operator:



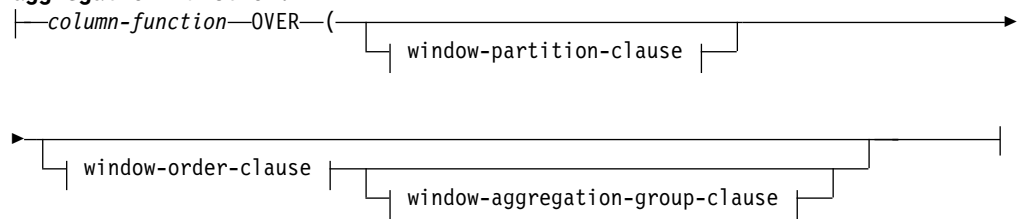
Note:

1 || may be used as a synonym for CONCAT.

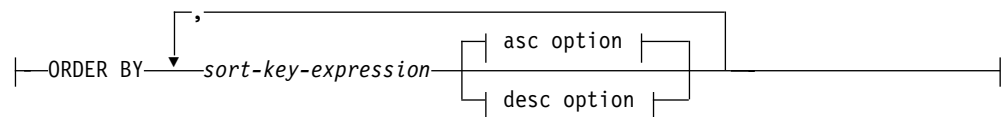
OLAP Functions

The following represents a correction to the "OLAP Functions" section under "Expressions" in Chapter 3.

aggregation-function:



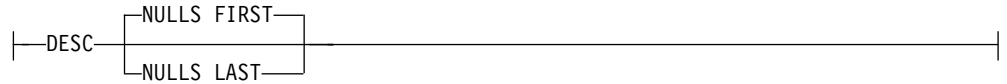
window-order-clause:



asc option:



desc option:



window-aggregation-group-clause:



group-end:



In the window-order-clause description:

NULLS FIRST

The window ordering considers null values before all non-null values in the sort order.

NULLS LAST

The window ordering considers null values after all non-null values in the sort order.

In the window-aggregation-group-clause description:

window-aggregation-group-clause

The aggregation group of a row R is a set of rows, defined relative to R in the ordering of the rows of R's partition. This clause specifies the aggregation group. If this clause is not specified, the default is the same as RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW, providing a cumulative aggregation result.

ROWS

Indicates the aggregation group is defined by counting rows.

RANGE

Indicates the aggregation group is defined by an offset from a sort key.

group-start

Specifies the starting point for the aggregation group. The aggregation group end is the current row. Specification of the group-start clause is equivalent to a group-between clause of the form "BETWEEN group-start AND CURRENT ROW".

group-between

Specifies the aggregation group start and end based on either ROWS or RANGE.

group-end

Specifies the ending point for the aggregation group. The aggregation group start

is the current row. Specification of the *group-end* clause is equivalent to a *group-between* clause of the form "BETWEEN CURRENT ROW AND *group-end*".

UNBOUNDED PRECEDING

Includes the entire partition preceding the current row. This can be specified with either ROWS or RANGE. Also, this can be specified with multiple sort-key-expressions in the window-order-clause.

UNBOUNDED FOLLOWING

Includes the entire partition following the current row. This can be specified with either ROWS or RANGE. Also, this can be specified with multiple sort-key-expressions in the window-order-clause.

CURRENT ROW

Specifies the start or end of the aggregation group based on the current row. If ROWS is specified, the current row is the aggregation group boundary. If RANGE is specified, the aggregation group boundary includes the set of rows with the same values for the *sort-key-expressions* as the current row. This clause cannot be specified in *group-bound2* if *group-bound1* specifies *value* FOLLOWING.

value **PRECEDING**

Specifies either the range or number of rows preceding the current row. If ROWS is specified, then *value* is a positive integer indicating a number of rows. If RANGE is specified, then the data type of *value* must be comparable to the type of the sort-key-expression of the window-order-clause. There can only be one sort-key-expression, and the data type of the sort-key-expression must allow subtraction. This clause cannot be specified in *group-bound2* if *group-bound1* is CURRENT ROW or *value* FOLLOWING.

value **FOLLOWING**

Specifies either the range or number of rows following the current row. If ROWS is specified, then *value* is a positive integer indicating a number of rows. If RANGE is specified, then the data type of *value* must be comparable to the type of the sort-key-expression of the window-order-clause. There can only be one sort-key-expression, and the data type of the sort-key-expression must allow addition.

Sequence Reference

The following information should be added to the end of the Expressions section (after "Subtype Treatment").

sequence-reference:

nextval-expression
prevval-expression

nextval-expression:

NEXTVAL FOR *sequence-name*

prevval-expression:

PREVVAL FOR *sequence-name*

NEXTVAL FOR *sequence-name*

A NEXTVAL expression returns the next value for the sequence specified by *sequence-name*.

PREVVAL FOR *sequence-name*

A PREVVAL expression returns the most recently generated value for the specified sequence for a previous statement within the current session. This value can be repeatedly referenced using PREVVAL expressions specifying the name of the sequence. There may be multiple instances of PREVVAL expressions specifying the same sequence name within a single statement and they all return the same value.

A PREVVAL expression can only be used if a NEXTVAL expression specifying the same sequence name has already been referenced in the current user session (in the current or a previous transaction) (SQLSTATE 51035).

Note:

- A new sequence number is generated when a NEXTVAL expression specifies the name of the sequence. However, if there are multiple instances of a NEXTVAL expression specifying the same sequence name within a query, the counter for the sequence is incremented only once for each row of the result.
- The most recently generated value for a sequence can be repeatedly referenced using a PREVVAL expression specifying the name of the sequence. There may be multiple instances of a PREVVAL expression specifying the same sequence name within a single statement.
- The same sequence number can be used as a unique key value in two separate tables by referencing the sequence number with a NEXTVAL expression for the first row (this generates the sequence value), and a PREVVAL expression for the other rows (this instance of PREVVAL refers to the sequence value generated by the NEXTVAL expression in the previous statement), as shown below:

```
INSERT INTO order(orderno, custno)
VALUES (NEXTVAL FOR order_seq, 123456);
```

```
INSERT INTO line_item (orderno, partno, quantity)
VALUES (PREVVAL FOR order_seq, 987654, 1);
```

- Examples of where NEXTVAL and PREVVAL expressions can be specified are:
 - select-statement or SELECT INTO statement: within the select-clause as long as the statement does not contain a DISTINCT keyword, a GROUP BY clause, an ORDER BY clause, a UNION keyword, an INTERSECT keyword, or EXCEPT keyword
 - INSERT statement: within a VALUES clause
 - INSERT statement: within the select-clause of the fullselect
 - UPDATE statement: within the select-clause of the fullselect of an expression in the SET clause (either searched or positioned UPDATE statement)
 - VALUES INTO statement: within the select-clause of the fullselect of an expression

- Examples of where NEXTVAL and PREVVAl expressions cannot be specified (SQLSTATE 428F9) are:
 - join condition of a full outer join
 - DEFAULT value for a column in a CREATE TABLE or ALTER TABLE statement
 - generated column definition in a CREATE TABLE or ALTER TABLE statement
 - condition of a CHECK constraint
 - CREATE TRIGGER statement
 - CREATE VIEW statement
 - CREATE METHOD statement
 - CREATE FUNCTION statement.
- In addition, a NEXTVAL expression cannot be specified (SQLSTATE 428F9) in :
 - CASE expression
 - parameter list of an aggregate function
 - subquery
 - SELECT statement that contains a DISTINCT operator
 - join condition of join
 - GROUP BY clause of a SELECT statement
 - SELECT statement that is combined with another SELECT statement using the UNION, INTERSECT, or EXCEPT set operator
 - nested table expression
 - parameter list of a table function
 - WHERE clause of a SELECT, DELETE, or UPDATE statement
 - ORDER BY clause
 - parameter list of a CALL statement.
- When a value is generated for a sequence, that value is consumed, and the next time that a value is needed, a new value will be generated. This is true even when the statement containing the NEXTVAL expression fails.
- If an INSERT statement includes a NEXTVAL expression in the VALUES list for the column, and if some error occurs at some point during the execution of the INSERT (it could be a problem in generating the next sequence value, or a problem with the value for another column), then an insertion failure occurs, and the value generated for the sequence is considered to be consumed. In some cases, reissuing the same INSERT statement might lead to success.

For example, consider an error that is the result of the existence of a unique index for the column for which NEXTVAL was used and the sequence value generated already exists in the index. It is possible that the next value generated for the sequence is a value that does not exist in the index and so the subsequent INSERT would succeed.
- If in generating a value for a sequence, the maximum value for the sequence is exceeded (or the minimum value for a descending sequence) and cycles are not permitted, then an error occurs (SQLSTATE 23522). In this case, the user could

ALTER the sequence to extend the range of acceptable values, or enable cycles for the sequence, or DROP and CREATE a new sequence with a different data type that has a larger range of values.

For example, a sequence may have been defined with a data type of SMALLINT, and eventually the sequence runs out of assignable values. To redefine the sequence as INTEGER, you would need to drop and recreate the sequence with the new definition.

- A reference to PREVVAL in a select statement of a cursor refers to a value that was generated for the specified sequence prior to the opening of the cursor. However, closing the cursor can affect the values returned by PREVVAL for the specified sequence in subsequent statements, or even for the same statement in the event that the cursor is reopened. This would be the case when the select statement of the cursor included a reference to NEXTVAL for the same sequence name.

Examples:: These examples assume that there is a table called "order" and that a sequence called "order_seq" is created as follows:

```
CREATE SEQUENCE order_seq
  START WITH 1
  INCREMENT BY 1
  NOMAXVALUE
  NOCYCLE
  CACHE 24
```

- Some examples of how to generate an "order_seq" sequence number with a NEXTVAL expression for the sequence created above:

```
INSERT INTO order(orderno, custno)
  VALUES (NEXTVAL FOR order_seq, 123456);
```

or,

```
UPDATE order
  SET orderno = NEXTVAL FOR order_seq
  WHERE custno = 123456;
```

or,

```
VALUES NEXTVAL FOR order_seq INTO :hv_seq;
```

Chapter 4. Functions

Enabling the New Functions and Procedures

Version 7 FixPaks deliver new SQL built-in scalar functions. Refer to the *SQL Reference* updates for a description of these new functions. The new functions are not automatically enabled on each database when the database server code is upgraded to the new service level. To enable these new functions, the system administrator must issue the command **db2updv7**, specifying each database at the server. This command makes an entry in the database that ensures that database objects created prior to executing this command use existing function signatures that may match the new function signatures.

For information on enabling the MQSeries functions (those defined in the MQDB2 schema), see 第173頁的『MQSeries』.

Scalar Functions

ABS or ABSVAL

► ABS ABSVAL (*expression*) ►

The schema is SYSIBM.

This function was first available in FixPak 2 of Version 7.1.

註: The SYSFUN version of the ABS (or ABSVAL) function continues to be available.

Returns the absolute value of the argument.

The argument is an expression that returns a value of any built-in numeric data type.

The result of the function has the same data type and length attribute as the argument. If the argument can be null or the database is configured with DFT_SQLMATHWARN set to yes, then the result can be null; if the argument is null, the result is the null value.

For example:

ABS(-51234)

returns an INTEGER with a value of 51234.

DECRYPT_BIN and DECRYPT_CHAR

► DECRYPT_BIN DECRYPT_CHAR ►

► (*—encrypted-data* , —password-string-expression) ►

The schema is SYSIBM.

This function was first available in FixPak 3 of Version 7.1.

The DECRYPT_BIN and DECRYPT_CHAR functions return a value that is the result of decrypting *encrypted-data*. The password used for decryption is either the *password-string-expression* value or the ENCRYPTION PASSWORD value (as assigned using the SET ENCRYPTION PASSWORD statement). The DECRYPT_BIN and DECRYPT_CHAR functions can only decrypt values that were encrypted using the ENCRYPT function (SQLSTATE 428FE).

encrypted-data

An expression that returns a CHAR FOR BIT DATA or VARCHAR FOR BIT DATA value that is a complete, encrypted data string that was encrypted using the ENCRYPT function.

password-string-expression

An expression that returns a CHAR or VARCHAR value with at least 6 bytes and

no more than 127 bytes (SQLSTATE 428FC). This should be the same password used to encrypt the data or decryption will result in an error (SQLSTATE 428FD). If the value of the password argument is null or not provided, the data will be decrypted using the ENCRYPTION PASSWORD value, which must have been set for the session (SQLSTATE 51039).

The result of the DECRYPT_BIN function is VARCHAR FOR BIT DATA. The result of the DECRYPT_CHAR function is VARCHAR. If the *encrypted-data* included a hint, the hint is not returned by the function. The length attribute of the result is the length attribute of the data type of *encrypted-data* minus 8 bytes. The actual length of the value returned by the function will match the length of the original string that was encrypted. If the *encrypted-data* includes bytes beyond the encrypted string, these bytes are not returned by the function. If the first argument can be null, the result can be null; if the first argument is null, the result is the null value.

If the data is decrypted on a different system using a code page other than the code page in which the encryption took place, it is possible that expansion may occur when converting the decrypted value to the database code page. In such situations, the *encrypted-data* value should be cast to a VARCHAR string with a larger number of bytes.

Also see 『ENCRYPT』 and page 353 『GETHINT』 for additional information on using this function.

Examples:

Example 1: This example uses the ENCRYPTION PASSWORD value to hold the encryption password.

```
SET ENCRYPTION PASSWORD = 'Ben123';
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832');
SELECT DECRYPT_CHAR(SSN)
FROM SSN;
```

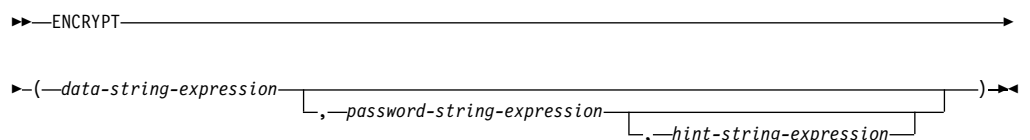
The value returned is '289-46-8832'.

Example 2: This example explicitly passes the encryption password.

```
SELECT DECRYPT_CHAR(SSN, 'Ben123')
FROM SSN;
```

The value returned is '289-46-8832'.

ENCRYPT



The schema is SYSIBM.

This function was first available in FixPak 3 of Version 7.1.

The ENCRYPT function returns a value that is the result of encrypting *data-string-expression*. The password used for encryption is either the *password-string-expression* value or the ENCRYPTION PASSWORD value (as assigned using the SET ENCRYPTION PASSWORD statement).

data-string-expression

An expression that returns a CHAR or VARCHAR value to be encrypted. The length attribute for the data type of *data-string-expression* is limited to 32663 without a *hint-string-expression* argument and 32631 when the *hint-string-expression* argument is specified (SQLSTATE 42815).

password-string-expression

An expression that returns a CHAR or VARCHAR value with at least 6 bytes and no more than 127 bytes (SQLSTATE 428FC). The value represents the password used to encrypt the *data-string-expression*. If the value of the password argument is null or not provided, the data will be encrypted using the ENCRYPTION PASSWORD value, which must have been set for the session (SQLSTATE 51039).

hint-string-expression

An expression that returns a CHAR or VARCHAR value up to 32 bytes that will help data owners remember passwords (for example, 'Ocean' as a hint to remember 'Pacific'). If a hint value is given, the hint is embedded into the result and can be retrieved using the GETHINT function. If this argument is null or not provided, no hint will be embedded in the result.

The result data type of the function is VARCHAR FOR BIT DATA.

The length attribute of the result is:

- When the optional hint parameter is specified, the length attribute of the non-encrypted data + 8 bytes + the number of bytes to the next 8 byte boundary + 32 bytes for the hint length.
- With no hint parameter, the length attribute of the non-encrypted data + 8 bytes + the number of bytes to the next 8 byte boundary.

If the first argument can be null, the result can be null; if the first argument is null, the result is the null value.

Notice that the encrypted result is longer than the *data-string-expression* value. Therefore, when assigning encrypted values, ensure that the target is declared with sufficient size to contain the entire encrypted value.

Notes:

- **Encryption Algorithm:** The internal encryption algorithm used is RC2 block cipher with padding, the 128-bit secret key is derived from the password using a MD2 message digest.
- **Encryption Passwords and Data:** It is the user's responsibility to perform password management. Once the data is encrypted only the password used to encrypt it can be used to decrypt it (SQLSTATE 428FD). Be careful when using CHAR variables to set password values as they may be padded with blanks. The encrypted result may contain null terminator and other non-printable characters.

Table Column Definition: When defining columns and types to contain encrypted data always calculate the length attribute as follows. For encrypted data with no hint: Maximum length of the non-encrypted data + 8 bytes + the number of bytes to the next 8 byte boundary = encrypted data column length.

For encrypted data with embedded hint :

Maximum length of the non-encrypted data + 8 bytes + the number of bytes to the next 8 byte boundary + 32 bytes for the hint length = encrypted data column length.

Any assignment or cast to a length shorter than the suggested data length may result in failed decryption in the future and **lost data**. Blanks are valid encrypted data values that may be truncated when stored in a column that is too short.

Sample Column Length Calculations

Maximum length of non-encrypted data	6 bytes
8 bytes	8 bytes
Number of bytes to the next 8 byte boundary	2 bytes

Encrypted data column length	16 bytes
Maximum length of non-encrypted data	32 bytes
8 bytes	8 bytes
Number of bytes to the next 8 byte boundary	8 bytes

Encrypted data column length	48 bytes

- **Administration of encrypted data:** Encrypted data can only be decrypted on servers that support the decryption functions that correspond to the ENCRYPT function. Hence, replication of columns with encrypted data should only be done to servers that support the DECRYPT_BIN or DECRYPT_CHAR function.

Also see page 350 『DECRYPT_BIN and DECRYPT_CHAR』 and 『GETHINT』 for additional information on using this function.

Examples:

Example 1: This example uses the ENCRYPTION PASSWORD value to hold the encryption password.

```
SET ENCRYPTION PASSWORD = 'Ben123';
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832');
```

Example 2: This example explicitly passes the encryption password.

```
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832', 'Ben123', '');
```

Example 3: The hint 'Ocean' is stored to help the user remember the encryption password of 'Pacific'.

```
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832', 'Pacific', 'Ocean');
```

GETHINT

▶—GETHINT—(—encrypted-data—)—————▶

The schema is SYSIBM.

This function was first available in FixPak 3 of Version 7.1.

The GETHINT function will return the password hint if one is found in the *encrypted-data*. A password hint is a phrase that will help data owners remember passwords (For example, 'Ocean' as a hint to remember 'Pacific').

encrypted-data

An expression that returns a CHAR FOR BIT DATA or VARCHAR FOR BIT DATA value that is a complete, encrypted data string that was encrypted using the ENCRYPT function (SQLSTATE 428FE).

The result of the function is VARCHAR(32). The result can be null; if the hint parameter was not added to the *encrypted-data* by the ENCRYPT function or the first argument is null, the result is the null value.

Also see page 350 『DECRYPT_BIN and DECRYPT_CHAR』 and page 351 『ENCRYPT』 for additional information on using this function.

Example:

In this example the hint 'Ocean' is stored to help the user remember the encryption password 'Pacific'.

```
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832', 'Pacific','ocean');
SELECT GETHINT(SSN)
FROM EMP;
```

The value returned is 'Ocean'.

IDENTITY_VAL_LOCAL

▶—IDENTITY_VAL_LOCAL—(—)—————▶

The schema is SYSIBM.

This procedure was first available in FixPak 3 of Version 7.1.

The IDENTITY_VAL_LOCAL function is a non-deterministic function that returns the most recently assigned value for an identity column, where the assignment occurred as a result of a single row INSERT statement using a VALUES clause. The function has no input parameters.

The result is a DECIMAL(31,0), regardless of the actual data type of the corresponding identity column.

The value returned by the function is the value assigned to the identity column of the table identified in the most recent single row INSERT statement. The INSERT statement must be made using a VALUES clause on a table containing an identity column. Also, the INSERT statement must be issued at the same level¹ (that is, the value is available locally at the level it was assigned, until it is replaced by the next assigned value).

1. A new level is initiated each time a trigger, function, or stored procedure is invoked.

The assigned value is either a value supplied by the user (if the identity column is defined as GENERATED BY DEFAULT), or an identity value generated by DB2.

The function returns a null value in the following situations:

- When a single row INSERT statement with a VALUES clause has not been issued at the current processing level for a table containing an identity column.
- When a COMMIT or ROLLBACK of a unit of work has occurred since the most recent INSERT statement that assigned a value².

The result of the function is not affected by the following:

- A single row INSERT statement with a VALUES clause for a table without an identity column.
- A multiple row INSERT statement with a VALUES clause.
- An INSERT statement with a fullselect.
- A ROLLBACK TO SAVEPOINT statement.

Notes:

- Expressions in the VALUES clause of an INSERT statement are evaluated prior to the assignments for the target columns of the INSERT statement. Thus, an invocation of an IDENTITY_VAL_LOCAL function inside the VALUES clause of an INSERT statement will use the most recently assigned value for an identity column from a previous INSERT statement. The function returns the null value if no previous single row INSERT statement with a VALUES clause for a table containing an identity column has been executed within the same level as the IDENTITY_VAL_LOCAL function.
- The identity column value of the table for which the trigger is defined can be determined within a trigger, by referencing the trigger transition variable for the identity column.
- The result of invoking the IDENTITY_VAL_LOCAL function from within the trigger condition of an insert trigger is a null value.
- It is possible that multiple before or after insert triggers exist for a table. In this case each trigger is processed separately, and identity values assigned by one triggered action are not available to other triggered actions using the IDENTITY_VAL_LOCAL function. This is true even though the multiple triggered actions are conceptually defined at the same level.
- It is not generally recommended to use the IDENTITY_VAL_LOCAL function in the body of a before insert trigger. The result of invoking the IDENTITY_VAL_LOCAL function from within the triggered action of a before insert trigger is the null value. The value for the identity column of the table for which the trigger is defined cannot be obtained by invoking the IDENTITY_VAL_LOCAL function within the triggered action of a before insert trigger. However, the value for the identity column can be obtained in the triggered action, by referencing the trigger transition variable for the identity column.

2. Unless the automatic commit is turned off, interfaces that automatically commit after each statement will return a null value when the function is invoked in separate statements.

- The result of invoking the `IDENTITY_VAL_LOCAL` function from within the triggered action of an after insert trigger³ is the value assigned to an identity column of the table identified in the most recent single row `INSERT` statement invoked in the same triggered action that had a `VALUES` clause for a table containing an identity column. If a single row `INSERT` statement with a `VALUES` clause for a table containing an identity column was not executed within the same triggered action, prior to the invocation of the `IDENTITY_VAL_LOCAL` function, then the function returns a null value.
- Since the results of the `IDENTITY_VAL_LOCAL` function are not deterministic, the result of an invocation of the `IDENTITY_VAL_LOCAL` function within the `SELECT` statement of a cursor can vary for each `FETCH` statement.
- The assigned value is the value actually assigned to the identity column (that is, the value that would be returned on a subsequent `SELECT` statement). This value is not necessarily the value provided in the `VALUES` clause of the `INSERT` statement, or a value generated by DB2. The assigned value could be a value specified in a `SET` transition variable statement, within the body of a before insert trigger, for a trigger transition variable associated with the identity column.
- The value returned by the function is unpredictable following a failed single row `INSERT` with a `VALUES` clause into a table with an identity column. The value may be the value that would have been returned from the function had it been invoked prior to the failed `INSERT`, or it may be the value that would have been assigned had the `INSERT` succeeded. The actual value returned depends on the point of failure and is therefore unpredictable.

Examples:

Example 1: Set the variable `IVAR` to the value assigned to the identity column in the `EMPLOYEE` table. If this insert is the first into the `EMPLOYEE` table, then `IVAR` would have a value of 1.

```
CREATE TABLE EMPLOYEE
(EMPNO  INTEGER GENERATED ALWAYS AS IDENTITY,
 NAME   CHAR(30),
 SALARY DECIMAL(5,2),
 DEPTNO SMALLINT)
```

Example 2: An `IDENTITY_VAL_LOCAL` function invoked in an `INSERT` statement returns the value associated with the previous single row `INSERT` statement, with a `VALUES` clause for a table with an identity column. Assume for this example that there are two tables, `T1` and `T2`. Both `T1` and `T2` have an identity column named `C1`. DB2 generates values in sequence, starting with 1, for the `C1` column in table `T1`, and values in sequence, starting with 10, for the `C1` column in table `T2`.

```
CREATE TABLE T1
(C1 INTEGER GENERATED ALWAYS AS IDENTITY,
 C2 INTEGER),
CREATE TABLE T2
(C1 DECIMAL(15,0) GENERATED BY DEFAULT AS IDENTITY
 (START WITH 10),
```

3. This applies to both `FOR EACH ROW` and `FOR EACH STATEMENT` after insert triggers.

```

        C2 INTEGER),
INSERT INTO T1 (C2) VALUES (5),
INSERT INTO T1 (C2) VALUES (6),
SELECT * FROM T1

```

which gives a result of:

C1	C2
1	5
2	6

and now, declaring the function for the variable IVAR:

```

VALUES IDENTITY_VAL_LOCAL() INTO :IVAR

```

At this point, the IDENTITY_VAL_LOCAL function would return a value of 2 in IVAR, because that was the value most recently assigned by DB2. The following INSERT statement inserts a single row into T2, where column C2 gets a value of 2 from the IDENTITY_VAL_LOCAL function.

```

INSERT INTO T2 (C2) VALUES (IDENTITY_VAL_LOCAL());
SELECT * FROM T2
WHERE C1 = DECIMAL(IDENTITY_VAL_LOCAL(),15,0)

```

returning a result of:

C1	C2
10.	2

Invoking the IDENTITY_VAL_LOCAL function after this insert results in a value of 10, which is the value generated by DB2 for column C1 of T2.

In a nested environment involving a trigger, use the IDENTITY_VAL_LOCAL function to retrieve the identity value assigned at a particular level, even though there might have been identity values assigned at lower levels. Assume that there are three tables, EMPLOYEE, EMP_ACT, and ACCT_LOG. There is an after insert trigger defined on EMPLOYEE that results in additional inserts into the EMP_ACT and ACCT_LOG tables.

```

CREATE TABLE EMPLOYEE
(EMPNO SMALLINT GENERATED ALWAYS AS IDENTITY (START WITH 1000),
NAME CHAR(30),
SALARY DECIMAL(5,2),
DEPTNO SMALLINT);

CREATE TABLE EMP_ACT
(ACNT_NUM SMALLINT GENERATED ALWAYS AS IDENTITY (START WITH 1),
EMPNO SMALLINT);

CREATE TABLE ACCT_LOG
(ID SMALLINT GENERATED ALWAYS AS IDENTITY (START WITH 100),
ACNT_NUM SMALLINT,
EMPNO SMALLINT);

CREATE TRIGGER NEW_HIRE
AFTER INSERT ON EMPLOYEE
REFERENCING NEW AS NEW_EMP
FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
INSERT INTO EMP_ACT (EMPNO)

```

```

VALUES (NEW_EMP.EMPNO);
INSERT INTO ACCT_LOG (ACNT_NUM EMPNO)
VALUES (IDENTITY_VAL_LOCAL(), NEW_EMP.EMPNO);
END

```

The first triggered INSERT statement inserts a row into the EMP_ACT table. This INSERT statement uses a trigger transition variable for the EMPNO column of the EMPLOYEE table, to indicate that the identity value for the EMPNO column of the EMPLOYEE table is to be copied to the EMPNO column of the EMP_ACT table. The IDENTITY_VAL_LOCAL function could not be used to obtain the value assigned to the EMPNO column of the EMPLOYEE table. This is because an INSERT statement has not been issued at this level of the nesting, and as such, if the IDENTITY_VAL_LOCAL function were invoked in the VALUES clause of the INSERT for EMP_ACT, then it would return a null value. This INSERT statement for the EMP_ACT table also results in the generation of a new identity column value for the ACNT_NUM column.

A second triggered INSERT statement inserts a row into the ACCT_LOG table. This statement invokes the IDENTITY_VAL_LOCAL function to indicate that the identity value assigned to the ACNT_NUM column of the EMP_ACT table in the previous INSERT statement in the triggered action is to be copied to the ACNT_NUM column of the ACCT_LOG table. The EMPNO column is assigned the same value as the EMPNO column of EMPLOYEE table.

From the invoking application (that is, the level at which the INSERT to EMPLOYEE is issued), set the variable IVAR to the value assigned to the EMPNO column of the EMPLOYEE table by the original INSERT statement.

```

INSERT INTO EMPLOYEE (NAME, SALARY, DEPTNO)
VALUES ('Rupert', 989.99, 50);

```

The contents of the three tables after processing the original INSERT statement and all of the triggered actions are:

```

SELECT EMPNO, SUBSTR(NAME,10) AS NAME, SALARY, DEPTNO
FROM EMPLOYEE;

```

EMPNO	NAME	SALARY	DEPTNO
1000	Rupert	989.99	50

```

SELECT ACNT_NUM, EMPNO
FROM EMP_ACT;

```

ACNT_NUM	EMPNO
1	1000

```

SELECT * FROM ACCT_LOG;

```

ID	ACNT_NUM	EMPNO
100	1	1000

The result of the IDENTITY_VAL_LOCAL function is the most recently assigned value for an identity column at the same nesting level. After processing the original INSERT statement and all of the triggered actions, the IDENTITY_VAL_LOCAL function returns a value of 1000, because this is the value assigned to the EMPNO column of the EMPLOYEE table. The following VALUES statement results in setting IVAR to 1000.

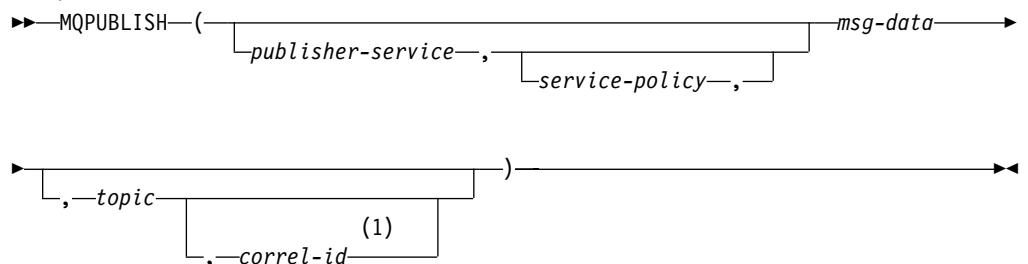
The insert into the EMP_ACT table (which occurred after the insert into the EMPLOYEE table and at a lower nesting level) has no affect on what is returned by this invocation of the IDENTITY_VAL_LOCAL function.

```
VALUES IDENTITY_VAL_LOCAL() INTO :IVAR;
```

LCASE and UCASE (Unicode)

In a Unicode database, the entire repertoire of Unicode characters is uppercase (or lowercase) based on the Unicode properties of these characters. Double-wide versions of ASCII characters, as well as Roman numerals, now convert to upper or lower case correctly.

MQPUBLISH



Note:

1 The *correl-id* cannot be specified unless a *service* and a *policy* are previously defined.

The schema is MQDB2.

The MQPUBLISH function publishes data to MQSeries. This function requires the installation of either MQSeries Publish/Subscribe or MQSeries Integrator. Please consult www.ibm.com/software/MQSeries for further details.

The MQPUBLISH function publishes the data contained in *msg-data* to the MQSeries publisher specified in *publisher-service*, and using the quality of service policy defined by *service-policy*. An optional topic for the message can be specified, and an optional user-defined message correlation identifier may also be specified. The function returns a value of '1' if successful or a '0' if unsuccessful.

publisher-service

A string containing the logical MQSeries destination where the message is to be sent. If specified, the *publisher-service* must refer to a publisher Service Point defined in the AMT.XML repository file. A service point is a logical end-point from which a message is sent or received. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface for further details. If *publisher-service* is not specified, then the DB2.DEFAULT.PUBLISHER will be used. The maximum size of *publisher-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy to be used in handling of this message. If specified, the *service-policy* must refer to a Policy defined in the AMT.XML repository file. A Service Policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and

message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 characters.

msg-data

A string expression containing the data to be sent via MQSeries. The maximum size is 4000 characters.

topic

A string expression containing the topic for the message publication. If no topic is specified, none will be associated with the message. The maximum size of *topic* is 40 characters. Multiple topics can be specified in one string (up to 40 characters long). Each topic must be separated by a colon. For example, "t1:t2:the third topic" indicates that the message is associated with all three topics: t1, t2, and "the third topic".

correl-id

An optional string expression containing a correlation identifier to be associated with this message. The *correl-id* is often specified in request and reply scenarios to associate requests with replies. If not specified, no correlation id will be added to the message. The maximum size of *correl-id* is 24 characters.

Examples

Example 1: This example publishes the string "Testing 123" to the default publisher service (DB2.DEFAULT.PUBLISHER) using the default policy (DB2.DEFAULT.POLICY). No correlation identifier or topic is specified for the message.

VALUES MQPUBLISH('Testing 123')

Example 2: This example publishes the string "Testing 345" to the publisher service "MYPUBLISHER" under the topic "TESTS". The default policy is used and no correlation identifier is specified.

VALUES MQPUBLISH('MYPUBLISHER','Testing 345', 'TESTS')

Example 3: This example publishes the string "Testing 678" to the publisher service "MYPUBLISHER" using the policy "MYPOLICY" with a correlation identifier of "TEST1". The message is published with topic "TESTS".

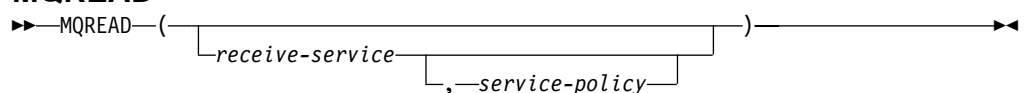
VALUES MQPUBLISH('MYPUBLISHER','MYPOLICY','Testing 678','TESTS','TEST1')

Example 4: This example publishes the string "Testing 901" to the publisher service "MYPUBLISHER" under the topic "TESTS" using the default policy (DB2.DEFAULT.POLICY) and no correlation identifier.

VALUES MQPUBLISH('Testing 901','TESTS')

All examples return the value '1' if successful.

MQREAD



The schema is MQDB2.

The MQREAD function returns a message from the MQSeries location specified by *receive-service*, using the quality of service policy defined in *service-policy*. Executing this operation does not remove the message from the queue associated with *receive-service*, but instead returns the message at the head of the queue. The return value is a VARCHAR(4000) containing the message. If no messages are available to be returned, a NULL is returned.

receive-service

A string containing the logical MQSeries destination from where the message is to be received. If specified, the *receive-service* must refer to a Service Point defined in the AMT.XML repository file. A service point is a logical end-point from where a message is sent or received. Service points definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface for further details. If *receive-service* is not specified, then the DB2.DEFAULT.SERVICE will be used. The maximum size of *receive-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy used in handling this message. If specified, the *service-policy* must refer to a Policy defined in the AMT.XML repository file. A Service Policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 characters.

Examples:

Example 1: This example reads the message at the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY).

VALUES MQREAD()

Example 2: This example reads the message at the head of the queue specified by the service "MYSERVICE" using the default policy (DB2.DEFAULT.POLICY).

VALUES MQREAD('MYSERVICE')

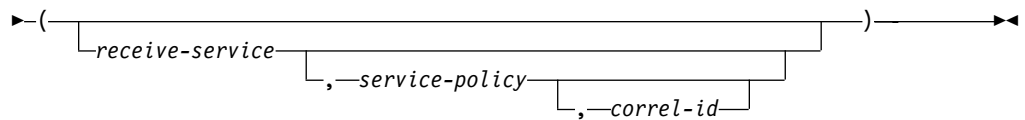
Example 3: This example reads the message at the head of the queue specified by the service "MYSERVICE", and using the policy "MYPOLICY".

VALUES MQREAD('MYSERVICE', 'MYPOLICY')

All of these examples return the contents of the message as a VARCHAR(4000) if successful. If no messages are available, then a NULL is returned.

MQRECEIVE

▶—MQRECEIVE—————▶



The schema is MQDB2.

The MQRECEIVE function returns a message from the MQSeries location specified by *receive-service*, using the quality of service policy *service-policy*. Performing this operation removes the message from the queue associated with *receive-service*. If the *correl-id* is specified, then the first message with a matching correlation identifier will be returned. If *correl-id* is not specified, then the message at the head of the queue will be returned. The return value is a VARCHAR(4000) containing the message. If no messages are available to be returned, a NULL is returned.

receive-service

A string containing the logical MQSeries destination from which the message is received. If specified, the *receive-service* must refer to a Service Point defined in the AMT.XML repository file. A service point is a logical end-point from which a message is sent or received. Service points definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface for further details. If *receive-service* is not specified, then the DB2.DEFAULT.SERVICE is used. The maximum size of *receive-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy to be used in the handling of this message. If specified, the *service-policy* must refer to a Policy defined in the AMT XML repository file⁴. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY is used. The maximum size of *service-policy* is 48 characters.

correl-id

A string containing an optional correlation identifier to be associated with this message. The *correl-id* is often specified in request and reply scenarios to associate requests with replies. If not specified, no correlation id will be specified. The maximum size of *correl-id* is 24 characters.

Examples:

Example 1: This example receives the message at the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY).

VALUES MQRECEIVE()

Example 2: This example receives the message at the head of the queue specified by the service "MYSERVICE" using the default policy (DB2.DEFAULT.POLICY).

VALUES MQRECEIVE('MYSERVICE')

4. A Service Policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details.

Example 3: This example receives the message at the head of the queue specified by the service "MYSERVICE" using the policy "MYPOLICY".

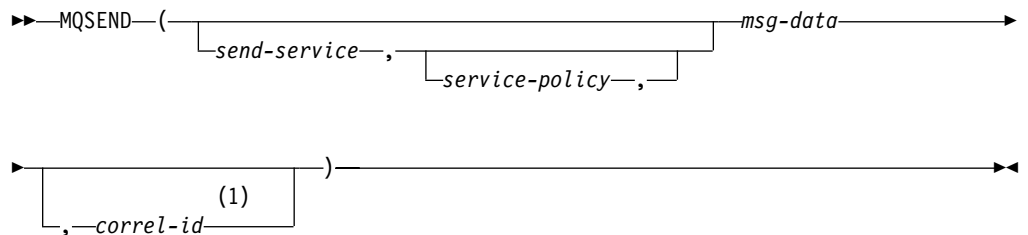
```
VALUES MQRECEIVE('MYSERVICE', 'MYPOLICY')
```

Example 4: This example receives the first message with a correlation id that matches '1234' from the head of the queue specified by the service "MYSERVICE" using the policy "MYPOLICY".

```
VALUES MQRECEIVE('MYSERVICE', 'MYPOLICY', '1234')
```

All these examples return the contents of the message as a VARCHAR(4000) if successful. If no messages are available, a NULL will be returned.

MQSEND



註:

- 1 The *correl-id* cannot be specified unless a *service* and a *policy* are previously defined.

The schema is MQDB2.

The MQSEND function sends the data contained in *msg-data* to the MQSeries location specified by *send-service*, using the quality of service policy defined by *service-policy*. An optional user defined message correlation identifier may be specified by *correl-id*. The function returns a value of '1' if successful or a '0' if unsuccessful.

msg-data

A string expression containing the data to be sent via MQSeries. The maximum size is 4000 characters.

send-service

A string containing the logical MQSeries destination where the message is to be sent. If specified, the *send-service* refers to a service point defined in the AMT.XML repository file. A service point is a logical end-point from which a message may be sent or received. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface manual for further details. If *send-service* is not specified, then the value of DB2.DEFAULT.SERVICE is used. The maximum size of *send-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy used in handling of this message. If specified, the *service-policy* must refer to a service policy defined in the AMT XML repository file. A Service Policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual

for further details. If *service-policy* is not specified, then a default value of DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 characters.

correl-id

An optional string containing a correlation identifier associated with this message. The *correl-id* is often specified in request and reply scenarios to associate requests with replies. If not specified, no correlation id will be sent. The maximum size of *correl-id* is 24 characters.

Examples:

Example 1: This example sends the string "Testing 123" to the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY), with no correlation identifier.

VALUES MQSEND('Testing 123')

Example 2: This example sends the string "Testing 345" to the service "MYSERVICE", using the policy "MYPOLICY", with no correlation identifier.

VALUES MQSEND('MYSERVICE','MYPOLICY','Testing 345')

Example 3: This example sends the string "Testing 678" to the service "MYSERVICE", using the policy "MYPOLICY", with correlation identifier "TEST3".

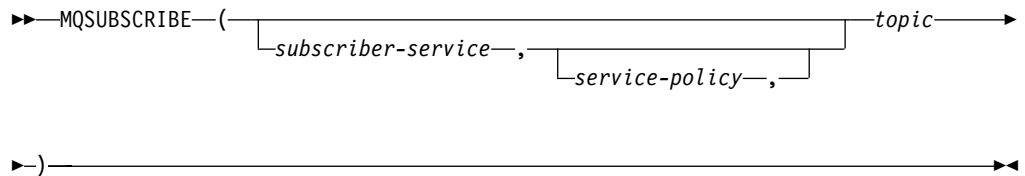
VALUES MQSEND('MYSERVICE','MYPOLICY','Testing 678','TEST3')

Example 4: This example sends the string "Testing 901" to the service "MYSERVICE", using the default policy (DB2.DEFAULT.POLICY), and no correlation identifier.

VALUES MQSEND('MYSERVICE','Testing 901')

All examples return a scalar value of '1' if successful.

MQSUBSCRIBE



The schema is MQDB2.

The MQSUBSCRIBE function is used to register interest in MQSeries messages published on a specified topic. The *subscriber-service* specifies a logical destination for messages that match the specified topic. Messages that match *topic* will be placed on the queue defined by *subscriber-service* and can be read or received through a subsequent call to MQREAD, MQRECEIVE, MQREADALL, or MQRECEIVEALL. This function requires the installation and configuration of an MQSeries based publish and subscribe system, such as MQSeries Integrator or MQSeries Publish/Subscribe. See www.ibm.com/software/MQSeries for further details.

The function returns a value of '1' if successful or a '0' if unsuccessful. Successfully executing this function will cause the publish and subscribe server to forward messages matching the topic to the service point defined by *subscriber-service*.

subscriber-service

A string containing the logical MQSeries subscription point to where messages matching *topic* will be sent. If specified, the *subscriber-service* must refer to a Subscribers Service Point defined in the AMT.XML repository file. Service points definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface manual for further details. If *subscriber-service* is not specified, then the DB2.DEFAULT.SUBSCRIBER will be used instead. The maximum size of *subscriber-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy to be used in handling the message. If specified, the *service-policy* must refer to a Policy defined in the AMT.XML repository file. A Service Policy defines a set of quality of service options to be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used instead. The maximum size of *service-policy* is 48 characters.

topic

A string defining the types of messages to receive. Only messages published with the specified topics will be received by this subscription. Multiple subscriptions may coexist. The maximum size of *topic* is 40 characters. Multiple topics can be specified in one string (up to 40 characters long). Each topic must be separated by a colon. For example, "t1:t2:the third topic" indicates that the message is associated with all three topics: t1, t2, and "the third topic".

Examples:

Example 1: This example registers an interest in messages containing the topic "Weather". The default *subscriber-service* (DB2.DEFAULT.SUBSCRIBER) is registered as the subscriber and the default *service-policy* (DB2.DEFAULT.POLICY) specifies the quality of service.

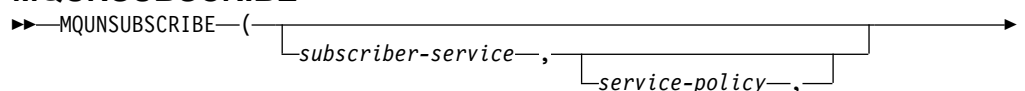
VALUES MQSUBSCRIBE('Weather')

Example 2: This example demonstrates a subscriber registering interest in messages containing "Stocks". The subscriber registers as "PORTFOLIO-UPDATES" with policy "BASIC-POLICY".

VALUES MQSUBSCRIBE('PORTFOLIO-UPDATES', 'BASIC-POLICY', 'Stocks')

All examples return a scalar value of '1' if successful.

MQUNSUBSCRIBE



The schema is MQDB2.

The MQUNSUBSCRIBE function is used to unregister an existing message subscription. The *subscriber-service*, *service-policy*, and *topic* are used to identify which subscription is cancelled. This function requires the installation and configuration of an MQSeries based publish and subscribe system, such as MQSeries Integrator or MQSeries Publish/Subscribe. See www.ibm.com/software/MQSeries for further details.

The function returns a value of '1' if successful or a '0' if unsuccessful. The result of successfully executing this function is that the publish and subscribe server will remove the subscription defined by the given parameters. Messages with the specified *topic* will no longer be sent to the logical destination defined by *subscriber-service*.

subscriber-service

If specified, the *subscriber-service* must refer to a Subscribers Service Point defined in the AMT.XML repository file. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface manual for further details. If *subscriber-service* is not specified, then the DB2.DEFAULT.SUBSCRIBER value is used. The maximum size of *subscriber-service* is 48 characters.

service-policy

If specified, the *service-policy* must refer to a Policy defined in the AMT.XML repository file. A Service Policy defines a set of quality of service options to be applied to this messaging operation. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 characters.

topic

A string specifying the subject of messages that are not to be received. The maximum size of *topic* is 40 characters. Multiple topics can be specified in one string (up to 40 characters long). Each topic must be separated by a colon. For example, "t1:t2:the third topic" indicates that the message is associated with all three topics: t1, t2, and "the third topic".

Examples:

Example 1: This example cancels an interest in messages containing the topic "Weather". The default subscriber-service (DB2.DEFAULT.SUBSCRIBER) is registered as the unsubscriber and the default service-policy (DB2.DEFAULT.POLICY) specifies the quality of service.

VALUES MQUNSUBSCRIBE('Weather')

Example 2: This example demonstrates a subscriber cancelling an interest in messages containing "Stocks". The subscriber is registered as "PORTFOLIO-UPDATES" with policy "BASIC-POLICY".

VALUES MQUNSUBSCRIBE('PORTFOLIO-UPDATES', 'BASIC-POLICY', 'Stocks')

These examples return a scalar value of '1' if successful and a scalar value of '0' if unsuccessful.

MULTIPLY_ALT

►MULTIPLY_ALT(*exact_numeric_expression*, *exact_numeric_expression*)◄

The schema is SYSIBM.

This function was first available in FixPak 2 of Version 7.1.

The MULTIPLY_ALT scalar function returns the product of the two arguments as a decimal value. It is provided as an alternative to the multiplication operator, especially when the sum of the precisions of the arguments exceeds 31.

The arguments can be any built-in exact numeric data type (DECIMAL, BIGINT, INTEGER, or SMALLINT).

The result of the function is a DECIMAL. The precision and scale of the result are determined as follows, using the symbols *p* and *s* to denote the precision and scale of the first argument, and the symbols *p'* and *s'* to denote the precision and scale of the second argument.

- The precision is $\text{MIN}(31, p + p')$
- The scale is:
 - 0 if the scale of both arguments is 0
 - $\text{MIN}(31, s+s')$ if $p+p'$ is less than or equal to 31
 - $\text{MAX}(\text{MIN}(3, s+s'), 31-(p-s+p'-s'))$ if $p+p'$ is greater than 31.

The result can be null if at least one argument can be null or the database is configured with DFT_SQLMATHWARN set to yes; the result is the null value if one of the arguments is null.

The MULTIPLY_ALT function is a better choice than the multiplication operator when performing decimal arithmetic where a scale of at least 3 is needed and the sum of the precisions exceeds 31. In these cases, the internal computation is performed so that overflows are avoided. The final result is then assigned to the result data type using truncation where necessary to match the scale. Note that overflow of the final result is still possible when the scale is 3.

The following is a sample comparing the result types using MULTIPLY_ALT and the multiplication operator.

Type of argument 1	Type of argument 2	Result using MULTIPLY_ALT	Result using multiplication operator
DECIMAL(31,3)	DECIMAL(15,8)	DECIMAL(31,3)	DECIMAL(31,11)
DECIMAL(26,23)	DECIMAL(10,1)	DECIMAL(31,19)	DECIMAL(31,24)
DECIMAL(18,17)	DECIMAL(20,19)	DECIMAL(31,29)	DECIMAL(31,31)
DECIMAL(16,3)	DECIMAL(17,8)	DECIMAL(31,9)	DECIMAL(31,11)

Type of argument 1	Type of argument 2	Result using MULTIPLY_ALT	Result using multiplication operator
DECIMAL(26,5)	DECIMAL(11,0)	DECIMAL(31,3)	DECIMAL(31,5)
DECIMAL(21,1)	DECIMAL(15,1)	DECIMAL(31,2)	DECIMAL(31,2)

Example:

Multiply two values where the data type of the first argument is DECIMAL(26, 3) and the data type of the second argument is DECIMAL(9,8). The data type of the result is DECIMAL(31,7).

```
values multiply_alt(98765432109876543210987.654,5.43210987)
1
-----
536504678578875294857887.5277415
```

Note that the complete product of these two numbers is 536504678578875294857887.52774154498 but the last 4 digits were truncated to match the scale of the result data type. Using the multiplication operator with the same values results in an arithmetic overflow since the result data type is DECIMAL(31,11) and the result value has 24 digits left of the decimal, but the result data type only supports 20 digits.

REC2XML

►► REC2XML (—*decimal-constant*—, —*format-string*—, —*row-tag-string*—) ►►

►) ►►

 ↓
 , —*column-name*—

The schema is SYSIBM.

The REC2XML function returns a string formatted with XML tags and containing column names and column values.

decimal-constant

The expansion factor for replacing column value characters. The decimal value must be greater than 0.0 and less than or equal to 6.0 (SQLSTATE 42820).

The *decimal-constant* value is used to calculate the result length of the function. For every column with a character data type, the length attribute of the column is multiplied by this expansion factor before it is added in to the result length.

To specify no expansion, use a value of 1.0. Specifying a value less than 1.0 reduces the calculated result length. If the actual length of the result string is greater than the calculated result length of the function, then an error is raised (SQLSTATE 22001).

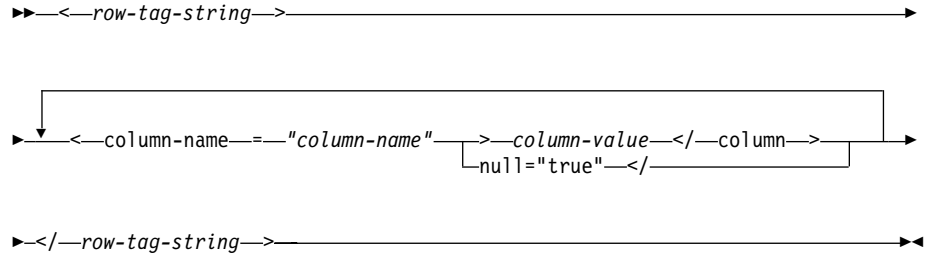
format-string

The string constant that specifies which format the function is to use during execution.

The *format-string* is case-sensitive, so the following values must be specified in uppercase to be recognized.

COLATTVAL or COLATTVAL_XML

These formats return a string with columns as attribute values.



Column names may or may not be valid XML attribute values. For those column names which are not valid XML attribute values, character replacement is performed on the column name before it is included in the result string.

Column values may or may not be valid XML element values. If the *format-string* COLATTVAL is specified, for those column values which are not valid XML element values, character replacement is performed on the column value before it is included in the result string. If the *format-string* COLATTVAL_XML is specified, character replacement is not performed on column values (note that character replacement is still performed on column names).

row-tag-string

A string constant that specifies the tag used for each row.

If an empty string is specified, then a value of 'row' is assumed.

If a string of one or more blank characters is specified, then no beginning *row-tag-string* or ending *row-tag-string* (including the angle bracket delimiters) will appear in the result string.

column-name

A qualified or unqualified name of a table column. The column must have one of the following data types (SQLSTATE 42815):

- numeric (SMALLINT, INTEGER, BIGINT, DECIMAL, NUMERIC, REAL, DOUBLE)
- character string (CHAR, VARCHAR)⁵
- datetime (DATE, TIME, TIMESTAMP)
- a user-defined type based on one of the above types

The same column name cannot be specified more than once (SQLSTATE 42734).

The result of the function is VARCHAR. The maximum length is 32672 bytes (SQLSTATE 54006).

Consider the following invocation:

5. A character string with a subtype of BIT DATA is not allowed.

REC2XML (dc, fs, rt, c₁, c₂, ..., c_n)

If the value of fs is "COLATTVAL" or "COLATTVAL_XML"

the result is the same as the following expression:

```
'<' CONCAT rt CONCAT '>' CONCAT y1 CONCAT y2 CONCAT ... CONCAT yn  
CONCAT '</' CONCAT rt CONCAT '>'
```

where y_n is equivalent to:

```
'<column name="' CONCAT xvcn CONCAT vn
```

and v_n is equivalent to:

```
'">' CONCAT rn CONCAT '</column>'
```

if the column is not null, and

```
'" null="true"/>'
```

if the column value is null.

xvc_n is equivalent to a string representation of the column name of c_n, where any characters appearing in table 29 are replaced with the corresponding representation. This ensures that the resulting string is a valid XML attribute or element value token.

r_n is equivalent to a string representation as indicated in table 28.

Result Column Values:

Based on the data type of the column and the actual *format-string* specified, the column values from the table may be transformed before being concatenated into the result string. The following table shows the transformations done on the column values.

Table 28. Column Values String Result

Data type of c _n	r _n
CHAR, VARCHAR	The value is a string. If the <i>format-string</i> does not end in the characters "_XML", then each character in c _n is replaced with the corresponding replacement representation from Table 29, as indicated. The length attribute is: dc * the length attribute of c _n .
SMALLINT, INTEGER, BIGINT, DECIMAL, NUMERIC, REAL, DOUBLE	The value is LTRIM(RTRIM(CHAR(c _n))). The length attribute is the result length of CHAR(c _n). The decimal character is always the period character.
DATE	The value is CHAR(c _n , ISO). The length attribute is the result length of CHAR(c _n , ISO).
TIME	The value is CHAR(c _n , JIS). The length attribute is the result length of CHAR(c _n , JIS)
TIMESTAMP	The value is CHAR(c _n). The length attribute is the result length of CHAR(c _n).

Character Replacement:

Depending on the value specified for the *format-string*, certain characters in column names and column values will be replaced to ensure that the column names form valid XML attribute values and the column values form valid XML element values.

Table 29. Character Replacements for XML Attribute Values and Element Values

< is replaced by <
> is replaced by >
" is replaced by "
& is replaced by &
' is replaced by '

Examples:

- Using the DEPARTMENT table, format the department table row, except the DEPTNAME and LOCATION columns, for department 'D01' into a string of valid XML. Since the data does not contain any of the characters which require replacement, the expansion factor will be 1.0 (no expansion). Also note that the MGRNO value is null for this row.

```
SELECT REC2XML (1.0, 'COLATTVAL', '', DEPTNO, MGRNO, ADMRDEPT)
FROM DEPARTMENT
WHERE DEPTNO = 'D01'
```

This example returns the following VARCHAR(117) string:

```
<row>
  <column name="DEPTNO">D01</column>
  <column name="MGRNO" null="true"/>
  <column name="ADMRDEPT">A00</column>
</row>
```

Note: REC2XML does not insert new line characters in the output. The above example output is formatted for the sake of readability.

- A 5-day university schedule introduces a class with the name '&43<FIE' to a table called CL_SCHED, with a new format for the CLASS_CODE column. Using the REC2XML function, this example formats an XML string with this new class data, except for the class end time.

The length attribute for the REC2XML call (see below) with an expansion factor of 1.0 would be 128 (11 for the '<row>' and '</row>' overhead, 21 for the column names, 75 for the '<column name=', '>', '</column>' and double quotes, 7 for the CLASS_CODE data, 6 for the DAY data, and 8 for the STARTING data). Since the '&' and '<' characters will be replaced, an expansion factor of 1.0 will not be sufficient. The length attribute of the function will need to support an increase from 7 to 14 characters for the new format CLASS_CODE data.

However, since it is known that the DAY value will never be more than 1 digit long, an extra 5 is calculated into the length that will never be used. Therefore, the expansion only needs to handle an increase of 2. Since CLASS_CODE is the only character string column in the argument list, this is the only column value to which the expansion factor applies. To get an increase of 2 for the length, an expansion factor of 9/7 (approximately 1.2857) would be needed. An expansion factor of 1.3 will be used.

```

SELECT REC2XML (1.3, 'COLATTVAL', 'record', CLASS_CODE, DAY, STARTING)
FROM CL_SCHED
WHERE CLASS_CODE = '&43<FIE'

```

Returns the following string:

```

<record>
  <column name="CLASS_CODE">&43<FIE</column>
  <column name="DAY">5</column>
  <column name="STARTING">06:45:00</column>
</record>

```

Note: REC2XML does not insert new line characters in the output. The above example output is formatted for the sake of readability.

- This example shows characters replaced in a column name.

```

SELECT REC2XML (1.3, 'COLATTVAL', '', Class, "time<noon")
FROM (SELECT Class_code, Starting
      FROM Cl_sched
      WHERE Starting < '12:00:00')
AS Early (Class, "time<noon")

```

This query returns the following string:

```

<row>
  <column name="CLASS">&43<FIE</column>
  <column name="time<noon">06:45:00</column>
</row>

```

Note: REC2XML does not insert new line characters in the output. The above example output is formatted for the sake of readability.

- Assume that new rows have been added to the EMP_RESUME table, storing the resumes as strings of valid XML. The COLATTVAL_XML *format-string* is used so character replacement will not be performed. None of the resumes are more than 3500 characters in length. The following query could be used to select the XML version of the resumes from the EMP_RESUME table and format it into an XML document fragment.

```

SELECT REC2XML (1.0, 'COLATTVAL_XML', '', EMPNO, RESUME_XML)
FROM (SELECT EMPNO, CAST(RESUME as VARCHAR(3500)) as RESUME_XML
      FROM EMP_RESUME
      WHERE RESUME_FORMAT = 'XML')
AS EMP_RESUME_XML

```

ROUND

▶—ROUND—(*expression1*, *expression2*)—▶

The schema is SYSIBM.

This function was first available in FixPak 2 of Version 7.1.

註: The SYSFUN version of the ROUND function continues to be available.

The ROUND function returns *expression1* rounded to *expression2* places to the right of the decimal point if *expression2* is positive or to the left of the decimal point if *expression2* is zero or negative.

If *expression1* is positive, a value of 5 is rounded to the next higher positive number. For example, ROUND(3.5,0) = 4. If *expression1* is negative, a value of 5 is rounded to the next lower negative number. For example, ROUND(-3.5,0) = -4.

expression1

An expression that returns a value of any built-in numeric data type.

expression2

An expression that returns a small or large integer. When the value of *expression2* is not negative, it specifies rounding to that number of places to the right of the decimal separator. When the value of *expression2* is negative, it specifies rounding to the absolute value of *expression2* places to the left of the decimal separator.

If *expression2* is not negative, *expression1* is rounded to the absolute value of *expression2* number of places to the right of the decimal point. If the value of *expression2* is greater than the scale of *expression1* then the value is unchanged except that the result value has a precision that is larger by 1. For example, ROUND(748.58,5) = 748.58 where the precision is now 6 and the scale remains 2.

If *expression2* is negative, *expression1* is rounded to the absolute value of *expression2*+1 number of places to the left of the decimal point.

If the absolute value of a negative *expression2* is larger than the number of digits to the left of the decimal point, the result is 0. For example, ROUND(748.58,-4) = 0.

The data type and length attribute of the result are the same as the data type and length attribute of the first argument, except that the precision is increased by one if the *expression1* is DECIMAL or NUMERIC and the precision is less than 31. For example, an argument with a data type of DECIMAL(5,2) results in DECIMAL(6,2). An argument with a data type of DECIMAL(31,2) results in DECIMAL(31,2). The scale is the same as the scale of the first argument.

If either argument can be null or the database is configured with DFT_SQLMATHWARN set to yes, the result can be null. If either argument is null, the result is the null value.

Examples:: Calculate the number 873.726 rounded to 2, 1, 0, -1, -2, -3, and -4 decimal places respectively.

```
VALUES (ROUND(873.726, 2),
        ROUND(873.726, 1),
        ROUND(873.726, 0),
        ROUND(873.726,-1),
        ROUND(873.726,-2),
        ROUND(873.726,-3),
        ROUND(873.726,-4) )
```

This example returns:

1	2	3	4	5	6	7
873.730	873.700	874.000	870.000	900.000	1000.000	0.000

Calculate a both positive and negative numbers.

```
VALUES (ROUND(3.5, 0),
        ROUND(3.1, 0),
        ROUND(-3.1, 0),
        ROUND(-3.5,0) )
```

This example returns:

```
  1    2    3    4
  ---  ---  ---  ---
  4.0  3.0 -3.0 -4.0
```

WEEK_ISO

Change the description of this function to the following:

The schema is SYSFUN.

Returns the week of the year of the argument as an integer value in the range 1-53. The week starts with Monday and always includes 7 days. Week 1 is the first week of the year to contain a Thursday, which is equivalent to the first week containing January 4. It is therefore possible to have up to 3 days at the beginning of a year appear in the last week of the previous year. Conversely, up to 3 days at the end of a year may appear in the first week of the next year.

The argument must be a date, timestamp, or a valid character string representation of a date or timestamp that is neither a CLOB nor a LONG VARCHAR.

The result of the function is INTEGER. The result can be null; if the argument is null, the result is the null value.

Example:

The following list shows examples of the result of WEEK_ISO and DAYOFWEEK_ISO.

DATE	WEEK_ISO	DAYOFWEEK_ISO
1997-12-28	52	7
1997-12-31	1	3
1998-01-01	1	4
1999-01-01	53	5
1999-01-04	1	1
1999-12-31	52	5
2000-01-01	52	6
2000-01-03	1	1

Table Functions

MQREADALL

```
►► MQREADALL ( [receive-service] [, -service-policy] [num-rows] )
►►
```

The schema is MQDB2.

The MQREADALL function returns a table containing the messages and message metadata from the MQSeries location specified by *receive-service*, using the quality of service policy *service-policy*. Performing this operation does not remove the messages from the queue associated with *receive-service*.

If *num-rows* is specified, then a maximum of *num-rows* messages will be returned. If *num-rows* is not specified, then all available messages will be returned. The table returned contains the following columns:

- MSG - a VARCHAR(4000) column containing the contents of the MQSeries message.
- CORRELID - a VARCHAR(24) column holding a correlation ID used to relate messages.
- TOPIC - a VARCHAR(40) column holding the topic that the message was published with, if available.
- QNAME - a VARCHAR(48) column holding the queue name where the message was received.
- MSGID - a CHAR(24) column holding the assigned MQSeries unique identifier for this message.
- MSGFORMAT - a VARCHAR(8) column holding the format of the message, as defined by MQSeries. Typical strings have a MQSTR format.

receive-service

A string containing the logical MQSeries destination from which the message is read. If specified, the *receive-service* must refer to a service point defined in the AMT.XML repository file. A service point is a logical end-point from which a message is sent or received. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface for further details. If *receive-service* is not specified, then the DB2.DEFAULT.SERVICE will be used. The maximum size of *receive-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy used in the handling of this message. If specified, the *service-policy* refers to a Policy defined in the AMT XML repository file. A service policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 characters.

num-rows

A positive integer containing the maximum number of messages to be returned by the function.

Examples:

Example 1: This example receives all the messages from the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). The messages and all the metadata are returned as a table.

```
SELECT *  
FROM table (MQREADALL()) T
```

Example 2: This example receives all the messages from the head of the queue specified by the service MYSERVICE, using the default policy (DB2.DEFAULT.POLICY). Only the MSG and CORRELID columns are returned.

```
SELECT T.MSG, T.CORRELID
FROM table (MQREADALL('MYSERVICE')) T
```

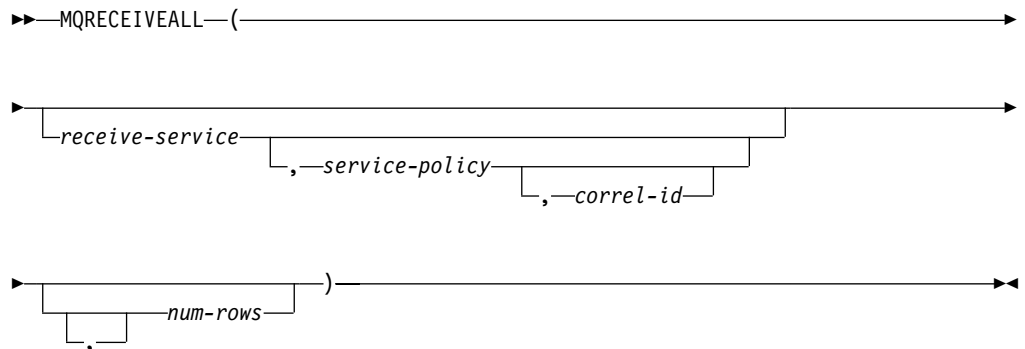
Example 3: This example reads the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). Only messages with a CORRELID of '1234' are returned. All columns are returned.

```
SELECT *
FROM table (MQREADALL()) T
WHERE T.CORRELID = '1234'
```

Example 4: This example receives the first 10 messages from the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). All columns are returned.

```
SELECT *
FROM table (MQREADALL(10)) T
```

MQRECEIVEALL



The schema is MQDB2.

The MQRECEIVEALL function returns a table containing the messages and message metadata from the MQSeries location specified by *receive-service*, using the quality of service policy *service-policy*. Performing this operation removes the messages from the queue associated with *receive-service*.

If a *correl-id* is specified, then only those messages with a matching correlation identifier will be returned. If *correl-id* is not specified, then the message at the head of the queue will be returned.

If *num-rows* is specified, then a maximum of *num-rows* messages will be returned. If *num-rows* is not specified, then all available messages are returned. The table returned contains the following columns:

- MSG - a VARCHAR(4000) column containing the contents of the MQSeries message.
- CORRELID - a VARCHAR(24) column holding a correlation ID used to relate messages.
- TOPIC - a VARCHAR(40) column holding the topic that the message was published with, if available.

- QNAME - a VARCHAR(48) column holding the queue name where the message was received.
- MSGID - a CHAR(24) column holding the assigned MQSeries unique identifier for this message.
- MSGFORMAT - a VARCHAR(8) column holding the format of the message, as defined by MQSeries. Typical strings have a MQSTR format.

receive-service

A string containing the logical MQSeries destination from which the message is received. If specified, the *receive-service* must refer to a service point defined in the AMT.XML repository file. A service point is a logical end-point from which a message is sent or received. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface manual for further details. If *receive-service* is not specified, then the DB2.DEFAULT.SERVICE will be used. The maximum size of *receive-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy used in the handling of this message. If specified, the *service-policy* refers to a Policy defined in the AMT XML repository file. A service policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 characters.

correl-id

An optional string containing a correlation identifier associated with this message. The *correl-id* is often specified in request and reply scenarios to associate requests with replies. If not specified, no correlation id is specified. The maximum size of *correl-id* is 24 characters.

num-rows

A positive integer containing the maximum number of messages to be returned by the function.

Examples:

Example 1: This example receives all the messages from the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). The messages and all the metadata are returned as a table.

```
SELECT *
FROM table (MQRECEIVEALL()) T
```

Example 2: This example receives all the messages from the head of the queue specified by the service MYSERVICE, using the default policy (DB2.DEFAULT.POLICY). Only the MSG and CORRELID columns are returned.

```
SELECT T.MSG, T.CORRELID
FROM table (MQRECEIVEALL('MYSERVICE')) T
```

Example 3: This example receives all of the message from the head of the queue specified by the service "MYSERVICE", using the policy "MYPOLICY". Only messages with a CORRELID of '1234' are returned. Only the MSG and CORRELID columns are returned.

```
SELECT T.MSG, T.CORRELID
FROM table (MQRECEIVEALL('MYSERVICE','MYPOLICY','1234')) T
```

Example 4: This example receives the first 10 messages from the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). All columns are returned.

```
SELECT *
FROM table (MQRECEIVEALL(10)) T
```

Procedures

GET_ROUTINE_SAR

► GET_ROUTINE_SAR (—*sarblob*—, —*type*—, —*routine_name_string*—) ◀

The schema is SYSFUN.

This procedure was first available in FixPak 3 of Version 7.1.

The GET_ROUTINE_SAR procedure retrieves the necessary information to install the same routine in another database server running the same level on the same operating system. The information is retrieved into a single BLOB string representing an SQL archive file. The invoker of the GET_ROUTINE_SAR procedure must have DBADM authority.

sarblob

An output argument of type BLOB(3M) that contains the routine SAR file contents.

type

An input argument of type CHAR(2) that specifies whether the type of routine, using one of the following values:

- **P** for a procedure.
- **SP** for the specific name of a procedure.

routine_name_string

An input argument of type VARCHAR(257) that specifies a qualified name of the routine. If no schema name is specified, the default is the CURRENT SCHEMA when the routine is processed.

Note: The *routine_name_string* cannot include the double quote character (").

The qualified name of the routine is used to determine which routine to retrieve. The routine that is found must be an SQL routine or an error is raised (SQLSTATE 428F7). When not using a specific name, this may result in more than one routine and an error is raised (SQLSTATE 42725). If this occurs, the specific name of the routine must be used to get the routine.

The SAR file must include a bind file which may not be available at the server. If the bind file cannot be found and stored in the SAR file, an error is raised (SQLSTATE 55045).

PUT_ROUTINE_SAR

► PUT_ROUTINE_SAR (—*sarblob*—, —*new_owner*—, —*use_register_flag*—) ◀

The schema is SYSFUN.

This procedure was first available in FixPak 3 of Version 7.1.

The PUT_ROUTINE_SAR procedure passes the necessary file to create an SQL routine at the server and then defines the routine. The invoker of the PUT_ROUTINE_SAR procedure must have DBADM authority.

sarblob

An input argument of type BLOB(3M) that contains the routine SAR file contents.

new_owner

An input argument of type VARCHAR(128) that contains an authorization-name used for authorization checking of the routine. The *new-owner* must have the necessary privileges for the routine to be defined. If *new-owner* is not specified, the authorization-name of the original routine definer is used.

use_register_flag

An input argument of type INTEGER that indicates whether or not the CURRENT SCHEMA and CURRENT PATH special registers are used to define the routine. If the special registers are not used, the settings for the default schema and SQL path are the settings used when the routine was originally defined. Possible values for *use-register-flag*:

- 0** Do not use the special registers of the current environment
- 1** Use the CURRENT SCHEMA and CURRENT PATH special registers.

If the value is 1, CURRENT SCHEMA is used for unqualified object names in the routine definition (including the name of the routine) and CURRENT PATH is used to resolve unqualified routines and data types in the routine definition. If the *use-registers-flag* is not specified, the behavior is the same as if a value of 0 was specified.

The identification information contained in *sarblob* is checked to confirm that the inputs are appropriate for the environment, otherwise an error is raised (SQLSTATE 55046). The PUT_ROUTINE_SAR procedure then uses the contents of the *sarblob* to define the routine at the server.

The contents of the *sarblob* argument are extracted into the separate files that make up the SQL archive file. The shared library and bind files are written to files in a temporary directory. The environment is set so that the routine definition statement processing is aware that compiling and linking are not required, and that the location of the shared library and bind files is available. The contents of the DDL file are then used to dynamically execute the routine definition statement.

Note: No more than one procedure can be concurrently installed under a given schema.

Processing of this statement may result in the same errors as executing the routine definition statement using other interfaces. During routine definition processing, the presence of the shared library and bind files is noted and the precompile, compile and link steps are skipped. The bind file is used during bind processing and the contents of both files are copied to the usual directory for an SQL routine.

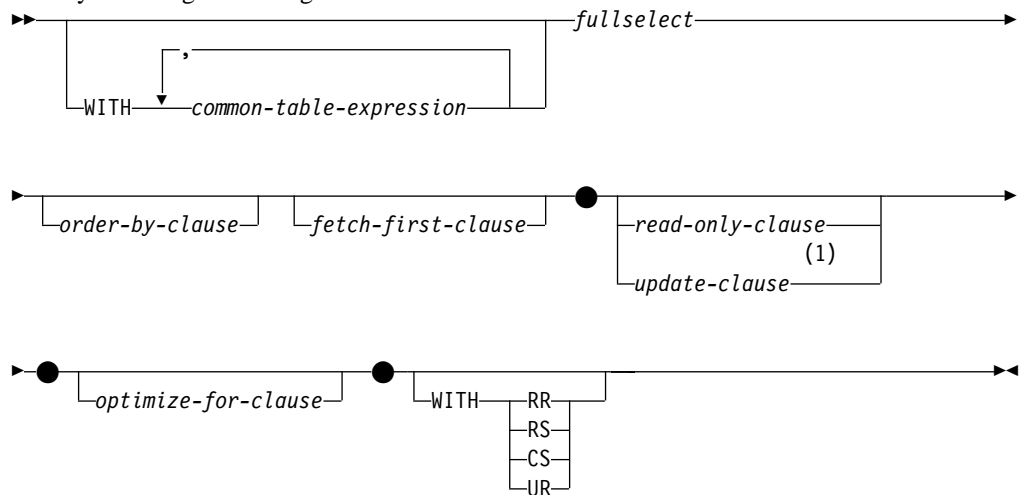
Note: If a GET ROUTINE or a PUT ROUTINE operation (or their corresponding procedure) fails to execute successfully, it will always return an error (SQLSTATE 38000), along with diagnostic text providing information about the cause of the failure. For example, if the procedure name provided to GET ROUTINE does not identify an SQL procedure, diagnostic "100, 02000" text will be returned, where "100" and "02000" are the SQLCODE and SQLSTATE, respectively, that identify the cause of the problem. The

SQLCODE and SQLSTATE in this example indicate that the row specified for the given procedure name was not found in the catalog tables.

Chapter 5. Queries

select-statement/syntax diagram

The syntax diagram changes to:



Note:

- 1 The update-clause and the order-by-clause cannot both be specified in the same select-statement.

Add the following paragraph to the description below the syntax diagram:

The optional WITH clause specifies the isolation level at which the select statement is executed.

- RR - Repeatable Read
- RS - Read Stability
- CS - Cursor Stability
- UR - Uncommitted Read

The default isolation level of the statement is the isolation level of the package in which the statement is bound.

select-statement/fetch-first-clause

The last paragraph in the description of the fetch-first-clause:

Specification of the fetch-first-clause in a select-statement makes the cursor not deletable (read-only). This clause cannot be specified with the FOR UPDATE clause.

is incorrect and should be removed.

Chapter 6. SQL Statements

Update of the Partitioning Key Now Supported

Update the partitioning key is now supported. The following text from various statements in Chapter 6 should be **deleted** only if the DB2_UPDATE_PART_KEY=ON:

註: If DB2_UPDATE_PART_KEY=OFF, then the restrictions still apply.

Statement: ALTER TABLE

Rules

- A partitioning key column of a table cannot be updated (SQLSTATE 42997).
- A nullable column of a partitioning key cannot be included as a foreign key column when the relationship is defined with ON DELETE SET NULL (SQLSTATE 42997).

Statement: CREATE TABLE

Rules

- A partitioning key column of a table cannot be updated (SQLSTATE 42997).
- A nullable column of a partitioning key cannot be included as a foreign key column when the relationship is defined with ON DELETE SET NULL (SQLSTATE 42997).

Statement: DECLARE GLOBAL TEMPORARY TABLE PARTITIONING KEY (column-name,...)

註: The partitioning key columns cannot be updated (SQLSTATE 42997).

Statement: UPDATE

Footnotes

- 108 A column of a partitioning key is not updatable (SQLSTATE 42997). The row of data must be deleted and inserted to change columns in a partitioning key.

Larger Index Keys for Unicode Databases

ALTER TABLE

The length of variable length columns that are part of any index, including primary and unique keys, defined when the registry variable DB2_INDEX_2BYTEVARLEN was on, can be altered to a length greater than 255 bytes. The fact that a variable length column is involved in a foreign key will no longer prevent the length of that column from being altered to larger than 255 bytes, regardless of the registry variable setting. However, data with length greater than 255 cannot be inserted into the table unless the column in the corresponding primary key has length greater than 255 bytes, which is only possible if the primary key was created with the registry variable ON.

CREATE INDEX

Indexes can be defined on variable length columns whose length is greater than 255 bytes if the registry variable DB2_INDEX_2BYTEVARLEN is ON.

CREATE TABLE

Primary and unique keys with variable keyparts can have a size greater than 255 if the registry variable DB2_INDEX_2BYTEVARLEN is ON. Foreign keys can be defined on variable length columns whose length is greater than 255 bytes.

ALTER SEQUENCE

ALTER SEQUENCE

The ALTER SEQUENCE statement modifies the attributes of a sequence by:

- Restarting the sequence
- Changing the increment between future sequence values
- Setting new minimum or maximum values
- Changing the number of cached sequence numbers
- Changing whether the sequence can cycle or not
- Changing whether sequence numbers must be generated in order of request

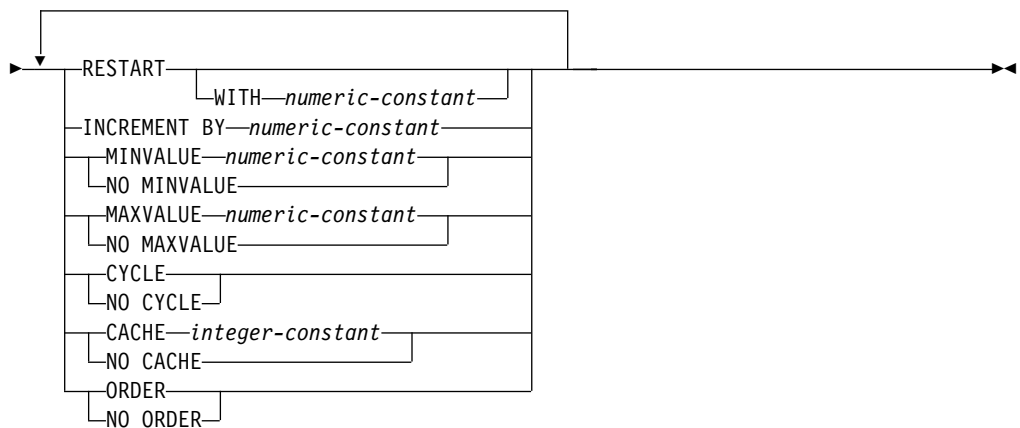
Invocation: This statement can be embedded in an application program or issued through the use of dynamic SQL statements. It is an executable statement that can be dynamically prepared. However, if the bind option DYNAMICRULES BIND applies, the statement cannot be dynamically prepared (SQLSTATE 42509).

Authorization: The privileges held by the authorization ID of the statement must include at least one of the following:

- Definer of the sequence
- The ALTERIN privilege for the schema implicitly or explicitly specified
- SYSADM or DBADM authority

Syntax:

▶▶ ALTER SEQUENCE *sequence-name* →



Description:

sequence-name

Identifies the particular sequence. The combination of name, and the implicit or explicit schema name must identify an existing sequence at the current server. If no sequence by this name exists in the explicitly or implicitly specified schema, an error (SQLSTATE 42704) is issued.

RESTART

Restarts the sequence. If *numeric-constant* is not specified, the sequence is restarted

at the value specified implicitly or explicitly as the starting value on the CREATE SEQUENCE statement that originally created the sequence.

WITH *numeric-constant*

Restarts the sequence with the specified value. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 42820) as long as there are no non-zero digits to the right of the decimal point (SQLSTATE 42894).

INCREMENT BY

Specifies the interval between consecutive values of the sequence. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 42820), and does not exceed the value of a large integer constant (SQLSTATE 42815), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA).

If this value is negative, then the sequence of values descends. If this value is positive, then the sequence of values ascends. If this value is 0 or greater than the range defined by MINVALUE and MAXVALUE, only one value will be generated, but the sequence is treated as an ascending sequence otherwise.

MINVALUE or NO MINVALUE

Specifies the minimum value at which a descending sequence either cycles or stops generating values, or an ascending sequence cycles to after reaching the maximum value.

MINVALUE *numeric-constant*

Specifies the numeric constant that is the minimum value. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 42820), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA), but the value must be less than or equal to the maximum value (SQLSTATE 42815).

NO MINVALUE

For an ascending sequence, the value is the START WITH value, or 1 if START WITH is not specified. For a descending sequence, the value is the minimum value of the data type associated with the sequence. This is the default.

MAXVALUE or NO MAXVALUE

Specifies the maximum value at which an ascending sequence either cycles or stops generating values, or a descending sequence cycles to after reaching the minimum value.

MAXVALUE *numeric-constant*

Specifies the numeric constant that is the maximum value. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 428FA), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA), but the value must be greater than or equal to the minimum value (SQLSTATE 42815).

NO MAXVALUE

For an ascending sequence, the value is the maximum value of the data type associated with the sequence. For a descending sequence, the value is the START WITH value, or -1 if START WITH is not specified. This is the default.

CYCLE or NOCYCLE

Specifies whether the sequence should continue to generate values after reaching either its maximum or minimum value. The boundary of the sequence can be reached either with the next value landing exactly on the boundary condition, or by overshooting it in which case the next value would be determined from wrapping around to the START WITH value if cycles were permitted.

CYCLE

Specifies that values continue to be generated for this sequence after the maximum or minimum value has been reached. If this option is used, after an ascending sequence reaches its maximum value, it generates its minimum value; or after a descending sequence reaches its minimum value, it generates its maximum value. The maximum and minimum values for the sequence determine the range that is used for cycling.

When CYCLE is in effect, then duplicate values can be generated for the sequence.

NO CYCLE

Specifies that values will not be generated for the sequence once the maximum or minimum value for the sequence has been reached. This is the default.

CACHE or NO CACHE

Specifies whether to keep some preallocated values in memory for faster access. This is a performance and tuning option.

CACHE *integer-constant*

Specifies the maximum number of sequence values that are preallocated and kept in memory. Preallocating and storing values in the cache reduces synchronous I/O to the log when values are generated for the sequence.

In the event of a system failure, all cached sequence values that have not been used in committed statements are lost (that is, they will never be used). The value specified for the CACHE option is the maximum number of sequence values that could be lost in case of system failure.

The minimum value is 2 (SQLSTATE 42815). The default value is CACHE 20.

NO CACHE

Specifies that values of the sequence are not to be preallocated. It ensures that there is not a loss of values in the case of a system failure, shutdown or database deactivation. When this option is specified, the values of the sequence are not stored in the cache. In this case, every request for a new value for the sequence results in synchronous I/O to the log.

NO ORDER or ORDER

Specifies whether the sequence numbers must be generated in order of request.

ORDER

Specifies that the sequence numbers are generated in order of request.

NO ORDER

Specifies that the sequence numbers do not need to be generated in order of request. This is the default.

After restarting a sequence or changing to `CYCLE`, it is possible for sequence numbers to be duplicate values of ones generated by the sequence previously.

Version:

- Only future sequence numbers are affected by the `ALTER SEQUENCE` statement.
- The data type of a sequence cannot be changed. Instead, drop and recreate the sequence specifying the desired data type for the new sequence.
- All cached values are lost when a sequence is altered.

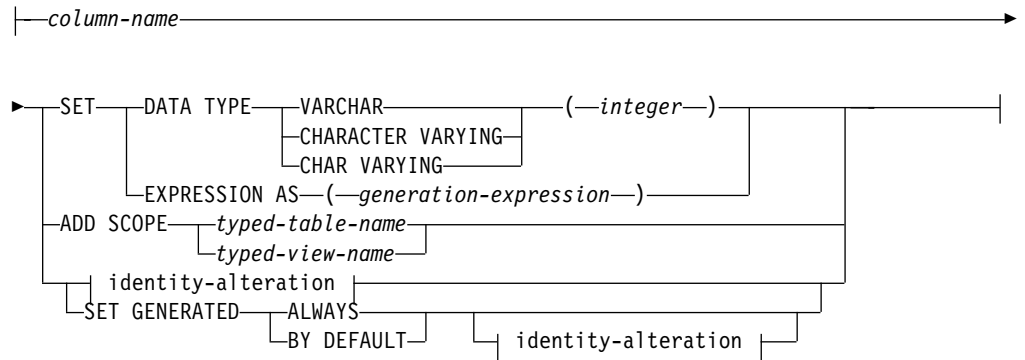
Examples *Example1:* A possible reason for specifying `RESTART` without a numeric value would be to reset the sequence to the `START WITH` value. In this example, the goal is to generate the numbers from 1 up to the number of rows in the table and then inserting the numbers into a column added to the table using temporary tables. Another use would be to get results back where all the resulting rows are numbered:

```
ALTER SEQUENCE org_seq
RESTART
SELECT NEXTVAL for org_seq, org.*
FROM org
```

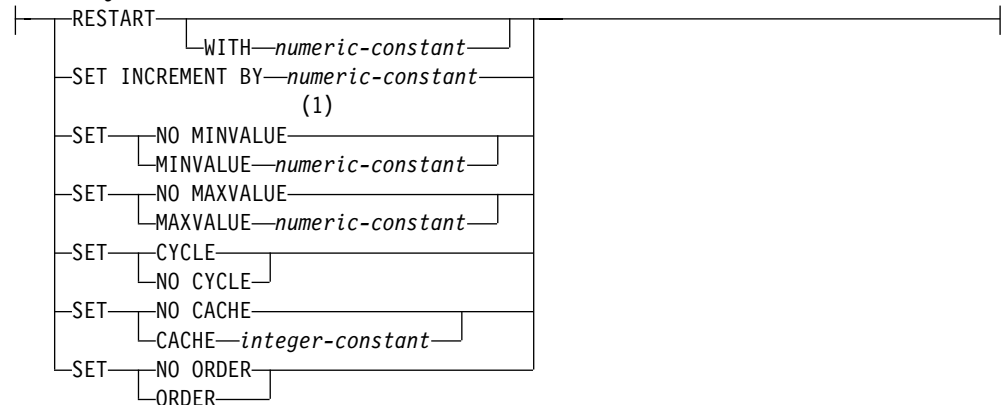
ALTER TABLE

Changes to syntax fragments:

column-alteration:



identity-alteration:



Note:

- 1 These parameters can be specified without spaces: NOMINVALUE, NOMAXVALUE, NOCYCLE, NOCACHE, and NOORDER. These single word versions are all acceptable alternatives to the two word versions.

Add the following parameters:

SET GENERATED

Specifies whether values are to be generated for the column always or only when a default value is needed.

ALWAYS

A value will always be generated for the column when a row is inserted or updated in the table. The column must already be defined as a generated column (SQLSTATE 42837).

BY DEFAULT

The value will be generated for the column when a row is inserted into the table, unless a value is specified. The column must already be defined as a generated column (SQLSTATE 42837).

RESTART or RESTART WITH *numeric-constant*

Resets the state of the sequence associated with the identity column. If WITH *numeric-constant* is not specified, then the sequence for the identity column is restarted at the value that was specified, either implicitly or explicitly, as the starting value when the identity column was originally created. The *numeric-constant* is an exact numeric constant that can be any positive or negative value that could be assigned to this column (SQLSTATE 42820) as long as there are no non-zero digits to the right of the decimal point (SQLSTATE 42894). The column must already be defined with the IDENTITY attribute (SQLSTATE 42837). The *numeric-constant* will be used as the next value for the column.

SET INCREMENT BY *numeric-constant*

Specifies the interval between consecutive values of the identity column. The column must already be defined with the IDENTITY attribute (SQLSTATE 42837). This value is any positive or negative value that could be assigned to this column (SQLSTATE 42820), and does not exceed the value of a large integer constant (SQLSTATE 42815), as long as there are no non-zero digits to the right of the decimal point (SQLSTATE 42894).

If this value is negative, then the sequence of values for this identity column descends. If this value is positive, then the sequence of values for this identity column ascends. If this value is 0, or is greater than the range defined by MINVALUE and MAXVALUE, then DB2 will only generate one value, but the sequence is treated as an ascending sequence otherwise.

SET MINVALUE *numeric-constant* or NO MINVALUE

Specifies the minimum value at which a descending identity column either cycles or stops generating values, or the value to which an ascending identity column cycles to after reaching the maximum value. The column must already be defined with the IDENTITY attribute (SQLSTATE 42837).

MINVALUE *numeric-constant*

Specifies the minimum numeric constant value . This value can be any positive or negative value that could be assigned to this column (SQLSTATE 42820), without non-zero digits existing to the right of the decimal point (SQLSTATE 42894), but the value must be less than the maximum value (SQLSTATE 42815).

NO MINVALUE

For an ascending sequence, the value is the START WITH value, or 1 if START WITH is not specified. For a descending sequence, the value is the minimum value of the data type of the column.

SET MAXVALUE *numeric-constant* **or NO MAXVALUE**

Specifies the maximum value at which an ascending identity column either cycles or stops generating values, or the value to which a descending identity column cycles to after reaching the minimum value. The column must already be defined with the IDENTITY attribute (SQLSTATE 42837).

MAXVALUE *numeric-constant*

Specifies the numeric constant that is the maximum value. This value can be any positive or negative value that could be assigned to this column (SQLSTATE 42820), without non-zero digits existing to the right of the decimal point (SQLSTATE 42894), but the value must be greater than the minimum value (SQLSTATE 42815).

NO MAXVALUE

For an ascending sequence, the value is the maximum value of the data type of the column. For a descending sequence, the value is the START WITH value, or -1 if START WITH is not specified.

SET CYCLE or NO CYCLE

Specifies whether this identity column should continue to generate values after generating either the maximum or minimum value. The column must already be defined with the IDENTITY attribute (SQLSTATE 42837).

CYCLE

Specifies that values continue to be generated for this column after the maximum or minimum value has been reached. If this option is used, then after an ascending identity column reaches the maximum value, it generates its minimum value; or after a descending sequence reaches the minimum value, it generates its maximum value. The maximum and minimum values for the identity column determine the range that is used for cycling.

When CYCLE is in effect, then duplicate values can be generated for an identity column. Although not required, if unique values are desired, a single-column unique index defined using the identity column will ensure uniqueness. If a unique index exists on such an identity column and a non-unique value is generated, then an error occurs (SQLSTATE 23505).

NO CYCLE

Specifies that values will not be generated for the identity column once the maximum or minimum value has been reached.

SET CACHE *integer-constant* or NO CACHE

Specifies whether to keep some preallocated values in memory for faster access. This is a performance and tuning option. The column must already be defined with the IDENTITY attribute (SQLSTATE 42837).

CACHE *integer-constant*

Specifies how many values of the identity sequence are preallocated and kept in memory. When values are generated for the identity column, preallocating and storing values in the cache reduces synchronous I/O to the log.

If a new value is needed for the identity column and there are no unused values available in the cache, then the allocation of the value requires waiting for I/O to the log. However, when a new value is needed for the identity column and there is an unused value in the cache, the allocation of that identity value can happen more quickly by avoiding the I/O to the log.

When a database manager is stopped (database deactivation, system failure, or shutdown, for example), all cached sequence values that have not been used in committed statements are lost (that is, they will never be used). The value specified for the CACHE option is the maximum number of values for the identity column that could be lost in case of system failure.

The minimum value is 2 (SQLSTATE 42615).

NO CACHE

Specifies that values for the identity column are not to be preallocated.

When this option is specified, the values of the identity column are not stored in the cache. In this case, every request for a new identity value results in synchronous I/O to the log.

SET ORDER or NO ORDER

Specifies whether the identity column values must be generated in order of request. The column must already be defined with the IDENTITY attribute (SQLSTATE 42837).

ORDER

Specifies that the identity column values are generated in order of request.

NO ORDER

Specifies that the identity column values do not need to be generated in order of request.

Compound SQL (Embedded)

A prepared COMMIT statement is not allowed in an ATOMIC compound SQL statement.

Compound Statement (Dynamic)

Compound Statement (Dynamic)

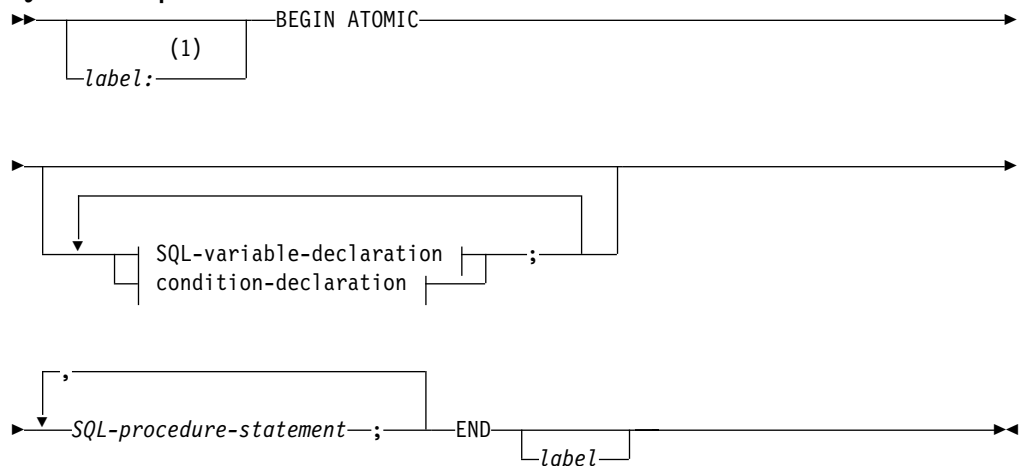
A compound statement groups other statements together into an executable block. You can declare SQL variables within a dynamically prepared atomic compound statement.

Invocation: This statement can be embedded in a trigger, SQL Function, or SQL Method, or issued through the use of dynamic SQL statements. It is an executable statement that can be dynamically prepared.

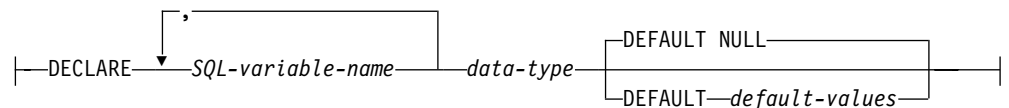
Authorization: No privileges are required to invoke a dynamic compound statement. However, the authorization ID of the compound statement must hold the necessary privileges to invoke the SQL statements embedded in the compound statement.

Syntax:

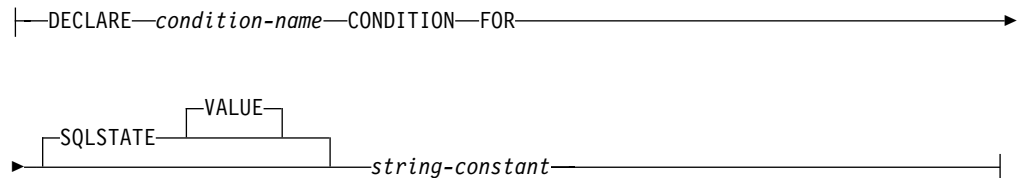
dynamic-compound-statement



SQL-variable-declaration:



condition-declaration:



Note:

- 1 A label can only be specified when the statement is in a function, method, or trigger definition.

Description:

label

Defines the label for the code block. If the beginning label is specified, it can be used to qualify SQL variables declared in the dynamic compound statement and can also be specified on a LEAVE statement. If the ending label is specified, it must be the same as the beginning label.

ATOMIC

ATOMIC indicates that, if an error occurs in the compound statement, all SQL statements in the compound statement will be rolled back and any remaining SQL statements in the compound statement are not processed.

SQL-procedure-statement

The following list of *SQL-control-statements* can be used within the dynamic compound statement:

- FOR Statement
- GET DIAGNOSTICS Statement
- IF Statement
- ITERATE Statement
- LEAVE Statement
- SIGNAL Statement
- WHILE Statement

The SQL statements that can be issued are:

- fullselect⁶
- Searched UPDATE
- Searched DELETE
- INSERT
- SET variable statement

SQL-variable-declaration

Declares a variable that is local to the dynamic compound statement.

SQL-variable-name

Defines the name of a local variable. DB2 converts all SQL variable names to uppercase. The name cannot:

- Be the same as another SQL variable within the same compound statement.
- Be the same as a parameter name.
- Be the same as column names.

If an SQL statement contains an identifier with the same name as an SQL variable and a column reference, DB2 interprets the identifier as a column.

data-type

Specifies the data type of the variable.

6. A common-table-expression may precede the fullselect

DEFAULT *default-values* or **NULL**

Defines the default for the SQL variable. The variable is initialized when the dynamic compound statement is called. If a default value is not specified, the variable is initialized to NULL.

condition-declaration

Declares a condition name and corresponding SQLSTATE value.

condition-name

Specifies the name of the condition. The condition name must be unique within the procedure body and can be referenced only within the compound statement in which it is declared.

FOR SQLSTATE *string-constant*

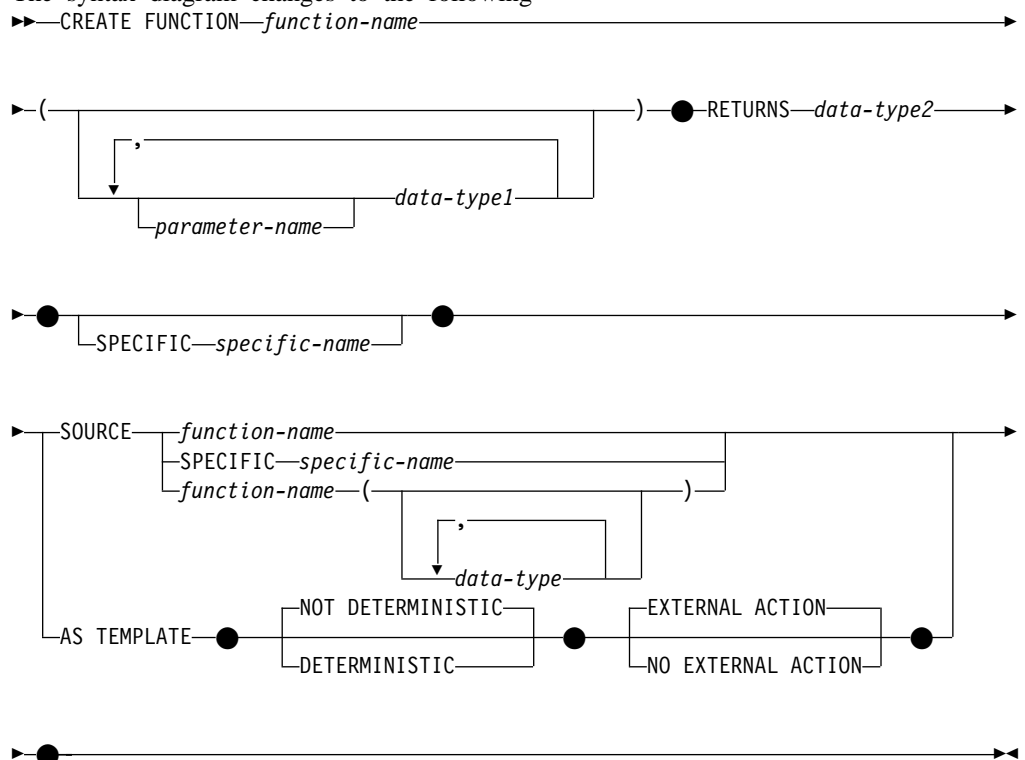
Specifies the SQLSTATE associated with the condition. The *string-constant* must be specified as five characters enclosed in single quotes, and cannot be '00000'.

Notes:

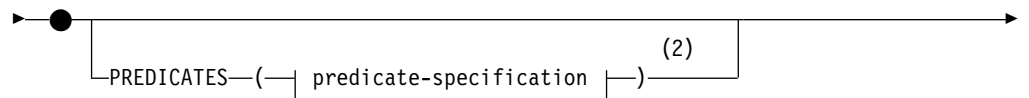
- Dynamic compound statements are compiled by DB2 as one single statement. This statement is effective for short scripts involving little control flow logic but significant data flow. For larger constructs with nested complex control flow, a better choice is to use SQL procedures for details on using SQL procedures.

CREATE FUNCTION (Source or Template)

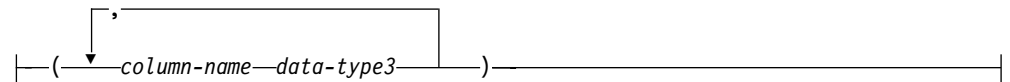
The syntax diagram changes to the following



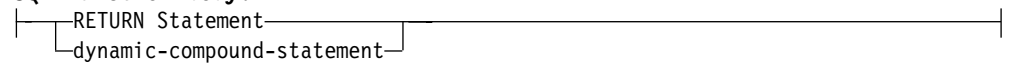
Add the following to the "Description" section:



column-list:



SQL-function-body:



Note:

- 1 NULL CALL may be specified in place of CALLED ON NULL INPUT
- 2 Valid only if RETURNS specifies a scalar result (data-type2)

Change the following parameters:

LANGUAGE SQL

Specifies that the function is written using SQL.

This parameter section replaces the "RETURN expression, NULL, WITH common-table-expression, fullselect" parameter section.

SQL-function-body

Specifies the body of the function. Parameter names can be referenced in the SQL-function-body. Parameter names may be qualified with the function name to avoid ambiguous references.

If the SQL-function-body is a dynamic compound statement, it must contain at least one RETURN statement and a RETURN statement must be executed when the function is called (SQLSTATE 42632). If the function is a table or row function, then it can contain only one RETURN statement which must be the last statement in the dynamic compound (SQLSTATE 429BD).

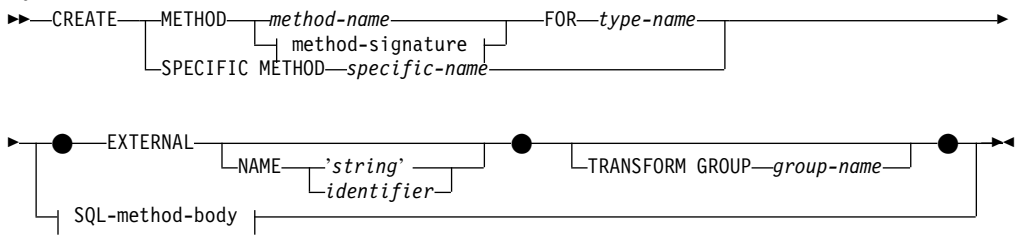
For additional details, see page 391 『Compound Statement (Dynamic)』 and page 419 『RETURN』.

CREATE METHOD

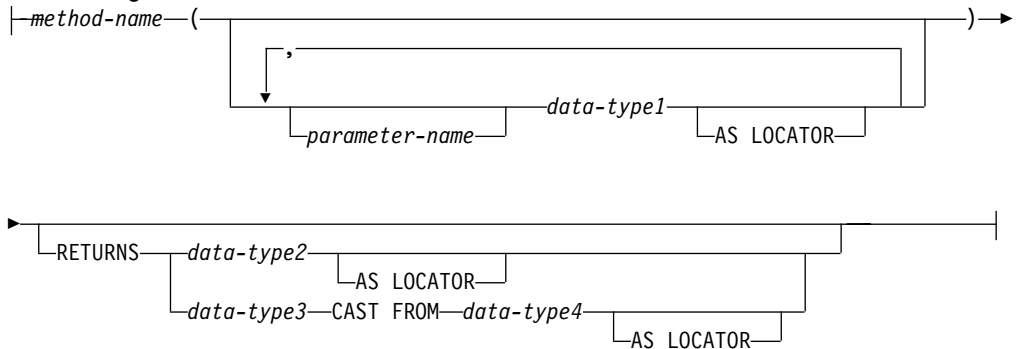
The syntax diagram changes to:

CREATE METHOD

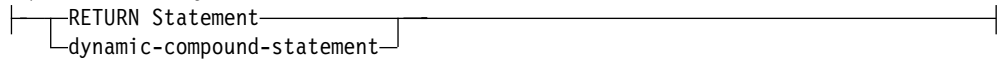
Syntax:



method-signature:



SQL-method-body:



The following parameters replace the "RETURN scalar-expression or NULL" section:

SQL-method-body

The `SQL-method-body` defines how the method is implemented if the method specification in `CREATE TYPE` is `LANGUAGE SQL`.

The `SQL-method-body` must comply with the following parts of the method specification:

- `DETERMINISTIC` or `NOT DETERMINISTIC` (SQLSTATE 428C2)
- `EXTERNAL ACTION` or `NO EXTERNAL ACTION` (SQLSTATE 428C2)
- `CONTAINS SQL` or `READS SQL DATA` (SQLSTATE 42985)

Parameter names can be referenced in the `SQL-method-body`. The subject of the method is passed to the method implementation as an implicit first parameter named `SELF`.

For additional details, see page 391 『Compound Statement (Dynamic)』 and page 419 『RETURN』.

CREATE SEQUENCE

CREATE SEQUENCE

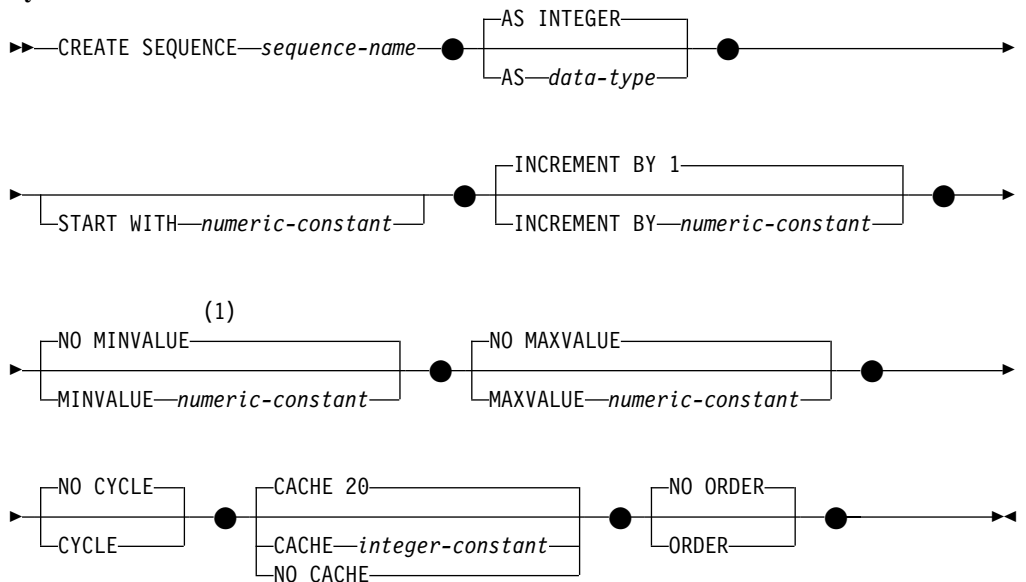
The CREATE SEQUENCE statement creates a sequence at the application server.

Invocation: This statement can be embedded in an application program or issued through the use of dynamic SQL statements. It is an executable statement that can be dynamically prepared. However, if the bind option DYNAMICRULES BIND applies, the statement cannot be dynamically prepared (SQLSTATE 42509).

Authorization: The privileges held by the authorization ID of the statement must include at least one of the following:

- CREATEIN privilege for the implicitly or explicitly specified schema
- SYSADM or DBADM authority

Syntax:



Note:

- 1 These parameters can be specified without spaces: NOMINVALUE, NOMAXVALUE, NOCYCLE, NOCACHE, and NOORDER. These single word versions are all acceptable alternatives to the two word versions.

Description:

sequence-name

Names the sequence. The combination of name, and the implicit or explicit schema name must not identify an existing sequence at the current server (SQLSTATE 42710).

The unqualified form of *sequence-name* is an SQL identifier. The qualified form is a qualifier followed by a period and an SQL identifier. The qualifier is a schema name.

If the sequence name is explicitly qualified with a schema name, the schema name cannot begin with 'SYS' or an error (SQLSTATE 42939) is raised.

AS *data-type*

Specifies the data type to be used for the sequence value. The data type can be any exact numeric type (SMALLINT, INTEGER, BIGINT or DECIMAL) with a scale of zero or a user-defined distinct type for which the source type is an exact numeric type with a scale of zero (SQLSTATE 42815). The default is INTEGER.

START WITH *numeric-constant*

Specifies the first value for the sequence. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 42820), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA). The default is MINVALUE for ascending sequences and MAXVALUE for descending sequences.

This value is not necessarily the value that a sequence would cycle to after reaching the maximum or minimum value of the sequence. The START WITH clause can be used to start a sequence outside the range that is used for cycles. The range used for cycles is defined by MINVALUE and MAXVALUE.

INCREMENT BY *numeric-constant*

Specifies the interval between consecutive values of the sequence. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 42820), and does not exceed the value of a large integer constant (SQLSTATE 42815), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA).

If this value is negative, then the sequence of values descends. If this value is positive, then the sequence of values ascends. If this value is 0 or greater than the range defined by MINVALUE and MAXVALUE, only one value will be generated, but the sequence is treated as an ascending sequence otherwise. The default is 1.

MINVALUE or NO MINVALUE

Specifies the minimum value at which a descending sequence either cycles or stops generating values, or an ascending sequence cycles to after reaching the maximum value.

MINVALUE *numeric-constant*

Specifies the numeric constant that is the minimum value. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 42820), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA), but the value must be less than or equal to the maximum value (SQLSTATE 42815).

NO MINVALUE

For an ascending sequence, the value is the START WITH value, or 1 if START WITH is not specified. For a descending sequence, the value is the minimum value of the data type associated with the sequence. This is the default.

MAXVALUE or NO MAXVALUE

Specifies the maximum value at which an ascending sequence either cycles or stops generating values, or a descending sequence cycles to after reaching the minimum value.

MAXVALUE *numeric-constant*

Specifies the numeric constant that is the maximum value. This value can be

any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 428FA), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA), but the value must be greater than or equal to the minimum value (SQLSTATE 42815).

NO MAXVALUE

For an ascending sequence, the value is the maximum value of the data type associated with the sequence. For a descending sequence, the value is the START WITH value, or -1 if START WITH is not specified. This is the default.

CYCLE or NO CYCLE

Specifies whether the sequence should continue to generate values after reaching either its maximum or minimum value. The boundary of the sequence can be reached either with the next value landing exactly on the boundary condition, or by overshooting it.

CYCLE

Specifies that values continue to be generated for this sequence after the maximum or minimum value has been reached. If this option is used, after an ascending sequence reaches its maximum value it generates its minimum value; after a descending sequence reaches its minimum value it generates its maximum value. The maximum and minimum values for the sequence determine the range that is used for cycling.

When CYCLE is in effect, then duplicate values can be generated for the sequence.

NO CYCLE

Specifies that values will not be generated for the sequence once the maximum or minimum value for the sequence has been reached. This is the default.

CACHE or NO CACHE

Specifies whether to keep some preallocated values in memory for faster access. This is a performance and tuning option.

CACHE *integer-constant*

Specifies the maximum number of sequence values that are preallocated and kept in memory. Preallocating and storing values in the cache reduces synchronous I/O to the log when values are generated for the sequence.

In the event of a system failure, all cached sequence values that have not been used in committed statements are lost (that is, they will never be used). The value specified for the CACHE option is the maximum number of sequence values that could be lost in case of system failure.

The minimum value is 2 (SQLSTATE 42815). The default value is CACHE 20.

NO CACHE

Specifies that values of the sequence are not to be preallocated. It ensures that there is not a loss of values in the case of a system failure, shutdown or database deactivation. When this option is specified, the values of the sequence are not stored in the cache. In this case, every request for a new value for the sequence results in synchronous I/O to the log.

NO ORDER or ORDER

Specifies whether the sequence numbers must be generated in order of request.

ORDER

Specifies that the sequence numbers are generated in order of request.

NO ORDER

Specifies that the sequence numbers do not need to be generated in order of request. This is the default.

Notes:

- It is possible to define a constant sequence, that is, one that would always return a constant value. This could be done by specifying the same value for MINVALUE or MAXVALUE, or by specifying an INCREMENT value of zero. In either case, in order to allow for NEXTVAL to generate the same value more than once, CYCLE must be specified. A constant sequence can be used as a numeric global variable. ALTER SEQUENCE can be used to adjust the values that will be generated for a constant sequence.
- A sequence can be cycled manually, by using the ALTER SEQUENCE statement. If NO CYCLE is implicitly or explicitly specified, the sequence can be restarted or extended using the ALTER SEQUENCE statement to cause values to continue to be generated once the maximum or minimum value for the sequence has been reached.
- Caching sequence numbers implies that a range of sequence numbers can be kept in memory for fast access. When an application accesses a sequence that can allocate the next sequence number from the cache, the sequence number allocation can happen quickly. However, if an application accesses a sequence that cannot allocate the next sequence number from the cache, the sequence number allocation may require having to wait for I/O operations to persistent storage. The choice of the value for CACHE should be done keeping in mind the performance and application requirements tradeoffs.
- The owner has the ALTER and USAGE privileges on the new sequence. Only the USAGE privilege can be granted by the owner and only to PUBLIC.
- The following syntax is also supported: NOMINVALUE, NOMAXVALUE, NOCYCLE, NOCACHE, and NOORDER.

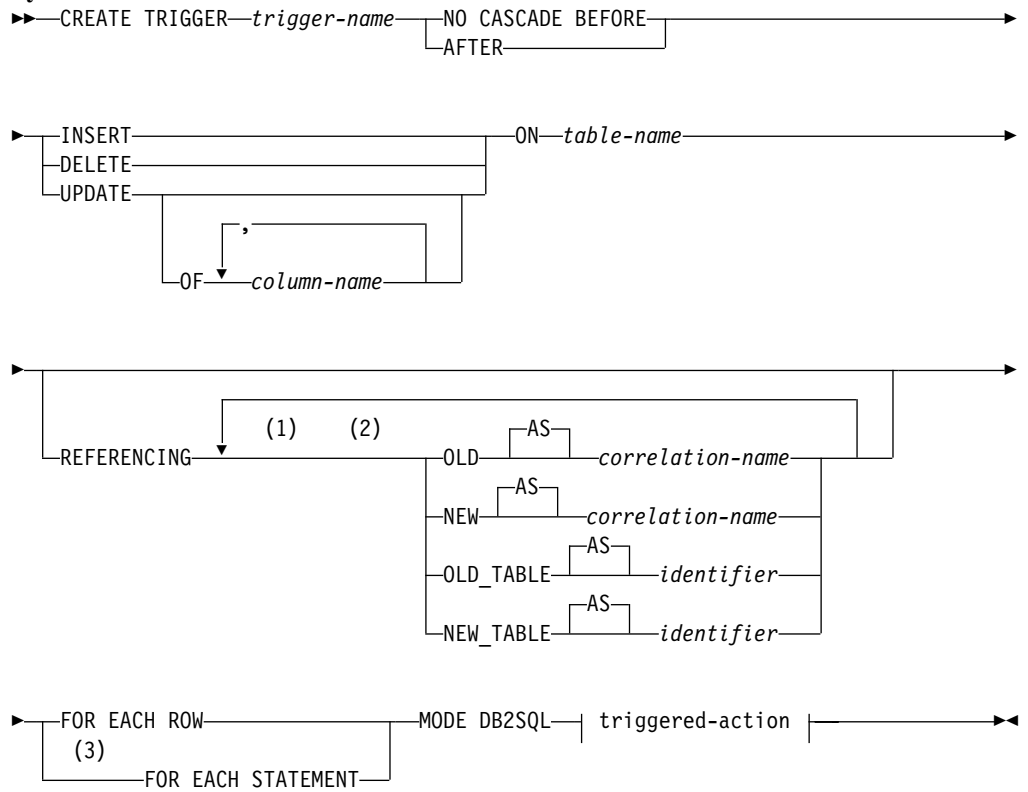
Examples *Example 1:* Create a sequence called org_seq:

```
CREATE SEQUENCE org_seq  
  START WITH 1  
  INCREMENT BY 1  
  NO MAXVALUE  
  NO CYCLE  
  CACHE 24
```

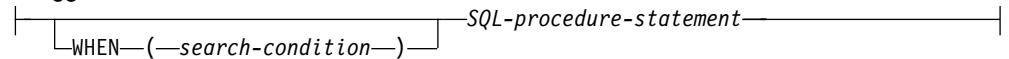
CREATE TRIGGER

CREATE TRIGGER

Syntax:



triggered-action:



Note:

- 1 OLD and NEW may only be specified once each.
- 2 OLD_TABLE and NEW_TABLE may only be specified once each and only for AFTER triggers.
- 3 FOR EACH STATEMENT may not be specified for BEFORE triggers.

Replace the description of "triggered-action" with the following:

triggered-action

Specifies the action to be performed when a trigger is activated. A triggered-action is composed of an *SQL-procedure-statement* and an optional condition for the execution of the *SQL-procedure-statement*.

WHEN (*search-condition*)

Specifies a condition that is true, false, or unknown. The *search-condition* provides a capability to determine whether or not a certain triggered action should be executed.

The associated action is performed only if the specified search condition evaluates as true. If the WHEN clause is omitted, the associated *SQL-procedure statement* is always performed.

SQL-procedure-statement

The *SQL-procedure-statement* can contain a dynamic compound statement or any of the SQL control statements listed in 第391頁的『Compound Statement (Dynamic)』.

If the trigger is a BEFORE trigger, then an *SQL-procedure-statement* can also include a fullselect or a SET variable statement (SQLSTATE 42987).

If the trigger is an AFTER trigger, then an *SQL-procedure-statement* can also include one of the following (SQLSTATE 42987):

- an INSERT SQL statement
- a searched UPDATE SQL statement
- a searched DELETE SQL statement
- a SET variable statement
- a fullselect ⁷

The *SQL-procedure-statement* cannot reference an undefined transition variable (SQLSTATE 42703) or a declared temporary table (SQLSTATE 42995).

The *SQL-procedure-statement* in a BEFORE trigger cannot reference a summary table defined with REFRESH IMMEDIATE (SQLSTATE 42997).

The *SQL-procedure-statement* in a BEFORE trigger cannot reference a generated column, other than the identity column, in the new transition variable (SQLSTATE 42989).

The Notes section changes to the following:

- The result of a fullselect specified in the *SQL-procedure-statement* is not available inside or outside of the trigger.
- **Inoperative triggers:** An *inoperative trigger* is a trigger that is no longer available and is therefore never activated. A trigger becomes inoperative if:
 - A privilege that the creator of the trigger is required to have for the trigger to execute is revoked.
 - An object such as a table, view or alias, upon which the triggered action is dependent, is dropped.
 - A view, upon which the triggered action is dependent, becomes inoperative.
 - An alias that is the subject table of the trigger is dropped.

In practical terms, an inoperative trigger is one in which a trigger definition has been dropped as a result of cascading rules for DROP or REVOKE statements. For example, when an view is dropped, any trigger with an *SQL-procedure-statement* defined using that view is made inoperative.

When a trigger is made inoperative, all packages with statements performing operations that were activating the trigger will be marked invalid. When the package is rebound

7. A common-table-expression may precede a fullselect.

(explicitly or implicitly) the **inoperative trigger is completely ignored**. Similarly, applications with dynamic SQL statements performing operations that were activating the trigger will also completely ignore any inoperative triggers.

The trigger name can still be specified in the DROP TRIGGER and COMMENT ON TRIGGER statements.

An inoperative trigger may be recreated by issuing a CREATE TRIGGER statement using the definition text of the inoperative trigger. This trigger definition text is stored in the TEXT column of SYSCAT.TRIGGERS. Note that there is no need to explicitly drop the inoperative trigger in order to recreate it. Issuing a CREATE TRIGGER statement with the same *trigger-name* as an inoperative trigger will cause that inoperative trigger to be replaced with a warning (SQLSTATE 01595).

Inoperative triggers are indicated by an X in the VALID column of the SYSCAT.TRIGGERS catalog view.

- **Errors executing triggers:** Errors that occur during the execution of triggered SQL statements are returned using SQLSTATE 09000 unless the error is considered severe. If the error is severe, the severe error SQLSTATE is returned. The SQLERRMC field of the SQLCA for non-severe error will include the trigger name, SQLCODE, SQLSTATE and as many tokens as will fit from the tokens of the failure.

The *SQL-procedure-statement* could include a SIGNAL SQLSTATE statement or contain a RAISE_ERROR function. In both these cases, the SQLSTATE returned is the one specified in the SIGNAL SQLSTATE statement or the RAISE_ERROR condition.

CREATE WRAPPER

Linux uses libraries called LIBDRDA.SO and LIBSQLNET.SO, not LIBDRDA.A and LIBSQLNET.A.

DECLARE CURSOR

Within the 『DECLARE CURSOR』 statement, near the end of the Notes section the following sentence should be changed from:

An ambiguous cursor is considered read-only if the BLOCKING bind option is ALL, otherwise it is considered deletable.

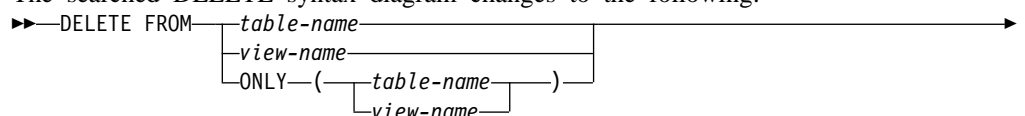
to:

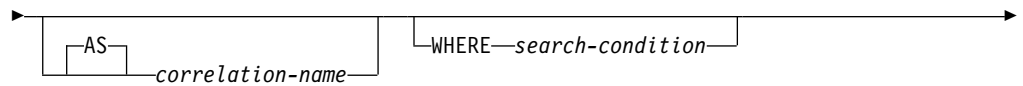
An ambiguous cursor is considered read-only if the BLOCKING bind option is ALL; otherwise, it is considered updatable.

The change is from the word 『deletable』 to the word 『updatable』.

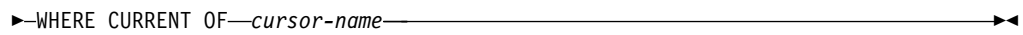
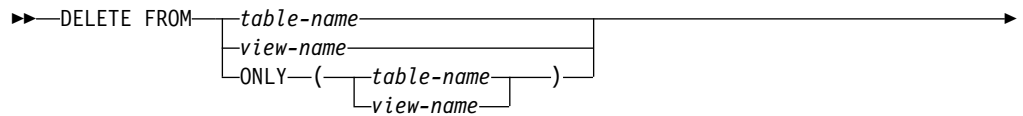
DELETE

The searched DELETE syntax diagram changes to the following:





Positioned DELETE:



Add the following to the description section:

WITH

Specifies the isolation level used when locating the rows to be deleted.

RR

Repeatable Read

RS

Read Stability

CS

Cursor Stability

UR

Uncommitted Read

The default isolation level of the statement is the isolation level of the package in which the statement is bound.

DROP

Add the following option:



Add the following parameters:

SEQUENCE *sequence-name* RESTRICT

Identifies the particular sequence that is to be dropped. The *sequence-name*, along with the implicit or explicit schema name, must identify an existing sequence at the current server. If no sequence by this name exists in the explicitly or implicitly specified schema, an error (SQLSTATE 42704) is raised.

The RESTRICT keyword enforces the rule that the sequence is not dropped if the definition of a table column refers to the sequence (through an IDENTITY column).

Note:

- System created sequences for IDENTITY columns cannot be dropped using the DROP sequence command.
- When a sequence is dropped, all privileges on the sequence are also dropped.

The table showing the dependencies that objects have on each other (Table 27) needs to be updated as follows:

New row: DROP SEQUENCE The entry at the intersection of the new row "DROP SEQUENCE" and the column "PACKAGE" will be an "A". The rest of the entries in this new row will be "-"

GRANT (Sequence Privileges)

GRANT (Sequence Privileges)

This form of the GRANT statement grants privileges on a user-defined sequence.

Invocation: This statement can be embedded in an application program or issued through the use of dynamic SQL statements. It is an executable statement that can be dynamically prepared. However, if the bind option DYNAMICRULES BIND applies, the statement cannot be dynamically prepared (SQLSTATE 42509).

Authorization: The privileges held by the authorization ID of the statement must include at least one of the following:

- Owner of the sequence
- SYSADM or DBADM authority

Syntax:

►► GRANT USAGE ON SEQUENCE *sequence-name* TO PUBLIC

Description:

USAGE

Grants the USAGE privilege for a sequence.

ON SEQUENCE *sequence-name*

Identifies the sequence on which the USAGE privilege is to be granted. The *sequence-name*, including the implicit or explicit schema qualifier, must uniquely identify an existing sequence at the current server. If no sequence by this name exists in the specified schema, an error (SQLSTATE 42704) is raised.

TO PUBLIC

Grants the USAGE privilege to all users.

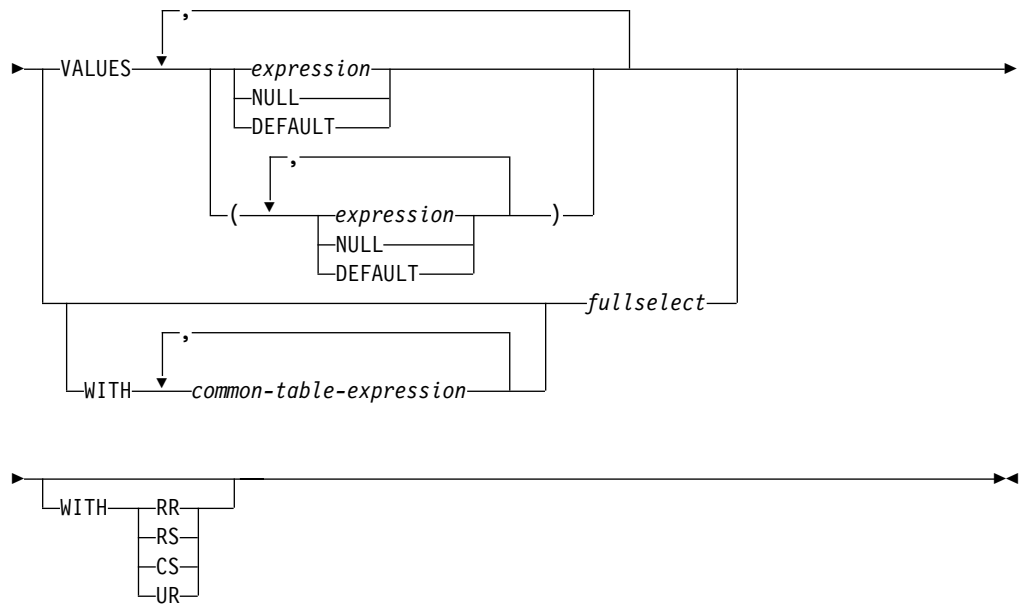
範例: *Example 1:* Grant any user the privilege on a sequence called MYNUM

```
GRANT USAGE ON SEQUENCE MYNUM TO PUBLIC
```

INSERT

Syntax diagram changes to:

►► INSERT INTO *table-name* | *view-name* (*column-name*)



Add the following to the description section:

WITH

Specifies the isolation level at which the fullselect is executed.

RR

Repeatable Read

RS

Read Stability

CS

Cursor Stability

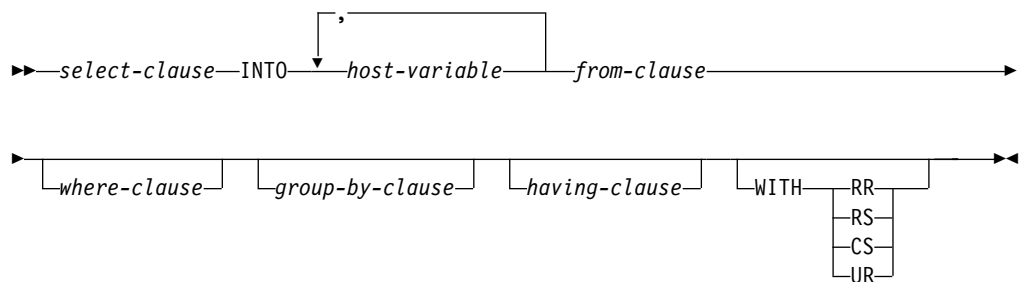
UR

Uncommitted Read

The default isolation level of the statement is the isolation level of the package in which the statement is bound.

SELECT INTO

The syntax diagram changes to:



Add the following to the description section:

WITH

Specifies the isolation level at which the `SELECT INTO` statement is executed.

RR

Repeatable Read

RS

Read Stability

CS

Cursor Stability

UR

Uncommitted Read

The default isolation level of the statement is the isolation level of the package in which the statement is bound.

SET ENCRYPTION PASSWORD

SET ENCRYPTION PASSWORD

The SET ENCRYPTION PASSWORD statement sets the password that will be used by the encryption and decryption functions. The password is not tied to DB2 authentication, and is used for data encryption only.

This statement is not under transaction control.

Invocation: The statement can be embedded in an application program or issued interactively. It is an executable statement that can be dynamically prepared.

Authorization: No authorization is required to execute this statement.

Syntax:

```
▶▶ SET ENCRYPTION PASSWORD   host-variable | string-constant ▶▶
```

Description: The ENCRYPTION PASSWORD can be used by the ENCRYPT, DECRYPT_BIN, and DECRYPT_CHAR built-in functions for password based encryption. The length must be between 6 and 127 inclusive. All characters must be specified in the exact case intended as there is no conversion to uppercase characters.

host-variable

A variable of type CHAR or VARCHAR. The length of the contents of the host-variable must be between 6 and 127 inclusive (SQLSTATE 428FC). It cannot be set to null. All characters must be specified in the exact case intended as there is no conversion to uppercase characters.

string-constant

A character string constant. The length must be between 6 and 127 inclusive (SQLSTATE 428FC).

規則:

- The initial ENCRYPTION PASSWORD value is the empty string ('').
- The host-variable or string-constant is transmitted to the database server using normal DB2 mechanisms.

附註:

- See 第351頁的『ENCRYPT』 and 第350頁的『DECRYPT_BIN and DECRYPT_CHAR』 for additional information on using this statement.

Example: *Example 1:* The following statement sets the ENCRYPTION PASSWORD.

```
SET ENCRYPTION PASSWORD = 'bubbalu'
```

SET transition-variable

This section changes to the following:

SET Variable

The SET Variable statement assigns values to local variables or to new transition variables. It is under transaction control.

Invocation: This statement can only be used as an SQL statement in either a dynamic compound statement, trigger, SQL function or SQL method.

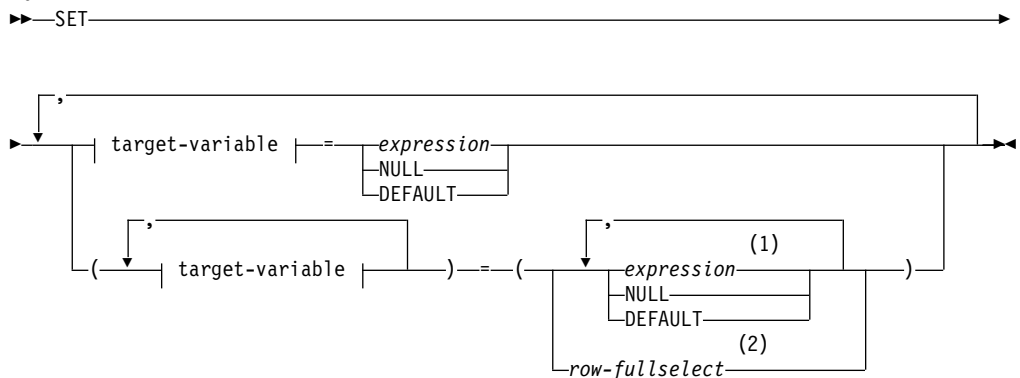
Authorization: To reference a transition variable, the privileges held by the authorization ID of the trigger creator must include at least one of the following:

- UPDATE of the columns referenced on the left hand side of the assignment and SELECT for any columns referenced on the right hand side.
- CONTROL privilege on the table (subject table of the trigger)
- SYSADM or DBADM authority.

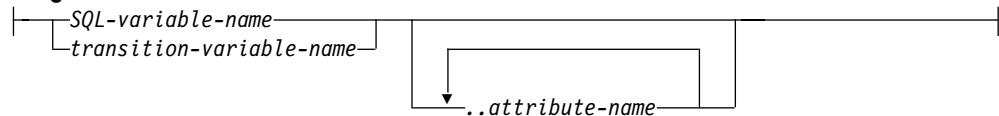
To execute this statement with a *row-fullselect* as the right hand side of the assignment, the privileges held by the authorization ID of either the trigger definer or the dynamic compound statement owner must also include at least one of the following, for each table or view referenced:

- SELECT privilege
- CONTROL privilege
- SYSADM or DBADM.

Syntax:



target-variable:



Note:

- 1 The number of expressions, NULLs and DEFAULTs must match the number of *target-variables*.
- 2 The number of columns in the select list must match the number of *target-variables*.

Description:

target-variable

Identifies the target variable of the assignment. A *target-variable* representing the same variable must not be specified more than once (SQLSTATE 42701).

SQL-variable-name

Identifies the SQL variable that is the assignment target. SQL variables must be declared before they are used. SQL variables can be defined in a dynamic compound statement.

transition-variable-name

Identifies the column to be updated in the transition row. A *transition-variable-name* must identify a column in the subject table of a trigger, optionally qualified by a correlation name that identifies the new value (SQLSTATE 42703).

..attribute name

Specifies the attribute of a structured type that is set (referred to as an *attribute assignment*). The *SQL-variable-name* or *transition-variable-name* specified must be defined with a user-defined structured type (SQLSTATE 428DP). The *attribute-name* must be an attribute of the structured type (SQLSTATE 42703). An assignment that does not involve the *..attribute name* clause is referred to as a conventional assignment.

expression

Indicates the new value of the *target-variable*. The expression is any expression of the type described in Chapter 2 of the SQL Reference. The expression can not include a column function except when it occurs within a scalar fullselect (SQLSTATE 42903). In the context of a CREATE TRIGGER statement, an *expression* may contain references to OLD and NEW transition variables and must be qualified by the *correlation-name* to specify which transition variable (SQLSTATE 42702).

NULL

Specifies the null value and can only be specified for nullable columns (SQLSTATE 23502). NULL cannot be the value in an attribute assignment (SQLSTATE 429B9), unless it was specifically cast to the data type of the attribute.

DEFAULT

Specifies that the default value should be used.

If *target-variable* is a column, the value inserted depends on how the column was defined in the table.

- If the column was defined using the WITH DEFAULT clause, then the value is set to the default defined for the column.
- If the column was defined using the IDENTITY clause, the value is generated by the database manager.
- If the column was defined without specifying the WITH DEFAULT clause, the IDENTITY clause, or the NOT NULL clause, then the value is NULL.
- If the column was defined using the NOT NULL clause and the IDENTITY clause is not used, or the WITH DEFAULT clause was not used or DEFAULT NULL was used, the DEFAULT keyword cannot be specified for that column (SQLSTATE 23502).

If *target-variable* is an SQL variable, then the value inserted is the default as specified or implied in the variable declaration.

row-fullselect

A fullselect that returns a single row with the number of columns corresponding to the number of target-variables specified for assignment. The values are assigned to each corresponding target-variable. If the result of the row-fullselect is no rows, then null values are assigned. In the context of a CREATE TRIGGER statement, a *row-fullselect* may contain references to OLD and NEW transition variables which must be qualified by their *correlation-name* to specify which transition variable to use (SQLSTATE 42702). An error is returned if there is more than one row in the result (SQLSTATE 21000).

Rule:

- The number of values to be assigned from expressions, NULLs and DEFAULTs or the *row-fullselect* must match the number of target-variables specified for assignment (SQLSTATE 42802).
- A SET Variable statement cannot assign an SQL variable and a transition variable in one statement (SQLSTATE 42997).
- Values are assigned to target-variables under the assignment rules described in Chapter 2 of the SQL Reference.

If the statement is used in a BEFORE UPDATE trigger, and the registry variable DB2_UPDATE_PART_KEY=OFF, then a *transition-variable* specified as *target-variable* cannot be a partitioning key column (SQLSTATE 42997).

Version:

- If more than one assignment is included, all *expressions* and *row-fullselects* are evaluated before the assignments are performed. Thus references to target-variables in an expression or row fullselect are always the value of the target-variable prior to any assignment in the single SET statement.
- When an identity column defined as a distinct type is updated, the entire computation is done in the source type, and the result is cast to the distinct type before the value is actually assigned to the column.⁸
- To have DB2 generate a value on a SET statement for an identity column, use the DEFAULT keyword:

```
SET NEW.EMPNO = DEFAULT
```

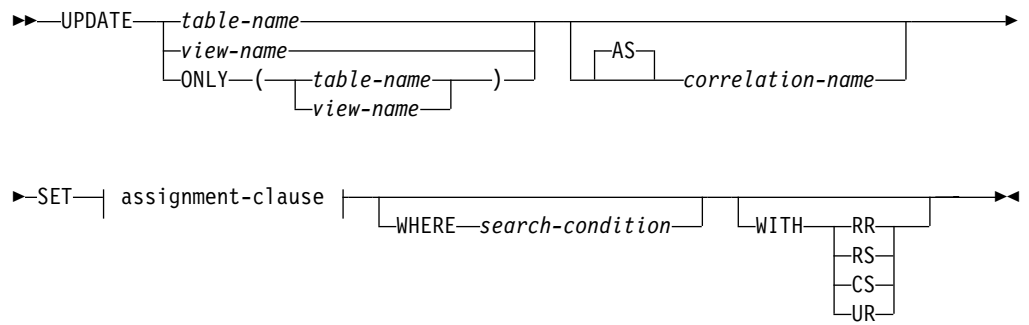
In this example, NEW.EMPNO is defined as an identity column, and the value used to update this column is generated by DB2.

The examples for this statement stay the same.

UPDATE

The searched UPDATE syntax diagram is changed to:

8. There is no casting of the previous value to the source type prior to the computation.



Add the following to the description section:

WITH

Specifies the isolation level at which the UPDATE statement is executed.

RR

Repeatable Read

RS

Read Stability

CS

Cursor Stability

UR

Uncommitted Read

The default isolation level of the statement is the isolation level of the package in which the statement is bound.

Chapter 7. SQL Procedures now called Chapter 7. SQL Control Statements

Control statements are SQL statements that allow SQL to be used in a manner similar to writing a program in a structured programming language. SQL control statements can be used in the body of a routine, trigger or a dynamic compound statement.

This chapter contains the syntax and descriptions of the supported SQL control statements, along with the SQL-procedure-statement.

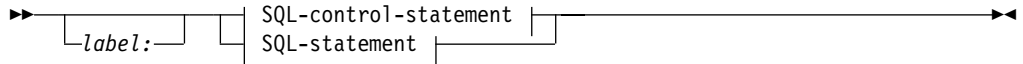
SQL Procedure Statement

The SQL Procedure Statement information changes to the following:

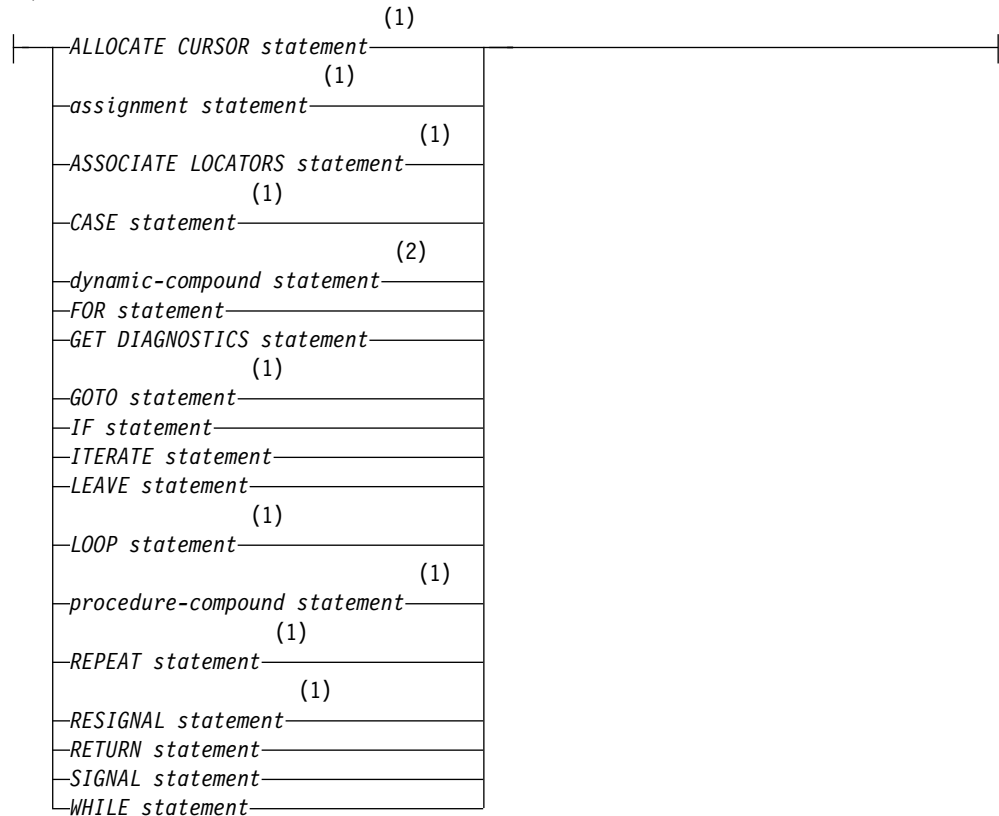
SQL Procedure Statement

This chapter contains syntax diagrams, semantic descriptions, rules, and examples of the use of the statements that constitute the procedure body of an SQL routine, trigger, or dynamic compound statement.

Syntax:



SQL-control-statement:



Note:

- 1 This statement is only supported in the scope of an SQL Procedure.
- 2 This statement is only supported within a trigger, SQL function, or SQL method. It must be the outermost statement.

Description:

label:

Specifies the label for an SQL procedure statement. The label must be unique within a list of SQL procedure statements, including any compound statements nested within the list. Note that compound statements that are not nested may use the same label. A list of SQL procedure statements is possible in a number of SQL control statements.

In the context of a trigger, an SQL function or method, or a dynamic compound statement, only the dynamic compound statement, the FOR statement and the WHILE statement may be labeled.

SQL-statement

In the body of an SQL procedure, all executable SQL statements can be contained, with the exception of the following:

- CONNECT
- CREATE any object other than indexes, tables, or views
- DESCRIBE
- DISCONNECT
- DROP any object other than indexes, tables, or views
- FLUSH EVENT MONITOR
- REFRESH TABLE
- RELEASE (connection only)
- RENAME TABLE
- RENAME TABLESPACE
- REVOKE
- SET CONNECTION
- SET INTEGRITY

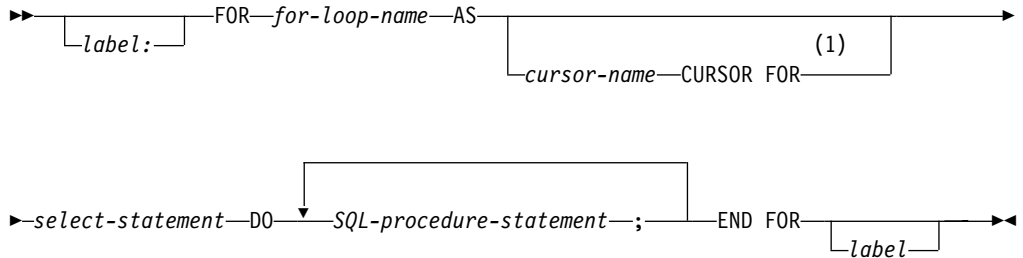
Note: You may include CALL statements within an SQL procedure body, but these CALL statements can only call another SQL procedure or a C procedure. CALL statements within an SQL procedure body cannot call other types of stored procedures.

FOR

FOR

The FOR statement executes a statement or group of statements for each row of a table.

Syntax:



Note:

- 1 This option can only be used in the context of an SQL Procedure.

Description:

label

Specifies the label for the FOR statement. If the beginning label is specified, that label can be used in LEAVE and ITERATE statements. If the ending label is specified, it must be the same as the beginning label.

for-loop-name

Specifies a label for the implicit compound statement generated to implement the FOR statement. It follows the rules for the label of a compound statement except that it cannot be used with and ITERATE or LEAVE statement within the FOR statement. The *for-loop-name* is used to qualify the column names returned by the specified *select-statement*.

cursor-name

Names the cursor that is used to select rows from the result table from the SELECT statement. If not specified, DB2 generates a unique cursor name.

select-statement

Specifies the SELECT statement of the cursor. All columns in the select list must have a name and there cannot be two columns with the same name.

In a trigger, function, method, or dynamic compound statement, the *select-statement* must consist of only a *fullselect* with optional common table expressions.

SQL-procedure-statement

Specifies a statement (or statements) to be invoked for each row of the table.

Rule:

- The select list must consist of unique column names and the table specified in the select list must exist when the procedure is created, or it must be a table created in a previous SQL procedure statement.
- The cursor specified in a for-statement cannot be referenced outside the for-statement and cannot be specified in an OPEN, FETCH, or CLOSE statement.

Example: In the following example, the for-statement is used to iterate over the entire employee table. For each row in the table, the SQL variable fullname is set to the last

name of the employee, followed by a comma, the first name, a blank space, and the middle initial. Each value for fullname is inserted into table tnames.

```

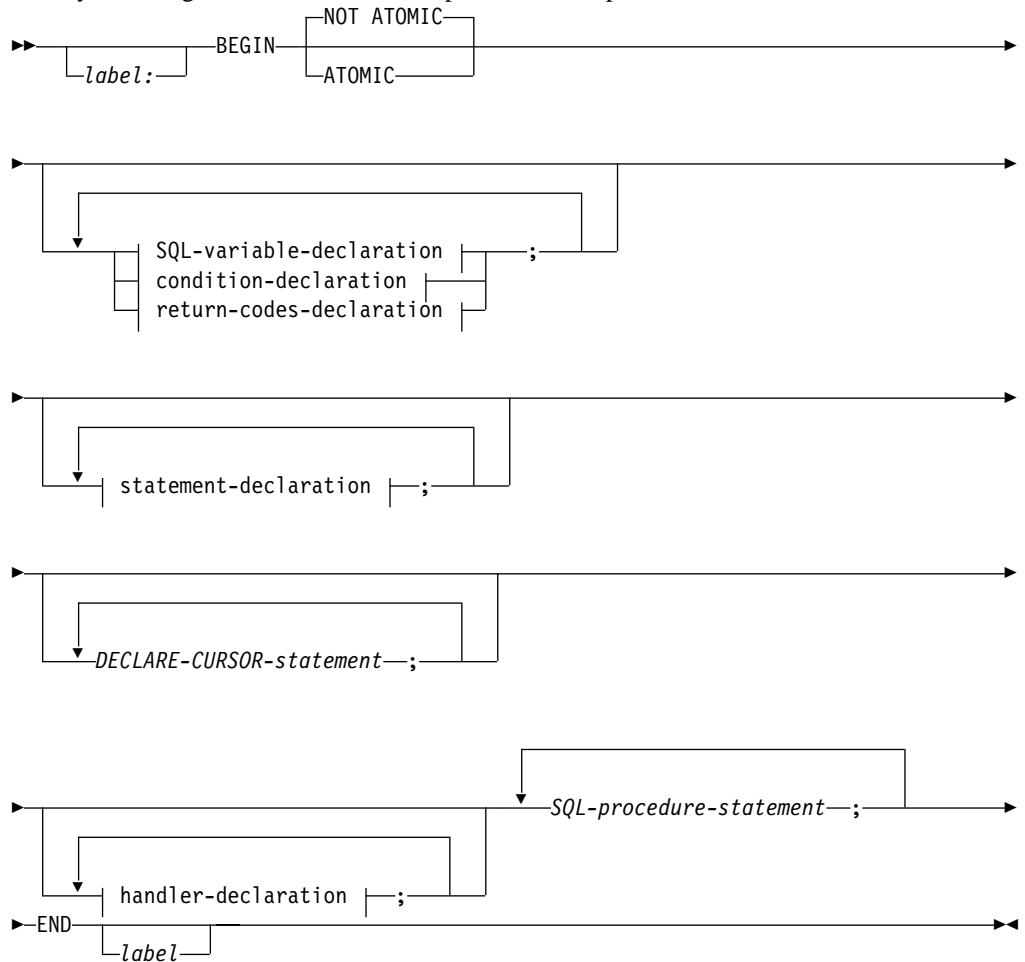
BEGIN
  DECLARE fullname CHAR(40);
  FOR v1 AS
    SELECT firstnme, midinit, lastname FROM employee
  DO
    SET fullname = lastname || ',' || firstnme || ' ' || midinit;
    INSERT INTO tnames VALUE (fullname);
  END FOR
END

```

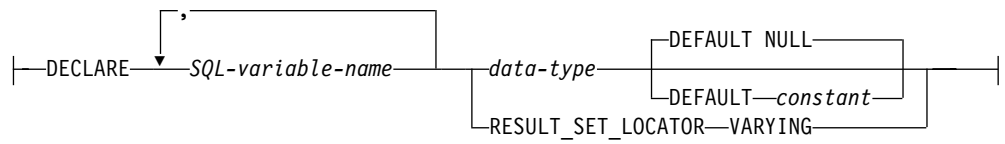
Compound Statement changes to Compound Statement (Procedure)

A procedure compound statement groups other statements together in an SQL procedure. You can declare SQL variables, cursors, and condition handlers within a compound statement.

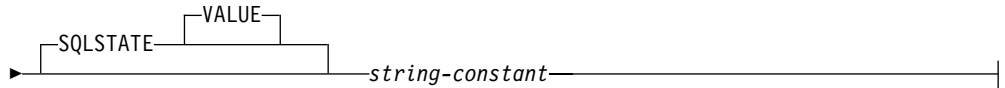
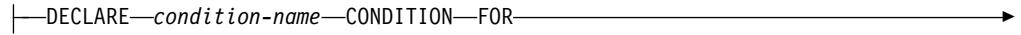
The syntax diagram now has a title: procedure-compound-statement.



SQL-variable-declaration:



condition-declaration:



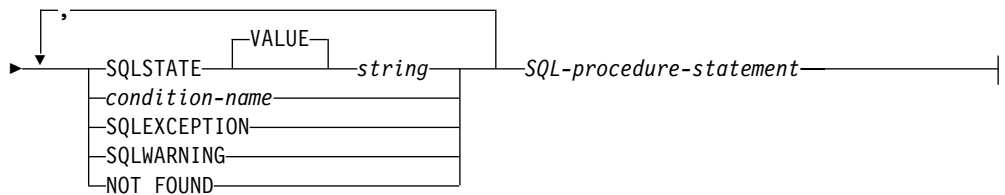
statement-declaration:



return-codes-declaration:



handler-declaration:



statement-declaration

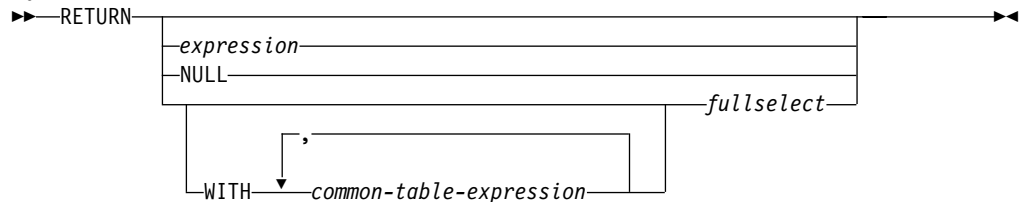
A *statement-declaration* declares a list of one or more names that are local to the compound statement. A statement name cannot be the same as another statement name within the same compound statement.

RETURN

RETURN

The RETURN statement is used to return from the routine. For SQL functions or methods, it returns the result of the function or method. For an SQL procedure, it optionally returns an integer status value.

Syntax:



Description:

expression

Specifies a value that is returned from the routine:

- If the routine is a function or method, one of *expression*, NULL, or *fullselect* must be specified (SQLSTATE 42630) and the data type of the result must be assignable to the RETURNS type of the routine (SQLSTATE 42866).
- A scalar expression (other than a scalar fullselect) cannot be specified for a table function (SQLSTATE 428F1).
- If the routine is a procedure, the data type of *expression* must be INTEGER (SQLSTATE 428E2). A procedure cannot return NULL or a fullselect.

NULL

Specifies that the function or method returns a null value of the data type defined in the RETURNS clause. NULL cannot be specified for a RETURN from a procedure.

WITH *common-table-expression*

Defines a common table expression for use with the fullselect that follows.

fullselect

Specifies the row or rows to be returned for the function. The number of columns in the fullselect must match the number of columns in the function result (SQLSTATE 42811). In addition, the static column types of the fullselect must be assignable to the declared column types of the function result, using the rules for assignment to columns (SQLSTATE 42866).

The *fullselect* cannot be specified for a RETURN from a procedure.

If the routine is a scalar function or method, then the fullselect must return one column (SQLSTATE 42823) and, at most, one row (SQLSTATE 21000).

If the routine is a row function, it must return, at most, one row (SQLSTATE 21505).

If the routine is a table function, it can return zero or more rows with one or more columns.

Rule:

- The execution of an SQL function or method must end with a RETURN (SQLSTATE 42632).
- In an SQL table or row function using a *dynamic-compound-statement*, the only RETURN statement allowed is the one at the end of the compound statement (SQLSTATE 429BD).

Version:

- When a value is returned from a procedure, the caller may access the value using:
 - the GET DIAGNOSTICS statement to retrieve the RETURN_STATUS when the SQL procedure was called from another SQL procedure
 - the parameter bound for the return value parameter marker in the escape clause CALL syntax (?=CALL...) in a CLI application
 - directly from the SQLCA returned from processing the CALL of an SQL procedure by retrieving the value of SQLERRD[0] when the SQLCODE is not less than zero (assume a value of -1 when SQLCODE is less than zero).

Examples: Use a RETURN statement to return from an SQL stored procedure with a status value of zero if successful, and -200 if not.

```

BEGIN
...
  GOTO FAIL
...
  SUCCESS: RETURN 0
  FAIL: RETURN -200
END

```

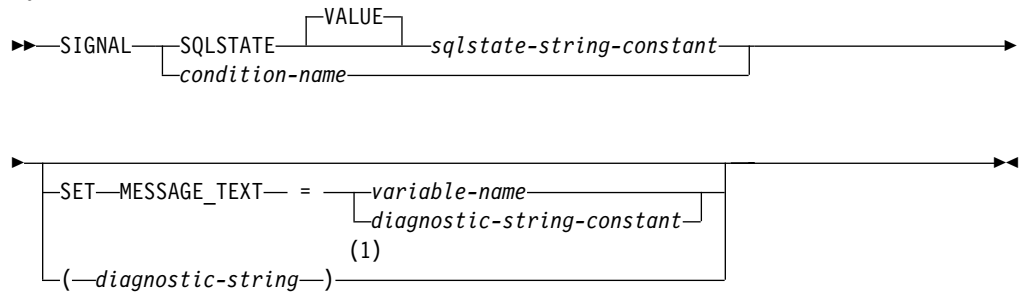
SIGNAL

The SIGNAL SQLSTATE Statement is no longer used, in favor of this usage.

SIGNAL

The SIGNAL statement is used to signal an error or warning condition. It causes an error or warning to be returned with the specified SQLSTATE, along with optional message text.

Syntax:



Note:

- 1 This option is only provided within the scope of a CREATE TRIGGER statement for compatibility with older versions of DB2.

Description:

SQLSTATE VALUE *sqlstate-string-constant*

The specified string constant represents an SQLSTATE. It must be a character string constant with exactly 5 characters that follow the rules for SQLSTATES:

- Each character must be from the set of digits ('0' through '9') or non-accented upper case letters ('A' through 'Z').
- The SQLSTATE class (first two characters) cannot be '00', since this represents successful completion.

In the context of either a dynamic compound statement, trigger, SQL function, or SQL method, the following rules must also be applied:

- The SQLSTATE class (first two characters) cannot be '01' or '02', since these are not error classes.
- If the SQLSTATE class starts with the numbers '0' through '6' or the letters 'A' through 'H', then the subclass (the last three characters) must start with a letter in the range of 'T' through 'Z'.
- If the SQLSTATE class starts with the numbers '7', '8', '9', or the letters 'I' through 'Z', then the subclass can be any of '0' through '9' or 'A' through 'Z'.

If the SQLSTATE does not conform to these rules, an error is raised (SQLSTATE 428B3).

condition-name

Specifies the name of the condition. The condition name must be unique within the procedure and can only be referenced within the compound statement in which it is declared.

SET MESSAGE_TEXT=

Specifies a string that describes the error or warning. The string is returned in the

SQLERRMC field of the SQLCA. If the actual string is longer than 70 bytes, it is truncated without warning. This clause can only be specified if a SQLSTATE or condition-name is also specified (SQLSTATE 42601).

variable-name

Identifies an SQL variable that must be declared within the compound statement. The SQL variable must be defined as a CHAR or VARCHAR data type.

diagnostic-string-constant

Specifies a character string constant that contains the message text.

diagnostic-string

An expression with a type of CHAR or VARCHAR that returns a character string of up to 70 bytes to describe the error condition. If the string is longer than 70 bytes, it will be truncated. This option is only provided within the scope of a CREATE TRIGGER statement, for compatibility with older versions of DB2. Regular use is not recommended.

Note:

- If a SIGNAL statement is issued, the SQLCODE that is assigned is:
 - +438 if the SQLSTATE begins with '01' or '02'
 - 438 otherwise
- If the SQLSTATE or condition indicates that an exception (SQLSTATE class other than '01' or '02') is signaled:
 - Then the exception is handled and control is transferred to a handler, provided that a handler exists in the same compound statement (or an outer compound statement) as the signal statement, and the compound statement contains a handler for the specified SQLSTATE, condition-name, or SQLEXCEPTION;
 - If the exception cannot be handled, then control is immediately returned to the end of the compound statement.
- If the SQLSTATE or condition indicates that a warning (SQLSTATE class '01') or not found condition (SQLSTATE class '02') is signaled:
 - Then the warning or not found condition is handled and control is transferred to a handler, provided that a handler exists in the same compound statement (or an outer compound statement) as the signal statement, and the compound statement contains a handler for the specified SQLSTATE, condition-name, SQLWARNING (if the SQLSTATE class is '01'), or NOT FOUND (if the SQLSTATE class is '02');
 - If the warning cannot be handled, then processing continues with the next statement.
- SQLSTATE values are comprised of a two-character class code value, followed by a three-character subclass code value. Class code values represent classes of successful and unsuccessful execution conditions.

Any valid SQLSTATE value can be used in the SIGNAL statement. However, it is recommended that programmers define new SQLSTATES based on ranges reserved for applications. This prevents the unintentional use of an SQLSTATE value that might be defined by the database manager in a future release.

- SQLSTATE classes that begin with the characters '7' through '9', or 'I' through 'Z' may be defined. Within these classes, any subclass may be defined.

- SQLSTATE classes that begin with the characters '0' through '6', or 'A' through 'H' are reserved for the database manager. Within these classes, subclasses that begin with the characters '0' through 'H' are reserved for the database manager. Subclasses that begin with the characters 'I' through 'Z' may be defined.

Examples: An SQL procedure for an order system that signals an application error when a customer number is not known to the application. The ORDERS table includes a foreign key to the CUSTOMER table, requiring that the CUSTNO exist before an order can be inserted.

```
CREATE PROCEDURE SUBMIT_ORDER
  (IN ONUM INTEGER, IN CNUM INTEGER,
   IN PNUM INTEGER, IN QNUM INTEGER)
  SPECIFIC SUBMIT_ORDER
  MODIFIES SQL DATA
  LANGUAGE SQL
  BEGIN
    DECLARE EXIT HANDLER FOR SQLSTATE VALUE '23503'
      SIGNAL SQLSTATE '75002'
      SET MESSAGE_TEXT = 'Customer number is not known';
    INSERT INTO ORDERS (ORDERNO, CUSTNO, PARTNO, QUANTITY)
      VALUES (ONUM, CNUM, PNUM, QNUM);
  END
```

Appendix A. SQL Limits

There is a change to Table 33, Database Manager Limits. With the registry variable DB2_INDEX_2BYTEVARLEN set to ON, the longest variable index key part (in bytes) can now be greater than 255.

Appendix D. Catalog Views

A new catalog view has been added:

SYSCAT.SEQUENCES

The view SYSCAT.SEQUENCES is automatically generated for databases created with FixPak 3 or later. For databases created prior to FixPak 3, run the **db2updv7** command in order to add the view to the database. See the Command Reference update in the Release Notes for details.

This catalog view is updated during normal operation, in response to SQL data definition statements, environment routines, and certain utilities. Data in the catalog view is available through normal SQL query facilities. Columns have consistent names based on the type of objects that they describe.

Table 30. Columns in SYSCAT.SEQUENCES Catalog View

Column Name	Data Type	Nullable	Description
SEQSHEMA	VARCHAR(128)		Schema of the sequence.
SEQNAME	VARCHAR(128)		Sequence name (generated by DB2 for an identity column).
DEFINER	VARCHAR(128)		Definer of the sequence.
OWNER	VARCHAR(128)		Owner of the sequence.
SEQID	INTEGER		Internal ID of the sequence.

Table 30. Columns in SYSCAT.SEQUENCES Catalog View (continue)

Column Name	Data Type	Nullable	Description
SEQTYPE	CHAR(1)		Sequence type S - Regular sequence
INCREMENT	DECIMAL(31,0)		Increment value.
START	DECIMAL(31,0)		Starting value.
MAXVALUE	DECIMAL(31,0)	Yes	Maximal value.
MINVALUE	DECIMAL(31,0)		Minimum value.
CYCLE	CHAR(1)		Whether cycling will occur when a boundary is reached: Y - cycling will occur N - cycling will not occur
CACHE	INTEGER		Number of sequence values to preallocate in memory for faster access. 0 indicates that values are not preallocated.
ORDER	CHAR(1)		Whether or not the sequence numbers must be generated in order of request: Y - sequence numbers must be generated in order of request N - sequence numbers are not required to be generated in order of request
DATATYPEID	INTEGER		For built-in types, the internal ID of the built-in type. For distinct types, the internal ID of the distinct type.
SOURCETYPEID	INTEGER		For a built-in type, this has a value of 0. For a distinct type, this is the internal ID of the built-in type that is the source type for the distinct type.
CREATE_TIME	TIMESTAMP		Time when the sequence was created.
ALTER_TIME	TIMESTAMP		Time when the last ALTER SEQUENCE statement was executed for this sequence.
PRECISION	SMALLINT		The precision defined for a sequence with a decimal or numeric type. Values are: 5 for a SMALLINT, 10 for INTEGER, and 19 for BIGINT.
ORIGIN	CHAR(1)		Sequence Origin U - User generated sequence S - System generated sequence
REMARKS	VARCHAR(254)	Yes	User supplied comments, or null.

IBM DB2 儲存程序建置器

Java 1.2 支援儲存程序建置器

DB2 儲存程序建置器支援 Java 儲存程序使用 Java 1.2 功能。另外，儲存程序建置器支援雙向語言，如阿拉伯語和希伯來語，所使用的是 Java 1.2 的 bi-di 支援。

此支援只用於 Windows NT 平台而已。

為了儲存程序建置器辨識和使用 Java 1.2 的功能，所以要進行安裝 Java 1.2。

安裝 Java 1.2：

1. JDK 1.2.2 在 DB2 UDB CD 的 DB2\bid\NT 目錄下。
ibm-inst-n122p-win32-x86.exe 是安裝程式，ibm-jdk-n122p-win32-x86.exe 是 JDK 分送。將兩個檔案複製到硬碟上的暫時目錄中，然後從該目錄執行安裝程式。
2. 將它安裝在 <DB2PATH\java\Java12 下，<DB2PATH 是 DB2 的安裝路徑。
3. JDK/JRE 安裝提示時，不要選取 JDK/JRE 為系統 VM。

Java 1.2 安裝成功後，以正常方式啟動儲存程序建置器。

欲使用 JDK 1.2 支援執行 Java 儲存程序，必須設定資料庫伺服器環境變數 DB2_USE_JDK12 為真，請使用下列指令：

```
DB2SET DB2_USE_JDK12=TRUE
```

同時，設定 JDK11_PATH 指向安裝 Java 1.2 支援的目錄。請使用下列指令來設定此路徑：

```
DB2 UPDATE DBM CFG USING JDK11_PATH
```

欲停止使用 Java 1.2，請在 <DB2PATH>\java\Java12 下解除安裝 JDK/JRE，或更改 <DB2PATH>\java\Java12 次目錄的名字。

重要事項：不要將 <DB2PATH>\java\Java12 和 <DB2PATH>\Java12 混淆。<DB2PATH>\Java12 是 DB2 安裝的一部份，並且包括 Java 1.2 的 JDBC 支援。

DB2 儲存程序的遠端除錯

若要使用 UNIX 和 Windows 平台的 Java 和 C 儲存程序之遠端除錯能力，請安裝 IBM Distributed Debugger。IBM Distributed Debugger 可於 Visual Age for Java Professional Edition CD 內取得。除錯器從屬站只在 Windows 平台上執行。支援伺服器平台包括：Windows、AIX 和 Solaris。請使用儲存程序建置器內建的 SQL 除錯能力，為 Windows 和 UNIX 平台除錯本端和遠端 SQL 儲存程序。此時無法支援 OS/2 平台。有關 DB2 for OS/390 儲存程序建置器的詳細資訊，請連接到下列網站：
<http://www-4.ibm.com/software/data/db2/os390/spb/exciting>

欲在 OS/390 平台進行 SQL 程序除錯，也必須有 OS/390 R1 產品的 IBM C/C++ 輔助工具。OS/390 R1 的 IBM C/C++ 輔助工具之詳細資訊，請連接到下列網站：
<http://www.ibm.com/software/ad/c390/pt/>

在 Windows、OS/2 或 UNIX 平台上開發 SQL 程序

在您能夠使用儲存程序建置器於 Windows、OS/2 或 UNIX 資料庫上順利地開發 SQL 程序之前，必須架構 SQL 程序的伺服器。有關如何架構 SQL 程序的伺服器，請參閱第307頁的『第 4 章 開發 Java Applets 和應用程式』。

在 Solaris 平台使用 DB2 儲存程序建置器

欲在 Solaris 平台使用 DB2 儲存程序建置器：

1. 下載和安裝 JDK 1.1.8。自 JavaSoft 網站下載 JDK 1.1.8。
2. 設定環境變數 JAVA_HOME 至安裝 JDK 的位置。
3. 設定 DB2 JDK11_PATH 至安裝 JDK 的目錄下。

欲設定 DB2 JDK11_PATH，請使用下列指令：

```
DB2 UPDATE DBM CFG USING JDK11_PATH.
```

已知問題與限制

- 目前 Windows 98 並不支援 SQL 程序。
- 關於 Java 儲存程序，JAR ID、類別名稱與方法名稱不可包含非 ASCII 字元。
- 在 AS/400 上下列 V4R4 PTFs 會引用至 OS/400 V4R4：
 - SF59674
 - SF59878
- 儲存程序儲存於資料庫時，具 FOR MIXED DATA 或 FOR SBCS DATA 次類型的儲存程序參數，就不會顯示在編輯器窗格的原始碼中。
- 目前的問題是 Java 原始碼是由資料庫擷取來的。擷取時程式碼的說明會呈現收合的方式。這會影響在非 ASCII 字碼頁上使用 DB2 儲存程序建置器的使用者，因為他們的從屬站與伺服器使用不同的字碼頁。

在繁體中文語言環境下使用 DB2 儲存程序建置器

在繁體中文語言環境下使用 JDK 或 JRE 1.1.8 時會發生問題。儲存程序建置器程式的圖形式介面 (包含功能表、編輯器文字、訊息等等) 無法適當顯示。解決方案是修改 font.properties.zh_TW 檔案，它會在下列目錄中：

```
sql1lib/java/jdk/lib  
sql1lib/java/jre/lib
```

變更：

```
monospaced.0=\u7d30\u660e\u9ad4,CHINESEBIG5_CHARSET,NEED_CONVERTED
```

改為：

```
monospaced.0=Courier New,ANSI_CHARSET
```

UNIX (AIX、Sun Solaris、Linux) 安裝與儲存程序建置器

在 Sun Solaris 安裝時，若您使用 Java Development Kit 或 Runtime 而不是在 AIX 上安裝 UDB 的方式，您就必須將環境變數 JAVA_HOME 設定到已安裝 Java 的路徑中 (也就是該目錄須含 /bin 與 /lib 目錄)。儲存程序建置器未支援 Linux 作業系統，但可在已支援的平台上建置或執行 DB2 UDB for Linux 系統上的儲存程序。

支援的平台包括用於從屬站的 AIX、Solaris 及 NT，與用於伺服器的 AIX、Solaris、NT、Linux、OS/2、HP-UX 及 NUMA-Q。

建置 OS/390 的 SQL 儲存程序

DB2 儲存程序建置器支援在 OS390 V7 伺服器的 DB2 UDB 上建置 SQL 儲存程序。

除錯 SQL 儲存程序

現在，SQL 儲存程序在 Windows 及 UNIX 平台上的除錯，已直接整合到 DB2 儲存程序建置器中。在除錯非隔離 (信任的) SQL 程序時，KEEPDARI 資料庫管理員設定選項可設定為 YES 或 NO；但是，在除錯隔離 (非信任的) SQL 程序時，必須設定為 YES (預設值)。請參閱儲存程序建置器的說明，以取得有關使用整合除錯器的其他資訊。

匯出 Java 儲存程序

DB2 儲存程序建置器現在支援匯出 Java 儲存程序。若要匯出 Java 儲存程序：

1. 在儲存程序資料夾上按一下滑鼠右鍵，按一下「匯出 Java 儲存程序」以開啓「匯出 Java 儲存程序」視窗。
2. 選取您要匯出的儲存程序，將它們移到「已選取的儲存程序」直欄。
3. 選取您想要的選項之後，請按一下「確定」。

在 OS/390 上插入儲存程序

針對執行 OS/390 的 DB2 儲存程序建置器版本 5 及更新版本，如果您使用精靈來插入儲存程序並指示無 WLM 環境選項，則產生的程式碼會包含以下的文字：NO WLM ENVIRONMENT。這行程式碼會讓儲存程序在 SPAS 位址空間中依預期執行。此修正程式解決存在於 DB2 儲存程序建置器版本 6 及更新版本中的問題。

以修正程式所產生的程式碼顯示如下：

```
CREATE PROCEDURE SYSPROC.Proc2 ( )
  RESULT SETS 1
  LANGUAGE SQL
  MODIFIES SQL DATA
  COLLID TEST
  NO WLM ENVIRONMENT
  ASUTIME NO LIMIT
  RUN OPTIONS 'NOTEST(ALL,*,VADTCP&9.112.14.91:*)'
-----
-- SQL Stored Procedure
-----
P1: BEGIN
  -- Declare cursor
  DECLARE cursor1 CURSOR WITH RETURN FOR
    SELECT * FROM SYSIBM.SYSPROCEDURES;
```

```
-- Cursor left open for client application
OPEN cursor1;

END P1
```

在工作站伺服器中設定 SQL 儲存程序的建置選項

在 UNIX 及 Windows 平台上使用 DB2 儲存程序建置器時，可以為所有的 SQL 儲存程序設定建置選項。這些建置選項包含下列的編譯器及前置編譯器 DB2 登記變數：

- DB2_SQLROUTINE_PREPOPTS
- DB2_SQLROUTINE_COMPILER_PATH
- DB2_SQLROUTINE_COMPILE_COMMAND
- DB2_SQLROUTINE_KEEP_FILES

雖然可以使用 `db2set` 指令設定這些登記變數，使用儲存程序建置器可消除實際存取資料庫伺服器以發出指令或停止，然後重新啟動伺服器使變更生效的需求。

若要開啓「SQL 儲存程序建置選項」視窗，請在專案概略表中的資料庫連線按一下滑鼠右鍵，然後按一下「SQL 儲存程序建置選項」。有關設定這些選項的詳細資訊，請參閱 DB2 儲存程序說明。

自動復新 OS/390 上儲存程序建置的 WLM 位址空間

當您順利地在 OS/390 上建置將在 WLM 執行的儲存程序之後，DB2 儲存程序建置器將自動復新 WLM 位址空間。

在 OS/390 開發 Java 儲存程序

「DB2 儲存程序建置器」支援 DB2 UDB for OS/390 版本 6 (或以上) 的 Java 儲存程序的開發。您可以建立新的或變更現存的 Java 儲存程序。

為 MQ Series 與 OLE DB 建置 DB2 表格使用者定義功能 (UDF)

「DB2 儲存程序建置器」提供精靈以協助您建立 MQSeries 與 OLE DB 的表格 UDF。您可以使用「建立 OLE DB 表格 UDF」精靈存取 OLE DB 資料提供者。此精靈會建立 OLE 表格 UDF 與選用的表格檢視。您可以使用「建立 MQSeries」表格 UDF 精靈來建立具有選項表格檢視的表格 UDF，以存取 MQSeries 訊息並剖析定位標籤格式中的資料。

Unicode 更新

簡介

Unicode 標準是書寫字元及文字方面全球通用的字元編碼方法。Unicode 是用多個位元組來表示一個字元。它定義了一種一致的方式來將多國語言文字編碼，使得文字資料可在國際間交換，並創立世界級軟體的基礎。

Unicode 提供以下兩種編碼方法。預設的編碼方法是 UTF-16 (16 位元的編碼格式)。UCS-2 是 UTF-16 的子集，它使用兩個位元組來表示一個字元。UCS-2 通常被認可為全球性的字碼頁，它能夠表示目前所有單位元組字碼頁及雙位元組字碼頁的全部必需字元。UCS-2 在 IBM 中註冊為字碼頁 1200。

另一個 Unicode 編碼格式是 UTF-8，它是位元組導向的，且是被設計來方便現有 ASCII 系統的使用。UTF-8 使用不定數目的位元組 (通常為 1-3，有時是 4) 來儲存每個字元。無變化的 ASCII 字元就儲存為單位元組。所有其它字元就使用多個位元組來儲存。通常 UTF-8 資料可依據非設計為多位元組字碼頁的字碼來視為擴充的 ASCII 資料。UTF-8 在 IBM 中註冊為字碼頁 1208。

應用程式要考慮到資料在區域字碼頁、UCS-2 和 UTF-8 之間做轉換時的需求，這是很重要的。例如，20 個字元在 UCS-2 中需要完整的 40 個位元組，UTF-8 則介於 20 到 60 個位元組之間 (視原先使用的字碼頁及文字而定)。

DB2 Unicode 資料庫及應用程式

使用 UTF-8 字碼集建立的 Unix、Windows 或 OS/2 DB2 Universal 資料庫可用來儲存 UCS-2 及 UTF-8 兩種格式的資料。這樣的資料庫可被視為 Unicode 資料庫。SQL CHAR 資料是使用 UTF-8 來編碼的，而 SQL GRAPHIC 資料是使用 UCS-2 來編碼的。這可以相當於在 CHAR 直欄儲存單位元組 (SBCS) 及多位元組 (MBCS) 字碼集，以及在 GRAPHIC 直欄儲存雙位元組 (DBCS) 字碼集。

應用程式的字碼頁可能和 DB2 用來儲存資料的字碼頁不一樣。在非 Unicode 資料庫中，在字碼頁不一樣時，資料庫管理程式會轉換從屬站及伺服器之間傳送的字元及圖形 (pure DBCS) 資料。在 Unicode 資料庫中，從屬站字碼頁與 UTF-8 之間的字元資料轉換會自動由資料庫管理程式來執行，但是所有的圖形 (UCS-2) 資料不經過轉換就在從屬站及伺服器之間傳送。

資料庫管理程式執行的字碼頁轉換

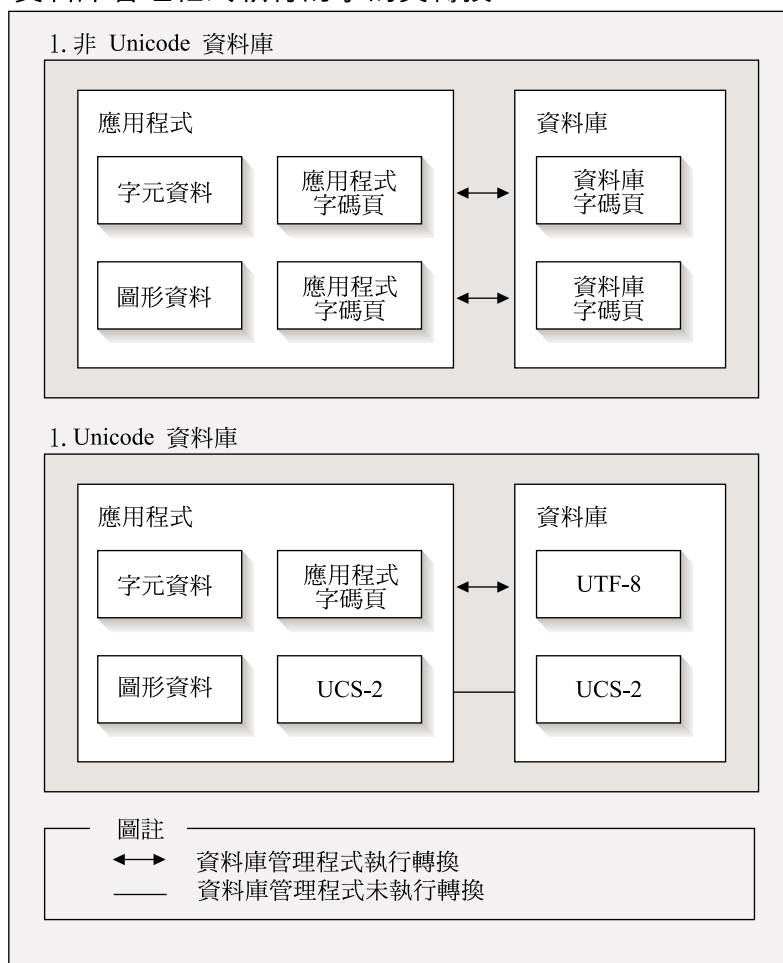


圖 7. 資料庫管理程式所執行的字碼頁轉換

註:

1. 當連接到 Unicode 資料庫，如果應用程式設定 `DB2CODEPAGE=1208`，則區域字碼頁是 UTF-8，所以不需要做字碼頁轉換。
2. 當已連接到 Unicode 資料庫，CLI 應用程式也可將字元資料當作圖形資料接收，以及將圖形資料當作字元資料。

應用程式是可以指定 UTF-8 字碼頁，以指出它將用 UCS-2 來傳送及接收圖形資料，並用 UTF-8 來傳送及接收字元資料。此應用程式字碼頁只支援 Unicode 資料庫。

使用 Unicode 時要考慮的其它項目為：

1. 資料庫字碼頁是在建立資料庫的時候決定的，且它的值預設會從作業系統語言環境 (或字碼頁) 來決定。CODESET 及 TERRITORY 關鍵字可用來明確地建立 Unicode DB2 資料庫。例如：

```
CREATE DATABASE unidb USING CODESET UTF-8 TERRITORY US
```
2. 應用程式字碼頁也預設為區域字碼頁，但是這可用下列方法由 UTF-8 來置換：
 - 使用此指令將應用程式字碼頁設定為 UTF-8 (1208)：

```
db2set DB2CODEPAGE=1208
```

- 對於 CLI/ODBC 應用程式，可呼叫 `SQLSetConnectAttr()` 及設定 `SQL_ATTR_ANSI_APP` 為 `SQL_AA_FALSE`。預設設定是 `SQL_AA_TRUE`。
3. GRAPHIC 直欄中的資料對於每個 Unicode 字元會使用完整的兩個位元組，而 CHAR 直欄中的資料對於每個 Unicode 字元會使用 1 到 3 個位元組。在 GRAPHIC 直欄字元方面的 SQL 限制通常是 CHAR 直欄字元的一半，但在位元組方面是一樣的。CHAR 直欄的最大字元長度是 254。GRAPHIC 直欄的最大字元長度是 127。若需詳細資訊，請參閱 *SQL Reference "Functions"* 章節中的 MAX。
 4. 圖形文字用 G 字首來與字元文字區別。例如：


```
SELECT * FROM mytable WHERE mychar = 'utf-8 data' AND mygraphic = G'ucs-2 data'
```

註：對於 Unicode 資料庫就不需要 G 字首。
請參閱第439頁的『Literals in Unicode Databases』，取得其餘資訊及更新的支援。
 5. 支援 CLI/ODBC 和 JDBC 應用程式不同於支援內含的應用程式。若需 CLI/ODBC 支援的相關資訊，請參閱第435頁的『CLI Guide and Reference(無中文版)』。
 6. UCS-2 資料的位元組次序可能因平台而異。DB2 在內部是使用 big-endian 格式。

文件更新

此文件更新下列有關在 DB2 版本 7.1 中使用 Unicode 的資訊：

- *SQL Reference*:
 - Chapter 3 Language Elements
 - Chapter 4 Functions
- *CLI Guide and Reference*:
 - Chapter 3. Using Advanced Features
 - Appendix C. DB2 CLI and ODBC
- *Data Movement Utilities Guide and Reference*, Appendix C. Export/Import/Load Utility File Formats

若需有關在 DB2 中使用 Unicode 的詳細資訊，請參照 *Administration Guide* 的 Appendix J. National Language Support (NLS)：“Unicode/UCS-2 and UTF-8 Support in DB2 UDB”。

SQL Reference(無中文版)

Chapter 3 Language Elements

Promotion of Data Types

In this section table 5 shows the precedence list for each data type. Please note:

1. For a Unicode database, the following are considered to be equivalent data types:
 - CHAR and GRAPHIC
 - VARCHAR and VARGRAPHIC
 - LONG VARCHAR and LONG VARGRAPHIC
 - CLOB and DBCLOB
2. In a Unicode database, it is possible to create functions where the only difference in the function signature is between equivalent CHAR and GRAPHIC data types, for

example, foo(CHAR(8)) and foo(GRAPHIC(8)). We strongly recommend that you do not define such duplicate functions since migration to a future release will require one of them to be dropped before the migration will proceed.

If such duplicate functions do exist, the choice of which one to invoke is determined by a two pass algorithm. The first pass attempts to find a match using the same algorithm as is used for resolving functions in a non-Unicode database. If no match is found, then a second pass will be done taking into account the following promotion precedence for CHAR and GRAPHIC strings:

GRAPHIC->CHAR->VARGRAPHIC->VARCHAR->LONG VARGRAPHIC->LONG VARCHAR->DBCLOB->CLOB

Casting Between Data Types

The following entry has been added to the list introduced as: "The following casts involving distinct types are supported":

- for a Unicode database, cast from a VARCHAR or VARGRAPHIC to distinct type DT with a source data type CHAR or GRAPHIC.

The following are updates to *Table 6. Supported Casts between Built-in Data Types*. Only the affected rows of the table are included.

表 31. Supported Casts between Built-in Data Types

Target Data Type →	C H A R	V A R C H A R	L O N G V A R C H A R	C L O B	G R A P H I C	V A R G R A P H I C	L O N G V A R G R A P H I C	D B C L O B
CHAR	Y	Y	Y	Y	Y ¹	Y ¹	-	-
VARCHAR	Y	Y	Y	Y	Y ¹	Y ¹	-	-
LONGVARCHAR	Y	Y	Y	Y	-	-	Y ¹	Y ¹
CLOB	Y	Y	Y	Y	-	-	-	Y ¹
GRAPHIC	Y ¹	Y ¹	-	-	Y	Y	Y	Y
VARGRAPHIC	Y ¹	Y ¹	-	-	Y	Y	Y	Y
LONGVARGRAPHIC	-	-	Y ¹	Y ¹	Y	Y	Y	Y
DBCLOB	-	-	-	Y ¹	Y	Y	Y	Y

¹ Cast is only supported for Unicode databases.

Assignments and Comparisons

Assignments and comparisons involving both character and graphic data are only supported when one of the strings is a literal. For function resolution, graphic literals and character literals will both match character and graphic function parameters.

The following are updates to *Table 7. Data Type Compatibility for Assignments and Comparisons*. Only the affected rows of the table, and the new footnote ⁶, are included:

Operands	Binary Integer	Decimal Number	Floating Point	Character String	Graphic String	Date	Time	Time-stamp	Binary String	UDT
Character String	No	No	No	Yes	Yes ⁶	¹	¹	¹	No ³	²
Graphic String	No	No	No	Yes ⁶	Yes	No	No	No	No	²

⁶ Only supported for Unicode databases.

String Assignments:

Storage Assignment

The last paragraph of this sub-section is modified as follows:

When a string is assigned to a fixed-length column and the length of the string is less than the length attribute of the target, the string is padded to the right with the necessary number of single-byte, double-byte, or UCS-2² blanks. The pad character is always a blank even for columns defined with the FOR BIT DATA attribute.

Retrieval Assignment

The third paragraph of this sub-section is modified as follows:

When a character string is assigned to a fixed-length variable and the length of the string is less than the length attribute of the target, the string is padded to the right with the necessary number of single-byte, double-byte, or UCS-2² blanks. The pad character is always a blank even for strings defined with the FOR BIT DATA attribute.

² UCS-2 defines several SPACE characters with different properties. For a Unicode database, the database manager always uses the ASCII SPACE at position x'0020' as UCS-2 blank. For an EUC database, the IDEOGRAPHIC SPACE at position x'3000' is used for padding GRAPHIC strings.

Conversion Rules for String Assignments

The following paragraph has been added to the end of this sub-section:

For Unicode databases, character strings can be assigned to a graphic column, and graphic strings can be assigned to a character column.

DBCS Considerations for Graphic String Assignments

The first paragraph of this sub-section has been modified as follows:

Graphic string assignments are processed in a manner analogous to that for character strings. For non-Unicode databases, graphic string data types are compatible only with other graphic string data types, and never with numeric, character string, or datetime data types. For Unicode databases, graphic string data types are compatible with character string data types.

String Comparisons:

Conversion Rules for Comparison

This sub-section has been modified as follows:

When two strings are compared, one of the strings is first converted, if necessary, to the encoding scheme and/or code page of the other string. For details, see the “Rules for String Conversions” section of Chapter 3 Language Elements in the *SQL Reference*.

Rules for Result Data Types

Character and Graphic Strings in a Unicode Database: This is a new sub-section inserted after the sub-section “Graphic Strings”.

In a Unicode database, character strings and graphic strings are compatible.

If one operand is...	And the other operand is...	The data type of the result is...
GRAPHIC(x)	CHAR(y) or GRAPHIC(y)	GRAPHIC(z) where $z = \max(x,y)$
VARGRAPHIC(x)	CHAR(y) or VARCHAR(y)	VARGRAPHIC(z) where $z = \max(x,y)$
VARCHAR(x)	GRAPHIC(y) or VARGRAPHIC(y)	VARGRAPHIC(z) where $z = \max(x,y)$
LONG VARGRAPHIC	CHAR(y) or VARCHAR(y)	LONG VARGRAPHIC or LONG VARCHAR
LONG VARCHAR	GRAPHIC(y) or VARGRAPHIC(y)	LONG VARGRAPHIC
DBCLOB(x)	CHAR(y) or VARCHAR(y)	DBCLOB(z) where $z = \max(x,y)$ or CLOB(y)
DBCLOB(x)	LONG VARCHAR	DBCLOB(z) where $z = \max(x,16350)$
CLOB(x)	GRAPHIC(y) or VARGRAPHIC(y)	DBCLOB(z) where $z = \max(x,y)$
CLOB(x)	LONG VARGRAPHIC	DBCLOB(z) where $z = \max(x,16350)$

Rules for String Conversions

The third point has been added to the following list in this section:

For each pair of code pages, the result is determined by the sequential application of the following rules:

- If the code pages are equal, the result is that code page.
- If either code page is BIT DATA (code page 0), the result code page is BIT DATA.
- In a Unicode database, if one code page denotes data in an encoding scheme different from the other code page, the result is UCS-2 over UTF-8 (that is, the graphic data type over the character data type).¹

- Otherwise, the result code page is determined by Table 8 of the “Rules for String Conversions” section of Chapter 3 Language Elements in the *SQL Reference*. An entry of 'first' in the table means the code page from the first operand is selected and an entry of 'second' means the code page from the second operand is selected.

¹ In a non-Unicode database, conversion between different encoding schemes is not supported.

Expressions

The following has been added:

In a Unicode database, an expression that accepts a character or graphic string will accept any string types for which conversion is supported.

With the Concatenation Operator: The following has been added to the end of this sub-section:

In a Unicode database, concatenation involving both character string operands and graphic string operands will first convert the character operands to graphic operands. Note that in a non-Unicode database, concatenation cannot involve both character and graphic operands.

Predicates

The following entry has been added to the list introduced by the sentence: “The following rules apply to all types of predicates”:

- In a Unicode database, all predicates that accept a character or graphic string will accept any string types for which conversion is supported.

Chapter 4 Functions

Scalar Functions

The following sentence has been added to the end of this section:

In a Unicode database, all scalar functions that accept a character or graphic string will accept any string types for which conversion is supported.

CLI Guide and Reference(無中文版)

Chapter 3. Using Advanced Features

The following is a new section for this chapter.

Writing a DB2 CLI Unicode Application

There are two main areas of support for DB2 CLI Unicode Applications:

1. The addition of a set of functions that can accept Unicode string arguments in place of ANSI string arguments.
2. The addition of new C and SQL data types to describe data as ANSI or Unicode data. The following sections provide more information for both of these areas. To be considered a Unicode application, the application must set the `SQL_ATTR_ANSI_APP` connection attribute to `SQL_AA_FALSE`, before a connection is made. This will ensure

that CLI will connect as a Unicode client, and all Unicode data will be sent in either UTF-8 for CHAR data or UCS-2 for GRAPHIC data.

Unicode Functions: The following is a list of the ODBC API functions that support both Unicode (W) and ANSI (A) versions (the function name will have a W for Unicode):

SQLBrowseConnect	SQLForeignKeys	SQLPrimaryKeys
SQLColAttribute	SQLGetConnectAttr	SQLProcedureColumns
SQLColAttributes	SQLGetConnectOption	SQLProcedures
SQLColumnPrivileges	SQLGetCursorName	SQLSetConnectAttr
SQLColumns	SQLGetDescField	SQLSetConnectOption
SQLConnect	SQLGetDescRec	SQLSetCursorName
SQLDataSources	SQLGetDiagField	SQLSetDescField
SQLDescribeCol	SQLGetDiagRec	SQLSetStmtAttr
SQLDriverConnect	SQLGetInfo	SQLSpecialColumns
SQLDrivers	SQLGetStmtAttr	SQLStatistics
SQLError	SQLNativeSQL	SQLTablePrivileges
SQLExecDirect	SQLPrepare	SQLTables

Unicode functions that always return, or take, string length arguments are passed as count-of-characters. For functions that return length information for server data, the display size and precision are described in number of characters. When the length (transfer size of the data) could refer to string or nonstring data, the length is described in octet lengths. For example, `SQLGetInfoW` will still take the length as count-of-bytes, but `SQLExecDirectW` will use count-of-characters. CLI will return result sets in either Unicode or ANSI, depending on the application's binding. If an application binds to `SQL_C_CHAR`, the driver will convert `SQL_WCHAR` data to `SQL_CHAR`. The driver manager maps `SQL_C_WCHAR` to `SQL_C_CHAR` for ANSI drivers but does no mapping for Unicode drivers.

New datatypes and valid conversions: There are two new CLI or ODBC defined data types, `SQL_C_WCHAR` and `SQL_WCHAR`. `SQL_C_WCHAR` indicates that the C buffer contains UCS-2 data. `SQL_WCHAR` indicates that a particular column or parameter marker contains Unicode data. For DB2 Unicode Servers, graphic columns will be described as `SQL_WCHAR`. Conversion will be allowed between `SQL_C_WCHAR` and `SQL_CHAR`, `SQL_VARCHAR`, `SQL_LONGVARCHAR` and `SQL_CLOB`, as well as between the graphic data types.

Table 32. Supported Data Conversions

SQL Data Type	SQL - CHAR	SQL - WCHAR	SQL - LONG	SQL - SHORT	SQL - TINYINT	SQL - FLOAT	SQL - DOUBLE	SQL - DATE	SQL - TIME	SQL - TIME STAMP	SQL - BINARY	SQL - BIT	SQL - DBCCHAR	SQL - CLOB	SQL - BLOB	SQL - DBCLOB	SQL - BIGINT	SQL - NUMERIC
BLOB	X	X									D				X			
CHAR	D	X	X	X	X	X	X	X	X	X	X	X					X	X
CLOB	D	X									X			X				
DATE	X	X						D		X								
DBCLOB		X									X		D			X		
DECIMAL	D	X	X	X	X	X	X				X	X					X	X
DOUBLE	X	X	X	X	X	X	D					X					X	X
FLOAT	X	X	X	X	X	X	D					X					X	X
GRAPHIC (Non-Unicode)	X	X											D					
GRAPHIC (Unicode)	X	X	X	X	X	X	X	X	X	X	X	X	D				X	
INTEGER	X	X	D	X	X	X	X					X					X	X
LONG VARCHAR	D	X									X							
LONG VARGRAPHIC (Non-Unicode)	X	X									X		D					
LONG VARGRAPHIC (Unicode)	X	X									X		D					
NUMERIC	D	X	X	X	X	X	X					X						X
REAL	X	X	X	X	X	D	X					X						X
SMALLINT	X	X	X	D	X	X	X					X					X	X
BIGINT	X	X	X	X	X	X	X				X	X					D	X
TIME	X	X							D	X								

Table 32. Supported Data Conversions (continue)

SQL Data Type	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	S Q L - C H A R	
TIMESTAMP	X	X						X	X	D								
VARCHAR	D	X	X	X	X	X	X	X	X	X	X	X					X	X
VARGRAPHIC (Non-Unicode)	X	X											D					
VARGRAPHIC (Unicode)	X	X	X	X	X	X	X	X	X	X	X	X	D				X	

Note:

- D** Conversion is supported. This is the default conversion for the SQL data type.
- X** All IBM DBMSs support the conversion.
- blank** No IBM DBMS supports the conversion.
 - Data is not converted to LOB Locator types, rather locators represent a data value, refer to Using Large Objects for more information.
 - SQL_C_NUMERIC is only available on 32-bit Windows operating systems.

Obsolete Keyword/Patch Value: Before Unicode applications were supported, applications that were written to work with single byte character data could be made to work with double byte graphic data by a series of cli ini file keywords, such as GRAPHIC=1,2 or 3, Patch2=7 etc. These workarounds presented graphic data as character data, and also affected the reported length of the data.

These keywords are no longer required for Unicode applications, and in fact should not be used otherwise there could be serious side effects. If it is not known if a particular application is a Unicode application, we suggest you try without any of the keywords that affect the handling of graphic data.

Literals in Unicode Databases: In Non-Unicode databases, data in LONG VARCHAR and LONG VARCHAR columns cannot be compared. Data in GRAPHIC/VARGRAPHIC and CHAR/VARCHAR columns can only be compared, or assigned to each other, using explicit cast functions since no implicit code page conversion is supported. This includes GRAPHIC/VARGRAPHIC and CHAR/VARCHAR literals where a GRAPHIC/VARGRAPHIC literal is differentiated from a CHAR/VARCHAR literal by a G prefix.

For Unicode databases, casting between GRAPHIC/VARGRAPHIC and CHAR/VARCHAR literals is not required. Also, a G prefix is not required in front of a GRAPHIC/VARGRAPHIC literal. Provided at least one of the arguments is a literal, implicit conversions occur. This allows literals with or without the G prefix to be used within statements that use either SQLPrepareW() or SQLExecDirect(). Literals for LONG VARCHARs still must have a G prefix.

For more information, please see "Casting Between Data Types" in Chapter 3 Language Elements of the *SQL Reference*.

New CLI Configuration Keywords: The following three keywords have been added to avoid any extra overhead when Unicode applications connect to a database.

1. **DisableUnicode**

Keyword Description:

Disables the underlying support for Unicode.

db2cli.ini Keyword Syntax:

DisableUnicode = 0 | 1

Default Setting:

0 (false)

DB2 CLI/ODBC Settings Tab:

This keyword cannot be set using the CLI/ODBC Settings notebook. The db2cli.ini file must be modified directly to make use of this keyword.

Usage Notes:

With Unicode support enabled, and when called by a Unicode application, CLI will attempt to connect to the database using the best client codepage possible to ensure there is no unnecessary data loss due to codepage conversion. This may increase the connection time as codepages are exchanged, or may cause codepage conversions on the client that did not occur before this support was added.

Setting this keyword to True will cause all Unicode data to be converted to the application's local codepage first, before the data is sent to the server. This can cause data loss for any data that cannot be represented in the local codepage.

2. **ConnectCodepage**

Keyword Description:

Specifies a specific codepage to use when connecting to the data source to avoid extra connection overhead.

db2cli.ini Keyword Syntax:

ConnectCodepage = 0 | 1 <any valid db2 codepage>

Default Setting:

0

DB2 CLI/ODBC Settings Tab:

This keyword cannot be set using the CLI/ODBC Settings notebook. The db2cli.ini file must be modified directly to make use of this keyword.

Usage Notes:

Non-Unicode applications always connect to the database using the application's local codepage, or the DB2Codepage environment setting. By default, CLI will ensure that Unicode applications will connect to Unicode databases using UTF-8 and UCS-2 codepages, and will connect to non-Unicode databases using the database's codepage. This ensures there is no unnecessary data loss due to codepage conversion.

This keyword allows the user to specify the database's codepage when connecting to a non-Unicode database in order to avoid any extra overhead on the connection.

Specify a value of 1 to cause SQLDriverConnect() to return the correct value in the output connection string, so the value can be used on future SQLDriverConnect() calls.

3. UnicodeServer**Keyword Description:**

Indicates that the data source is a Unicode Server. Equivalent to setting ConnectCodepage=1208.

db2cli.ini Keyword Syntax:

UnicodeServer = 0 | 1

Default Setting:

0

DB2 CLI/ODBC Settings Tab:

This keyword cannot be set using the CLI/ODBC Settings notebook. The db2cli.ini file must be modified directly to make use of this keyword.

Usage Notes:

This keyword is equivalent to ConnectCodepage=1208, and is added only for convenience. Set this keyword to avoid extra connect overhead when connecting to DB2 for OS/390 Version 7 or higher. There is no need to set this keyword for DB2 for Windows, DB2 for Unix or DB2 for OS/2 databases, since there is no extra processing required.

Appendix C. DB2 CLI and ODBC

The following is a new section added to this appendix.

ODBC Unicode Applications

A Unicode ODBC application sends and retrieves character data primarily in UCS-2. It does this by calling Unicode versions of the ODBC functions ('W' suffix) and by indicating Unicode data types. The application does not explicitly specify a local code page. The application can still call the ANSI functions and pass local code page strings.

For example, the application may call `SQLConnectW()` and pass the DSN, User ID and Password as Unicode arguments. It may then call `SQLExecDirectW()` and pass in a Unicode SQL statement string, and then bind a combination of ANSI local code page buffers (`SQL_C_CHAR`) and Unicode buffers (`SQL_C_WCHAR`). The database data types may be local code page or UCS-2 and UTF-8.

If a CLI application calls `SQLConnectW` or calls `SQLSetConnectAttr` with `SQL_ATTR_ANSI_APP` set to `SQL_AA_FALSE`, the application is considered a Unicode application. This means all CHAR data is sent and received from the database in UTF-8 format. The application can then fetch CHAR data into `SQL_C_CHAR` buffers in local code page (with possible data loss), or into `SQL_C_WCHAR` buffers in UCS-2 without any data loss.

If the application does not do either of the two calls above, CHAR data is converted to the applications local codepage at the server. This means CHAR data fetched into `SQL_C_WCHAR` may suffer data loss.

If the `DB2CODEPAGE` instance variable is set (using `db2set`) to code page 1208 (UTF-8), the application will receive all CHAR data in UTF-8 since this is now the local code page. The application must also ensure that all CHAR input data is also in UTF-8. ODBC also assumes that all `SQL_C_WCHAR` data is in the native endian format. CLI will perform any required byte-reversal for `SQL_C_WCHAR`.

ODBC Unicode Versus Non-Unicode Applications: This release of DB2 Universal Database contains the `SQLConnectW()` API. A Unicode driver must export `SQLConnectW` in order to be recognized as a Unicode driver by the driver manager. It is important to note that many ODBC applications (such as Microsoft Access and Visual Basic) call `SQLConnectW()`. In previous releases of DB2 Universal Database, DB2 CLI has not supported this API, and thus was not recognized as a Unicode driver by the ODBC driver manager. This caused the ODBC driver manager to convert all Unicode data to the application's local code page. With the added support of the `SQLConnectW()` function, these applications will now connect as Unicode applications and DB2 CLI will take care of all required data conversion.

DB2 CLI now accepts Unicode APIs (with a suffix of "W") and regular ANSI APIs. ODBC defines a set of functions with a suffix of "A", but the driver manager does not pass ANSI functions with the "A" suffix to the driver. Instead, it converts these functions to ANSI function calls without the suffix, and then passes them to the driver.

An ODBC application that calls the `SQLConnectW()` API is considered a Unicode application. Since the ODBC driver manager will always call the `SQLConnectW()` API regardless of what version the application called, ODBC introduced the `SQL_ATTR_ANSI_APP` connect attribute to notify the driver if the application should be considered an ANSI or UNICODE application. If `SQL_ATTR_ANSI_APP` is not set to `SQL_AA_FALSE`, DB2 CLI converts all Unicode data to the local code page before sending it to the server.

Appendix C. Export/Import/Load Utility File Formats

The following update has been added to this Appendix:

The export, import, and load utilities are not supported when they are used with a Unicode client connected to a non-Unicode database. Unicode client files are only supported when the Unicode client is connected to a Unicode database.

第7篇 連接到主電腦系統

連接環境補充資料	445
在 VM 環境下安裝應用程式伺服器	445
CLI/ODBC/JDBC 架構 PATCH1 和 PATCH2 設定 值	445

連接環境補充資料

在 VM 環境下安裝應用程式伺服器

新增下列句子到「提供區段資訊」章節，「定義應用程式伺服器」次章節的第一句子之後：

RDB_NAME 由 SQLSTART EXEC 提供，可作為 DBNAME 參數。

CLI/ODBC/JDBC 架構 PATCH1 和 PATCH2 設定值

CLI/ODBC/JDBC 驅動程式的架構可以由從屬站架構輔助程式、ODBC 驅動程式管理程式（若系統有安裝此程式）、編輯 db2cli.ini 檔來完成。詳細資料，請參閱安裝與架構補充或 *CLI Guide and Reference*。

您可以透過 db2cli.ini 檔，或者透過 SQLDriverConnect() 或 SQLBrowseConnect() CLI API，指定 PATCH1 及 PATCH2 關鍵字值，來修改 DB2 CLI/ODBC 驅動程式的預設行為。

您可以將使用者想要設定的所有關鍵字加在一起，來指定 PATCH1 關鍵字。例如，如果指定修補程式 1、2 及 8，則 PATCH1 的值將是 11。下列是驅動程式上每一關鍵字值及其效果的說明：

- 1 - 這使得驅動程式將搜尋 "count(exp)"，並以 "count(distinct exp)" 置換它。因為有些版本的 DB2 支援 "count(exp)" 語法，且該語法是由某些 ODBC 應用程式所建立的，所以需要這種效果。當伺服器不支援 "count(exp)" 語法時，Microsoft 應用程式需要它。
- 2 - 部份 ODBC 應用程式會被設陷，當 SQLGetTypeInfo() 函數中對於 LITERAL_PREFIX 或 LITERAL_SUFFIX 直欄傳回 SQL_NULL_DATA 時。這會強制驅動程式改而傳回一個空字串。Impromptu 2.0 會需要用到。
- 4 - 這會強制驅動程式將輸入時間戳記資料視為日期資料，如果時間和時間戳記的小數部份是零。Microsoft Access 會需要用到。
- 8 - 如果時間戳記的日期部份為 1899-12-30，這會強制驅動程式將輸入時間戳記資料視為時間資料。Microsoft Access 需要這種效果。
- 16 - 不使用。

- 32 - 這會強制驅動程式不傳回關於 SQL_LONGVARCHAR、SQL_LONGVARBINARY 及 SQL_LONGVARGRAPHIC 直欄的資訊。在應用程式看來，就好像並不支援長欄位。Lotus 123 會需要用到。
- 64 - 這會強制驅動程式以 NULL 值終止圖形輸出字串。雙位元組環境中的 Microsoft Access 需要這種效果。
- 128 - 這會強制驅動程式讓 "SELECT Config, nValue FROM MSysConf" 查詢到伺服器去。目前驅動程式會傳回錯誤，相關 SQLSTATE 值是 S0002（找不到表格）。如果使用者已在資料庫中建立了此架構表格且希望應用程式可以存取時會需要用到。
- 256 - 這會強制驅動程式在 SQLStatistics() 呼叫中先傳回主要鍵直欄。目前，驅動程式會依照索引名稱傳回已排序的索引，那是標準的 ODBC 行為。
- 512 - 這會強制驅動程式在 SQLGetFunctions() 中對 SQL_API_SQLTABLEPRIVILEGES 及 SQL_API_SQLCOLUMNPRIVILEGES 兩者傳回 FALSE。
- 1024 - 這會強制驅動程式在 SQLExecute() 或 SQLExecDirect() 中傳回 SQL_SUCCESS，而非 SQL_NO_DATA_FOUND，如果執行的 UPDATE 或 DELETE 陳述式沒有對任何橫列發生作用。Visual Basic 應用程式會需要用到。
- 2048 - 不使用。

- 4096 - 在自動 COMMIT 模式中時，於關閉游標後，這會強制驅動程式不發出 COMMIT。
- 8192 - 這會強制驅動程式在呼叫儲存程序後傳回額外的結果集。這個結果集是一個單一橫列的結果集，由儲存程序的輸出值組成。可被 Powerbuild 應用程式來存取。
- 32768 - 這會強制驅動程式使 Microsoft Query 應用程式能使用

- DB2 MVS 同義字。
- 65536 - 這會強制驅動程式自行插入一個 "G" 到實際上為圖形文字的字元文字前面。這個修補程式在雙位元組環境中工作時一定要提供。
 - 131072 - 這會強制驅動程式將時間戳記直欄視為一個 CHAR(26) 直欄，在它是一個唯一索引的一部份時。Microsoft 應用程式會需要用到。
 - 262144 - 這會強制驅動程式使用 pseudo-catalog 表格 db2cli.procedures，而非 SYSCAT.PROCEDURES 及 SYSCAT.PROCPARMS 表格。
 - 524288 - 這會強制驅動程式使用 SYSTEM TABLE SCHEMA，而非 TABLE_SCHEMA，在對 DB2/400 V3.x 系統作系統查詢時。這樣可提昇效能。
 - 1048576 - 這會強制驅動程式透過 SQLPutData() 將零長度的字串視為 SQL_NULL_DATA。

PATCH2 關鍵字不同於 PATCH1 關鍵字。在這種情況中，將使用逗點分隔字元來指定多個修補程式。例如，如果指定了修補程式 1、4 及 5，則 PATCH2 將具有值 "1,4,5"。下列是驅動程式上每一關鍵字值及其效果的說明：

- 1 - 這會強制驅動程式將 CALL 陳述式中的儲存程序名稱轉換為大寫字體。
- 2 - 不使用。
- 3 - 這會強制驅動程式將綱目呼叫的所有引數轉換為大寫字體。
- 4 - 這會強制驅動程式對於綱目呼叫 (也就是 SQLColumns()、SQLProcedureColumns() 等等) 傳回類似版本 2.1.2 的結果集，而非類似版本 5 的結果集。
- 5 - 這會強制驅動程式不要最佳化輸入 VARCHAR 直欄的處理，指向資料的指標和指向長度的指標在記憶體中是連續的。
- 6 - 這會強制驅動程式傳回不支援可捲動游標的訊息。Visual Basic 程式會需要用到，如果 DB2 從屬站是版本 5，且伺服器是 DB2 UDB 版本 5。
- 7 - 這會強制驅動程式將所有 GRAPHIC 直欄資料類型對映到 CHAR 直欄資料類型。在雙位元組環境中需要這種效果。
- 8 - 這會強制驅動程式忽略綱目呼叫中的目錄搜尋引數。
- 9 - 游標提早關閉時不要 COMMIT
- 10 - 不使用
- 11 - 報告支援目錄名稱，(VB 儲存程序)
- 12 - 從綱目呼叫引數除去雙引號，(Visual Interdev)
- 13 - 請勿將 db2cli.ini 的關鍵字添加到輸出連接字串
- 14 - 忽略 SQLProcedures() 及 SQLProcedureColumns() 上的綱目名稱
- 15 - 使用句點代表字元輸出中的小數點符號
- 16 - 強迫傳回每一開啓的說明資訊
- 17 - 請勿在說明上傳回直欄名稱
- 18 - 嘗試以參數記號取代文字
- 19 - 目前，DB2 MVS V4.1 不支援 ODBC 語法，因為它容許在外部結合子句中的 ON 子句可有括弧。
開啓這個 PATCH2 會導致 IBM DB2 ODBC 驅動程式在外部結合子句在 ODBC ESC 序列中時將連字號除去。這個 PATCH2 只有在不利於 DB2 MVS 4.1 時才使用。
- 20 - 目前，MVS 上的 DB2 不支援參數記號為兩個運算元 (expression ? BETWEEN ?) 的 BETWEEN 述詞。開啓這個修補程式會導致 IBM ODBC 驅動程式將述詞覆寫為 (expression >= ? and expression <= ?)。
- 21 - 將儲存程序的所有僅 OUTPUT 參數設定為 SQL NULL DATA
- 22 - 這個 PATCH2 會導致 IBM ODBC 驅動程式報告不支援外部結合 (OUTER Join)。這適用於在使用外部結合時陳述式時會產生 SELECT DISTINCT col1 或 ORDER BY col1 的應用程式，其中 col1 的長度大於 254 個字元，且造成 DB2 UDB 傳回錯誤 (因為 DB2 UDB 在此用法上不支援大於 254 個位元組的直欄)
- 23 - 請勿最佳化透過 cbColDef=0 結合的參數之輸入
- 24 - 存取對映時間的實例為字元
- 25 - 存取小數直欄的實例 - 除去字元陳述尾端的零
- 26 - 不將 SQL 程式 464 傳回給應用程式 - 表示已傳回結果集
- 27 - 迫使 SQLTables 使用 TABLETYPE 關鍵字值，即使應用程式指定了一個有效值
- 28 - 說明實際的直欄為雙倍直欄
- 29 - 小數直欄的 ADO 實例 - 除去 x 值的前導零，其中 x 為 1 > x > -1 (某些 MDAC 版本需要此效果)
- 30 - 快取最佳化時，停止使用儲存程序
- 31 - 回報 SQLStatistics 呼叫別名的統計值
- 32 - 置換 SQL 程式 -727 原因碼 4 處理程序
- 33 - 在轉換成字元時，傳回時間戳記的 ISO 版本 (相對於 ODBC 版本)

- 34 - 以 CHAR 名稱來回報 CHAR FOR BIT DATA 直欄
- 35 - SQL_DESC_BASE_TABLE_NAMER 被要求下回報一個無效的表格名稱 - ADO 唯讀最佳化
- 36 - 保留
- 37 - 保留

第8篇 一般資訊

一般資訊	451	使用 SMIT 解除安裝 DLFM 元件可能會除去額外的檔案集	474
DB2 Universal Database Business Intelligence 快速導覽	451	在 Windows 2000 上使用 CLP 時發生錯誤	474
DB2 Everywhere 已改為 DB2 Everyplace	451	SQL1035N	474
滑鼠需求	451	SQL 輔助程式的加強功能	474
試圖從 DB2 Run-time Client 連結導致「找不到連結檔案」錯誤	452	Linux 上的 DB2 的 Gnome 和 KDE 桌面整合	474
Search 探索 (Search Discovery)	452	在 Windows 2000 Terminal Server 管理模式下執行 DB2	475
HP-UX 11 的記憶體視窗	452	備份與復置指令的線上說明	475
dlfm client_conf 失效時使用者之動作	453	"Warehouse Manager" 應為 "DB2 Warehouse Manager"	475
很少見的事件：複製常駐程式在 dlfm 停止時沒有停止	453		
解除安裝 DB2 DFS 從屬站啟動程式	453		
Windows NT 從屬站身分驗證	453		
在建立子程序時 AutoLoader 可能會當掉	454		
DATALINK 復置	454		
Windows NT (CS/NT) 的 IBM Communications 伺服器定義使用者 ID 和通行碼	454		
節點定義	454		
聯合系統的限制	455		
DataJoiner 限制	455		
Windows NT 的希伯來文資訊型錄管理程式	455		
Microsoft SNA Server 和 SNA 多站更新 (兩階段確定) 支援	456		
啟動 Windows 之後無法啟動 DB2 的 SNA SPM	456		
DB2 管理伺服器的語言環境設定	456		
捷徑無效	457		
Windows NT 及 Windows 2000 上的 DB2 服務帳戶需求	457		
在版本 6 中建立的 Query Patroller 使用者失去了 EXECUTE 專用權	457		
Query Patroller 限制	457		
需要確定所有會在資料倉儲中心(DWC)中使用的使用者定義程式	458		
資料倉儲中心命令行匯出的新選項	458		
Backup Services APIs (XBSA)	458		
OS/390 代理程式	459		
安裝概觀	459		
安裝明細	460		
設置額外的代理程式函數	462		
用觸發程式排程倉儲步驟 (XTClient)	464		
轉換程式	464		
存取 DB2 系列之外的資料庫	468		
執行 DB2 for OS/390 公用程式	469		
抄寫	470		
代理程式記載	471		
Windows NT 上的從屬站端快取	472		
Enterprise Edition UNIX CD-ROM 的試用產品	472		
DB2 Connect Enterprise Edition UNIX CD-ROM 的試用產品	473		
捨棄 Data Links Manager	473		

一般資訊

DB2 Universal Database Business Intelligence 快速導覽

「快速導覽」不適用於 DB2 for Linux 或 Linux/390。

以小的系統字型執行「快速導覽」，可獲得最佳效果。您可能需要調整 Web 瀏覽器的字型大小，以正確地在 OS/2 上檢視「快速導覽」。請參照 Web 瀏覽器的說明，以取得調整字型大小的資訊。若要正確地檢視「快速導覽」（僅適用於 SBCS），建議您使用 8 點的 Helv 字型。針對日本及韓國客戶，建議您使用 8 點的 Mincho 字型。設定字型的喜好設定時，請確定選取「喜好設定」視窗「字型」頁的「使用預設字型，置換文件指定的字型」選項。

在某些例子裡，可能會在第二個瀏覽器視窗後面啟動「快速導覽」。若要更正這個問題，請關閉「快速導覽」，並遵循第9頁的『啟動 Netscape 時產生的錯誤訊息』裡的步驟。

啟動「快速導覽」時，您可能會接收到類似下列兩行的 JavaScript 錯誤：

```
file:/C:/Program Files/SQLLIB/doc/html/db2qt/index4e.htm, 65 行：
```

```
視窗未定義。
```

這個 JavaScript 錯誤會阻止「快速導覽」啟動頁 index4e.htm，在啟動「快速導覽」後自動關閉。可藉由關閉顯示 index4e.htm 的瀏覽器視窗來關閉「快速導覽」啟動頁。

在「新特性介紹」區段，「資料管理」主題的下方提到在 7.1 版中支援「即時日誌保存支援」。事實上並不是這樣。其中也提到：

日誌檔的大小從 4GB 增加到 32GB。

句子應改為：

全部的現行日誌空間已從 4GB 增加到 32GB。

說明 DB2 Data Links Manager 的區段中有一個句子如下：

而且，可支援 Veritas XBSA 介面的使用，
以便使用 NetBackup 作備份及復置。

句子應改為：

而且，可支援 XBSA 介面來做檔案歸檔及回復。
儲存體管理程式支援的 XBSA 介面包含
Legato NetWorker 及 Veritas NetBackup。

DB2 Everywhere 已改為 DB2 Everyplace

DB2 Everywhere 的名稱現在已改為 DB2 Everyplace。

滑鼠需求

除了視窗之外，所有的平台都需要滑鼠來使用其中的工具。

試圖從 DB2 Run-time Client 連結導致「找不到連結檔案」錯誤

因為 DB2 Run-time Client 沒有整組的連結檔案，所以無法完成 GUI 工具的連結，只可以從 DB2 Administration Client 完成 GUI 工具的連結。

Search 探索 (Search Discovery)

Search 探索只支援廣播媒體。例如，Search 探索將無法透過 ATM 配接卡來作用。但是，此項限制並不適用於 Known 探索。

HP-UX 11 的記憶體視窗

記憶體視窗是供大型 HP 64 位元機器的使用者所用，因他們想要利用 32 位元應用程式超過 1.75GB 的共用記憶體。如果您是執行 64 位元版本的 DB2，就不需要記憶體視窗。記憶體視窗可使每一程序或程序群組各有一個 1 GB 的共用記憶體可用。這將容許案例具有自己的 1GB 共用記憶體，再加上 0.75GB 的廣域共用記憶體。如果使用者想要善用這個優點，他們可以執行多個案例，每一案例均在自己的視窗中執行。下列是使用記憶體視窗的先決條件：

- DB2 EE 環境
 - 修補程式：Extension Software 12/98 及 PHKL_17795。
 - 必須對案例設定 \$DB2INSTANCE 變數。
 - 您在記憶體視窗下執行的每一個 DB2 案例，在 /etc/services.window 檔案中都有登錄。例如：

```
db2instance1 50
db2instance2 60
```

備註：名稱及 ID 之間僅能有一個空格。

- 您必須使用 TCP/IP 迴路方法，執行任何您要在伺服器上執行且需要多個單一陳述式的 DB2 指令。這是因為當記憶體視窗完成第一個陳述式的程序時，shell 將終止執行。「DB2 服務程式」知道如何完成此項作業。
- 任何要在記憶體視窗下所執行的案例內執行 DB2 指令，必須以 db2win (位在 sqllib/bin 中) 作為字首。例如：

```
db2win db2start
db2win db2stop
```

- 記憶體視窗外執行的任何 DB2 指令 (但記憶體視窗正在執行時) 應該傳回 1042。例如：

```
db2win db2start <== OK
db2 connect to db <==SQL1042
db2stop <==SQL1042
db2win db2stop <== OK
```

- DB2 EEE 環境
 - 修補程式：Extension Software 12/98 及 PHKL_17795。
 - 必須對案例設定 \$DB2INSTANCE 變數。
 - DB2_ENABLE_MEM_WINDOWS 登錄變數必須設定為 TRUE。
 - 您在記憶體視窗下所執行的每一個案例的每一個邏輯節點，在 /etc/services.window 檔案中都有登錄。每一個登錄的第一欄將成為與埠號連接的案例名稱。例如：

```

=== $HOME/sqllib/db2nodes.cfg for db2instance1 ===
5 host1 0
7 host1 1
9 host2 0

=== $HOME/sqllib/db2nodes.cfg for db2instance2 ===
1 host1 0
2 host2 0
3 host2 1

=== /etc/services.window on host1 ===
db2instance10 50
db2instance11 55
db2instance20 60

=== /etc/services.window on host2 ===
db2instance10 30
db2instance20 32
db2instance21 34

```

- 您不能用 db2win 作 DB2 指令的開頭，它只適用在 EE 環境下。

dlfm client_conf 失效時使用者之動作

在 DLFM 從屬站中，如果 dlfm client_conf 因為某些理由而失敗，DB2 型錄中的 "stale" 項目或許是因素之一。解決方案為發出下列指令：

```

db2 uncatalog db <dbname>
db2 uncatalog node <node alias>
db2 terminate

```

然後，再嘗試 dlfm client_conf。

很少見的事件：複製常駐程式在 dlfm 停止時沒有停止

這種情況很少發生。當使用者發出 dlfm stop 指令或不正常關機時，dlfm_copyd (copy 常駐程式) 沒有停止。如果發生這種情況，在嘗試重新啟動 dlfm 之前，發出 dlfm shutdown 指令。

解除安裝 DB2 DFS 從屬站啟動程式

解除安裝 DB2 DFS 從屬站啟動程式之前，root 應該確定 DFS 檔案不在使用中，而且沒有使用者在 DFS 檔案空間開啓 Shell。以 root 身分發出下列指令：

```
stop.dfs dfs_cl
```

檢查 /... 不再被裝載：

```
mount | grep -i dfs
```

如果沒有這樣做，DB2 DFS 從屬站啟動程式解除安裝時，該機器需要重新啟動。

Windows NT 從屬站身分驗證

Windows NT 的環境下，新的 DB2 登錄變數 DB2DOMAINLIST 與現存的從屬站身分驗證機制相結合。Windows NT 伺服器的 DB2 可使用此變數定義一個或多個 Windows NT 網域。只有定義於此列示中的網域使用者的連線或連接要求才會被接受。

只有在單純的 Windows NT 網域環境下執行 DB2 伺服器從屬站版本 7 的情況下，才可使用此登錄變數。

有關此登錄變數的設定，請參考 *Administration Guide: Performance* 內的「DB2 Registry and Environment Variables」部份。

在建立子程序時 AutoLoader 可能會當掉

AIX 4.3.3 包含了一個修正程式，用以解決在建立子程序時可能導致 AutoLoader 中斷的 libc 問題。AutoLoader 是一個多重執行緒的程式。其中的一個執行緒會起始另一個程序。起始一個子程序，會造成母項的記憶體影像建立在子項中。

很可能會發生：由 libc.a 用來管理多重執行緒從相同程序中之資料堆配置記憶體的鎖定，被非起始的 (non-forking) 執行緒所保留。因為非起始的執行緒不會存在於子程序，這個鎖定將不會釋出給子項，因而造成有時母項會中斷。

DATALINK 復置

資料庫復置之後所進行的離線備份復置，無論有或沒有 ROLL-FORWARD，都不會牽扯到快速一致性的處理程序。此情況下，檔案鏈結控制項之下有 DATALINK 直欄的所有表格將被置於資料鏈結一致性擱置 (DRP) 狀態下。

Windows NT (CS/NT) 的 IBM Communications 伺服器定義使用者 ID 和通行碼

如果您使用 APPC 作為遠端 DB2 從屬站與 DB2 伺服器連接之通訊協定，同時，您把 CS/NT 當成 SNA 產品使用，請確定您已在 CS/NT 架構檔內正確的設定了下列關鍵字。通常您可以在 `x:\ibmcs\private` 目錄下找在該檔。

節點定義

TG_SECURITY_BEHAVIOR

此參數容許使用者決定，如果 TP 不是機密保護的架構，節點是如何處理出現在 ATTACH 內的安全資訊。

IGNORE_IF_NOT_DEFINED

此參數容許使用者決定是否讓機密保護參數出現在 ATTACH 內，同時，若 TP 不是機密保護的架構時可忽略它們。

若您使用 IGNORE_IF_NOT_DEFINED，則無須於 CS/NT 定義使用者 ID 和通行碼。

VERIFY_EVEN_IF_NOT_DEFINED

此參數容許使用者決定是否讓機密保護參數出現在 ATTACH 內，並驗證它們，即使 TP 未架構為機密保護。這是預設值。

若您使用 VERIFY_EVEN_IF_NOT_DEFINED，則必須於 CS/NT 內定義使用者 ID 和通行碼。

欲定義 CS/NT 使用者 ID 和通行碼，請執行下列步驟：

1. 開始 --> 程式集 --> **IBM Communications Server --> SNA Node Configuration**。即開啓「歡迎使用 Communications Server 架構視窗」。
2. 選擇您要修改的架構檔。按一下「下一步」。所選擇的架構實務範例視窗開啓。

3. 強調顯示 CPI-C, APPC 或 5250 模擬程式。按一下「完成」。即可開啓 Communications Server SNA 節點視窗。
4. 按一下 CPI-C 和 APPC 旁邊的 [+] 號。
5. 按一下 LU6.2 機密保護旁邊的 [+] 號。
6. 右邊按一下使用者通行碼，並選取**建立**。定義使用者 ID 通行碼的視窗開啓。
7. 填寫使用者 ID 和通行碼。按一下**確定**。按一下**完成**以接受變更。

聯合系統的限制

下列是適用於聯合系統的限制：

- 包含暱名的查詢不支援 Oracle 資料類型 NCHAR、NVARCHAR2、NCLOB 及 BFILE。
- 「控制中心」不支援「建立伺服器選項」、「變更伺服器選項」及「捨棄伺服器選項」指令。若要發出以上任一指令，必須使用命令行處理器 (CLP)。
- 針對包含暱名的查詢，DB2 UDB 不是永遠遵照 DFT_SQLMATHWARN 資料庫的架構選項。反之，DB2 UDB 直接從遠端資料來源傳回運算表示式錯誤或警告，而忽略 DFT_SQLMATHWARN 設定。
- CREATE SERVER OPTION 陳述式不容許 COLSEQ 伺服器選項設定為 'I' (針對具有不區分大小寫對照順序的資料來源)。
- 指定無效的選項時，將傳回 ALTER NICKNAME 陳述式。
- 針對 Oracle、Microsoft SQL Server 及 Sybase 資料來源，數值資料類型無法對映到 DB2 的 BIGINT 資料類型。依預設值，Oracle 的數字 (p,s) 資料類型 (10 <= p <= 18 而 s = 0) 會對映到 DB2 的 DECIMAL 資料類型。

DataJoiner 限制

聯合環境內發出的分散式要求會受限為唯讀作業。

Windows NT 的希伯來文資訊型錄管理程式

「資訊型錄管理程式」元件可適用於希伯來文，並且在 DB2 Warehouse Manager for Windows NT CD 上提供。

希伯來文的翻譯提供於 IL_ICM.ZIP 這個壓縮檔中，位於 DB2 Warehouse Manager for Windows NT CD 的 DB2\IL 目錄裡。

若要安裝希伯來文翻譯的「資訊型錄管理程式」，請先安裝英文版的 DB2 Warehouse Manager for Windows NT，以及 Windows NT 希伯來文版的所有先決條件。

安裝 DB2 Warehouse Manager for Windows NT 之後，從 DB2\IL 目錄將 IL_ICM.ZIP 檔案解壓縮到安裝 DB2 Warehouse Manager for Windows NT 的目錄上。請確定解壓縮程式已選擇正確的選項，以在壓縮檔案裡建立目錄結構。

檔案解壓縮後，全域環境變數 LC_ALL 必須從 En_US 變更為 Iw_IL。若要變更設定：

1. 開啓 Windows NT 控制台，並在**系統**圖示上按兩下。
2. 在**系統內容**視窗，按一下**環境**標籤，然後尋找**系統變數**區段中的 **LC_ALL** 變數。
3. 按一下變數，以顯示**數值編輯**方框裡的值。將值從 **En_US** 變更為 **Iw_IL**。

4. 按一下**設定**按鈕。
5. 關閉**系統內容**視窗及**控制台**。

現在已安裝希伯來文版的「資訊型錄管理程式」。

Microsoft SNA Server 和 SNA 多站更新 (兩階段確定) 支援

當 Microsoft SNA Server 是使用中的 SNA 產品時，主電腦和 AS/400 應用程式無法存取使用 SNA 兩階段確定的 DB2 UDB 伺服器。任何表示支援以上功能的 DB2 UDB 出版品都是不正確的。必須要有 IBM Communications Server for Windows NT 5.02 版或以上的版本。

註：使用 DB2 UDB for Windows 來存取主電腦及 AS/400 資料庫伺服器的應用程式可以使用 Microsoft SNA Server 版本 4 Service Pack 3 或以上版本來使用 SNA 兩階段確定。

啓動 Windows 之後無法啓動 DB2 的 SNA SPM

若您使用 Microsoft SNA Server 版本 4 SP3 或以上的版本，請驗證 DB2 的 SNA SPM 可在重新啓動之後正常啓動。檢查 `\sqllib\<<案例名稱>\db2diag.log` 檔案，以取得類似下列所示的項目：

```
2000-04-20-13.18.19.958000 Instance:DB2 Node:000
PID:291(db2syscs.exe) TID:316 Appid:none
common_communication sqlccspmconnmgr APPC init Probe:19
SPM0453C 因為 Microsoft SNA Server 未啓動，所以同步點管理程式未啓動。
2000-04-20-13.18.23.033000 Instance:DB2 Node:000
PID:291(db2syscs.exe) TID:302 Appid:none
common_communication sqlccsna_start_listen Probe:14
DIA3001E 未成功啓動 "SNA SPM"。

2000-04-20-13.18.23.603000 Instance:DB2 Node:000
PID:291(db2syscs.exe) TID:316 Appid:none
common_communication sqlccspmconnmgr_listener Probe:6
DIA3103E APPC 通信協定支援時發生錯誤。APPC verb "APPC(DISPLAY 1
BYTE)". 主要傳回碼 "F004"。次要傳回碼 "00000000"。
```

若 db2diag.log 檔裡有這些項目，且時間戳記和您最近重新啓動的時間相符，您必須：

1. 呼叫 db2stop。
2. 啓動 SnaServer 服務程式 (若尚未啓動)。
3. 呼叫 db2start。

再次檢查 db2diag.log 檔，以確定沒有再添加其他項目。

DB2 管理伺服器的語言環境設定

請確定 DB2 管理伺服器案例的語言環境相容於 DB2 案例的語言環境。否則，DB2 案例無法與 DB2 管理伺服器溝通。

若 LANG 環境變數沒有在 DB2 管理伺服器的使用者設定檔裡設定，DB2 管理伺服器將以預設的系統語言環境啓動。若沒有定義預設的系統語言環境，DB2 管理伺服器將會以字碼頁 819 來啓動。若 DB2 案例使用某種 DBCS 語言環境，且 DB2 管理伺服器是以字碼頁 819 來啓動，則案例將無法與 DB2 管理伺服器溝通。DB2 管理伺服

器和 DB2 案例的語言環境必須相容。例如，在簡體中文 Linux 系統上，應該在 DB2 管理伺服器的使用者設定檔裡設定 "LANG=zh_CN"。

捷徑無效

在某些語言裡，對於 UNIX 系統和 OS/2 上的「控制中心」來說，有些鍵盤捷徑無效。請使用滑鼠來選取選項。

Windows NT 及 Windows 2000 上的 DB2 服務帳戶需求

在 DB2 for Windows NT 或 Windows 2000 的安裝期間，安裝程式會建立數個 Windows 服務，並指定每一個服務的服務帳戶。若要適當地執行 DB2，安裝程式會授與下列使用者權利給與 DB2 服務有關聯的服務帳戶：

- 作為作業系統的一部份
- 建立 token 物件
- 增加配額
- 以服務方式登入
- 更換程序層 token。

如果您要使用不同的服務帳戶供 DB2 服務使用，您必須將這些使用者權利授與服務帳戶。

除了這些使用者權利，服務帳戶對於安裝 DB2 產品的目錄也必須有寫入的權利。

DB2 Administration Server 服務 (DB2DAS00 服務) 的服務帳戶也必須擁有啟動及停止其它 DB2 服務 (也就是說，服務帳戶必須隸屬於 Power Users 群組) 的權限，以及對於任何它所管理的 DB2 案例擁有 DB2 SYSADM 權限。

在版本 6 中建立的 Query Patroller 使用者失去了 EXECUTE 專用權

因為部份新的儲存程序 (IWM.DQPGROUP、IWM.DQPVALUR、IWM.DQPCALCT 及 IWM.DQPINJOB) 新增到 Query Patroller 版本 7 中，Query Patroller 版本 6 中的現存使用者就於那些資料包不再保有 EXECUTE 專用權。可自動更正此問題的應用程式已新增到 FixPak 1 中了。

當您嘗試使用 DQP 查詢管理來修改 DQP 使用者資訊時，請不要試著從使用者清單中移除現有的使用者。

Query Patroller 限制

因為 JVM (Java Virtual Machine) 平台限制，在 HP-UX 及 NUMA-Q 上並不支援「查詢啟動程式」。除此之外，在 NUMA-Q 上並不支援「Query Patroller 追蹤器」。如果需要所有的 Query Patroller 從屬站工具，建議您使用不同的平台 (比如 Windows NT) 來對 HP-UX 或 NUMA-Q 伺服器執行這些工具。

需要確定所有會在資料倉儲中心(DWC)中使用的使用者定義程式

如果您要將「DB2 儲存程序建置器」所建立的儲存程序當作「資料倉儲中心(DWC)」中的使用者定義程式來使用，您必須插入下列陳述式到儲存程序 `con.close()`；陳述式前面：

```
con.commit();
```

如果沒有插入這個陳述式，儲存程序所做的變更會在儲存程序從 DWC 執行時被 Rollback。

對於 DWC 中所有的使用者定義程式，必須要明確地 COMMIT 任何內含的 DB2 函數，以使變更能在資料庫中生效；也就是說，您必須將 COMMIT 陳述式加到使用者定義程式中。

資料倉儲中心命令行匯出的新選項

命令行匯出到標示檔有了一個新的選項，`/B`。此選項無法透過「資料倉儲中心」介面來使用。

iwh2exp2 指令的新語法為：

```
iwh2exp2 filename.INP dbname userid password [PREFIX=table_schema] [/S] [/R] [/B]
```

其中

- filename.INP 是 INP 檔案的完整路徑名稱
- dbname 是資料倉儲中心控制資料庫的名稱
- userid 是用來登入資料庫的使用者 ID
- password 是用來登入資料庫的密碼
- 選用參數為：
 - PREFIX=table_schema：控制資料庫表格的表格綱目 (預設值為 IWH)
 - /S：匯出倉選取步驟的時程表
 - /R：不要匯出倉選取步驟的倉儲來源
 - /B：不要匯出倉選取步驟的 contributing 步驟

註：如果指定了 `/R` 或 `/B`，當匯入結果標示檔時，倉儲來源或 contributing 步驟必須已經存在，否則會傳回錯誤。

Backup Services APIs (XBSA)

Backup Services APIs (XBSA) 已被英國的 Open Group 定義為一個開放的應用程式程式設計介面，可用於需要資料儲存管理的應用程式或機能之間作備份或保存。這記載在 "Open Group Technical Standard System Management: Backup Services API (XBSA)"，Document Number C425 (ISBN: 1-85912-056-3)。

爲了支援這個介面，新建了兩個 DB2 登錄變數，目前支援 AIX、HP、Solaris 及 Windows NT：

DB2_VENDOR_INI

指向一個包含所有廠商環境設定的檔案。值會在資料庫管理程式啟動時獲得。

DB2_XBSA_LIBRARY

指向廠商提供的 XBSA 檔案庫。在 AIX 上，如果名稱不是 `shr.o`，那麼設定值必須包括共用物件。HP、Solaris 及 Windows NT 不要求共用物件名稱。例如，若要使用 Legato 的 NetWorker Business Suite Module for DB2，登記變數必須設定如下：

```
db2set DB2_XBSA_LIBRARY="/usr/lib/libxdb2.a(bsashr10.o)"
```

XBSA 介面可透過 BACKUP DATABASE 或 RESTORE DATABASE 指令來呼叫。例如：

```
db2 backup db sample use XBSA
db2 restore db sample use XBSA
```

OS/390 代理程式

這個文件中有什麼呢？安裝 OS/390 與其特性

本文提供如何安裝 OS/390 代理程式的指示及其特性的相關資訊。關於安裝程序的快速複查，請參閱「安裝概觀」，關於詳細的指示，請參閱「安裝細節」。關於代理程式特性的相關資訊，請參閱「設置額外的代理程式功能」、「轉換程式」及「存取 DB2 系列以外的資料庫」。

概觀

DB2「倉儲中心」包括 OS/390 代理程式。您可以在 DB2 Universal Database for OS/390 及其它資料庫之間使用代理程式來通信，包括其它平台上的 DB2 資料庫及非 DB2 資料庫。代理程式可與使用 ODBC 連線的資料來源通信。代理程式在 OS/390 UNIX 系統服務上執行。它需要 OS/390 V2R6 或以上版本，並且與 DB2 for OS/390 版本 5、6 及 7 相容。

OS/390 代理程式支援下列作業：

- 從來源 DB2 資料庫複製資料到目標 DB2 資料庫
- 取樣表格或檔案的內容
- 執行使用者定義程式
- 透過 IBM 的 DataJoiner 程式存取非 DB2 資料庫
- 透過 Cross Access 的 Classic Connect 存取 VSAM 或 IMS 資料
- 執行 DB2 Universal Database for OS/390 公用程式
- 執行「IBM 資料傳導延遲器」的引用工作

安裝概觀

這些步驟概要說明安裝程序。在「安裝明細」一節中提供了有關這些步驟的更多資訊。

1. 從 DB2 Universal Database for OS/390 磁帶安裝 OS/390 代理程式。
2. 更新您設定檔中的環境變數。
3. 設定連線：
 - 核心及代理常駐程式之間。
 - 代理程式及其將存取的資料庫之間。
4. 本端連結 CLI，連結到任何遠端資料庫。
5. 設置您的 ODBC 起始設定檔案。
6. 設置授權，則使用者：
 - 可執行代理常駐程式。
 - 對規劃 DSNACLI 有執行權限。

- 對記載及 ODBC 追蹤目錄有讀寫的授權(如果需要)。
7. 啟動代理常駐程式。

安裝明細

安裝 OS/390 代理程式

OS/390 代理程式附隨於 DB2 Universal Database for OS/390 版本 7 磁帶中。請參閱磁帶所附的程式目錄，以取得安裝 OS/390 代理程式的詳細資訊。

在安裝 OS/390 代理程式之前，您必須先在 DB2 子系統上引用 APAR PQ36585 或 PQ36586。

更新您設定檔中的環境變數

變數會將代理程式指向多種 DB2 檔案庫、輸出目錄等等。

下列範例是範例 .profile 檔案的內容。 .profile 檔案會定義環境變數，且它必須在啟動代理常駐程式的使用者的起始目錄中：

```
export VWS_LOGGING=/usr/lpp/DWC/logs
export VWP_LOG=/usr/lpp/DWC/vwp.log
export VWS_TEMPLATES=/usr/lpp/DWC/
export DSNAOINI=/usr/lpp/DWC/dsnaoini
export LIBPATH=/usr/lpp/DWC/:$LIBPATH
export PATH=/usr/lpp/DWC/:$PATH
export STEPLIB=DSN710.SDSNEXIT:DSN710.SDSNLOAD
```

設置連線

若要設定核心及常駐程式連線，請將下面這行加入 /etc/services 檔案或 TCPIP.ETC.SERVICES 檔案：

```
vwkernel 11000/tcp
vwd 11001/tcp
vwlogger 11002/tcp
```

若要在 OS/390 代理程式及資料庫間設定連線，將任何遠端資料庫新增到 OS/390 通信資料庫 (CDB)。這裡是部份範例 CDB INSERT：

```
INSERT INTO SYSIBM.LOCATIONS
(LOCATION, LINKNAME, PORT)
VALUES
('NTDB', 'VWNT704', '60002');

INSERT INTO SYSIBM.IPNAMES
(LINKNAME, SECURITY_OUT, USERNAMES, IPADDR)
VALUES
('VWNT704', 'P', 'O', 'VWNT704.STL.IBM.COM');

INSERT INTO SYSIBM.USERNAMES
(TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
VALUES
('O', 'MVSUID', 'VWNT704', 'NTUID', 'NTPW');
```

若需設置連線的詳細資訊，請參閱 *DB2 UDB for OS/390 Installation Guide* (GC26-9008-00) 中的「Connecting Distributed Database Systems」章節。

連結 CLI

因爲 OS/390 代理程式使用 CLI 來與 DB2 溝通，您必須將 CLI 規劃連結到所有您的代理程式要存取的遠端資料庫。這裡是一些本端 DB2 for OS/390 資料庫的範例連結資料包陳述式：

```
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLICS) ISO(CS)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLINC) ISO(NC)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIRR) ISO(RR)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIRS) ISO(RS)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIUR) ISO(UR)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIMS)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIC1)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIC2)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIF4)
```

這裡是一些執行在 Windows NT 上的 DB2 資料庫的範例 bind package 陳述式：

```
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLICS) ISO(CS)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLINC) ISO(NC)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIRR) ISO(RR)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIRS) ISO(RS)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIUR) ISO(UR)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIC1)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIC2)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIQR)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIF4)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIV1)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIV2)
```

這裡是一個範例 bind 陳述式，將 CLI 資料包一起連結到規劃中：

```
BIND PLAN(DWC6CLI) PKLIST(*.DWC6CLI.* )
```

設置 ODBC 起始設定檔案

範例 ODBC 起始設定檔案 (inisamp) 內含在 usr/lpp/DWC/ 目錄中。您可以編輯此檔案以配合您的系統使用，或者建立您自己的檔案。若要確定檔案正確地運作，請驗證已適當地架構：

- DSNAOINI 環境變數必須指向起始設定檔案。
- 檔名必須使用 dsnaoini.location_name 的命名慣例。
- 檔案必須包括 CONNECTTYPE=2 及 MVSATTACHTYPE=CAF 參數。

若需連結 CLI 與 DSNAOINI 檔案的詳細資訊，請參閱 *DB2 UDB for OS/390 ODBC Guide and Reference(SC26-9005)*。

設置授權

OS/390 代理程式是一個常駐程式程序。您可以使用一般的 UNIX 安全性或 OS/390 UNIX 安全性來執行代理常駐程式。

因爲代理程式需要常駐程式權限，請定義這些代理程式可執行檔爲「RACF 程式控制」：

- libt1s4d.dll
- iwhcomnt.dll
- vwd

欲在 RACF 程式控制中定義可執行的程式，請切換至「資料倉儲中心」可執行檔所在的目錄，執行下列指令：

```
extattr +p libt1s4d.dll
extattr +p iwhcomnt.dll
extattr +p vwd
```

若要使用有 +p 參數的 extattr 指令，則對於 BPX.FILEATTR.PROGCTL FACILITY 類別您至少必須擁有讀取權限。下列範例顯示用來將此許可權給予使用者 ID SMORG 的 RACF 指令：

```
RDEFINE FACILITY BPX.FILEATTR.PROGCTL UACC(NONE)
PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY) ID(SMORG) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

關於授權的詳細資訊，請參閱 *OS/390 UNIX System Services Planning (SC28-1890)*。

啓動代理常駐程式

在您完成架構系統之後，請啓動代理常駐程式：

1. Telnet 到 OS/390 上的 UNIX Systems Services，透過 OS/390 host 及 USS 埠。
2. 啓動代理常駐程式：
 - 若要在背景啓動常駐程式，請在命令行中輸入 vwd。
 - 若要在背景啓動常駐程式，請輸入
vwd>/usr/lpp/DWC/logs/vwd.log 2>&1 &

若要驗證 OS/390 代理常駐程式是否在執行中，請在 UNIX Shell 命令行輸入下列指令：

```
ps -e | grep vwd
```

或者，在 OS/390 主控台輸入 D OMVS,a=all，並搜尋字串 vwd

設置額外的代理程式函數

DB2 Warehouse Manager 資料包包括下列使用者定義程式：

- vwpftp：執行 FTP 指令檔。
- vwpmvs：提出 JCL 工作串流。
- vwprcpy：使用 FTP 複製檔案。
- XTClient：從屬站觸發程式。
- etidlmvs：ETI (Evolutionary Technologies International) 提供的公用程式；可刪除 MVS 上的檔案。
- etircmvs：ETI 提供的公用程式；可在 MVS 主電腦上執行 FTP。
- etiexmvs：ETI 提供的公用程式；可在 MVS 上執行 JCL。

除此之外，您可以在「資料倉儲中心」建立使用者定義程式及儲存程序。OS/390 代理程式支援任何可在 UNIX 系統服務下執行的可執行檔。

一個使用者定義程式會指定給一個或多個步驟。當您執行使用者定義程式時，會發生下列動作：

- 代理程式啓動。
- 代理程式執行使用者定義程式。
- 使用者定義程式傳回回覆碼及回饋檔案給代理程式。
- 代理程式將結果傳回核心。

爲了在 OS/390 上執行 ETI，您必須先將 FixPack 2 套用到 DB2 Universal Database 版本 7.1。

使用 VWP_LOG 環境變數來定義一個目錄，供使用者定義的程式寫入輸出。

如果您使用使用者定義程式來用 FTP 提出工作，您必須先建立要提出的 JCL 及資料。JCL 中的工作名稱必須是 USERIDx，其中 x 是一個單一字元的字母或數字 (範例：MYUSERA)。包含在 JCL 中的 MSGCLASS 及 SYSOUT 檔案的輸出類別必須指定 JES-held 輸出類別。

限制：所提出工作的最大 LRECL 是 254 個字元。JES 只會掃描 JCL 的前 72 個字元。

變更「資料倉儲中心」FTP 支援的模板

「資料倉儲中心」安裝了一個 JCL 模板，供使用 FTP 轉送檔案。如果您計劃讓 OS/390 代理程式使用 FTP 指令 GET 或 PUT 從某 OS/390 主電腦傳送檔案到另一個遠端主電腦，您需要爲 OS/390 系統變更 JCL 模板中的帳戶資訊：

1. 以一個有權限在 /usr/lpp/DWC 目錄中複製及更新檔案的 ID 登入。
2. 尋找 ftp.jcl，並用新的檔名 systemname.ftp.jcl 複製檔案，其中 *systemname* 是 OS/390 系統的名稱。
3. 建立此檔案的副本給每一個您要在其上執行轉換程式 vwpmvs 或 ETI extract 的 OS/390 系統。例如，如果您要在 STLMVS1 上執行任一個程式，建立一個名稱爲 STLMVS1.ftp.jcl 的檔案副本。
4. 使用文字編輯程式來自訂 JCL，以符合您站台的需求。變更帳戶資訊以符合您的 MVS 系統的標準帳戶資訊。請勿變更任何方括弧 (I) 內的參數，比如 [USERID] 及 [FTPFILE]。(方括弧分別爲十六進位字元 x'AD' 及 x'BD'。若您的 TSO 終端機類型在 SPF 選項 0 中未設成 3278A，這些值可能會顯示成特殊字元，而不是方括弧。這並不會是問題，如果您沒有變更 x'AD' 或 x'BD'，或者任何字元之間的資料。)
5. 更新環境變數 VWS_TEMPLATES，指向複製的模板檔案的目錄。

「資料倉儲中心」包含此範例 JCL 模板：

```
//[USERID]A JOB , 'PUT/GET',  
// CLASS=A,  
// USER=&SYSUID,  
// NOTIFY=&SYSUID,  
// TIME=(,30),  
// MSGCLASS=H  
//STEP1 EXEC PGM=FTP,PARM='( EXIT'  
//INPUT DD DSN=[FTPFILE],DISP=SHR  
//OUTPUT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*
```

取樣表格或檔案的內容

您可使用 OS/390 代理程式來取樣 DB2 表格與純文字檔的內容，比如 UNIX 系統服務檔案及 OS/390 原生純文字檔。您也可使用 OS/390 代理程式以 Classic Connect 取樣 IMS 或 VSAM 檔案的內容。對於純文字檔，代理程式會察看檔案定義的內容中的參數來判定檔案格式。

用觸發程式排程倉儲步驟 (XTClient)

使用觸發程式可從 OS/390 平台排程倉儲步驟。您或是 OS/390 工作排程程式可以提出一個會觸發「資料倉儲中心」步驟的工作。如果步驟順利完成，JCL 中的觸發函式步驟會傳回回覆碼 0。

在 OS/390 UNIX 系統服務上必須安裝 JDK 1.1.8 或更新版本以使用觸發程式。若要啟動觸發函式，請先在執行倉儲伺服器的機器上啟動 XTServer。此程序可見於資料倉儲中心管理手冊的第 5 章，標題為「從資料倉儲中心之外啟動步驟」。XTServer 啟動之後，請在 OS/390 上啟動 XTClient。下列範例顯示啟動觸發函式的 JCL 範例。

```
//DBA1A JOB 1,'XTCLIENT',CLASS=A,MSGCLASS=H,
//      MSGLEVEL=(1,1),REGION=4M,NOTIFY=&SYSUID
//*****
//* submit iwhetrig
//*****
//BRADS EXEC PGM=BPXBATCH,
//      PARM=('sh cd /usr/lpp/DWC/; java XTClient 9.317.171.133 1100x
//      9 drummond pw bvmvs2nt 1 1 100')
//STDOUT DD PATH='/tmp/xtclient.stdout',
//      PATHOPTS=(OWRONLY,OCREAT),
//      PATHMODE=SIRWXU
//STDERR DD PATH='/tmp/xtclient.stderr',
//      PATHOPTS=(OWRONLY,OCREAT),
//      PATHMODE=SIRWXU
//
```

註：上述範例 JCL 碼顯示如何繼續參數到下一行。方法是鍵入參數，一直到第 71 個直欄，在第 72 個直欄放一個 'X'，然後在下一行的第 16 個直欄繼續。

參數 (cd /usr/lpp/DWC/;) 的第一個部分將目錄變更到安裝 OS/390 代理程式的位置。參數的第二部份啟動 XTClient 並傳遞下列 8 個參數：

- 「資料倉儲中心」伺服器主電腦名稱或 IP 位址
- 「資料倉儲中心」伺服器埠 (通常為 11009)
- 「資料倉儲中心」使用者 ID
- 「資料倉儲中心」通行碼
- 要執行的步驟名稱
- 「資料倉儲中心」伺服器指令，其中：
 - 1 = 移入步驟
 - 2 = 將步驟提昇到測試模式
 - 3 = 將步驟提昇到生產模式
 - 4 = 將步驟降級到測試模式
 - 5 = 將步驟降級到開發模式
- 是否等待步驟完成的選項，其中 1= 是，0 = 否
- 最大列數 (使用 0 或空白來提取所有橫列)

轉換程式

簡介

這 12 個轉換程式是 Java 儲存程序，提供一些基本的資料轉換。爲了執行這些轉換程式，您必須先在 DB2 子系統上設定 Java 儲存程序。關於這些轉換程式的其他資訊，請參閱 *IBM DB2 Universal Database Data Warehouse Center 管理手冊 版本 7*，SC40-0496-00。

```
IWH.CLEAN
IWH.PERIODTABLE
IWH.KEYTABLE
IWH.CHISQUARE
IWH.CORRELATION
IWH.STATISTICS
IWH.INVERTDATA
IWH.PIVOTDATA
IWH.REGRESSION
IWH.ANOVA
IWH.SUBTOTAL
IWH.MOVINGAVERAGE
```

設置「Java 儲存程序」

這些指令是完整指示的精簡版，說明如何設定 Java 儲存程序，請參閱 *Application Programming Guide and Reference for Java™*，SC26-9018

1. 在 DB2 子系統上引用 PTFs UQ46170 和 UQ46114。
2. 在 OS/390 系統上安裝 Visual Age for Java 2.0 或以上版本。
3. 在 DB2 上安裝 JDBC，將 JDBC 套件連結至 DB2 子系統。
4. 在 DB2 子系統上設定 RRS 和 DB2 WLM 儲存程序。
5. 設定您 DB2 的 Java 儲存程序。包括爲 Java 儲存程序位址空間建立一個 Java WLM 啓動程序。
6. 在 WLM 之下，您必須將 Java WLM 啓動程序結合一個 WLM 環境名稱。請使用 WLM 應用程式環境畫面「建立應用程式環境」來結合環境名稱和 JCL 程序。
7. 在 CREATE PROCEDURE 或 ALTER PROCEDURE 的 WLM_ENVIRONMENT 選項上指定 WLM 應用程式環境名稱，將儲存程序或使用者定義的函數結合一個應用程式環境。
8. 確定您的 DB2 啓動作業的擁有者具備存取權來存取 Java WLM 啓動程序中的檔案庫。

倉儲轉換程式的設置步驟

這些指令是完整指示的精簡版，請參閱 *IBM DB2 Universal Database Data Warehouse Center 管理手冊 版本 7*，SC40-0496-00

1. 將 Fixpack 3 引用至 DB2 Universal Database for NT 版本 7，或將「倉儲控制資料庫」更新成 TRANSREGISTERED = 1 和 TRANSFENCED=1。

欲更新「倉儲控制資料庫」，請在 DB2 Universal Database 命令行處理器上輸入下列 SQL：

```
CONNECT TO your_vw_control_database
UPDATE IWH.INFORESOURCE SET TRANSREGISTERED = '1' WHERE SUBDBTYPE = 'DB2 MVS'
UPDATE IWH.INFORESOURCE SET TRANSFENCED = '1' WHERE SUBDBTYPE = 'DB2 MVS'
```

2. 定義轉換程式到 DB2

- 若您有 DB2 for OS/390 版本 7，請使用 /usr/lpp/DWC/createXfSQLV7 中的 SQL 陳述式。

- 若您有 DB2 for OS/390 版本 6，請使用 /usr/lpp/DWC/createXfSQL 中的 SQL 陳述式。
- 若您有 DB2 for OS/390 版本 5，請使用 /usr/lpp/DWC/createXfSQL 中註解的 SQL 陳述式。移除所有 CREATE PROCEDURE 陳述式的註解。然後移除 INSERT INTO SYSIBM.SYSPROCEDURES 陳述式的註解，用來定義 DB2 for OS/390 版本 5 的轉換程式。

當您設定 Java 儲存程序時，您使用 WLM 來結合 Java WLM 啟動程序和一個 WLM 環境名稱。環境名稱指定於 CREATE PROCEDURE 陳述式的 WLM ENVIRONMENT 選項中。DSNWLMJ 是內含於上述轉換程式定義中的 WLM 環境名稱。您可新增 DSNWLMJ 的 WLM 關聯名稱，或將每一個轉換程式定義的 WLM ENVIRONMENT 選項變更為您已結合至啟動程序的名稱。

3. 從 UNIX 系統服務設定鏈結來連結 IWH710.SIWHLOAD 中的轉換程式載入模組。
 - 以 Telnet 連接至您 OS/390 主電腦系統的 UNIX 系統服務。
 - 切換至您安裝 OS/390 代理程式的目錄。預設安裝目錄是 /usr/lpp/DWC。
 - 若您使用 DB2 V7，請跳至步驟 4。若您使用 DB2 V5 或 V6，請在安裝目錄中編輯 trlinks 資料集。在此行的第一欄中加上一個 # 符號來註解此行：

```
ln -e IWHXFV7 xf.j11;
```

移除第一欄的 # 符號來移除此行的註解。

```
#ln -e IWHXF xf.j11;
```

儲存您的變更。

- 鍵入 trlinks 並按下輸入。這會在目錄中建立一個 xf.j11 鏈結，用來指示代理程式載入 IWHXF 或 IWHXFV7 模組。
4. 透過 APF 授權 IWH710.SIWHPDSE，然後新增至 DB2 Java 儲存程序啟動程序的 STEPLIB 連接。
 5. 將您的 xf.j11 鏈結所在的目錄 (預設值：/usr/lpp/DWC) 新增至 WLM 環境資料集中的 CLASSPATH 和 LIBPATH 環境變數。
 - 若您不確定 WLM 資料集的位置，請查看您的 DB2 Java 儲存程序啟動程序。您的 WLM 環境資料集是您的 JAVAENV DD 卡指向的資料集。
 6. 啟動儲存程序，然後建立並執行倉儲步驟。

Java 儲存程序的限制

一個儲存程序的標記中的 Java 物件僅於 DB2 for OS/390 版本 7 中才支援。因此，轉換程式在 DB2 for OS/390 版本 5 或 6 的參數中不支援 NULL 值。在這些版本中，若您傳遞 NULL 值，則會視為零來處理。版本 5 和 6 轉換程式將零參數視為空字串。

DB2 僅於 DB2 for OS/390 版本 7 中支援 COMMIT SQL 陳述式。INVERTDATA 儲存程序可在儲存程序內捨棄並重建表格；因此需要 commit 陳述式。基於此理由，DB2 for OS/390 版本 5 或版本 6 不支援 IWH.INVERTDATA。

DB2 for OS/390 不支援 Java 使用者定義的函數，所以 390 平台上不支援 IWH.FORMATDATE。

Java 儲存程序的範例啟動程序

(說明於「DB2 for OS/390 Application Programming Guide and Reference for Java」)：

```

//DSNWLMJ PROC DB2SSN=DSN,NUMTCB=5,APPLENV=DSNWLMJ <-- WLM ENVIRONMENT value in
CREATE PROC
//*****
//* THIS PROC IS USED TO START THE WLM-ESTABLISHED SPAS *
//* ADDRESS SPACE FOR THE DSNWLMJ APPLICATION ENVIRONMENT *
//* V WLM,APPLENV=DSNWLMJ,RESUME *
//*****
//DSNWLMJ EXEC PGM=DSNX9WLM,TIME=1440,REGION=0M,
// PARM='&DB2SSN, &NUMTCB, &APPLENV'
//STEPLIB DD DSN=DSN.TESTLIB,DISP=SHR
// DD DSN=IWH710.SIWHPDSE,DISP=SHR <--This has the transformers in it
// DD DSN=DSN.HPJSP.PDSE.JDBC,DISP=SHR <--HPJ DLLs from HPJ setup
// DD DSN=SYS1.PP.PDSELINK,DISP=SHR <--HPJ runtime libraries
// DD DSN=DSN710.SDSNEXIT,DISP=SHR
// DD DSN=DSN710.SDSNLOAD,DISP=SHR
// DD DSN=SYS1.SCEERUN,DISP=SHR
// DD DSN=DSN.PDSE,DISP=SHR <--HPJ setup info
//JAVAENV DD DSN=DSN.WLMENVJ.JSPENV,DISP=SHR <--Environment variables, see below
//CEEDUMP DD SYSOUT=A
//DSSPRINT DD SYSOUT=A
//JSPDEBUG DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSPRINT DD SYSOUT=A

```

範例環境變數資料集

(說明於「DB2 for OS/390 Application Programming Guide and Reference for Java」):

```

ENVAR("TZ=PST07",
"DB2SQLJPROPERTIES=/usr/lpp/db2/jdbc/db2710/classes/db2sqljjdbc.properties",
"LIBPATH=/usr/lpp/DWC",
"VWSPATH=/usr/lpp/DWC",
"CLASSPATH=/usr/lpp/db2/jdbc/db2710/classes:/usr/lpp/DWC:/usr/lpp/hpj/lib"),
MSGFILE(JSPDEBUG)

```

轉換程式的國家語言支援

OS/390 代理程式產生的大部份訊息皆傳送至 NT 平台來解譯，所以大多數情況下，訊息語言系根據 UDB for NT 如何安裝而定。轉換程式是例外。OS/390 代理程式提供下列訊息檔案供轉換程式使用：

File name:	For language:
Xf.properties_Fi_FI	Finnish in Finland
Xf.properties_No_NO	Norwegian in Norway
Xf.properties_Ru_RU	Russian in Russia
Xf.properties_Zh_CN	Chinese in China (People's Republic of China)
Xf.properties_Zh_TW	Chinese in Taiwan
Xf.properties_Da_DK	Danish in Denmark
Xf.properties_De_DE	German in Germany
Xf.properties_En_US	English in U.S.
Xf.properties_Es_ES	Spanish in Spain
Xf.properties_Fr_FR	French in France
Xf.properties_It_IT	Italian in Italy
Xf.properties_Ja_JP	Japanese in Japan
Xf.properties_Ko_KR	Korean in Korea
Xf.properties_Pt_BR	Portugese in Brazil
Xf.properties_Sv_SE	Swedish in Sweden

若您的轉換程式訊息不是英文，請選取上述其中一個檔案，將其內容複製到 Xf.properties。

存取 DB2 系列之外的資料庫

若要存取非 DB2 Universal Database 系列的資料庫，OS/390 代理程式使用 DataJoiner。DataJoiner 可讓代理程式使用通常的 DRDA 串流，彷彿它就是 UDB 資料庫。如果 ODBC 要求被導引到一個非 DB2 系列資料庫來源，DataJoiner 會呼叫附加層次的程式碼來存取外部的資料庫。

DataJoiner 可以存取 Oracle、Sybase、Informix、Microsoft SQL Server、Teradata 及任何其它執行在 Windows NT、AIX 或 Sun Solaris 作業環境上有 ODBC 驅動程式的資料庫。OS/390 代理程式可存取作為來源的 DataJoiner，但作為目標就不可以。DataJoiner 不支援二階段的 COMMIT。雖然 DataJoiner 在版本 2.1 及 2.1.1 中支援 TCP/IP 作為應用程式要求器，但它並沒有應用程式伺服器。因為 OS/390 代理程式需要應用程式伺服器來使用 TCP/IP，請改用 SNA 連線來從 OS/390 存取 DataJoiner。

在 OS/390 上存取 IMS 及 VSAM

Classic Connect 可個別從倉儲代理程式購買及安裝。OS/390 代理程式透過 Classic Connect ODBC 驅動程式來存取 IMS 及 VSAM。Classic Connect 可讓您設置 IMS 及 VSAM 資料集的 DB2-like 定義，然後使用 ODBC 來存取它們。

OS/390 代理程式會根據要求是否導引到 Classic Connect 或 DB2，來載入正確的 ODBC 驅動程式。如果您要存取 DB2 來源，代理程式會載入 DB2 ODBC 驅動程式。如果您正在存取 VSAM 或 IMS 來源，代理程式會載入 Classic Connect ODBC 驅動程式。然後就會處理代理程式要求。

設置 Classic Connect ODBC 驅動程式與倉儲存取

Classic Connect 可個別從 OS/390 代理程式購買及安裝。Classic Connect 可將單一檔案或一個檔案的一部份視為一或多個關聯式表格。您必須對映 IMS 和 VSAM 資料，才能讓 Classic Connect 存取。您可手動對映資料或使用 Microsoft Windows Classic Connect 非關聯式資料對映程式。

1. 在 OS/390 上安裝 Classic Connect Data Server。
2. 選用性：在 NT 上安裝 Classic Connect Data Mapper 產品。
3. 定義 Classic Connect 的邏輯表定義，讓 Classic Connect 可依關係來存取資料。您可使用資料對映程式來建立 IMS 和 VSAM 結構的定義，或手動建立定義。
4. 在您設定 Classic Connect 之後，您可設定對於倉儲的存取權限：
 - a. 建立一個 Classic Connect .ini 檔案。

範例 Classic Connect 應用程式架構檔 cxa.ini 在 /usr/lpp/DWC/ 目錄中，且它會在這裏複製給您：

```
* national language for messages
NL = US English
* resource master file
NL CAT = usr/lpp/DWC/v4r1m00/msg/engcat
FETCH BUFFER SIZE = 32000
DEFLOC = CXASAMP
USERID = uid
USERPASSWORD = pwd
DATASOURCE = DJX4DWC tcp/9.112.46.200/1035
MESSAGE POOL SIZE = 1000000
```

- b. 更新 .ini 檔中的 DATASOURCE 行。這一行包含一個資料來源名稱及一個通信協定位址。資料來源名稱必須與 Classic Connect 資料伺服器上定義的 Query

Processor 名稱相對應，位置在資料伺服器的配置檔中的 QUERY PROCESSOR SERVICE INFO ENTRY。通信協定位址可在同一個檔案中的 TCP/IP SERVICE INFO 登錄找到。這個檔案裡的 USERID 及 USERPASSWORD 會在定義倉儲資料來源時用到。

- c. 匯出 CXA_CONFIG 環境變數到 Classic Connect 程式檔，它們通常與 .ini 檔案位在相同的目錄。
- d. 更新 LIBPATH 環境變數，將路徑加到 Classic Connect 程式檔，通常和 .ini 檔案在相同的目錄中。
- e. 選用性：使用測試程式 cxasamp 來驗證安裝：在您的 .ini 檔案所在的目錄中輸入 cxasamp。location/uid/pwd 是定義於 .ini 檔案中的 data source name/userid/userpassword。
- f. 定義資料來源到倉儲，就如同您定義任何 DB2 資料來源。

您不需要去更新 dsnaoini 檔案，因為 DB2 for OS/390 沒有驅動程式管理員。Classic Connect 的驅動程式管理員內建在 OS/390 代理程式中。

執行 DB2 for OS/390 公用程式

您必須將 APAR PQ44904 引用至 OS/390 代理程式，以便使用代理程式來執行公用程式。

DSNUTILS 是一個在 WLM 與 RRS 環境中執行的 DB2 for OS/390 儲存程序。您可以用它來執行任何藉由使用者定義儲存程序介面所安裝的 DB2 公用程式。DB2 for OS/390 LOAD、REORG 及 RUNSTATS 公用程式提供內容表，供您用來變更公用程式如何執行。欲變更公用程式的內容，請在公用程式的「處理模型產生器」視窗上按一下右鍵，再按一下內容。

Warehouse Manager 亦提供 DSNUTILS 的介面，讓您在 Warehouse Manager 步驟中可併入 DB2 公用程式。

若要設定 DSNUTILS 儲存程序：

1. 當安裝 DB2 時，執行 DSNTIJSJ 工作來設定和連結 DSNUTILS 儲存程序。請確定 DSNUTILS 的定義有 PARAMETER STYLE GENERAL。
2. 啟用 WLM 管理的儲存程序。
3. 設定 RRS 及 WLM 環境。
4. 執行 DB2 提供的範例批次 DSNUTILS 程式。(此步驟非必要，但建議執行。)
5. 將 DSNUTILS 規劃與 DSNCLI 規劃作連結，如此 CLI 就可以呼叫儲存程序：

```
BIND PLAN(DSNAOCLI) PKLIST(*.DSNAOCLI.*, *.DSNUTILS.*)
```
6. 使用 Warehouse Manager 設定步驟並執行該步驟。母體類型應該是 APPEND。若不是，則執行公用程式之前，Warehouse Manager 會刪除表格中的所有項目。

使用 LOAD 公用程式在 DB2 for OS/390 表格之間複製資料

假設您想要將表格卸載至一個純文字檔來複製表格，然後將純文字檔載入至另一個表格。您通常必須卸載資料，編輯載入控制陳述式來卸載產生的資料，然後載入資料。透過倉儲，您可指定您要重新載入至不同的表格，不必在步驟之間停止和手動編輯控制陳述式。做法如下：使用 Reorg/Generic 介面來建立一個步驟，其中使用 UNLOAD 公用程式或 REORG TABLESPACE 公用程式來卸載一個檔案。這兩個公用程式會產生兩

個輸出資料集，其中一個含有表格資料，另一個含有可輸入至 LOAD 的公用程式控制陳述式。在公用程式所產生的控制陳述式中， INTO TABLE 表格名稱是已卸載的表格名稱。以下顯示您可在 Reorg Unload 步驟中使用的 DSNUTILS 參數範例：

表 33. Reorg Unload 步驟的內容

UTILITY_ID	REORGULX
RESTART	NO
UTSTMT	REORG TABLESPACE DBVW.USAINENT UNLOAD EXTERNAL
UTILITY_NAME	REORG TABLESPACE
RECDSN	DBVW.DSNURELD.RECDSN
RECDEVT	SYSDA
RECSPACE	50
PNCHDSN	DBVW.DSNURELD.PNCHDSN
PNCHDEVT	SYSDA
PNCHSPACE	3

使用 Reorg/Generic DSNUTILS 介面來建立一個載入步驟。DSNUTILS 公用程式陳述式參數通常指定一個公用程式控制陳述式。倉儲公用程式介面亦容許公用程式陳述式欄位中出現檔名。您可使用關鍵字 FILE 來指定包含有效控制陳述式的檔案，以及使用關鍵字 TABLE 來指定您要載入的表格名稱。若要使用 LOAD 公用程式來使用先前範例的輸出，請在 LOAD 內容中引用下列參數值：

註： 在 UTSTMT 欄位中，鍵入一個載入陳述式，或鍵入 REORG 公用程式和 UNLOAD EXTERNAL 選項所輸出的檔案名稱。

表 34. LOAD 步驟內容

UTILITY_ID	LOADREORG
RESTART	NO
UTSTMT	:FILE:DBVW.DSNURELD.PNCHDSN:TABLE:[DBVW].INVENTORY
UTILITY_NAME	LOAD
RECDSN	DBVW.DSNURELD.RECDSN
RECDEVT	SYSDA

這對於相同或不同 DB2 子系統上的任何 DB2 for OS/390 來源和目標表格皆有用。控制陳述式純文字檔可以是 HFS 或原始 MVS 檔案。

若需更多有關 OS/390 上可用的 DSNUTILS 與 DB2 公用程式的詳細資訊，請參閱 DB2 for OS/390 Utility Guide and Reference。

抄寫

您可以使用 OS/390 代理程式來將 Data Propagator 抄寫引用步驟自動化。抄寫需要有一個來源資料庫、一個控制資料庫，以及一個目標資料庫。這些可以是不同或相同的資料庫。一個擷取工讀取 DB2 日誌來判斷來源資料庫中已新增、更新或變更的列。然後，此工作將變更寫入一個變更資料表。然後會執行引用工作將變更套用到目標資料

庫中。DB2 Warehouse Manager 可建立抄寫步驟以自動化引用工作。使用 Warehouse Manager 定義要執行的引用工作類型及執行的時間。您必須將 SASNLINK 檔案庫匯出到 steplib 環境變數。

新增抄寫支援到「資料倉儲中心」模板

「資料倉儲中心」包括了一個 JCL 模板給抄寫支援。如果您計劃要使用這個 OS/390 代理程式來執行引用程式，則您必須為您的 OS/390 系統變更此模板中的帳戶及資料集資訊。若要變更模板：

1. 以一個有權限在 /usr/lpp/DWC/ 目錄中複製及更新檔案的 ID 登入。
2. 尋找 apply.jcl，並將此檔案另存為 systemname.apply.jcl，其中 systemname 是 MVS 系統的名稱。例如，如果在 STLMVS1 上，建立的檔案副本名為 STLMVS1.apply.jcl。
3. 使用文字編輯程式來自訂 JCL，以符合您站台的需求。變更帳戶資訊以符合您的 MVS 系統的標準帳戶資訊，並變更 STEPLIB DD 的資料集及 MSGS DD。
4. 如果需要，變更 EXEC 卡上的程式名稱。若需變更程式名稱的詳細資訊，請參閱 DB2 指南與參考手冊。請勿變更任何方括弧([]) 內的參數，比如 [USERID] 及 [APPLY_PARMS]。(方括弧分別為十六進位字元 x'AD' 及 x'BD'。若您的 TSO 終端機類型在 SPF 選項 0 中未設成 3278A，這些值可能會顯示成特殊字元，而不是方括弧。這並不會是問題，如果您沒有變更 x'AD' 或 x'BD'，或者任何字元之間的資料。)
5. 更新環境變數 VWS_TEMPLATES，指向複製的模板檔案的目錄。

下列範例顯示「資料倉儲中心」中的 JCL 模板：

引用 JCL 模板：

```
//[USERID]A JOB ,MSGCLASS=H,MSGLEVEL=(1,1),
// REGION=2M,TIME=1440,NOTIFY=&SYSUID
//* DON'T CHANGE THE FIRST LINE OF THIS TEMPLATE.
//* THE REMAINING JCL SHOULD BE MODIFIED FOR YOUR SITE.
//*****
//* RUN APPLY/MVS ON OS/390 DB2 6.1 *
//*****
//ASNARUN EXEC PGM=ASNAPV66,REGION=10M,
// [APPLY_PARMS]
//STEPLIB DD DISP=SHR,DSN=DPROPR.V6R1M0.SASNLINK
// DD DISP=SHR,DSN=DSN610.SDSNLOAD
//MSGS DD DSN=DPROPR.V2R1M0A.MSGS,DISP=SHR
//ASNASPL DD DSN=&&ASNASPL,DISP=(NEW,DELETE,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(10,1)),
// DCB=(RECFM=VB,BLKSIZE=6404)
//SYSTEM DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//
```

代理程式記載

許多 DB2 Warehouse Manager 元件 (比如伺服器、日誌器、代理程式及部份「資料倉儲中心」) 會將日誌寫入到日誌目錄 (指定在 VWS_LOGGING 環境變數中)。日誌檔為純文字。

您可從「資料倉儲中心」啓動代理程式日誌。在左窗格中，以滑鼠右鍵按一下**倉儲**，再按一下**內容**。在「追蹤層次」標籤上，將設定值變更成您要的追蹤層次。代理程式追蹤支援層次 0-4：

- 層次 1 - 進入/離開追蹤
- 層次 2 - 層次 1 加上除錯追蹤
- 層次 3 - 層次 2 加上資料追蹤
- 層次 4 - 內部緩衝區追蹤

當追蹤的設定高於層次 1，效能將會較慢。只有除錯時才應開啓追蹤。追蹤資訊會儲存在 AGNTxxx.LOG 檔案中，環境資訊會儲存在 AGNTxxx.SET 檔案中。

Windows NT 上的從屬站端快取

如果使用者嘗試去存取一台 Windows NT Server 機器上的 READ PERM DB 檔案，而在機器上是使用含有效符記的共用磁碟機來建立 DB2 資料鏈結，檔案會如預期開啓。然而，在此之後，使用相同符記的後續開啓要求並不會送到伺服器，而是由從屬站上的快取來服務。即使在符記到期之後，使用者仍可以看到檔案的內容，因為項目仍然在快取之中。然而，如果檔案是在一台 Windows NT Workstation 上，則此問題並不會發生。

解決方案是在 Windows NT Server 上將登錄項目

\\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services

Lanmanserver\Parameters\EnableOpLocks 設定為 0。藉由此登錄設定，每當透過共用磁碟機從從屬工作站存取伺服器上的檔案時，要求一定會送到伺服器，而不會由從屬站快取來服務。因此，所有要求都會重新驗證符記。

此解決方案的負面影響是這樣會影響所有透過共用磁碟機從伺服器存取檔案的整體效能。即使有此設定，如果檔案是透過對映到伺服器本身的共用磁碟機來存取的（當無法從不同的從屬站機器），結果是仍會從快取來服務要求。因此，符記過期並沒有生效。

註： 在所有的情況下，如果檔案存取是本端存取，而非透過共用磁碟機，符記驗證及後續的符記過期會如預期發生。

Enterprise Edition UNIX CD-ROM 的試用產品

適用於 UNIX 平台版本 6 及版本 7 的 DB2 Universal Database (UDB) Enterprise Edition (EE) CD-ROM，包含一套 90 天的 DB2 Connect Enterprise Edition (CEE) 試用版。由於 DB2 Connect 功能是建立於 DB2 UDB EE 產品中，您不需要在已安裝 DB2 UDB EE 的系統上安裝 DB2 CEE 產品就能使用 DB2 Connect 功能。如果您安裝了 90 天的 DB2 CEE 試用版，並且決定昇級至授權版本，就必須購買 DB2 CEE 產品並安裝 DB2 CEE 授權碼。您不需要重新安裝產品。在 DB2 EE 或 DB2 CEE for UNIX 快速入門一書中提供了安裝授權碼的指示。

如果您已隨著 EE 的安裝，安裝試用的 CEE 產品，卻不想要永久地安裝 CEE，您可以遵循下列指示，除去 90 天的 CEE 試用版。如果您除去 Connect EE 的試用版，將仍然可以使用 DB2 EE 的 DB2 Connect 功能。

若要除去 DB2 Connect 版本 7，請從各個平台解除安裝下列的檔案集：

- 在 AIX，解除安裝 db2_07_01.clic 檔案集。

- 在 NUMA-Q 及 Solaris 作業環境，解除安裝 db2clic71 套裝軟體。
- 在 Linux，解除安裝 db2clic71-7.1.0-x RPM。
- 在 HP-UX，解除安裝 DB2V7CONN.clic 檔案集。

若要除去 DB2 Connect 版本 6，請從各個平台解除安裝下列的檔案集：

- 在 AIX，解除安裝 db2_06_01.clic 檔案集。
- 在 NUMA-Q 及 Solaris 作業環境，解除安裝 db2cplic61 套裝軟體。
- 在 Linux，解除安裝 db2cplic61-6.1.0-x RPM。
- 在 HP-UX，解除安裝 DB2V6CONN.clic 檔案集。

DB2 Connect Enterprise Edition UNIX CD-ROM 的試用產品

適用於 UNIX 平台版本 6 及版本 7 的 DB2 Connect Enterprise Edition (EE) CD-ROM，包含一套 90 天的 DB2 Universal Database (UDB) Enterprise Edition (EE) 試用版。提供 90 天的 DB2 UDB EE 試用版作為評估之用，對 DB2 Connect 的運作並不是必要的。

如果您安裝了 90 天的 DB2 UDB EE 試用版，並且決定昇級至授權版本，就必須購買 DB2 UDB EE 產品並安裝 DB2 UDB EE 授權碼。您不需要重新安裝產品。在 DB2 EE 或 DB2 CEE for UNIX 快速入門一書中提供了安裝授權碼的指示。如果您已隨著 Connect EE 的安裝，安裝試用的 UDB EE 產品，卻不想要永久地安裝 CEE，您可以遵循下列指示，除去 90 天的 EE 試用版。如果您除去 DB2 UDB EE 的試用版，將不影響 DB2 Connect EE 的功能。

若要除去 DB2 UDB EE 版本 7，請從各個平台解除安裝下列的檔案集：

- 在 AIX，解除安裝 db2_07_01.elic 檔案集。
- 在 NUMA-Q 及 Solaris 作業環境，解除安裝 db2elic71 套裝軟體。
- 在 Linux，解除安裝 db2elic71-7.1.0-x RPM。
- 在 HP-UX，解除安裝 DB2V7ENTP.elic 檔案集。

若要除去 DB2 UDB EE 版本 6，請從各個平台解除安裝下列的檔案集：

- 在 AIX，解除安裝 db2_06_01.elic 檔案集。
- 在 NUMA-Q 及 Solaris 作業環境，解除安裝 db2elic61 套裝軟體。
- 在 Linux，解除安裝 db2elic61-6.1.0-x RPM。
- 在 HP-UX，解除安裝 DB2V6ENTP.elic 檔案集。

捨棄 Data Links Manager

您現在可以捨棄指定資料庫的 DB2 Data Links Manager。某些資料鍵結相關的 SQL 要求之處理程序與公用程式一樣 (例如，備份/復置)，包含了與已架構資料庫的 DLM 之間的通信。在以前，DB2 無法捨棄已架構的 DLM，即使它尚未被選用。這導致 SQL 及公用程式處理程序附加的額外負荷。一旦新增 DLM，在要求的處理程序中與它通信的引擎，可能會導致某些 SQL 要求的失敗 (例如，除去表格/表格空間/資料庫)。

使用 SMIT 解除安裝 DLFM 元件可能會除去額外的檔案集

在從安裝 Data Links Manager 的 AIX 機器上解除安裝 DB2 (版本 5、6 或 7) 之前，請遵循下列步驟：

1. 以 root 的身分，使用下列指令備份 /etc/vfs：

```
cp -p /etc/vfs /etc/vfs.bak
```
2. 解除安裝 DB2。
3. 以 root 的身分，用步驟 1 所做的備份取代 /etc/vfs：

```
cp -p /etc/vfs.bak /etc/vfs
```

在 Windows 2000 上使用 CLP 時發生錯誤 SQL1035N

如果安裝 DB2 到只有某些使用者 (例如，管理者) 才具有寫入權的目錄，一般使用者會在嘗試使用 DB2 命令行處理器時收到錯誤 SQL1035N。

若要解決此問題，應在所有使用者都具有寫入權的目錄中安裝 DB2。

SQL 輔助程式的加強功能

SQL 輔助程式工具現在允許使用者為表格結合指定不同於 "=" 的結合運算子。加強功能的「結合類型」對話框 (在 SQL 輔助程式工具的「結合」頁上按一下「結合類型」按鈕就能啟動該對話框)，包含結合運算子的下拉列示。

可用的運算子有 "="、"<>"、"<"、">"、"<=" 及 ">="。SQL 輔助程式是協助使用者建立簡式 SQL 陳述式的工具。它可用於「命令中心」(「交談式」標籤)、「控制中心」(「建立概略表」及「建立觸發函式」對話框)、儲存程序建置器 (「插入 SQL 儲存程序」精靈) 及「資料倉儲中心」(SQL 處理步驟)。

Linux 上的 DB2 的 Gnome 和 KDE 桌面整合

DB2 現在提供一組公用程式來建立 DB2 桌上管理程式資料夾和圖示，用以在 Gnome 和 KDE 桌上管理程式上啟動最常用的 DB2 工具，供以 Intel 為基礎的 Linux 分送程式使用。DB2 Version 7.2 依預設值會安裝這些工具，且可於安裝之後用來建立和除去一或多個使用者的桌上管理程式圖示。

欲新增一組桌上管理程式圖示給一或多個使用者，請使用下列指令：

```
db2icons <user1> [<user2> <user3>...]
```

註：請注意，若在 Gnome 或 KDE 桌上管理程式環境執行時產生圖示，使用者可能需要手動復新桌上管理程式才能看見新圖示。

欲除去一或多個使用者的一組桌上管理程式圖示，請使用下列指令：

```
db2rmicons <user1> [<user2> <user3>...]
```

註：您必須具備足夠的權限才能產生或除去其他使用者的圖示。通常，如果您是一般使用者的話，**db2icons** 和 **db2rmicons** 可用來建立或除去您自己的圖示，但唯有您是 root 使用者或是具有權限寫入指定之使用者起始目錄的另一個使用者，您才能夠用這兩個指令來建立或除去別人的圖示。

在 Windows 2000 Terminal Server 管理模式下執行 DB2

對於 DB2 UDB Version 7.1, FixPak 3 及更高的版本，DB2 可在 Windows 2000 Terminal Server 管理模式下執行。在此之前，您無法在 Windows 2000 Terminal Server 管理模式的從屬站階段作業下執行 DB2。

備份與復置指令的線上說明

Incorrect information appears when you type db2 ? backup. 正確的輸出為：

```
BACKUP DATABASE database-alias [USER username [USING password]]
[TABLESPACE (tblspace-name [ {,tblspace-name} ... ])] [ONLINE]
[INCREMENTAL [DELTA]] [USE TSM [OPEN num-sess SESSIONS]] |
TO dir/dev [ {,dir/dev} ... ] | LOAD lib-name [OPEN num-sess SESSIONS]
[WITH num-buff BUFFERS] [BUFFER buffer-size] [PARALLELISM n]
[WITHOUT PROMPTING]
```

當您鍵入 db2 ? restore 時，出現不正確資訊。正確的輸出為：

```
RESTORE DATABASE source-database-alias { restore-options | CONTINUE | ABORT }";
```

```
restore-options:";
[USER username [USING password]] [{TABLESPACE [ONLINE] |};
TABLESPACE (tblspace-name [ {,tblspace-name} ... ]) [ONLINE] |";
HISTORY FILE [ONLINE]]] [INCREMENTAL [ABORT]]";
[{{USE TSM [OPEN num-sess SESSIONS] |};
FROM dir/dev [ {,dir/dev} ... ] | LOAD shared-lib";
[OPEN num-sess SESSIONS}}] [TAKEN AT date-time] [TO target-directory]";
[INTO target-database-alias] [NEWLOGPATH directory]";
[WITH num-buff BUFFERS] [BUFFER buffer-size]";
[DLREPORT file-name] [REPLACE EXISTING] [REDIRECT] [PARALLELISM n]";
[WITHOUT ROLLING FORWARD] [WITHOUT DATALINK] [WITHOUT PROMPTING]";
```

"Warehouse Manager" 應為 "DB2 Warehouse Manager"

所有出現在產品螢幕和產品文件的 "Warehouse Manager" 都應讀為 "DB2 Warehouse Manager"。

第9篇 附加資訊

附加資訊

DB2 Universal Database 和 DB2 Connect 線上支援

DB2 資訊的完整和保持最新來源，包括有關此文件出版後問題探索的資訊，請使用 DB2 Universal Database & DB2 Connect 線上支援網站：

<http://www.ibm.com/software/data/db2/udb/winos2unix/support>。

DB2 Magazine

有關 DB2 產品系列的最新資訊，您可以訂閱免費的 DB2 Magazine。線上版的雜誌可於 <http://www.db2mag.com> 取得；在網頁上也有訂閱雜誌的指示。

第10篇 附錄與後記

附錄. 注意事項

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。要知道在您所在之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 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，其它非 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。

在這本書或文件中可能包含著 IBM 所擁有之專利或專利申請案。本書使用者並不享有前述專利之任何授權。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，請聯絡您國家的 IBM 智慧財產部門，或者用書面方式寄到：

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

下列段落若與該國之法律條款抵觸，即視為不適用： IBM 僅以現狀提供本書，而不提供任何明示或默示之保證 (包括但不限於可售性或符合特定效用的保證)。若有些地區在某些交易上並不允許排除上述保證，則該排除無效。

本書中可能會有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。同時，IBM 得隨時改進並 (或) 變動本書中所提及的產品及 (或) 程式。

本書對於非 IBM 網站的援引只是為了方便而提供，並不對這些網站作任何認可。該些網站上的內容並非本 IBM 產品內容的一部份，用戶使用該網站時應自行承擔風險。

當您提供資訊給 IBM 時，您即授權予 IBM 以其認為適當的方式來使用或分送資訊，而不必對您負起任何責任。

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它程式 (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 Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

上述資料的取得有其特殊要件，在某些情況下必須付費方得使用。

本書所描述的任何程式及其所有可用的授權著作是由 IBM 所提供，並受到「IBM 客戶合約」、「國際程式授權合約」或雙方之間任何同等合約條款之規範。

此間所含之任何效能資料，皆是得自控制的環境之下；因此不同作業環境之下所得的結果，可能會有很大差異。部份測量可能是在開發中的系統上執行，因此不保證可以從一般的系統獲致相同的結果。甚至有部份的測量，是利用插補法而得的估計值，其實際結果可能會有不同。本書的使用者應根據其特有的環境，驗證出適用的資料。

本書所提及之非 IBM 產品資訊，係一由產品的供應商，或其出版的聲明或其它公開管道取得。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性、或任何對產品的其它主張是否完全無誤。如果您對非 IBM 產品的性能有任何的疑問，請逕向該產品的供應商查詢。

有關 IBM 未來動向的任何陳述，僅代表 IBM 的目標而已，並可能於未事先聲明的情況下有所變動或撤回。

本書中含有日常商業活動所用的資料及報告範例。爲了提供完整的說明，這些範例包括個人、公司、廠牌和產品名稱。這些名稱全屬虛構，若與任何公司的名稱和住址雷同，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、銷售或散佈符合作業平台 (用於執行所撰寫的範例程式) 之應用程式設計介面的應用程式等目的，以任何形式複製、修改及散佈這些範例程式，而無需付費給 IBM。但這些範例皆未經過完整的測試。因此，IBM 不會保證或暗示這些程式的穩定性、服務能力或功能。

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：

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

商標

下列術語 (以星號 (*) 標示) 是 IBM 公司在美國、其它國家或兩者的商標。

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extenders	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	VisualAge
eNetwork	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
	WIN-OS/2

下列術語是其它公司的商標或註冊商標：

Microsoft、Windows、Windows NT 是微軟公司的商標或註冊商標。

Java 以及所有與 Java 有關的商標與標章，以及 Solaris 是 Sun Microsystems, Inc. 在美國、其它國家或兩者的商標。

Tivoli 與 NetView 是 Tivoli Systems Inc. 在美國、其它國家或兩者的商標。

UNIX 是 X/Open Company Limited 在美國、其它國家或兩者的註冊商標，須經該公司授權始可使用。

其它公司、產品或服務名稱 (以兩顆星號 (**)) 標示) 可能是其它公司的商標或服務標誌。

索引

索引順序以中文字，英文字，及特殊符號之次序排列。

〔四劃〕

- 元件追蹤資料檔名 227
- 分割的資料庫
 - 新增節點時發生錯誤 4
- 日文
 - Linux 上的語言環境設定 6

〔六劃〕

- 名稱與位址篩選, Trillium 216
- 多位置更新測試連接
 - 案例版本不相容 190
- 安裝
 - 必要的服務帳戶於 Windows 457
 - 在使用可移除式磁碟機時中斷 6
 - 停止 DB2 程序於 15
- 安裝先決條件
 - 適用 UDB EE 及 CEE 於 Linux/390 16
- 安裝, 安裝, 移轉, 可執行檔 209

〔七劃〕

- 快速導覽 451
- 抄寫
 - 表格及直欄名稱的限制 154
 - 非 IBM 伺服器 153
 - 問題判斷 156
 - 規劃 155
 - 實務範例 153
 - 隨處更新(update-anywhere)先決條件 155
 - DATALINK 154
- 抄寫分析程式 156
- 身分驗證
 - 使用 DB2DOMAINLIST 依據網域 453

〔八劃〕

- 協助工具 457
- 協助工具特性 3
- 表格結構 159
- 非 DB2 來源, Sybase 的範例登錄 228
- 非 DB2 倉儲來源, AIX 227
- 非 DB2 倉儲來源, Solaris 作業環境 227

〔九劃〕

- 建立資料庫精靈
 - 設定擴充大小 197
- 查詢結果, 檢視 193
- 紅旗 Linux
 - 簡體中文語言環境 6

〔十劃〕

- 修補層次
 - 於 Solaris 版本 2.6 4
- 記憶體視窗
 - 於 HP-UX 11 452
- 訊息
 - 擷取程式和引用程式 159
- 訊息, MQ Series 224

〔十一劃〕

- 從屬站
 - 下載 15
- 控制中心
 - 工具設定筆記本 193
 - 用來管理 VM 及 VSE 資料庫 189
 - 並行 Applet 的限制 193
 - 使用 Internet Explorer 的錯誤 6
 - 於 OS/390 平台 190
 - 疑難排解 192
 - 雙向支援
 - 於 Windows NT 189
 - OS/2 上的 Infopop 顯示問題 192
 - OS/390 必要的修正程式 191
 - UNIX 平台上的疑難排解 192
 - Version 6, using to administer satellite environment 120
 - Windows 95 上的 Script 中心問題 193
- 移轉
 - 於 Windows 2000 15
 - 執行 db2iupdt 來更新案例 16
 - 發生錯誤於 6
 - 概略表的問題 15
- 處理模型產生器
 - 資料倉儲中心 243
- 軟體需求, ERwin 標示語言, 資料倉儲中心, DWC, 資訊型錄管理程式, ICM 209

〔十二劃〕

- 備份服務 APIs
 - DB2 支援 458

- 復置
 - DATALINK 直欄的問題 454
- 無效的捷徑錯誤 189
- 登記變數
 - 支援備份服務 APIs 458
 - DB2DOMAINLIST 453
- 程序, 安裝前停止 15

〔十三劃〕

- 傳訊, MQ Series 223
- 匯入, 正在匯入, 標示語言檔, DWC 211
- 匯入, 標示語言檔, ICM, 資訊型錄管理程式 212
- 搜尋線上資訊
 - 於 Solaris 作業系統 9
- 概略表
 - 移轉後無法使用 15
- 解除安裝
 - DB2 DFS 從屬站啟動程式 453
- 資料倉儲中心
 - 使用儲存程序 458
 - iwh2exp2 指令語法 458
 - OLE DB 支援 226
- 資料篩選, 名稱與位址篩選 216
- 資訊型錄管理程式
 - 公用程式 267
 - 適用於希伯來文 455

〔十四劃〕

- 精靈
 - 建立資料庫 197
 - MQSeries 輔助 197
 - OLE DB 輔助 197
- 綱目模型產生器
 - 資料倉儲中心 244
- 語言環境
 - 相容於「管理伺服器」和案例 456
 - 紅旗 Linux 上的簡體中文 6
 - Linux 上的日文及簡體中文 6
- 遠端從屬站存取
 - 架構 Communication Server for NT 454

〔十五劃〕

- 增量 COMMIT 與 replace 227
- 標示語言, DWC, ICM, ERwin 檔, ER1 檔 209
- 範例內容, 檢視 193

衛星管理中心

Version 6, using to administer satellite environment 120

〔十六劃〕

錯誤訊息

找不到連結檔案 452
移轉期間 6
無效的捷徑 189
新增節點至分割的資料庫時 4
檔案拒絕存取 190
SQL10012N 192

〔十七劃〕

儲存程序

於資料倉儲中心 458

檔案拒絕存取錯誤 190

聯合系統

外層 33
限制 33
暱名 33

聯合系統的限制 455

〔十八劃〕

擴充大小

建立資料庫精靈中的設定 197

擷取程式和引用程式

於 AS/400 156
從應用程式內啟動 160
錯誤訊息 159

簡體中文

Linux 上的語言環境設定 6

雙向語言支援

使用 Windows NT 上的控制中心 189

A

accessing multiple indexes 96

Address Windowing Extensions (AWE) 96

Adobe Acrobat Reader

使用非英文語言環境來存取 10

ADT transforms 330

AIX 4.3.3

AutoLoader 在建立子程序後當掉 454

alert center 63

ALTER SEQUENCE statement

detailed description 384

ambiguous cursor 403

app_ctl_heap_sz 101

Archive Log 131

archiving and retrieving

user exit program 86

archiving logs on demand 78

assigning strings to columns, rules for 433

assignments

strings, basic rules for 433

AS/400

擷取程式和引用程式 156

authorization

granting create on sequence 406

public create on sequence 406

AutoLoader

在建立子程序時當掉 454

B

backup

incremental 73

offline

from split mirrored image 76

to named pipes 75

Baltic Rim languages

code page support 60

binding database utilities using Run-Time

Client 329

buffer pools

AWE 96

C

catalog views

SEQUENCES 423

character conversion

rules for operations combining

strings 434

rules when comparing strings 434

character string

assignment, overview 433

CLI

writing a Unicode application 435

CLI functions

SQLBindFileToParam 333

SQLNextResult 333

CLI interface

code page settings 60

CLI stored procedures

autobinding 333

limitations 332

clients

three-tier environment 331

clone databases

creating 71

code pages

Baltic Rim languages 60

in applications using CLI 60

mapping to locales 57

column

string assignment, basic rules for 433

Commit.tag template

example values 246

tokens 246

Common Warehouse Metamodel

XML 支援 243

compound SQL, using 332

Compound Statement (Dynamic)

variables 391

compound statements

dynamic 90

concatenation

operator 435

configuration parameter

MIN_DEC_DIV_3 100

configuration parameters

app_ctl_heap_sz 101

maxappls 101

maxlocks 99

mon_heap_sz 101

softmax 101

containers

adding to an SMS table space 67

Control Center

running translated versions 59

conversion rules

for operations combining strings 434

for string comparisons 434

copy 常駐程式 453

crash recovery

using db2inidb tool 71

CREATE PROCEDURE statement

dynamic compound statement 391

FOR statement 416

RETURN statement 419

SIGNAL statement 421

SQL procedure statement 414

CREATE SEQUENCE statement

detailed description 397

cursors

attributes, default 331

concurrency 331

keyset-driven

updating 331

scrollable 331

keyset-driven 331

OS/390 331

server-side 331

static 331

troubleshooting applications 332

sensitivity 331

type 331

updatable 331

values concurrency 331

CWM

XML 支援 243

D

data encryption 70
Data Links File System
 incompatible with Norton Utilities 56
Data Movement
 limits of Unicode clients 442
data types
 promotion in a Unicode database 431
DataJoiner
 限制 455
DATALINK
 抄寫 154
 復置時發生問題 454
DAYOFWEEK_ISO scalar function 336
DB2 Connect
 prerequisites on Sun Cluster 2.2 54
DB2 control server
 minimum requirements 109
 response file installation 109
 Version 6, upgrading 113
DB2 Data Links Manager
 interactions with recovery 82
DB2 Enterprise Edition
 setting up as satellite 109
 prerequisites 109
 Version 6, using as DB2 control server 113
DB2 Personal Edition
 setting up as satellite 109
 prerequisites 109
DB2 Relational Connect
 在 Windows NT 上安裝 34
 安裝於 AIX、Solaris 與 Linux 35
DB2 Server for VM & VSE
 從控制中心管理資料庫 189
db2ArchiveLog 290
DB2CODEPAGE registry variable
 in applications using CLI 60
db2expln
 running 108
db2inidb tool 71
db2iupdt 指令，升級 DB2 後執行 16
DB2MSCS utility, machine reboot to set PATH 54
db2start
 on Windows 95, 98 and ME 163
db2_all 87
DB2_AWE registry variable 97
DB2_INDEX_2BYTEVARLEN 67
deadlocks 166
DECLARE CURSOR 403
DECRYPT function
 detailed description 350
 values and arguments 350

decrypting information
 DECRYPT function 350
DFS 從屬站啟動程式
 解除安裝 453
distributed databases
 setup for updating 53
dlfm client_conf
 失敗理由 453
dl_expint setting recommendation 100
DMS containers
 creating and extending in parallel 63
DMS table spaces
 parallel creation and resizing 69
dual logging 78
DWC
 CWM XML 支援 243
 OLE DB 支援 226
dynamic compound statement 391
dynamic compound statements 90
dynexpln
 running 108

E

enable_MQFunctions 183, 185
ENCRYPT function
 detailed description 351
 values and arguments 351
encrypting information
 ENCRYPT function 351
 GETHINT function 353
encryption
 data 70
error messages
 executing triggers 403
ERwin, 描述資料, 取出, 匯入 208
expression
 concatenation operator 435
 string 435
expressions
 NEXTVAL 66
 PREVVAL 66
 sequences 346

F

FOR statement 416
ForeignKeyAdditional.tag template
 example values 250
 tokens 249
ForeignKey.tag template
 example values 248
 tokens 246
functions
 DECRYPT 70

functions (繼續)
 ENCRYPT 70
 GETHINT 70
 in a Unicode database 435
 procedures 378
 scalar
 DECRYPT 350
 ENCRYPT 351
 GETHINT 353
 IDENTITY_VAL_LOCAL 354
 MQPUBLISH 359
 MQREAD 360
 MQRECEIVE 361
 MQSEND 363
 MQSUBSCRIBE 364
 MQUNSUBSCRIBE 365
 REC2XML 368
 table
 MQREADALL 374
 MQRECEIVEALL 376

G

gateways
 three-tier environment 331
GETHINT function
 detailed description 353
 values and arguments 353
governor configuration file 98
GRANT
 CREATE ON SEQUENCE 406
GRANT (Sequence Privileges) statement
 detailed description 406
GUI 工具
 連結錯誤 452

H

high availability 71
High Availability on Solaris 54
HP and Sun Solaris
 backup and restore support 79
HP-UX
 不支援「查詢啟動程式」 457
HP-UX 11
 記憶體視窗 452

I

identity column
 altering 68
IDENTITY columns 67
IDENTITY_VAL_LOCAL function
 detailed description 354
 values and arguments 354

incremental backup and recovery 73

index keys

larger 90

indexes

accessing multiple 96

larger keys 90

inoperative triggers

detailed description 402

Internet Explorer

使用控制中心的錯誤 6

IPX/SPX 通信協定支援，於 Windows

2000 15

isolation levels

in DELETE statement 404

in INSERT statement 407

in SELECT INTO statement 408

in SELECT statement 381

in UPDATE statement 413

statement-level 89

iw2exp2 指令

新的選項 458

J

Java methods

in CREATE PROCEDURE or CREATE

FUNCTION 327

JDBC

scrollable cursors, accessing 331

JDBC/ODBC/CLI static profiling,

limitations 329

JDK 1.1

安裝路徑 192

L

Linux

日文及簡體中文的語言環境設定 6

設定控制中心環境 16

locale

compatibility between Administration

Server and instance 60

locale settings

in applications using CLI 60

locales

national language versions 57

lock waits 166

locking, optimistic 331

locklist

maxlocks configuration parameter 99

log

mirroring 78

logs

archiving on demand 78

M

maxappls 101

Microsoft SQL Server 資料來源

字碼頁 47

增進效能 44

環境變數 43

鏈結至 DB2 44

DSN 名稱 45

MERANT 外層 44

MERANT 檔案庫 42

ODBC 追蹤 47

ODBC 驅動程式 42

mirroring

logs 78

mon_heap_sz 101

MQ Functions 183

MQ 功能 185

MQPUBLISH function

detailed description 359

values and arguments 359

MQREAD function

detailed description 360

values and arguments 360

MQREADALL function

detailed description 374

values and arguments 374

MQRECEIVE function

detailed description 361

values and arguments 361

MQRECEIVEALL function

detailed description 376

values and arguments 376

MQSEND function

detailed description 363

values and arguments 363

MQSeries 輔助精靈 197

MQSUBSCRIBE function

detailed description 364

values and arguments 364

MQUNSUBSCRIBE function

detailed description 365

values and arguments 365

N

named pipes

backing up to 75

naming conventions

character limits 55

naming rules 56

national language support 60

Netscape

用來存取線上資訊 9

錯誤訊息 9, 195

next key locking 166

NEXTVAL 66

nextval-expression 346

NLV 支援

Unicode 429

Norton Utilities

incompatible with Data Links File

System 56

NUMA-Q

不支援「查詢啟動程式」或「追蹤器」

457

O

ODBC

fetch orientations 331

scrollable cursors, accessing 331

ODBC Unicode applications 440

OLE DB 支援 226

OLE DB 輔助精靈 197

on demand log archiving 78

Open Client, 非 DB2 倉儲來源和 227

operand

string 435

Oracle 資料來源

環境變數 35

Linux 外層 35

Solaris 外層 35

OS/2

安裝控制中心 190

訊息所需的 JDK 層次 309

P

parallel recovery 75

partitioning keys

updating 53

performance monitor 63

prefetchers

in parallel DMS container creation and

extension 63

PREVVAL 66

prevval-expression 346

PrimaryKeyAdditional.tag template

example values 253

tokens 252

PrimaryKey.tag template

example values 251

tokens 250

privileges

USAGE 70

Q

- Query Enabler
 - 在 HP-UX 及 NUMA-Q 上不受支援 457
- Query Patroller
 - 在移轉之後遺失專用權 457
- Query Patroller 追蹤器
 - 在 NUMA-Q 上不受支援 457

R

- REC2XML function
 - detailed description 368
 - values and arguments 368
- recovery
 - incremental 73
 - interaction with DB2 Data Links Manager 82
 - parallel 75
- registry variables
 - DB2_AWE 97
 - new and changed 103
- replicated summary tables 53, 94
- response file installation
 - DB2 control server keywords 109
 - satellite-specific keywords 110
- restore
 - incremental 73
- return identity column value
 - IDENTITY_VAL_LOCAL function 354
- RETURN statement 419
- routines
 - procedures 378

S

- satellite environment
 - administering with Version 6 衛星管理中心 120
 - DB2 Personal Edition satellite 109
 - DB2 Workgroup Edition satellite 109
 - installation considerations 110
 - interactive installation of satellite 111
 - prerequisites 109
 - response file installation of satellite 110
- satellites
 - DB2 Personal Edition 109
 - DB2 Workgroup Edition 109
 - installation considerations 110
 - interactive installation 111
 - response file installation 110
- scalar functions
 - DAYOFWEEK_ISO 336
 - WEEK_ISO 336

- schema names
 - character limits 55
- Search 探索 452
- sequence invocation 346
- sequences 67
 - altering 68
 - creating 65
 - dropping 69
 - invoking 346
 - nextval-expression 346
 - prevval-expression 346
 - privileges 70
- sequences, description 319
- SET CONSTRAINTS statement
 - replaced by SET INTEGRITY 57
- SET ENCRYPTION PASSWORD 70
- SET ENCRYPTION PASSWORD statement
 - detailed description 409
- SET INTEGRITY statement
 - replacing SET CONSTRAINTS 57
- SET Variable statement
 - detailed description 410
- SIGNAL statement 421
- SIGTTIN 87
- SMS table spaces
 - adding containers 67
- SNA Server
 - 從主電腦或 AS/400 應用程式存取 UDB 456
- SNA SPM
 - 於重新啟動之後啟動驗證 456
- softmax 101
- Solaris
 - 不支援 UltraSparc 之前的版本 4
 - V2.6 所需的修補層次 4
- Solaris 作業系統
 - 64 位元
 - 架構 169
 - 限制 169
- Solaris 作業環境
 - 搜尋線上資訊於 9
- sort heap size 98
- sort heap threshold 98
- split mirror
 - as a standby database 72
- split mirror handling
 - online 71
- SQL procedure
 - DECLARE statement 391
 - dynamic compound statement 391
 - FOR statement 416
 - RETURN statement 419
 - SIGNAL statement 421
 - variables 391
- SQL statements
 - ALTER SEQUENCE 384

- SQL statements (繼續)
 - CREATE SEQUENCE 397
 - GRANT (Sequence Privileges) 406
 - SET ENCRYPTION PASSWORD 409
 - SET Variable 410
- SQL variables 391
- SQLBindFileToParam CLI function,
 - correction 333
- SQLNextResult function 333
- SQL, compound 332
- starting DB2
 - using db2start on Windows 95, 98 and ME 163
- statement-level isolation 89
- static profiling, in JDBC/ODBC/CLI applications 329
- stdin 87
- string
 - expression 435
 - operand 435
- sub-element statistics 92
- Sun Cluster 2.2
 - DB2 Connect prerequisites 54
- Sun Solaris and HP
 - backup and restore support 79
- suspended I/O function
 - supporting continuous availability 71
- SWITCH ONLINE clause 69
- Sybase
 - 在 Windows 上與 ICM 不相容 7
- Sybase Open Client 36
- Sybase 資料來源 36
 - 字碼頁 42
 - 增進效能 39
 - 環境變數 37
 - 鏈結至 DB2 38
- system catalog statistics
 - collecting distribution statistics 91

T

- tablespaces
 - setting to ONLINE state 69
- template
 - Commit.tag
 - example values 246
 - token 246
 - ForeignKeyAdditional.tag
 - example values 250
 - tokens 249
 - ForeignKey.tag
 - example values 248
 - tokens 246
 - PrimaryKeyAdditional.tag
 - example values 253
 - tokens 252

template (繼續)

- PrimaryKey.tag
 - example values 251
 - tokens 250

triggered SQL statements

- SET Variable statement 410

triggers

- error messages 403
- inoperative 402

U

Unicode

- 文件更新 431
- 資料庫及應用程式 429
- functions in 435
- in ODBC applications 440

Unicode clients

- data movement limitations 442

USAGE privilege 70

user IDs

- character limits 55

V

Veritas

- high availability on Solaris 54

VI

- setting up for use with DB2 87

W

Web 瀏覽器

- 建議 - Windows 2000 9

WEEK_ISO scalar function 336

Windows

- Sybase 及 ICM 間的不相容 7

Windows 2000

- 建議的 Web 瀏覽器 9
- 移轉於 15
- IPXX/SPX 通信協定支援 15

X

XML 支援

- Common Warehouse Metamodel (CWM) 243



Printed in Singapore