

IBM® DB2 Universal Database™



應用程式開發手冊：建置及執行應用程式

版本 8

IBM® DB2 Universal Database™



應用程式開發手冊：建置及執行應用程式

版本 8

使用此資訊及其支援的產品之前，請先閱讀注意事項下的一般資訊。

本文件含有 IBM 的所有權資訊。它是依據軟體使用權同意書而提供的，並受到著作權法的保護。本書中的資訊不包括任何產品保證，且其陳述也不得延伸解釋。

您可以線上訂購 IBM 出版品，或透過當地的 IBM 業務代表來訂購。

- 若要線上訂購出版品，請造訪位於 www.ibm.com/shop/publications/order 的「IBM 出版品中心 (IBM Publications Center)」。
- 若要尋找您當地的 IBM 業務代表，請造訪位於 www.ibm.com/planetwide 的「IBM 全球聯絡站名錄 (IBM Directory of Worldwide Contacts)」。

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

當您傳送資訊給 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。

© Copyright International Business Machines Corporation 1993 - 2002. All rights reserved.

目錄

關於本書	vii	移轉應用程式	44
		下一步要做什麼	52
第 1 篇 應用程式開發環境	1	第 3 章 範例程式及相關檔案	53
第 1 章 DB2 支援	3	範例檔案	53
DB2 Application Development Client	3	範例程式：結構及設計	58
資料庫管理程式案例	5	依語言及應用程式介面的範例程式	65
DB2 支援的伺服器	7	C/C++ 範例	65
用於開發及執行應用程式的 DB2 支援軟體	8	DB2 CLI 範例	68
依平台支援的軟體	9	JDBC 範例	70
用於開發及執行應用程式的 AIX 支援軟體	9	SQLJ 範例	72
用於開發及執行應用程式的 HP-UX 支援軟體	10	Java WebSphere 範例	74
用於開發及執行應用程式的 Linux 支援軟體	11	Java 外掛程式範例	75
用於開發及執行應用程式的 Solaris 支援軟體	13	COBOL 範例	76
用於開發及執行應用程式的 Windows 支援軟體	14	SQL 程序範例	80
第 2 章 安裝	17	Visual Basic 範例	82
一般安裝資訊	17	Visual C++ 範例	83
設置應用程式開發環境	17	Windows Management Instrumentation 範例	84
更新「資料庫管理程式架構檔」	19	物件連結和內嵌 (OLE) 範例	84
設置 Java 環境	20	物件連結和內嵌資料庫 (OLE DB) 表格函數範例	85
設定 SQL 程序環境	21	命令行處理器 (CLP) 範例	86
UNIX	22	REXX 範例	86
設置 UNIX 應用程式開發環境	22	日誌管理使用者跳出程式範例	88
UNIX 環境變數設定值	23	建置檔、make 檔及錯誤檢查公用程式	90
Java	24	建置檔	90
SQL 程序	28	make 檔	93
Windows	32	錯誤檢查公用程式	96
設置 Windows 應用程式開發環境	32	第 2 篇 建置及執行沒有平台限制的 應用程式	99
Java	35	第 4 章 Java	101
SQL 程序	37	Java 範例程式	101
範例資料庫	39	Java Applet 注意事項	103
設置範例資料庫	39	JDBC	105
建立範例資料庫	39	建置 JDBC Applet	105
在主電腦或 AS/400 及 iSeries 伺服器上建立範例資料庫	41	建置 JDBC 應用程式	106
編目範例資料庫	42	建置 JDBC 常式	107
連結範例資料庫公用程式	43	SQLJ	110
		建置 SQLJ 程式	110

建置 SQLJ Applet	111	C 多緒應用程式的建置 Script	154
建置 SQLJ 應用程式	113	VisualAge C++	155
SQLJ 應用程式及 Applet 的 UNIX 建置 Script	114	在 AIX 上建置 C++ 應用程式	155
UNIX 的 SQLJ 應用程式選項	115	C++ 應用程式的建置 Script	157
SQLJ 應用程式及 Applet 的 Windows 批次檔	116	AIX C++ 應用程式編譯及鏈結選項	158
Windows 的 SQLJ 應用程式選項	118	在 AIX 上建置 C++ 常式	159
建置 SQLJ 常式	119	C++ 常式的建置 Script	162
SQLJ 常式的 UNIX 建置 Script	120	AIX C++ 常式編譯及鏈結選項	163
UNIX 的 SQLJ 儲存程序選項	122	在 AIX 上建置 C++ 多緒應用程式	164
SQLJ 常式的 Windows 批次檔	123	C++ 多緒應用程式的建置 Script	166
Windows 的 SQLJ 儲存程序選項	124	VisualAge C++ 架構檔	166
第 5 章 SQL 程序	127	利用架構檔建置 VisualAge C++ 程式	166
建立 SQL 程序	127	利用架構檔建置 C++ DB2 API 應用程式	167
在 UNIX 上利用從屬站應用程式呼叫 SQL 程序	128	利用架構檔建置 C++ 內嵌式 SQL 應用程式	168
在 Windows 上利用從屬站應用程式呼叫 SQL 程序	129	利用架構檔建置 C++ 儲存程序	169
保留 SQL 程序的中間檔	131	利用架構檔建置 C++ 使用者定義的函數	170
自訂 SQL 程序的前置編譯及鏈結選項	132	IBM COBOL Set for AIX	171
備份及復置 SQL 程序	133	在 AIX 上架構 IBM COBOL 編譯器	171
分送編譯後的 SQL 程序	134	在 AIX 上建置 IBM COBOL 應用程式	172
重新連結 SQL 程序	135	IBM COBOL 應用程式的建置 Script	174
第 6 章 呼叫儲存程序	137	AIX IBM COBOL 應用程式編譯及鏈結選項	174
利用 CALL 陳述式呼叫儲存程序	137	在 AIX 上建置 IBM COBOL 常式	175
第 3 篇 建置及執行特定平台專用的應用程式	139	IBM COBOL 常式的建置 Script	177
第 7 章 AIX	141	AIX IBM COBOL 常式編譯及鏈結選項	177
重要注意事項	141	Micro Focus COBOL	179
常式的 AIX 匯出檔案	142	在 AIX 上架構 Micro Focus COBOL 編譯器	179
AIX 常式及 CREATE 陳述式	142	在 AIX 上建置 Micro Focus COBOL 應用程式	179
取代 AIX 共用檔案庫	143	Micro Focus COBOL 應用程式的建置 Script	181
在 AIX 上安裝 COBOL 的注意事項	143	AIX Micro COBOL 應用程式編譯及鏈結選項	182
IBM C	144	在 AIX 上建置 Micro Focus COBOL 常式	183
在 AIX 上建置 C 應用程式	144	Micro Focus COBOL 常式的建置 Script	184
C 應用程式的建置 Script	146	AIX Micro COBOL 常式編譯及鏈結選項	185
AIX C 應用程式編譯及鏈結選項	146	REXX	186
在 AIX 上建置 C 常式	148	在 AIX 上建置 REXX 應用程式	186
C 常式的建置 Script	151	第 8 章 HP-UX	189
AIX C 常式編譯及鏈結選項	152	HP-UX C	189
在 AIX 上建置 C 多緒應用程式	153	在 HP-UX 上建置 C 應用程式	189
		C 應用程式的建置 Script	191
		HP-UX C 應用程式編譯及鏈結選項	192

在 HP-UX 上建置 C 常式	193	在 Linux 上建置 C++ 多緒應用程式.	241
C 常式的建置 Script	197	C++ 多緒應用程式的建置 Script	241
HP-UX C 常式編譯及鏈結選項	198		
在 HP-UX 上建置 C 多緒應用程式	199		
C 多緒應用程式的建置 Script	200		
HP-UX C++	201		
在 HP-UX 上建置 C++ 應用程式.	201		
C++ 應用程式的建置 Script.	202		
HP-UX C++ 應用程式編譯及鏈結選項	203		
在 HP-UX 上建置 C++ 常式	205		
C++ 常式的建置 Script	209		
HP-UX C++ 常式編譯及鏈結選項.	210		
在 HP-UX 上建置 C++ 多緒應用程式	211		
C++ 多緒應用程式的建置 Script	212		
Micro Focus COBOL	213		
在 HP-UX 上架構 Micro Focus COBOL 編譯器	213		
在 HP-UX 上建置 Micro Focus COBOL 應用程式.	214		
Micro Focus COBOL 應用程式的建置 Script	216		
HP-UX Micro Focus COBOL 應用程式編譯及鏈結選項	216		
在 HP-UX 上建置 Micro Focus COBOL 常式	217		
Micro Focus COBOL 常式的建置 Script	219		
HP-UX Micro Focus COBOL 常式編譯及鏈結選項.	220		
第 9 章 Linux	221		
Linux C	221		
在 Linux 上建置 C 應用程式	221		
C 應用程式的建置 Script	223		
Linux C 應用程式編譯及鏈結選項	224		
在 Linux 上建置 C 常式	225		
C 常式的建置 Script	228		
Linux C 常式編譯及鏈結選項	229		
在 Linux 上建置 C 多緒應用程式	230		
C 多緒應用程式的建置 Script	231		
Linux C++	232		
在 Linux 上建置 C++ 應用程式	232		
C++ 應用程式的建置 Script.	233		
Linux C++ 應用程式編譯及鏈結選項.	234		
在 Linux 上建置 C++ 常式.	235		
C++ 常式的建置 Script	239		
Linux C++ 常式編譯及鏈結選項	239		
		第 10 章 Solaris	243
		Forte C	243
		在 Solaris 上建置 C 應用程式.	243
		C 應用程式的建置 Script	245
		Solaris C 應用程式編譯及鏈結選項	246
		在 Solaris 上建置 C 常式	248
		C 常式的建置 Script	251
		Solaris C 常式編譯及鏈結選項.	252
		在 Solaris 上建置 C 多緒應用程式	253
		C 多緒應用程式的建置 Script	255
		Forte C++	255
		在 Solaris 上建置 C++ 應用程式.	256
		C++ 應用程式的建置 Script.	257
		Solaris C++ 應用程式編譯及鏈結選項	258
		在 Solaris 上建置 C++ 常式	260
		C++ 常式的建置 Script	263
		Solaris C++ 常式編譯及鏈結選項.	264
		在 Solaris 上建置 C++ 多緒應用程式	265
		C++ 多緒應用程式的建置 Script	267
		Micro Focus COBOL	267
		在 Solaris 上架構 Micro Focus COBOL 編譯器	267
		在 Solaris 上建置 Micro Focus COBOL 應用程式	268
		Micro Focus COBOL 應用程式的建置 Script	270
		Solaris Micro Focus COBOL 應用程式編譯及鏈結選項	270
		在 Solaris 上建置 Micro Focus COBOL 常式	271
		Micro Focus COBOL 常式的建置 Script	273
		Solaris Micro Focus COBOL 常式編譯及鏈結選項	273
		第 11 章 Windows 作業系統	275
		WCHARTYPE CONVERT 前置編譯選項	275
		物件連結和內嵌資料庫 (OLE DB) 表格函數	276
		Windows Management Instrumentation (WMI)	277
		Microsoft Visual Basic	277
		利用 Visual Basic 建置 ADO 應用程式	277
		利用 Visual Basic 建置 RDO 應用程式	280
		利用 Visual Basic 的物件連結和內嵌 (OLE) 自動化	282

Microsoft Visual C++	283
利用 Visual C++ 建置 ADO 應用程式	283
利用 Visual C++ 的物件連結和內嵌 (OLE) 自動化	285
在 Windows 上建置 C/C++ 應用程式	286
C/C++ 應用程式的批次檔	287
Windows C/C++ 應用程式編譯及鏈結選項	288
在 Windows 上建置 C/C++ 常式	290
C/C++ 常式的批次檔	293
Windows C/C++ 常式編譯及鏈結選項	294
IBM VisualAge COBOL	296
在 Windows 上架構 IBM COBOL 編譯器	296
在 Windows 上建置 IBM COBOL 應用程式	297
IBM COBOL 應用程式的批次檔	298
Windows IBM COBOL 應用程式編譯及鏈結選項	299
在 Windows 上建置 IBM COBOL 常式	299
IBM COBOL 常式的批次檔	301
Windows IBM COBOL 常式編譯及鏈結選項	301
Micro Focus COBOL	302
在 Windows 上架構 Micro Focus COBOL 編譯器	302
在 Windows 上建置 Micro Focus COBOL 應用程式	303
Micro Focus COBOL 應用程式的批次檔	305
Windows Micro COBOL 應用程式編譯及鏈結選項	305
在 Windows 上建置 Micro Focus COBOL 常式	306
Micro Focus COBOL 常式的批次檔	307
Windows Micro COBOL 常式編譯及鏈結選項	308
物件 REXX	308
在 Windows 上建置 Object REXX 應用程式	308

第 4 篇 附錄與後記 311

附錄 A. DB2 Universal Database 技術資訊	313
DB2 Universal Database 技術資訊概觀	313
DB2 文件的 FixPak	313
DB2 技術資訊的種類	313
從 PDF 檔案列印 DB2 書籍	321
訂購 DB2 印刷書籍	321
存取線上說明	322
從瀏覽器存取「DB2 資訊中心」以尋找主題	324
從管理工具存取「DB2 資訊中心」以尋找產品資訊	325
直接從「DB2 HTML 文件 CD」檢視線上技術文件	327
更新已安裝於您的機器上的 HTML 文件	327
將檔案從 DB2 HTML 文件 CD 複製到 Web 伺服器	329
使用 Netscape 4.x 來搜尋 DB2 文件的疑難排解	329
搜尋 DB2 文件	330
線上 DB2 疑難排解資訊	331
協助工具	332
鍵盤輸入及導引	332
容易使用的顯示	332
替代的警戒提示	332
與輔助技術相容	333
容易使用的文件	333
DB2 教學指導	333
從瀏覽器存取的「DB2 資訊中心」	334
附錄 B. 注意事項	337
商標	340
索引	343
洽詢 IBM	349
產品資訊	349

關於本書

Application Development Guide 一書共三冊，它說明關於撰寫程式碼、除錯、建置和執行 DB2 應用程式時您必須知道的資訊：

- *Application Development Guide: Programming Client Applications* 包含您要撰寫在 DB2 從屬站執行的獨立式 DB2 應用程式時必須知道的資訊。它包括有關下列各項的資訊：
 - DB2 支援的程式設計介面。關於 DB2 Developer's Edition、支援的程式設計介面、建立 Web 應用程式的機能和 DB2 提供的程式設計特性 (例如常式和觸發函式)，均有提供其高階說明。
 - DB2 應用程式應該遵循的一般結構。有提供關於如何維護資料庫中的資料值和關係的建議，有說明授權注意事項，並提供關於如何測試及除錯應用程式的資訊。
 - 內含的 SQL，包含動態及靜態。有說明內含的 SQL 的一般注意事項，以及在 DB2 應用程式中使用靜態和動態 SQL 時會碰到的一些特殊議題。
 - 支援的主電腦及解譯語言，例如 C/C++、COBOL、Perl 及 REXX，以及如何在以這些語言撰寫的應用程式中使用內含的 SQL。
 - Java (包括 JDBC 和 SQLj) 及關於建置使用 WebSphere Application Server 的 Java 應用程式的注意事項。
 - DB2 Server 的 IBM OLE DB Provider。有提供關於 OLE DB 服務、元件及內容的 IBM OLE DB Provider 支援的一般資訊。也有提供關於對 ActiveX Data Object (ADO) 使用 OLE DB 介面的 Visual Basic 和 Visual C++ 應用程式的特定資訊。
 - 國家語言支援議題。說明一般主題，諸如對照順序、字碼頁和語言環境的衍生及字元轉換。亦說明較特定的議題，諸如 DBCS 字碼頁、EUC 字集和適合日文和繁體中文 EUC 和 UCS-2 環境的議題。
 - 交易管理。說明適合執行多位置更新的應用程式及執行並行交易的應用程式的議題。
 - 分割的資料庫環境中的應用程式。說明在分割的資料庫環境中的引導式 DSS、本端略過、緩衝插入和疑難排解應用程式。
 - 常用的應用程式技術。有提供關於如何使用已產生的直欄和身分直欄、已宣告的暫時表格以及如何使用儲存點來管理交易等資訊。
 - 支援使用於內含的 SQL 應用程式中的 SQL 陳述式。

- 存取主電腦和 iSeries 環境的應用程式。說明專屬於內含的 SQL 應用程式 (它們存取主電腦及 iSeries 環境) 的議題。
- EBCDIC 二進位對照的擬態。
- *Application Development Guide: Programming Server Applications* 包括您必須知道的伺服器端程式設計，包括常式、大型物件、使用者定義類型及觸發函式。它包括有關下列各項的資訊：
 - 常式 (儲存程序、使用者定義的函數及方法)，包括：
 - 常式效能、機密保護、檔案庫管理注意事項及限制。
 - 登錄及撰寫常式，包括 CREATE 陳述式及除錯。
 - 程序參數模式及參數處理。
 - 程序結果集。
 - UDF 特性，包括 scratchpad 及純量和表格函數。
 - SQL 程序，包括除錯及狀況處理。
 - 參數樣式、授權及外部常式的連結。
 - C、Java 和 OLE 自動化常式的語言特定注意事項。
 - 呼叫常式
 - 函數選取及傳遞特殊類型和 LOB 至函數。
 - 字碼頁及常式。
 - 大型物件，包括 LOB 使用情況及定位器、參考變數和 CLOB 資料。
 - 使用者定義的特殊類型，包括完全符合、定義及捨棄 UDT、建立含結構化類型的表格、對特定的應用程式使用特殊類型及類型化表格、操作特殊類型和它們之間的強制轉型，以及執行特殊類型的比較和指定，包括特殊類型化直欄的 UNION 作業。
 - 使用者定義的結構化類型，包括儲存案例及個體化、結構化類型階層、定義結構化類型行為、方法的動態分派、比較、強制轉型及建構子函數，和結構化類型的 mutator 及 observer 方法。
 - 類型化表格，包括建立、捨棄、替代、儲存物件、定義系統產生的物件識別字以及物件識別字直欄的限制。
 - 參考類型，包括類型化表格中各物件之間的關係、與參考資料的語義關係以及參照整合性對限定範圍的參照。
 - 類型化表格和類型化概略表，包括以直欄類型表示結構化類型、轉換函數及轉換群組、主語言程式對映及結構化類型主變數。
 - 觸發函式，包括 INSERT、UPDATE 及 DELETE 觸發函式、有參照限制的交談、建立準則、顆粒狀、啟動時間、轉移變數和表格、觸發動作、多重觸發動作以及觸發函式、限制和常式之間的協同作用。

- 應用程式開發手冊：建置及執行應用程式 包含要在 DB2 支援的作業系統上建置及執行 DB2 應用程式必須知道的資訊：
 - AIX
 - HP-UX
 - Linux
 - Solaris
 - Windows

它包括有關下列各項的資訊：

- 如何設定應用程式開發環境，包括 Java 及 SQL 程序的特定指示、如何設定範例資料庫以及如何從舊版 DB2 移轉應用程式。
- 用來建置應用程式的 DB2 支援的伺服器及軟體，包括支援的編譯器和直譯器。
- DB2 範例程式檔、make 檔、建置檔及錯誤檢查公用程式檔案。
- 如何建置及執行 Java applet、應用程式和常式。
- 如何建置及執行 SQL 程序。
- 如何建置及執行 C/C++ 應用程式及常式。
- 如何建置及執行 IBM 和 Micro Focus COBOL 應用程式及常式。
- 如何在 AIX 及 Windows 上建置及執行 REXX 應用程式。
- 如何在 Windows 上使用 Visual Basic 及 Visual C++ 來建置及執行含有 ActiveX Data Objects (ADO) 的應用程式。
- 如何在 Windows 上使用 Visual C++ 來建置及執行含有遠端資料物件的應用程式。

第 1 篇 應用程式開發環境

第 1 章 DB2 支援

DB2 Application Development Client	3	用於開發及執行應用程式的 Linux 支援軟體	11
資料庫管理程式案例	5	用於開發及執行應用程式的 Solaris 支援軟體	13
DB2 支援的伺服器	7	用於開發及執行應用程式的 Windows 支援軟體	14
用於開發及執行應用程式的 DB2 支援軟體	8		
依平台支援的軟體	9		
用於開發及執行應用程式的 AIX 支援軟體	9		
用於開發及執行應用程式的 HP-UX 支援軟體	10		

這一冊的 Application Development Guide 說明 DB2 所支援的應用程式開發。提供了您要用來開發 DB2 應用程式所需環境的設置資訊，以及關於在這個環境中編譯、鏈結與執行這些應用程式的逐步指示。並說明如何在下列平台中使用 DB2 Application Development (DB2 AD) Client for DB2 Universal Database 版本 8 建置應用程式：

- AIX
- HP-UX
- Linux
- Solaris 作業環境
- Windows 作業系統

DB2 Application Development Client

DB2 Application Development (DB2 AD) Client 提供開發應用程式所需的工具及環境，存取施行 DRDA 的 DB2 伺服器及應用程式伺服器。

您可以使用安裝的 DB2 AD Client 來建置及執行 DB2 應用程式。也可以在這些 DB2 從屬站上執行 DB2 應用程式：

- DB2 Run-Time Client
- DB2 Administration Client

本書說明要使用於各平台的 DB2 AD Client，包括下列各項：

- **C/C++、COBOL 及 Fortran 的前置編譯器**，(前提是該平台支援語言)。
- **內嵌式 SQL 應用程式支援**，包括程式設計檔案庫、併入檔及程式碼範例。
- **DB2 Call Level Interface (DB2 CLI) 應用程式支援**，包括程式設計檔案庫、併入檔及程式碼範例，以開發易於移轉到 ODBC 及使用 ODBC SDK 編譯的應用程式。您可以從 Microsoft for Windows 作業系統，及許多其它支援平台的不

同廠商處取得 ODBC SDK。就 Windows 作業系統而言，DB2 從屬站包含一個 ODBC 驅動程式，該驅動程式支援使用 Microsoft ODBC SDK 開發的應用程式。就所有其它平台而言，DB2 從屬站包含一個選擇性安裝的 ODBC 驅動程式，該驅動程式支援可以使用該平台的 ODBC SDK 開發的應用程式。僅有 DB2 Clients for Windows 作業系統含有 ODBC 驅動程式管理程式。

- **DB2 Java Enablement**，包括 DB2 Java Database Connectivity (DB2 JDBC) 支援 (可用於開發 Java 應用程式及 Applet)，及 DB2 內嵌式 SQL for Java (DB2 SQLJ) 支援 (可用於開發 Java 內嵌式 SQL 應用程式及 Applet)。
- **Java Development Kit (JDK)**。IBM for AIX 的 JDK 1.3.1 及 Java Runtime Environment (JRE) 1.3.1、for Linux 及 for Windows 的 IBM Developer Kit 及 Runtime Environment 版本 1.3.1、for HP-UX 的 HP-UX Software Developer's Kit 及 Runtime Environment 1.3.1，以及 Sun Microsystems for Solaris 的 Java Development Kit 1.3.1。除了 for Solaris 外，如果選取任何需要 Java 執行的元件進行安裝，將安裝每一個作業系統的個別 JDK。若未選取任一個，仍可選取 JDK 來進行安裝。在 Solaris 上，使用者必須安裝 DB2 所提供的 JDK 1.3.1。
- **Windows 作業系統上的 ActiveX Data Objects (ADO) 及「物件連結和內嵌 (OLE) 自動化常式」(UDF 及儲存程序)**，含有在 Microsoft Visual Basic 及 Microsoft Visual C++ 中執行的程式碼範例。此外，還有在 Microsoft Visual Basic 中已施行且含 Remote Data Objects (RDO) 的程式碼範例。
- **物件連結和內嵌資料庫 (OLE DB) 表格函數**，位於 Windows 作業系統上。
- **DB2 開發中心**，一種圖形式應用程式，用以支援常式 (儲存程序及使用者定義的函數) 及結構化類型的迅速開發。「開發中心」提供了支援整個 DB2 系列 (其範圍從工作站到 z/OS) 的單一開發環境。您可以啟動「開發中心」作為獨立式應用程式，或是從 DB2 Universal Database 中心 (如控制中心、命令中心或作業中心) 啟動它。「開發中心」是利用 Java 來實施的，而且所有資料庫連線都是藉由使用 Java Database Connectivity (JDBC) API 來管理的。「開發中心」同時也提供「DB2 開發外掛程式」，供下列每一個開發環境使用：
 - Microsoft Visual C++ 版本 6
 - Microsoft Visual Basic 版本 6
 - Microsoft Visual InterDev 版本 6
- **交談式 SQL** 經由「命令中心」或「命令行處理器 (CLP)」，產生 SQL 陳述式的原型，或對資料庫執行特別查詢。
- **一組記錄的 API**，可讓其它應用程式開發工具直接在它們的產品內執行 DB2 的前置編譯器支援。例如，AIX 上的 IBM COBOL 使用這個介面。您可以從這個匿名 FTP 網站的 PDF 檔 (prepapi.pdf) 取得「前置編譯器服務 API」組的相關資訊：

<ftp://ftp.software.ibm.com/ps/products/db2/info/>

- **SQL92 及 MVS 一致旗號控制**，它會識別應用程式中不符合 ISO/ANSI SQL92 Entry Level 標準的內嵌式 SQL 陳述式，或識別 DB2 for z/OS 及 OS/390 不支援的內嵌式 SQL 陳述式。如果您將工作站上所開發的應用程式移轉到另一個平台，「旗號控制」會經由顯示語法不相容，來節省您的時間。

相關參照:

- *Command Reference* 中的『PRECOMPILE Command』
- 第 9 頁的『用於開發及執行應用程式的 AIX 支援軟體』
- 第 10 頁的『用於開發及執行應用程式的 HP-UX 支援軟體』
- 第 11 頁的『用於開發及執行應用程式的 Linux 支援軟體』
- 第 13 頁的『用於開發及執行應用程式的 Solaris 支援軟體』
- 第 14 頁的『用於開發及執行應用程式的 Windows 支援軟體』

資料庫管理程式案例

DB2[®] 支援在同一台機器上有多個資料庫管理程式案例。資料庫管理程式案例具有它自己的架構檔、目錄及資料庫。

每一個資料庫管理程式案例可以管理數個資料庫。不過給定的資料庫僅屬於一個案例。下圖顯示此關係。

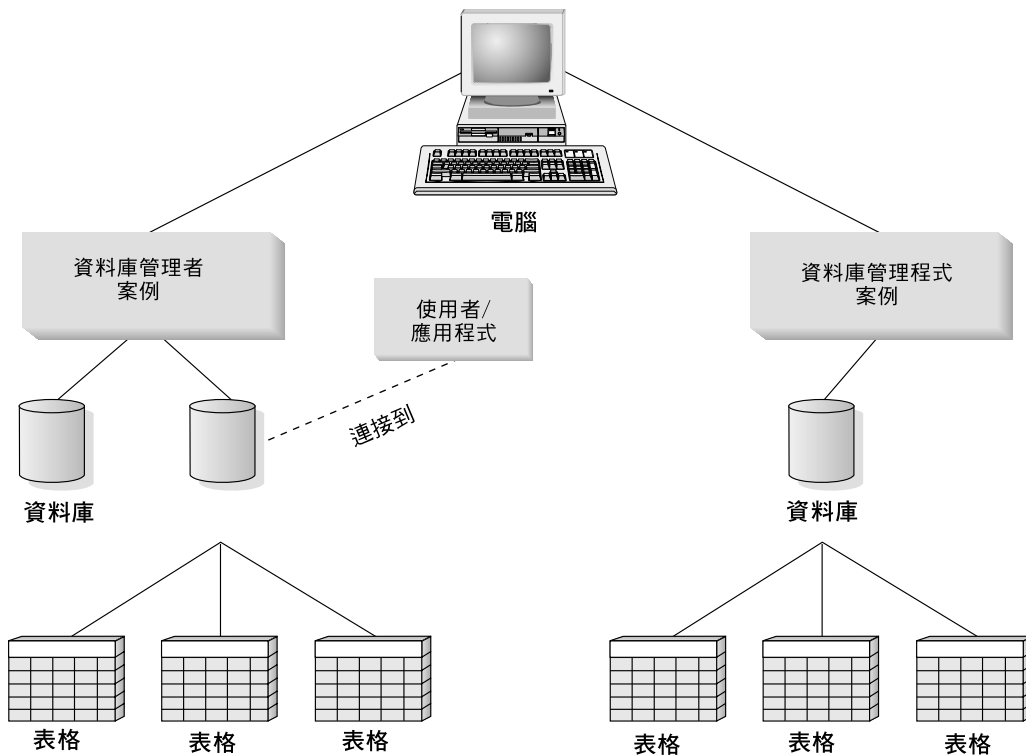


圖 1. 資料庫管理程式案例

資料庫管理程式案例使您能夠在同一台機器上具有多個資料庫環境。例如，您可以同時擁有一個用於開發的資料庫管理程式案例及另一個用於生產的案例。

使用 UNIX[®] 伺服器，您就可以在不同資料庫管理程式案例上，具有不同的 DB2 版本。例如，您可以讓某一個資料庫管理程式案例執行 DB2 Universal Database 版本 7.1，並讓另一個執行 DB2 Universal Database 版本 8.1。在 DB2 版本 8 之前，版本層次內僅支援一個版次及 FixPak 層次。例如，DB2 版本 7.1 及 DB2 版本 7.2 不能並存於 UNIX 伺服器中。使用 DB2 版本 8，可以有多個 FixPak 層次同時存在於同一 UNIX 伺服器。

若使用 Windows[®] 伺服器，您必須在每一個資料庫管理者案例中使用相同的 DB2 版本、版次及 FixPak 層次。您不能讓一個資料庫管理程式案例執行 DB2 Universal Database 版本 7.1，並讓另一個案例執行 DB2 Universal Database 版本 8.1。

您需要知道每一個您使用的案例的下列項目：

案例名稱

就 UNIX 平台而言，這是指您在建立資料庫管理程式案例時指定的有效使用者名稱。

就 Windows 作業系統而言，這是最多 8 個字元的英數字串。在安裝期間內，會建立一個名為 DB2 的案例。

案例目錄

案例位在其中的起始目錄。

就 UNIX 平台而言，案例目錄是 \$HOME/sqllib，其中 \$HOME 是案例擁有者的起始目錄。

就 Windows 作業系統而言，案例目錄為 %DB2PATH%\instance_name。變數 %DB2PATH% 決定 DB2 的安裝所在。%DB2PATH% 的預設安裝值是 \Program Files\IBM\SQLLIB，所以根據安裝 DB2 的磁碟機，%DB2PATH% 將指向 drive:\Program Files\IBM\SQLLIB，除非預設值遭到變更。

Windows 伺服器上的案例路徑是依據下列之一所建立的：

```
%DB2PATH%\%DB2INSTANCE%
```

(例如，C:\Program Files\IBM\SQLLIB\DB2)

或是，如果定義 DB2INSTPROF：

```
%DB2INSTPROF%\%DB2INSTANCE%
```

(例如，C:\PROFILES\DB2)

DB2INSTPROF 環境變數是用在 Windows 伺服器上，以支援在從屬站機器僅具有唯讀存取權的網路磁碟機上執行 DB2。假如是這樣的話，會設定 DB2 指向 drive:\Program Files\IBM\SQLLIB，而且設定 DB2INSTPROF 指向本端路徑 (例如，C:\PROFILES)，該路徑會包含全部案例特定資訊如目錄和架構，因為 DB2 對這些檔案需要具有更新權。

DB2 支援的伺服器

您可以使用 DB2 AD 從屬站來開發將在特定作業系統上執行的應用程式。不過，應用程式可以在下列作業系統伺服器存取遠端資料庫：

- DB2 for AIX
- DB2 for HP-UX
- DB2 for Linux
- DB2 for OS/2
- DB2 for NUMA-Q

- DB2 for Solaris
- DB2 for Windows NT/2000/XP/.NET 伺服器
- 遵守 DRDA 的應用系統伺服器，如：
 - DB2 for z/OS 及 OS/390
 - DB2 for AS/400 及 iSeries
 - DB2 for VSE & VM (先前稱為 SQL/DS for VM 及 VSE)
 - 遵守 DRDA 的應用系統伺服器，它們來自資料庫廠商而非 IBM。

註：

1. DB2 版本 8 HP-UX 64 位元伺服器不支援執行 DB2 版本 7 64 位元本端應用程式。
2. DB2 for OS/2 不適用於 DB2 版本 8
3. DB2 for NUMA-Q 在 PTX 作業系統上執行，而且僅適用於 DB2 版本 7
4. DB2 版本 8 Windows 64 位元伺服器僅支援針對 SQL 要求的 DB2 版本 6 及版本 7 32 位元從屬站連線。不支援版本 7 64 位元從屬站連線。

用於開發及執行應用程式的 DB2 支援軟體

DB2 版本 8 支援下列作業系統的編譯器、直譯器及相關軟體：

- AIX
- HP-UX
- Linux
- Solaris
- Windows 作業系統

DB2 支援這些作業系統每一個的 32 位元版本及 64 位元版本。在大部份情況中，於這些作業系統的 32 位元及 64 位元環境中建置應用程式會有差異。不過，DB2 支援在所有支援的 64 位元作業系統環境 (Linux IA64 除外) 上執行 32 位元應用程式及常式 (儲存程序及使用者定義的函數)。

已提供給這些作業系統每一個的編譯器資訊假定您將使用該作業系統的 DB2 前置編譯器，而非使用可能已內建在所列出的編譯器之一的前置編譯器支援。

若需最新的 DB2 編譯器資訊及相關的軟體更新，請探訪下列 DB2 應用程式開發網站：

<http://www.ibm.com/software/data/db2/udb/ad>

請注意，下列關於軟體支援的要點：

- **Fortran 和 REXX**。DB2 不會增強 Fortran 和 REXX 的特性超過 DB2 Universal Database 版本 5.2 中的這些語言的支援層次。
- **Perl**。列印時，Perl Database Interface (Perl DBI) 版本 0.93 或更新版本的 DB2 UDB 驅動程式 (DBD::DB2) 版次 0.76 適用於 AIX、HP-UX、Linux、Solaris 及 Windows。您可以從下列網址下載最新的驅動程式：

<http://www.ibm.com/software/data/db2/perl>

- **PHP**。PHP 可以作為從 Web 應用程式存取 DB2 的方法。PHP 是 HTML 內嵌式跨平台伺服器端 Script 語言。它支援使用統一 ODBC 存取方法對 DB2 的存取，在該方法中，用戶級的 PHP 使用 ODBC 呼叫與 DB2 通信。與標準的 ODBC 不同，使用「統一 ODBC」方法，通信會直接到達 DB2 CLI 層，而不透過 ODBC 層。如需 PHP 與 DB2 搭配使用的其它資訊，請搜尋 DB2 支援網站：

www.ibm.com/software/data/db2/udb/winos2unix/support

相關參照:

- 第 9 頁的『用於開發及執行應用程式的 AIX 支援軟體』
- 第 10 頁的『用於開發及執行應用程式的 HP-UX 支援軟體』
- 第 11 頁的『用於開發及執行應用程式的 Linux 支援軟體』
- 第 13 頁的『用於開發及執行應用程式的 Solaris 支援軟體』
- 第 14 頁的『用於開發及執行應用程式的 Windows 支援軟體』

依平台支援的軟體

用於開發及執行應用程式的 AIX 支援軟體

DB2 for AIX 支援下列作業系統：

AIX/6000

具有維護層次 9 的版本 4.3.3，以及更新版本

具有維護層次 1 的版本 5.1.0，以及更新版本

DB2 for AIX 支援下列程式設計語言及編譯器：

C IBM C for AIX 版本 5.0

C++ IBM VisualAge C++ 版本 5.0

COBOL

IBM COBOL Set for AIX 版本 1.1

Micro Focus COBOL Server Express 版本 2.0.10

Fortran

IBM XL Fortran for AIX 版本 4.1 (僅適用於 32 位元) 及 5.1.0 (32 位元及 64 位元)

Java IBM 適用於 AIX 的 Java Development Kit (JDK) 版本 1.3.1 及 Java Runtime Environment (JRE) 版本 1.3.1 (DB2 必須安裝的項目)

註: 如果任何選取的安裝元件需要 Java 執行，則 DB2 將安裝這個 JDK。若未選取任一個，仍可選取 JDK 來進行安裝。

Perl Perl 5.004_04 或更新版本，DBI 0.93 或更新版本

REXX IBM AIX REXX/6000 AISPO 產品編號：5764-057
IBM Object REXX for AIX 版本 1.1
REXXSAA 4.00

註: REXX 支援僅適用於 32 位元

如需 DB2 for AIX 軟體支援更新，請探訪 DB2 應用程式開發網站：

<http://www.ibm.com/software/data/db2/udb/ad>

相關參照:

- 第 8 頁的『用於開發及執行應用程式的 DB2 支援軟體』
- *DB2 伺服器快速入門* 中的『已分割之 DB2 伺服器的安裝需求 (AIX)』
- 第 146 頁的『AIX C 應用程式編譯及鏈結選項』
- 第 152 頁的『AIX C 常式編譯及鏈結選項』
- 第 158 頁的『AIX C++ 應用程式編譯及鏈結選項』
- 第 163 頁的『AIX C++ 常式編譯及鏈結選項』
- 第 174 頁的『AIX IBM COBOL 應用程式編譯及鏈結選項』
- 第 177 頁的『AIX IBM COBOL 常式編譯及鏈結選項』
- 第 182 頁的『AIX Micro COBOL 應用程式編譯及鏈結選項』
- 第 185 頁的『AIX Micro COBOL 常式編譯及鏈結選項』
- *DB2 伺服器快速入門* 中的『DB2 伺服器的安裝需求 (AIX)』

用於開發及執行應用程式的 HP-UX 支援軟體

DB2 for HP-UX 支援下列作業系統：

HP-UX

版本 11 及 11i

DB2 for HP-UX 支援下列程式設計語言及編譯器：

C HP C Compiler 版本 B.11.11.02

C++ HP aC++ 版本 A.03.31

COBOL

Micro Focus COBOL 版本 4.1

Fortran

HP Fortran/9000 版本 10.0

HP-UX F77 B.11.00.01

Java Hewlett-Packard 的 Software Developer's Kit 及 Runtime Environment 1.3.1 for HP-UX 11.0 和 11i PA-RISC (DB2 必須安裝的項目)

註： 如果任何選取的安裝元件需要 Java 執行，則 DB2 將安裝這個 SDK。
若未選取任一個，仍可選取 SDK 來進行安裝。

Perl Perl 5.004_04 或更新版本，DBI 0.93 或更新版本

如需 DB2 for HP-UX 軟體支援更新，請探訪 DB2 應用程式開發網站：

<http://www.ibm.com/software/data/db2/udb/ad>

相關參照：

- 第 8 頁的『用於開發及執行應用程式的 DB2 支援軟體』
- *DB2 伺服器快速入門* 中的『已分割之 DB2 伺服器的安裝需求 (HP-UX)』
- 第 192 頁的『HP-UX C 應用程式編譯及鏈結選項』
- 第 198 頁的『HP-UX C 常式編譯及鏈結選項』
- 第 203 頁的『HP-UX C++ 應用程式編譯及鏈結選項』
- 第 210 頁的『HP-UX C++ 常式編譯及鏈結選項』
- 第 216 頁的『HP-UX Micro Focus COBOL 應用程式編譯及鏈結選項』
- 第 220 頁的『HP-UX Micro Focus COBOL 常式編譯及鏈結選項』
- *DB2 伺服器快速入門* 中的『DB2 伺服器的安裝需求 (HP-UX)』

用於開發及執行應用程式的 Linux 支援軟體

DB2 for Linux for Intel x86 (32 位元架構) 支援下列作業系統環境：

Linux kernel 版本 2.4.9 或更新版本，glibc 版本 2.2.4 或更新版本，以及 rpm (必須安裝)

S/390 上的 DB2 for Linux 支援下列作業系統環境：

下列其中一項：

- RedHat v7.2
- S/390 上的 SuSE SLES-7 for Linux

DB2 for Linux for Intel x86 及 S/390 支援下列程式設計語言及編譯器：

C/C++ 對於 Linux on Intel：GNU/Linux gcc 及 g++ 版本 2.95.3 及 2.96

對於 Linux/390：GNU/Linux gcc 及 g++ 版本 2.95.3

請探訪 DB2 應用程式開發網站以取得支援 Linux on Intel 的 gcc 及 g++ 版本 3.0 以及支援 Linux/390 的未來版本資訊：

<http://www.ibm.com/software/data/db2/udb/ad>

Java 對於 Intel 上的 Linux：IBM Developer Kit 及 Runtime Environment for Linux，Java 2 Technology Edition，版本 1.3.1，32 位元版本 (DB2 必須安裝的項目)

對於 Linux/390：IBM zSeries Developer Kit for Linux，Java 2 Technology Edition (在 Sun 1.3.1 SDK 層次，且 DB2 必須安裝它)

註：如果任何選取的安裝元件需要 Java 執行，則 DB2 將安裝適當的 Developer Kit。若未選取任一個，仍可選取 Developer Kit 來進行安裝。

Perl Perl 5.004_04 或更新版本，DBI 0.93 或更新版本

REXX 對於 Linux on Intel：Object REXX Interpreter for Linux 版本 2.1

對於 Linux/390：Object REXX 2.2.0 for Linux/390

IA64 上的 DB2 for Linux 支援下列作業系統環境：

下列其中一項：

- Red Hat 7.2
- IA64 上的 SuSE SLES-7 for Linux

IA64 上的 DB2 for Linux 支援下列程式設計語言及編譯器：

C GNU/Linux gcc 版本 3.0.2

C++ GNU/Linux g++ 版本 3.0.2

Java IBM Developer Kit 及 Runtime Environment for Linux，Java 2 Technology Edition，版本 1.3.1，64 位元版本 (DB2 必須安裝的項目)。若要使用這個 JDK，您亦須安裝了 gcc 3.0.2 及 gcc3 libstdc++ 執行時間檔案庫。

註：如果任何選取的安裝元件需要 Java 執行，則 DB2 將安裝這個 Developer Kit。若未選取任一個，仍可選取 Developer Kit 來進行安裝。

Perl Perl 5.6

註：Linux IA64 不支援執行 DB2 32 位元應用程式或常式（儲存程序及使用者定義的函數）。

如需 DB2 for Linux 軟體支援更新，請探訪 DB2 應用程式開發網站：

<http://www.ibm.com/software/data/db2/udb/ad>

相關參照：

- 第 8 頁的『用於開發及執行應用程式的 DB2 支援軟體』
- *DB2 伺服器快速入門* 中的『已分割之 DB2 伺服器的安裝需求 (Linux)』
- 第 224 頁的『Linux C 應用程式編譯及鏈結選項』
- 第 229 頁的『Linux C 常式編譯及鏈結選項』
- 第 234 頁的『Linux C++ 應用程式編譯及鏈結選項』
- 第 239 頁的『Linux C++ 常式編譯及鏈結選項』
- *DB2 Personal Edition 快速入門* 中的『DB2 Personal Edition (Linux) 的安裝需求』
- *DB2 伺服器快速入門* 中的『DB2 伺服器的安裝需求 (Linux)』

用於開發及執行應用程式的 Solaris 支援軟體

DB2 for Solaris 支援下列作業系統：

Solaris

Solaris 7、Solaris 8 及 Solaris 9

DB2 for Solaris 支援下列程式設計語言及編譯器：

C Forte C 版本 5.0、6 及 6.1

註：這些編譯器版本通常稱為 SPARCompiler。

C++ Forte C++ 版本 5.0、6 及 6.1

註：這些編譯器版本通常稱為 SPARCompiler。

COBOL

Micro Focus COBOL Server Express 版本 2.0.10

Fortran

SPARCompiler Fortran 版本 4.2 及 5.0

Java Sun Microsystems 的 Java Development Kit (JDK) for Solaris 版本 1.3.1 (隨 DB2 一起出貨)

Perl Perl 5.004_04 或更新版本，DBI 0.93 或更新版本

如需 DB2 for Solaris 軟體支援更新，請探訪 DB2 應用程式開發網站：

<http://www.ibm.com/software/data/db2/udb/ad>

相關參照:

- 第 8 頁的『用於開發及執行應用程式的 DB2 支援軟體』
- *DB2 伺服器快速入門* 中的『已分割之 DB2 伺服器的安裝需求 (Solaris 作業環境)』
- 第 246 頁的『Solaris C 應用程式編譯及鏈結選項』
- 第 252 頁的『Solaris C 常式編譯及鏈結選項』
- 第 258 頁的『Solaris C++ 應用程式編譯及鏈結選項』
- 第 264 頁的『Solaris C++ 常式編譯及鏈結選項』
- 第 270 頁的『Solaris Micro Focus COBOL 應用程式編譯及鏈結選項』
- 第 273 頁的『Solaris Micro Focus COBOL 常式編譯及鏈結選項』
- *DB2 伺服器快速入門* 中的『DB2 伺服器的安裝需求 (Solaris)』

用於開發及執行應用程式的 Windows 支援軟體

DB2 for Windows 32 位元作業系統支援下列：

Microsoft Windows XP

Microsoft Windows .NET

Microsoft Windows 2000

Microsoft Windows NT

具有 Service Pack 6a 或更新的版本 4.0。

Microsoft Windows ME

Microsoft Windows 98

DB2 for Windows 32 位元作業系統支援下列程式設計語言：

Basic Microsoft Visual Basic 版本 4.2 及版本 5.0

Microsoft Visual Basic 6.0 Professional Edition

C/C++ Microsoft Visual C++ 版本 5.0 和 6.0

32 位元應用程式版的 Intel C++ Compiler 版本 5 或更新版本

COBOL

Micro Focus COBOL 版本 4.0.20

Micro Focus COBOL Net Express 版本 3.1.0

IBM VisualAge COBOL 版本 2.0

REXX IBM Object REXX for Windows NT/95 版本 1.1

如需取得 IBM Object REXX for Windows 的相關資訊，請探訪：

<http://www.ibm.com/software/ad/obj-rexx/>

Java IBM Developer Kit 及 Runtime Environment for Windows，Java 2 Technology Edition，版本 1.3.1，32 位元版本 (DB2 必須安裝的項目)

註： 如果任何選取的安裝元件需要 Java 執行，則 DB2 將安裝這個 Developer Kit。若未選取任一個，仍可選取 Developer Kit 來進行安裝。

Sun Microsystems 的 Java Development Kit (JDK) 1.3.1 for Win32

Perl Perl 5.004_04、DBI 0.93

Microsoft Windows Scripting Host

版本 5.1

DB2 for Windows 64 位元作業系統支援下列：

Microsoft Windows XP 64 位元版

Microsoft Windows .NET Server 64 位元版

DB2 for Windows 64 位元作業系統支援下列程式設計語言：

C/C++ Intel C++ Compiler for Itanium 版本 6.0

適用於 Intel Itanium 架構的 Microsoft C/C++ 編譯器

Java IBM Developer Kit 及 Runtime Environment for Windows，Java 2 Technology Edition，版本 1.3.1，64 位元版本 (DB2 必須安裝的項目)

註： 如果任何選取的安裝元件需要 Java 執行，則 DB2 將安裝這個 Developer Kit。若未選取任一個，仍可選取 Developer Kit 來進行安裝。

Microsoft Windows Scripting Host

版本 5.1

註: Windows .NET Server 包括下列全部:

- Windows .NET Web Server
- Windows .NET Standard Server
- Windows .NET Enterprise Server
- Windows .NET Datacenter Server

如需 DB2 for Windows 軟體支援更新, 請探訪 DB2 應用程式開發網站:

<http://www.ibm.com/software/data/db2/udb/ad>

相關參照:

- 第 8 頁的『用於開發及執行應用程式的 DB2 支援軟體』
- *DB2 伺服器快速入門* 中的『DB2 伺服器的安裝需求 (Windows)』
- 第 288 頁的『Windows C/C++ 應用程式編譯及鏈結選項』
- 第 294 頁的『Windows C/C++ 常式編譯及鏈結選項』
- 第 299 頁的『Windows IBM COBOL 應用程式編譯及鏈結選項』
- 第 301 頁的『Windows IBM COBOL 常式編譯及鏈結選項』
- 第 305 頁的『Windows Micro COBOL 應用程式編譯及鏈結選項』
- 第 308 頁的『Windows Micro COBOL 常式編譯及鏈結選項』
- *DB2 Personal Edition 快速入門* 中的『DB2 Personal Edition (Windows) 的安裝需求』
- *DB2 伺服器快速入門* 中的『已分割之 DB2 伺服器的安裝需求 (Windows)』

第 2 章 安裝

一般安裝資訊	17	Windows	32
設置應用程式開發環境	17	設置 Windows 應用程式開發環境	32
更新「資料庫管理程式架構檔」.	19	Java	35
設置 Java 環境	20	設置 Windows Java 環境	35
設定 SQL 程序環境	21	Windows Java 環境設定值	36
UNIX	22	SQL 程序	37
設置 UNIX 應用程式開發環境	22	設置 Windows SQL 程序環境	37
UNIX 環境變數設定值	23	範例資料庫	39
Java	24	設置範例資料庫	39
設置 UNIX Java 環境	24	建立範例資料庫	39
設置 AIX Java 環境	25	在主電腦或 AS/400 及 iSeries 伺服器上建 立範例資料庫	41
設置 HP-UX Java 環境	26	編目範例資料庫	42
設置 Linux Java 環境	26	連結範例資料庫公用程式	43
設置 Solaris Java 環境	28	移轉應用程式	44
SQL 程序	28	下一步要做什麼	52
設置 UNIX SQL 程序環境	28		
UNIX 預設 DB2 SQLROUTINE COMPILE COMMAND 值	30		

一般安裝資訊

關於 DB2 CLI 安裝資訊，請參閱 *CLI Guide and Reference*。

設置應用程式開發環境

爲了建置並執行 DB2 應用程式，您必須使用作業系統支援的其中一種程式設計語言的編譯器或直譯器。您必須設置 DB2 環境並架構它，以符合開發需求。隨後將有某些程序，可讓您從前一版 DB2 移轉 DB2 應用程式。此外，基於測試目的，您可能想要建立 DB2 範例資料庫。

先決條件:

最好先建立一個非 DB2 應用程式，確定 DB2 支援且您計劃使用之編譯器或直譯器的環境已設定妥當。然後，如果您遇到任何問題，請參閱您的編譯器或直譯器所附的文件。

將「應用程式開發從屬站」安裝在您正在使用的從屬站或伺服器工作站。如果要從遠端從屬站開發應用程式，請確定您的從屬站機器可以聯絡 DB2 資料庫伺服器常駐的機器。同時，亦請確定您的從屬站可以順利地連接資料庫。您可以使用命令行處理器 (CLP) 或從屬站架構輔助程式 (CCA) 來測試連線。

程序:

若要設置您的應用程式開發環境，請：

1. 除非預設值是可接受的，不然更新資料庫管理程式架構檔
2. 如果您將利用 DB2 CLI、Java 或 SQL 程序來設計程式，則必須先架構您的環境，然後再利用下列中的指示來執行平台特有的變更：
 - 設置 CLI 環境
 - 設置 Java 環境
 - 設置 SQL 程序環境
3. 利用下列中的指示來架構您的作業系統環境：
 - 設置 UNIX 環境
 - 設置 Windows 環境
4. 選用的：設置範例資料庫

相關概念:

- 第 5 頁的『資料庫管理程式案例』
- 第 44 頁的『移轉應用程式』

相關作業:

- 第 19 頁的『更新「資料庫管理程式架構檔」』
- *CLI Guide and Reference, Volume 1* 中的『Setting Up the CLI Environment』
- 第 20 頁的『設置 Java 環境』
- 第 21 頁的『設定 SQL 程序環境』
- 第 22 頁的『設置 UNIX 應用程式開發環境』
- 第 32 頁的『設置 Windows 應用程式開發環境』
- 第 39 頁的『設置範例資料庫』

相關參照:

- 第 3 頁的『DB2 Application Development Client』
- 第 7 頁的『DB2 支援的伺服器』
- 第 8 頁的『用於開發及執行應用程式的 DB2 支援軟體』
- 第 9 頁的『用於開發及執行應用程式的 AIX 支援軟體』
- 第 10 頁的『用於開發及執行應用程式的 HP-UX 支援軟體』
- 第 11 頁的『用於開發及執行應用程式的 Linux 支援軟體』
- 第 13 頁的『用於開發及執行應用程式的 Solaris 支援軟體』
- 第 14 頁的『用於開發及執行應用程式的 Windows 支援軟體』

更新「資料庫管理程式架構檔」

此檔案包含用於開發應用程式的重要設定值。

對於常式 (儲存程序及 UDF)，關鍵字 `KEEPFENCED` 具有預設值 `YES`。這會讓常式程序保持在作用中。如果您要開發常式，則可能要多測試幾次將相同的共用檔案庫載入。此預設設定可能會干擾檔案庫的重新載入。在開發常式時，最好將此關鍵字的值變更為 `NO`，然後在您準備好要載入最終版的共用檔案庫時，再將它變更回 `YES`。

對於安全執行緒常式，用來執行常式的程序在完成後仍會留在案例 (程序內僅有用來呼叫常式的執行緒才會終止)。基於這個理由，在開發新常式時，您應該將常式定義為非安全執行緒。然後，若適當的話，請啟用常式，以便在置於生產時，能在安全執行緒模式中執行。

註: 在舊版 DB2 中，`KEEPFENCED` 稱之為 `KEEPDARI`。

對於 Java 應用程式開發，您需要以安裝 Java Development Kit (JDK) 的路徑，更新 `JDK_PATH` 關鍵字。

註: 在舊版 DB2 中，`JDK_PATH` 稱之為 `JDK11_PATH`。

程序:

若要變更這些設定值，請輸入：

```
db2 update dbm cfg using <keyword> <value>
```

例如，若要將關鍵字 `KEEPFENCED` 設定為 `NO`，請輸入：

```
db2 update dbm cfg using KEEPFENCED NO
```

若要將 `JDK_PATH` 關鍵字設定為 `/home/db2inst/jdk13`，請輸入：

```
db2 update dbm cfg using JDK_PATH /home/db2inst/jdk13
```

若要檢視資料庫管理程式架構檔中的現行設定值，請輸入：

```
db2 get dbm cfg
```

註: 在 Windows，您需要在 DB2 命令視窗中輸入這些命令。

相關作業:

- 第 20 頁的『設置 Java 環境』

相關參照:

- *SQL Reference, Volume 2* 中的『CREATE FUNCTION statement』

- *SQL Reference, Volume 2* 中的『CREATE PROCEDURE statement』
- *Command Reference* 中的『GET DATABASE MANAGER CONFIGURATION Command』
- *Command Reference* 中的『RESET DATABASE MANAGER CONFIGURATION Command』

設置 Java 環境

您可以使用適合平台的 Java Development Kit (JDK) 開發 Java 程式，以存取 DB2 資料庫。JDK 包括 Java Database Connectivity (JDBC)，這是一種動態 SQL API for Java。

在 DB2 從屬站和伺服器上，所提供的 DB2 JDBC 支援是作為 Java Enablement 選項的一部份。有了這項支援，您可以建置和執行 JDBC 應用程式和 Applet。這些只含有動態 SQL，並使用 Java 呼叫介面來將 SQL 陳述式傳送到 DB2。

所提供的 DB2 Java 內嵌式 SQL (SQLJ) 支援是作為 DB2 AD Client 的一部份。除了 DB2 JDBC 支援外，有了 DB2 SQLJ 支援之後，您可建置和執行 SQLJ Applet 和應用程式。其中含有靜態 SQL，並使用連結到 DB2 資料庫的內嵌式 SQL 陳述式。

DB2 AD Client 提供的 SQLJ 支援包括：

- DB2 SQLJ 轉換程式 `sqlj`，以 Java 原始陳述式取代 SQLJ 程式中的內嵌式 SQL 陳述式，並產生序列設定檔，此設定檔含有關於 SQLJ 程式中的 SQL 作業之資訊。
- DB2 SQLJ 設定檔自訂程式 `db2profc`，它前置編譯儲存在序列設定檔中的 SQL 陳述式，自訂它們成為執行函數呼叫，以及在 DB2 資料庫中產生資料包。
- DB2 SQLJ 設定檔印表機 `db2profp`，它以純文字方式列印 DB2 自訂版的設定檔內容。

程序：

欲執行 DB2 Java 應用程式，您必須安裝及呼叫提供原生緒支援的 Java Virtual Machine (JVM)。欲執行使用原生緒的 Java 應用程式，您可以在命令中使用 `-native` 選項。例如，欲執行 Java 範例應用程式 `DbInfo.class`，您可以使用下列命令：

```
java -native DbInfo
```

您可以藉由將 `THREADS_FLAG` 環境變數設為 "native"，指定原生緒作為某些 Java Virtual Machine 的預設緒支援。本文件假設原生緒支援為預設值。請參閱您的 JVM 文件，取得在系統中設定原生緒為預設值的相關指示。

欲執行 DB2 Java Applet，您可以呼叫提供原生緒或綠色緒支援的 Java Virtual Machine。

確定上述項目已安裝且正常運作之後，您可以遵循下列之一中的步驟，設定特殊作業系統 Java 環境：

- 設置 UNIX Java 環境
- 設置 Windows Java 環境

關於最新 DB2 Java 應用程式開發更新組件，請探訪下列網頁：

<http://www.ibm.com/software/data/db2/java>

相關作業：

- 第 24 頁的『設置 UNIX Java 環境』
- 第 35 頁的『設置 Windows Java 環境』

相關參照：

- *Command Reference* 中的『db2profrc - DB2 SQLj Profile Customizer Command』
- *Command Reference* 中的『db2profp - DB2 SQLj Profile Printer Command』

設定 SQL 程序環境

需使用兩個 DB2 登錄變數來完成編譯器架構：

DB2_SQLROUTINE_COMPILER_PATH

指定為設定編譯器環境變數之 Script 的路徑名稱。

DB2_SQLROUTINE_COMPILE_COMMAND

指定 DB2 用來編譯為 SQL 程序而建立之 C 檔案的完整命令。

您可使用 db2set 命令，或可從「開發中心」中使用「SQL 儲存程序建置選項」對話框，來設定這些 DB2 登錄變數值。若使用「SQL 儲存程序建置選項」對話框，無需實際存取資料庫伺服器，亦無需重新啟動資料庫伺服器，即可使變更生效。

底下的步驟示範 db2set 命令。

先決條件：

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」及 DB2 支援的 C 或 C++ 編譯器。

程序：

若要架構 SQL 程序環境，請執行下列之一中的步驟，取決於您的平台而定：

- 設置 UNIX SQL 程序環境
- 設置 Windows SQL 程序環境

相關概念:

- *Administration Guide: Performance* 中的『DB2 registry and environment variables』

相關作業:

- 第 28 頁的『設置 UNIX SQL 程序環境』
- 第 37 頁的『設置 Windows SQL 程序環境』
- 第 127 頁的『建立 SQL 程序』
- 第 137 頁的『利用 CALL 陳述式呼叫儲存程序』
- 第 128 頁的『在 UNIX 上利用從屬站應用程式呼叫 SQL 程序』
- 第 129 頁的『在 Windows 上利用從屬站應用程式呼叫 SQL 程序』
- 第 134 頁的『分送編譯後的 SQL 程序』
- 第 135 頁的『重新連結 SQL 程序』

UNIX

關於 UNIX DB2 CLI 安裝資訊，請參閱 *CLI Guide and Reference*。

設置 UNIX 應用程式開發環境

您需要設定資料庫案例的環境變數。每一個資料庫管理程式案例都有兩個檔案，db2profile 及 db2cshrc，它們都是用來設定該案例環境變數的 Script。

程序:

執行適用於您正在使用之 Shell 的正確 Script：

對於 **bash** 或 **Korn shell** 而言：

```
. $HOME/sqlllib/db2profile
```

對 **C shell** 而言：

```
source $HOME/sqlllib/db2cshrc
```

其中 \$HOME 是案例擁有者的起始目錄。

如果您在 .profile 或 .login 檔中包括了這個命令，則在您登入時，命令就會自動執行。

如果您將使用 ODBC 或 DB2 CLI、Java 或 SQL 程序，請執行下列主題中的步驟：

- 設置 UNIX ODBC 環境
- 設置 UNIX Java 環境
- 設置 UNIX SQL 程序環境

相關概念:

- 第 23 頁的『UNIX 環境變數設定值』

相關作業:

- *CLI Guide and Reference, Volume 1* 中的『Setting Up the UNIX ODBC Environment』
- 第 24 頁的『設置 UNIX Java 環境』
- 第 28 頁的『設置 UNIX SQL 程序環境』

相關參照:

- 第 9 頁的『用於開發及執行應用程式的 AIX 支援軟體』
- 第 10 頁的『用於開發及執行應用程式的 HP-UX 支援軟體』
- 第 11 頁的『用於開發及執行應用程式的 Linux 支援軟體』
- 第 13 頁的『用於開發及執行應用程式的 Solaris 支援軟體』

UNIX 環境變數設定值

根據您所在的 UNIX® 平台，將在 db2profile (用於 bash 或 korn shell) 或 db2cshrc (用於 C shell) 中，設定下列環境變數值，而且這些檔案的呼叫將放置在 .profile (bash 或 korn shell) 或 .login (C shell) 檔。

AIX :

- PATH，包括數個包括 sqllib/bin 的 DB2® 目錄
- LIBPATH，包括目錄 sqllib/lib (請參閱底下備註)

HP-UX :

- PATH，包括數個包括 sqllib/bin 的 DB2 目錄
- SHLIB_PATH (32 位元及 64 位元) 或 LD_LIBRARY_PATH (64 位元)，包括目錄 sqllib/lib (請參閱底下備註)

Linux 及 Solaris :

- PATH，包括數個包括 sqllib/bin 的 DB2 目錄
- LD_LIBRARY_PATH，包括目錄 sqllib/lib (請參閱底下備註)

註: 如果您要在 64 位元 DB2 案例中執行本端 32 位元應用程式，則 LIBPATH、SHLIB_PATH 及 LD_LIBRARY_PATH 應該含有 sqllib/lib32 代替 sqllib/lib。

空白檔案 `sqllib/userprofile` 及 `sqllib/usercshrc` 是在案例建立期間建立的，以容許使用者放置他們自己的案例環境設定值。在任何 DB2 FixPak 或未來版本安裝的案例更新 (`db2iupdt`) 期間，均不會對這些檔案加以修改。如果您不想要 `db2profile` 或 `db2cshrc` Script 中的新環境設定值，可以使用相對應的「使用者」Script 來置換它們。在 `db2profile` 或 `db2cshrc` Script 結束時，即會呼叫這個「使用者」Script。在案例移轉期間 (`db2imigr`)，會將使用者 Script 複製過去，以便繼續使用您的環境修改。這些使用者 Script 僅在 DB2 版本 7 及以後的版本中可用。

相關作業:

- 第 22 頁的『設置 UNIX 應用程式開發環境』

Java

設置 UNIX Java 環境

爲了在具有 DB2 JDBC 支援的 UNIX 上執行 JDBC 和 SQLJ 程式，所以在資料庫管理程式檔 `db2profile` 和 `db2cshrc` 中有包括用來更新 Java 環境的命令。建立 DB2 案例之後，修改 `.bashrc`、`.profile` 及/或 `.cshrc`，使得：

1. `THREADS_FLAG` 設定爲 "native"。(僅限在 HP-UX、Linux 及 Solaris)
2. `CLASSPATH` 包括：
 - "." (現行目錄)
 - 檔案 `sqllib/java/db2java.zip`
 - 檔案 `sqllib/java/db2jcc.jar`

若要建置 SQLJ 程式，則也會更新 `CLASSPATH` 來包括下面檔案：

```
sqllib/java/sqlj.zip
```

若要執行 SQLJ 程式，則也會更新 `CLASSPATH` 來包括下面檔案：

```
sqllib/java/runtime.zip
```

程序:

若要執行 DB2 Java 常式 (儲存程序及 UDF)，您必須在伺服器上更新 DB2 資料庫管理程式架構，以便併入 JDK 安裝在該機器上的路徑。您可在伺服器命令行上輸入下列命令來執行此動作：

```
db2 update dbm cfg using JDK_PATH /home/db2inst/jdk13
```

其中 `/home/db2inst/jdk13` 是安裝 JDK 的路徑。

您可在伺服器上輸入下列命令，檢查 DB2 資料庫管理程式架構，以驗證 JDK_PATH 欄位的值是否正確：

```
db2 get dbm cfg
```

您可能需要將輸出重新導向至某檔案以方便檢視。JDK_PATH 欄位出現在靠近輸出起始處。

確定上述項目已安裝且正常運作之後，您可以遵循下列之一中的步驟，設定特定 UNIX 作業系統環境：

- 設置 AIX Java 環境
- 設置 HP-UX Java 環境
- 設置 Linux Java 環境
- 設置 Solaris Java 環境

相關作業:

- 第 25 頁的『設置 AIX Java 環境』
- 第 26 頁的『設置 HP-UX Java 環境』
- 第 26 頁的『設置 Linux Java 環境』
- 第 28 頁的『設置 Solaris Java 環境』
- 第 19 頁的『更新「資料庫管理程式架構檔」』

相關參照:

- *Command Reference* 中的『GET DATABASE MANAGER CONFIGURATION Command』
- *Command Reference* 中的『RESET DATABASE MANAGER CONFIGURATION Command』

設置 AIX Java 環境

程序:

欲在具有 DB2 JDBC 支援的 AIX 上建置 Java 應用程式，您需要：

1. IBM for AIX 的 Java Development Kit (JDK) 版本 1.3.1 及 Java Runtime Environment (JRE) 版本 1.3.1。
2. DB2 Java Enablement，隨附於適用 AIX 從屬站及伺服器的 DB2 Universal Database 版本 8 中。

相關概念:

- 第 101 頁的『Java 範例程式』
- 第 103 頁的『Java Applet 注意事項』

相關作業:

- 第 39 頁的『設置範例資料庫』

相關參照:

- 第 9 頁的『用於開發及執行應用程式的 AIX 支援軟體』

設置 HP-UX Java 環境

程序:

欲在具有 DB2 JDBC 支援的 HP-UX 上建置 Java 應用程式，您需要在您的開發機器上安裝及架構下列項目：

1. Hewlett-Packard 的 Software Developer's Kit 及 Runtime Environment 1.3.1 for HP-UX 11.0 和 11i PA-RISC。
2. DB2 Java Enablement，隨附於適用 HP-UX 從屬站及伺服器的 DB2 Universal Database 版本 8 中。

若要執行 Java 常式 (儲存程序及使用者定義的函數)，請確定共用檔案庫路徑與下列內容類似：

```
export SHLIB_PATH=$JAVADIR/jre/lib/PA_RISC:\
                $JAVADIR/jre/lib/PA_RISC/classic:\
                $HOME/sqllib/lib:\
                /usr/lib:$SHLIB_PATH
```

其中 \$JAVADIR 通常會設為 /opt/java1.3 (Java SDK 的預設位置)。

註: DB2 不支援常式使用 HotSpot Java Virtual Machine。

相關概念:

- 第 101 頁的『Java 範例程式』
- 第 103 頁的『Java Applet 注意事項』

相關作業:

- 第 39 頁的『設置範例資料庫』

相關參照:

- 第 10 頁的『用於開發及執行應用程式的 HP-UX 支援軟體』

設置 Linux Java 環境

程序:

欲在具有 DB2 JDBC 支援的 Linux 上建置 Java 應用程式，您需要在您的開發機器上安裝及架構下列項目：

1. 下列其中一項：
 - 對於 Intel 上的 Linux：IBM Developer Kit 及 Runtime Environment for Linux，Java 2 Technology Edition，版本 1.3.1，32 位元版本
 - 對於 Linux/390：IBM zSeries Developer Kit for Linux，Java 2 Technology Edition
 - 對於 IA64 上的 Linux：IBM Developer Kit 及 Runtime Environment for Linux，Java 2 Technology Edition，版本 1.3.1，64 位元版本
2. DB2 Java Enablement，隨附於適用 Linux 從屬站及伺服器的 DB2 Universal Database 版本 8 中。

若要執行 Java 儲存程序或使用者定義的函數，則 Linux 執行時間鏈結器必須能夠存取某些 Java 共用檔案庫，而且 DB2 必須能夠載入這些檔案庫及 Java 虛擬機器。既然執行此載入的程式是以 `setuid` 專用權執行，因此它僅會在 `/lib` 或 `/usr/lib` 中尋找相依檔案庫。

您可以將 Java 共用檔案庫的位置新增到 `/etc/ld.so.conf`，但是我們建議在 `/usr/lib` 中建立符號鏈結，以指向這些檔案庫。若是 IBM JDK 1.3，您需要指向 `libjava.so`、`libjvm.so` 及 `libhpi.so` 的符號鏈結。您可以用 `root` 身分執行下列命令，來建立符號鏈結：

```
cd /usr/lib
ln -fs JAVAHOME/jre/bin/libjava.so .
ln -fs JAVAHOME/jre/bin/classic/libjvm.so .
ln -fs JAVAHOME/jre/bin/libhpi.so .
```

其中 `JAVAHOME` 是 JDK 的基本目錄。如果 DB2 找不到這些檔案庫，則在嘗試執行 Java 常式時，您將得到 -4301 錯誤，而且在管理通知日誌中將有一則訊息，指出找不到檔案庫。

如果您決定將 Java 共用檔案庫的位置新增到 `/etc/ld.so.conf`，您必須以 `root` 身分執行下列命令，復新執行時間鏈結器快取：

```
bash# ldconfig
```

相關概念:

- 第 101 頁的『Java 範例程式』
- 第 103 頁的『Java Applet 注意事項』

相關作業:

- 第 39 頁的『設置範例資料庫』

相關參照:

- 第 11 頁的『用於開發及執行應用程式的 Linux 支援軟體』

設置 Solaris Java 環境

程序:

欲在具有 DB2 JDBC 支援的 Solaris 作業環境中建置 Java 應用程式，您必須在開發機器中安裝及架構下列各項：

1. Sun Microsystems 的 Java Development Kit (JDK) for Solaris 版本 1.3.1。
2. DB2 Java Enablement，隨附於適用 Solaris 從屬站及伺服器的 DB2 Universal Database 版本 8 中。

相關概念:

- 第 101 頁的『Java 範例程式』
- 第 103 頁的『Java Applet 注意事項』

相關作業:

- 第 39 頁的『設置範例資料庫』

相關參照:

- 第 13 頁的『用於開發及執行應用程式的 Solaris 支援軟體』

SQL 程序

設置 UNIX SQL 程序環境

SQL 程序需要使用您的編譯器架構，在伺服器上設定兩個環境變數：
DB2_SQLROUTINE_COMPILER_PATH 及
DB2_SQLROUTINE_COMPILE_COMMAND。

限制:

案例擁有者必須屬於隔離 ID 的主群組。

程序:

在您首次編譯儲存程序時，DB2 會產生一可執行的 Script 檔
\$HOME/sqlllib/function/routine/sr_cpath (包含有編譯器環境變數的預設值)。如果預設值不適合您的編譯器，請執行下列之一：

- 編輯 sr_cpath Script 檔。

- 設定 DB2_SQLROUTINE_COMPILER_PATH DB2 登錄變數，以包含其它可執行 Script 的完整路徑名稱 (此可執行 Script 可指定您需要的設定值)。

若安裝有 Application Development Client，則會提供預設編譯命令，它可至少用於每個伺服器平台所支援的編譯器之一：

AIX IBM C for AIX 版本 5.0

HP-UX

HP aC++ 版本 A.03.31

Linux

GNU/Linux g++

Solaris

Forte C++ 版本 5.0、6 及 6.1

若要使用其它編譯器，或自訂預設命令，您必須對 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登錄變數進行如下設定：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=<compile_command>
```

其中 <compile_command> 為 C 或 C++ 編譯命令，包括建立常式所必需的選項及參數。必須執行 db2start 命令，設定方可生效。

在編譯命令中，請使用關鍵字 SQLROUTINE_FILENAME 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。只有對 AIX，才使用關鍵字 SQLROUTINE_ENTRY 取代進入點名稱。在設定 IBM C for AIX 版本 5.0 之 DB2_SQLROUTINE_COMPILE_COMMAND 預設值的命令中，可以看到使用這兩個關鍵字的範例：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=xlc -I$HOME/sqllib/include \  
SQLROUTINE_FILENAME.c -bE:SQLROUTINE_FILENAME.exp -e SQLROUTINE_ENTRY \  
-o SQLROUTINE_FILENAME -L$HOME/sqllib/lib -ldb2
```

相關作業:

- 第 131 頁的『保留 SQL 程序的中間檔』
- 第 132 頁的『自訂 SQL 程序的前置編譯及連結選項』
- 第 133 頁的『備份及復置 SQL 程序』
- 第 127 頁的『建立 SQL 程序』
- 第 137 頁的『利用 CALL 陳述式呼叫儲存程序』
- 第 128 頁的『在 UNIX 上利用從屬站應用程式呼叫 SQL 程序』
- 第 134 頁的『分送編譯後的 SQL 程序』

相關參照:

- 第 30 頁的『UNIX 預設 DB2 SQLROUTINE COMPILE COMMAND 值』
- 第 9 頁的『用於開發及執行應用程式的 AIX 支援軟體』
- 第 10 頁的『用於開發及執行應用程式的 HP-UX 支援軟體』
- 第 11 頁的『用於開發及執行應用程式的 Linux 支援軟體』
- 第 13 頁的『用於開發及執行應用程式的 Solaris 支援軟體』

UNIX 預設 DB2 SQLROUTINE COMPILE COMMAND 值

以下命令可在支援的 UNIX 伺服器平台上設定 C 或 C++ 編譯器的 DB2_SQLROUTINE_COMPILE_COMMAND 預設值：

AIX 若要使用 IBM C for AIX 版本 5.0，請輸入：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=x1c -I$HOME/sql1lib/include \
SQLROUTINE_FILENAME.c -bE:SQLROUTINE_FILENAME.exp \
-e SQLROUTINE_ENTRY -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -ldb2
```

若要使用 IBM VisualAge C++ 5.0，請輸入：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=x1c -I$HOME/sql1lib/include \
SQLROUTINE_FILENAME.C -bE:SQLROUTINE_FILENAME.exp \
-e SQLROUTINE_ENTRY -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -ldb2
```

若尚未設定 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登錄變數，則此為預設編譯命令。

註：欲在 AIX 上編譯 64 位元 SQL 程序，請新增 -q64 選項到上述命令中。

若要使用 IBM VisualAge C++ for AIX 版本 5 漸進式編譯器，請輸入：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacb1d"
```

若您未在 vacb1d 命令之後指定配置檔，則 DB2 會在首次嘗試建立 SQL 程序時，建立以下預設配置檔：

```
$HOME/sql1lib/function/routine/sqlproc.icc
```

您可以在設定 DB2_SQLROUTINE_COMPILE_COMMAND 的 DB2 登錄值時，指定自己的配置檔：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacb1d \
$HOME/sql1lib/function/sqlproc.icc"
```

HP-UX

若要使用 HP C Compiler 版本 B.11.11.02，請輸入：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc +DAportable +ul -Aa +z \
-I$HOME/sql1lib/include -c SQLROUTINE_FILENAME.c; \
ld -b -o SQLROUTINE_FILENAME SQLROUTINE_FILENAME.o \
-L$HOME/sql1lib/lib -ldb2
```

若要使用 HP aC++ 版本 A.03.31，請輸入：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=aCC +DAportable +u1 +z -ext \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.C -b \  
-o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -ldb2
```

若尚未設定 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登錄變數，則此為預設編譯命令。

註：若要在 HP-UX 上編譯 64 位元 SQL 程序，請除去 +DAportable 選項，並將 +DA2.0W 選項新增至上述命令中。

Linux 若要使用 GNU/Linux gcc，請輸入：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc -fpic \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-shared -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -ldb2
```

若要使用 GNU/Linux g++，請輸入：

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=g++ -fpic \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.C \  
-shared -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -ldb2
```

若尚未設定 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登錄變數，則此為預設編譯命令。

Solaris

若要使用 Forte C 版本 5.0，請輸入：

```
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
```

若要使用 Forte C++ 版本 5.0，請輸入：

```
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
```

若尚未設定 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登錄變數，則此為預設編譯命令。

註：

1. 將編譯器選項 -xarch=v8plusa 新增至預設編譯器命令，是為了避免編譯器在與 libdb2.so 鏈結時，不產生有效可執行檔的問題。
2. 若要在 Solaris 上編譯 64 位元 SQL 程序，請除去 -xarch=v8plusa 選項，並將 -xarch=v9 選項新增至上述命令中。

相關作業:

- 第 28 頁的『設置 UNIX SQL 程序環境』

Windows

關於 Windows DB2 CLI 安裝資訊，請參閱 *CLI Guide and Reference*。

設置 Windows 應用程式開發環境

在 Windows XP、Windows .Net Server、Windows NT 或 Windows 2000 中安裝 DB2 AD Client 時，安裝程式會以環境變數 INCLUDE、LIB、PATH、DB2PATH 及 DB2INSTANCE 更新架構登錄。預設案例是 DB2。在 Windows 98 或 Windows ME 中安裝 DB2 AD Client 時，安裝程式會更新 autoexec.bat 檔。

您可以置換這些環境變數，設定機器或目前登入使用者的值。變更這些環境變數時需要特別小心。請不要更新 DB2PATH 環境變數。DB2INSTANCE 應該只定義在使用者環境層次。若使用 DB2INSTDEF 登錄變數，以便於未設定時可以定義預設的案例名稱，則不再需要 DB2INSTANCE。

程序:

若要置換環境變數設定值，請使用下列其中一項：

- Windows XP 控制台
- Windows .NET Server 控制台
- Windows NT 控制台
- Windows 2000 控制台
- Windows 98 或 Windows ME 命令視窗
- Windows 98 或 Windows ME autoexec.bat 檔

在命令中使用變數 %DB2PATH% 時，請以雙引號括住完整路徑，如 `set LIB="%DB2PATH%\lib";%LIB%`。此變數的預設安裝值是 `\Program Files\IBM\SQLLIB`，其中含有空格，所以如果不使用雙引號會造成錯誤。

此外，爲了執行 DB2 應用程式，您必須採取下列特定步驟：

- 在建置 C 或 C++ 程式時，您必須確定 INCLUDE 環境變數含有 %DB2PATH%\INCLUDE 作爲第一個目錄。

例如，Microsoft Visual C++ 編譯器環境設定檔 `Vc\bin\vcvars32.bat` 具有下列命令：

```
set INCLUDE=%MSVCDir%\INCLUDE;%MSVCDir%\...\ATL\INCLUDE;%INCLUDE%
```

欲以 DB2 使用此檔案，請先將會設定 %DB2PATH%\INCLUDE 路徑的 %INCLUDE%，從清單的尾端移到開頭，如下所示：

```
set INCLUDE=%INCLUDE%;%MSVCDir%\INCLUDE;%MSVCDir%\...\ATL\INCLUDE;
```

- 建置 Micro Focus COBOL 程式時，請設定 COBCPY 環境變數去指向 %DB2PATH%\INCLUDE\cobol_mf。
- 建置 IBM COBOL 程式時，請設定 SYSLIB 環境變數去指向 %DB2PATH%\INCLUDE\cobol_a。
- 確定 LIB 環境變數是指向 %DB2PATH%\lib，作法如下：

```
set LIB="%DB2PATH%\lib";%LIB%
```

註：若要從 32 位元環境啟用跨開發 64 位元應用程式，您必須確定 LIB 環境變數指向 %DB2PATH%\lib\Win64。這表示您必須從預設設定值變更它。依預設，環境變數指向 %DB2PATH%\lib，這是用於在 32 位元環境上開發 32 位元應用程式，或是在 64 位元環境上開發 64 位元應用程式。

- 確定遠端資料庫的伺服器上已設定 DB2COMM 環境變數。
- 確定伺服器及從屬站上已啟動機密保護服務，以進行 SERVER 身分驗證，視 CLIENT 身分驗證的層次而定。若要以手動方式啟動機密保護服務，請使用 NET START DB2NTSECSERVER 命令。

一般而言，唯一需要將機密保護服務設定成自動啟動的時機，就是工作站扮演連接至伺服器的 DB2 從屬站，而伺服器已架構成具有「從屬站身分驗證」的功能。若要自動啟動機密保護服務，請執行下列動作：

Windows NT

1. 按一下「開始」按鈕。
2. 按一下「設定」。
3. 按一下「控制台」。
4. 在「控制台」中，按一下「服務」。
5. 在「服務」視窗中，強調顯示「DB2 安全伺服器」。
6. 若未出現「已啟動」和「自動」設定，請按一下「啟動」。
7. 按一下「自動」。
8. 按一下「確定」。
9. 重新啟動機器，使設定生效。

Windows 2000 及 Windows .NET Server

1. 按一下「開始」按鈕。
2. 按一下「設定」。
3. 按一下「控制台」。

4. 按一下「管理工具」。
5. 按一下「服務」。
6. 在「服務」視窗中，強調顯示「DB2 安全伺服器」。
7. 如果沒有列出「已啟動」及「自動」設定，請在頂端的功能表上按一下「動作」。
8. 按一下「內容」。
9. 請確定您是在「一般事項」標籤中。
10. 從「啟動類型」下拉式功能表中，選取「自動」。
11. 按一下「確定」。
12. 重新啟動機器，使設定生效。

Windows XP

1. 按一下「開始」按鈕。
2. 按一下「設定」。
3. 按一下「控制台」。
4. 按一下「效能及維護」。
5. 按一下「管理工具」。
6. 按一下「服務」。
7. 在「服務」視窗中，強調顯示「DB2 安全伺服器」。
8. 如果沒有列出「已啟動」及「自動」設定，請在頂端的功能表上按一下「動作」。
9. 按一下「內容」。
10. 請確定您是在「一般事項」標籤中。
11. 從「啟動類型」下拉式功能表中，選取「自動」。
12. 按一下「確定」。
13. 重新啟動機器，使設定生效。

Windows XP、Windows .NET Server、Windows NT 或 Windows 2000 環境上的資料庫管理程式是當作服務來實施，因此，一旦啟動了服務，儘管可能發生問題，都不會傳回錯誤或警告。這表示執行 db2start 或 NET START 命令時，即使無法啟動通信子系統，亦不傳回任何警告。因此，使用者恆應檢查事件日誌或「DB2 管理通知」日誌，找出這些命令在執行時可能發生的錯誤。

如果您將使用 DB2 CLI、Java 或 SQL 程序，請繼續進行適當的作業：

- 設置 Windows CLI 環境
- 設置 Windows Java 環境

- 設置 Windows SQL 程序環境

相關作業:

- *CLI Guide and Reference, Volume 1* 中的『Setting Up the Windows CLI Environment』
- 第 35 頁的『設置 Windows Java 環境』
- 第 37 頁的『設置 Windows SQL 程序環境』

相關參照:

- 第 14 頁的『用於開發及執行應用程式的 Windows 支援軟體』

Java

設置 Windows Java 環境

本主題提供在 Windows 環境中建置並執行 DB2 Java 程式所需的資訊。

程序:

若要在具有 DB2 JDBC 支援的 Windows 作業系統上建置 Java 應用程式，您必須在開發機器上安裝和架構下列各項：

1. 下列其中一項：
 - IBM Developer Kit 及 Runtime Environment for Windows, Java 2 Technology Edition, 版本 1.3.1
 - Sun Microsystems 的 Java Development Kit (JDK) 1.3.1 for Win32
2. DB2 Java Enablement, 隨附於適用 Windows 從屬站及伺服器的 DB2 Universal Database 版本 8 中。

若要執行 DB2 Java 常式 (儲存程序及 UDF)，您必須在伺服器上更新 DB2 資料庫管理程式架構，以便併入 JDK 安裝在該機器上的路徑。您可在伺服器命令行上輸入下列命令來執行此動作：

```
db2 update dbm cfg using JDK_PATH c:\jdk13
```

其中 `c:\jdk13` 是安裝 JDK 的路徑。

如果安裝 JDK 的路徑之目錄名稱中有一或多個空格，您可以用單引號括住路徑。例如：

```
db2 update dbm cfg using JDK_PATH 'c:\Program Files\jdk13'
```

或使用短格式且沒有空格的目錄名稱：

```
db2 update dbm cfg using JDK_PATH c:\progra~1\jdk13
```

您可在伺服器上輸入下列命令，檢查 DB2 資料庫管理程式架構，以驗證 JDK_PATH 欄位的值是否正確：

```
db2 get dbm cfg
```

您可能需要將輸出重新導向至某檔案以方便檢視。JDK_PATH 欄位出現在靠近輸出起始處。

下列命令可以放置在批次檔中，以設定 IBM Java Development Kit 的 Java 環境。批次檔必須在 DB2 命令視窗中執行。請確定您作出了所有必需的路徑變更來適應您的特定環境。其他受支援的 JDK 可使用類似的命令。

下列命令可讓範例批次檔設定 Sun JDK 1.3.1 環境：

```
set JDKPATH=D:\JAVA\SUNjdk131
set PATH=%JDKPATH%\bin;%PATH%
set CLASSPATH=%CLASSPATH%;%JDKPATH%\lib\jdbc2_0-stdext.jar
db2 update dbm cfg using JDK_PATH %JDKPATH%
db2 terminate
db2stop
db2start
```

批次檔必須在 DB2 命令視窗中執行。

相關概念:

- 第 36 頁的『Windows Java 環境設定值』
- 第 101 頁的『Java 範例程式』
- 第 103 頁的『Java Applet 注意事項』

相關作業:

- 第 39 頁的『設置範例資料庫』

相關參照:

- 第 14 頁的『用於開發及執行應用程式的 Windows 支援軟體』

Windows Java 環境設定值

欲在具有 DB2® JDBC 支援的 Windows® 平台上執行 JDBC 及 SQLJ 程式，在安裝 DB2 時，系統會自動更新 CLASSPATH 以併入：

- "." (現行目錄)
- 檔案 sqllib\java\db2java.zip
- 檔案 sqllib\java\db2jcc.jar

若要建置 SQLJ 程式，則也會更新 CLASSPATH 來包括下面檔案：


```
sqllib\java\sqlj.zip
```

若要執行 SQLJ 程式，則也會更新 CLASSPATH 來包括下面檔案：

```
sqllib\java\runtime.zip
```

註：DB2 版本 8 不支援 Microsoft Software Developer's Kit for Java。它既無法用於 SQLJ 自行設定，也無法用於執行類型 2 JDBC 應用程式。

相關作業：

- 第 35 頁的『設置 Windows Java 環境』

SQL 程序

設置 Windows SQL 程序環境

下列 Windows 作業系統支援 SQL 程序：Windows NT、Windows 2000、Windows XP 及 Windows .NET Server。

「SQL 程序」需要利用伺服器上的編譯器架構，設定兩個環境變數：DB2_SQLROUTINE_COMPILER_PATH 及 DB2_SQLROUTINE_COMPILE_COMMAND。如果將您的編譯器環境變數設定為 SYSTEM 變數，將不需要任何架構。

限制：

嘗試設定變數 DB2_SQLROUTINE_COMPILER_PATH 時，64 位元 Windows 會存在一個問題，因為它需要完整的檔案路徑，並且不容許引數/開關。因此，如果使用者指定了下列，它將無法運作：

```
db2set DB2_SQLROUTINE_COMPILER_PATH="C:\MsSdk64\SetEnv.bat /XP64 /RETAIL"
```

解決之道在於建立另一個批次檔，利用適當的旗號呼叫 Microsoft 的設定批次檔，範例如下：

```
db2set DB2_SQLROUTINE_COMPILER_PATH="C:\MsSdk64\SetEnvXP64.bat"
```

其中 C:\MsSdk64\SetEnvXP64.bat 的內容如下：

```
call C:\MsSdk64\SetEnv.bat /XP64 /RETAIL
```

這個問題不會發生在 Windows 32 位元環境，因為 vcvars32.bat 不需要也不接受任何參數。

程序：

假設 C++ 編譯器已安裝在 C: 磁碟機，請設定如下的
DB2_SQLROUTINE_COMPILER_PATH DB2 登錄變數：

對於 Microsoft Visual C++ 版本 5.0：

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\devstudio\vc\bin\vcvars32.bat"
```

對於 Microsoft Visual C++ 版本 6.0：

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\Micros~1\vc98\bin\vcvars32.bat"
```

如有必要，可變更磁碟機或路徑，以反映 C++ 編譯器在系統中的位置。

對於編譯命令，請使用關鍵字 SQLROUTINE_FILENAME 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。若使用 Microsoft
Visual C++ 版本 5.0 及 6.0，則為：

```
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"
```

若尚未設定 DB2_SQLROUTINE_COMPILE_COMMAND DB2 登錄變數，則此為
預設編譯命令。

底下是 Microsoft Visual C++ 6.0 版編譯器之環境設定命令的範例。執行下列命令
的方法為將它們剪下並貼至批次檔，然後在 DB2 命令視窗中執行檔案。請確保對
於特定的環境作出所有必要的變更（其中包括路徑設定）：

```
@echo on  
rem Setting the SQL PROCEDURE environment:  
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\Micros~1\vc98\bin\vcvars32.bat"  
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"  
@echo off
```

相關作業：

- 第 131 頁的『保留 SQL 程序的中間檔』
- 第 132 頁的『自訂 SQL 程序的前置編譯及連結選項』
- 第 133 頁的『備份及復置 SQL 程序』
- 第 127 頁的『建立 SQL 程序』
- 第 137 頁的『利用 CALL 陳述式呼叫儲存程序』
- 第 129 頁的『在 Windows 上利用從屬站應用程式呼叫 SQL 程序』

- 第 134 頁的『分送編譯後的 SQL 程序』

相關參照:

- 第 14 頁的『用於開發及執行應用程式的 Windows 支援軟體』

範例資料庫

設置範例資料庫

若要使用 DB2 提供的範例程式，您必須在伺服器工作站上建立 `sample` 資料庫。若要使用遠端從屬站來存取伺服器的 `sample` 資料庫，您必須在從屬工作站上編目 `sample` 資料庫。

另外，若要使用遠端從屬站來存取伺服器的 `sample` 資料庫，而伺服器上執行不同版本的 DB2 或不同的作業系統，則需要將資料庫公用程式 (包括 DB2 CLI 公用程式檔案) 連結到 `sample` 資料庫。

程序:

以下是設置 `sample` 資料庫的步驟：

1. 建立範例資料庫
2. 編目範例資料庫
3. 連結範例資料庫公用程式

相關作業:

- 第 39 頁的『建立範例資料庫』
- 第 42 頁的『編目範例資料庫』
- 第 43 頁的『連結範例資料庫公用程式』

建立範例資料庫

您可以使用 `db2samp1` 命令，在命令行上建立 `sample` 資料庫。

先決條件:

若要建立資料庫，您必須具有「系統管理員 (SYSADM)」或「系統控制 (SYSCTRL)」權限。SYSADM 及 SYSCTRL 分別是 DB2 的第一及第二高權限層次。

程序:

若要建立資料庫，請在伺服器上執行下列動作：

1. 確定 `db2samp1` (用來建立 `sample` 資料庫的程式) 的位置在您的路徑中。`db2profile` 或 `db2cshrc` 檔案會將 `db2samp1` 放入您的路徑中，因此除非改變，否則位置保持不變。

- 在 UNIX 伺服器上，`db2samp1` 位於：

```
$HOME/sql1lib/bin
```

其中 `$HOME` 是 DB2 案例擁有者的起始目錄

- 在 Windows，`db2samp1` 位於：

```
%DB2PATH%\bin
```

其中 `%DB2PATH%` 即是 DB2 的安裝所在

2. 請確定 `DB2INSTANCE` 環境變數已設定為您要建立 `sample` 資料庫的案例名稱。如果未設定，您可以使用下列命令來設定：

- 在 UNIX：

您可以對 `bash` 或 `Korn shell` 進行設定，請輸入：

```
DB2INSTANCE=instance_name  
export DB2INSTANCE
```

對 `C shell` 執行此動作輸入下列命令：

```
setenv DB2INSTANCE instance_name
```

- 在 Windows 上輸入：

```
set DB2INSTANCE=instance_name
```

其中 `instance_name` 即是資料庫案例的名稱

3. 在輸入 `db2samp1` 後緊接著輸入您要建立範例資料庫的位置，來建立 `sample` 資料庫。在 UNIX 平台上，這是 `path` (例如，"`$HOME`")，輸入方式如下：

```
db2samp1 path
```

例如：

```
db2samp1 $HOME
```

在 Windows 上，這是 `drive` (例如，"`C:`")，其輸入方式如下：

```
db2samp1 drive
```

例如：

```
db2samp1 C:
```

如果您未指定路徑或磁碟機，則安裝程式將範例表格安裝在資料庫管理程式架構檔中的 DFTDBPATH 參數所指定的預設路徑或磁碟機中。資料庫與在其中建立它的案例具有相同的身分驗證類型。

相關作業:

- 第 41 頁的『在主電腦或 AS/400 及 iSeries 伺服器上建立範例資料庫』
- 第 42 頁的『編目範例資料庫』
- 第 43 頁的『連結範例資料庫公用程式』

在主電腦或 AS/400 及 iSeries 伺服器上建立範例資料庫

如果您要對照「主電腦」伺服器 (如 DB2 UDB for z/OS 及 OS/390，或 AS/400 及 iSeries 伺服器) 執行範例程式，則必須建立一個資料庫，其中包含了 SQL Reference 中所述的範例表格。

註: 您需要 DB2 Connect，才能連接主電腦伺服器。

限制:

工作站上的 DB2 與主電腦系統上的 DB2 存在一些 SQL 語法及 DB2 命令方面的差異。當存取 DB2 UDB for z/OS 及 OS/390，或 DB2 for AS/400 及 iSeries 上的資料庫時，請確定您的程式使用這些資料庫系統支援的 SQL 陳述式及前置編譯/連結選項。

程序:

若要建立資料庫，請：

1. 使用 db2samp1，在 DB2 工作站伺服器案例中建立 sample 資料庫。
2. 連接 sample 資料庫。
3. 將範例表格資料匯出到檔案中。
4. 連接主電腦資料庫。
5. 建立範例表格。
6. 從工作站伺服器上資料匯出所在的檔案匯入範例表格資料。

相關概念:

- *Data Movement Utilities Guide and Reference* 中的『Export Overview』
- *Data Movement Utilities Guide and Reference* 中的『Import Overview』

相關作業:

- 第 42 頁的『編目範例資料庫』

- 第 43 頁的『連結範例資料庫公用程式』

相關範例:

- 『exp samp.sqb -- Export and import tables with table data to a DRDA database (IBM COBOL)』
- 『tbmove.sqc -- How to move table data (C)』
- 『tbmove.sqC -- How to move table data (C++)』

編目範例資料庫

若要从遠端從屬站來存取伺服器的 `sample` 資料庫，您必須在從屬工作站上編目 `sample` 資料庫。

您不需要在伺服器工作站上編目 `sample` 資料庫，因為當您建立它時，即已在伺服器上編目它。

編目會以從屬站應用程式想要存取的資料庫名稱，來更新從屬工作站上的資料庫目錄。當處理從屬站要求時，資料庫管理程式會使用已編目的名稱，來尋找資料庫，並與它連接。

程序:

若要在遠端從屬工作站上編目範例資料庫，請輸入：

```
db2 catalog database sample as sample at node nodename
```

其中 *nodename* 即是伺服器節點的名稱

您也須先將遠端節點編入目錄中，方可與資料庫連接。

相關作業:

- 安裝與架構補充資料 中的『編目從屬站上的 TCP/IP 節點』
- 安裝與架構補充資料 中的『使用 CLP 將資料庫編目』
- 安裝與架構補充資料 中的『編目 DB2 從屬站上的 NetBIOS 節點』
- 安裝與架構補充資料 中的『編目從屬站上的「具名管道」節點』
- 安裝與架構補充資料 中的『編目 DB2 從屬站上的 APPC 節點』
- 第 43 頁的『連結範例資料庫公用程式』

連結範例資料庫公用程式

如果您要從執行不同版本的 DB2，或在不同作業系統上執行 DB2 的遠端從屬站，存取伺服器上的 `sample` 資料庫，則您需要將資料庫公用程式 (包括 DB2 CLI 公用程式) 連結至 `sample` 資料庫。

連結作業會建立資料包，於應用程式執行時，供資料庫管理程式用來存取資料庫。可以對前置編譯期間建立的連結檔案，指定 `BIND` 命令，明確地完成連結作業。

程序:

隨著您使用的從屬工作站平台，連結資料庫公用程式的方法有所不同。

在 UNIX 從屬工作站：

1. 輸入下列命令，連接 `sample` 資料庫：

```
db2 connect to sample user userid using password
```

其中 *userid* 及 *password* 是 `sample` 資料庫所在位置的案例之使用者 ID 及通行碼。

2. 經由輸入下列命令，使公用程式與資料庫連結：

```
db2 bind BNDPATH@db2ubind.lst blocking all sqlerror continue \  
messages bind.msg grant public
```

```
db2 bind BNDPATH@db2cli.lst blocking all sqlerror continue \  
messages cli.msg grant public
```

其中 *BNDPATH* 是連結檔案所在的路徑，如 `$HOME/sqlllib/bnd`，其中 `$HOME` 是 DB2 案例擁有者的起始目錄

3. 經由檢查連結訊息檔 `bind.msg` 及 `cli.msg`，驗證是否已完成連結。

在執行 Windows 作業系統的從屬工作站：

1. 從「開始」功能表中，選取「程式集」。
2. 從「程式集功能表」(或是從 Windows XP 上的「所有程式」) 選取 IBM DB2。
3. 在「IBM DB2」功能表中，選取「DB2 命令」視窗。
此時出現命令視窗。
4. 輸入下列命令，連接 `sample` 資料庫：

```
db2 connect to sample user userid using password
```

其中 *userid* 及 *password* 是 `sample` 資料庫所在位置的案例之使用者 ID 及通行碼。

- 經由輸入下列命令，使公用程式與資料庫連結：

```
db2 bind "%DB2PATH%\bnd\@db2ubind.lst" blocking all
sqlerror continue messages bind.msg grant public
```

```
db2 bind "%DB2PATH%\bnd\@db2cli.lst" blocking all
sqlerror continue messages cli.msg grant public
```

其中 %DB2PATH% 是指安裝 DB2 的路徑。

- 結束命令視窗，檢查連結訊息檔案 bind.msg 及 cli.msg，驗證是否成功完成連結。

對於所有存取主電腦伺服器的從屬站，請指定下列 .lst 檔之一，代替 db2ubind.lst：

ddcsmvs.lst

適用於 DB2 for z/OS 及 OS/390

ddcsvm.lst

適用於 DB2 for VM

ddcsvse.lst

適用於 DB2 for VSE

ddcs400.lst

適用於 DB2 for AS/400 及 iSeries

例如：

- 如果要從 UNIX 從屬站存取 DB2 for z/OS 及 OS/390 伺服器，請輸入：

```
db2 bind BNDPATH/@ddcsmvs.lst blocking all sqlerror continue \
messages bind.msg grant public
```
- 如果要從 Windows 從屬站存取 DB2 for z/OS 及 OS/390 伺服器，請輸入：

```
db2 bind "%DB2PATH%\bnd\@ddcsmvs.lst" blocking all
sqlerror continue messages bind.msg grant public
```

相關參照：

- Command Reference* 中的『BIND Command』

移轉應用程式

DB2 版本 8 支援下列 DB2 版本的移轉：

- DB2 版本 6
- DB2 版本 7.1
- DB2 版本 7.2

- DataJoiner® 版本 2.1.x

當移轉到更新版本的 DB2 時，會自動移轉您的資料庫及節點目錄。若要從其他舊版的 DB2 移轉，首先必須移轉到上述支援此移轉的支援版本之一，然後從該版本移轉到 DB2 版本 8。

從下列特定環境移轉時，請記住下列幾點：

Java 反對使用類型 3 驅動程式 (之前稱為「網路」驅動程式)。DB2 Java applet 應該移轉到類型 4 驅動程式。若要轉換類型 3 JDBC applet 以使用新的類型 4 驅動程式，請進行下列變更：

1. 類型 4 驅動程式保存檔是 db2jcc.jar。在與 applet 相關的 .html 檔中，將保存檔從 db2java.zip 變更為 db2jcc.jar。將 db2jcc.jar 複製到 Web 伺服器。
2. 類型 4 驅動程式類別名稱是 com.ibm.db2.jcc.DB2Driver。在 applet.java 檔中，將類型 3 驅動程式類別名稱 COM.ibm.db2.jdbc.net.DB2Driver 變更為類型 4 類別名稱。如果 applet 使用 javax.sql.DataSource 來取得連線，則不能有驅動程式類別的參照。
3. 類型 3 及類型 4 驅動程式都使用同一格式的資料來源 URL：jdbc:db2://server:portnumber/dbname。然而，三部份：server、portnumber 及 dbname 在兩個驅動程式中具有不同意義。
類型 3 驅動程式是三層模型，具有從屬站 (執行 applet 的瀏覽器)、JDBC Applet Server 及 DB2 伺服器。URL 中的 server 及 portnumber 代表 JDBC Applet Server。dbname 是執行 JDBC Applet Server 的系統上已編目的資料庫別名。
類型 4 驅動程式從屬站直接連接到 DB2 伺服器，以便 server 及 portnumber 是 DB2 伺服器 TCP/IP 監聽者的伺服器及埠號。dbname 是 DB2 伺服器系統上已編目的資料庫別名。
如果 applet 使用 DriverManager.getConnection 以連接 DB2，請以類型 4 驅動程式的新 URL 更新 .java 檔及 (必要時) .html 檔。
4. 如果 applet 利用 COM.ibm.db2.jdbc.DB2DataSource，則類別 com.ibm.jcc.db2.DB2SimpleDataSource 的新 javax.sql.DataSource 必須加以建立。applet 必須加以更新，才能使用這個新類別。

註：類型 4 驅動程式不支援 JDBC 2.0 Optional Package API，將連線置於儲存池 (javax.sql.ConnectionPoolDataSource、javax.sql.PooledConnection)。如果 applet 利用這些類型 3 特性，它將無法移轉到類型 4 驅動程式。

32 位元至 64 位元環境

可能不需要將您的應用程式從 32 位元移轉至 64 位元。DB2 版本 8 支援在所有 64 位元 DB2 案例 (Linux IA64 除外) 中執行現存的 32 位元本端應用程式。爲了能夠做到這樣，使用者必須重新連結 32 位元應用程式，然後以適當的檔案庫路徑來執行它。例如，若要在 AIX 上的 64 位元 DB2 案例中執行 32 位元應用程式，請：

- 在 bash 或 Korn shell 輸入：

```
export LIBPATH=$HOME/sqlllib/lib32
my32bitapp
```

- 在 C shell 輸入：

```
setenv LIBPATH $HOME/sqlllib/lib32
my32bitapp
```

仍在 32 位元伺服器上執行時，如果您想要將 32 位元應用程式移轉到 64 位元作業環境，請使用 LONGERROR 前置編譯選項，以準備移轉應用程式。在 32 位元環境中，將 LONGERROR 設爲 YES，以便前置編譯器每當發現長整數類型的主變數時，就會傳回一個錯誤。然後，請遵循下列步驟：

1. 不使用長類型的主變數，除非需要長類型。請改用新的可攜性主變數：sqlint32 或 sqluint32。例如：

```
EXEC SQL BEGIN DECLARE SECTION;
    long y;    /* this declaration generates an error on 64 bit */
    sqlint32 x; /* this declaration is acceptable for 64 bit */
EXEC SQL END DECLARE SECTION;
```

2. 對照 64 位元伺服器上的資料庫前置編譯應用程式。如此將爲要移轉的應用程式建立新的資料包。
3. 在 64 位元模式中編譯應用程式。
4. 以新的 64 位元 DB2 檔案庫鏈結應用程式。
5. 將應用程式連結到 64 位元伺服器上的資料庫。

HP-UX

如果您是從 HP-UX 版本 10 或更早的版本將 DB2 移轉到 HP-UX 版本 11，您的 DB2 程式必須以 HP-UX 版本 11 上的 DB2 重新執行前置編譯 (如果它們併入內嵌式 SQL)，且必須重新編譯。包括所有 DB2 應用程式、儲存程序、使用者定義函數及使用者跳出程式都需進行此項作業。並且，在 HP-UX 版本 11 上編譯的 DB2 程式不能在 HP-UX 版本 10 或更早版本上執行。在 HP-UX 版本 10 編譯及執行的 DB2 程式可從遠端連接到 HP-UX 版本 11 伺服器。

Micro Focus COBOL

透過 DB2 版本 2.1.1 或更早版本來進行前置編譯，並以 Micro Focus

COBOL 進行編譯的任何現存的應用程式應該以現行版本的 DB2 重新進行前置編譯，然後以 Micro Focus COBOL 重新進行編譯。如果這些以 IBM® 前置編譯器的較早版本開發的應用程式未重新進行前置編譯，則在發生異常終止時，資料庫可能會損毀。

底下是開發應用程式時要記住的幾點。這些注意事項會讓您的應用程式更具可攜性：

- 在 UNIX 上，僅於應用程式中使用預設檔案庫搜尋路徑 `/usr/lib:/lib`。在 Windows® 作業系統，使用下列命令以確定 LIB 環境變數指向 `%DB2PATH%\lib`：

```
set LIB=%DB2PATH%\lib;%LIB%
```

並且在預設路徑及您使用的 DB2 版本之間建立符號鏈結。確定鏈結是連到您應用程式所需要的最小 DB2 層次。關於設定鏈結的資訊，請參閱關於您平台的快速入門書籍。

- 如果您的應用程式需要特定的 DB2 版本，請在您的應用程式中編寫所指定 DB2 版本的路徑。例如，如果您的 AIX® 應用程式需要 DB2 版本 5，請編寫 `/usr/lpp/db2_05_00/lib`。通常，您不需要這麼做。
- 當您在建置生產而非內部開發的應用程式時，應用程式中的路徑不應指向 UNIX 中 `sqllib/lib` 目錄的案例擁有者拷貝，或指向 Windows 作業系統上的 `sqllib\lib` 目錄。這會讓應用程式高度依賴特定使用者名稱和環境。
- 一般而言，請不要使用下列環境變數來變更特殊環境中的搜尋路徑：`LIBPATH` (AIX)、`SHLIB_PATH` (HP-UX 32 位元)、`LD_LIBRARY_PATH` (HP-UX 64 位元、Linux 及 Solaris)，以及 `LIB` (Windows)。這些變數會置換在環境中執行之應用程式內所指定的搜尋路徑，所以應用程式可能找不到所需的檔案庫或檔案。
- 在 DB2 Universal Database™ 版本 6、7 及 8 中，含有字串語義的所有字元陣列項目都是 `char` 類型，而非其它變項，例如，`unsigned char`。您使用 DB2 Universal Database 版本 6、版本 7 或版本 8 編寫的任何應用程式都應遵循此慣例。

如果您具有使用 `unsigned char` 的 DB2 版本 1 應用程式，則編譯器會發出警告或錯誤訊息，因為版本 1 應用程式中的 `unsigned char`，與版本 6.1、版本 7 或版本 8 函數原型中的 `char` 之間發生類型衝突。如果發生這種情形，請使用編譯器選項 `-DSQLOLDCHAR` 來解決問題。

在 UNIX 平台上，如果您具有來自舊 DB2 版次版本的應用程式，並且要它們在同一部機器中舊版及 DB2 版本 8 的資料庫案例上同時執行，則您必須對環境作某些變更。若要決定需作何種變更，請回答下列問題後，複查「狀況」這一節，查看是否有任何狀況適合您的情況。

此處以 AIX 系統來解釋所提出的要點。相同概念適用其它 UNIX 平台，但其細節可能不同，例如環境變數和特定命令。

問題

問題 1: 如何將前一版的應用程式鏈結到 DB2 從屬站執行檔案庫，例如，AIX 中的 libdb2.a？

若要決定可執行檔的內嵌式共用檔案庫搜尋路徑，請在可執行檔常駐的目錄 (可能是 /usr/bin) 或您的案例目錄中使用下列其中一個系統命令：

AIX `dump -H executable_filename`

HP-UX

`chatr executable_filename`

Linux `objdump -p executable_filename`

Solaris

`dump -Lv executable_filename`

其中 *executable_filename* 是應用程式的可執行檔名稱。

下列是 AIX C 範例應用程式 `dbcat` (取自 DB2 案例的範例子目錄 `/home/dbinst/samples/c`) 的範例傾出報表，來自 DB2 版本 7.2：

C 應用程式 dbcat 的範例傾出報表

```

-----
dbcat:

  ***Loader Section***

                        Loader Header Information
VERSION#                #SYMtableENT      #RELOCent              LENidSTR
  0x00000001            0x0000000f        0x00000015            0x00000047

#IMPfilID              OFFfidSTR          LENstrTBL              OFFstrTBL
  0x00000003            0x00000284          0x0000007f            0x000002cb

  ***Import File Strings***
INDEX  PATH                                BASE                                MEMBER
  0     /home/db2inst/sqllib/lib:/usr/lib:/lib
  1                                     libc.a                             shr.o
  2                                     libdb2.a                            shr.o
-----

```

行 0 (零) 顯示可執行檔尋找所鏈結之共用檔案庫時搜尋的目錄路徑。行 1 和 2 顯示應用程式所鏈結的共用檔案庫。

根據應用程式的建立方式，您會看到下列不同路徑：

`/usr/lpp/db2_07_01_0000/lib`、`INSTHOME/sqllib/lib` (其中 `INSTHOME` 是資料庫案例擁有者的起始目錄) 或只有 `/usr/lib:/lib` 組合。

問題 2：如何在系統上架構 DB2 執行檔案庫？

在安裝 DB2 版本 1、2、5、6.1 或 7 其中之一時，有一個選用性步驟，可建立系統預設的共用檔案庫路徑 `/usr/lib` 與 DB2 安裝路徑 (其中含有 DB2 從屬站執行檔案庫) 之間的符號鏈結。

AIX 上不同 DB2 版本之安裝路徑如下：

版本 1

`/usr/lpp/db2_01_01_0000/lib`

版本 2

`/usr/lpp/db2_02_01/lib`

版本 5

`/usr/lpp/db2_05_00/lib`

版本 6.1

`/usr/lpp/db2_06_01/lib`

版本 7

`/usr/lpp/db2_07_01/lib`

版本 8

`/usr/opt/db2_08_01/lib`

無論使用的是哪一種版本，執行時共用檔案庫皆命名為 `libdb2.a`。

任何時候，只能以其中一個版本的檔案庫作為預設值。DB2 提供此預設值，讓您建立應用程式時，該應用程式不必依據特定的 DB2 版本。

問題 3：是否需要在環境中指定不同的搜尋路徑？

您可以在 AIX 上使用 `LIBPATH` 環境變數，在 HP-UX 上使用 `SHLIB_PATH` 環境變數，以及在 Linux 和 Solaris 上使用 `LD_LIBRARY_PATH` 環境變數，置換於應用程式中所編寫的共用檔案庫 搜尋路徑。您可以使用「問題 1」回覆所提供適合您平台的系統命令，來查看檔案庫搜尋路徑。

狀況

解決上面的問題後，您可能需要變更環境。請閱讀下面所列示的狀況。如果其中一個狀況適用於您的情況，請依該狀況作必要的變更。

狀況 1：如果版本 7 應用程式從 AIX 預設共用檔案庫路徑 `/usr/lib/libdb2.a` 載入共用檔案庫，且

- 如果從 `/usr/lib/libdb2.a` 至 `/usr/lpp/db2_07_01/lib/libdb2.a` 間有符號鏈結，且資料庫伺服器是 DB2 Universal Database for AIX 版本 8，請執行下列其中一項：

- 變更符號鏈結以指向：

`/usr/opt/db2_08_01/lib/libdb2.a`

若您是 `root` 使用者，您可以使用 `"db2ln"` 命令變更鏈結，如下所示：

`/usr/opt/db2_08_01/cfg/db2ln`

- 設定 `LIBPATH` 環境變數以指向 `/usr/opt/db2_08_01/lib` 或 `INSTHOME/sql/lib/lib`，其中 `INSTHOME` 是版本 8 DB2 案例擁有者的起始目錄。
- 在應用程式（從屬站）案例與伺服器案例之間架構 TCP/IP 連線。

- 如果 `/usr/lib/libdb2.a` 與 `/usr/opt/db2_08_01/lib/libdb2.a` 間有符號鏈結，且資料庫伺服器是 DB2 版本 7，則在應用程式（從屬站）案例與伺服器案例間架構一個 TCP/IP 連線。

狀況 2：如果版本 7 應用程式從 DB2 版本 7 案例擁有者的 `$HOME` 路徑 (`$HOME/sqlllib/lib/libdb2.a`) 載入共用檔案庫，且資料庫伺服器是 DB2 Universal Database for AIX 版本 8，請執行下列其中一項：

- 將應用程式案例移轉成與資料庫伺服器案例相同的版本。
- 設定 `LIBPATH` 環境變數以指向 `/usr/opt/db2_08_01/lib` 或 `INSTHOME/sqlllib/lib`，其中 `INSTHOME` 是版本 8 案例擁有者的起始目錄。
- 在應用程式（從屬站）案例與伺服器案例之間架構 TCP/IP 連線。

狀況 3：如果版本 7 應用程式從 DB2 版本 7 安裝路徑 (`/usr/lpp/db2_07_01/lib/libdb2.a`) 載入共用檔案庫，且資料庫伺服器是 DB2 Universal Database for AIX 版本 8，請執行下列其中一項：

- 設定 `LIBPATH` 環境變數以指向 `/usr/opt/db2_08_01/lib` 或 `INSTHOME/sqlllib/lib`，其中 `INSTHOME` 是資料庫案例擁有者的起始目錄。
- 在應用程式（從屬站）案例與伺服器案例之間架構 TCP/IP 連線。

狀況 4：如果版本 7 應用程式從 DB2 Universal Database for AIX 版本 8 安裝路徑 (`/usr/opt/db2_08_01/lib/libdb2.a`) 載入共用檔案庫，且資料庫伺服器是 DB2 版本 7，則在應用程式（從屬站）案例與伺服器案例間架構 TCP/IP 連線。

相關概念：

- *DB2 伺服器快速入門* 中的『移轉建議』
- *Application Development Guide: Programming Client Applications* 中的『JDBC 2.1 Core API Restrictions by the DB2 JDBC Type 4 Driver』
- *Application Development Guide: Programming Client Applications* 中的『JDBC 2.1 Optional Package API Support by the DB2 JDBC Type 4 Driver』

相關作業：

- 第 17 頁的『設置應用程式開發環境』
- *安裝與架構補充資料* 中的『建立 DB2 檔案的鏈結』
- *DB2 伺服器快速入門* 中的『移轉資料庫』
- *DB2 伺服器快速入門* 中的『移轉解譯表格』
- *DB2 伺服器快速入門* 中的『移轉案例 (UNIX)』
- *DB2 伺服器快速入門* 中的『將 V6 或 V7 DB2 伺服器離線以供 DB2 移轉』
- *DB2 伺服器快速入門* 中的『移轉 DB2 (Windows)』

- *DB2 伺服器快速入門* 中的『移轉 DB2 (UNIX)』

相關參照:

- *DB2 伺服器快速入門* 中的『移轉限制』
- *Administration Guide: Planning* 中的『Version 8 incompatibilities between releases』
- *Administration Guide: Planning* 中的『Version 7 incompatibilities between releases』

下一步要做什麼

一旦您的環境安裝好，就可以開始建置您的 DB2 應用程式。下一章將會討論範例程式及相關檔案，包括建置檔。以下章節將使用建置檔及範例，以指示您如何在您的程式設計環境中編譯、鏈結及執行應用程式。請依照您個別應用程式的開發所需要，閱讀特定的章節。

第 3 章 範例程式及相關檔案

範例檔案	53	Visual C++ 範例	83
範例程式：結構及設計.	58	Windows Management Instrumentation 範例	84
依語言及應用程式介面的範例程式	65	物件連結和內嵌 (OLE) 範例.	84
C/C++ 範例	65	物件連結和內嵌資料庫 (OLE DB) 表格函數 範例.	85
DB2 CLI 範例	68	命令行處理器 (CLP) 範例	86
JDBC 範例	70	REXX 範例	86
SQLJ 範例	72	日誌管理使用者跳出程式範例	88
Java WebSphere 範例	74	建置檔、make 檔及錯誤檢查公用程式.	90
Java 外掛程式範例	75	建置檔	90
COBOL 範例.	76	make 檔	93
SQL 程序範例	80	錯誤檢查公用程式	96
Visual Basic 範例	82		

本章說明 DB2 支援的所有平台其程式設計語言的範例程式及相關檔案。將提出依照 DB2 元件結構為基礎的範例設計，以及提供一連串 DB2 範例與其個別說明。它還解釋了 DB2 的建置檔、make 檔及錯誤檢查公用程式的使用。

範例檔案

範例程式隨附於 DB2[®] Application Development (DB2 AD) Client。並非所有範例程式均可用在所有平台或支援的程式設計語言上。您可以使用這些範例程式作為範本，建立您自己的應用程式，以及作為學習工具，瞭解 DB2 功能。

Db2 範例程式僅以「現狀」提供，不具任何保證。使用者（而非 IBM）要承擔品質、效能及任何錯誤修復的全部風險。

除了範例程式檔外，DB2 會在 `sqllib/samples` (UNIX) 下及 `sqllib\samples` (Windows) 下的範例目錄中提供其他範例檔案。這些包括要編譯及鏈結範例程式的建置檔及 `make` 檔、將鏈結至大部份範例程式的錯誤檢查公用程式，以及協助應用程式開發的不同 Script 檔。例如，提供 Script 以在數個語言子目錄中將儲存程序及 UDF 編目及取消編目。每一個範例目錄都有一個 README 檔，用以說明目錄中所包含的檔案。

同時，會提供大部份範例程式檔案的 HTML 版本，您可在線上文件中存取它們。這些「以 HTML 表示的範例」會鏈結至文件主題，以示範它們所說明的功能。關鍵字 (如 SQL 陳述式及 DB2 API) 是以 HTML 表示的範例內的熱鏈結，所以使用者可以直接跳至說明它們的文件。大部份以 HTML 表示的範例在檔案頂端的註解區段中都具有一個鏈結，以鏈結至顯示執行已編譯範例程式所產生的典型結果

的範例輸出檔。請注意，在大部份情況中，實際輸出是與機器及平台相依的，所以您從執行相同程式所收到的輸出可能會有所不同。

底下表格將依平台顯示主要支援程式設計語言/API 的範例目錄及 README 檔。README 檔是線上文件中的熱鏈結，而且它們含有的範例清單具有範例檔原始碼的熱鏈結。您也可以在此列出的範例目錄中存取範例檔案。對於目錄路徑，使用 UNIX 樣式斜線，如同 `samples/c` 所示一般，不過，其中目錄僅適用於 Windows，如同 `samples\VB\ADO` 所示一般。

表 1. 依平台的範例 README 檔

平台 → 語言	AIX	HP-UX	Linux	Solaris	Windows
C samples/c	README	README	README	README	README
C++ samples/cpp	README	README	README	README	README
CLI samples/cli	README	README	README	README	README
JDBC samples/java	README	README	README	README	README
SQLJ samples/java	README	README	README	README	README
IBM COBOL samples/cobol	README	n/a	n/a	n/a	README
Micro Focus COBOL samples/cobol_mf	README	README	n/a	README	README
Visual Basic samples\VB\ADO	n/a	n/a	n/a	n/a	ReadMe.txt
SQL 程序 samples/sqlproc	README	README	README	README	README
CLP samples/clp	README	README	README	README	README

每一種支援語言的範例程式副檔名不同，而在每一種語言中，內嵌式 SQL 與非內嵌式 SQL 程式之範例程式副檔名也不同。某一語言中，程式群組的副檔名也會不同。這些不同的範例檔的副檔名均在下列表格中加以分類：

按語言分類的範例副檔名

第 55 頁的表 2.

按程式群組分類的範例副檔名

表 3.

表 2. 按語言分類的範例副檔名

語言	目錄	內嵌式 SQL 程式	非內嵌式 SQL 程式
C	samples/c samples/cli (CLI 程式)	.sqc	.c
C++	samples/cpp	.sqc (UNIX) .sqx (Windows)	.C (UNIX) .cxx (Windows)
COBOL	samples/cobol samples/cobol_mf	.sqb	.cbl
JAVA	samples/java/jdbc samples/java/sqlj samples/java/WebSphere samples/java/plugin	.sqlj	.java
REXX	samples/rexx	.cmd	.cmd
Visual Basic	samples\VB\ADO samples\VB\MTS samples\VB\RDO		.bas .frm .vbp
Visual C++	samples\VC\ADO		.cpp .dsp .dsw

表 3. 按程式群組分類的範例副檔名

範例群組	目錄	副檔名
CLP	samples/clp	.db2
OLE	samples\ole\msvb (Visual Basic) samples\ole\msvc (Visual C++)	.bas .vbp (Visual Basic) .cpp (Visual C++)
OLE DB	samples\oledb	.db2
SQL 程序	samples/sqlproc	.db2 (SQL 程序 Script) .c (CLI 從屬站應用程式) .sqc (內嵌式 C 從屬站應用程式) .java (JDBC 從屬站應用程式)
使用者跳出程式	samples/c	.cadsm (UNIX & Windows) .cdisk (UNIX & Windows) .ctape (UNIX) .cxbsa (UNIX & Windows)

註:

目錄區隔字元

UNIX[®] 上的目錄區隔字元是 /。在 Windows[®] 上，它是 \。在這些表格中，除非目錄僅用在 Windows，否則將使用 UNIX 區隔字元。

內嵌式 SQL 程式

需要前置編譯，但 REXX 內嵌式 SQL 程式除外，因為該程式在執行時會解譯內嵌式 SQL 陳述式。

IBM[®] COBOL 範例

只有在 AIX[®] 及 Windows 32 位元作業系統的 cobol 子目錄中提供。

Micro Focus COBOL 範例

只有在 AIX、HP-UX、Solaris 作業環境及 Windows 32 位元作業系統的 cobol_mf 子目錄中提供。

Java[™] 範例

包括 Java Database Connectivity (JDBC) applet、應用程式及常式、內嵌式 SQL for Java (SQLJ) applet、應用程式，以及常式。此外，也包括「DB2 控制中心」的 WebSphere 範例及外掛程式範例檔。所有支援的 DB2 平台上皆提供 Java 範例。

REXX 範例

只在 AIX 及 Windows 32 位元作業系統上提供。

CLP 範例

執行 SQL 陳述式的命令行處理器 Script。

OLE 範例

適用於 Microsoft[®] Visual Basic 及 Microsoft Visual C++ 中的「物件連結和內嵌 (OLE)」，只在 Windows 作業系統上提供。

Visual Basic 範例

ActiveX 資料物件、遠端資料物件及 Microsoft Transaction Server 範例，僅在 Windows 作業系統上提供。

Visual C++ 範例

ActiveX 資料物件範例，只在 Windows 作業系統上提供。

使用者跳出程式範例

用來保存及擷取資料庫日誌檔的「日誌管理使用者跳出程式」。這些檔案必須以 .c 副檔名加以更名，並編譯為 C 語言程式。

您可以在已安裝 DB2 的目錄的 samples 次目錄中，找到範例程式。每一種支援的語言均有子目錄。下列範例告訴您如何在每一個支援的平台上，找出以 C 或 C++ 撰寫而成的範例。

- 在 UNIX 平台上。

您可以在您的資料庫案例目錄下的 `sqllib/samples/c` 中，找到內嵌式 SQL 及 DB2 API 程式的 C 原始程式；DB2 CLI 程式的 C 原始程式是在 `sqllib/samples/cli` 中。關於範例表格中範例程式的其它資訊，請參閱 DB2 案例下適當 `samples` 次目錄中的 README 檔案。README 檔將含有任何未列示在本書中的附加範例。

- **在 Windows 作業系統上。**

您可以在 DB2 安裝目錄下的 `sqllib\samples\c` 中，找到內嵌式 SQL 及 DB2 API 程式的 C 原始碼；DB2 CLI 程式的 C 原始碼是在 `sqllib\samples\cli` 中。關於範例表格中程式的附加資訊，請參閱適當 `sqllib\samples` 子目錄中的 README 檔案。README 檔將含有任何未列示在本書中的附加範例。

範例程式目錄在大多數平台上是唯讀的。在您可以變更或開發範例程式之前，請將它們複製到您的工作目錄中。

相關概念:

- 第 90 頁的『建置檔』
- 第 93 頁的『make 檔』
- 第 96 頁的『錯誤檢查公用程式』
- 第 58 頁的『範例程式：結構及設計』

相關參照:

- 第 65 頁的『C/C++ 範例』
- 第 68 頁的『DB2 CLI 範例』
- 第 70 頁的『JDBC 範例』
- 第 72 頁的『SQLJ 範例』
- 第 80 頁的『SQL 程序範例』
- 第 82 頁的『Visual Basic 範例』
- 第 83 頁的『Visual C++ 範例』
- 第 84 頁的『物件連結和內嵌 (OLE) 範例』
- 第 85 頁的『物件連結和內嵌資料庫 (OLE DB) 表格函數範例』
- 第 86 頁的『命令行處理器 (CLP) 範例』
- 第 88 頁的『日誌管理使用者跳出程式範例』
- 第 76 頁的『COBOL 範例』
- 第 74 頁的『Java WebSphere 範例』
- 第 75 頁的『Java 外掛程式範例』
- 第 84 頁的『Windows Management Instrumentation 範例』

範例程式：結構及設計

C、CLI、C++、Java 及 Visual Basic ADO 中的大部份 DB2® 範例會加以組織，以反映資料庫元件的物件型設計模型。範例會群組成代表不同 DB2 層次的種類。範例所屬的層次是以範例名稱開頭的兩個字元字首來指出 (對於 Java，第一個字元將大寫)。並不會以每一個「應用程式設計介面」的範例中代表所有層次，但是就整體範例而言，層次的代表如下：

字首	DB2 層次
il	安裝映像檔層次
cl	從屬站層次
in	案例層次
db	資料庫層次
ts	表格空間層次
tb	表格層次
dt	資料類型層次

這些層次顯示階層式結構。安裝映像檔層次是 DB2 的最上層。在這個層次之下，從屬站層次應用程式可以存取不同案例；一個案例可以具有一個或多個資料庫；一個資料庫具有表格存在於其中的表格空間，以及依序保存不同類型之資料的表格空間。

這個設計不包括所有 DB2 範例。部份範例的目的在於示範存取資料的不同方法。這些方法是這些範例的主要方法，所以這些方法以如下類似的方式代表它們：

字首	程式設計方法
fn	SQL 函數
sp	儲存程序
ud	使用者定義的函數

除了這些種類外，還有一組教學指導範例，以介紹資料庫程式設計的基本概念。這些範例會使用範例設計所代表的部份較簡單函數，而且以字元 "tut" 開頭。

有其他範例未包括在這個設計中，如「日誌管理使用者跳出程式」範例、COBOL、Visual C++、REXX 中的範例、「物件連結和內嵌 (OLE)」範例、CLP Script，以及「SQL 程序」。

底下是在這個設計中實施的範例程式。

註: Java™ 程式名稱具有第一個字元以及有時候以大寫字體表示的其他字元，而且在教學指導範例名稱中沒有底線。Visual Basic 範例具有一些以大寫字體表示的字元 (但不是第一個字元)。

表 4. DB2 Universal Database 中的範例程式設計

範例程式	說明
教學指導範例	
tut_mod	如何修改表格資料。 支援的語言/API：C、C++、CLI、JDBC
tut_read	如何讀取表格。 支援的語言/API：C、C++、CLI、JDBC
tut_use	如何使用資料庫。 支援的語言/API：C、C++、CLI
安裝映像檔層次	
ilinfo	如何取得/設定安裝層次資訊。 支援的語言/API：CLI、JDBC
從屬站層次	
cliExeSQL	如何執行 SQL 陳述式。 支援的語言/API：Visual Basic/ADO
cli_info	如何取得/設定從屬站層次資訊。 支援的語言/API：C、C++、CLI、Visual Basic/ADO
clihandl	如何配置及釋放 handle。 支援的語言/API：CLI
clisqlca	如何使用 SQLCA 資料。 支援的語言/API：CLI
clisnap	如何擷取從屬站層次 snapshot。 支援的語言/API：C、C++
案例層次	
inattach	如何連接/分離案例。 支援的語言/API：C、C++

表 4. DB2 Universal Database 中的範例程式設計 (繼續)

範例程式	說明
inauth	如何顯示案例層次中的權限。 支援的語言/API：C、C++
ininfo	如何取得及設定案例層次資訊。 支援的語言/API：C、C++、CLI
insnap	如何擷取案例層次 snapshot。 支援的語言/API：C、C++
instart	如何停止及啓動現行本端案例。 支援的語言/API：C、C++
資料庫層次	
dbauth	如何授與/顯示/取消資料庫層次中的權限。 支援的語言/API：C、C++、JDBC、SQLJ
dbcfg	如何架構資料庫及資料庫管理程式參數。 支援的語言/API：C、C++
dbconn	如何連接資料庫/中斷連線。 支援的語言/API：C、C++、CLI、JDBC、SQLJ、Visual Basic/ADO
dbcreate	如何建立及捨棄資料庫。 支援的語言/API：C、C++
dbinfo	如何在資料庫層次上取得及設定資訊。 支援的語言/API：C、C++、CLI、JDBC、Visual Basic/ADO
dbinline	如何使用列入 SQL 程序語言。 支援的語言/API：C
dbCommit	如何在資料庫層次上動態控制自動確定。 支援的語言/API：Visual Basic/ADO
dbmcon	如何連接多個資料庫/中斷連線。 支援的語言/API：C、C++、CLI、JDBC、SQLJ
dbmconx	如何連接多個資料庫/中斷連線。 支援的語言/API：CLI

表 4. DB2 Universal Database 中的範例程式設計 (繼續)

範例程式	說明
dbmigrat	如何移轉資料庫。 支援的語言/API：C、C++
dbnative	如何利用 ODBC 跳出子句轉換陳述式。 支援的語言/API：CLI、JDBC
dbpkg	如何使用資料包。 支援的語言/API：C、C++
dbrecov	如何回復資料庫。 支援的語言/API：C、C++
dbsample	如何建立包括主電腦及 AS/400® 表格及概略表的範例資料庫。 支援的語言/API：C
DbSeq	如何建立、變更及捨棄資料庫中的順序。 支援的語言/API：JDBC
dbsnap	如何擷取資料庫層次 snapshot。 支援的語言/API：C、C++
dbthrds	如何使用執行緒。 支援的語言/API：C、C++
dbuse	如何使用資料庫物件。 支援的語言/API：C、C++、CLI、JDBC、SQLJ
dbusemx	資料庫物件如何與內嵌式 SQL 一起使用。 支援的語言/API：CLI
表格空間層次	
tscreate	如何建立捨棄緩衝池及表格空間。 支援的語言/API：C、C++
tsinfo	如何在表格空間層次取得資訊。 支援的語言/API：C、C++

表格層次

表 4. DB2 Universal Database 中的範例程式設計 (繼續)

範例程式	說明
tbconstr	<p>如何使用表格限制。</p> <p>支援的語言/API：C、C++、CLI、JDBC、SQLJ</p>
tbcreate	<p>如何建立、變更及捨棄表格。</p> <p>支援的語言/API：C、C++、CLI、JDBC、SQLJ</p>
TbGenCol	<p>如何使用產生的直欄。</p> <p>支援的語言/API：JDBC</p>
tbident	<p>如何使用身分直欄。</p> <p>支援的語言/API：C、JDBC、SQLJ</p>
tbinfo	<p>如何在表格層次上取得及設定資訊。</p> <p>支援的語言/API：C、C++、CLI、JDBC、SQLJ</p>
tbmod	<p>如何修改表格中的資訊。</p> <p>支援的語言/API：C、C++、CLI、JDBC、SQLJ</p>
tbmove	<p>如何移動表格資料。</p> <p>支援的語言/API：C、C++</p>
tbpriv	<p>如何授與/顯示/取消表格層次專用權。</p> <p>支援的語言/API：C、C++、JDBC、SQLJ</p>
tbread	<p>如何讀取表格中的資訊。</p> <p>支援的語言/API：C、C++、CLI、JDBC、SQLJ</p>
tbreorg	<p>如何重組表格。</p> <p>支援的語言/API：C、C++</p>
tbsavept	<p>如何使用外部儲存點。</p> <p>支援的語言/API：C</p>
tbtemp	<p>如何使用已宣告的暫時表格。</p> <p>支援的語言/API：C、JDBC</p>
tbtrig	<p>如何在表格上使用觸發函式。</p> <p>支援的語言/API：C、C++、JDBC、SQLJ</p>

資料類型層次

表 4. DB2 Universal Database 中的範例程式設計 (繼續)

範例程式	說明
dtformat	如何使用載入及匯入資料格式擴充。 支援的語言/API：C
dtHier	如何擷取階層式資料。 支援的語言/API：Visual Basic/ADO
dtinfo	如何取得資料類型的相關資訊。 支援的語言/API：CLI、JDBC
dtlob	如何讀取及寫入 LOB 資料。 支援的語言/API：C、C++、CLI、JDBC、Visual Basic/ADO
dtstruct	如何建立、使用及捨棄結構化類型及類型化表格的階層結構。 支援的語言/API：C++
dtudt	如何建立/使用/捨棄使用者定義的特殊類型。 支援的語言/API：C、C++、CLI、JDBC、SQLJ
DB2 函數	
fnuse	如何使用 SQL 函數。 支援的語言/API：C、C++
儲序程序	
spcall	如何呼叫儲存程序。 支援的語言/API：CLI、Visual Basic/ADO
spclient	在 spserver 中呼叫儲存程序的從屬站應用程式。 支援的語言/API：C、C++、CLI、JDBC、SQLJ
spserver	spclient 所呼叫的儲存程序常式。 支援的語言/API：C、C++、CLI、JDBC、SQLJ
使用者定義的函數	
udfcli	在 udfsrv 中呼叫 UDF 的從屬站應用程式。 支援的語言/API：C、C++、CLI、JDBC、SQLJ
udfsrv	udfcli 所呼叫的使用者定義的函數檔案庫。 支援的語言/API：C、C++、CLI、JDBC、SQLJ

表 4. DB2 Universal Database 中的範例程式設計 (繼續)

範例程式	說明
udfjcli	在 udfjsrv 中呼叫「參數樣式 Java」使用者定義的函數的從屬站應用程式。 支援的語言/API：JDBC、SQLJ
udfjsrv	udfjcli 所呼叫的「參數樣式 Java」使用者定義的函數檔案庫。 支援的語言/API：JDBC、SQLJ
udfsqlcl	在 udfsqlsv 中呼叫 Java 使用者定義的函數的從屬站應用程式。 支援的語言/API：JDBC
udfsqlsv	udfsqlcl 所呼叫、具有 SQL 陳述式的 Java 使用者定義的函數檔案庫。 支援的語言/API：JDBC
udfUse	如何建立及使用 UDT 及 UDF。 支援的語言/API：Visual Basic/ADO

相關概念:

- 第 90 頁的『建置檔』
- 第 93 頁的『make 檔』
- 第 96 頁的『錯誤檢查公用程式』
- 第 53 頁的『範例檔案』

相關參照:

- 第 65 頁的『C/C++ 範例』
- 第 68 頁的『DB2 CLI 範例』
- 第 70 頁的『JDBC 範例』
- 第 72 頁的『SQLJ 範例』
- 第 80 頁的『SQL 程序範例』
- 第 82 頁的『Visual Basic 範例』
- 第 83 頁的『Visual C++ 範例』
- 第 84 頁的『物件連結和內嵌 (OLE) 範例』
- 第 85 頁的『物件連結和內嵌資料庫 (OLE DB) 表格函數範例』
- 第 86 頁的『命令行處理器 (CLP) 範例』
- 第 88 頁的『日誌管理使用者跳出程式範例』
- 第 76 頁的『COBOL 範例』

- 第 74 頁的『Java WebSphere 範例』
- 第 75 頁的『Java 外掛程式範例』
- 第 84 頁的『Windows Management Instrumentation 範例』
- 第 86 頁的『REXX 範例』

依語言及應用程式介面的範例程式

C/C++ 範例

註: 副檔名:

C .c (非內嵌式 SQL)
 .sqc (內嵌式 SQL)
C++ .C (非內嵌式 SQL)
 .sqC (內嵌式 SQL)

表 5. C/C++ 範例程式檔

範例程式名稱	程式說明
教學指導範例 - 示範基本資料庫作業的程式。	
tut_mod.sqc, tut_mod.sqC	如何修改表格資料。
tut_read.sqc, tut_read.sqC	如何讀取表格。
tut_use.sqc, tut_use.sqC	如何使用資料庫。
從屬站層次 - 處理 DB2 之從屬站層次的範例。	
cli_info.c, cli_info.C	如何取得及設定從屬站層次資訊。
clisnap.c, clisnap.C	如何在從屬站層次中擷取 snapshot。
案例層次 - 處理 DB2 之案例層次的範例。	
inattach.c, inattach.C	如何連接/分離案例。
inauth.sqc, inauth.sqC	如何顯示案例層次中的權限。
ininfo.c, ininfo.C	如何取得及設定案例層次資訊。
insnap.c, insnap.C	如何在案例層次中擷取 snapshot。
instart.c, instart.C	如何停止及啟動現行本端案例。

表 5. C/C++ 範例程式檔 (繼續)

範例程式名稱	程式說明
資料庫層次 - 處理 DB2 中資料庫物件的範例。	
dbauth.sqc, dbauth.sqC	如何授與/顯示/取消資料庫層次中的權限
dbcfg.sqc, dbcfg.sqC	如何架構資料庫及資料庫管理程式參數。
dbconn.sqc, dbconn.sqC	如何連接資料庫及切斷與資料庫的連接。
dbcreate.c, dbcreate.C	如何建立及捨棄資料庫。
dbinfo.c, dbinfo.C	如何在資料庫層次上取得及設定資訊。
dbinline.sqc	如何使用列入 SQL 程序語言。
dbmcon.sqc, dbmcon.sqC	如何連接多個資料庫及中斷連線。
dbmcon1.h, dbmcon1.h	dbmcon1.sqc、dbmcon1.sqC 的標頭檔
dbmcon1.sqc, dbmcon1.sqC	dbmcon.sqc、dbmcon.sqC 的支援檔案。
dbmcon2.h, dbmcon2.h	dbmcon2.sqc、dbmcon2.sqC 的標頭檔
dbmcon2.sqc, dbmcon2.sqC	dbmcon.sqc、dbmcon.sqC 的支援檔案。
dbmigrat.c, dbmigrat.C	如何移轉資料庫。
dbpkg.sqc, dbpkg.sqC	如何使用資料包。
dbrecov.sqc, dbrecov.sqC	如何回復資料庫。
dbsample.sqc	如何建立包括主電腦及 AS/400 表格及概略表的範例資料庫。
dbsnap.c, dbsnap.C	如何在資料庫層次中擷取 snapshot。
dbthrds.sqc, dbthrds.sqC	如何使用執行緒。
dbuse.sqc, dbuse.sqC	如何使用資料庫物件。
表格空間層次 - 處理 DB2 之表格空間層次的範例。	
tscreate.sqc, tscreate.sqC	如何建立及捨棄緩衝池及表格空間。
tsinfo.sqc, tsinfo.sqC	如何在表格空間層次取得資訊。

表格層次 - 處理 DB2 中表格物件的範例。

表 5. C/C++ 範例程式檔 (繼續)

範例程式名稱	程式說明
tbconstr.sqc, tbconstr.sqC	如何使用表格限制。
tbcreate.sqc, tbcreate.sqC	如何建立、變更及除去表格。
tbident.sqc	如何使用身分直欄。
tbinfo.sqc, tbinfo.sqC	如何在表格層次上取得及設定資訊。
tbmod.sqc, tbmod.sqC	如何修改表格中的資訊。
tbmovesqc, tbmove.sqC	如何移動表格資料。
tbpriv.sqc, tbpriv.sqC	如何授與/顯示/取消表格層次專用權。
tbread.sqc, tbread.sqC	如何讀取表格中的資訊。
tbreorg.sqc, tbreorg.sqC	如何重組表格。
tbsavept.sqc	如何使用外部儲存點。
tbtemp.sqc	如何使用已宣告的暫時表格。
tbtrig.sqc, tbtrig.sqC	如何在表格上使用觸發函式。
資料類型層次 - 處理資料類型的範例。	
dtformat.sqc	如何使用載入及匯入資料格式擴充。
dtlob.sqc, dtlob.sqC	如何讀取及寫入 LOB 資料。
dtstruct.sqc	如何建立、使用及捨棄結構化類型及類型化表格的階層結構。
dtudt.sqc, dtudt.sqC	如何建立、使用及捨棄使用者定義的特殊類型。
DB2 函數層次	
fnuse.sqc, fnuse.sqC	如何使用 SQL 函數。
儲存程序層次 - 示範儲存程序的範例。	
spcat	spserver 程式的儲存程序型錄 Script。這個 Script 會呼叫 spdrop.db2 及 spcreate.db2。
spcreate.db2	用來發出 CREATE PROCEDURE 陳述式，CLP Script。
spdrop.db2	從型錄中捨棄儲存程序的 CLP Script。
spclient.sqc, spclient.sqC	用來呼叫在 spserver.sqc、spserver.sqC 中宣告之伺服器常式的從屬站程式。

表 5. C/C++ 範例程式檔 (繼續)

範例程式名稱	程式說明
spserver.sqc, spserver.sqC	在伺服器上建置及執行的儲存程序常式。
UDF 層次 - 示範使用者定義的函數的範例。	
udfcli.sqc, udfcli.sqC	在 udfsrv.c、udfsrv.C 中呼叫使用者定義的函數的從屬站應用程式。
udfsrv.c, udfsrv.C	udfcli.sqc、udfcli.sqC 所呼叫的使用者定義的函數 ScalarUDF。
udfemcli.sqc, udfemcli.sqC	呼叫內嵌式 SQL 使用者定義的函數檔案庫 udfemsrv 的從屬站應用程式。
udfemsrv.sqc, udfemsrv.sqC	udfemcli 所呼叫的內嵌式 SQL 使用者定義的函數檔案庫。
其它	
utilsnap.c, utilsnap.C	snapshot 監督程式範例的公用程式。

相關概念:

- 第 53 頁的『範例檔案』
- 第 58 頁的『範例程式：結構及設計』

DB2 CLI 範例

表 6. 範例 CLI 程式檔

範例程式名稱	程式說明
教學指導範例 - 示範基本資料庫作業的程式。	
tut_mod.c	如何修改表格資料。
tut_read.c	如何讀取表格。
tut_use.c	如何使用資料庫。
安裝映像檔層次 - 處理 DB2 及 CLI 的安裝映像檔層次範例。	
ilinfo.c	如何取得及設定安裝層次資訊 (如 CLI 驅動程式的版本)。
從屬站層次 - 處理 DB2 之從屬站層次的範例。	
cli_info.c	如何取得及設定從屬站層次資訊。
clihandl.c	如何配置及釋放 handle。
clisqlca.c	如何使用 SQLCA 資料。
案例層次 - 處理 DB2 之案例層次的範例。	
ininfo.c	如何取得及設定案例層次資訊。

表 6. 範例 CLI 程式檔 (繼續)

範例程式名稱	程式說明
資料庫層次 - 處理 DB2 中資料庫物件的範例。	
dbcongui.c	如何利用「圖形式使用者介面 (GUI)」連接資料庫。
dbconn.c	如何連接資料庫及切斷與資料庫的連接。
dbinfo.c	如何在資料庫層次上取得及設定資訊。
dbmcon.c	如何連接多個資料庫及中斷連線。
dbmconx.c	如何利用內嵌式 SQL 連接多個資料庫及中斷連線。
dbmconx1.h	dbmconx1.sqc 的標頭檔。
dbmconx1.sqc	dbmconx 程式的內嵌式 SQL 檔案。
dbmconx2.h	dbmconx2.sqc 的標頭檔。
dbmconx2.sqc	dbmconx 程式的內嵌式 SQL 檔案。
dbnative.c	如何將含有 ODBC escape 子句的陳述式轉換為資料來源特定格式。
dbuse.c	如何使用資料庫物件。
dbusemx.sqc	資料庫物件如何與內嵌式 SQL 一起使用。
表格層次 - 處理 DB2 中表格物件的範例。	
tbconstr.c	如何使用表格限制。
tbcreate.c	如何建立、變更及捨棄表格。
tbinfo.c	如何在表格層次上取得及設定資訊。
tbmod.c	如何修改表格中的資訊。
tbread.c	如何讀取表格中的資訊。
資料類型層次 - 處理資料類型的範例。	
dtinfo.c	如何取得資料類型的相關資訊。
dtlob.c	如何讀取及寫入 LOB 資料。
dtudt.c	如何建立、使用及捨棄使用者定義的特殊類型。
儲存程序層次 - 示範儲存程序的範例。	
spcat	spserver 程式的儲存程序型錄 Script。這個 Script 會呼叫 spdrop.db2 及 spcreate.db2。
spcreate.db2	用來發出 CREATE PROCEDURE 陳述式，CLP Script。
spdrop.db2	從型錄中捨棄儲存程序的 CLP Script。
spclient.c	用來呼叫在 spserver.c 中宣告之伺服器函數的從屬站程式。
spserver.c	在伺服器上建置及執行的儲存程序函數。
spclires.c	示範多個結果集的 SQLMoreResults 與 SQLNextResults 之間差異的從屬站應用程式。
spcall.c	呼叫任何儲存程序的從屬站程式。

表 6. 範例 CLI 程式檔 (繼續)

範例程式名稱	程式說明
UDF 層次 - 示範使用者定義函數的範例。	
udfcli.c	呼叫 udfsrv.c 中使用者定義函數的從屬站應用程式。
udfsrv.c	udfcli.c 所呼叫的使用者定義的函數 ScalarUDF。
共同公用程式檔	
utilcli.c	CLI 範例中所使用的公用程式函數。
utilcli.h	CLI 範例中所使用的公用程式函數的標頭檔。

相關概念:

- 第 53 頁的『範例檔案』
- 第 58 頁的『範例程式：結構及設計』

JDBC 範例

表 7. 範例 JDBC 程式檔

範例程式名稱	程式說明
教學指導範例 - 示範基本資料庫作業的程式。	
TutMod.java	如何修改表格資料。
TutRead.java	如何讀取表格。
安裝映像檔層次 - 處理 DB2 之安裝映像檔層次的範例。	
IlInfo.java	如何取得及設定安裝層次資訊。
資料庫層次 - 處理 DB2 中資料庫物件的範例。	
DbAuth.java	如何授與/顯示/取消資料庫層次中的權限。
DbConn.java	如何連接資料庫及切斷與資料庫的連接。
DbInfo.java	如何在資料庫層次上取得及設定資訊。
DbMCon.java	如何連接多個資料庫及中斷連線。
DbNative.java	如何將含有 ODBC escape 子句的陳述式轉換為資料來源特定格式。
DbSeq.java	如何建立、變更及捨棄資料庫中的順序。
DbUse.java	如何使用資料庫物件。
表格層次 - 處理 DB2 中表格物件的範例。	
TbConstr.java	如何使用表格限制。
TbCreate.java	如何建立、變更及除去表格。
TbGenCol.java	如何使用產生的直欄。
TbIdent.java	如何使用「身分直欄」。

表 7. 範例 JDBC 程式檔 (繼續)

範例程式名稱	程式說明
TbInfo.java	如何在表格層次上取得及設定資訊。
TbMod.java	如何修改表格中的資訊。
TbPriv.java	如何授與/顯示/取消表格層次專用權。
TbRead.java	如何讀取表格中的資訊。
TbTemp.java	如何使用「已宣告的暫時表格」。
TbTrig.java	如何在表格上使用觸發函式。
資料類型層次 - 處理資料類型的範例。	
DtInfo.java	如何取得資料類型的相關資訊。
DtLob.java	如何讀取及寫入 LOB 資料。
DtUdt.java	如何建立、使用及捨棄使用者定義的特殊類型。
Applet - 示範 applet 的範例。	
Applt.java	如何使用 applet。
儲存程序 - 示範儲存程序的範例。	
spscat	spserver 程式的儲存程序型錄 Script。這個 Script 會呼叫 SpDrop.db2 及 SpCreate.db2。
SpCreate.db2	用來發出 CREATE PROCEDURE 陳述式，CLP Script。
SpDrop.db2	從型錄中捨棄儲存程序的 CLP Script。
SpClient.java	用來呼叫在 SpServer.java 中宣告之伺服器函數的從屬站程式。
SpServer.java	在伺服器上建置及執行的儲存程序函數。
UDFs - 示範使用者定義的函數的範例。	
UDFcli.java	呼叫使用者定義的函數檔案庫 UDFsrv 的從屬站應用程式。
UDFsrv.java	UDFcli.java 所呼叫的使用者定義的函數。
udfcat	UDFsrv 程式的 UDF 型錄 Script。這個 Script 會呼叫 UDFDrop.db2 及 UDFCreate.db2。
UDFDrop.db2	從型錄捨棄 UDF 的 CLP Script。
UDFCreate.db2	用來發出 CREATE PROCEDURE 陳述式，CLP Script。
UDFjcli.java	呼叫使用者定義的函數檔案庫 UDFjsrv 的從屬站應用程式。
UDFjsrv.java	UDFjcli.java 所呼叫的使用者定義的函數。
udfjcat	UDFjsrv 程式的 UDF 型錄 Script。這個 Script 會呼叫 UDFjDrop.db2 及 UDFjCreate.db2。
UDFjDrop.db2	從型錄捨棄 UDF 的 CLP Script。
UDFjCreate.db2	用來發出 CREATE PROCEDURE 陳述式，CLP Script。
UDFsCreate.db2	如何將 UDFsqlsv.java 中所含有的 UDF 編目

表 7. 範例 JDBC 程式檔 (繼續)

範例程式名稱	程式說明
UDFsDrop.db2	如何將 UDFsqlsv.java 中所含有的 UDF 解除編目
UDFsqlcl.java	在 UDFsqlsv.java 中呼叫 UDF
UDFsqlsv.java	UDFsqlcl.java 所呼叫、具有 SQL 陳述式的使用者定義的函數
Java Bean - 示範 Java Bean 類別的範例。	
CreateEmployee.java	如何建立員工記錄。
GeneratePayroll.java	如何按部門產生薪資報表。

相關概念:

- 第 101 頁的『Java 範例程式』
- 第 53 頁的『範例檔案』
- 第 58 頁的『範例程式：結構及設計』

相關參照:

- 第 72 頁的『SQLJ 範例』
- 第 74 頁的『Java WebSphere 範例』
- 第 75 頁的『Java 外掛程式範例』

SQLJ 範例

表 8. 範例 SQLJ 程式檔

範例程式名稱	程式說明
教學指導範例 - 示範基本資料庫作業的程式。	
TutMod.sqlj	如何修改表格資料。
TutRead.sqlj	如何讀取表格。
資料庫層次 - 處理 DB2 中資料庫物件的範例。	
DbAuth.sqlj	如何授與/顯示/取消資料庫層次中的權限。
DbConn.sqlj	如何連接資料庫及切斷與資料庫的連接。
DbMCon.java	如何連接多個資料庫及中斷連線。
DbMCon1.sqlj	DbMCon.java 的支援檔案。
DbMCon2.sqlj	DbMCon.java 的支援檔案。
DbUse.sqlj	如何使用資料庫物件。
表格層次 - 處理 DB2 中表格物件的範例。	
TbConstr.sqlj	如何使用表格限制。

表 8. 範例 SQLJ 程式檔 (繼續)

範例程式名稱	程式說明
TbCreate.sqlj	如何建立、變更及除去表格。
TbIdent.sqlj	如何使用身分直欄。
TbInfo.sqlj	如何在表格層次上取得及設定資訊。
TbMod.sqlj	如何修改表格中的資訊。
TbPriv.sqlj	如何授與/顯示/取消表格層次專用權。
TbRead.sqlj	如何讀取表格中的資訊。
TbTrig.sqlj	如何在表格上使用觸發函式。
資料類型層次 - 處理資料類型的範例。	
DtUdt.sqlj	如何建立、使用及捨棄使用者定義的特殊類型。
Applet 層次 - 示範 applet 的範例。	
Applt.sqlj	如何使用 applet。
儲存程序層次 - 示範儲存程序的範例。	
spcat	SpServer 程式的儲存程序型錄 Script。這個 Script 會呼叫 SpDrop.db2 及 SpCreate.db2。
SpCreate.db2	用來發出 CREATE PROCEDURE 陳述式，CLP Script。
SpDrop.db2	從型錄中捨棄儲存程序的 CLP Script。
SpClient.sqlj	用來呼叫在 SpServer.sqlj 中宣告之伺服器函數的從屬站程式。
SpServer.sqlj	在伺服器上建置及執行的儲存程序函數。
SpIterat.sqlj	SpServer.sqlj 的疊代類別檔。
UDF 層次 - 示範使用者定義的函數的範例。	
UDFcli.sqlj	呼叫使用者定義的函數檔案庫 UDFsrv 的從屬站應用程式。
UDFsrv.java	UDFcli 所呼叫的使用者定義的函數。
udfcats	UDFsrv 程式的 UDF 型錄 Script。這個 Script 會呼叫 UDFDrop.db2 及 UDFCreate.db2。
UDFDrop.db2	從型錄捨棄 UDF 的 CLP Script。
UDFCreate.db2	用來發出 CREATE PROCEDURE 陳述式，CLP Script。
UDFjcli.sqlj	呼叫使用者定義的函數檔案庫 UDFjsrv 的從屬站應用程式。
UDFjsrv.java	UDFjcli 所呼叫的使用者定義的函數。
udfjcats	UDFjsrv 程式的 UDF 型錄 Script。這個 Script 會呼叫 UDFjDrop.db2 及 UDFjCreate.db2。
UDFjDrop.db2	從型錄捨棄 UDF 的 CLP Script。
UDFjCreate.db2	用來發出 CREATE PROCEDURE 陳述式，CLP Script。

Java Bean - 示範 Java Bean 類別的範例。

表 8. 範例 SQLJ 程式檔 (繼續)

範例程式名稱	程式說明
CreateEmployee.sqlj	如何建立員工記錄。
GeneratePayroll.sqlj	如何按部門產生薪資報表。

相關概念:

- 第 101 頁的『Java 範例程式』
- 第 53 頁的『範例檔案』
- 第 58 頁的『範例程式：結構及設計』

相關參照:

- 第 70 頁的『JDBC 範例』
- 第 74 頁的『Java WebSphere 範例』
- 第 75 頁的『Java 外掛程式範例』

Java WebSphere 範例

表 9. Java WebSphere 範例檔

範例程式名稱	程式說明
AccessEmployee.ear	這個 Enterprise ARchive (.EAR) 檔是由四個含有 32 個不同 .class、.JSP 及 .HTML 檔之模組所組成。可以輕易地使用 IBM WebSphere Application Server 部署的這個 EAR 檔示範 Java 從屬站如何與 Enterprise Java Beans (EJB) 互動，以存取 DB2 中所儲存的資料。AccessEmployee.ear 檔位於 samples/java/WebSphere 目錄中。

相關概念:

- 第 101 頁的『Java 範例程式』
- 第 53 頁的『範例檔案』
- 第 58 頁的『範例程式：結構及設計』

相關參照:

- 第 70 頁的『JDBC 範例』
- 第 72 頁的『SQLJ 範例』
- 第 75 頁的『Java 外掛程式範例』

Java 外掛程式範例

表 10. Java 控制中心外掛程式範例檔

範例程式名稱	程式說明
Example1.java	如何新增新的工具列按鈕到「控制中心」工具列。
Example2.java	如何新增新的功能表動作到「控制中心資料庫」物件。
Example3.java	如何在「控制中心」樹狀結構中的「資料庫」物件下新增新的物件。
Example3Folder.java	如何在「控制中心」樹狀結構中的「資料庫」物件下新增新的物件。
Example4.java	如何除去「資料庫」及「案例」物件的「架構」功能表項目。
Example5.java	如何從「表格」除去「變更」動作。
Example6.java	如何停用「資料庫架構」預設按鈕。

相關概念:

- *Administration Guide: Implementation* 中的『Introducing the plug-in architecture for the Control Center』
- *Administration Guide: Implementation* 中的『Control Center plug-in performance considerations』
- *Administration Guide: Implementation* 中的『Compiling and running the example plugins』
- 第 101 頁的『Java 範例程式』
- 第 53 頁的『範例檔案』
- 第 58 頁的『範例程式：結構及設計』
- *Administration Guide: Implementation* 中的『Writing plugins as Control Center extensions』

相關作業:

- *Administration Guide: Implementation* 中的『Creating a plugin that adds a toolbar button』
- *Administration Guide: Implementation* 中的『Setting attributes for a plugin tree object』

相關參照:

- 第 70 頁的『JDBC 範例』
- 第 72 頁的『SQLJ 範例』
- 第 74 頁的『Java WebSphere 範例』

COBOL 範例

註: COBOL 範例不是建構在用於 C、CLI、C++ 及 Java 範例的 DB2 層次設計。

表 11. 沒有內嵌式 SQL 的 COBOL DB2 API 範例程式

範例程式	內含 API
checkerr.cbl	<ul style="list-style-type: none">• sqlaintp - 取得錯誤訊息• sqllogstt - 取得 SQLSTATE 訊息
client.cbl	<ul style="list-style-type: none">• sqlqryc - 查詢從屬站• sqlesetc - 設定從屬站
d_dbconf.cbl	<ul style="list-style-type: none">• sqleatin - 連接• sqledtin - 分離• sqlfddb - 取得資料庫架構預設值
d_dbmcon.cbl	<ul style="list-style-type: none">• sqleatin - 連接• sqledtin - 分離• sqlfdsys - 取得資料庫管理程式架構預設值
db_udcs.cbl	<ul style="list-style-type: none">• sqleatin - 連接• sqlecrea - 建立資料庫• sqledrpd - 捨棄資料庫
dbcat.cbl	<ul style="list-style-type: none">• sqlecadb - 編目資料庫• sqledcls - 關閉資料庫目錄掃描• sqledgne - 取得下一個資料庫目錄登錄• sqledosd - 開啓資料庫目錄掃描• sqleuncd - 取消編目資料庫
dbcmt.cbl	<ul style="list-style-type: none">• sqledcgd - 變更資料庫註解• sqledcls - 關閉資料庫目錄掃描• sqledgne - 取得下一個資料庫目錄登錄• sqledosd - 開啓資料庫目錄掃描• sqleisig - 安裝信號處理程式

表 11. 沒有內嵌式 SQL 的 COBOL DB2 API 範例程式 (繼續)

範例程式	內含 API
dbconf.cb1	<ul style="list-style-type: none"> • sqleatin - 連接 • sqlcrea - 建立資料庫 • sqldrpd - 捨棄資料庫 • sqlfrdb - 重設資料庫架構 • sqlfudb - 更新資料庫架構 • sqlfxdb - 取得資料庫架構
dbinst.cb1	<ul style="list-style-type: none"> • sqleatcp - 連接及變更通行碼 • sqleatin - 連接 • sqledtin - 分離 • sqlgins - 取得案例
dbmconf.cb1	<ul style="list-style-type: none"> • sqleatin - 連接 • sqledtin - 分離 • sqlfrsys - 重設資料庫管理程式架構 • sqlfusys - 更新資料庫管理程式架構 • sqlfxsys - 取得資料庫管理程式架構
dbsnap.cb1	<ul style="list-style-type: none"> • sqleatin - 連接 • sqlmonss - 取得 Snapshot
dbstart.cb1	<ul style="list-style-type: none"> • sqlpstart - 啓動資料庫管理程式
dbstop.cb1	<ul style="list-style-type: none"> • sqlfrce - 強迫終止應用程式 • sqlpstp - 停止資料庫管理程式
dcscat.cb1	<ul style="list-style-type: none"> • sqlgdad - 編目 DCS 資料庫 • sqlgdcl - 關閉 DCS 目錄掃描 • sqlgdcl - 解編 DCS 資料庫 • sqlgdge - 取得資料庫的 DCS 目錄登錄 • sqlgdgt - 取得 DCS 目錄登錄 • sqlgdsc - 開啓 DCS 目錄掃描
ebcdicdb.cb1	<ul style="list-style-type: none"> • sqleatin - 連接 • sqlcrea - 建立資料庫 • sqldrpd - 捨棄資料庫
migrate.cb1	<ul style="list-style-type: none"> • sqlmgdb - 移轉資料庫

表 11. 沒有內嵌式 SQL 的 COBOL DB2 API 範例程式 (繼續)

範例程式	內含 API
monreset.cbl	<ul style="list-style-type: none"> • sqlcatin - 連接 • sqlmrset - 重設監督程式
monsz.cbl	<ul style="list-style-type: none"> • sqlcatin - 連接 • sqlmonss - 取得 Snapshot • sqlmonsz - 估計 sqlmonss() 輸出緩衝區所需的大小
nodecat.cbl	<ul style="list-style-type: none"> • sqlctnd - 編目節點 • sqlencls - 關閉節點目錄掃描 • sqlengne - 取得下一個節點目錄登錄 • sqlenops - 開啓節點目錄掃描 • sqluncn - 解編節點
restart.cbl	<ul style="list-style-type: none"> • sqlerstd - 重新啓動資料庫
setact.cbl	<ul style="list-style-type: none"> • sqlsact - 設定帳戶字串
sws.cbl	<ul style="list-style-type: none"> • sqlcatin - 連接 • sqlmon - 取得/更新監督程式開關

表 12. COBOL DB2 API 內嵌式 SQL 範例程式

範例程式	內含 API
dbauth.sqb	<ul style="list-style-type: none"> • sqluadcu - 取得授權
dbstat.sqb	<ul style="list-style-type: none"> • db2Reorg - 重組表格 • db2Runstats - 執行統計
expsamp.sqb	<ul style="list-style-type: none"> • sqluexpr - 匯出 • sqluimpr - 匯入
impexp.sqb	<ul style="list-style-type: none"> • sqluexpr - 匯出 • sqluimpr - 匯入
loadqry.sqb	<ul style="list-style-type: none"> • db2LoadQuery - 載入查詢
rebind.sqb	<ul style="list-style-type: none"> • sqlarbdn - 重新連結

表 12. COBOL DB2 API 內嵌式 SQL 範例程式 (繼續)

範例程式	內含 API
tabscont.sqb	<ul style="list-style-type: none"> • sqlbctcq - 關閉表格空間配置區查詢 • sqlbftcq - 提取表格空間配置區查詢 • sqlbotcq - 開啓表格空間配置區查詢 • sqlbtcq - 表格空間配置區查詢 • sqlfmem - 可用記憶體
tabspace.sqb	<ul style="list-style-type: none"> • sqlbctsq - 關閉表格空間查詢 • sqlbftpq - 提取表格空間查詢 • sqlbgts - 取得表格空間統計值 • sqlbmtsq - 表格空間查詢 • sqlbotsq - 開啓表格空間查詢 • sqlbstpq - 單一表格空間查詢 • sqlfmem - 可用記憶體
tload.sqb	<ul style="list-style-type: none"> • sqluexpr - 匯出 • sqluload - 載入 • sqluvqdp - 停止表格的表格空間
tspace.sqb	<ul style="list-style-type: none"> • sqlbctcq - 關閉表格空間配置區查詢 • sqlbctsq - 關閉表格空間查詢 • sqlbftcq - 提取表格空間配置區查詢 • sqlbftpq - 提取表格空間查詢 • sqlbgts - 取得表格空間統計值 • sqlbmtsq - 表格空間查詢 • sqlbotcq - 開啓表格空間配置區查詢 • sqlbotsq - 開啓表格空間查詢 • sqlbstpq - 單一表格空間查詢 • sqlbstsc - 設定表格空間配置區 • sqlbtcq - 表格空間配置區查詢 • sqlfmem - 可用記憶體

表 13. 沒有 DB2 API 的 COBOL 內嵌式 SQL 範例程式

範例程式名稱	程式說明
advsql.sqb	示範如何使用進階的 SQL 表示式，如 CASE、CAST 及純量全選。
cursor.sqb	示範如何使用靜態 SQL 來使用游標。

表 13. 沒有 DB2 API 的 COBOL 內嵌式 SQL 範例程式 (繼續)

範例程式名稱	程式說明
delet.sqb	示範將從資料庫中刪除項目的靜態 SQL。
dynamic.sqb	示範如何使用動態 SQL 來使用游標。
joinsql.sqb	示範如何使用進階 SQL 結合表示式。
lobeval.sqb	示範 LOB 定位器的用途，以及延遲實際 LOB 資料的評估。
lobfile.sqb	示範 LOB 檔案 handle 的用途。
lobloc.sqb	示範 LOB 定位器的用途。
openftch.sqb	示範如何使用靜態 SQL 來提取、更新及刪除橫列。
static.sqb	示範靜態 SQL 如何擷取資訊。
tabsql.sqb	示範進階 SQL 表格表示式的使用方式。
trigsq1.sqb	示範如何使用進階 SQL 觸發函式及限制。
updat.sqb	示範如何使用靜態 SQL 來更新資料庫。
varinp.sqb	示範如何使用參數記號，將變數輸入「內嵌式動態 SQL」陳述式呼叫中。

相關概念:

- 第 53 頁的『範例檔案』

SQL 程序範例

表 14. SQL 程序範例程式檔

範例程式名稱	程式說明
basecase.db2	UPDATE_SALARY 程序會提升 "sample" 資料庫中 "staff" 表格的 "empno" IN 參數所識別的員工薪資。程序會根據使用 "rating" IN 參數的 CASE 陳述式，判斷提升幅度。
basecase.sqc	呼叫 UPDATE_SALARY 程序。
baseif.db2	UPDATE_SALARY_IF 程序會提升 "sample" 資料庫中 "staff" 表格的 "empno" IN 參數所識別的員工薪資。程序會根據使用 "rating" IN 參數的 IF 陳述式，判斷提升幅度。
baseif.sqc	呼叫 UPDATE_SALARY_IF 程序。
dynamic.db2	CREATE_DEPT_TABLE 程序會使用動態 DDL 來建立新的表格。表格的名稱取決於程序的 IN 參數值。
dynamic.sqc	呼叫 CREATE_DEPT_TABLE 程序。

表 14. SQL 程序範例程式檔 (繼續)

範例程式名稱	程式說明
iterate.db2	ITERATOR 程序會使用 FETCH 迴圈，從 "department" 表格中擷取資料。如果 "deptno" 直欄的值不是 'D11'，則會將修改的資料插入 "department" 表格。如果 "deptno" 直欄的值是 'D11'，則 ITERATE 陳述式會將控制串流傳回 LOOP 陳述式的開頭。
iterate.sqc	呼叫 ITERATOR 程序。
leave.db2	LEAVE_LOOP 程序會計算在 "not_found" 條件 handler 呼叫 LEAVE 陳述式之前，在 LOOP 陳述式中執行的 FETCH 作業數。LEAVE 陳述式會讓控制串流跳出迴圈，並完成儲存程序。
leave.sqc	呼叫 LEAVE_LOOP 程序。
loop.db2	LOOP_UNTIL_SPACE 程序會計算在游標擷取到一系列值為空格 (' ') 的直欄 "midinit" 之前，在 LOOP 陳述式中執行的 FETCH 作業數。loop 陳述式會讓控制串流跳出迴圈，並完成儲存程序。
loop.sqc	呼叫 LOOP_UNTIL_SPACE 程序。
nestcase.db2	BUMP_SALARY 程序會使用巢狀 CASE 陳述式，提升 "sample" 資料庫中 "staff" 表格的 dept IN 參數所識別的部門員工薪資。
nestcase.sqc	呼叫 BUMP_SALARY 程序。
nestif.db2	BUMP_SALARY_IF 程序會使用巢狀 IF 陳述式，提升 "sample" 資料庫中 "staff" 表格的 dept IN 參數所識別的部門員工薪資。
nestif.sqc	呼叫 BUMP_SALARY_IF 程序。
nestedsp.db2	OUT_AVERAGE、OUT_MEDIAN 及 MAX_SALARY 程序會從範例資料庫的 "staff" 表格傳回平均值、中位值及最大值。
NestedSP.java	呼叫 OUT_AVERAGE 程序。
repeat.db2	REPEAT_STMT 程序會計算在游標無法再擷取更多橫列之前，在 repeat 陳述式中執行的 FETCH 作業數。條件 handler 會讓控制串流跳出重複的迴圈，並完成儲存程序。
repeat.sqc	呼叫 REPEAT_STMT 程序。
resultset.c	呼叫 MEDIAN_RESULT_SET 程序，顯示中間薪資，然後顯示 SQL 程序所產生的結果集。此從屬站是使用 CLI API 撰寫的，可以接受結果集。
resultset.db2	MEDIAN_RESULT_SET 程序會取得 "sample" 資料庫中 "staff" 表格的 "dept" IN 參數所識別的部門員工之中間薪資。中間值會被指派為薪資 OUT 參數，並傳回 "resultset" 從屬站。然後程序會開啓 WITH RETURN 游標，傳回其薪資大於中間值的員工結果集。程序會將結果集傳回從屬站。
spserver.db2	此 CLP Script 中的 SQL 程序會示範基本錯誤處理常式、巢狀儲存程序呼叫，並將結果集傳回從屬站應用程式或呼叫應用程式。您可以使用 CLI 範例目錄中的 "spcall" 應用程式，來呼叫程序。也可以使用 "spclient" 應用程式 (位於 C 及 CPP 範例目錄)，呼叫沒有傳回結果集的程序。

表 14. SQL 程序範例程式檔 (繼續)

範例程式名稱	程式說明
whiles.db2	DEPT_MEDIAN 程序會取得 "sample" 資料庫中 "staff" 表格的 "dept" IN 參數所識別的部門員工之中間薪資。中間值會被指派給薪資 OUT 參數，並傳回 "whiles" 從屬站。然後，whiles 從屬站會列印中間薪資。
whiles.sqc	呼叫 DEPT_MEDIAN 程序。

相關概念:

- 第 53 頁的『範例檔案』

Visual Basic 範例

表 15. Visual Basic ADO 範例程式檔

範例程式名稱	程式說明
從屬站層次	
cliExeSQL.bas	如何執行 SQL 陳述式。
cli_Info.bas	如何取得/設定從屬站層次資訊。
資料庫層次	
dbConn.bas	如何連接資料庫/中斷連線。
dbInfo.bas	如何在資料庫層次上取得及設定資訊。
dbCommit.bas	如何在資料庫層次上動態控制自動確定。
資料類型層次	
dtHier.bas	如何擷取階層式資料。
dtLob.bas	如何讀取及寫入 LOB 資料。
儲序程序	
spCall.bas	如何呼叫儲存程序。
使用者定義的函數	
udfUse.bas	如何建立及使用 UDT 及 UDF。

表 16. Visual Basic RDO 及 MTS 範例程式檔

範例程式名稱	程式說明
Bank.vbp	一種 RDO 程式，可建立及維護銀行分行的資料，能夠在客戶帳戶上執行交易。此程式可以利用使用者指定的資料庫，因為包含 DDL 來建立必要的表格，供應用程式用來儲存資料。這個程式的檔案位於 samples\VB\RDO。

表 16. Visual Basic RDO 及 MTS 範例程式檔 (繼續)

範例程式名稱	程式說明
db2com.vbp	<p>此 Visual Basic 專案示範如何使用 Microsoft 交易伺服器來更新資料庫。專案會建立從屬站程式 db2mts.vbp 所用的伺服器 DLL，以及四個類別模組：</p> <ul style="list-style-type: none"> • UpdateNumberColumn.cls • UpdateRow.cls • UpdateStringColumn.cls • VerifyUpdate.cls <p>此程式中，sample 資料庫內會建立一個暫時表格 DB2MTS。這個程式的檔案位於 samples\VB\MTS。</p>
db2mts.vbp	<p>此為從屬站程式的 Visual Basic 專案，使用 Microsoft 交易伺服器來呼叫由 db2com.vbp 所建立的伺服器 DLL。這個程式的檔案位於 samples\VB\MTS。</p>

相關概念:

- 第 53 頁的『範例檔案』
- 第 58 頁的『範例程式：結構及設計』

相關參照:

- 第 84 頁的『Windows Management Instrumentation 範例』

Visual C++ 範例

表 17. Visual C++ 範例程式檔

範例程式名稱	程式說明
BLOBAccess.dsw	<p>此範例示範如何使用 Microsoft Visual C++ 來高亮度標示 ADO/Blob 存取。類似於 Visual Basic 範例 Blob.vbp。BLOB 範例具有兩個 main 函數：</p> <ol style="list-style-type: none"> 1. 從 Sample 資料庫中讀取 BLOB，顯示在螢幕上。 2. 從檔案中讀取 BLOB，並插入資料庫中。(匯入)
VarChar.dsp	<p>一個 Visual C++ 程式，使用 ADO 將 VarChar 資料當作文字欄位來存取。提供圖形式使用者介面，供使用者用來檢視及更新 sample 資料庫中 ORG 表格的資料。</p>

相關概念:

- 第 53 頁的『範例檔案』

Windows Management Instrumentation 範例

表 18. Windows Management Instrumentation 範例程式檔

範例檔名	檔案說明
backupdb.vbs	如何備份資料庫
createdb.vbs	如何建立及捨棄資料庫。
listsvr.vbs	如何列舉伺服器案例及啟動/停止 DB2 案例。
perfmon.mof	perfmon.vbs 的 MOF 檔。
perfmon.vbs	如何取得 DB2 效能計數器。備註：首先您必須執行 "mofcomp perfmon.mof"。
regvar.mof	regvar.vbs 的 MOF 檔。
regvar.vbs	如何取得 DB2 登錄變數。備註：首先您必須執行 "mofcomp regvar.mof"。
restoredb.vbs	如何復置資料庫。
rollfwddb.vbs	如何 RollForward 資料庫。
updatedbcfg.vbs	如何取得及更新資料庫架構。
updatedbmcfg.vbs	如何取得及更新資料庫管理程式架構。

相關概念:

- 第 53 頁的『範例檔案』
- 第 277 頁的『Windows Management Instrumentation (WMI)』

相關參照:

- 第 82 頁的『Visual Basic 範例』

物件連結和內嵌 (OLE) 範例

表 19. 物件連結和內嵌 (OLE) 範例程式

範例程式名稱	程式說明
sales	示範有關 Microsoft Excel 銷售試算表的向上捲動查詢 (使用 Visual Basic)。
names	查詢 Lotus Notes 通訊錄 (使用 Visual Basic)。
inbox	透過 OLE/Messaging 查詢 Microsoft Exchange 信箱電子郵件訊息 (使用 Visual Basic)。
invoice	以電子郵件附件的形式傳送 Microsoft Word 發票文件的 OLE 自動化使用者定義的函數 (使用 Visual Basic)。
bcounter	一種 OLE 自動化使用者定義的函數，示範使用案例變數的 scratch pad (在 Visual Basic 中執行)。
ccounter	計數器 OLE 自動化使用者定義的函數 (使用 Visual C++)。

表 19. 物件連結和內嵌 (OLE) 範例程式 (繼續)

範例程式名稱	程式說明
salarysrv	計算 sample 資料庫 STAFF 表格內中間薪資的 OLE 自動化儲存程序 (使用 Visual Basic)。
salarycltvc	一種 Visual C++ DB2 CLI 範例，會呼叫 Visual Basic 儲存程序 salarysrv。
salarycltvb	一種 Visual Basic DB2 CLI 範例，會呼叫 Visual Basic 儲存程序 salarysrv。
salsvado	一種 OLE 自動化儲存程序 (在 32 位元 Visual Basic 及 ADO 中執行)，會藉由計算新建立的表格 STAFF2 中的中間薪資來示範輸出參數，並藉由從該表格擷取薪資來示範結果集。
salcldo	一種 Visual Basic 從屬站，會呼叫 Visual Basic 儲存程序 salsvado。
testcli	一種 OLE 自動化內嵌式 SQL 從屬站應用程式，會呼叫儲存程序 tstsrv (在 Visual C++ 中執行)。
tstsrv	一種 OLE 自動化儲存程序，會示範在從屬站及儲存程序間不同的傳送類型 (在 Visual C++ 中執行)。

相關概念:

- 第 53 頁的『範例檔案』

相關參照:

- 第 85 頁的『物件連結和內嵌資料庫 (OLE DB) 表格函數範例』

物件連結和內嵌資料庫 (OLE DB) 表格函數範例

表 20. 物件連結和內嵌資料庫 (OLE DB) 表格函數

範例程式名稱	程式說明
jet.db2	Microsoft.Jet.OLEDB.3.51 提供者
mapi.db2	MAPI 的 INTERSOLV 連接 OLE DB
msdaora.db2	Oracle 的 Microsoft OLE DB 提供者
msdasql.db2	ODBC 驅動程式的 Microsoft OLE DB 提供者
msidxs.db2	Microsoft OLE DB 索引伺服器提供者
notes.db2	Notes 的 INTERSOLV 連接 OLE DB
samprov.db2	Microsoft OLE DB 範例提供者
sqloledb.db2	SQL 伺服器的 Microsoft OLE DB 提供者

相關概念:

- 第 53 頁的『範例檔案』

相關參照:

- 第 84 頁的『物件連結和內嵌 (OLE) 範例』

命令行處理器 (CLP) 範例

表 21. 命令行處理器 (CLP) 範例 Script。

範例檔名	檔案說明
const.db2	建立一個具有 CHECK CONSTRAINT 子句的表格。
cte.db2	示範一個常用的表格表示式。
flt.db2	示範遞迴查詢。
join.db2	示範表格的外部結合。
stock.db2	示範如何使用起始動作定義。
testdata.db2	使用 DB2 內建函數 (如 RAND() 及 TRANSLATE())，將具有隨機產生的測試資料表格的移入。

相關概念:

- 第 53 頁的『範例檔案』

REXX 範例

表 22. REXX 範例程式檔

範例檔名	檔案說明
blobfile.cmd	示範「二進位大型物件 (BLOB)」操作。
chgis1.cmd	示範 CHANGE ISOLATION LEVEL API。
client.cmd	示範 SET CLIENT 及 QUERY CLIENT API。
d_dbconf	示範 API：GET DATABASE CONFIGURATION DEFAULTS
d_dbmcon	示範 API：GET DATABASE MANAGER CONFIGURATION DEFAULTS
db_udcs	示範 CREATE DATABASE 及 DROP DATABASE API 以模擬 DB2 for MVS/ESA CCSID 500 (EBCDIC 國際化) 對照順序的對照行為
dbauth	示範 GET AUTHORIZATIONS API
dbcatt	示範下列 API： CATALOG DATABASE CLOSE DATABASE DIRECTORY SCAN GET NEXT DATABASE DIRECTORY ENTRY OPEN DATABASE DIRECTORY SCAN UNCATALOG DATABASE
dbcmt	示範下列 API： CHANGE DATABASE COMMENT GET ERROR MESSAGE INSTALL SIGNAL HANDLER

表 22. REXX 範例程式檔 (繼續)

範例檔名	檔案說明
dbconf	<p>示範下列 API :</p> <p>CREATE DATABASE DROP DATABASE GET DATABASE CONFIGURATION RESET DATABASE CONFIGURATION UPDATE DATABASE CONFIGURATION</p>
dbinst	<p>示範下列 API :</p> <p>ATTACH TO INSTANCE DETACH FROM INSTANCE GET INSTANCE</p>
dbmconf	<p>示範下列 API :</p> <p>GET DATABASE MANAGER CONFIGURATION RESET DATABASE MANAGER CONFIGURATION UPDATE DATABASE MANAGER CONFIGURATION</p>
dbstart	<p>示範 START DATABASE MANAGER API</p>
dbstat	<p>示範下列 API :</p> <p>REORGANIZE TABLE RUN STATISTICS</p>
dbstop	<p>示範下列 API :</p> <p>FORCE USERS STOP DATABASE MANAGER</p>
dcscat	<p>示範下列 API :</p> <p>ADD DCS DIRECTORY ENTRY CLOSE DCS DIRECTORY SCAN GET DCS DIRECTORY ENTRY FOR DATABASE GET DCS DIRECTORY ENTRIES OPEN DCS DIRECTORY SCAN UNCATALOG DCS DIRECTORY ENTRY</p>
dynamic	<p>示範如何使用動態 SQL 來使用 "CURSOR"</p>
ebcdicdb	<p>示範 CREATE DATABASE 及 DROP DATABASE API 以模擬 DB2 for MVS/ESA CCSID 037 (EBCDIC 美式英文) 對照順序的對照行為</p>
impexp	<p>示範 EXPORT 及 IMPORT API</p>
lobeval	<p>示範延遲資料庫內的 LOB 計算</p>
lobfile	<p>示範 LOB 檔案 handle 的使用</p>
lobloc	<p>示範 LOB 定位器的使用</p>
lobval	<p>示範 LOB 的使用</p>
migrate	<p>示範 MIGRATE DATABASE API</p>

表 22. REXX 範例程式檔 (繼續)

範例檔名	檔案說明
nodecat	示範下列 API： CATALOG NODE CLOSE NODE DIRECTORY SCAN GET NEXT NODE DIRECTORY ENTRY OPEN NODE DIRECTORY SCAN UNCATALOG NODE
quitab	示範 API：QUIESCE TABLESPACES FOR TABLE
rechist	示範下列 API： CLOSE RECOVERY HISTORY FILE SCAN GET NEXT RECOVERY HISTORY FILE ENTRY OPEN RECOVER HISTORY FILE SCAN PRUNE RECOVERY HISTORY FILE ENTRY UPDATE RECOVERY HISTORY FILE ENTRY
restart	示範 RESTART DATABASE API
sqlecsrx	對照順序的範例
updat	使用動態 SQL 以更新資料庫

相關概念:

- 第 53 頁的『範例檔案』

相關作業:

- 第 186 頁的『在 AIX 上建置 REXX 應用程式』
- 第 308 頁的『在 Windows 上建置 Object REXX 應用程式』

日誌管理使用者跳出程式範例

註: 列示在下列表格中的每一個來源檔頂端都會提供編譯「日誌管理使用者跳出程式」程式的指示。

表 23. 日誌管理使用者跳出範例程式檔。

範例檔名	檔案說明
db2uext2.cadsm	這是一個利用 Tivoli Storage Manager (TSM) API 來保存及擷取資料庫日誌檔的範例「使用者跳出程式」。此範例提供呼叫的審核追蹤 (每一個選項儲存在不同的檔案中), 包括時間戳記和已接收參數。本範例也提供發生錯誤的呼叫之追蹤, 包括時間戳記與用來判斷問題所在之已隔離出錯誤的字串。您可以停用這些選項。該檔案必須更名為 db2uext2.c 並編譯為 C 程式。在 UNIX 及 Windows 作業系統上提供。 註: 在 AIX 上使用 TSM API Client 的應用程式必須使用 xlc_r 或 xlc_r 編譯器呼叫來建立, 而不是使用 xlc 或 xlc, 即使應用程式為單緒。這會確保檔案庫提供安全的緒。如果您擁有一個使用不提供安全緒之檔案庫編譯的應用程式, 則可以引用 Fixtest IC21925E 或與您的應用程式提供者連絡。Fixtest 可在 index.storsys.ibm.com 匿名 FTP 伺服器上找到。
db2uext2.cdisk	此為範例「使用者跳出程式」, 利用所在特定平台的系統複製命令。此程式可保存、擷取資料庫日誌檔, 並提供呼叫的審核追蹤 (每一個選項儲存在不同的檔案中), 包括時間戳記和收到的參數。本範例也提供發生錯誤的呼叫之錯誤追蹤, 包括時間戳記與用來判斷問題所在之已隔離出錯誤的字串。您可以停用這些選項。該檔案必須更名為 db2uext2.c 並編譯為 C 程式。在 UNIX 及 Windows 作業系統上提供。
db2uext2.ctape	此為範例「使用者跳出程式」, 利用所在特定 UNIX 平台的系統磁帶命令。程式可以保存及擷取資料庫日誌檔。系統磁帶命令的所有限制即為此使用者跳出程式的限制。此範例提供呼叫的審核追蹤 (每一個選項儲存在不同的檔案中), 包括時間戳記和已接收參數。本範例也提供發生錯誤的呼叫之錯誤追蹤, 包括時間戳記與用來判斷問題所在之已隔離出錯誤的字串。您可以停用這些選項。該檔案必須更名為 db2uext2.c 並編譯為 C 程式。只適用於 UNIX 平台。
db2uext2.cxbsa	這是一個利用 XBSA API 來「保存」及「擷取」資料庫日誌檔案的範例「使用者跳出程式」。該範例提供呼叫的審核追蹤 (每一個選項儲存在不同的檔案中), 包括時間戳記和已接收參數。本範例也提供發生錯誤的呼叫之錯誤追蹤, 包括時間戳記與用來判斷問題所在之已隔離出錯誤的字串。您可以停用這些選項。該檔案必須更名為 db2uext2.c 並編譯為 C 程式。在 UNIX 及 Windows 作業系統上提供。

相關概念:

- 第 53 頁的『範例檔案』

相關參照:

- *Data Recovery and High Availability Guide and Reference* 中的『Tivoli Storage Manager』

建置檔、make 檔及錯誤檢查公用程式

建置檔

用來示範建置範例程式的檔案在 UNIX 上稱為 Script 檔，在 Windows 上稱為批次檔。我們一般稱這些檔案為建置檔。它們含有建議的編譯及鏈結命令，適合支援的平台編譯器使用。

DB2 在支援的平台上提供各種語言的建置檔，以便使用其中已建置的各類型程式，每一種語言的範例程式都在相同的目錄中。下列表格列示不同類型的建置檔，用於建置不同類型的程式。除非另有指示，不然這些建置檔適用於所有支援平台上的支援語言。Windows 上的建置檔具有 .bat (批次) 副檔名，不包括在表格中。若是 UNIX 平台，則沒有副檔名。

表 24. DB2 建置檔

建置檔	建置的程式類型
bldapp	應用程式
bldrtn	常式 (儲存程序及 UDF)
bldsqlj	Java SQLJ 應用程式
bldsqljs	Java SQLJ 常式 (儲存程序及 UDF)
bldmc	C/C++ 多連線應用程式
bldmt	UNIX C/C++ 多緒應用程式
bldcli	SQL 程序的 CLI 從屬站應用程式，位於 sqlproc samples 子目錄。
blddevm	AIX 及 Windows 事件監督程式 evm (僅限 C 語言)

註: bldcli 檔同於 samples/cli 目錄中的 bldapp 檔。名稱不同的原因在於內含的 C bldapp 檔也包括在 samples/sqlproc 目錄中。

下列表格依平台及程式設計語言，以及它們所在的目錄來列示建置檔。在線上文件中，建置檔名稱是 HTML 中來源檔的熱鏈結。使用者也可以在適當的範例目錄中存取文字檔。

表 25. 按語言及平台的建置檔

平台 → 語言	AIX	HP-UX	Linux	Solaris	Windows
C samples/c	bldapp bldrtn bldmt bldmc bldevm	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp.bat bldrtn.bat bldmc.bat bldevm.bat
C++ samples/cpp	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp bldrtn bldmt bldmc	bldapp.bat bldrtn.bat bldmc.bat
CLI samples/cli	bldapp bldrtn bldmc	bldapp bldrtn bldmc	bldapp bldrtn bldmc	bldapp bldrtn bldmc	bldapp.bat bldrtn.bat bldmc.bat
SQLJ samples/java/sqlj	bldsqlj bldsqljs	bldsqlj bldsqljs	bldsqlj bldsqljs	bldsqlj bldsqljs	bldsqlj.bat bldsqljs.bat
IBM COBOL samples/cobol	bldapp bldrtn	n/a	n/a	n/a	bldapp.bat bldrtn.bat
Micro Focus COBOL samples/cobol_mf	bldapp bldrtn	bldapp bldrtn	n/a	bldapp bldrtn	bldapp.bat bldrtn.bat

建置檔是用在建置應用程式及常式的文件中，因為它們會非常清楚地示範 DB2 建議支援編譯器使用的編譯及鏈結選項。通常有許多其它編譯和鏈結選項，使用者可自由地試驗它們。請參閱提供的編譯和鏈結選項的編譯器文件。除了建置和範例程式以外，軟體開發者也可使用建置檔建置自己的程式。使用者可修改範例程式，將之作爲模版以協助他們的程式設計開發。

爲了方便起見，建置檔是設計成使用編譯器容許的檔名來建置來源檔。這方面不像 make 檔，make 檔會將程式名稱寫到檔案內。make 檔會存取建置檔，以編譯及鏈結它們所製作的程式。建置檔在 UNIX 上使用 \$1 變數，在 Windows 作業系統上使用 %1 變數，替代內部的程式名稱。這些變數名稱中的遞增號碼替代可能需要的其它引數。

建置檔能夠快速而容易地試驗，因為每一個建置檔都適合特定種類的程式建置，如獨立式應用程式、常式 (儲存程序及 UDF) 或其他特殊化程式類型，如多連線或多緒程式。只要編譯器支援某類型建置檔要建置的特定類型程式就會提供該種類建置檔。

由建置檔所產生的物件及可執行檔會在每一次程式建置時自動改寫，即使來源檔沒有修改也一樣。當使用 `make` 檔時，情況不是如此。它這表示軟體開發者可以重建現存的程式，而不用刪除之前的物件及可執行檔，或修改來源檔。

建置檔含有 `sample` 資料庫的預設設定。若使用者正在存取另一個資料庫，則只要提供另一個參數就可改寫預設值。若他們持續使用其它資料庫，他們可能需要將這個資料庫名稱 (取代 `sample`) 寫在建置檔本身之內。

內嵌式 `SQL` 程式所使用的建置檔會呼叫另一個檔案 `embprep`，該檔案中含有內嵌式 `SQL` 程式的前置編譯及連結步驟。這些步驟也許需要使用者 `ID` 及通行碼的選用參數，取決於內嵌式 `SQL` 程式的建置位置。

除了 `SQLJ` 之外，如果軟體開發者是在資料庫所在的伺服器案例中建置程式，則使用者 `ID` 及通行碼，則對兩者而言是共同的，也就不需要提供。換句話說，如果軟體開發者是在不同的案例中，如從遠端存取伺服器資料庫的從屬站機器，則必須提供這些參數。

即使存取本端資料庫，`SQLJ` 建置檔也需要 `db2prof.c` 自訂器的使用者 `ID` 和通行碼。這會遵循類型 4 驅動程式的慣例。

最後軟體開發者可依需要修改建置檔。除了在建置檔中變更資料庫名稱以外 (如上述)，軟體開發者可輕易將其它參數寫在檔案內，變更編譯和鏈結選項，或變更預設 `DB2` 案例路徑。建置檔的簡單、直接及特定的本質使您能夠輕易地依需要修改它們。

相關概念:

- 第 93 頁的『`make` 檔』
- 第 96 頁的『錯誤檢查公用程式』
- 第 53 頁的『範例檔案』

相關參照:

- 第 9 頁的『用於開發及執行應用程式的 `AIX` 支援軟體』
- 第 10 頁的『用於開發及執行應用程式的 `HP-UX` 支援軟體』
- 第 11 頁的『用於開發及執行應用程式的 `Linux` 支援軟體』
- 第 13 頁的『用於開發及執行應用程式的 `Solaris` 支援軟體』
- 第 14 頁的『用於開發及執行應用程式的 `Windows` 支援軟體』

make 檔

支援的編譯器的每一個範例目錄，包含一個用來建置目錄內大部份提供的範例程式的 `make` 檔。`make` 檔會呼叫建置檔來編譯及鏈結每一個程式。`make` 檔的語法和其命令的輸出，在某些重要方面與建置檔不同。不過，藉由使用 `make` 檔作為建置檔的「前端」，使用者就能夠利用 `make` 檔的簡單但強大的命令：

make <program_name>

編譯和鏈結指定的程式。

make all

編譯和鏈結列示在 `make` 檔的所有程式。

make clean

刪除列示在 `make` 檔的全部程式的全部中間檔（如目的檔）。

make cleanall

刪除列示在 `make` 檔的全部程式的全部中間檔和可執行檔。

Java™ 通常不會使用 `make` 檔，而且 `make` 可執行檔不會隨著 Java Development Kit (JDK) 一起出貨。不過，DB2® 會提供 `make` 檔作為 Java 範例的選項，以防使用者想要 `make` 命令的便利性。若要使用 Java `make` 檔，您必須具有可用的 `make` 可執行檔，通常它會與另一個語言編譯器一起使用。

底下是 DB2 依平台提供給主要程式設計語言/API 的 `make` 檔，以及放置它們的範例目錄。這些是線上文件中的熱鏈結，而且它們所建置的範例程式會在它們之內鏈結。這些檔案也可以在範例目錄中加以存取。

表 26. 依平台的範例 `make` 檔

平台 → 語言	AIX	HP-UX	Linux	Solaris	Windows
C samples/c	make 檔	make 檔	make 檔	make 檔	make 檔
C++ samples/cpp	make 檔	make 檔	make 檔	make 檔	make 檔
CLI samples/cli	make 檔	make 檔	make 檔	make 檔	make 檔
JDBC samples/java/jdbc	make 檔	make 檔	make 檔	make 檔	make 檔
SQLJ samples/java/sqlj	make 檔	make 檔	make 檔	make 檔	make 檔
IBM COBOL samples/cobol	make 檔	n/a	n/a	n/a	make 檔
Micro Focus COBOL samples/cobol_mf	make 檔	make 檔	n/a	make 檔	make 檔
SQL 程序 samples/sqlproc	make 檔	make 檔	make 檔	make 檔	make 檔

不像建置檔，`make` 檔不會改寫它列示之程式的現存中間檔和可執行檔。使用 `make all` 命令可更快建立部份這些檔案的可執行檔（若其它檔案已有可執行檔），因為 `make all` 會忽略這些檔案。但它也需要 `make clean` 及 `make cleanall` 命令，以便在不需要現存目的檔和可執行檔時除去它們。

`make` 檔可用於程式開發。因為它們需要將程式名稱寫在檔案本身，所以您可能認為 `make` 檔比建置檔不容易使用，但如果您要的是 `make` 命令的能力與便利的話，那麼不妨考慮使用 `make` 檔。

`make` 檔會將它們所呼叫的程式組織在變數所代表的數個從屬站及伺服器程式種類之下（請參閱 `make` 檔以取得詳細資訊）。如果您要新增程式到 `make` 檔，請確定您已新增它，可供正確變數存取它。例如，可以在任何從屬站（位於伺服器本端或遠端）上執行的程式置於 `client_run` 變數之下。

您也需要在 `cleanall` 變數下指定程式名稱，以確定所產生的可執行檔可被 `make cleanall` 命令刪除。此外，如果它是內嵌式 SQL 程式，請在 `clean` 變數下指定因前置編譯而建立的非內嵌式 SQL 檔，以便 `make clean` 命令（以及呼叫它的 `make cleanall` 命令）將刪除非內嵌式 SQL 檔。

此外，您需要以正確的語法指定新檔案，以呼叫適當的建置檔來編譯及鏈結程式。

若要正確地估計新檔案需要新增至哪一個範例 `make` 檔，底下是位於 AIX® C `make` 檔的內嵌式 SQL 程式 `dbauth` 所在之處。

```
*****
#
#           2f - make client_run
*****

client_run : \
    cli_info clisnap \
    dbauth dbconn dbcreate dbinfo dbmcon \
. . .
*****
#
#           2g - make clean
*****

clean :
    $(ERASE) *.o
    $(ERASE) *.DEL *.TXT *.MSG
    $(ERASE) dbauth.c dbcfg.c dbconn.c dbmcon.c dbmcon1.c dbmcon2.c
. . .
*****
#
#           2h - make cleanall
*****

cleanall : \
    clean
    $(ERASE) *.bnd
    $(ERASE) cli_info clisnap
    $(ERASE) dbauth dbcfg dbconn dbcreate dbinfo dbmcon dbmcon1 dbmcon2
. . .
*****
#
#           3b - regular samples, embedded SQL
*****

dbauth :
    $(BLDAPP) dbauth $(ALIAS) $(UID) $(PWD)
```

上面最後一行的程式名稱後的三個變數 `ALIAS`、`UID` 及 `PWD` 分別代表資料庫的別名、使用者 ID 及通行碼。這些變數將傳遞至建置檔 (在這個情況中，`BLDAPP` 變數所代表的 `bldap` 建置檔)。如果程式使用內嵌式 SQL，則 `ALIAS`、`UID` 及 `PWD` 將輪流傳遞至建置檔所呼叫的 `embprep` 前置編譯及連結 Script。在使用 `make` 檔前，您可能需要變更這些變數值。依預設，`ALIAS` 將設為 `sample` 資料庫，而 `UID` 及 `PWD` 沒有設定值。

UID 及 PWD 是選用參數，因此若使用者已經在與伺服器資料庫相同的案例中工作的話，便不需要設定這些選用參數。然而，如果不是這種情況，例如，如果使用者自從屬站機器遠端連接伺服器，則使用者將需要修改 `make` 檔以提供正確值給 UID 及 PWD 變數，才能存取資料庫。

對於多連線程式，C、CLI 及 C++ `make` 檔也具有第二個資料庫別名 `ALIAS2`，預設是設為 `sample2` 資料庫。相對應的使用者 ID 和通行碼 `UID2` 及 `PWD2` 沒有設定值。如同 UID 及 PWD 變數一般，若第二個資料庫是在本端存取，則它們將不需要值。

`make` 檔同時也定義 `ERASE` 變數，以便在呼叫 `make clean` 及 `make cleanall` 命令時刪除檔案。在 UNIX，這會設為 `rm -f`；在 Windows®，它會設為 `del`。

相關概念:

- 第 90 頁的『建置檔』
- 第 96 頁的『錯誤檢查公用程式』
- 第 53 頁的『範例檔案』

相關參照:

- 第 9 頁的『用於開發及執行應用程式的 AIX 支援軟體』
- 第 10 頁的『用於開發及執行應用程式的 HP-UX 支援軟體』
- 第 11 頁的『用於開發及執行應用程式的 Linux 支援軟體』
- 第 13 頁的『用於開發及執行應用程式的 Solaris 支援軟體』
- 第 14 頁的『用於開發及執行應用程式的 Windows 支援軟體』

錯誤檢查公用程式

DB2® AD Client 提供數個公用程式檔。這些檔案有一些函數，可以錯誤檢查及列印錯誤資訊。在範例目錄中提供每一種語言版本的公用程式檔。在與某一應用程式一起使用時，錯誤檢查公用程式檔可提供有用的錯誤資訊，以便更易於除錯 DB2 程式。大部份的錯誤檢查公用程式會使用 DB2 API `GET SQLSTATE MESSAGE (sqllogstt)` 及 `GETERROR MESSAGE (sqlaintp)` 以取得適當的 `SQLSTATE` 及 `SQLCA` 資訊，這些資訊與程式執行時所發生的問題相關。DB2 CLI 公用程式檔 `utilcli.c` 不使用這些 DB2 API；相反地，它使用同等的 DB2 CLI 陳述式。所有錯誤檢查公用程式印出的說明性錯誤訊息，讓軟體開發者能夠迅速瞭解問題。

有些 DB2 程式，如常式 (儲存程序和使用者定義的函數)，不必使用這些公用程式。Java™ 也不需要這些公用程式，因為在發生異常狀況時，`SQLException` 物件會被擲出。

以下是 DB2 支援的不同程式設計語言的編譯器，所使用的錯誤檢查公用程式檔：

checkerr.cb1

對於 IBM COBOL 程式

checkerr.cb1

對於 Micro Focus COBOL 程式

utilcli.c

供 CLI 程式使用

utilcli.h

utilcli.c 的標頭檔

utilapi.c

供 C 非內嵌式 SQL 程式使用

utilapi.h

utilapi.c 的標頭檔

utilemb.sqc

供 C 內嵌式 SQL 程式使用

utilemb.h

utilemb.sqc 的標頭檔

utilapi.C

供 C++ 非內嵌式 SQL 程式使用。

utilapi.h

utilapi.C 的標頭檔

utilemb.sqC

供 C++ 內嵌式 SQL 程式使用

utilemb.h

utilemb.sqC 的標頭檔

要使用公用程式函數之前，必須先編譯公用程式檔，然後編譯在建立目標程式的可執行檔時鏈結的目的檔。在 `samples` 目錄中的 `makefile` 及建置檔，可供需要錯誤檢查公用程式的程式執行上述作業。

下列範例示範如何在 DB2 程式中使用錯誤檢查公用程式。 `utilemb.h` 標頭檔會定義 `EMB_SQL_CHECK` 巨集，該巨集可替代函數 `SqlInfoPrint()` 及 `TransRollback()`：

```
#define EMB_SQL_CHECK( MSG_STR )          \
    if( SqlInfoPrint( MSG_STR, &sqlca, __LINE__, __FILE__) != 0 ) \
        TransRollback( );
```

SqlInfoPrint() 會檢查 SQLCODE 旗號。它會將此旗號指示的特定錯誤之任何相關可用資訊列印出來。它也會指向原始程式中發生錯誤的位置。 TransRollback() 可讓公用程式檔安全地在發生錯誤的交易後執行失敗回復。它需要內嵌式 SQL 陳述式，連接資料庫並執行失敗後回復。下列範例說明 C 程式 dbuse 如何使用巨集呼叫公用程式函數，並提供 SqlInfoPrint() 函數的 MSG_STR 參數值 "Delete with host variables -- Execute"：

```
EXEC SQL DELETE FROM org
      WHERE deptnumb = :hostVar1 AND
            division = :hostVar2;
EMB_SQL_CHECK("Delete with host variables -- Execute");
```

EMB_SQL_CHECK 巨集會確保如果 DELETE 陳述式失效，則交易仍可安全地失敗後回復，並列印適當的錯誤訊息。

鼓勵軟體開發者在建立自己的 DB2 程式時使用和展開這些錯誤檢查公用程式。

相關概念:

- 第 90 頁的『建置檔』
- 第 93 頁的『make 檔』
- 第 53 頁的『範例檔案』

第 2 篇 建置及執行沒有平台限制的應用程式

第 4 章 Java

Java 範例程式	101	UNIX 的 SQLJ 應用程式選項	115
Java Applet 注意事項	103	SQLJ 應用程式及 Applet 的 Windows 批 次檔	116
JDBC	105	Windows 的 SQLJ 應用程式選項	118
建置 JDBC Applet	105	建置 SQLJ 常式	119
建置 JDBC 應用程式	106	SQLJ 常式的 UNIX 建置 Script	120
建置 JDBC 常式	107	UNIX 的 SQLJ 儲存程序選項	122
SQLJ	110	SQLJ 常式的 Windows 批次檔	123
建置 SQLJ 程式	110	Windows 的 SQLJ 儲存程序選項	124
建置 SQLJ Applet	111		
建置 SQLJ 應用程式	113		
SQLJ 應用程式及 Applet 的 UNIX 建置 Script	114		

本章提供建置 Java applet 及應用程式的詳細資訊。關於最新 DB2 Java 應用程式開發更新組件，請探訪下列網頁：

<http://www.ibm.com/software/data/db2/java>

Java 範例程式

DB2[®] 提供範例程式來示範建立和執行只用動態 SQL 的 JDBC 程式，以及使用靜態 SQL 的 SQLJ 程式。在 java 範例目錄下有 JDBC 及 SQLJ 範例的個別目錄。底下是 UNIX[®] 及 Windows[®] 作業系統上的 Java 範例目錄結構：

- 在 UNIX：

sqllib/samples/java

含有所有子目錄中 Java 範例程式的 README 檔。

sqllib/samples/java/jdbc

含有 JDBC 範例程式檔。

sqllib/samples/java/sqlj

含有 SQLJ 範例程式。

sqllib/samples/java/WebSphere

含有 WebSphere 範例程式。

sqllib/samples/java/plugin

含有「DB2 控制中心」的外掛程式範例檔。

sqllib/samples/java/plugin/doc

含有外掛程式介面的 javadoc 檔。

- 在 Windows :

sqllib\samples\java

含有所有子目錄中 Java 範例程式的 README 檔。

sqllib\samples\java\jdbc

含有 JDBC 範例程式。

sqllib\samples\java\sqlj

含有 SQLJ 範例程式。

sqllib\samples\java\Websphere

含有 Websphere 範例程式。

sqllib\samples\java\plugin

含有「DB2 控制中心」的外掛程式範例檔。

sqllib\samples\java\plugin\doc

含有外掛程式介面的 javadoc 檔。

SQLJ 範例目錄含有為 Java 程式建置內嵌式 SQL 的建置檔 (UNIX 上的 Script, Windows 上的批次檔)。JDBC 目錄不含有建置檔, 因為在命令行上使用 javac 來建置 JDBC 程式是如此的簡單, 以致不需要建置檔。

JDBC 及 SQLJ 範例目錄全都含有選用的 make 檔。make 檔不會與廣泛地與 Java 一起使用, 而且 Java Development Kits (JDKs) 不會隨著 make 可執行檔一起提供。DB2 提供 Java 範例 make 檔, 以防使用者想要它們提供的便利性。每一個 Java makefile 都會在 JDBC 或 SQLJ 範例目錄中建置所有提供的範例程式。您可以使用與另一個語言編譯器一起使用的 make 程式, 如 gnumake。

提供兩個 SQLJ 建置檔: UNIX 上的 bldsqlj 或 Windows 上的 bldsqlj.bat, 用以建置 SQLJ applet 及應用程式, 以及 UNIX 上的 bldsqljs 或 Windows 上的 bldsqljs.bat, 用以建置 SQLJ 常式 (儲存程序及使用者定義的函數)。

相關作業:

- 第 20 頁的『設置 Java 環境』
- 第 105 頁的『建置 JDBC Applet』
- 第 106 頁的『建置 JDBC 應用程式』
- 第 107 頁的『建置 JDBC 常式』
- 第 111 頁的『建置 SQLJ Applet』
- 第 113 頁的『建置 SQLJ 應用程式』
- 第 119 頁的『建置 SQLJ 常式』
- 第 110 頁的『建置 SQLJ 程式』

相關參照:

- 第 70 頁的『JDBC 範例』
- 第 72 頁的『SQLJ 範例』
- 第 74 頁的『Java WebSphere 範例』
- 第 75 頁的『Java 外掛程式範例』

Java Applet 注意事項

您可以使用 Java™ applet 來存取 DB2® 資料庫。使用它們時，請記住下列幾點：

1. 如果您要使用現在反對使用的類型 3 驅動程式 (又稱為「網路」驅動程式)，則 Java Applet 使用的 db2java.zip 檔案與 JDBC Applet 伺服器處於相同的 FixPak 層次很重要。在正常情況下，db2java.zip 從 JDBC Applet 伺服器正在執行的 Web 伺服器載入。這會確保相配。不過，如果您設定的 Java Applet 是從不同的位置載入 db2java.zip，則會發生不符情況。在兩個檔案之間使 FixPak 層次相符在連線時間嚴格執行。如果偵測出不符，則連線遭拒，並且從屬站接收到下列其中一項異常狀況：

- 如果 db2java.zip 是 DB2 版本 7 FixPak 2 或更新版本：

```
COM.ibm.db2.jdbc.DB2Exception: [IBM] [JDBC 驅動程式]
CLI0621E 未支援的 JDBC 伺服器設定。
```

- 如果 db2java.zip 在 FixPak 2 之前：

```
COM.ibm.db2.jdbc.DB2Exception: [IBM] [JDBC 驅動程式]
CLI0601E 已結束無效陳述式 handle 或陳述式。
SQLSTATE=S1000
```

如果發生不符情況，則 JDBC Applet 伺服器在 jdbcerr.log 檔案中記載下列其中一項訊息：

- 如果 JDBC Applet 伺服器是 DB2 版本 7 FixPak 2 或更新版本：

```
jdbcFSQLConnect: JDBC Applet Server 及從屬站 (db2java.zip)
版本不相配。無法繼續連線。 , einfo= -111
```

- 如果 JDBC Applet 伺服器在 FixPak 2 之前：

```
jdbcServiceConnection(): 接收到無效要求。 , einfo= 0
```

若要測試您的 JDBC 環境，您可以使用 sqllib\samples\java (Windows) 或 sqllib\samples\java (UNIX) 中的範例檔案 db2JDBCVersion.java。db2JDBCVersion 程式檢查目前使用之 DB2 JDBC 驅動程式的版本，以及 JDBC 環境是否已為其正確設定。

強烈建議使用者將他們的 applet 移轉到類型 4 驅動程式。

2. 就由數個 Java 類別組成的大型 JDBC 或 SQLJ Applet 而言，您可選擇將它的類別備份至單一 JAR 檔。就 SQLJ Applet 而言，您也必須將序列設定檔及其類別放置一處。若選擇這樣做，請將 JAR 檔新增至 "Applet" 標籤中的 archive 參數。關於詳細資訊，請參閱 JDK 版本 1.3 文件。

就 SQLJ Applet 而言，有些瀏覽器尚未支援從一個與 Applet 相關的資源檔載入序列化物件。例如，在那些瀏覽器載入提供的範例 Applet Applet 時會出現下列錯誤訊息：

```
java.lang.ClassNotFoundException: Applet_SJProfile0
```

其解決之道，是使用一個公用程式將序列化設定檔轉換成以 Java 類別格式儲存的設定檔。此公用程式是一個稱為

sqlj.runtime.profile.util.SerProfileToClass 的 Java 類別。它將序列化設定檔資源檔視為輸入，並產生一個含有設定檔的 Java 類別作為輸出。您可使用下列其中一個命令來轉換設定檔：

```
profconv Applet_SJProfile0.ser
```

或

```
java sqlj.runtime.profile.util.SerProfileToClass Applet_SJProfile0.ser
```

結果會建立類別 Applet_SJProfile0.class。使用 .class 格式的設定檔取代 Applet 使用的 .ser 格式的所有設定檔，問題就會解決。

3. 您可能需要將檔案 db2java.zip（就 SQLJ Applet 而言，也包括檔案 runtime.zip）放在一個由數個可從網站下載的 Applet 共用的目錄。這些檔案位於 Windows[®] 作業系統上的 sqllib\java 目錄，以及位於 UNIX 上的 sqllib/java 目錄。您可能需要將 codebase 參數新增至 HTML 檔中的 Applet 標籤以識別此目錄。關於詳細資訊，請參閱 JDK 版本 1.3 文件。
4. 自從 DB2 版本 5.2 之後，信號處理常式已新增至 JDBC Applet 伺服器（監聽者）db2jd，使它更加健全。因此使用者無法使用 CTRL-C 命令來結束 db2jd。因此，終止監聽者的唯一方法就是使用 kill -9 (Unix) 或「工作管理程式」(Windows) 來結束此程序。
5. 關於在 Web 伺服器上執行 DB2 Java Applet 的資訊（尤其是在 Domino[™] Go Webserver），請探訪：

```
http://www.ibm.com/software/data/db2/db2lotus/gojava.htm
```

相關作業：

- 第 20 頁的『設置 Java 環境』
- 第 105 頁的『建置 JDBC Applet』

JDBC

建置 JDBC Applet

Applet 示範動態 SQL Java applet 如何存取 DB2 資料庫。

程序:

您可以使用現在反對使用的類型 3 驅動程式 (又稱為「網路」驅動程式) 或類型 4 驅動程式。與這兩個驅動程式連接的區段將呈現於下。強烈建議您將您的 applet 移轉到類型 4 驅動程式。

若要在命令行中輸入命令，來建置並執行 JDBC applet (Applet)，請確定 Web 伺服器已安裝並正在您的 DB2 機器 (伺服器或從屬站) 上執行，或是在從屬站機器的工作目錄中輸入下列命令，使用附於 Java Development Kit 的 Applet 檢視器來進行作業：

```
Appletviewer Applet.html
```

與類型 3 (「網路」) 驅動程式連接

若要與類型 3 驅動程式連接，首先根據檔案中的指示來修改 Applet.html 檔。然後，在您的 Applet.html 中指定的 TCP/IP 埠上啟動 JDBC Applet 伺服器。例如，若在 Applet.html 中您指定 param name=port value='6789'，那麼您應該會輸入下列命令：

```
db2jstrt 6789
```

請確定連線字串中的 JDBC 埠號是建議的預設值。如果您能確定號碼與另一埠號不衝突，則可變更此項。不要使用資料庫埠號 "50000"。

與類型 4 驅動程式連接

若要與類型 4 驅動程式連接，請根據檔案中的指示來修改 Applet.html 檔，不過，指定的 TCP/IP 埠號應該是資料庫埠號 "50000"。

建置 Applet

1. 使用這個命令編譯 Applet.java 以產生檔案 Applet.class：

```
javac Applet.java
```

2. 請確定您的 Web 瀏覽器可以存取您的工作目錄。如果無法存取工作目錄，請將 Applet.class 和 Applet.html 複製到可存取的目錄中。

3. 若使用類型 3 驅動程式，請將 Windows 上的 `sqllib\java\db2java.zip` 或 UNIX 上的 `sqllib/java/db2java.zip` 複製到 `Applt.class` 及 `Applt.html` 所在的同一目錄。

若使用類型 4 驅動程式，請將 Windows 上的 `sqllib\java\db2jcc.jar` 或 UNIX 上的 `sqllib/java/db2jcc.jar` 複製到 `Applt.class` 及 `Applt.html` 所在的同一目錄。

4. 在從屬站機器上，啓動 Web 瀏覽器（它必須支援 Java 1.3）並載入 `Applt.html`。

您也可使用 `Java make` 檔建置此程式。

相關概念:

- 第 103 頁的『Java Applet 注意事項』

相關作業:

- 第 106 頁的『建置 JDBC 應用程式』
- 第 107 頁的『建置 JDBC 常式』
- 第 111 頁的『建置 SQLJ Applet』

相關參照:

- 第 70 頁的『JDBC 範例』

相關範例:

- 『`Applt.java` -- A Java applet that use JDBC applet driver to access a database (JDBC)』

建置 JDBC 應用程式

`DbInfo` 示範一個存取 DB2 資料庫的動態 SQL Java 應用程式。

程序:

透過在命令行上輸入的命令來建置和執行此應用程式：

1. 使用這個命令編譯 `DbInfo.java` 以產生檔案 `DbInfo.class`：

```
javac DbInfo.java
```

2. 使用此命令對應用程式執行 `java` 直譯器：

```
java DbInfo
```

您也可使用 `Java makefile` 建置此程式。

註: 如果您要在 UNIX 上的 64 位元 DB2 案例中執行 Java 應用程式，但 JDK 是 32 位元，您必須在執行應用程式前，變更 DB2 檔案庫路徑。例如，在 AIX 上：

- 若使用 bash 或 Korn shell：

```
export LIBPATH=$HOME/sql1lib/lib32
```

- 若使用 C shell：

```
setenv LIBPATH $HOME/sql1lib/lib32
```

相關作業:

- 第 105 頁的『建置 JDBC Applet』
- 第 107 頁的『建置 JDBC 常式』
- 第 113 頁的『建置 SQLJ 應用程式』

相關參照:

- 第 70 頁的『JDBC 範例』

相關範例:

- 『DbInfo.java -- How to get/set info in a database (JDBC)』

建置 JDBC 常式

DB2 提供範例程式，示範 JDBC 常式 (儲存程序及使用者定義的函數)。這些常式位於 UNIX 上的 `samples/java/jdbc` 目錄，Windows 上的 `samples\java\jdbc` 目錄。常式是在伺服器上編譯及儲存。從屬站應用程式呼叫時，它們會存取伺服器資料庫並將資訊傳回從屬站應用程式。

程序:

下列範例告訴您如何建置由下列構成的常式：

- 儲存程序
- 沒有 SQL 陳述式的使用者定義的函數
- 有 SQL 陳述式的使用者定義的函數

儲存程序

Spserver 示範動態 SQL PARAMETER STYLE JAVA 儲存程序。

若要從命令行於伺服器上建置並執行這個程式，請：

1. 請使用下列命令編譯 Spserver.java，以產生檔案 Spserver.class：

```
javac Spserver.java
```

2. 複製 Spserver.class 到 Windows 作業系統的 sqllib\function 目錄，或到 UNIX 的 sqllib/function 目錄。
3. 接著，在伺服器上執行 spcat Script，將常式編目。輸入：

```
spcat
```

這個 Script 將連接到範例資料庫、將取消常式編目 (若之前已呼叫 Spdrop.db2 將它們編目的話)，然後呼叫 Spcreate.db2 來將它們編目，最後切斷資料庫連線。您也可以個別地執行 Spdrop.db2 及 Spcreate.db2 Script。

4. 然後，停止資料庫後再重新啟動，以辨識新的類別檔。必要時，請將類別檔的檔案模式設為「讀取」，以便隔離使用者可以讀取它。
5. 編譯並執行 Spclient 從屬站應用程式以存取儲存程序類別。

沒有 SQL 陳述式的使用者定義的函數

UDFsrv 是沒有 SQL 陳述式的使用者定義的函數。DB2 提供 JDBC 從屬站應用程式 (UDFcli) 及 SQLJ 從屬站應用程式 (UDFcli)，它們都可以存取 UDFsrv 檔案庫。

若要從命令行於伺服器上建置並執行 UDF 程式，請：

1. 使用此命令編譯 UDFsrv.java 以產生檔案 UDFsrv.class：

```
javac UDFsrv.java
```

2. 複製 UDFsrv.class 到 Windows 作業系統的 sqllib\function 目錄，或到 UNIX 的 sqllib/function 目錄。
3. 若要存取 UDFsrv 檔案庫，您可使用 JDBC 或 SQLJ 從屬站應用程式。從屬站程式的這兩個版本都含有 CREATE FUNCTION SQL 陳述式，您可以使用它，利用資料庫來登錄 UDFsrv 中所含有的 UDF。從屬站程式的這兩個版本同時也含有一旦 UDF 登錄完成後就可以使用它們的 SQL 陳述式。

有 SQL 陳述式的使用者定義的函數

UDFsqlsv 是含有 SQL 陳述式的使用者定義的函數。DB2 提供 JDBC 從屬站應用程式 (UDFsqlcl) 來存取 UDFsqlsv 檔案庫。

若要從命令行於伺服器上建置並執行 UDF 程式，請：

1. 使用下列命令編譯 UDFsqlsv.java，以產生檔案 UDFsqlsv.class：

```
javac UDFsqlsv.java
```

2. 複製 UDFsqlsv.class 到 Windows 作業系統的 sqllib\function 目錄，或到 UNIX 的 sqllib/function 目錄。

3. 若要存取 UDFsqlsv 檔案庫，請使用從屬站程式 UDFsqlcl，它含有 CREATE FUNCTION SQL 陳述式，您可以使用它，利用資料庫來登錄 UDFsqlsv 中所含有的 UDF。從屬站程式同時也含有一旦 UDF 登錄完成後就可以使用它們的 SQL 陳述式。

您也可以使用 Java make 檔來建置上述的程式。

相關作業:

- 第 105 頁的『建置 JDBC Applet』
- 第 106 頁的『建置 JDBC 應用程式』
- 第 119 頁的『建置 SQLJ 常式』

相關參照:

- 第 70 頁的『JDBC 範例』

相關範例:

- 『spcat -- To catalog SQLj stored procedures on UNIX』
- 『SpClient.java -- Call a variety of types of stored procedures from SpServer.java (JDBC)』
- 『SpCreate.db2 -- How to catalog the stored procedures contained in SpServer.java』
- 『SpDrop.db2 -- How to uncatalog the stored procedures contained in SpServer.java』
- 『SpServer.java -- Provide a variety of types of stored procedures to be called from (JDBC)』
- 『UDFcli.java -- Call the UDFs in UDFsrv.java (JDBC)』
- 『UDFCreate.db2 -- How to catalog the Java UDFs contained in UDFsrv.java』
- 『UDFDrop.db2 -- How to uncatalog the Java UDFs contained in UDFsrv.java』
- 『UDFsCreate.db2 -- How to catalog the UDFs contained in UDFsqlsv.java』
- 『UDFsDrop.db2 -- How to uncatalog the UDFs contained in UDFsqlsv.java』
- 『UDFsqlcl.java -- Call the UDFs in UDFsqlsv.java (JDBC)』
- 『UDFsqlsv.java -- Provide UDFs to be called by UDFsqlcl.java (JDBC)』
- 『UDFsrv.java -- Provide UDFs to be called by UDFcli.java (JDBC)』

建置 SQLJ 程式

DB2 提供建置檔，以建置 SQLJ 範例程式。對於 applet 及應用程式，您可以使用 UNIX 上的 `bldsqj` Script，或是 Windows 上的 `bldsqj.bat` 批次檔。對於常式（儲存程序及使用者定義的函數），您可以使用 UNIX 上的 `bldsqjls` Script，或是 Windows 上的 `bldsqjls.bat` 批次檔。

附於 DB2 的 SQLJ 轉換程式將編譯轉換的 `.java` 檔成爲 `.class` 檔。因此，建置檔不使用 `java` 編譯器。

註：在舊版 DB2 中，`db2prof` 命令已使用格式爲 `-url=jdbc:db2:dbname` 的 URL，其中 `dbname` 是在本端編目的資料庫別名。新的格式遵循 DB2 類型 4 JDBC 驅動程式的慣例：`-url=jdbc:db2://hostname:portnumber/dbname`，其中 `hostname` 是 DB2 伺服器的名稱、`portnumber` 是 DB2 伺服器的 TCP/IP 監聽者埠號，而 `dbname` 是已在「DB2 伺服器」上編目的資料庫別名。這表示您必須架構「DB2 伺服器」，才能進行 TCP/IP 連線。

程序：

若要使用 IBM Java Development Kit for UNIX 及 Windows 作業系統，建置及執行 SQLJ 程式，您必須使用下列適用於您作業系統的命令，關閉 JDK 的 JIT (just-in-time) 編譯器：

UNIX：

若爲 bash 或 Korn shell：

```
export JAVA_COMPILER=NONE
```

若爲 C shell：

```
setenv JAVA_COMPILER NONE
```

Windows：

```
SET JAVA_COMPILER=NONE
```

若要建置不同類型的 DB2 SQLJ 程式，請參閱下列：

- 建置 SQLJ Applet
- 建置 SQLJ 應用程式
- 建置 SQLJ 常式

相關概念：

- 第 101 頁的『Java 範例程式』

- 第 103 頁的『Java Applet 注意事項』

相關作業:

- 第 111 頁的『建置 SQLJ Applet』
- 第 113 頁的『建置 SQLJ 應用程式』
- 第 119 頁的『建置 SQLJ 常式』

相關參照:

- 第 72 頁的『SQLJ 範例』

建置 SQLJ Applet

下列步驟顯示如何建置 Applet 範例，示範一個存取 DB2 資料庫的 SQLJ applet。這些步驟會使用建置檔 bldsqlj (UNIX) 或 bldsqlj.bat (Windows)。這個建置檔含有要建置 SQLJ applet 或應用程式的命令。

建置檔最多使用 6 個參數：UNIX 上的 \$1、\$2、\$3、\$4、\$5 及 \$6，以及 Windows 上的 %1、%2、%3、%4、%5 及 %6。第一個參數指定程式名稱。第二個參數指定資料庫案例的使用者 ID，第三個參數指定通行碼。第四個參數指定伺服器名稱。第五個參數指定埠號。第六個參數則指定資料庫名稱。除了第一個參數 (程式名稱) 外，都可以使用預設值。請參閱建置檔，以取得使用預設參數值的詳細資訊。

程序:

您可以使用現在反對使用的類型 3 驅動程式 (又稱為「網路」驅動程式) 或類型 4 驅動程式。與這兩個驅動程式連接的區段將呈現於下。強烈建議您將您的 applet 移轉到類型 4 驅動程式。

若要執行這個 applet，請確定 Web 伺服器已安裝並正在您的 DB2 機器 (伺服器或從屬站) 上執行，或是您可以在從屬站機器的工作目錄中輸入下列命令，使用附於 Java Development Kit 的 Applet 檢視器來進行作業：

```
Appletviewer Applet.html
```

與類型 3 (「網路」) 驅動程式連接

若要與類型 3 驅動程式連接，首先根據檔案中的指示來修改 Applet.html 檔。然後，在您在 Applet.html 中指定的 TCP/IP 埠上啟動 JDBC Applet 伺服器。例如，若在 Applet.html 中您指定 param name=port value='6789'，那麼您應該會輸入下列命令：

```
db2jstrt 6789
```

請確定連線字串中的 JDBC 埠號是建議的預設值。如果您能確定號碼與另一埠號不衝突，則可變更此項。不要使用資料庫埠號 50000。

與類型 4 驅動程式連接

若要與類型 4 驅動程式連接，請根據檔案中的指示來修改 `Applt.html` 檔，不過，指定的 TCP/IP 埠號應該是資料庫埠號 50000。

建置 Applet

1. 使用下列命令來建置 Applet：

```
blsqlj Applt <userid> <password> <server_name> <port_number> <db_name>
```

其中除了程式名稱外的所有參數都可以具有預設值，其說明在建置檔中。

2. 請確定您的 Web 瀏覽器或 applet 檢視器 (若正在使用它) 可以存取您的工作目錄。如果無法存取您的目錄，請複製下列檔案到可存取的目錄：

<code>Applt.html</code>	<code>Applt.class</code>
<code>Applt_Cursor1.class</code>	<code>Applt_Cursor2.class</code>
<code>Applt_SJProfileKeys.class</code>	<code>Applt_SJProfile0.ser</code>

3. 若使用類型 3 驅動程式，請將 Windows 上的 `sqllib\java\runtime.zip` 及 `sqllib\java\db2java.zip` 或 UNIX 上的 `sqllib/java/runtime.zip` 及 `sqllib/java/db2java.zip` 複製到 `Applt.class` 及 `Applt.html` 所在的同一目錄。

若使用類型 4 驅動程式，請將 Windows 上的 `sqllib\java\runtime.zip` 及 `sqllib\java\db2jcc.jar` 或 UNIX 上的 `sqllib/java/runtime.zip` 及 `sqllib/java/db2jcc.jar` 複製到 `Applt.class` 及 `Applt.html` 所在的同一目錄。

4. 在從屬站機器上，啟動 Web 瀏覽器 (它必須支援 JDK 1.3) 或 applet 檢視器，並載入 `Applt.html`。

您也可使用 Java make 檔建置此程式。

相關概念:

- 第 103 頁的『Java Applet 注意事項』

相關作業:

- 第 105 頁的『建置 JDBC Applet』
- 第 113 頁的『建置 SQLJ 應用程式』
- 第 119 頁的『建置 SQLJ 常式』

相關參照:

- 第 115 頁的『UNIX 的 SQLJ 應用程式選項』

- 第 118 頁的『Windows 的 SQLJ 應用程式選項』
- 第 72 頁的『SQLJ 範例』

相關範例:

- 『Applt.sqlj -- An SQLJ applet that uses a JDBC applet driver to access a database (SQLj)』
- 『blsqlj.bat -- Builds a Java embedded SQL (SQLJ) application or applet on Windows』
- 『blsqlj -- Builds Java embedded SQL (SQLJ) applications and applets on UNIX』

建置 SQLJ 應用程式

下列步驟顯示如何建置 TbMod 範例，示範一個存取 DB2 資料庫的 SQLJ 應用程式。這些步驟會使用建置檔 blsqlj (UNIX) 或 blsqlj.bat (Windows)。這個建置檔含有要建置 SQLJ applet 或應用程式的命令。

建置檔最多使用 6 個參數：UNIX 上的 \$1、\$2、\$3、\$4、\$5 及 \$6，以及 Windows 上的 %1、%2、%3、%4、%5 及 %6。第一個參數指定程式名稱。第二個參數指定資料庫案例的使用者 ID，第三個參數指定通行碼。第四個參數指定伺服器名稱。第五個參數指定埠號。第六個參數則指定資料庫名稱。除了第一個參數 (程式名稱) 外，都可以使用預設值。請參閱建置檔，以取得使用預設參數值的詳細資訊。

程序:

若要利用建置檔 blsqlj (UNIX) 或 blsqlj.bat (Windows)，建置 TbMod，請輸入這個命令：

```
blsqlj TbMod <userid> <password> <server_name> <port_number> <db_name>
```

其中除了程式名稱外的所有參數都可以具有預設值，其說明在建置檔中。

使用此命令對應用程式執行 Java 直譯器：

```
java TbMod
```

您也可使用 Java makefile 建置此程式。

註: 如果您要在 UNIX 上的 64 位元 DB2 案例中執行 Java 應用程式，但 JDK 是 32 位元，您必須在執行應用程式前，變更 DB2 檔案庫路徑。例如，在 AIX 上：

- 若使用 bash 或 Korn shell：

```
export LIBPATH=$HOME/sql1lib/lib32
```

- 若使用 C shell :

```
setenv LIBPATH $HOME/sql1lib/lib32
```

相關作業:

- 第 106 頁的『建置 JDBC 應用程式』
- 第 111 頁的『建置 SQLJ Applet』
- 第 119 頁的『建置 SQLJ 常式』

相關參照:

- 第 115 頁的『UNIX 的 SQLJ 應用程式選項』
- 第 118 頁的『Windows 的 SQLJ 應用程式選項』
- 第 72 頁的『SQLJ 範例』

相關範例:

- 『bldsqlj.bat -- Builds a Java embedded SQL (SQLJ) application or applet on Windows』
- 『bldsqlj -- Builds Java embedded SQL (SQLJ) applications and applets on UNIX』
- 『TbMod.sqlj -- How to modify table data (SQLj)』

SQLJ 應用程式及 Applet 的 UNIX 建置 Script

```
#!/bin/sh
# SCRIPT: bldsqlj
# Builds UNIX Java embedded SQL (SQLJ) applications
# Usage: bldsqlj prog_name (requires hardcoding user ID and password)
#       bldsqlj prog_name userid password
#       bldsqlj prog_name userid password server_name
#       bldsqlj prog_name userid password server_name port_number
#       bldsqlj prog_name userid password server_name port_number db_name
#
# Defaults:
#       userid      = $USER variable
#       password    = $PSWD variable
#       server_name = local hostname
#       port_number = 50000
#       db_name     = sample

# To add defaults for user ID (USER) and password (PSWD)
# Uncomment the following and add the values in the quotes
# USER=""
# PSWD=""

# Translate and compile the SQLJ source file
# and bind the package to the database.
if ( ( ( $# == 1 ) ) && [ $USER != "" && $PSWD != "" ] ) ||
    ( ( $# >= 3 && $# <= 6 ) )
then
```

```

sqlj $1.sqlj

if (($# == 1))
then
    db2profrc -url=jdbc:db2://$(hostname):50000/sample \
        -user=$USER -password=$PSWD \
        -preoptions="package using $1" $1_SJProfile0
elif (($# == 3))
then
    db2profrc -url=jdbc:db2://$(hostname):50000/sample -user=$2 \
        -password=$3 -preoptions="package using $1" $1_SJProfile0
elif (($# == 4))
then
    db2profrc -url=jdbc:db2://$4:50000/sample -user=$2 -password=$3 \
        -preoptions="package using $1" $1_SJProfile0
elif (($# == 5))
then
    db2profrc -url=jdbc:db2://$4:$5/sample -user=$2 -password=$3 \
        -preoptions="package using $1" $1_SJProfile0
else
    db2profrc -url=jdbc:db2://$4:$5/$6 -user=$2 -password=$3 \
        -preoptions="package using $1" $1_SJProfile0
fi
else
echo 'Usage: bldsqlj prog_name (requires hardcoding user ID and password)'
echo '    bldsqlj prog_name userid password'
echo '    bldsqlj prog_name userid password server_name'
echo '    bldsqlj prog_name userid password server_name port_number'
echo '    bldsqlj prog_name userid password server_name port_number
    db_name'
echo ''
echo '    Defaults:'
echo '        userid      = '$USER
echo '        password    = '$PSWD
echo '        server_name = '$(hostname)
echo '        port_number = 50000'
echo '        db_name     = sample'
fi

```

UNIX 的 SQLJ 應用程式選項

下列表格含有在 UNIX 上的 bldsqlj 建置 Script 中使用的 SQLJ 轉換程式及前置編譯選項。這些是 DB2 建議的選項，您可以使用它們，在 UNIX 平台上建置 SQLJ 應用程式及 applet。

bldsqlj 的轉換程式及前置編譯選項

sqlj	SQLJ 轉換程式（也編譯此程式）。
\$1.sqlj	SQLJ 來源檔。
db2profc	DB2 for Java 設定檔自訂程式。
-url	指定一個 JDBC URL 建立資料庫連線，如 jdbc:db2://servername:50000/sample。
-user	指定使用者 ID。
-password	指定通行碼。
-preoptions	使用字串 package using \$1 指定資料庫的資料包名稱，\$1 是 SQLJ 來源檔名稱。
\$1_SJProfile0	指定程式的序列化設定檔。

相關作業:

- 第 111 頁的『建置 SQLJ Applet』
- 第 113 頁的『建置 SQLJ 應用程式』

相關參照:

- 第 122 頁的『UNIX 的 SQLJ 儲存程序選項』

相關範例:

- 『bldsqlj -- Builds Java embedded SQL (SQLJ) applications and applets on UNIX』

SQLJ 應用程式及 Applet 的 Windows 批次檔

```
@echo off
rem BATCH FILE: bldsqlj.bat
rem Builds Windows Java embedded SQL (SQLJ) applications and applets
rem To add defaults for user ID (USER) and password (PSWD)
rem Uncomment the following and add the appropriate values
rem set USER=
rem set PSWD=

goto start
:usage
echo Usage: bldsqlj prog_name (requires hardcoding user ID and password)
echo      bldsqlj prog_name userid password
```



```

echo      blsqlj prog_name userid password server_name
echo      blsqlj prog_name userid password server_name port_number
echo      blsqlj prog_name userid password server_name port_number db_name
echo.
echo      Defaults:
echo      userid      = %USER%
echo      password    = %PSWD%
echo      server_name = %COMPUTERNAME%
echo      port_number = 50000
echo      db_name     = sample
goto exit

:start
rem Translate and compile the SQLJ source file
rem and bind the package to the database.
if "%1" == "" goto usage
if "%2" == "" goto case1
if "%3" == "" goto usage
if "%4" == "" goto case3
if "%5" == "" goto case4
if "%6" == "" goto case5
if "%7" == "" goto case6
goto usage

:case1
if "%USER%" == "" goto usage
if "%PSWD%" == "" goto usage
if "%COMPUTERNAME%" == "" goto nohostname
sqlj %1.sqlj
db2profcc -url=jdbc:db2://%COMPUTERNAME%:50000/sample -user=%USER%
        -password=%PSWD% -preoptions="package using %1" %1_SJProfile0
goto continue

:case3
if "%COMPUTERNAME%" == "" goto nohostname
sqlj %1.sqlj
db2profcc -url=jdbc:db2://%COMPUTERNAME%:50000/sample -user=%2
        -password=%3 -preoptions="package using %1" %1_SJProfile0
goto continue

:case4
sqlj %1.sqlj
db2profcc -url=jdbc:db2://%4:50000/sample -user=%2 -password=%3
        -preoptions="package using %1" %1_SJProfile0
goto continue

:case5
sqlj %1.sqlj
db2profcc -url=jdbc:db2://%4:%5/sample -user=%2 -password=%3
        -preoptions="package using %1" %1_SJProfile0
goto continue

:case6
sqlj %1.sqlj
db2profcc -url=jdbc:db2://%4:%5/%6 -user=%2 -password=%3

```

```

        -preoptions="package using %1" %1_SJProfile0
        goto continue

:continue
rem Put any post building steps here
goto exit

:nohostname
echo Local server name (hostname) could not be determined.
echo.
goto usage

:exit
@echo on

```

Windows 的 SQLJ 應用程式選項

下列表格含有用在 Windows 作業系統上的 `bldsqlj.bat` 批次檔中的 SQLJ 轉換程式及前置編譯選項。這些是 DB2 建議的選項，您可以使用它們，建置 SQLJ 常式 (儲存程序及使用者定義的函數)。

bldsqlj.bat 的轉換程式及前置編譯選項	
sqlj	SQLJ 轉換程式 (也編譯此程式)。
%1.sqlj	SQLJ 來源檔。
db2profrc	DB2 for Java 設定檔自訂程式。
-url	指定一個 JDBC URL 建立資料庫連線，如 <code>jdbc:db2://servername:50000/sample</code> 。
-user	指定使用者 ID。
-password	指定通行碼。
-preoptions	使用字串 "package using %1" 指定資料庫的資料包名稱，%1 是 SQLJ 來源檔名稱。
%1_SJProfile0	指定程式的序列化設定檔。

相關作業:

- 第 111 頁的『建置 SQLJ Applet』
- 第 113 頁的『建置 SQLJ 應用程式』

相關參照:

- 第 124 頁的『Windows 的 SQLJ 儲存程序選項』

相關範例:

- 『bldsqlj.bat -- Builds a Java embedded SQL (SQLJ) application or applet on Windows』

建置 SQLJ 常式

DB2 提供範例程式，示範 SQLJ 常式 (儲存程序及使用者定義的函數)。這些常式位於 UNIX 上的 `samples/java/sqlj` 目錄，Windows 上的 `samples\java\sqlj` 目錄。常式是在伺服器上編譯及儲存。從屬站應用程式呼叫時，它們會存取伺服器資料庫並將資訊傳回從屬站應用程式。

在同一目錄中，DB2 也提供含有要建置常式之命令的建置檔 `bldsqljs` (UNIX) 或 `bldsqljs.bat` (Windows)。

建置檔最多使用 6 個參數：UNIX 上的 `$1`、`$2`、`$3`、`$4`、`$5` 及 `$6`，以及 Windows 上的 `%1`、`%2`、`%3`、`%4`、`%5` 及 `%6`。第一個參數指定程式名稱。第二個參數指定資料庫案例的使用者 ID，第三個參數指定通行碼。第四個參數指定伺服器名稱。第五個參數指定埠號。第六個參數則指定資料庫名稱。除了第一個參數 (程式名稱) 外，都可以使用預設值。請參閱建置檔，以取得使用預設參數值的詳細資訊。

程序:

下列範例告訴您如何利用儲存程序來建置類別檔。

`Spserver` 示範 `PARAMETER STYLE JAVA` 儲存程序，使用 `JDBC` 應用程式驅動程式存取 `DB2` 資料庫。

若要利用建置檔 `bldsqljs` (UNIX) 或 `bldsqljs.bat` (Windows)，建置此儲存程序類別：

1. 請輸入下列命令：

```
bldsqljs Spserver <userid> <password> <server_name> \  
<port_number> <db_name>
```

其中除了程式名稱外的所有參數都可以具有預設值，其說明在建置檔中。

2. 接著，在伺服器上執行 `spcat` Script，將常式編目。輸入：

```
spcat
```

這個 Script 將連接到範例資料庫、將取消常式編目 (若之前已呼叫 Spdrop.db2 將它們編目的話)，然後呼叫 Spcreate.db2 來將它們編目，最後切斷資料庫連線。您也可以個別地執行 Spdrop.db2 及 Spcreate.db2 Script。

3. 然後，停止資料庫後再重新啟動，以辨識新的類別檔。必要時，請將類別檔的檔案模式設為「讀取」，以便隔離使用者可以讀取它。
4. 建置及執行 Spclient 從屬站應用程式，以呼叫儲存程序。您可以利用應用程式建置檔 bldsqlj (UNIX) 或 bldsqlj.bat (Windows)，建置 Spclient。

您也可以使用 Java make 檔來建置上述的程式。

相關作業:

- 第 107 頁的『建置 JDBC 常式』
- 第 111 頁的『建置 SQLJ Applet』
- 第 113 頁的『建置 SQLJ 應用程式』

相關參照:

- 第 122 頁的『UNIX 的 SQLJ 儲存程序選項』
- 第 124 頁的『Windows 的 SQLJ 儲存程序選項』
- 第 72 頁的『SQLJ 範例』

相關範例:

- 『bldsqljs.bat -- Builds a Java embedded SQL (SQLJ) stored procedure on Windows』
- 『bldsqljs -- Builds Java embedded SQL (SQLJ) stored procedures on UNIX』
- 『spcat -- To catalog SQLj stored procedures on UNIX』
- 『SpClient.sqlj -- Call a variety of types of stored procedures from SpServer.sqlj (SQLj)』
- 『SpCreate.db2 -- How to catalog the stored procedures contained in SpServer.sqlj』
- 『SpDrop.db2 -- How to uncatalog the stored procedures contained in SpServer.sqlj』
- 『SpIterat.sqlj -- Iterator class file for SpServer.sqlj (SQLj)』
- 『SpServer.sqlj -- Provide a variety of types of stored procedures to be called from (SQLj)』

SQLJ 常式的 UNIX 建置 Script

```
#!/bin/sh
# SCRIPT: bldsqljs
# Builds UNIX Java embedded SQL (SQLJ) routines
# Usage: bldsqljs prog_name (requires hardcoding user ID and password)
```

```

#      blsqljs prog_name userid password
#      blsqljs prog_name userid password server_name
#      blsqljs prog_name userid password server_name port_number
#      blsqljs prog_name userid password server_name port_number db_name
#
#      Defaults:
#      userid      = $USER variable
#      password    = $PSWD variable
#      server_name = local hostname
#      port_number = 50000
#      db_name     = sample

# To add defaults for user ID (USER) and password (PSWD)
# Uncomment the following and add the values in the quotes
# USER=""
# PSWD=""

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Translate and compile the SQLJ source file
# and bind the package to the database.
if ( (( $# == 1 )) && [ $USER != "" && $PSWD != "" ] ) ||
  (( $# >= 3 && $# <= 6 ))
then
  sqlj $1.sqlj

  if (($# == 1))
  then
    db2profcc -url=jdbc:db2://$(hostname):50000/sample \
      -user=$USER -password=$PSWD \
      -preoptions="package using $1" $1_SJProfile0
  elif (($# == 3))
  then
    db2profcc -url=jdbc:db2://$(hostname):50000/sample -user=$2 \
      -password=$3 -preoptions="package using $1" $1_SJProfile0
  elif (($# == 4))
  then
    db2profcc -url=jdbc:db2://$4:50000/sample -user=$2 -password=$3 \
      -preoptions="package using $1" $1_SJProfile0
  elif (($# == 5))
  then
    db2profcc -url=jdbc:db2://$4:$5/sample -user=$2 -password=$3 \
      -preoptions="package using $1" $1_SJProfile0
  else
    db2profcc -url=jdbc:db2://$4:$5/$6 -user=$2 -password=$3 \
      -preoptions="package using $1" $1_SJProfile0
  fi

# Copy the *.class and *.ser files to the 'function' directory.
rm -f $DB2PATH/function/$1*.class
rm -f $DB2PATH/function/$1*.ser
cp $1*.class $DB2PATH/function

```

```

cp $1*.ser $DB2PATH/function
else
echo 'Usage: blsqljs prog_name (requires hardcoding user ID and password)'
echo '      blsqljs prog_name userid password'
echo '      blsqljs prog_name userid password server_name'
echo '      blsqljs prog_name userid password server_name port_number'
echo '      blsqljs prog_name userid password server_name port_number
db_name'
echo ''
echo '      Defaults:'
echo '      userid      = '$USER
echo '      password    = '$PSWD
echo '      server_name = '$(hostname)
echo '      port_number = 50000'
echo '      db_name     = sample'
fi

```

UNIX 的 SQLJ 儲存程序選項

下列表格含有用在 UNIX 上的 blsqljs 建置 Script 中的 SQLJ 轉換程式及前置編譯選項。這些是 DB2 建議的選項，您可以使用它們，在 UNIX 平台上建置 SQLJ 常式（儲存程序及使用者定義的函數）。

blsqljs 的轉換程式及前置編譯選項	
sqlj	SQLJ 轉換程式（也編譯此程式）。
\$1.sqlj	SQLJ 來源檔。
db2profrc	DB2 for Java 設定檔自訂程式。
-url	指定一個 JDBC URL 建立資料庫連線，如 jdbc:db2://servername:50000/sample。
-user	指定使用者 ID。
-password	指定通行碼。
-preoptions	使用字串 "package using \$1" 指定資料庫的資料包名稱，\$1 是 SQLJ 來源檔名稱。
\$1_SJProfile0	指定程式的序列化設定檔。

相關作業:

- 第 119 頁的『建置 SQLJ 常式』

相關參照:

- 第 115 頁的『UNIX 的 SQLJ 應用程式選項』

相關範例:

- 『bldsqljs -- Builds Java embedded SQL (SQLJ) stored procedures on UNIX』

SQLJ 常式的 Windows 批次檔

```
@echo off
rem BATCH FILE: bldsqljs.bat
rem Builds Windows Java embedded SQL (SQLJ) routines

rem To add defaults for user ID (USER) and password (PSWD)
rem Uncomment the following and add the appropriate values
rem set USER=
rem set PSWD=

goto start
:usage
echo Usage: bldsqljs prog_name (requires hardcoding user ID and password)
echo         bldsqljs prog_name userid password
echo         bldsqljs prog_name userid password server_name
echo         bldsqljs prog_name userid password server_name port_number
echo         bldsqljs prog_name userid password server_name port_number db_name
echo.
echo         Defaults:
echo             userid      = %USER%
echo             password    = %PSWD%
echo             server_name = %COMPUTERNAME%
echo             port_number = 50000
echo             db_name     = sample
goto exit

:start
rem Translate and compile the SQLJ source file
rem and bind the package to the database.
if "%DB2PATH%" == "" goto nodb2cmd
if "%1" == "" goto usage
if "%2" == "" goto case1
if "%3" == "" goto usage
if "%4" == "" goto case3
if "%5" == "" goto case4
if "%6" == "" goto case5
if "%7" == "" goto case6
goto usage

:case1
if "%USER%" == "" goto usage
if "%PSWD%" == "" goto usage
if "%COMPUTERNAME%" == "" goto nohostname
sqlj %1.sqlj
db2profrc -url=jdbc:db2://%COMPUTERNAME%:50000/sample -user=%USER%
          -password=%PSWD% -preoptions="package using %1" %1_SJProfile0
```

```

goto continue

:case3
if "%COMPUTERNAME%" == "" goto nohostname
sqlj %1.sqlj
db2profcc -url=jdbc:db2://%COMPUTERNAME%:50000/sample -user=%2
          -password=%3 -preoptions="package using %1" %1_SJProfile0
goto continue

:case4
sqlj %1.sqlj
db2profcc -url=jdbc:db2://%4:50000/sample -user=%2 -password=%3
          -preoptions="package using %1" %1_SJProfile0
goto continue

:case5
sqlj %1.sqlj
db2profcc -url=jdbc:db2://%4:%5/sample -user=%2 -password=%3
          -preoptions="package using %1" %1_SJProfile0
goto continue

:case6
sqlj %1.sqlj
db2profcc -url=jdbc:db2://%4:%5/%6 -user=%2 -password=%3
          -preoptions="package using %1" %1_SJProfile0
goto continue

:continue
rem Copy the *.class and *.ser files to the 'function' directory.
copy %1*.class %DB2PATH%\function\
copy %1*.ser %DB2PATH%\function\
goto exit

:nodb2cmd
echo DB2 command line environment not initialized. Please run db2cmd
and try again.
goto exit

:nohostname
echo Local server name (hostname) could not be determined.
echo.
goto usage

:exit
@echo on

```

Windows 的 SQLJ 儲存程序選項

下列表格含有用在 Windows 作業系統上的 bldsqljs.bat 批次檔中的 SQLJ 轉換程式及前置編譯選項。這些是 DB2 建議的選項，您可以使用它們，建置 SQLJ 常式 (儲存程序及使用者定義的函數)。

bldsqljs.bat 的轉換程式及前置編譯選項

sqlj	SQLJ 轉換程式 (也編譯此程式)。
%1.sqlj	SQLJ 來源檔。
db2profrc	DB2 for Java 設定檔自訂程式。
-url	指定一個 JDBC URL 建立資料庫連線，如 jdbc:db2://servername:50000/sample。
-user	指定使用者 ID。
-password	指定通行碼。
-preoptions	使用字串 "package using %1" 指定資料庫的資料包名稱，%1 是 SQLJ 來源檔名稱。
%1_SJProfile0	指定程式的序列化設定檔。

相關作業:

- 第 119 頁的『建置 SQLJ 常式』

相關參照:

- 第 118 頁的『Windows 的 SQLJ 應用程式選項』

相關範例:

- 『bldsqljs.bat -- Builds a Java embedded SQL (SQLJ) stored procedure on Windows』

第 5 章 SQL 程序

建立 SQL 程序	127	保留 SQL 程序的中間檔	131
在 UNIX 上利用從屬站應用程式呼叫 SQL 程序	128	自訂 SQL 程序的前置編譯及連結選項	132
在 Windows 上利用從屬站應用程式呼叫 SQL 程序	129	備份及復置 SQL 程序	133
		分送編譯後的 SQL 程序	134
		重新連結 SQL 程序	135

建立 SQL 程序

「DB2 命令行處理器」Script (結尾為 .db2 副檔名者)，位於 UNIX 上的 `sqllib/samples/sqlproc` 目錄及 Windows 上的 `sqllib\samples\sqlproc` 目錄，會執行 CREATE PROCEDURE 陳述式以在伺服器上建立儲存程序。每一個 CLP Script 都有同名的相對應從屬站應用程式檔案，其副檔名表示它的語言及應用程式介面：`.sqc` (代表 C 內嵌式 SQL)、`.c` (代表 DB2 CLI) 或 `.java` (代表 JDBC)。

程序:

在執行 CREATE PROCEDURE CLP Script 之前，請使用下列命令連接 sample 資料庫：

```
db2 connect to sample user userid using password
```

其中 *userid* 及 *password* 是 sample 資料庫所在位置的案例之使用者 ID 及通行碼。

欲執行 `rsultset.db2` Script 檔中所含的 CREATE PROCEDURE 陳述式，請輸入下列命令：

```
db2 -td@ -vf rsultset.db2
```

現在，可以呼叫 SQL 程序。

相關作業:

- 第 137 頁的『利用 CALL 陳述式呼叫儲存程序』
- 第 128 頁的『在 UNIX 上利用從屬站應用程式呼叫 SQL 程序』
- 第 129 頁的『在 Windows 上利用從屬站應用程式呼叫 SQL 程序』
- 第 134 頁的『分送編譯後的 SQL 程序』
- 第 135 頁的『重新連結 SQL 程序』

相關範例:

- 『resultset.db2 -- To create the MEDIAN_RESULT_SET SQL procedure 』

在 UNIX 上利用從屬站應用程式呼叫 SQL 程序

您可以建置並執行從屬站應用程式，在 UNIX 上呼叫 SQL 程序。DB2 在 `sqllib/samples/sqlproc` 目錄中提供範例從屬站程式，以呼叫可從同時由它提供的範例命令行處理器 Script 建立的 SQL 程序。DB2 CLI、C 內嵌式 SQL 及 JDBC 有從屬站來源檔。

同時，也會提供建置 Script (JDBC 除外) 以建置這些應用程式介面的從屬站程式：

- `bldcli` 含有要建置 SQL 程序之 DB2 CLI 從屬站應用程式的命令。這個檔案同於 `sqllib/samples/cli` 目錄中的 `bldapp` Script 檔。
- `bldapp` 含有要建置 SQL 程序之 C 內嵌式 SQL 從屬站應用程式的命令。這個檔案同於 `sqllib/samples/c` 目錄中的 `bldapp` Script 檔。

程序：

根據您要使用的應用程式介面，您可以遵循這些範例，以建置並執行從屬站程式來呼叫 SQL 程序：

DB2 CLI

欲從來源檔 `resultset.c` 建置 DB2 CLI 從屬站應用程式 `resultset`，請輸入：

```
bldcli resultset
```

此命令會建立可執行檔 `resultset`。

欲呼叫儲存程序，請輸入可執行檔名稱、您要連接的資料庫名稱、及資料庫案例的使用者 ID 與通行碼，以執行範例從屬站應用程式：

```
resultset database userid password
```

C 內嵌式 SQL

若要從來源檔 `basecase.sqc` 建置內嵌式 SQL 從屬站應用程式 `basecase`，請輸入 Script 檔名稱、可執行檔名稱、您要連接的資料庫，以及資料庫案例的使用者 ID 及通行碼：

```
bldapp basecase database userid password
```

結果產生可執行檔 `basecase`。

若要呼叫儲存程序，請輸入下列命令來執行範例從屬站應用程式：

```
basecase database userid password
```

JDBC 若要從來源檔 `NestedSP.java` 建置 JDBC 從屬站應用程式 `NestedSP`，請編譯來源檔：

```
javac NestedSP.java
```

結果是類別檔 `NestedSP.class`。

若要呼叫儲存程序，請在應用程式上執行 Java 直譯器：

```
java NestedSP userid password database
```

相關作業:

- 第 137 頁的『利用 CALL 陳述式呼叫儲存程序』
- 第 129 頁的『在 Windows 上利用從屬站應用程式呼叫 SQL 程序』
- 第 134 頁的『分送編譯後的 SQL 程序』
- 第 135 頁的『重新連結 SQL 程序』

相關範例:

- 『basecase.sqc -- To call the UPDATE_SALARY SQL procedure』
- 『resultset.c -- To call the MEDIAN_RESULT_SET SQL procedure』

在 Windows 上利用從屬站應用程式呼叫 SQL 程序

您可以建置並執行從屬站應用程式，在 Windows 上呼叫 SQL 程序。DB2 在 `sqllib\samples\sqlproc` 目錄中提供範例從屬站程式，以呼叫可從同時由它提供的範例命令行處理器建立的 SQL 程序。DB2 CLI、C 內嵌式 SQL 及 JDBC 有從屬站來源檔。

提供批次檔 (JDBC 除外) 以建置這些應用程式介面的從屬站程式：

- `bldcli.bat` 含有要建置 SQL 程序之 DB2 CLI 從屬站應用程式的命令。這個檔案同於 `sqllib\samples\cli` 目錄中的 `bldapp.bat` 檔。
- `bldapp.bat` 含有要建置 SQL 程序之 C 內嵌式 SQL 從屬站應用程式的命令。這個檔案同於 `sqllib\samples\c` 目錄中的 `bldapp.bat` 檔。

程序:

根據您要使用的應用程式介面，您可以遵循這些範例，以建置並執行從屬站程式來呼叫 SQL 程序：

DB2 CLI

欲從來源檔 `resultset.c` 建置 DB2 CLI 從屬站應用程式 `resultset`，請輸入：

```
bldcli resultset
```

此命令會建立可執行檔 `resultset.exe`。

欲呼叫儲存程序，請輸入可執行檔名稱、您要連接的資料庫名稱、及資料庫案例的使用者 ID 與通行碼，以執行範例從屬站應用程式：

```
resultset database userid password
```

C 內嵌式 SQL

若要從來源檔 `basecase.sqc` 建置內嵌式 SQL 從屬站應用程式 `basecase`，請輸入批次檔名稱、可執行檔名稱、您要連接的資料庫，以及含有資料庫之案例的使用者 ID 及通行碼：

```
bldapp basecase database userid password
```

結果是可執行檔 `basecase.exe`。

若要呼叫儲存程序，請輸入下列命令來執行範例從屬站應用程式：

```
basecase database userid password
```

JDBC 若要從來源檔 `NestedSP.java` 建置 JDBC 從屬站應用程式 `NestedSP`，請編譯來源檔：

```
javac NestedSP.java
```

結果是類別檔 `NestedSP.class`。

若要呼叫儲存程序，請在應用程式上執行 Java 直譯器：

```
java NestedSP userid password database
```

相關作業:

- 第 137 頁的『利用 CALL 陳述式呼叫儲存程序』
- 第 128 頁的『在 UNIX 上利用從屬站應用程式呼叫 SQL 程序』
- 第 134 頁的『分送編譯後的 SQL 程序』
- 第 135 頁的『重新連結 SQL 程序』

相關範例:

- 『`basecase.sqc` -- To call the UPDATE_SALARY SQL procedure』
- 『`resultset.c` -- To call the MEDIAN_RESULT_SET SQL procedure』

保留 SQL 程序的中間檔

當您發出 CREATE PROCEDURE 陳述式時，DB2 會建立許多中間檔，如果 DB2 順利完成陳述式，這些檔案就會被刪除。如果 SQL 程序沒有如預期地執行，保留檔案以檢查 DB2 所建立的 SQC、C 及訊息日誌檔可能有助於解決問題。

程序:

欲保留 DB2 在順利執行 CREATE PROCEDURE 陳述式期間內所建立的檔案，您必須將伺服器上的 DB2_SQLROUTINE_KEEP_FILES DB2 登錄變數的值設定為 1、y 或 yes，如下列命令所示：

```
db2set DB2_SQLROUTINE_KEEP_FILES=1
```

接著，停止並重新啟動 DB2，讓變更生效。

然後，在未能順利建立 SQL 程序時，可能留下一些中間檔，您將必須手動刪除之。這些檔案位於下列目錄中：

UNIX

```
$HOME/sqllib/function/routine/sqlproc/<db_name>/<schema_name>/tmp
```

Windows

```
sqllib\function\routine\sqlproc\<db_name>\<schema_name>\tmp
```

其中 <db_name> 及 <schema_name> 是用來建立 SQL 程序的資料庫及綱目。

相關作業:

- 第 132 頁的『自訂 SQL 程序的前置編譯及連結選項』
- 第 133 頁的『備份及復置 SQL 程序』
- 第 127 頁的『建立 SQL 程序』
- 第 137 頁的『利用 CALL 陳述式呼叫儲存程序』
- 第 128 頁的『在 UNIX 上利用從屬站應用程式呼叫 SQL 程序』
- 第 129 頁的『在 Windows 上利用從屬站應用程式呼叫 SQL 程序』
- 第 134 頁的『分送編譯後的 SQL 程序』
- 第 135 頁的『重新連結 SQL 程序』

自訂 SQL 程序的前置編譯及連結選項

您可以設定 DB2_SQLROUTINE_PREPOPTS DB2 登錄變數，自訂前置編譯及連結選項。這些選項不能在程序層次上自訂。

程序:

欲指定 SQL 程序的自訂前置編譯選項，請使用下列命令，將 DB2 前置編譯器所使用的前置編譯選項清單放入 DB2 登錄：

```
db2set DB2_SQLROUTINE_PREPOPTS=options
```

其中 *options* 指定 DB2 前置編譯器所使用的前置編譯選項清單。僅接受下列選項：

```
BLOCKING {UNAMBIG | ALL | NO}
DATETIME {DEF | USA | EUR | ISO | JIS | LOC}
DEGREE {1 | degree-of-parallelism | ANY}
DYNAMICRULES {BIND | RUN}
EXPLAIN {NO | YES | ALL}
EXPLAINSAP {NO | YES | ALL}
FEDERATED {NO | YES}
INSERT {DEF | BUF}
ISOLATION {CS |RR |UR |RS |NC}
QUERYOPT optimization-level
SYNCPOINT {ONEPHASE | TWOPHASE | NONE}
```

範例。若要使資料包對日期及「可重複讀取」使用 ISO 格式以作為隔離層次，請指定如下的選項：

```
db2set DB2_SQLROUTINE_PREPOPTS="DATETIME ISO ISOLATION RR"
```

接著，停止並重新啟動 DB2，讓變更生效。

相關作業:

- 第 131 頁的『保留 SQL 程序的中間檔』
- 第 133 頁的『備份及復置 SQL 程序』
- 第 127 頁的『建立 SQL 程序』
- 第 137 頁的『利用 CALL 陳述式呼叫儲存程序』
- 第 128 頁的『在 UNIX 上利用從屬站應用程式呼叫 SQL 程序』
- 第 129 頁的『在 Windows 上利用從屬站應用程式呼叫 SQL 程序』
- 第 134 頁的『分送編譯後的 SQL 程序』
- 第 135 頁的『重新連結 SQL 程序』

備份及復置 SQL 程序

在建立 SQL 程序時，所產生的共用動態鏈結檔案庫 (DLL) 會保留在資料庫型錄中，以及來源文字、資料包及相關檔案也會保留在其中。因此，當您執行資料庫備份時，將儲存此全部資訊。

程序:

在資料庫回復時，將會除去隸屬於回復資料庫之檔案系統上的所有 SQL 程序可執行檔。若將索引建立配置參數 `indexrec` 設定為 `RESTART`，則將從型錄表格中取出所有 SQL 程序可執行檔，並在下次連線時放回到檔案系統中。否則，將在首次執行 SQL 程序時，取出 SQL 可執行檔。

將可執行檔放回到以下目錄中：

UNIX `$HOME/sqllib/function/routine/sqlproc/<database_name>`

Windows

`sqllib\function\routine\sqlproc\<database_name>`

其中 `<database_name>` 代表建立 SQL 程序時所使用的資料庫。

若在復置作業之後，首次嘗試連接資料庫時，傳回以下內容：

```
SQL2048N 在存取物件 "SQL PROCEDURE FILES" 時發生錯誤。  
原因碼："7"。
```

利用 `db2stop` 停止 DB2，利用 `db2start` 重新啟動。

相關作業:

- 第 131 頁的『保留 SQL 程序的中間檔』
- 第 132 頁的『自訂 SQL 程序的前置編譯及連結選項』
- 第 127 頁的『建立 SQL 程序』
- 第 137 頁的『利用 CALL 陳述式呼叫儲存程序』
- 第 128 頁的『在 UNIX 上利用從屬站應用程式呼叫 SQL 程序』
- 第 129 頁的『在 Windows 上利用從屬站應用程式呼叫 SQL 程序』
- 第 134 頁的『分送編譯後的 SQL 程序』
- 第 135 頁的『重新連結 SQL 程序』

分送編譯後的 SQL 程序

當您定義 SQL 程序時，即將其轉換為 C 程式，並在依據目標資料庫進行預先編譯及連結之後，對之進行編譯及鏈結，以建立共用檔案庫。在編譯及鏈結步驟中，需要在資料庫伺服器機器上安裝有 C 或 C++ 編譯器。然而，一旦您定義了 SQL 程序，您就可以編譯後的形式，將其分送到使用同一作業系統及同一 DB2 版本，卻不一定存取 C 或 C++ 編譯器的 DB2 資料庫伺服器機器。出現這些狀況時，DB2 可容許使用者以編譯後的形式，從一資料庫中取出 SQL 程序，並將其以編譯後的形式安裝於其它伺服器機器的另一資料庫中。

DB2 可提供命令行介面，亦可提供程式設計介面，以進行取出及安裝作業。命令行介面包含兩個 CLP 命令：GET ROUTINE 及 PUT ROUTINE。程式設計介面包含兩個內建儲存程序：GET_ROUTINE_SAR 及 PUT_ROUTINE_SAR。

程序：

若要将編譯後的 SQL 程序從一資料庫伺服器分送到另一資料庫伺服器，請執行下列步驟：

1. 建置應用程式，其中包括定義為該應用程式之一部份的 SQL 程序。
2. 在測試程序之後，可藉由發出 GET ROUTINE 命令或呼叫 GET_ROUTINE_SAR 儲存程序，取出每個程序的編譯後版本，並放入其它檔案中。將這些檔案複製到您的分送媒體中 (如有必要)。

範例。假設 SCHEMA1.MYPROC 是存在於來源伺服器上的「SQL 程序」。下列命令將建立名為 myproc.sar 的檔案，其含有可分送的程序表示式。

```
GET ROUTINE INTO myproc.sar FROM PROCEDURE SCHEMA1.MYPROC
```

3. 藉由發出 PUT ROUTINE 命令，或呼叫 PUT_ROUTINE_SAR 儲存程序，使用在上個步驟中建立的檔案，在每台伺服器上安裝每個程序之編譯後的版本。

範例。下列命令將把來自前一個範例的程序安裝在目標伺服器：

```
PUT ROUTINE FROM myproc.sar
```

每台資料庫伺服器必須擁有相同的作業系統及 DB2 層次。

註：如果 GET ROUTINE 或 PUT ROUTINE 作業 (或它們的相對應程序) 無法順利執行，它恆會傳回一個錯誤以及診斷文字，以提供失敗原因的相關資訊。例如，如果提供給 GET ROUTINE 的程序名稱無法識別 SQL 程序，則將傳回診斷 "-204, 42704" 文字，其中 "-204" 及 "42704" 分別代表 SQLCODE 及 SQLSTATE，用以識別問題的原因。這個範例中的 SQLCODE 及 SQLSTATE 指出 GET ROUTINE 命令中所提供的程序名稱尚未定義。

相關作業：

- 第 21 頁的『設定 SQL 程序環境』
- 第 127 頁的『建立 SQL 程序』
- 第 137 頁的『利用 CALL 陳述式呼叫儲存程序』
- 第 128 頁的『在 UNIX 上利用從屬站應用程式呼叫 SQL 程序』
- 第 129 頁的『在 Windows 上利用從屬站應用程式呼叫 SQL 程序』
- 第 135 頁的『重新連結 SQL 程序』

相關參照:

- *SQL Reference, Volume 1* 中的『GET_ROUTINE_SAR procedure』
- *SQL Reference, Volume 1* 中的『PUT_ROUTINE_SAR procedure』
- *Command Reference* 中的『GET ROUTINE Command』
- *Command Reference* 中的『PUT ROUTINE Command』

重新連結 SQL 程序

程序:

若要重新連結對應於 SQL 程序的資料包，請呼叫 SYSPROC.REBIND_ROUTINE_PACKAGE 內建儲存程序。

例如，如果名為 MYSCHEMA.MYPROC 的 SQL 程序存在於資料庫中，則可以從命令行處理器 (CLP) 發出下列命令來重新連結它的資料包：

```
CALL SYSPROC.REBIND_ROUTINE_PACKAGE('P', 'MYSCHEMA.MYPROC', 'CONSERVATIVE')
```

其中 'P' 指出 'MYSCHEMA.MYPROC' 是程序名稱。第一個參數的值 'SP' 將指出 'MYSCHEMA.MYPROC' 是特定的程序名稱。'CONSERVATIVE' 指出應該引用傳統的重新連結語義。如需傳統重新連結的詳細資訊，請參閱底下相關鏈結中的 REBIND 命令。

相關作業:

- 第 21 頁的『設定 SQL 程序環境』
- 第 131 頁的『保留 SQL 程序的中間檔』
- 第 132 頁的『自訂 SQL 程序的前置編譯及連結選項』
- 第 133 頁的『備份及復置 SQL 程序』
- 第 127 頁的『建立 SQL 程序』
- 第 137 頁的『利用 CALL 陳述式呼叫儲存程序』
- 第 128 頁的『在 UNIX 上利用從屬站應用程式呼叫 SQL 程序』

- 第 129 頁的『在 Windows 上利用從屬站應用程式呼叫 SQL 程序』
- 第 134 頁的『分送編譯後的 SQL 程序』

相關參照:

- *Command Reference* 中的『REBIND Command』

第 6 章 呼叫儲存程序

利用 CALL 陳述式呼叫儲存程序 137

本章提供在命令行上以 CALL 陳述式呼叫 DB2 儲存程序的詳細資訊。

關於最新 DB2 應用程式開發更新組件，請探訪下列網頁：

<http://www.ibm.com/software/data/db2/udb/ad>

利用 CALL 陳述式呼叫儲存程序

您可以從 DB2 命令行處理器介面使用 call SQL 陳述式來呼叫儲存程序。呼叫的儲存程序必須已定義在型錄中。

程序:

若要呼叫儲存程序，首先必須連接資料庫：

```
db2 connect to sample user userid using password
```

其中 *userid* 及 *password* 是 *sample* 資料庫所在位置的案例之使用者 ID 及通行碼。

若要使用 call 陳述式，請輸入儲存程序名稱，以及任何 IN 或 INOUT 參數，或 '?' 以作為每個 OUT 參數的位置保留符號。

儲存程序的參數已在程式來源檔中儲存程序的 CREATE PROCEDURE 陳述式中給定。例如，在 SQL 程序來源檔 *whiles.db2* 中，DEPT_MEDIAN 程序的 CREATE PROCEDURE 陳述式會開始：

```
CREATE PROCEDURE DEPT_MEDIAN  
(IN deptNumber SMALLINT, OUT medianSalary DOUBLE)
```

若要呼叫此程序，您必須在 IN 參數 *deptNumber* 中放入一個有效的 SMALLINT 值，並在 OUT 參數中放入問號 '?'。DEPT_MEDIAN 程序會存取範例資料庫的 STAFF 表格。變數 *deptNumber* 指定給 STAFF 表格的 DEPT 直欄，以便您可以從 DEPT 直欄取得有效值：例如，值 51。

現在，您可以使用程序名稱、IN 參數值及 OUT 參數值 (問號 '?')，輸入 call 陳述式。程序的參數必須以括弧括住，如下所示：

```
db2 call dept_median (51, ?)
```

註: 在 UNIX 平台上，括弧對於命令 Shell 具有特殊意義，所以它們的前面須有 "\ " 字元，或是以引號括住，如下所示：

```
db2 "call dept_median (51, ?)"
```

如果您正在使用命令行處理器的交談模式，將不會使用引號。
在執行上述命令後，您應會接收到此結果：

```
Value of output parameters
-----
Parameter Name : MEDIANSALARY
Parameter Value : +1.76545000000000E+004

DB20000I The SQL command completed successfully.
```

相關作業:

- 第 128 頁的『在 UNIX 上利用從屬站應用程式呼叫 SQL 程序』
- 第 129 頁的『在 Windows 上利用從屬站應用程式呼叫 SQL 程序』
- 第 134 頁的『分送編譯後的 SQL 程序』

相關範例:

- 『whiles.db2 -- To create the DEPT_MEDIAN SQL procedure 』
- 『whiles.sqc -- To call the DEPT_MEDIAN SQL procedure 』

第 3 篇 建置及執行特定平台專用的應用程式

第 7 章 AIX

重要注意事項	141	利用架構檔建置 C++ 儲存程序	169
常式的 AIX 匯出檔案	142	利用架構檔建置 C++ 使用者定義的函數	170
AIX 常式及 CREATE 陳述式	142	IBM COBOL Set for AIX	171
取代 AIX 共用檔案庫	143	在 AIX 上架構 IBM COBOL 編譯器	171
在 AIX 上安裝 COBOL 的注意事項	143	在 AIX 上建置 IBM COBOL 應用程式	172
IBM C	144	IBM COBOL 應用程式的建置 Script	174
在 AIX 上建置 C 應用程式	144	AIX IBM COBOL 應用程式編譯及鏈結選 項	174
C 應用程式的建置 Script	146	在 AIX 上建置 IBM COBOL 常式	175
AIX C 應用程式編譯及鏈結選項	146	IBM COBOL 常式的建置 Script	177
在 AIX 上建置 C 常式	148	AIX IBM COBOL 常式編譯及鏈結選項	177
C 常式的建置 Script	151	Micro Focus COBOL	179
AIX C 常式編譯及鏈結選項	152	在 AIX 上架構 Micro Focus COBOL 編譯 器	179
在 AIX 上建置 C 多緒應用程式	153	在 AIX 上建置 Micro Focus COBOL 應用 程式	179
C 多緒應用程式的建置 Script	154	Micro Focus COBOL 應用程式的建置 Script	181
VisualAge C++	155	AIX Micro COBOL 應用程式編譯及鏈結選 項	182
在 AIX 上建置 C++ 應用程式	155	在 AIX 上建置 Micro Focus COBOL 常式	183
C++ 應用程式的建置 Script	157	Micro Focus COBOL 常式的建置 Script	184
AIX C++ 應用程式編譯及鏈結選項	158	AIX Micro COBOL 常式編譯及鏈結選項	185
在 AIX 上建置 C++ 常式	159	REXX	186
C++ 常式的建置 Script	162	在 AIX 上建置 REXX 應用程式	186
AIX C++ 常式編譯及鏈結選項	163		
在 AIX 上建置 C++ 多緒應用程式	164		
C++ 多緒應用程式的建置 Script	166		
VisualAge C++ 架構檔	166		
利用架構檔建置 VisualAge C++ 程式	166		
利用架構檔建置 C++ DB2 API 應用程式	167		
利用架構檔建置 C++ 內嵌式 SQL 應用程 式	168		

本章提供在 AIX 上建置應用程式的詳細資訊。關於適用於 AIX 的最新 DB2 應用程式開發更新組件，請探訪下列網頁：

<http://www.ibm.com/software/data/db2/udb/ad>

重要注意事項

本章提供在各種支援的編譯器上建置 DB2 應用程式的 AIX 特定資訊。

常式的 AIX 匯出檔案

外部常式係在伺服器上編譯，並在伺服器上的共用檔案庫中儲存及執行。編譯常式時會建立這些共用檔案庫。

AIX 需要您提供一個匯出檔，這個檔案會指定檔案庫中哪一些廣域函數可從其外部呼叫。這個檔案必須包括檔案庫中所有常式的名稱。其他 UNIX 平台僅匯出檔案庫中的所有廣域函數。這是 AIX 匯出檔的例子：

```
#! spserver export file
outlanguage
```

匯出檔 `spserver.exp` 會列出儲存程序 `outlanguage`。鏈結器使用 `spserver.exp`，建立共用檔案庫 `spserver`，來包含 `outlanguage` 儲存程序。

AIX 鏈結器文件有關於匯出檔的進一步資訊。

相關概念:

- 第 142 頁的『AIX 常式及 CREATE 陳述式』

AIX 常式及 CREATE 陳述式

下列說明編譯和鏈結常式之間的關係，以及您在 `CREATE FUNCTION` 陳述式的 `EXTERNAL NAME` 子句中提供的資訊。

當您編譯及鏈結您的程式時，您可以使用一個利用 `-bE:` 選項指定的匯出檔案，來識別外部函數。

假設檔案庫 `myrtns` 含有三個常式：`modify`、`remove` 及 `add`。將 `modify` 識別為預設進入點，做法為將它放置在匯出檔案中，作為鏈結步驟中所鏈結的第一個登錄。同時，也將 `remove` 及 `add` 函數併入匯出檔案，指出它們是額外可匯出的函數。

在鏈結步驟中，您指定：

```
-bE:myrtns.exp
```

識別匯出檔 `myrtns.exp`。

匯出檔的樣子如下：

```
  modify
remove
add
```

最後，常式的 EXTERNAL NAME 子句 (利用 modify、remove 及 add 函數來加以實施) 編寫如下：

```
EXTERNAL NAME '/u/mydir/routines/myrtns!modify'
```

及

```
EXTERNAL NAME '/u/mydir/routines/myrtns!remove'
```

及

```
EXTERNAL NAME '/u/mydir/routines/myrtns!add'
```

註：所使用的預設路徑將是 `sqllib/function`。這表示 EXTERNAL NAME 子句是否指定如下：

```
EXTERNAL NAME 'myrtns!modify'
```

DB2 將嘗試從 `sqllib/function` 載入 `myrtns`。

相關概念：

- 第 142 頁的『常式的 AIX 匯出檔案』

取代 AIX 共用檔案庫

程序：

在建置共用檔案庫之後，通常會將它複製到 DB2 將從其中存取它的目錄內。當嘗試取代常式共用檔案庫時，您應該執行 `/usr/sbin/slibclean`，來清除 AIX 共用檔案庫快取，或是從目標目錄中除去檔案庫，然後從來源目錄中，將檔案庫複製到目標目錄中。不然，複製作業可能會失敗，因為 AIX 保有所參照的檔案庫的快取，且不容許檔案庫被改寫。

在 AIX 上安裝 COBOL 的注意事項

因為 AIX 載入常式及解析它們之間的檔案庫參照的方式，所以就有關於應該如何安裝 COBOL 的需求。當 COBOL 程式在執行期間，載入一個共用檔案庫 (常式) 時，這些需求即變成一個因素。

當載入一個常式時，也須載入它所參照的檔案庫鏈。當 AIX 搜尋一個僅被您的程式間接參照的檔案庫時，它必須使用已編入檔案庫中，用來參照它的路徑，而這個路徑是由語言供應商 (IBM COBOL 或 Micro Focus COBOL) 所提供的。這個路徑非常可能不是編譯器安裝所在的同一個路徑。如果找不到鏈中的檔案庫，常式的載入將失敗，而且您將收到 `SQLCODE -444`。

欲確保這種情況不會發生，請在您想要放置編譯器的位置上安裝它，然後從安裝目錄中對所有語言檔案庫，建立與 `/usr/lib` (這是一個當需要載入一個檔案庫時，幾乎恆會搜尋的目錄) 的符號式鏈結。您可以使檔案庫與 `sqllib/function` (常式目錄) 產生鏈結，但這僅對一個資料庫案例有作用；`/usr/lib` 對機器上的所有使用者均有作用。

相關作業:

- 第 22 頁的『設置 UNIX 應用程式開發環境』
- 第 171 頁的『在 AIX 上架構 IBM COBOL 編譯器』
- 第 179 頁的『在 AIX 上架構 Micro Focus COBOL 編譯器』

IBM C

關於 DB2 CLI 應用程式及常式的建置資訊，請參閱 *CLI Guide and Reference*。

在 AIX 上建置 C 應用程式

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/c` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

建置檔 `bldapp` 含有要建置 DB2 應用程式的命令。

第一個參數 `$1`，指定您的來源檔的名稱。這是唯一必要的參數，也是沒有內嵌式 SQL DB2 API 程式唯一需要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 `$2`，指定您要連接的資料庫名稱；第三個參數 `$3` 指定資料庫的使用者 ID，而 `$4` 指定通行碼。

對於內嵌式 SQL 程式，`bldapp` 會傳送參數以前置編譯及連結 Script `embprep`。如果未提供任何資料庫名稱，將使用預設 `sample` 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

下列範例告訴您如何建置及執行 DB2 API 及內嵌式 SQL 應用程式。

若要從來源檔 `cli_info.c` 建置 DB2 API 非內嵌式 SQL 範例程式 `cli_info`，請輸入：

```
bldapp cli_info
```

結果是可執行檔 `cli_info`。

欲執行可執行檔，請輸入可執行檔檔名：

```
cli_info
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 `tbmod.sqc` 建置內嵌式 SQL 應用程式 `tbmod`：

1. 如果連接同一案例上的 `sample` 資料庫，請輸入：

```
bldapp tbmod
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp tbmod database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp tbmod database userid password
```

結果是可執行檔 `tbmod`。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 `sample` 資料庫，則只要輸入可執行檔檔名：

```
tbmod
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
tbmod database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
tbmod database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關作業:

- 第 148 頁的『在 AIX 上建置 C 常式』

相關參照:

- 第 146 頁的『AIX C 應用程式編譯及鏈結選項』
- 第 65 頁的『C/C++ 範例』

相關範例:

- 『`bldapp` -- Builds AIX C application programs』
- 『`cli_info.c` -- Set and get information at the client level (C)』
- 『`embprep` -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』

- 『tbmod.sqc -- How to modify table data (C)』

C 應用程式的建置 Script

```

#!/bin/sh
# SCRIPT: bldapp
# Builds AIX C application programs
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqlllib

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG=-q64
else
    EXTRA_CFLAG=
fi

# If embedded SQL program, precompile and bind it.
if [ -f "$1.sqc" ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.c error-checking utility.
    xlc $EXTRA_CFLAG -I$DB2PATH/include -c utilemb.c
else
    # Compile the utilapi.c error-checking utility.
    xlc $EXTRA_CFLAG -I$DB2PATH/include -c utilapi.c
fi

# Compile the program.
xlc $EXTRA_CFLAG -I$DB2PATH/include -c $1.c

if [ -f "$1.sqc" ]
then
    # Link the program with utilemb.o
    xlc $EXTRA_CFLAG -o $1 $1.o utilemb.o -ldb2 -L$DB2PATH/lib
else
    # Link the program with utilapi.o
    xlc $EXTRA_CFLAG -o $1 $1.o utilapi.o -ldb2 -L$DB2PATH/lib
fi

```

AIX C 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 AIX IBM C 編譯器建置 C 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp 建置 Script 中所示範一般。

bldapp 的編譯及鏈結選項

編譯選項：

xlc IBM C 編譯器。

\$EXTRA_CFLAG

如果 'BUILD_64BIT=true' 未加上註解，則含有 "-q64" 值；否則，即不含任何值。

-I\$DB2PATH/include

指定 DB2 併入檔的位置。例如：\$HOME/sqllib/include。

-c 僅執行編譯；沒有任何鏈結。編譯及鏈結是不同的步驟。

鏈結選項：

xlc 使用編譯器作為鏈結器的前端。

\$EXTRA_CFLAG

如果 'BUILD_64BIT=true' 未加上註解，則含有 "-q64" 值；否則，即不含任何值。

-o \$1 指定可執行檔程式。

\$1.o 指定程式目的檔。

utilemb.o

如果是內嵌式 SQL 程式，則有內嵌式 SQL 公用程式的目的檔，可執行錯誤檢查。

utilapi.o

如果不是內嵌式 SQL 程式，則有 DB2 API 公用程式目的檔，可執行錯誤檢查。

-ldb2 鏈結 DB2 檔案庫。

-L\$DB2PATH/lib

指定 DB2 執行程式共用檔案庫的位置。例如：\$HOME/sqllib/lib。如果您未指定 -L 選項，則編譯器將採用下列路徑：/usr/lib:/lib。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 144 頁的『在 AIX 上建置 C 應用程式』

相關參照：

- 第 152 頁的『AIX C 常式編譯及鏈結選項』

相關範例：

- 『bldapp -- Builds AIX C application programs』

在 AIX 上建置 C 常式

DB2 會提供建置 Script，來編譯及鏈結 C 程式。這些都位於 `sqllib/samples/c` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script `bldrtn` 含有要建置常式 (儲存程序及使用者定義的函數) 的命令。Script 會編譯常式，並將它們放入可被資料庫管理程式載入及可被從屬站應用程式呼叫的共用檔案庫。

第一個參數 `$1`，指定您的來源檔的名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。

資料庫參數是選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。因此，既然儲存程序必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

程序:

下列範例告訴您如何利用下列建置常式共用檔案庫：

- 儲存程序
- 非內嵌式 SQL 使用者定義的函數 (UDF)
- 內嵌式 SQL 使用者定義的函數 (UDF)

儲存程序共用檔案庫

若要从來源檔 `spserver.sqc` 建置範例程式 `spserver`：

1. 如果連接到 `sample` 資料庫，請輸入建置 Script 名稱及程式名稱：

```
bldrtn spserver
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn spserver database
```

Script 會將共用檔案庫複製到路徑 `sqllib/function` 中的伺服器。

2. 接著，在伺服器上執行 `spcat` Script，將常式編目：

```
spcat
```

這個 Script 將連接到範例資料庫、將取消常式編目 (若之前已呼叫 `spdrop.db2` 將它們編目的話)，然後呼叫 `spcreate.db2` 來將它們編目，最後切斷資料庫連線。您也可以個別地呼叫 `spdrop.db2` 及 `spcreate.db2` Script。

3. 然後，停止資料庫後再重新啟動，以辨識新的共用檔案庫。

一旦建置共用檔案庫 `spserver` 完成，您即可建置存取共用檔案庫的從屬站應用程式 `spclient`。

您可以使用 Script `bldapp` 建置 `spclient`。

欲呼叫共用檔案庫中的儲存程序，請輸入下列命令以執行範例從屬站應用程式：

```
spclient database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式可存取共用檔案庫 `spserver`，並在伺服器資料庫中執行許多儲存程序函數。輸出會傳回從屬站應用程式。

非內嵌式 SQL UDF 共用檔案庫

若要從來源檔 `udfsrv.c` 建置使用者定義的函數程式 `udfsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfsrv
```

Script 會將 UDF 複製到 `sqllib/function` 目錄。

一旦您建置了 `udfsrv`，您就可以建置呼叫它的從屬站應用程式 `udfcli`。已提供 DB2 CLI 及此程式的內嵌式 SQL 版本。您可以使用 Script `bldapp`，從來源檔 `udfcli.c` (位於 `sqllib/samples/cli`) 建置 DB2 CLI `udfcli` 從屬站程式。

您可以使用 Script `bldapp`，從來源檔 `udfcli.sqc` (位於 `sqllib/samples/c`) 建置內嵌式 SQL `udfcli` 從屬站程式。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

內嵌式 SQL UDF 共用檔案庫

當連接到 `sample` 資料庫時，若要從來源檔 `udfemsrv.sqc` 建置內嵌式 SQL 使用者定義的函數程式 `udfemsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfemsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn udfemsrv database
```

Script 會將 UDF 複製到 `sqllib/function` 目錄。

一旦您建置了 `udfemsrv`，您就可以建置呼叫它的從屬站應用程式 `udfemcli`。您可以使用 Script `bldapp`，從來源檔 `udfemcli.sqc` (位於 `sqllib/samples/c`) 建置 `udfemcli` 從屬站程式。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfemcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfemsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關作業:

- 第 144 頁的『在 AIX 上建置 C 應用程式』

相關參照:

- 第 152 頁的『AIX C 常式編譯及鏈結選項』
- 第 65 頁的『C/C++ 範例』

相關範例:

- 『bldrtn -- Builds AIX C routines (stored procedures and UDFs)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcat -- To catalog C stored procedures on UNIX』
- 『spcliient.sqc -- Call various stored procedures (C)』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqc』
- 『spserver.sqc -- A variety of types of stored procedures (C)』
- 『udfcli.sqc -- Call a variety of types of user-defined functions (C)』
- 『udfemcli.sqc -- Call a variety of types of embedded SQL user-defined functions. (C)』
- 『udfemsrv.sqc -- Call a variety of types of embedded SQL user-defined functions. (C)』
- 『udfsrv.c -- Call a variety of types of user-defined functions (C)』

C 常式的建置 Script

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds AIX C routines (stored procedures and UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1ib

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
```

```

    EXTRA_CFLAG=-q64
else
    EXTRA_CFLAG=
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sql" ]
then
    ./embprep $1 $2
fi

# Compile the program.
xlc_r $EXTRA_CFLAG -I$DB2PATH/include -c $1.c

# Link the program using the export file $1.exp,
xlc_r $EXTRA_CFLAG -qmkshrobj -o $1 $1.o -ldb2 -L$DB2PATH/lib -bE:$1.exp

# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

AIX C 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 AIX IBM C 編譯器建置 C 常式（儲存程序及使用者定義的函數），如同 bldrtn 建置 Script 中所示範一般。

bldrtn 的編譯及鏈結選項	
編譯選項：	
xlc_r	需要使用多緒版本的 IBM C 編譯器，因為常式可能在其他常式 (THREADSAFE) 執行所在的同一程序中執行，或是在引擎本身 (NOT FENCED) 中執行。
\$EXTRA_CFLAG	如果 'BUILD_64BIT=true' 未加上註解，則含有 "-q64" 值；否則，即不含任何值。
-I\$DB2PATH/include	指定 DB2 併入檔的位置。例如：\$HOME/sqllib/include。
-c	僅執行編譯；沒有任何鏈結。編譯及鏈結是不同的步驟。

bldrtn 的編譯及鏈結選項

鏈結選項：

xlc_r 使用多緒版本的編譯器作為鏈結器的前端。

\$EXTRA_CFLAG

如果 'BUILD_64BIT=true' 未加上註解，則含有 "-q64" 值；否則，即不含任何值。

-qmkshrobj

建立共用檔案庫。

-o \$1 指定輸出檔名稱。

\$1.o 指定目的檔。

-ldb2 與 DB2 檔案庫鏈結。

-L\$DB2PATH/lib

指定 DB2 執行程式共用檔案庫的位置。例如：\$HOME/sqllib/lib。如果您未指定 -L 選項，則編譯器將採用下列路徑： /usr/lib:/lib。

-bE:\$1.exp

指定匯出檔。匯出檔含有常式的清單。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 148 頁的『在 AIX 上建置 C 常式』

相關參照：

- 第 146 頁的『AIX C 應用程式編譯及鏈結選項』

相關範例：

- 『bldrtn -- Builds AIX C routines (stored procedures and UDFs)』

在 AIX 上建置 C 多緒應用程式

必須編譯 AIX 上的 C 多緒應用程式，並且鏈結 xlc_r 編譯器代替 xlc 編譯器，或者鏈結 xlc_r 編譯器代替 xlc 編譯器 (就 C++ 而言)。_r 版本會為多緒編譯設定適當的前置處理器定義，並為鏈結器提供適當的緒檔案庫名稱。

您可從編譯器文件中取得使用多緒編譯器前端設定編譯器及鏈結旗號的附加資訊。

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 sqllib/samples/c 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

在 `sqllib/samples/c` 中的 Script 檔 `bldmt` 含有建置內嵌式 SQL 多緒程式的命令。第一個參數 \$1，指定您的來源檔的名稱。第二個參數 \$2，指定您想要與其連接的資料庫的名稱。參數 \$3 指定資料庫的使用者 ID，而 \$4 則指定使用者 ID 的通行碼。只需要第一個參數 (來源檔名稱)。資料庫名稱、使用者 ID 及通行碼均為可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

除了 `xlc_r` 編譯器以及沒有鏈結的公用程式檔這兩點以外，其它編譯及鏈結選項均和內嵌式 SQL Script 檔 `bldapp` 所使用的選項相同。

程序:

若要從來源檔 `dbthrds.sqc` 建置多緒範例程式 `dbthrds`，請輸入：

```
bldmt dbthrds
```

結果是可執行檔 `dbthrds`。若要對 `sample` 資料庫執行可執行檔，請輸入這個可執行檔名稱：

```
dbthrds
```

相關概念:

- 第 90 頁的『建置檔』

相關作業:

- 第 144 頁的『在 AIX 上建置 C 應用程式』

相關參照:

- 第 146 頁的『AIX C 應用程式編譯及鏈結選項』
- 第 65 頁的『C/C++ 範例』

相關範例:

- 『`bldmt -- Builds AIX C multi-threaded applications`』
- 『`dbthrds.sqc -- How to use multiple context APIs on UNIX (C)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

C 多緒應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldmt
# Builds AIX C multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib
```

```

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG=-q64
else
    EXTRA_CFLAG=
fi

# If embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
fi

# Compile the program.
xlc_r $EXTRA_CFLAG -I$DB2PATH/include -c $1.c

# Link the program.
xlc_r $EXTRA_CFLAG -o $1 $1.o -L$DB2PATH/lib -ldb2

```

VisualAge C++

在 AIX 上建置 C++ 應用程式

DB2 會提供建置 Script，來編譯及鏈結 C++ 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/cpp` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

建置檔 `bldapp` 含有要建置 DB2 API 及內嵌式 SQL 應用程式的命令。

第一個參數 `$1`，指定您的來源檔的名稱。這是唯一必要的參數，也是沒有內嵌式 SQL DB2 API 程式唯一需要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 `$2`，指定您要連接的資料庫名稱；第三個參數 `$3` 指定資料庫的使用者 ID，而 `$4` 指定通行碼。

對於內嵌式 SQL 程式，`bldapp` 會傳送參數以前置編譯及連結 Script `embprep`。如果未提供任何資料庫名稱，將使用預設 `sample` 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

下列範例告訴您如何建置及執行 DB2 API 及內嵌式 SQL 應用程式。

若要從來源檔 `cli_info.C` 建置非內嵌式 SQL 範例程式 `cli_info`，請輸入：

```
bldapp cli_info
```

結果是可執行檔 `cli_info`。您可以輸入下面的命令，對 `Sample` 資料庫執行可執行檔：

```
cli_info
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 `tbmod.sqC` 建置內嵌式 SQL 應用程式 `tbmod`：

1. 如果連接同一案例上的 `sample` 資料庫，請輸入：

```
bldapp tbmod
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp tbmod database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp tbmod database userid password
```

結果是可執行檔 `tbmod`。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 `sample` 資料庫，則只要輸入可執行檔檔名：

```
tbmod
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
tbmod database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
tbmod database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關作業:

- 第 159 頁的『在 AIX 上建置 C++ 常式』

相關參照:

- 第 158 頁的『AIX C++ 應用程式編譯及鏈結選項』
- 第 65 頁的『C/C++ 範例』

相關範例:

- 『`bldapp -- Builds AIX C++ applications`』
- 『`cli_info.C -- Set and get information at the client level (C++)`』

- 『tbmod.sqlC -- How to modify table data (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』

C++ 應用程式的建置 Script

```

#!/bin/sh
# SCRIPT: bldapp
# Builds AIX C++ applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG=-q64
else
    EXTRA_CFLAG=
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqlC" ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.C error-checking utility.
    x1C $EXTRA_CFLAG -I$DB2PATH/include -c utilemb.C
else
    # Compile the utilapi.C error-checking utility.
    x1C $EXTRA_CFLAG -I$DB2PATH/include -c utilapi.C
fi

# Compile the program.
x1C $EXTRA_CFLAG -I$DB2PATH/include -c $1.C

if [ -f $1".sqlC" ]
then
    # Link the program with utilemb.o
    x1C $EXTRA_CFLAG -o $1 $1.o utilemb.o -ldb2 -L$DB2PATH/lib
else
    # Link the program with utilapi.o
    x1C $EXTRA_CFLAG -o $1 $1.o utilapi.o -ldb2 -L$DB2PATH/lib
fi

```

AIX C++ 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 AIX IBM VisualAge C++ 編譯器建置 C++ 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp 建置 Script 中所示範一般。

bldapp 的編譯及鏈結選項	
編譯選項：	
x1C	VisualAge C++ 編譯器。
EXTRA_CFLAG	
	如果 'BUILD_64BIT=true' 未加上註解，則含有 "-q64" 值；否則，即不含任何值。
-I\$DB2PATH/include	
	指定 DB2 併入檔的位置。例如：\$HOME/sqlllib/include。
-c	
	僅執行編譯；沒有任何鏈結。編譯及鏈結是不同的步驟。
鏈結選項：	
x1C	使用編譯器作為鏈結器的前端。
EXTRA_CFLAG	
	如果 'BUILD_64BIT=true' 未加上註解，則含有 "-q64" 值；否則，即不含任何值。
-o \$1	指定可執行檔程式。
\$1.o	指定程式目的檔。
utilapi.o	
	包括非內嵌式 SQL 程式之 API 公用程式目的檔。
utilemb.o	
	包括內嵌式 SQL 程式之內嵌式 SQL 公用程式目的檔。
-ldb2	與 DB2 檔案庫鏈結。
-L\$DB2PATH/lib	
	指定 DB2 執行程式共用檔案庫的位置。例如：\$HOME/sqlllib/lib。若未指定 -L 選項，編譯器會假設下列路徑 /usr/lib:/lib。
請參閱您的編譯器文件，取得其他編譯器選項。	

相關作業：

- 第 155 頁的『在 AIX 上建置 C++ 應用程式』
- 第 168 頁的『利用架構檔建置 C++ 內嵌式 SQL 應用程式』
- 第 167 頁的『利用架構檔建置 C++ DB2 API 應用程式』

相關參照：

- 第 163 頁的『AIX C++ 常式編譯及鏈結選項』

相關範例:

- 『bldapp -- Builds AIX C++ applications』

在 AIX 上建置 C++ 常式

DB2 會提供建置 Script，來編譯及鏈結 C++ 程式。這些都位於 `sqllib/samples/cpp` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script 檔 `bldrtn` 含有要建置常式的命令。Script 檔會編譯常式，並將它放入可被資料庫管理程式載入及可被從屬站應用程式呼叫的共用檔案庫。

第一個參數 \$1，指定您的來源檔的名稱。第二個參數 \$2，指定您想要與其連接的資料庫的名稱。

資料庫參數是選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。因此，既然儲存程序必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

程序:

下列範例告訴您如何利用下列建置常式共用檔案庫：

- 儲存程序
- 非內嵌式 SQL 使用者定義的函數 (UDF)
- 內嵌式 SQL 使用者定義的函數 (UDF)

儲存程序共用檔案庫

若要從來源檔 `spserver.sqc` 建置範例程式 `spserver`：

1. 如果連接到 `sample` 資料庫，請輸入建置 Script 名稱及程式名稱：

```
bldrtn spserver
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn spserver database
```

Script 檔會將共用檔案庫複製到伺服器的路徑 `sqllib/function` 中。

2. 接著，在伺服器上執行 `spcat` Script，將常式編目：

```
spcat
```

這個 Script 將連接到範例資料庫、將取消常式編目 (若之前已呼叫 `spdrop.db2` 將它們編目的話)，然後呼叫 `spcreate.db2` 來將它們編目，最後切斷資料庫連線。您也可以個別地呼叫 `spdrop.db2` 及 `spcreate.db2` Script。

3. 然後，停止資料庫後再重新啟動，以辨識新的共用檔案庫。

一旦建置共用檔案庫 `spserver` 完成，您即可建置存取共用檔案庫的從屬站應用程式 `spclient`。您可以使用 Script 檔 `bldapp` 建置 `spclient`。

欲呼叫共用檔案庫中的儲存程序，請輸入下列命令以執行範例從屬站應用程式：

```
spclient database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式可存取共用檔案庫 `spserver`，並在伺服器資料庫中執行許多儲存程序函數。輸出會傳回從屬站應用程式。

非內嵌式 SQL UDF 共用檔案庫

若要從來源檔 `udfsrv.C` 建置使用者定義的函數程式 `udfsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfsrv
```

Script 檔會將 UDF 複製到 `sqllib/function` 目錄。

必要時，請設定 UDF 的檔案模式，以便資料庫管理程式可以存取它。

一旦您建置了 `udfsrv`，您就可以建置呼叫它的從屬站應用程式 `udfcli`。您可以使用 Script 檔 `bldapp` 從來源檔 `udfcli.sqc` 建置 `udfcli`。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

內嵌式 SQL UDF 共用檔案庫

當連接到 `sample` 資料庫時，若要從來源檔 `udfemsrv.sql` 建置內嵌式 SQL 使用者定義的函數程式 `udfemsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfemsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn udfemsrv database
```

Script 檔會將 UDF 複製到 `sqllib/function` 目錄。

一旦您建置了 `udfemsrv`，您就可以建置呼叫它的從屬站應用程式 `udfemcli`。您可以使用 Script 檔 `bldapp` 從來源檔 `udfemcli.sql` 建置 `udfemcli`。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfemcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfemsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關作業:

- 第 155 頁的『在 AIX 上建置 C++ 應用程式』

相關參照:

- 第 163 頁的『AIX C++ 常式編譯及鏈結選項』
- 第 65 頁的『C/C++ 範例』

相關範例:

- 『bldrtn -- Builds AIX C++ routines (stored procedures and UDFs)』
- 『spclient.sqC -- Call various stored procedures (C++)』
- 『spserver.sqC -- A variety of types of stored procedures (C++)』
- 『udfcli.sqC -- Call a variety of types of user-defined functions (C++)』
- 『udfemcli.sqC -- Call a variety of types of embedded SQL user-defined functions. (C++)』
- 『udfemsrv.sqC -- Call a variety of types of embedded SQL user-defined functions. (C++)』
- 『udfsrv.C -- Call a variety of types of user-defined functions (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcat -- To catalog C stored procedures on UNIX』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc
』
- 『spdrow.db2 -- How to uncatalog the stored procedures contained in spserver.sqc
』

C++ 常式的建置 Script

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds AIX C++ routines (stored procedures and UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql/lib

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
```

```

    EXTRA_CFLAG=-q64
else
    EXTRA_CFLAG=
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqC" ]
then
    ./embprep $1 $2
fi

# Compile the program.
x1C_r $EXTRA_CFLAG -I$DB2PATH/include -c $1.C

# Link using export file $1.exp, creating shared library $1
x1C_r $EXTRA_CFLAG -qmksrobj -o $1 $1.o -L$DB2PATH/lib -ldb2 -bE $1.exp

# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

AIX C++ 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 AIX VisualAge C++ 編譯器建置 C++ 常式 (儲存程序及使用者定義的函數)，如同 bldrtn 建置 Script 中所示範一般。

bldrtn 的編譯及鏈結選項	
編譯選項：	
x1C_r	需要使用多緒版本的 IBM VisualAge C++ 編譯器，因為常式可能在其他常式 (THREADSAFE) 執行所在的同一程序中執行，或是在引擎本身 (NOT FENCED) 中執行。
\$EXTRA_CFLAG	如果 'BUILD_64BIT=true' 未加上註解，則含有 "-q64" 值；否則，即不含任何值。
-I\$DB2PATH/include	指定 DB2 併入檔的位置。例如：\$HOME/sqllib/include。
-c	僅執行編譯；沒有任何鏈結。編譯及鏈結是不同的步驟。

bldrtn 的編譯及鏈結選項

鏈結選項：

x1C_r 使用多緒版本的編譯器作為鏈結器的前端。

\$EXTRA_CFLAG

如果 'BUILD_64BIT=true' 未加上註解，則含有 "-q64" 值；否則，即不含任何值。

-qmkshrojb

建立共用檔案庫。

-o \$1 將輸出檔指定為共用檔案庫檔案。

\$1.o 指定程式目的檔。

-L\$DB2PATH/lib

指定 DB2 執行程式共用檔案庫的位置。例如：\$HOME/sqllib/lib。如果您未指定 -L 選項，則編譯器將採用下列路徑： /usr/lib:/lib。

-ldb2 與 DB2 檔案庫鏈結。

-bE \$1.exp

指定匯出檔。匯出檔含有常式的清單。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 148 頁的『在 AIX 上建置 C 常式』
- 第 169 頁的『利用架構檔建置 C++ 儲存程序』
- 第 170 頁的『利用架構檔建置 C++ 使用者定義的函數』

相關參照：

- 第 158 頁的『AIX C++ 應用程式編譯及鏈結選項』

相關範例：

- 『bldrtn -- Builds AIX C++ routines (stored procedures and UDFs)』

在 AIX 上建置 C++ 多緒應用程式

必須編譯 AIX 上的 C++ 多緒應用程式，並且鏈結 x1C_r 編譯器代替 x1C 編譯器，或者鏈結 x1c_r 編譯器代替 x1c 編譯器（就 C 而言）。_r 版本會為多緒編譯設定適當的前置處理器定義，並為鏈結器提供適當的緒檔案庫名稱。

您可從編譯器文件中取得使用多緒編譯器前端設定編譯器及鏈結旗號的附加資訊。

DB2 會提供建置 Script，來編譯及鏈結 C++ 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/cpp` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script `bldmt` 含有要建置多緒應用程式的命令。

第一個參數 `$1`，指定您的來源檔的名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。參數 `$3` 指定資料庫的使用者 ID，而 `$4` 則指定使用者 ID 的通行碼。只需要第一個參數 (來源檔名稱)。資料庫名稱、使用者 ID 及通行碼均為可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

除了前面討論的 `x1C_r` 編譯器以及沒有鏈結的這兩點公用程式檔以外，編譯及鏈結選項均與內嵌式 SQL Script 檔 `bldapp` 中所使用的選項相同。

程序:

若要從來源檔 `dbthrd.sqC` 建置多緒範例程式 `dbthrds`，請輸入：

```
bldmt dbthrds
```

結果是可執行檔 `dbthrds`。若要對 `sample` 資料庫執行可執行檔，請輸入這個可執行檔名稱：

```
dbthrds
```

相關概念:

- 第 90 頁的『建置檔』

相關作業:

- 第 155 頁的『在 AIX 上建置 C++ 應用程式』

相關參照:

- 第 158 頁的『AIX C++ 應用程式編譯及鏈結選項』
- 第 65 頁的『C/C++ 範例』

相關範例:

- 『`bldmt -- Builds AIX C++ multi-threaded applications`』
- 『`dbthrd.sqC -- How to use multiple context APIs on UNIX (C++)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

C++ 多緒應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldmt
# Builds AIX C++ multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Precompile and bind the program.
./embprep $1 $2 $3 $4

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG=-q64
else
    EXTRA_CFLAG=
fi

# Compile the program.
xlc_r $EXTRA_CFLAG -I$DB2PATH/include -c $1.c

# Link the program.
xlc_r $EXTRA_CFLAG -o $1 $1.o -L$DB2PATH/lib -ldb2
```

VisualAge C++ 架構檔

註：關於 CLI 應用程式及常式的建置資訊，請參閱 *CLI Guide and Reference*。

利用架構檔建置 VisualAge C++ 程式

VisualAge C++ 版本 5.0 具有漸進式編譯器及批次模式編譯器。當批次模式編譯器使用 `make` 檔及建置檔時，漸進式編譯器將改用架構檔。請參閱附於 VisualAge C++ 版本 5.0 的文件以瞭解這種情況。

DB2 爲您可使用 VisualAge C++ 編譯器建置的不同類型 DB2 程式提供一些架構檔。

程序：

若要使用某 DB2 架構檔，您必須先對想要編譯的程式名稱設定環境變數。然後使用 VisualAge C++ 提供的命令編譯該程式。底下主題說明如何使用 DB2 提供的架構檔來編譯不同類型的程式：

- 利用架構檔建置 C++ 內嵌式 SQL 應用程式
- 利用架構檔建置 C++ DB2 API 應用程式
- 利用架構檔建置 C++ 儲存程序
- 利用架構檔建置 C++ 使用者定義的函數

相關作業:

- 第 168 頁的『利用架構檔建置 C++ 內嵌式 SQL 應用程式』
- 第 167 頁的『利用架構檔建置 C++ DB2 API 應用程式』
- 第 169 頁的『利用架構檔建置 C++ 儲存程序』
- 第 170 頁的『利用架構檔建置 C++ 使用者定義的函數』
- 第 144 頁的『在 AIX 上建置 C 應用程式』
- 第 148 頁的『在 AIX 上建置 C 常式』
- 第 155 頁的『在 AIX 上建置 C++ 應用程式』
- 第 159 頁的『在 AIX 上建置 C++ 常式』

利用架構檔建置 C++ DB2 API 應用程式

架構檔 `api.icc` (位於 `sqllib/samples/c` 及 `sqllib/samples/cpp`) 可讓您在 AIX 上以 C 及 C++ 建置 DB2 API 應用程式。

程序:

若要使用架構檔，從來源檔 `cli_info.c` 建置 DB2 API 範例程式 `cli_info`，請執行下列動作：

1. 輸入下列命令將 API 環境變數設定為程式名稱：

- 若為 `bash` 或 Korn shell：

```
export API=cli_info
```

- 若為 C shell：

```
setenv API cli_info
```

2. 若工作目錄有 `api.ics` 檔 (使用 `api.icc` 檔建置另一個程式產生該檔案)，請使用此命令刪除 `api.ics` 檔：

```
rm api.ics
```

不必刪除針對您再次建置的相同程式而產生的現存 `api.ics` 檔。

3. 編譯範例程式，請輸入：

```
vacbld api.icc
```

註: vacbld 命令是由 VisualAge C++ 提供的。

結果是可執行檔 cli_info。您可以輸入可執行檔名稱，以執行該程式：

```
cli_info
```

相關作業:

- 第 168 頁的『利用架構檔建置 C++ 內嵌式 SQL 應用程式』
- 第 169 頁的『利用架構檔建置 C++ 儲存程序』
- 第 170 頁的『利用架構檔建置 C++ 使用者定義的函數』

利用架構檔建置 C++ 內嵌式 SQL 應用程式

架構檔 emb.icc (位於 sqllib/samples/c 及 sqllib/samples/cpp) 可讓您在 AIX 上以 C 及 C++ 建置 DB2 內嵌式 SQL 應用程式。

程序:

若要使用架構檔，從來源檔 tbmod.sqc 建置內嵌式 SQL 應用程式 tbmod，請執行下列動作：

1. 輸入下列命令將 EMB 環境變數設定為程式名稱：

- 若為 bash 或 Korn shell：

```
export EMB=tbmod
```

- 若為 C shell：

```
setenv EMB tbmod
```

2. 若工作目錄有 emb.ics 檔 (使用 emb.icc 檔建置另一個程式產生該檔案)，請使用此命令刪除 emb.ics 檔：

```
rm emb.ics
```

不必刪除針對您再次建置的相同程式而產生的現存 emb.ics 檔。

3. 編譯範例程式，請輸入：

```
vacbld emb.icc
```

註: vacbld 命令是由 VisualAge C++ 提供的。

結果是可執行檔 tbmod。您可以輸入可執行檔名稱，以執行該程式：

```
tbmod
```

相關作業:

- 第 167 頁的『利用架構檔建置 C++ DB2 API 應用程式』

- 第 169 頁的『利用架構檔建置 C++ 儲存程序』
- 第 170 頁的『利用架構檔建置 C++ 使用者定義的函數』

利用架構檔建置 C++ 儲存程序

架構檔 `stp.icc` (位於 `sqllib/samples/c` 及 `sqllib/samples/cpp`) 可讓您在 AIX 上以 C 及 C++ 建置 DB2 內嵌式 SQL 儲存程序。

程序:

若要使用架構檔，從來源檔 `spserver.sqc` 建置內嵌式 SQL 儲存程序共用檔案庫 `spserver`，請執行下列動作：

1. 輸入下列命令將 STP 環境變數設定為程式名稱：

- 若為 `bash` 或 `Korn shell`：

```
export STP=spserver
```

- 若為 `C shell`：

```
setenv STP spserver
```

2. 若工作目錄有 `stp.ics` 檔 (使用 `stp.icc` 檔建置另一個程式產生該檔案)，請使用此命令刪除 `stp.ics` 檔：

```
rm stp.ics
```

不必刪除針對您再次建置的相同程式而產生的現存 `stp.ics` 檔。

3. 編譯範例程式，請輸入：

```
vacbld stp.icc
```

註: `vacbld` 命令是由 VisualAge C++ 提供的。

儲存程序共用檔案庫會複製到路徑 `sqllib/function` 中的伺服器。

接著，在伺服器上執行 `spcat` Script，將共用檔案庫中的儲存程序編目：

```
spcat
```

這個 Script 將連接到範例資料庫、將取消儲存程序編目 (若之前已呼叫 `spdrop.db2` 將它們編目的話)，然後呼叫 `spcreate.db2` 來將它們編目，最後切斷資料庫連線。您也可以個別地呼叫 `spdrop.db2` 及 `spcreate.db2` Script。

然後，停止資料庫後再重新啟動，以辨識新的共用檔案庫。必要時，請設定共用檔案庫的檔案模式，以便 DB2 案例可以存取檔案庫。

一旦建置了儲存程序共用檔案庫 `spserver` 後，您就可以建置從屬站應用程式 `spclient`，呼叫其中的儲存程序。您可以使用架構檔 `emb.icc` 建置 `spclient`。

若要呼叫儲存程序，請輸入下列命令來執行範例從屬站應用程式：

```
spclient database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample` 或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為有效的通行碼。

從屬站應用程式可存取共用檔案庫 `spserver`，並在伺服器資料庫中執行許多儲存程序函數。輸出會傳回從屬站應用程式。

相關作業：

- 第 168 頁的『利用架構檔建置 C++ 內嵌式 SQL 應用程式』
- 第 167 頁的『利用架構檔建置 C++ DB2 API 應用程式』
- 第 170 頁的『利用架構檔建置 C++ 使用者定義的函數』

利用架構檔建置 C++ 使用者定義的函數

架構檔 `udf.icc` (位於 `sqllib/samples/c` 及 `sqllib/samples/cpp`) 可讓您在 AIX 上以 C 及 C++ 建置使用者定義的函數。

程序：

若要使用架構檔，從來源檔 `udfsrv.c` 建置使用者定義的函數程式 `udfsrv`，請執行下列動作：

1. 輸入下列命令將 UDF 環境變數設定為程式名稱：

- 若為 `bash` 或 `Korn shell`：

```
export UDF=udfsrv
```

- 若為 `C shell`：

```
setenv UDF udfsrv
```

2. 若工作目錄有 `udf.ics` 檔 (使用 `udf.icc` 檔建置另一個程式產生該檔案)，請使用此命令刪除 `udf.ics` 檔：

```
rm udf.ics
```

不必刪除針對您再次建置的相同程式而產生的現存 `udf.ics` 檔。

3. 編譯範例程式，請輸入：

```
vacbld udf.icc
```

註： `vacbld` 命令是由 VisualAge C++ 提供的。

UDF 檔案庫會複製到路徑 `sqllib/function` 中的伺服器。

必要時請設定使用者定義的函數的檔案模式，使 DB2 案例能執行它。

一旦您建置了 `udfsrv`，您就可以建置呼叫它的從屬站應用程式 `udfcli`。已提供 DB2 CLI 及此程式的內嵌式 SQL 版本。

您可以使用架構檔 `cli.icc`，從來源檔 `udfcli.c` (位於 `sqllib/samples/cli`) 建置 DB2 CLI `udfcli` 程式。

您可以使用架構檔 `emb.icc`，從來源檔 `udfcli.sqc` (位於 `sqllib/samples/c`) 建置內嵌式 SQL `udfcli` 程式。

欲呼叫 UDF，請輸入可執行檔名稱，以執行範例呼叫應用程式：

```
udfcli
```

呼叫應用程式會從 `udfsrv` 檔案庫呼叫 `ScalarUDF` 函數。

相關作業：

- 第 168 頁的『利用架構檔建置 C++ 內嵌式 SQL 應用程式』
- 第 167 頁的『利用架構檔建置 C++ DB2 API 應用程式』
- 第 169 頁的『利用架構檔建置 C++ 儲存程序』

IBM COBOL Set for AIX

在 AIX 上架構 IBM COBOL 編譯器

如果您想要開發內嵌式 SQL 及 DB2 API 呼叫的應用程式，而且您將使用 IBM COBOL Set for AIX 編譯器，則下列是您需要採取的步驟。

程序：

- 當您使用命令行處理器命令 `db2 prep`，來前置編譯您的應用程式時，請使用 `target ibmcob` 選項。

- 請勿在您的來源檔中使用 Tab 鍵字元。
- 您可以在您的來源檔的第一行中，使用 PROCESS 及 CBL 關鍵字，來設定編譯選項。
- 如果您的應用程式僅內嵌式 SQL，且沒有任何 DB2 API 呼叫，則您不需要使用 pgmname(mixed) 編譯選項。如果您使用 DB2 API 呼叫，則您必須使用 pgmname(mixed) 編譯選項。
- 如果您使用的是 IBM COBOL Set for AIX 編譯器的「System/390 主電腦資料類型支援」功能，則應用程式的 DB2 併入檔會位於下列目錄：

```
$HOME/sqlllib/include/cobol_i
```

若使用提供的 Script 檔建置 DB2 範例程式，必須變更在這些 Script 檔中指定的併入檔路徑以指向 cobol_i 目錄而不是 cobol_a 目錄。

若不是使用 IBM COBOL Set for AIX 編譯器的「System/390 主電腦資料類型支援」特性，或使用此編譯器的舊版本，則應用程式的 DB2 併入檔會位於下列目錄：

```
$HOME/sqlllib/include/cobol_a
```

指定 COPY 檔名，來併入 .cbl 副檔名，方式如下：

```
COPY "sql.cbl".
```

相關概念:

- 第 143 頁的『在 AIX 上安裝 COBOL 的注意事項』

相關作業:

- 第 22 頁的『設置 UNIX 應用程式開發環境』
- 第 172 頁的『在 AIX 上建置 IBM COBOL 應用程式』
- 第 175 頁的『在 AIX 上建置 IBM COBOL 常式』

在 AIX 上建置 IBM COBOL 應用程式

DB2 會提供建置 Script，來編譯及鏈結 COBOL 內嵌式 SQL 及 DB2 API 程式。這些都位於 sqlllib/samples/cobol 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

建置檔 bldapp 含有要建置 DB2 應用程式的命令。

第一個參數 \$1，指定您的來源檔的名稱。對沒有內嵌式 SQL 的程式而言，這是唯一必要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 \$2，指定您要連接的資料庫名稱；第三個參數 \$3 指定資料庫的使用者 ID，而 \$4 指定通行碼。

對於內嵌式 SQL 程式，bldapp 會傳送參數以前置編譯及連結 Script embprep。如果未提供任何資料庫名稱，將使用預設 sample 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

若要從來源檔 client.cbl 建置非內嵌式 SQL 範例程式 client，請輸入：

```
bldapp client
```

結果產生可執行檔 client。您可以輸入下面的命令，對 Sample 資料庫執行可執行檔：

```
client
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 updat.sqb 建置內嵌式 SQL 應用程式 updat：

1. 如果連接同一案例上的 sample 資料庫，請輸入：

```
bldapp updat
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp updat database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp updat database userid password
```

結果產生是可執行檔 updat。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 sample 資料庫，則只要輸入可執行檔檔名：

```
updat
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
updat database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
updat database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關作業:

- 第 175 頁的『在 AIX 上建置 IBM COBOL 常式』

相關參照:

- 第 174 頁的『AIX IBM COBOL 應用程式編譯及鏈結選項』
- 第 76 頁的『COBOL 範例』

相關範例:

- 『bldapp -- Builds AIX COBOL applications』
- 『client.cbl -- How to set and query a client (IBM COBOL)』
- 『embprep -- To prep and bind a COBOL embedded SQL sample on AIX』
- 『updat.sqb -- How to update, delete and insert table data (IBM COBOL)』

IBM COBOL 應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldapp
# Builds AIX COBOL applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# If an embedded SQL program, precompile and bind it.
if [ -f "$1.sqb" ]
then
  ./embprep $1 $2 $3 $4
fi

# Compile the checkerr.cbl error checking utility.
cob2 -qpgmname\(\mixed\) -qlib -I$DB2PATH/include/cobol_a \
  -c checkerr.cbl

# Compile the program.
cob2 -qpgmname\(\mixed\) -qlib -I$DB2PATH/include/cobol_a \
  -c $1.cbl

# Link the program.
cob2 -o $1 $1.o checkerr.o -L$DB2PATH/lib -ldb2
```

AIX IBM COBOL 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 AIX IBM COBOL Set 編譯器建置 COBOL 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp 建置 Script 中所示範一般。

bldapp 的編譯及鏈結選項

編譯選項：

- cob2** IBM COBOL Set 編譯器
- qpgmname*(mixed\)***
指示編譯器允許 CALL 具有混合字體名稱的檔案庫進入點。
- qlib** 指示編譯器處理 COPY 陳述式。
- I\$DB2PATH/include/cobol_a**
指定 DB2 併入檔的位置。例如：`$HOME/sql1lib/include/cobol_a`。
- c** 僅執行編譯；沒有任何鏈結。編譯及鏈結是不同的步驟。

鏈結選項：

- cob2** 使用編譯器作為鏈結器的前端。
- o \$1** 指定可執行檔程式。
- \$1.o** 指定程式目的檔。
- checkerr.o**
包括公用程式目的檔以便檢查錯誤。
- L\$DB2PATH/lib**
指定 DB2 執行程式共用檔案庫的位置。例如：`$HOME/sql1lib/lib`。如果您未指定 **-L** 選項，則編譯器將採用下列路徑：`/usr/lib:/lib`。
- ldb2** 與資料庫管理程式檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 172 頁的『在 AIX 上建置 IBM COBOL 應用程式』

相關參照：

- 第 177 頁的『AIX IBM COBOL 常式編譯及鏈結選項』

相關範例：

- 『bldapp -- Builds AIX COBOL applications』

在 AIX 上建置 IBM COBOL 常式

DB2 會提供建置 Script，來編譯及鏈結 COBOL 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sql1lib/samples/cobol` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

位於 `sqllib/samples/cobol` 的 Script `bldrtn` 含有要建置常式 (儲存程序) 的命令。Script 會編譯常式，並將它放入可被從屬站應用程式呼叫的共用檔案庫。

第一個參數 `$1`，指定您的來源檔的名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。既然共用檔案庫必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

只有第一個參數是必要的，即來源檔名稱。Script 會使用來源檔名稱 `$1` 作為共用檔案庫名稱。資料庫名稱是可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

程序:

當連接到範例資料庫時，若要從來源檔 `outsrv.sqb` 建置範例程式 `outsrv`，請輸入：

```
bldrtn outsrv
```

如果連接的是另一個資料庫，亦請併入資料庫名稱：

```
bldrtn outsrv database
```

Script 檔會將共用檔案庫複製到伺服器的路徑 `sqllib/function` 中。

一旦建置了常式共用檔案庫 `outsrv`，您就可以建置從屬站應用程式 `outcli`，在檔案庫內呼叫常式。您可以使用 Script 檔 `bldapp` 建置 `outcli`。

若要呼叫常式，請輸入下列命令以執行範例從屬站應用程式：

```
outcli database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample` 或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式可存取共用檔案庫 `outsrv`，並在伺服器資料庫中執行名稱相同的常式，然後將輸出傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關作業:

- 第 172 頁的『在 AIX 上建置 IBM COBOL 應用程式』

相關參照:

- 第 177 頁的『AIX IBM COBOL 常式編譯及鏈結選項』
- 第 76 頁的『COBOL 範例』

相關範例:

- 『bldrtn -- Builds AIX COBOL routines (stored procedures)』
- 『embprep -- To prep and bind a COBOL embedded SQL sample on AIX』
- 『outcli.sqb -- Call stored procedures using the SQLDA structure (IBM COBOL)』
- 『outsrv.sqb -- Demonstrates stored procedures using the SQLDA structure (IBM COBOL)』

IBM COBOL 常式的建置 Script

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds AIX COBOL routines (stored procedures)
# Usage: bldrtn <program_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Precompile and bind the program.
./embprep $1 $2

# Compile the checkerr.cbl error checking utility.
cob2 -qpgmname\(mixed\) -qlib -I$DB2PATH/include/cobol_a \
    -c checkerr.cbl

# Compile the program.
cob2 -qpgmname\(mixed\) -qlib -c -I$DB2PATH/include/cobol_a $1.cbl

# Link the program creating shared library $1 with export file $1.exp
cob2 -o $1 $1.o checkerr.o -bnoentry -bE:$1.exp \
    -L$DB2PATH/lib -ldb2

# Copy the shared library to the sqllib/function directory of the DB2 instance.
# This assumes the user has write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

AIX IBM COBOL 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 IBM COBOL C++ 編譯器在 AIX 上建置 COBOL 常式 (儲存程序)，如同 bldrtn 建置 Script 中所示範一般。

bldrtn 的編譯及鏈結選項

編譯選項：

- cob2** IBM COBOL Set 編譯器
- qpgmname\ (mixed\)**
指示編譯器允許 CALL 具有混合字體名稱的檔案庫進入點。
- qlib** 指示編譯器處理 COPY 陳述式。
- c** 僅執行編譯；沒有任何鏈結。本書假定編譯及鏈結是分開的步驟。
- I\$DB2PATH/include/cobol_a**
指定 DB2 併入檔的位置。例如：\$HOME/sqllib/include/cobol_a。

鏈結選項：

- cob2** 使用編譯器來鏈結編輯。
- o \$1** 將輸出檔指定為共用檔案庫檔案。
- \$1.o** 指定儲存程序目的檔。
- checkerr.o**
包括公用程式目的檔以便檢查錯誤。
- bnoentry**
不要指定共用檔案庫的預設進入點。
- bE:\$1.exp**
指定匯出檔。匯出檔含有儲存程序的清單。
- L\$DB2PATH/lib**
指定 DB2 執行程式共用檔案庫的位置。例如：\$HOME/sqllib/lib。如果您未指定 -L 選項，則編譯器將採用下列路徑： /usr/lib:/lib。
- ldb2** 與資料庫管理程式檔案庫鏈結。
- 請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 175 頁的『在 AIX 上建置 IBM COBOL 常式』

相關參照：

- 第 174 頁的『AIX IBM COBOL 應用程式編譯及鏈結選項』

相關範例：

- 『bldrtn -- Builds AIX COBOL routines (stored procedures)』

Micro Focus COBOL

在 AIX 上架構 Micro Focus COBOL 編譯器

如果您想要利用 Micro Focus COBOL 編譯器開發含有內嵌式 SQL 及 DB2 API 呼叫的應用程式，請執行下列動作。

程序:

- 當您在使用命令行處理器命令 `db2 prep` 前置編譯您的應用程式時，請使用 `target mfcob` 選項。
- 您必須在 Micro Focus COBOL 環境變數 `COBCPY` 中，併入 DB2 COBOL COPY 檔案目錄。COBCPY 環境變數會指定 COPY 檔的位置。Micro Focus COBOL 的 DB2 COPY 檔常駐在資料庫案例目錄下的 `sqllib/include/cobol_mf` 中。

若要包括目錄，請輸入：

- 在 `bash` 或 `Korn shell` 輸入：

```
export COBCPY=$COBCPY:$HOME/sqllib/include/cobol_mf
```

- 在 `C shell` 輸入：

```
setenv COBCPY $COBCPY:$HOME/sqllib/include/cobol_mf
```

註：您可能想要在 `.profile` 或 `.login` 檔中設定 `COBCPY`。

相關概念:

- 第 143 頁的『在 AIX 上安裝 COBOL 的注意事項』

相關作業:

- 第 22 頁的『設置 UNIX 應用程式開發環境』
- 第 179 頁的『在 AIX 上建置 Micro Focus COBOL 應用程式』
- 第 183 頁的『在 AIX 上建置 Micro Focus COBOL 常式』

在 AIX 上建置 Micro Focus COBOL 應用程式

DB2 會提供建置 Script，來編譯及鏈結 Micro Focus COBOL 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/cobol_mf` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

建置檔 `bldap` 含有要建置 DB2 應用程式的命令。

第一個參數 \$1，指定您的來源檔的名稱。對沒有內嵌式 SQL 的程式而言，這是唯一必要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 \$2，指定您要連接的資料庫名稱；第三個參數 \$3 指定資料庫的使用者 ID，而 \$4 指定通行碼。

對於內嵌式 SQL 程式，bldapp 會傳送參數以前置編譯及連結 Script embprep。如果未提供任何資料庫名稱，將使用預設 sample 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

若要從來源檔 client.cbl 建置非內嵌式 SQL 範例程式 client，請輸入：

```
bldapp client
```

結果產生可執行檔 client。您可以輸入下面的命令，對 Sample 資料庫執行可執行檔：

```
client
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 updat.sqb 建置內嵌式 SQL 應用程式 updat：

1. 如果連接同一案例上的 sample 資料庫，請輸入：

```
bldapp updat
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp updat database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp updat database userid password
```

結果產生是可執行檔 updat。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 sample 資料庫，則只要輸入可執行檔檔名：

```
updat
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
updat database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
updat database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關作業:

- 第 183 頁的『在 AIX 上建置 Micro Focus COBOL 常式』

相關參照:

- 第 182 頁的『AIX Micro COBOL 應用程式編譯及鏈結選項』
- 第 76 頁的『COBOL 範例』

相關範例:

- 『bldapp -- Builds AIX Micro Focus COBOL applications』
- 『client.cbl -- How to set and query a client (MF COBOL)』
- 『updat.sqb -- How to update, delete and insert table data (MF COBOL)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』

Micro Focus COBOL 應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldapp
# Builds AIX Micro Focus COBOL applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
    embprep $1 $2 $3 $4
fi

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Compile the checkerr.cbl error checking utility.
cob -c -x checkerr.cbl

# Compile the program.
cob -c -x $1.cbl

# Link the program.
cob -x -o $1 $1.o checkerr.o -L$DB2PATH/lib -ldb2 -ldb2gmf
```

AIX Micro COBOL 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Micro Focus COBOL 編譯器在 AIX 上建置 COBOL 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp 建置 Script 中所示範一般。

bldapp 的編譯及鏈結選項	
編譯選項：	
cob	COBOL 編譯器。
-c	僅執行編譯；沒有任何鏈結。
-x	與 -c 一起使用時，將產生物件檔。
鏈結選項：	
cob	使用編譯器作為鏈結器的前端。
-x	產生可執行程式。
-o \$1	指定可執行檔程式。
\$1.o	指定程式目的檔。
-L\$DB2PATH/lib	指定 DB2 執行程式共用檔案庫的位置。例如：\$HOME/sql1lib/lib。如果您未指定 -L 選項，則編譯器將採用下列路徑： /usr/lib:/lib。
-ldb2	鏈結 DB2 檔案庫。
-ldb2gmf	鏈結 Micro Focus COBOL 的 DB2 例外處理程式檔案庫。
請參閱您的編譯器文件，取得其他編譯器選項。	

相關作業：

- 第 179 頁的『在 AIX 上建置 Micro Focus COBOL 應用程式』

相關參照：

- 第 185 頁的『AIX Micro COBOL 常式編譯及鏈結選項』

相關範例：

- 『bldapp -- Builds AIX Micro Focus COBOL applications』

在 AIX 上建置 Micro Focus COBOL 常式

DB2 會提供建置 Script，來編譯及鏈結 Micro Focus COBOL 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/cobol_mf` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script `bldrtn` 含有要建置常式 (儲存程序) 的命令。Script 會編譯常式來源檔，並將其放入可被從屬站應用程式呼叫的共用檔案庫。

第一個參數 `$1`，指定您的來源檔的名稱。Script 會使用來源檔名稱，作為共用檔案庫名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。既然共用檔案庫必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

只有第一個參數是必要的，即來源檔名稱。資料庫名稱是可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

程序:

當連接到範例資料庫時，若要從來源檔 `outsrv.sqb` 建置範例程式 `outsrv`，請輸入：

```
bldrtn outsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn outsrv database
```

Script 檔會將共用檔案庫複製到伺服器的路徑 `sqllib/function` 中。

一旦建置了儲存程序 `outsrv`，您就可以建置從屬站應用程式 `outcli`，呼叫該儲存程序。您可以使用 Script 檔 `bldapp` 建置 `outcli`。

若要呼叫儲存程序，請輸入下列命令來執行範例從屬站應用程式：

```
outcli database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample` 或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式可以存取共用檔案庫 `outsrv`，並在伺服器資料庫中執行名稱相同的儲存程序函數。然後將輸出傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關作業:

- 第 179 頁的『在 AIX 上建置 Micro Focus COBOL 應用程式』

相關參照:

- 第 185 頁的『AIX Micro COBOL 常式編譯及鏈結選項』
- 第 76 頁的『COBOL 範例』

相關範例:

- 『`bldrtn -- Builds AIX Micro Focus COBOL routines (stored procedures)`』
- 『`outcli.sqb -- Call stored procedures using the SQLDA structure (MF COBOL)`』
- 『`outsrv.sqb -- Demonstrates stored procedures using the SQLDA structure (MF COBOL)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

Micro Focus COBOL 常式的建置 Script

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds AIX Micro Focus COBOL routines (stored procedures)
# Usage: bldrtn <program_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# Precompile and bind the program.
./embprep $1 $2

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Compile the program.
cob -c -x $1.cbl

# Link the program.
cob -x -o $1 $1.o -Q -bnoentry \
-Q -bI:$DB2PATH/lib/db2g.imp -L$DB2PATH/lib -ldb2 -ldb2gmf
```

```
# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

AIX Micro COBOL 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Micro Focus COBOL 編譯器在 AIX 上建置 COBOL 常式 (儲存程序)，如同 bldrtn 建置 Script 中所示範一般。

bldrtn 的編譯及鏈結選項	
編譯選項：	
cob	COBOL 編譯器。
-c	僅執行編譯；沒有任何鏈結。本書假定編譯及鏈結是分開的步驟。
-x	與 -c 選項一起使用時，編譯至物件模組。
鏈結選項：	
cob	使用編譯器作為鏈結器的前端。
-x	產生共用檔案庫。
-o \$1	指定可執行檔程式。
\$1.o	指定程式目的檔。
-Q -bnoentry	不要指定共用檔案庫的預設進入點。
-Q -bI:\$DB2PATH/lib/db2g.imp	提供 DB2 應用程式檔案庫的進入點的清單。
-L\$DB2PATH/lib	指定 DB2 執行程式共用檔案庫的位置。例如：\$HOME/sqllib/lib。如果您未指定 -L 選項，則編譯器將採用下列路徑： /usr/lib:/lib。
-ldb2	鏈結 DB2 檔案庫。
-ldb2gmf	鏈結 Micro Focus COBOL 的 DB2 例外處理程式檔案庫。
請參閱您的編譯器文件，取得其他編譯器選項。	

相關作業:

- 第 183 頁的『在 AIX 上建置 Micro Focus COBOL 常式』

相關參照:

- 第 182 頁的『AIX Micro COBOL 應用程式編譯及鏈結選項』

相關範例:

- 『bldrtn -- Builds AIX Micro Focus COBOL routines (stored procedures)』

REXX

在 AIX 上建置 REXX 應用程式

以下內容說明如何在 AIX 上建置 REXX 應用程式。DB2 for AIX 支援典型的 REXX，也支援 Object REXX。Object REXX 是 REXX 語言的物件導向版本。物件導向功能已加入傳統的 REXX 中，但未改變其原有的函數及命令。Object REXX 直譯器已經過進一步改良，提供下列額外的支援：

- 類別、物件及方法
- 傳訊及多型性
- 單一及多重繼承

Object REXX 與傳統的 REXX 完全相容。本節中提及 REXX 時，表示所有 REXX 版本，包括 Object REXX。

您不必前置編譯或連結 REXX 程式。

程序:

欲在 AIX 上執行 DB2 REXX/SQL 程式，您必須設定 LIBPATH 環境變數，來包括 DB2 安裝目錄下的 lib。

若為 bash 或 Korn shell，請輸入：

```
export LIBPATH=$LIBPATH:/lib:/usr/lib:/usr/opt/db2_08_01/lib
```

若為 C shell，請輸入：

```
setenv LIBPATH $LIBPATH:/lib:/usr/lib:/usr/opt/db2_08_01/lib
```

在 AIX 上，您的應用程式檔案可以具有任何副檔名。您可以使用下列兩種方法之一，來執行您的應用程式：

1. 在 Shell 命令提示字元中，輸入 *rexx name*，其中 *name* 為您的 REXX 程式的名稱 (包括副檔名，若存在的話)。
2. 如果您的 REXX 程式的第一行含有 "魔術數字" (#!)，並識別 REXX/6000 直譯器常駐的目錄，則您可以經由在 Shell 命令提示字元中輸入您的 REXX 程式，來執行它。例如，如果 REXX/6000 直譯器檔案位在 /usr/bin 目錄中，併入下列作為您的 REXX 程式的第一行：

```
#!/usr/bin/rexx
```

然後，經由在 Shell 命令提示字元中輸入下列命令，使程式成爲可執行檔：

```
chmod +x name
```

經由在 Shell 命令提示字元中，輸入您的 REXX 程式的檔名來執行它。

REXX 範例程式位在目錄 `sql1lib/samples/rexx` 中。若要執行範例 REXX 程式 `updat.cmd`，請輸入：

```
updat.cmd
```

相關作業:

- 第 22 頁的『設置 UNIX 應用程式開發環境』

相關參照:

- 第 86 頁的『REXX 範例』

第 8 章 HP-UX

HP-UX C	189	在 HP-UX 上建置 C++ 多緒應用程式	211
在 HP-UX 上建置 C 應用程式	189	C++ 多緒應用程式的建置 Script	212
C 應用程式的建置 Script	191	Micro Focus COBOL	213
HP-UX C 應用程式編譯及鏈結選項	192	在 HP-UX 上架構 Micro Focus COBOL 編譯器	213
在 HP-UX 上建置 C 常式	193	在 HP-UX 上建置 Micro Focus COBOL 應用程式	214
C 常式的建置 Script	197	Micro Focus COBOL 應用程式的建置 Script	216
HP-UX C 常式編譯及鏈結選項	198	HP-UX Micro Focus COBOL 應用程式編 譯及鏈結選項	216
在 HP-UX 上建置 C 多緒應用程式	199	在 HP-UX 上建置 Micro Focus COBOL 常式	217
C 多緒應用程式的建置 Script	200	Micro Focus COBOL 常式的建置 Script	219
HP-UX C++.	201	HP-UX Micro Focus COBOL 常式編譯及 鏈結選項	220
在 HP-UX 上建置 C++ 應用程式	201		
C++ 應用程式的建置 Script	202		
HP-UX C++ 應用程式編譯及鏈結選項	203		
在 HP-UX 上建置 C++ 常式	205		
C++ 常式的建置 Script	209		
HP-UX C++ 常式編譯及鏈結選項	210		

本章提供在 HP-UX 上建置 DB2 應用程式的詳細資訊。關於適用於 HP-UX 的最新 DB2 應用程式開發更新組件，請探訪下列 DB2 應用程式開發網頁：

<http://www.ibm.com/software/data/db2/udb/ad>

HP-UX C

關於 DB2 CLI 應用程式及常式的建置資訊，請參閱 *CLI Guide and Reference*。

在 HP-UX 上建置 C 應用程式

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/c` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

建置 Script `bldapp` 含有要建置 DB2 應用程式的命令。

第一個參數 \$1，指定您的來源檔的名稱。這是唯一必要的參數，也是沒有內嵌式 SQL DB2 API 程式唯一需要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 \$2，指定您要連接的資料庫名稱；第三個參數 \$3 指定資料庫的使用者 ID，而 \$4 指定通行碼。

對於內嵌式 SQL 程式，bldapp 會傳送參數以前置編譯及連結 Script embprep。如果未提供任何資料庫名稱，將使用預設 sample 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

下列範例告訴您如何建置及執行 DB2 API 及內嵌式 SQL 應用程式。

若要從來源檔 cli_info.c 建置 DB2 API 非內嵌式 SQL 範例程式 cli_info，請輸入：

```
bldapp cli_info
```

結果是可執行檔 cli_info。

欲執行可執行檔，請輸入可執行檔檔名：

```
cli_info
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 tbmod.sqc 建置內嵌式 SQL 應用程式 tbmod：

1. 如果連接同一案例上的 sample 資料庫，請輸入：

```
bldapp tbmod
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp tbmod database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp tbmod database userid password
```

結果是可執行檔 tbmod。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 sample 資料庫，則只要輸入可執行檔檔名：

```
tbmod
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
tbmod database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
tbmod database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』
- 第 192 頁的『HP-UX C 應用程式編譯及鏈結選項』

相關範例:

- 『bldapp -- Builds HP-UX C applications』
- 『cli_info.c -- Set and get information at the client level (C)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『tbmod.sqc -- How to modify table data (C)』

C 應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldapp
# Builds HP-UX C applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1ib

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG="+DA2.0W"
else
    EXTRA_CFLAG="+DAportable"
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.c error-checking utility.
    cc $EXTRA_CFLAG -Ae -I$DB2PATH/include -c utilemb.c
else
    # Compile the utilapi.c error-checking utility.
    cc $EXTRA_CFLAG -Ae -I$DB2PATH/include -c utilapi.c
fi

# Compile the program.
cc $EXTRA_CFLAG -Ae -I$DB2PATH/include -c $1.c

if [ -f $1".sqc" ]
then
    # Link the program with utilemb.o.
    cc $EXTRA_CFLAG -o $1 $1.o utilemb.o -L$DB2PATH/lib -ldb2
```

```

else
  # Link the program with utilapi.o.
  cc $EXTRA_CFLAG -o $1 $1.o utilapi.o -L$DB2PATH/lib -ldb2
fi

```

HP-UX C 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 HP-UX C 編譯器建置 C 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp 建置 Script 中所示範一般。

bldapp 的編譯及鏈結選項	
編譯選項：	
cc	C 編譯器。
\$EXTRA_CFLAG	含有不同值，取決於啓用了 32 位元或 64 位元支援而定。若是 32 位元，值為 +DAportable；若是 64 位元，值為 +DA2.0W。
+DAportable (僅限 32 位元)	產生在 PA_RISC 1.x 及 2.0 工作站與伺服器中可相容的的程式碼。如果關心應用程式的可移轉性，就應該使用這個選項。沒有這個選項的建置將產生更適合於建置機器之處理器層次的程式碼，但是無法在舊 PA-RISC 處理器層次上運作。關於詳細資訊，請參閱編譯器文件。
+DA2.0W (僅限 64 位元)	必須用來產生 64 位元程式碼。
-Ae	啓用 HP ANSI 延伸模式。
-I\$DB2PATH/include	指定 DB2 併入檔的位置。
-c	僅執行編譯；沒有任何鏈結。編譯及鏈結是不同的步驟。

bldapp 的編譯及鏈結選項

鏈結選項：

cc 使用編譯器作為鏈結器的前端。

\$EXTRA_CFLAG

含有不同值，取決於啓用了 32 位元或 64 位元支援而定。若是 32 位元，值為 +DAportable；若是 64 位元，值為 +DA2.0W。

+DAportable (僅限 32 位元)

使用在 PA_RISC 1.x 及 2.0 工作站與伺服器中相容的的程式碼。如果關心應用程式的可移轉性，就應該使用這個選項。沒有這個選項的建置將產生更適合於建置機器之處理器層次的程式碼，但是無法在舊 PA-RISC 處理器層次上運作。關於詳細資訊，請參閱編譯器文件。

+DA2.0W (僅限 64 位元)

必須用來產生 64 位元程式碼。

-o \$1 指定可執行檔。

\$1.o 指定程式目的檔。

utilemb.o

如果是內嵌式 SQL 程式，則有內嵌式 SQL 公用程式的目的檔，可執行錯誤檢查。

utilapi.o

如果是非內嵌式 SQL 程式，則有 DB2 API 公用程式目的檔，可執行錯誤檢查。

-L\$DB2PATH/lib

指定 DB2 執行程式共用檔案庫的位置。例如：-L\$DB2PATH/lib。如果您未指定 -L 選項，將使用 /usr/lib:/lib。

-ldb2 與 DB2 檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 189 頁的『在 HP-UX 上建置 C 應用程式』

相關範例：

- 『bldapp -- Builds HP-UX C applications』

在 HP-UX 上建置 C 常式

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 sqllib/samples/c 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script 檔 `bldrtn` 含有要建置常式 (儲存程序及使用者定義的函數) 的命令。Script 檔會編譯常式，並將它放入可被資料庫管理程式載入及可被從屬站應用程式呼叫的共用檔案庫。

第一個參數 `$1`，指定您的來源檔的名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。

資料庫參數是選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。因此，既然儲存程序必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

程序:

下列範例告訴您如何利用下列建置常式共用檔案庫：

- 儲存程序
- 非內嵌式 SQL 使用者定義的函數 (UDF)
- 內嵌式 SQL 使用者定義的函數 (UDF)

儲存程序共用檔案庫

若要從來源檔 `spserver.sqc` 建置範例程式 `spserver`：

1. 如果連接到 `sample` 資料庫，請輸入建置檔名稱及程式名稱：

```
bldrtn spserver
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn spserver database
```

Script 檔會將共用檔案庫複製到路徑 `sqllib/function` 中的伺服器。

2. 接著，在伺服器上執行 `spcat` Script，將常式編目：

```
spcat
```

這個 Script 將連接到範例資料庫、將取消常式編目 (若之前已呼叫 `spdrop.db2` 將它們編目的話)，然後呼叫 `spcreate.db2` 來將它們編目，最後切斷資料庫連線。您也可以個別地呼叫 `spdrop.db2` 及 `spcreate.db2` Script。

3. 然後，停止資料庫後再重新啟動，以辨識新的共用檔案庫。

一旦建置共用檔案庫 `spserver` 完成，您即可建置存取共用檔案庫的從屬站應用程式 `spclient`。

您可以使用 Script 檔 `bldapp` 建置 `spclient`。

欲呼叫共用檔案庫中的儲存程序，請輸入下列命令以執行範例從屬站應用程式：

```
spclient database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為有效的通行碼。

從屬站應用程式可存取共用檔案庫 `spserver`，並在伺服器資料庫中執行許多儲存程序函數。輸出會傳回從屬站應用程式。

非內嵌式 SQL UDF 共用檔案庫

若要從來源檔 `udfsrv.c` 建置使用者定義的函數程式 `udfsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfsrv
```

Script 檔會將 UDF 複製到 `sqllib/function` 目錄。

一旦您建置了 `udfsrv`，您就可以建置呼叫它的從屬站應用程式 `udfcli`。已提供 DB2 CLI 及此程式的內嵌式 SQL 版本。您可以使用 Script 檔 `bldapp`，從來源檔 `udfcli.c` (位於 `sqllib/samples/cli`) 建置 DB2 CLI `udfcli` 從屬站程式。

您可以使用 Script 檔 `bldapp`，從來源檔 `udfcli.sqc` (位於 `sqllib/samples/c`) 建置內嵌式 SQL `udfcli` 從屬站程式。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfcli database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

爲有效的通行碼。

從屬站應用程式會存取共用檔案庫 `udfsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

內嵌式 SQL UDF 共用檔案庫

當連接到 `sample` 資料庫時，若要從來源檔 `udfemsrv.sql` 建置內嵌式 SQL 使用者定義的函數程式 `udfemsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfemsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn udfemsrv database
```

Script 檔會將 UDF 檔案庫複製到 `sqllib/function` 目錄。

一旦您建置了 `udfemsrv`，您就可以建置呼叫它的從屬站應用程式 `udfemcli`。您可以使用 Script 檔 `bldapp`，從來源檔 `udfemcli.sql` (位於 `sqllib/samples/c`) 建置 `udfemcli` 從屬站程式。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfemcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲有效的通行碼。

從屬站應用程式會存取共用檔案庫 `udfemsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』

- 第 198 頁的『HP-UX C 常式編譯及鏈結選項』

相關範例:

- 『bldrtn -- Builds HP-UX C routines (stored procedures and UDFs)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcat -- To catalog C stored procedures on UNIX』
- 『spcli.sqc -- Call various stored procedures (C)』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqc』
- 『spserver.sqc -- A variety of types of stored procedures (C)』
- 『udfcli.sqc -- Call a variety of types of user-defined functions (C)』
- 『udfemcli.sqc -- Call a variety of types of embedded SQL user-defined functions. (C)』
- 『udfemsrv.sqc -- Call a variety of types of embedded SQL user-defined functions. (C)』
- 『udfsrv.c -- Call a variety of types of user-defined functions (C)』

C 常式的建置 Script

```

#!/bin/sh
# SCRIPT: bldrtn
# Builds HP-UX C routines (stored procedures and UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG="+DA2.0W"
else
    EXTRA_CFLAG="+DAportable"
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2

```

```

fi

# Compile the program.
cc $EXTRA_CFLAG +u1 +z -Ae -I$DB2PATH/include \
  -D_POSIX_C_SOURCE=199506L -c $1.c

# Link the program to create a shared library
ld -b -o $1 $1.o -L$DB2PATH/lib -ldb2 -lpthread

# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

HP-UX C 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 HP-UX C 編譯器建置 C 常式 (儲存程序及使用者定義的函數)，如同 `bldrtn` 建置 Script 中所示範一般。

bldrtn 的編譯及鏈結選項	
編譯選項：	
cc	C 編譯器。
\$EXTRA_CFLAG	含有不同值，取決於啓用了 32 位元或 64 位元支援而定。若是 32 位元，值爲 +DAportable；若是 64 位元，值爲 +DA2.0W。
+DAportable (僅限 32 位元)	產生在 PA_RISC 1.x 及 2.0 工作站與伺服器中可相容的的程式碼。如果關心應用程式的可移轉性，就應該使用這個選項。沒有這個選項的建置將產生更適合於建置機器之處理器層次的程式碼，但是無法在舊 PA-RISC 處理器層次上運作。關於詳細資訊，請參閱編譯器文件。
+DA2.0W (僅限 64 位元)	必須用來產生 64 位元程式碼。
+u1	容許未調整的資料存取。僅在您使用未調整的資料時才使用。
+z	產生與位置無關的程式碼。
-Ae	啓用 HP ANSI 延伸模式。
-I\$DB2PATH/include	指定 DB2 併入檔的位置。例如： <code>-I\$DB2PATH/include</code> 。
-D_POSIX_C_SOURCE=199506L	需要 POSIX 執行緒檔案庫選項以確定已定義 <code>_REENTRANT</code> ，因爲常式可能在其他常式 (THREADSAFE) 執行所在的同一程序中執行，或是在引擎本身 (NOT FENCED) 中執行。
-c	僅執行編譯；沒有任何鏈結。本書假定編譯及鏈結是分開的步驟。

bldrtn 的編譯及鏈結選項

鏈結選項：

- ld** 使用鏈結器來鏈結。
- b** 建立共用檔案庫，而非正常的可執行檔。
- o \$1** 將輸出檔指定為共用檔案庫檔案。
- \$1.o** 指定程式目的檔。
- L\$DB2PATH/lib**
指定 DB2 執行程式共用檔案庫的位置。例如：\$HOME/sql1lib/lib。如果您未指定 -L 選項，將使用 /usr/lib:/lib。
- ldb2** 與 DB2 檔案庫鏈結。
- lpthread**
與 POSIX 執行緒檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 193 頁的『在 HP-UX 上建置 C 常式』

相關範例：

- 『bldrtn -- Builds HP-UX C routines (stored procedures and UDFs)』

在 HP-UX 上建置 C 多緒應用程式

HP-UX 提供 POSIX 緒檔案庫及 DCE 緒檔案庫。DB2 僅支援使用 POSIX 緒檔案庫的多緒應用程式。

HP-UX 上的多緒應用程式必須定義 `_REENTRANT`，才能執行編譯。HP-UX 文件建議以 `-D_POSIX_C_SOURCE=199506L` 進行編譯。如此也可確保已定義 `_REENTRANT`。應用程式也需要鏈結 `-lpthread`。

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sql1lib/samples/c` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script 檔 `bldmt` 含有要建置多緒應用程式的命令。它最多使用 4 個參數，以變數 `$1`、`$2`、`$3` 及 `$4` 代表。

第一個參數 `$1`，指定您的來源檔的名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。第三個參數 `$3`，指定資料庫的使用者 ID，而 `$4` 則指定通行碼。只需要第一個參數 (來源檔名稱)。資料庫名稱、使用者 ID 及通行碼均為可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

程序:

若要從來源檔 `dbthrrds.sqc` 建置範例程式 `dbthrrds`，請輸入：

```
bldmt dbthrrds
```

結果是可執行檔 `dbthrrds`。若要對 `sample` 資料庫執行可執行檔，請輸入這個可執行檔名稱：

```
dbthrrds
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』

相關範例:

- 『`bldmt -- Builds HP-UX C multi-threaded applications`』
- 『`dbthrrds.sqc -- How to use multiple context APIs on UNIX (C)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

C 多緒應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldmt
# Builds HP-UX C multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# To compile 64-bit programs, uncomment the following line.
# BUILD_64BIT=true

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG="+DA2.0W"
else
    EXTRA_CFLAG="+DAportable"
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
fi
```

```
# Compile the program.
cc $EXTRA_CFLAG -Ae -I$DB2PATH/include -D_POSIX_C_SOURCE=199506L -c $1.c

# Link the program
cc $EXTRA_CFLAG -o $1 $1.o -L$DB2PATH/lib -ldb2 -lpthread
```

HP-UX C++

在 HP-UX 上建置 C++ 應用程式

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/cpp` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

建置 Script `bldapp` 含有要建置 DB2 API 及內嵌式 SQL 應用程式的命令。Script 最多使用四個參數。

第一個參數 `$1`，指定您的來源檔的名稱。這是唯一必要的參數，也是沒有內嵌式 SQL DB2 API 程式唯一需要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 `$2`，指定您要連接的資料庫名稱；第三個參數 `$3` 指定資料庫的使用者 ID，而 `$4` 指定通行碼。

對於內嵌式 SQL 程式，`bldapp` 會傳送參數以前置編譯及鏈結 Script `embprep`。如果未提供任何資料庫名稱，將使用預設 `sample` 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

下列範例告訴您如何建置及執行 DB2 API 及內嵌式 SQL 應用程式。

若要從來源檔 `cli_info.C` 建置非內嵌式 SQL 範例程式 `cli_info`，請輸入：

```
bldapp cli_info
```

結果是可執行檔 `cli_info`。您可以輸入下面的命令，對 `Sample` 資料庫執行可執行檔：

```
cli_info
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 `tbmod.sqC` 建置內嵌式 SQL 應用程式 `tbmod`：

1. 如果連接同一案例上的 `sample` 資料庫，請輸入：

```
bldapp tbmod
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp tbmod database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp tbmod database userid password
```

結果是可執行檔 `tbmod`。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 `sample` 資料庫，則只要輸入可執行檔檔名：

```
tbmod
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
tbmod database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
tbmod database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』
- 第 203 頁的『HP-UX C++ 應用程式編譯及鏈結選項』

相關範例:

- 『`bldapp -- Builds HP-UX C++ applications`』
- 『`cli_info.C -- Set and get information at the client level (C++)`』
- 『`tbmod.sqC -- How to modify table data (C++)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

C++ 應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldapp
# Builds HP-UX C++ applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib
```

```

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG="+DA2.0W"
else
    EXTRA_CFLAG="+DAportable"
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqC" ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.C error-checking utility.
    aCC $EXTRA_CFLAG -ext -I$DB2PATH/include -c utilemb.C
else
    # Compile the utilapi.C error-checking utility.
    aCC $EXTRA_CFLAG -ext -I$DB2PATH/include -c utilapi.C
fi

# Compile the program.
aCC $EXTRA_CFLAG -ext -I$DB2PATH/include -c $1.C

if [ -f $1".sqC" ]
then
    # Link the program with utilemb.o.
    aCC $EXTRA_CFLAG -o $1 $1.o utilemb.o -L$DB2PATH/lib -ldb2
else
    # Link the program with utilapi.o.
    aCC $EXTRA_CFLAG -o $1 $1.o utilapi.o -L$DB2PATH/lib -ldb2
fi

```

HP-UX C++ 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 HP-UX C++ 編譯器建置 C++ 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp 建置 Script 中所示範一般。

bldapp 的編譯及鏈結選項

編譯選項：

aCC HP aC++ 編譯器。

\$EXTRA_CFLAG

含有不同值，取決於啓用了 32 位元或 64 位元支援而定。若是 32 位元，值為 +DAportable；若是 64 位元，值為 +DA2.0W。

+DAportable (僅限 32 位元)

產生在 PA_RISC 1.x 及 2.0 工作站與伺服器中可相容的的程式碼。如果關心應用程式的可移轉性，就應該使用這個選項。沒有這個選項的建置將產生更適合於建置機器之處理器層次的程式碼，但是無法在舊 PA-RISC 處理器層次上運作。關於詳細資訊，請參閱編譯器文件。

+DA2.0W (僅限 64 位元)

必須用來產生 64 位元程式碼。

-ext 容許不同 C++ 副檔名包含「很長的」支援。

-I\$DB2PATH/include

指定 DB2 併入檔的位置。例如：\$HOME/sql/lib/include

-c 僅執行編譯；沒有任何鏈結。本書假定編譯及鏈結是分開的步驟。

bldapp 的編譯及鏈結選項

鏈結選項：

aCC 使用 HP aC++ 編譯器作為鏈結器的前端。

\$EXTRA_CFLAG

含有不同值，取決於啓用了 32 位元或 64 位元支援而定。若是 32 位元，值為 +DAportable；若是 64 位元，值為 +DA2.0W。

+DAportable (僅限 32 位元)

使用在 PA_RISC 1.x 及 2.0 工作站與伺服器中相容的的程式碼。如果關心應用程式的可移轉性，就應該使用這個選項。沒有這個選項的建置將產生更適合於建置機器之處理器層次的程式碼，但是無法在舊 PA-RISC 處理器層次上運作。關於詳細資訊，請參閱編譯器文件。

+DA2.0W (僅限 64 位元)

必須用來產生 64 位元程式碼。

-o \$1 指定可執行檔。

\$1.o 指定程式目的檔。

utilemb.o

如果是內嵌式 SQL 程式，則有內嵌式 SQL 公用程式的目的檔，可執行錯誤檢查。

utilapi.o

如果是非內嵌式 SQL 程式，則有 DB2 API 公用程式目的檔，可執行錯誤檢查。

-L\$DB2PATH/lib

指定 DB2 執行程式共用檔案庫的位置。例如：\$HOME/sql1lib/lib。如果您未指定 -L 選項，將使用 /usr/lib:/lib。

-ldb2 與 DB2 檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 201 頁的『在 HP-UX 上建置 C++ 應用程式』

相關範例：

- 『bldapp -- Builds HP-UX C++ applications』

在 HP-UX 上建置 C++ 常式

DB2 會提供建置 Script，來編譯及鏈結 C++ 內嵌式 SQL 及 DB2 API 程式。這些都位於 sql1lib/samples/cpp 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script bldrtn 含有要建置常式 (儲存程序或使用者定義的函數) 的命令。Script 會編譯常式，並將它放入可被從屬站應用程式呼叫的共用檔案庫。

第一個參數 \$1，指定您的來源檔的名稱。第二個參數 \$2，指定您想要與其連接的資料庫的名稱。

資料庫參數是選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。因此，既然共用檔案庫必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

程序:

下列範例告訴您如何利用下列建置常式共用檔案庫：

- 儲存程序
- 非內嵌式 SQL 使用者定義的函數 (UDF)
- 內嵌式 SQL 使用者定義的函數 (UDF)

儲存程序共用檔案庫

若要從來源檔 `spserver.sql` 建置範例程式 `spserver`：

1. 如果連接到 `sample` 資料庫，請輸入建置檔名稱及程式名稱：

```
bldrtn spserver
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn spserver database
```

Script 會將儲存程序複製到路徑 `sqllib/function` 中的伺服器。

2. 接著，在伺服器上執行 `spcat` Script，將常式編目：

```
spcat
```

這個 Script 將連接到範例資料庫、將取消常式編目 (若之前已呼叫 `spdrop.db2` 將它們編目的話)，然後呼叫 `spcreate.db2` 來將它們編目，最後切斷資料庫連線。您也可以個別地呼叫 `spdrop.db2` 及 `spcreate.db2` Script。

3. 然後，停止資料庫後再重新啟動，以辨識新的共用檔案庫。

一旦建置共用檔案庫 `spserver` 完成，您即可建置存取共用檔案庫的從屬站應用程式 `spclient`。

您可以使用 Script 檔 `bldapp` 建置 `spclient`。

若要呼叫儲存程序，請輸入下列命令來執行範例從屬站應用程式：

```
spclient database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式可存取共用檔案庫 `spserver`，並在伺服器資料庫中執行許多儲存程序函數。輸出會傳回從屬站應用程式。

非內嵌式 SQL UDF 共用檔案庫

若要從來源檔 `udfsrv.C` 建置使用者定義的函數程式 `udfsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfsrv
```

Script 檔會將 UDF 複製到 `sqllib/function` 目錄。

一旦您建置了 `udfsrv`，您就可以建置呼叫它的從屬站應用程式 `udfcli`。您可以使用 Script `bldapp` 從來源檔 `udfcli.sqc` 建置 `udfcli`。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

內嵌式 SQL UDF 共用檔案庫

當連接到 `sample` 資料庫時，若要從來源檔 `udfemsrv.sqC` 建置內嵌式 SQL 使用者定義的函數程式 `udfemsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfemsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn udfemsrv database
```

Script 檔會將 UDF 複製到 `sqllib/function` 目錄。

一旦您建置了 `udfemsrv`，您就可以建置呼叫它的從屬站應用程式 `udfemcli`。您可以使用 Script `bldapp` 從來源檔 `udfemcli.sqC` 建置 `udfemcli`。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfemcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfemsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』
- 第 210 頁的『HP-UX C++ 常式編譯及鏈結選項』

相關範例:

- 『`bldrtn -- Builds HP-UX C++ routines (stored procedures and UDFs)`』
- 『`spclient.sqC -- Call various stored procedures (C++)`』
- 『`spserver.sqC -- A variety of types of stored procedures (C++)`』
- 『`udfcli.sqC -- Call a variety of types of user-defined functions (C++)`』

- 『udfemcli.sqlC -- Call a variety of types of embedded SQL user-defined functions. (C++)』
- 『udfemsrv.sqlC -- Call a variety of types of embedded SQL user-defined functions. (C++)』
- 『udfsrv.C -- Call a variety of types of user-defined functions (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcat -- To catalog C stored procedures on UNIX』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sql』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sql』

C++ 常式的建置 Script

```

#!/bin/sh
# SCRIPT: bldrtn
# Builds HP-UX C++ routines (stored procedures and UDFs)
# Usage: bldsrv <prog_name> [ <db_name> ]

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1ib

if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG="+DA2.0W"
else
    EXTRA_CFLAG="+DAportable"
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqlC" ]
then
    ./embprep $1 $2
fi

# Compile the program. First ensure it is coded with extern "C".
aCC $EXTRA_CFLAG +u1 +z -ext -mt -I$DB2PATH/include -c $1.C

# Link the program to create a shared library.
aCC $EXTRA_CFLAG -mt -b -o $1 $1.o -L$DB2PATH/lib -ldb2

```

```
# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

HP-UX C++ 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 HP-UX C++ 編譯器建置 C++ 常式 (儲存程序及使用者定義的函數)，如同 bldrtn 建置 Script 中所示範一般。

bldrtn 的編譯及鏈結選項	
編譯選項：	
aCC	HP aC++ 編譯器。
\$EXTRA_CFLAG	含有不同值，取決於啓用了 32 位元或 64 位元支援而定。若是 32 位元，值為 +DAportable；若是 64 位元，值為 +DA2.0W。
+DAportable (僅限 32 位元)	產生在 PA_RISC 1.x 及 2.0 工作站與伺服器中可相容的的程式碼。如果關心應用程式的可移轉性，就應該使用這個選項。沒有這個選項的建置將產生更適合於建置機器之處理器層次的程式碼，但是無法在舊 PA-RISC 處理器層次上運作。關於詳細資訊，請參閱編譯器文件。
+DA2.0W (僅限 64 位元)	必須用來產生 64 位元程式碼。
+u1	容許未調整的資料存取。
+z	產生與位置無關的程式碼。
-ext	容許不同 C++ 副檔名包含「很長的」支援。
-mt	需要容許 HP aC++ 編譯器的執行緒支援，因為常式可能在其他常式 (THREADSAFE) 執行所在的同一程序中執行，或是在引擎本身 (NOT FENCED) 中執行。
-I\$DB2PATH/include	指定 DB2 併入檔的位置。例如：\$DB2PATH/include
-c	僅執行編譯；沒有任何鏈結。本書假定編譯及鏈結是分開的步驟。

bldrtn 的編譯及鏈結選項

鏈結選項：

aCC 使用 HP aC++ 編譯器作為鏈結器的前端。

\$EXTRA_CFLAG

含有不同值，取決於啓用了 32 位元或 64 位元支援而定。若是 32 位元，值為 +DAportable；若是 64 位元，值為 +DA2.0W。

+DAportable (僅限 32 位元)

使用在 PA_RISC 1.x 及 2.0 工作站與伺服器中相容的程式碼。如果關心應用程式的可移轉性，就應該使用這個選項。沒有這個選項的建置將產生更適合於建置機器之處理器層次的程式碼，但是無法在舊 PA-RISC 處理器層次上運作。關於詳細資訊，請參閱編譯器文件。

+DA2.0W (僅限 64 位元)

必須用來產生 64 位元程式碼。

-mt 需要容許 HP aC++ 編譯器的執行緒支援，因為常式可能在其他常式 (THREADSAFE) 執行所在的同一程序中執行，或是在引擎本身 (NOT FENCED) 中執行。

-b 建立共用檔案庫，而非正常的可執行檔。

-o \$1 指定可執行檔。

\$1.o 指定程式目的檔。

-L\$DB2PATH/lib

指定 DB2 執行程式共用檔案庫的位置。例如：**-L\$DB2PATH/lib**。如果您未指定 **-L** 選項，將使用 **/usr/lib:/lib**。

-ldb2 與 DB2 檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 205 頁的『在 HP-UX 上建置 C++ 常式』

相關範例：

- 『bldrtn -- Builds HP-UX C++ routines (stored procedures and UDFs)』

在 HP-UX 上建置 C++ 多緒應用程式

HP-UX 提供 POSIX 緒檔案庫及 DCE 緒檔案庫。在 HP-UX 上 DB2 僅支援使用 POSIX 緒檔案庫的多緒應用程式。

對於 HP-UX C++ 編譯器，在編譯及鏈結步驟中，**-mt** 必須用於多緒應用程式。

DB2 會提供建置 Script，來編譯及鏈結 C++ 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/cpp` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script `bldmt` 含有要建置多緒應用程式的命令。它最多使用 4 個參數，以變數 `$1`、`$2`、`$3` 及 `$4` 代表。

第一個參數 `$1`，指定您的來源檔的名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。第三個參數 `$3`，指定資料庫的使用者 ID，而 `$4` 則指定通行碼。只需要第一個參數 (來源檔名稱)。資料庫名稱、使用者 ID 及通行碼均為可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

程序:

若要從來源檔 `dbthrds.sqlC` 建置範例程式 `dbthrds`，請輸入：

```
bldmt dbthrds
```

結果是可執行檔 `dbthrds`。若要對 `sample` 資料庫執行可執行檔，請輸入這個可執行檔名稱：

```
dbthrds
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』

相關範例:

- 『`bldmt -- Builds HP-UX C++ multi-threaded applications`』
- 『`dbthrds.sqlC -- How to use multiple context APIs on UNIX (C++)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

C++ 多緒應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldmt
# Builds HP-UX C++ multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib
```



```

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true
if [ "$BUILD_64BIT" != "" ]
then
    EXTRA_CFLAG="+DA2.0W"
else
    EXTRA_CFLAG="+DAportable"
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqC" ]
then
    ./embprep $1 $2 $3 $4
fi

# Compile the program.
aCC $EXTRA_CFLAG -ext -mt -I$DB2PATH/include -c $1.C

# Link the program
aCC $EXTRA_CFLAG -mt -o $1 $1.o -L$DB2PATH/lib -ldb2

```

Micro Focus COBOL

在 HP-UX 上架構 Micro Focus COBOL 編譯器

若開發含有內嵌式 SQL 和 DB2 API 呼叫的應用程式，而且使用 Micro Focus COBOL 編譯器，請記住以下幾點：

程序：

- 當您在使用命令行處理器命令 `db2 prep` 前置編譯您的應用程式時，請使用 `target mfcob` 選項。
- 若要使用內建式前置編譯器前端、執行時直譯器或 Animator 除錯器，您必須執行 Micro Focus 提供的 `mrts` 命令，新增 DB2 Generic API 進入點到 Micro Focus 執行時模組 `rts32`。您也必須執行 `mkcheck`，更新檢查檔。如果沒有執行，就會收到 `SQLGSTRT` 的 173 錯誤。

您必須執行 `mrts` 及 `mkcheck`，方可在建置 COBOL 應用程式前，設定您的環境。在執行 `mrts` 及 `mkcheck` 前，必須以下列步驟設定 `COBOPT`：

1. 以 `root` 身份登入。
2. 在目錄 `$COBDIR/src/rts` 中輸入：

```

COBOPT=/opt/IBM/db2/V8.1/lib/db2mrts.args; export COBOPT
ksh mrts
mv $COBDIR/rts32 $COBDIR/rts32.orig
cp rts32 $COBDIR/rts32

```

其中 `$COBDIR` 是安裝 COBOL 編譯器的目錄，通常為 `/opt/cobol/cobdir`。

3. 您也必須重建 check 可執行檔，Hewlett-Packard 將該檔案隨附在本產品中。如果沒有重建位在 \$COBDIR 目錄中的 check 可執行檔，則使用 cob -C SQL 執行的編譯嘗試會失效，並附有執行時系統 173 錯誤，因為 DB2 前置處理器會呼叫 DB2 檔案庫。欲重建 check，您應將 \$COBDIR 目錄下的 src/sql 目錄變更為根目錄，並執行 mkcheck Script。一旦 Script 完成，您就必須將產生的 check 可執行檔移到 \$COBDIR 目錄。在目錄 \$COBDIR/src/sql 中輸入：

```
COBOPT=/opt/IBM/db2/V8.1/lib/db2mkrts.args; export COBOPT
ksh mkcheck
mv $COBDIR/check $COBDIR/check.orig
cp check $COBDIR/check
```

現在，您可以使用下列目錄中提供的引數來執行 mkrts：

```
/opt/IBM/db2/V8.1/lib/db2mkrts.args
```

- 您必須在 Micro Focus COBOL 環境變數 COBCPY 中，併入 DB2 COBOL COPY 檔案目錄。COBCPY 環境變數會指定 COPY 檔的位置。Micro Focus COBOL 的 DB2 COPY 檔常駐在資料庫案例目錄下的 sqllib/include/cobol_mf 中。

若要包括目錄：

- 在 bash 或 Korn shell 上輸入：

```
export COBCPY=$COBCPY:$HOME/sqllib/include/cobol_mf
```

- 在 C shell 上輸入：

```
setenv COBCPY ${COBCPY}:${HOME}/sqllib/include/cobol_mf
```

註：您可能想要在 .profile 或 .login 檔中設定 COBCPY。

相關作業：

- 第 217 頁的『在 HP-UX 上建置 Micro Focus COBOL 常式』
- 第 214 頁的『在 HP-UX 上建置 Micro Focus COBOL 應用程式』

在 HP-UX 上建置 Micro Focus COBOL 應用程式

DB2 會提供建置 Script，來編譯及鏈結 Micro Focus COBOL 內嵌式 SQL 及 DB2 API 程式。這些都位於 sqllib/samples/cobol_mf 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

建置 Script bldapp 含有要建置 DB2 應用程式的命令。

第一個參數 \$1，指定您的來源檔的名稱。對沒有內嵌式 SQL 的程式而言，這是唯一必要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 \$2，指定您要連接的資料庫名稱；第三個參數 \$3 指定資料庫的使用者 ID，而 \$4 指定通行碼。

對於內嵌式 SQL 程式，bldapp 會傳送參數以前置編譯並連結檔案 embprep。如果未提供任何資料庫名稱，將使用預設 sample 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

若要從來源檔 client.cbl 建置非內嵌式 SQL 範例程式 client，請輸入：

```
bldapp client
```

結果產生可執行檔 client。您可以輸入下面的命令，對 Sample 資料庫執行可執行檔：

```
client
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 updat.sqb 建置內嵌式 SQL 應用程式 updat：

1. 如果連接同一案例上的 sample 資料庫，請輸入：

```
bldapp updat
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp updat database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp updat database userid password
```

結果產生可執行檔 updat。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 sample 資料庫，則只要輸入可執行檔檔名：

```
updat
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
updat database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
updat database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 216 頁的『HP-UX Micro Focus COBOL 應用程式編譯及鏈結選項』
- 第 76 頁的『COBOL 範例』

相關範例:

- 『bldapp -- Builds HP-UX Micro Focus COBOL applications』
- 『client.cbl -- How to set and query a client (MF COBOL)』
- 『updat.sqb -- How to update, delete and insert table data (MF COBOL)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』

Micro Focus COBOL 應用程式的建置 Script

```

#!/bin/sh
# SCRIPT: bldapp
# Builds HP-UX Micro Focus COBOL applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# If an embedded SQL program, precompile and bind it.
if [ -f "$1".sqb ]
then
    embprep $1 $2 $3 $4
fi

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Compile the checkerr.cbl error checking utility.
cob +DAportable -cx checkerr.cbl

# Compile the program.
cob +DAportable -cx $1.cbl

# Link the program.
cob +DAportable -x $1.o checkerr.o -L$DB2PATH/lib -ldb2 -ldb2gmf

```

HP-UX Micro Focus COBOL 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Micro Focus COBOL 編譯器在 HP-UX 上建置 COBOL 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp 建置 Script 中所示範一般。

bldapp 的編譯及鏈結選項

編譯選項：

cob Micro Focus COBOL 編譯器。

+DAportable

產生在 PA_RISC 1.x 及 2.0 工作站與伺服器中可相容的的程式碼。如果關心應用程式的可移轉性，就應該使用這個選項。沒有這個選項的建置將產生更適合於建置機器之處理器層次的程式碼，但是無法在舊 PA-RISC 處理器層次上運作。關於詳細資訊，請參閱編譯器文件。

-cx 編譯至物件模組。

鏈結選項：

cob 使用編譯器作為鏈結器的前端。

+DAportable

使用在 PA_RISC 1.x 及 2.0 工作站與伺服器中相容的的程式碼。如果關心應用程式的可移轉性，就應該使用這個選項。沒有這個選項的建置將產生更適合於建置機器之處理器層次的程式碼，但是無法在舊 PA-RISC 處理器層次上運作。關於詳細資訊，請參閱編譯器文件。

-x 指定可執行程式。

\$1.o 包括程式目的檔。

checkerr.o

包括公用程式目的檔以便檢查錯誤。

-L\$DB2PATH/lib

指定 DB2 執行程式共用檔案庫的位置。

-ldb2 鏈結 DB2 檔案庫。

-ldb2gmf

鏈結 Micro Focus COBOL 的 DB2 例外處理程式檔案庫。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 214 頁的『在 HP-UX 上建置 Micro Focus COBOL 應用程式』

相關範例：

- 『bldapp -- Builds HP-UX Micro Focus COBOL applications』

在 HP-UX 上建置 Micro Focus COBOL 常式

DB2 會提供建置 Script，來編譯及鏈結 Micro Focus COBOL 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/cobol_mf` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script `bldrtn` 含有要建置常式 (儲存程序) 的命令。Script 會編譯常式，並將它放入伺服器上可被從屬站應用程式呼叫的共用檔案庫。它會使用 1 或 2 個參數，在 `bldrtn` 內以變數 `$1` 及 `$2` 代表。

第一個參數 `$1`，指定您的來源檔的名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。既然共用檔案庫必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

只有第一個參數是必要的，即來源檔名稱。資料庫名稱是可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。Script 會使用來源檔名稱 `$1` 作為共用檔案庫名稱。

程序:

當連接到範例資料庫時，若要從來源檔 `outsrv.sqb` 建置範例程式 `outsrv`，請輸入：

```
bldrtn outsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn outsrv database
```

Script 會將共用檔案庫複製到 `sqllib/function` 目錄。

一旦建置了共用檔案庫 `outsrv`，您就可以建置從屬站應用程式 `outcli`，呼叫其中的常式。您可以使用 Script 檔 `bldapp` 建置 `outcli`。

若要呼叫常式，請輸入下列命令以執行從屬站應用程式：

```
outcli database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample` 或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式將存取共用檔案庫 `outsrv`，以在伺服器資料庫中執行名稱相同的常式，然後將輸出傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 220 頁的『HP-UX Micro Focus COBOL 常式編譯及鏈結選項』
- 第 76 頁的『COBOL 範例』

相關範例:

- 『bldrtn -- Builds HP-UX Micro Focus COBOL routines (stored procedures)』
- 『outcli.sqb -- Call stored procedures using the SQLDA structure (MF COBOL)』
- 『outsrv.sqb -- Demonstrates stored procedures using the SQLDA structure (MF COBOL)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』

Micro Focus COBOL 常式的建置 Script

```

#!/bin/sh
# SCRIPT: bldrtn
# Builds HP-UX Micro Focus COBOL routines (stored procedures)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# If an embedded SQL program, precompile and bind it.
if [ -f "$1.sqb" ]
then
    embprep $1 $2
fi

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Compile the program.
cob +DAportable +z -cx $1.cbl

# Link the program.
ld -b -o $1 $1.o -L$DB2PATH/lib -ldb2 -ldb2gmf \
    -L$COBDIR/coblib -lcobol -lcrtn

# Copy the shared library to the sqllib/function subdirectory.
# The user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

HP-UX Micro Focus COBOL 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Micro Focus COBOL 編譯器在 HP-UX 上建置 COBOL 常式 (儲存程序)，如同 bldrtn 建置 Script 中所示範一般。

bldrtn 的編譯及鏈結選項	
編譯選項：	
cob	COBOL 編譯器。
+DAportable	產生在 PA_RISC 1.x 及 2.0 工作站與伺服器中可相容的程式碼。如果關心應用程式的可移轉性，就應該使用這個選項。沒有這個選項的建置將產生更適合於建置機器之處理器層次的程式碼，但是無法在舊 PA-RISC 處理器層次上運作。關於詳細資訊，請參閱編譯器文件。
+z	產生與位置無關的程式碼。
-cx	編譯至物件模組。
鏈結選項：	
ld	使用鏈結器來鏈結。
-b	建立共用檔案庫，而非正常的可執行檔。
-o \$1	指定可執行檔。
\$1.o	包括程式目的檔。
-L\$DB2PATH/lib	指定 DB2 執行程式共用檔案庫的位置。
-ldb2	鏈結 DB2 共用檔案庫。
-ldb2gmf	鏈結 Micro Focus COBOL 的 DB2 例外處理程式檔案庫。
-L\$COBDIR/coblib	指定 COBOL 執行期檔案庫的位置。
-lcobol	鏈結 COBOL 檔案庫。
-lcrtn	鏈結 crtn 檔案庫。
請參閱您的編譯器文件，取得其他編譯器選項。	

相關作業：

- 第 217 頁的『在 HP-UX 上建置 Micro Focus COBOL 常式』

相關範例：

- 『bldrtn -- Builds HP-UX Micro Focus COBOL routines (stored procedures)』

第 9 章 Linux

Linux C	221	Linux C++	232
在 Linux 上建置 C 應用程式	221	在 Linux 上建置 C++ 應用程式	232
C 應用程式的建置 Script	223	C++ 應用程式的建置 Script.	233
Linux C 應用程式編譯及鏈結選項	224	Linux C++ 應用程式編譯及鏈結選項.	234
在 Linux 上建置 C 常式	225	在 Linux 上建置 C++ 常式.	235
C 常式的建置 Script	228	C++ 常式的建置 Script	239
Linux C 常式編譯及鏈結選項	229	Linux C++ 常式編譯及鏈結選項	239
在 Linux 上建置 C 多緒應用程式	230	在 Linux 上建置 C++ 多緒應用程式.	241
C 多緒應用程式的建置 Script	231	C++ 多緒應用程式的建置 Script	241

本章提供在 Linux 上建置應用程式的詳細資訊。關於適用於 Linux 的最新 DB2 應用程式開發更新組件，請探訪下列網頁：

<http://www.ibm.com/software/data/db2/udb/ad>

Linux C

關於 DB2 CLI 應用程式及常式的建置資訊，請參閱 *CLI Guide and Reference*。

在 Linux 上建置 C 應用程式

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/c` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

建置檔 `bldapp` 含有要建置 DB2 應用程式的命令。

第一個參數 \$1，指定您的來源檔的名稱。這是唯一必要的參數，也是沒有內嵌式 SQL DB2 API 程式唯一需要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 \$2，指定您要連接的資料庫名稱；第三個參數 \$3 指定資料庫的使用者 ID，而 \$4 指定通行碼。

對於內嵌式 SQL 程式，`bldapp` 會傳送參數以前置編譯及連結 Script `embprep`。如果未提供任何資料庫名稱，將使用預設 `sample` 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

下列範例告訴您如何建置及執行 DB2 API 及內嵌式 SQL 應用程式。

若要從來源檔 `cli_info.c` 建置 DB2 API 非內嵌式 SQL 範例程式 `cli_info`，請輸入：

```
bldapp cli_info
```

結果是可執行檔 `cli_info`。

欲執行可執行檔，請輸入可執行檔檔名：

```
cli_info
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 `tbmod.sqc` 建置內嵌式 SQL 應用程式 `tbmod`：

1. 如果連接同一案例上的 `sample` 資料庫，請輸入：

```
bldapp tbmod
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp tbmod database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp tbmod database userid password
```

結果是可執行檔 `tbmod`。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 `sample` 資料庫，則只要輸入可執行檔檔名：

```
tbmod
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
tbmod database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
tbmod database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』
- 第 224 頁的『Linux C 應用程式編譯及鏈結選項』

相關範例:

- 『`bldapp -- Builds Linux C applications`』
- 『`cli_info.c -- Set and get information at the client level (C)`』

- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『tbmod.sqc -- How to modify table data (C)』

C 應用程式的建置 Script

```

#!/bin/sh
# SCRIPT: bldapp
# Builds Linux C applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,-rpath$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.c error-checking utility.
    gcc -I$DB2PATH/include -c utilemb.c
else
    # Compile the utilapi.c error-checking utility.
    gcc -I$DB2PATH/include -c utilapi.c
fi

# Compile the program.
gcc -I$DB2PATH/include -c $1.c

if [ -f $1".sqc" ]
then
    # Link the program with utilemb.o.
    gcc -o $1 $1.o utilemb.o $EXTRA_LFLAG \
        -L$DB2PATH/lib -ldb2
else
    # Link the program with utilapi.o.
    gcc -o $1 $1.o utilapi.o $EXTRA_LFLAG \
        -L$DB2PATH/lib -ldb2
fi

```

Linux C 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Linux C 編譯器建置 C 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp 建置 Script 中所示範一般。

bldapp 的編譯及鏈結選項	
編譯選項：	
gcc	GNU/Linux C 編譯器。
-I\$DB2PATH/include	指定 DB2 併入檔的位置。
-c	僅執行編譯；沒有任何鏈結。此 Script 檔有不同的編譯及鏈結步驟。
鏈結選項：	
gcc	使用編譯器作為鏈結器的前端。
-o \$1	指定可執行檔。
\$1.o	指定目的檔。
utilemb.o	如果是內嵌式 SQL 程式，則有內嵌式 SQL 公用程式的目的檔，可執行錯誤檢查。
utilapi.o	如果是非內嵌式 SQL 程式，則有 DB2 API 公用程式物件檔，可執行錯誤檢查。
\$EXTRA_LFLAG	如果 'RUNTIME=true' 未加上註解，將含有值 "-Wl,-rpath,\$DB2PATH/lib"；不然，沒有任何值。
-Wl,-rpath,\$DB2PATH/lib	指定在執行時 DB2 共用檔案庫的位置。例如：\$HOME/sql1lib/lib。
-L\$DB2PATH/lib	指定在鏈結時 DB2 共用檔案庫的位置。例如：\$HOME/sql1lib/lib。如果您未指定 -L 選項，將使用 /usr/lib:/lib。
-ldb2	與 DB2 檔案庫鏈結。
請參閱您的編譯器文件，取得其他編譯器選項。	

相關作業：

- 第 221 頁的『在 Linux 上建置 C 應用程式』

相關範例：

- 『bldapp -- Builds Linux C applications』

在 Linux 上建置 C 常式

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/c` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script `bldrtn` 含有要建置常式 (儲存程序或使用者定義的函數) 的命令。Script 檔會編譯常式，並將它放入可被從屬站應用程式呼叫的共用檔案庫。

第一個參數 \$1，指定您的來源檔的名稱。第二個參數 \$2，指定您想要與其連接的資料庫的名稱。

資料庫參數是選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。因此，既然儲存程序必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

程序:

下列範例告訴您如何利用下列建置常式共用檔案庫：

- 儲存程序
- 非內嵌式 SQL 使用者定義的函數 (UDF)
- 內嵌式 SQL 使用者定義的函數 (UDF)

儲存程序共用檔案庫

若要從來源檔 `spserver.sqc` 建置範例程式 `spserver`：

1. 如果連接到 `sample` 資料庫，請輸入建置 Script 名稱及程式名稱：

```
bldrtn spserver
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn spserver database
```

Script 會將儲存程序複製到路徑 `sqllib/function` 中的伺服器。

2. 接著，在伺服器上執行 `spcat` Script，將常式編目：

```
spcat
```

這個 Script 將連接到範例資料庫、將取消常式編目 (若之前已呼叫 `spdrop.db2` 將它們編目的話)，然後呼叫 `spcreate.db2` 來將它們編目，最後切斷資料庫連線。您也可以個別地呼叫 `spdrop.db2` 及 `spcreate.db2` Script。

3. 然後，停止資料庫後再重新啟動，以辨識新的共用檔案庫。

一旦建置共用檔案庫 `spserver` 完成，您即可建置存取共用檔案庫的從屬站應用程式 `spclient`。

您可以使用 Script `bldapp` 建置 `spclient`。

若要呼叫儲存程序，請輸入下列命令來執行範例從屬站應用程式：

```
spclient database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式可存取共用檔案庫 `spserver`，並在伺服器資料庫中執行許多儲存程序函數。輸出會傳回從屬站應用程式。

非內嵌式 SQL UDF 共用檔案庫

若要從來源檔 `udfsrv.c` 建置使用者定義的函數程式 `udfsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfsrv
```

Script 檔會將 UDF 複製到 `sqllib/function` 目錄。

必要時，請設定 UDF 的檔案模式，以便資料庫管理程式可以存取它。

一旦您建置了 `udfsrv`，您就可以建置呼叫它的從屬站應用程式 `udfcli`。您可以使用 Script 檔 `bldapp` 從來源檔 `udfcli.sqc` 建置 `udfcli`。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfcli database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

內嵌式 SQL UDF 共用檔案庫

當連接到 `sample` 資料庫時，若要從來源檔 `udfemsrv.sqlc` 建置內嵌式 SQL 使用者定義的函數程式 `udfemsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfemsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn udfemsrv database
```

Script 檔會將 UDF 檔案庫複製到 `sqllib/function` 目錄。

必要時，請設定 UDF 的檔案模式，以便從屬站應用程式資料庫管理程式可以存取它。

一旦您建置了 `udfemsrv`，您就可以建置呼叫它的從屬站應用程式 `udfemcli`。您可以使用 Script 檔 `bldapp`，從來源檔 `udfemcli.sqlc` (位於 `sqllib/samples/c`) 建置 `udfemcli` 從屬站程式。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfemcli database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為有效的通行碼。

從屬站應用程式會存取共用檔案庫 `udfemsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』
- 第 229 頁的『Linux C 常式編譯及鏈結選項』

相關範例:

- 『bldrtn -- Builds Linux C routines (stored procedures or UDFs)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcat -- To catalog C stored procedures on UNIX』
- 『spclient.sqc -- Call various stored procedures (C)』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqc』
- 『spserver.sqc -- A variety of types of stored procedures (C)』
- 『udfcli.sqc -- Call a variety of types of user-defined functions (C)』
- 『udfemcli.sqc -- Call a variety of types of embedded SQL user-defined functions. (C)』
- 『udfemsrv.sqc -- Call a variety of types of embedded SQL user-defined functions. (C)』
- 『udfsrv.c -- Call a variety of types of user-defined functions (C)』

C 常式的建置 Script

```

#!/bin/sh
# SCRIPT: bldrtn
# Builds Linux C routines (stored procedures or UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# Set the runtime path since routines run as setuid.
EXTRA_LFLAG="-Wl,-rpath$DB2PATH/lib"

# If an embedded SQL program, precompile and bind it.
if [ -f "$1.sqc" ]
then
    ./embprep $1 $2
fi

```



```

# Compile the program.
gcc -fpic -I$DB2PATH/include -c $1.c -D_REENTRANT

# Link the program and create a shared library
gcc -shared -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2 -lpthread

# Copy the shared library to the function subdirectory.
# The user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Linux C 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Linux C 編譯器建置 C 常式 (儲存程序及使用者定義的函數)，如同 bldrtn 建置 Script 中所示範一般。

bldrtn 的編譯及鏈結選項	
編譯選項：	
gcc	GNU/Linux C 編譯器。
-fpic	產生位置獨立碼。
-I\$DB2PATH/include	指定 DB2 併入檔的位置。
-c	僅執行編譯；沒有任何鏈結。此 Script 檔有不同的編譯及鏈結步驟。
-D_REENTRANT	需要定義 <code>_REENTRANT</code> ，因為常式可能在其他常式 (THREADSAFE) 執行所在的同一程序中執行，或是在引擎本身 (NOT FENCED) 中執行。

bldrtn 的編譯及鏈結選項

鏈結選項：

gcc 使用編譯器作為鏈結器的前端。

-shared
建立共用檔案庫。

-o \$1 指定可執行檔。

\$1.o 包括程式目的檔。

\$EXTRA_LFLAG

含有值 "-Wl,-rpath,\$DB2PATH/lib" 以設定自從常式當作 `setuid` 執行後的執行時間路徑。

-Wl,-rpath,\$DB2PATH/lib
指定在執行時 DB2 共用檔案庫的位置。例如：`$HOME/sql1lib/lib`。

-L\$DB2PATH/lib
指定在鏈結時 DB2 共用檔案庫的位置。例如：`$HOME/sql1lib/lib`。如果您未指定 `-L` 選項，將使用 `/usr/lib:/lib`。

-ldb2 與 DB2 檔案庫鏈結。

-lpthread
與 POSIX 執行緒檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 225 頁的『在 Linux 上建置 C 常式』

相關範例：

- 『`bldrtn -- Builds Linux C routines (stored procedures or UDFs)`』

在 Linux 上建置 C 多緒應用程式

使用 Linux C 的多緒應用程式必須使用 `-D_REENTRANT` 進行編譯，並使用 `-lpthread` 鏈結。

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sql1lib/samples/c` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script `bldmt` 含有要建置多緒應用程式的命令。

第一個參數 \$1，指定您的來源檔的名稱。第二個參數 \$2，指定您想要與其連接的資料庫的名稱。第三個參數 \$3，指定資料庫的使用者 ID，而 \$4 則指定通行碼。只需要第一個參數 (來源檔名稱)。資料庫名稱、使用者 ID 及通行碼均為可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

程序:

若要從來源檔 `dbthdrs.sqc` 建置範例程式 `dbthdrs`，請輸入：

```
bldmt dbthdrs
```

結果是可執行檔 `dbthdrs`。若要對 `sample` 資料庫執行可執行檔，請輸入下列命令：

```
dbthdrs
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』

相關範例:

- 『`bldmt -- Builds Linux C multi-threaded applications`』
- 『`dbthdrs.sqc -- How to use multiple context APIs on UNIX (C)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

C 多緒應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldmt
# Builds Linux C multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql/lib

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,-rpath$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi
```

```
# If an embedded SQL program, precompile and bind it.
if [ -f $1".sql" ]
then
  ./embprep $1 $2 $3 $4
fi

# Compile the program.
gcc -I$DB2PATH/include -c $1.c -D_REENTRANT

# Link the program.
gcc -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2 -lpthread
```

Linux C++

在 Linux 上建置 C++ 應用程式

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/cpp` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

建置 Script `bldapp`，位於 `sqllib/samples/cpp`，含有要建置 DB2 API 及內嵌式 SQL 應用程式的命令。

第一個參數 `$1`，指定您的來源檔的名稱。這是唯一必要的參數，也是沒有內嵌式 SQL DB2 API 程式唯一需要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 `$2`，指定您要連接的資料庫名稱；第三個參數 `$3` 指定資料庫的使用者 ID，而 `$4` 指定通行碼。

對於內嵌式 SQL 程式，`bldapp` 會傳送參數以前置編譯及連結 Script `embprep`。如果未提供任何資料庫名稱，將使用預設 `sample` 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

下列範例告訴您如何建置及執行 DB2 API 及內嵌式 SQL 應用程式。

若要從來源檔 `cli_info.C` 建置非內嵌式 SQL 範例程式 `cli_info`，請輸入：

```
bldapp cli_info
```

結果是可執行檔 `cli_info`。您可以輸入下面的命令，對 `sample` 資料庫執行可執行檔：

```
cli_info
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 `tbmod.sqC` 建置內嵌式 SQL 應用程式 `tbmod`：

1. 如果連接同一案例上的 `sample` 資料庫，請輸入：

```
bldapp tbmod
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp tbmod database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp tbmod database userid password
```

結果是可執行檔 `tbmod`。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 `sample` 資料庫，則只要輸入可執行檔檔名：

```
tbmod
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
tbmod database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
tbmod database userid password
```

相關概念：

- 第 90 頁的『建置檔』

相關參照：

- 第 65 頁的『C/C++ 範例』
- 第 234 頁的『Linux C++ 應用程式編譯及鏈結選項』

相關範例：

- 『`bldapp -- Builds Linux C++ applications`』
- 『`cli_info.C -- Set and get information at the client level (C++)`』
- 『`tbmod.sqC -- How to modify table data (C++)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

C++ 應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldapp
# Builds Linux C++ applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
```

```

# The default is the standard instance path.
DB2PATH=$HOME/sql/lib

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,-rpath$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.C error-checking utility.
    g++ -I$DB2PATH/include -c utilemb.C
else
    # Compile the utilapi.C error-checking utility.
    g++ -I$DB2PATH/include -c utilapi.C
fi

# Compile the program.
g++ -I$DB2PATH/include -c $1.C

if [ -f $1".sqc" ]
then
    # Link the program with utilemb.o
    g++ -o $1 $1.o utilemb.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2
else
    # Link the program with utilapi.o
    g++ -o $1 $1.o utilapi.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2
fi

```

Linux C++ 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Linux C++ 編譯器建置 C++ 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp 建置 Script 中所示範一般。

bldapp 的編譯及鏈結選項	
編譯選項：	
g++	GNU/Linux C++ 編譯器。
-I\$DB2PATH/include	指定 DB2 併入檔的位置。
-c	僅執行編譯；沒有任何鏈結。此 Script 檔有不同的編譯及鏈結步驟。

bldapp 的編譯及鏈結選項

鏈結選項：

g++ 使用編譯器作為鏈結器的前端。

-o \$1 指定可執行檔。

\$1.o 包括程式目的檔。

utilemb.o

如果是內嵌式 SQL 程式，則有內嵌式 SQL 公用程式的目的檔，可執行錯誤檢查。

utilapi.o

如果是非內嵌式 SQL 程式，則有 DB2 API 公用程式物件檔，可執行錯誤檢查。

\$EXTRA_LFLAG

如果 'RUNTIME=true' 未加上註解，將含有值 "-Wl,-rpath,\$DB2PATH/lib"；不然，沒有任何值。

-Wl,-rpath,\$DB2PATH/lib

指定在執行時 DB2 共用檔案庫的位置。例如：\$HOME/sqllib/lib。

-L\$DB2PATH/lib

指定在鏈結時 DB2 共用檔案庫的位置。例如：\$HOME/sqllib/lib。如果您未指定 -L 選項，將使用 /usr/lib:/lib。

-ldb2 與 DB2 檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 232 頁的『在 Linux 上建置 C++ 應用程式』

相關範例：

- 『bldapp -- Builds Linux C++ applications』

在 Linux 上建置 C++ 常式

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 sqllib/samples/c 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script bldrtn 含有要建置常式 (儲存程序及使用者定義的函數) 的命令。Script 會編譯常式，並將它放入可被從屬站應用程式呼叫的共用檔案庫。它使用 1 或 2 個參數。

第一個參數 \$1，指定您的來源檔的名稱。第二個參數 \$2，指定您想要與其連接的資料庫的名稱。

資料庫參數是選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

程序:

下列範例告訴您如何利用下列建置常式共用檔案庫：

- 儲存程序
- 非內嵌式 SQL 使用者定義的函數 (UDF)
- 內嵌式 SQL 使用者定義的函數 (UDF)

儲存程序共用檔案庫

若要從來源檔 `spserver.sql` 建置範例程式 `spserver`：

1. 如果連接到 `sample` 資料庫，請輸入建置檔名稱及程式名稱：

```
bldrtn spserver
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn spserver database
```

`Script` 檔會將儲存程序複製到伺服器的路徑 `sqllib/function` 中。

2. 接著，在伺服器上執行 `spcat` `Script`，將常式編目：

```
spcat
```

這個 `Script` 將連接到範例資料庫、將取消常式編目 (若之前已呼叫 `spdrop.db2` 將它們編目的話)，然後呼叫 `screate.db2` 來將它們編目，最後切斷資料庫連線。您也可以個別地呼叫 `spdrop.db2` 及 `screate.db2` `Script`。

3. 然後，停止資料庫後再重新啟動，以辨識新的共用檔案庫。

一旦建置共用檔案庫 `spserver` 完成，您即可建置存取共用檔案庫的從屬站應用程式 `spclient`。您可以使用 `Script` `bldapp` 建置 `spclient`。

若要呼叫儲存程序，請輸入下列命令來執行範例從屬站應用程式：

```
spclient database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式可存取共用檔案庫 `spsrv`，並在伺服器資料庫中執行許多儲存程序函數。輸出會傳回從屬站應用程式。

非內嵌式 SQL UDF 共用檔案庫

若要從來源檔 `udfsrv.C` 建置使用者定義的函數程式 `udfsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfsrv
```

Script 檔會將 UDF 複製到 `sqllib/function` 目錄。

一旦您建置了 `udfsrv`，您就可以建置呼叫它的從屬站應用程式 `udfcli`。您可以使用 Script `bldapp` 從來源檔 `udfcli.sqC` 建置 `udfcli`。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

內嵌式 SQL UDF 共用檔案庫

當連接到 `sample` 資料庫時，若要從來源檔 `udfemsrv.sqC` 建置內嵌式 SQL 使用者定義的函數程式 `udfemsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfemsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn udfemsrv database
```

Script 檔會將 UDF 複製到 `sql/lib/function` 目錄。

一旦您建置了 `udfemsrv`，您就可以建置呼叫它的從屬站應用程式 `udfemcli`。您可以使用 Script `bldapp` 從來源檔 `udfemcli.sqC` 建置 `udfemcli`。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfemcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfemsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』
- 第 239 頁的『Linux C++ 常式編譯及鏈結選項』

相關範例:

- 『`bldrtn -- Builds Linux C++ routines (stored procedures and UDFs)`』
- 『`spclient.sqC -- Call various stored procedures (C++)`』
- 『`spserver.sqC -- A variety of types of stored procedures (C++)`』
- 『`udfcli.sqC -- Call a variety of types of user-defined functions (C++)`』
- 『`udfemcli.sqC -- Call a variety of types of embedded SQL user-defined functions. (C++)`』
- 『`udfemsrv.sqC -- Call a variety of types of embedded SQL user-defined functions. (C++)`』

- 『udfsrv.C -- Call a variety of types of user-defined functions (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcat -- To catalog C stored procedures on UNIX』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqc』

C++ 常式的建置 Script

```

#!/bin/sh
# SCRIPT: bldrtn
# Builds Linux C++ routines (stored procedures and UDFs)
# Usage: bldsrv <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1ib

# Set the runtime path since routines run as setuid.
EXTRA_LFLAG="-Wl,-rpath$DB2PATH/lib"

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2
fi

# Compile the program.
g++ -fpic -I$DB2PATH/include -c $1.C -D_REENTRANT

# Link the program and create a shared library.
g++ -shared -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2 -lpthread

# Copy the shared library to the function subdirectory.
# The user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Linux C++ 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Linux C++ 編譯器建置 C++ 常式（儲存程序及使用者定義的函數），如同 bldrtn 建置 Script 中所示範一般。

bldrtn 的編譯及鏈結選項

編譯選項：

- g++** GNU/Linux C++ 編譯器。
- fpic** 產生位置獨立碼。
- I\$DB2PATH/include**
指定 DB2 併入檔的位置。
- c** 僅執行編譯；沒有任何鏈結。此 Script 檔有不同的編譯及鏈結步驟。
- D_REENTRANT**
需要定義 `_REENTRANT`，因為常式可能在其他常式 (THREADSAFE) 執行所在的同一程序中執行，或是在引擎本身 (NOT FENCED) 中執行。

鏈結選項：

- g++** 使用編譯器作為鏈結器的前端。
- shared**
建立共用檔案庫。
- o \$1** 指定可執行檔。
- \$1.o** 包括程式目的檔。
- \$EXTRA_LFLAG**
含有值 `"-Wl,-rpath,$DB2PATH/lib"` 以設定自從常式當作 `setuid` 執行後的執行時間路徑。
- Wl,-rpath,\$DB2PATH/lib**
指定在執行時 DB2 共用檔案庫的位置。例如：`$HOME/sql1lib/lib`。
- L\$DB2PATH/lib**
指定在鏈結時 DB2 固定及共用檔案庫的位置。例如：`$HOME/sql1lib/lib`。如果您未指定 `-L` 選項，將使用 `/usr/lib:/lib`。
- ldb2** 與 DB2 檔案庫鏈結。
- lpthread**
與 POSIX 執行緒檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 235 頁的『在 Linux 上建置 C++ 常式』

相關範例：

- 『`bldrtn -- Builds Linux C++ routines (stored procedures and UDFs)`』

在 Linux 上建置 C++ 多緒應用程式

使用 Linux C++ 的多緒應用程式必須使用 `-D_REENTRANT` 進行編譯，並使用 `-lpthread` 鏈結。

DB2 會提供建置 Script，來編譯及鏈結 C++ 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/cpp` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script 檔 `bldmt` 含有要建置內嵌式 SQL 多緒程式的命令。它最多使用四個參數。

第一個參數 `$1`，指定您的來源檔的名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。第三個參數 `$3`，指定資料庫的使用者 ID，而 `$4` 則指定通行碼。只需要第一個參數 (來源檔名稱)。資料庫名稱、使用者 ID 及通行碼均為可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

程序:

若要從來源檔 `dbthrs.sql` 建置範例程式 `dbthrs`，請輸入：

```
bldmt dbthrs
```

結果是可執行檔 `dbthrs`。若要對 `sample` 資料庫執行可執行檔，請輸入下列命令：

```
dbthrs
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』

相關範例:

- 『`bldmt -- Builds Linux C++ multi-threaded applications`』
- 『`dbthrs.sql -- How to use multiple context APIs on UNIX (C++)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

C++ 多緒應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldmt
# Builds Linux C++ multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
```

```
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-Wl,-rpath$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
fi

# Compile the program.
g++ -I$DB2PATH/include -c $1.C -D_REENTRANT

# Link the program.
g++ -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2 -lpthread
```

第 10 章 Solaris

Forte C	243	在 Solaris 上建置 C++ 多緒應用程式	265
在 Solaris 上建置 C 應用程式.	243	C++ 多緒應用程式的建置 Script	267
C 應用程式的建置 Script	245	Micro Focus COBOL	267
Solaris C 應用程式編譯及鏈結選項	246	在 Solaris 上架構 Micro Focus COBOL 編譯器	267
在 Solaris 上建置 C 常式	248	在 Solaris 上建置 Micro Focus COBOL 應用程式	268
C 常式的建置 Script	251	Micro Focus COBOL 應用程式的建置 Script	270
Solaris C 常式編譯及鏈結選項.	252	Solaris Micro Focus COBOL 應用程式編譯及鏈結選項	270
在 Solaris 上建置 C 多緒應用程式	253	在 Solaris 上建置 Micro Focus COBOL 常式	271
C 多緒應用程式的建置 Script	255	Micro Focus COBOL 常式的建置 Script	273
Forte C++	255	Solaris Micro Focus COBOL 常式編譯及鏈結選項	273
在 Solaris 上建置 C++ 應用程式.	256		
C++ 應用程式的建置 Script.	257		
Solaris C++ 應用程式編譯及鏈結選項	258		
在 Solaris 上建置 C++ 常式	260		
C++ 常式的建置 Script	263		
Solaris C++ 常式編譯及鏈結選項.	264		

本章提供在 Solaris 作業環境上建置應用程式的詳細資訊。關於適用於 Solaris 的最新 DB2 應用程式開發更新組件，請探訪下列網頁：

<http://www.ibm.com/software/data/db2/udb/ad>

Forte C

註：Forte C 在之前是稱為 "SPARCompiler C"。

關於 DB2 CLI 應用程式及常式的建置資訊，請參閱 *CLI Guide and Reference*。

在 Solaris 上建置 C 應用程式

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/c` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

建置 Script `bldapp` 含有要建置 DB2 應用程式的命令。它最多使用四個參數。

第一個參數 \$1，指定您的來源檔的名稱。這是唯一必要的參數，也是沒有內嵌式 SQL DB2 API 程式唯一需要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 \$2，指定您要連接的資料庫名稱；第三個參數 \$3 指定資料庫的使用者 ID，而 \$4 指定通行碼。

對於內嵌式 SQL 程式，bldapp 會傳送參數以前置編譯及連結 Script embprep。如果未提供任何資料庫名稱，將使用預設 sample 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

下列範例告訴您如何建置及執行 DB2 API 及內嵌式 SQL 應用程式。

若要從來源檔 cli_info.c 建置 DB2 API 非內嵌式 SQL 範例程式 cli_info，請輸入：

```
bldapp cli_info
```

結果是可執行檔 cli_info。

欲執行可執行檔，請輸入可執行檔檔名：

```
cli_info
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 tmod.sqc 建置內嵌式 SQL 應用程式 tmod：

1. 如果連接同一案例上的 sample 資料庫，請輸入：

```
bldapp tmod
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp tmod database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp tmod database userid password
```

結果是可執行檔 tmod。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 sample 資料庫，則只要輸入可執行檔檔名：

```
tmod
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
tmod database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
tmod database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』
- 第 246 頁的『Solaris C 應用程式編譯及鏈結選項』

相關範例:

- 『bldapp -- Builds Solaris C applications』
- 『cli_info.c -- Set and get information at the client level (C)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『tbmod.sqc -- How to modify table data (C)』

C 應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldapp
# Builds Solaris C applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1ib

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    CFLAG_ARCH=v9
else
    CFLAG_ARCH=v8plusa
fi

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-R$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.c error-checking utility.
    cc -xarch=$CFLAG_ARCH -I$DB2PATH/include -c utilemb.c
else
```

```

# Compile the utilapi.c error-checking utility.
cc -xarch=$CFLAG_ARCH -I$DB2PATH/include -c utilapi.c
fi

# Compile the program.
cc -xarch=$CFLAG_ARCH -I$DB2PATH/include -c $1.c

if [ -f $1".sqc" ]
then
# Link the program with utilemb.o
cc -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilemb.o $EXTRA_LFLAG \
-L$DB2PATH/lib -ldb2
else
# Link the program with utilapi.o
cc -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilapi.o $EXTRA_LFLAG \
-L$DB2PATH/lib -ldb2
fi

```

Solaris C 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Forte C 編譯器建置 C 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp 建置 Script 中所示範一般。

bldapp 的編譯及鏈結選項	
編譯選項：	
cc	C 編譯器。
-xarch=\$CFLAG_ARCH	這個選項確定當編譯器與 libdb2.so 鏈結時，將產生有效的可執行檔。 \$CFLAG_ARCH 將設為 "v8plusa" (32 位元)，或設為 "v9" (64 位元)。
-I\$DB2PATH/include	指定 DB2 併入檔的位置。例如：\$HOME/sql/lib/include
-c	僅執行編譯；沒有任何鏈結。此 Script 檔有不同的編譯及鏈結步驟。

bldapp 的編譯及鏈結選項

鏈結選項：

cc 使用編譯器作為鏈結器的前端。

-xarch=\$CFLAG_ARCH

這個選項確定當編譯器與 libdb2.so 鏈結時，將產生有效的可執行檔。
\$CFLAG_ARCH 將設為 "v8plusa" (32 位元)，或設為 "v9" (64 位元)。

-mt 在多緒支援中鏈結。與 libdb2 鏈結時，它是必需的。

註： 如果使用 POSIX 執行緒，則不管 DB2 應用程式是否為執行緒，亦須利用 -lpthread 來鏈結。

-o \$1 指定可執行檔。

\$1.o 包括程式目的檔。

utilemb.o

如果是內嵌式 SQL 程式，則有內嵌式 SQL 公用程式的目的檔，可執行錯誤檢查。

utilapi.o

如果不是內嵌式 SQL 程式，則有 DB2 API 公用程式目的檔，可執行錯誤檢查。

\$EXTRA_LFLAG

如果 'RUNTIME=true' 未加上註解，將含有值 "-R\$DB2PATH/lib"；不然，沒有任何值。

-R\$DB2PATH/lib

指定在執行時 DB2 共用檔案庫的位置。例如：\$HOME/sql1lib/lib。

-L\$DB2PATH/lib

指定在鏈結時 DB2 共用檔案庫的位置。例如：\$HOME/sql1lib/lib。如果您未指定 -L 選項，將使用 /usr/lib:/lib。

-ldb2 與 DB2 檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 243 頁的『在 Solaris 上建置 C 應用程式』

相關範例：

- 『bldapp -- Builds Solaris C applications』

在 Solaris 上建置 C 常式

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/c` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script `bldrtn` 含有要建置常式 (儲存程序及使用者定義的函數) 的命令。Script 檔會編譯常式，並將它放入可被資料庫管理程式載入及可被從屬站應用程式呼叫的共用檔案庫。

第一個參數 `$1`，指定您的來源檔的名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。

資料庫參數是選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。因此，既然儲存程序必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

程序:

下列範例告訴您如何利用下列建置常式共用檔案庫：

- 儲存程序
- 非內嵌式 SQL 使用者定義的函數 (UDF)
- 內嵌式 SQL 使用者定義的函數 (UDF)

儲存程序共用檔案庫

若要從來源檔 `spserver.sqc` 建置範例程式 `spserver`：

1. 如果連接到 `sample` 資料庫，請輸入建置 Script 名稱及程式名稱：

```
bldrtn spserver
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn spserver database
```

Script 檔會將共用檔案庫複製到 `sqllib/function` 目錄上的伺服器。

2. 接著，在伺服器上執行 `spcat` Script，將常式編目：

```
spcat
```

這個 Script 將連接到範例資料庫、將取消常式編目 (若之前已呼叫 `spdrop.db2` 將它們編目的話)，然後呼叫 `spcreate.db2` 來將它們編目，最後切斷資料庫連線。您也可以個別地呼叫 `spdrop.db2` 及 `spcreate.db2` Script。

3. 然後，停止資料庫後再重新啟動，以辨識新的共用檔案庫。

一旦建置共用檔案庫 `spserver` 完成，您即可建置存取共用檔案庫的從屬站應用程式 `spclient`。

您可以使用 Script 檔 `bldapp`，建置 `spclient`。

欲呼叫共用檔案庫中的儲存程序，請輸入下列命令以執行範例從屬站應用程式：

```
spclient database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式可存取共用檔案庫 `spserver`，並在伺服器資料庫中執行許多儲存程序函數。輸出會傳回從屬站應用程式。

非內嵌式 SQL UDF 共用檔案庫

若要從來源檔 `udfsrv.c` 建置使用者定義的函數程式 `udfsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfsrv
```

Script 檔會將 UDF 複製到 `sqllib/function` 目錄。

一旦您建置了 `udfsrv`，您就可以建置呼叫它的從屬站應用程式 `udfcli`。已提供 DB2 CLI 及此程式的內嵌式 SQL 版本。您可以使用 Script `bldapp`，從來源檔 `udfcli.c` (位於 `sqllib/samples/cli`) 建置 DB2 CLI `udfcli` 從屬站程式。

您可以使用 Script 檔 `bldapp`，從來源檔 `udfcli.sqc` (位於 `sqllib/samples/c`) 建置內嵌式 SQL `udfcli` 從屬站程式。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

內嵌式 SQL UDF 共用檔案庫

當連接到 `sample` 資料庫時，若要從來源檔 `udfemsrv.sql` 建置內嵌式 SQL 使用者定義的函數程式 `udfemsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfemsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn udfemsrv database
```

Script 檔會將 UDF 檔案庫複製到 `sqllib/function` 目錄。

一旦您建置了 `udfemsrv`，您就可以建置呼叫它的從屬站應用程式 `udfemcli`。您可以使用 Script 檔 `bldapp`，從來源檔 `udfemcli.sql` (位於 `sqllib/samples/c`) 建置 `udfemcli` 從屬站程式。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfemcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfemsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』
- 第 252 頁的『Solaris C 常式編譯及鏈結選項』

相關範例:

- 『bldrtn -- Builds Solaris C routines (stored procedures or UDFs)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcat -- To catalog C stored procedures on UNIX』
- 『spcli.sql -- Call various stored procedures (C)』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sql』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sql』
- 『spserver.sql -- A variety of types of stored procedures (C)』
- 『udfcli.sql -- Call a variety of types of user-defined functions (C)』
- 『udfemcli.sql -- Call a variety of types of embedded SQL user-defined functions. (C)』
- 『udfemsrv.sql -- Call a variety of types of embedded SQL user-defined functions. (C)』
- 『udfsrv.c -- Call a variety of types of user-defined functions (C)』

C 常式的建置 Script

```

#!/bin/sh
# SCRIPT: bldrtn
# Builds Solaris C routines (stored procedures or UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1ib

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    CFLAG_ARCH=v9
else
    CFLAG_ARCH=v8plusa
fi

```

```

# Set the runtime path since routines run as setuid.
EXTRA_LFLAG="-R$DB2PATH/lib"

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sql" ]
then
  ./embprep $1 $2
fi

# Compile the program.
cc -xarch=$CFLAG_ARCH -mt -DUSE_UI_THREADS -Kpic \
  -I$DB2PATH/include -c $1.c

# Link the program and create a shared library
cc -xarch=$CFLAG_ARCH -mt -G -o $1 $1.o $EXTRA_LFLAG \
  -L$DB2PATH/lib -ldb2

# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Solaris C 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Forte C 編譯器建置 C 常式 (儲存程序及使用者定義的函數)，如同 bldrtn 建置 Script 中所示範一般。

bldrtn 的編譯及鏈結選項	
編譯選項：	
cc	C 編譯器。
-xarch=\$CFLAG_ARCH	這個選項確定當編譯器與 libdb2.so 鏈結時，將產生有效的可執行檔。 \$CFLAG_ARCH 將設為 "v8plusa" (32 位元)，或設為 "v9" (64 位元)。
-mt	需要容許多緒支援，因為常式可能在其他常式 (THREADSAFE) 執行所在的同一 程序中執行，或是在引擎本身 (NOT FENCED) 中執行。
-DUSE_UI_THREADS	容許 Sun 的 "UNIX International" 執行緒 API。
-Kpic	為共用檔案庫產生與位置無關的程式碼。
-I\$DB2PATH/include	指定 DB2 併入檔的位置。
-c	僅執行編譯；沒有任何鏈結。此 Script 檔有不同的編譯及鏈結步驟。

bldrtn 的編譯及鏈結選項

鏈結選項：

cc 使用編譯器作為鏈結器的前端。

-xarch=\$CFLAG_ARCH

這個選項確定當編譯器與 `libdb2.so` 鏈結時，將產生有效的可執行檔。
`$CFLAG_ARCH` 將設為 "v8plusa" (32 位元)，或設為 "v9" (64 位元)。

-mt 這是必要的，因為 DB2 檔案庫是利用 `-mt` 來鏈結。

-G 建立共用檔案庫。

-o \$1 指定可執行檔。

\$1.o 包括程式目的檔。

\$EXTRA_LFLAG

含有值 `"-R$DB2PATH/lib"` 以設定自從常式當作 `setuid` 執行後的執行時間路徑。

-R\$DB2PATH/lib

指定在執行時 DB2 共用檔案庫的位置。例如：`$HOME/sql1lib/lib`。

-L\$DB2PATH/lib

指定在鏈結時 DB2 共用檔案庫的位置。例如：`$HOME/sql1lib/lib`。如果您未指定 `-L` 選項，將使用 `/usr/lib:/lib`。

-ldb2 與 DB2 檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 225 頁的『在 Linux 上建置 C 常式』

相關範例：

- 『`bldrtn -- Builds Linux C routines (stored procedures or UDFs)`』

在 Solaris 上建置 C 多緒應用程式

DB2 支援使用 SUN 及 POSIX 執行緒檔案庫的多緒應用程式。預設值是 Sun 執行緒。在 Solaris 上使用 Forte C 的多緒應用程式，必須使用 `-mt` 來編譯及鏈結。如此可傳送 `-D_REENTRANT` 到前置處理器，以及傳送 `-lthread` 到鏈結器。您也需要指定編譯定義 `-DUSE_UI_THREADS`，以使用 Sun 的 "Unix International" 執行緒 API。

註：如果您想要使用 POSIX 執行緒，則必須新增編譯器選項

`-D_POSIX_PTHREAD_SEMANTICS`，這個選項將容許函數的 POSIX 變式，如

`getpwnam_r()`，而且也會新增鏈結選項 `-lpthread`。如果您要使用提供的 `bldmt Script`，則亦須刪除 `-DUSE_UI_THREADS` 定義。

DB2 會提供建置 `Script`，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/c` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

`Script bldmt` 含有要建置多緒應用程式的命令。

第一個參數 `$1`，指定您的來源檔的名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。第三個參數 `$3`，指定資料庫的使用者 ID，而 `$4` 則指定通行碼。只需要第一個參數 (來源檔名稱)。資料庫名稱、使用者 ID 及通行碼均為可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

程序:

若要從來源檔 `dbthrds.sqc` 建置範例程式 `dbthrds`，請輸入：

```
bldmt dbthrds
```

結果是可執行檔 `dbthrds`。若要對 `sample` 資料庫執行可執行檔，請輸入下列命令：

```
dbthrds
```

註：對於擁有許多連線的多緒程式，核心程式參數 `semsys:seminfo_semume` 及 `shmsys:shminfo_shmseg` 可能要設定成預設值以外的值。請在 `db2osconf` 公用程式上參閱底下的相關鏈結，以取得要設定給這些參數的建議值。

相關概念:

- 第 90 頁的『建置檔』
- *Command Reference* 中的『`db2osconf - Utility for Kernel Parameter Values Command`』

相關參照:

- 第 65 頁的『C/C++ 範例』

相關範例:

- 『`bldmt -- Builds Solaris C multi-threaded applications`』
- 『`dbthrds.sqc -- How to use multiple context APIs on UNIX (C)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

C 多緒應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldmt
# Builds Solaris C multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# To compile 64-bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    CFLAG_ARCH=v9
else
    CFLAG_ARCH=v8plusa
fi

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-R$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
fi

# Compile the program.
cc -xarch=$CFLAG_ARCH -mt -DUSE_UI_THREADS -I$DB2PATH/include -c $1.c

# Link the program.
cc -xarch=$CFLAG_ARCH -mt -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2
```

Forte C++

註: Forte C++ 在之前是稱為 SPARCompiler C++

在 Solaris 上建置 C++ 應用程式

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqlllib/samples/cpp` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

建置 Script `bldapp` 含有要建置 DB2 API 及內嵌式 SQL 應用程式的命令。它最多使用四個參數。

第一個參數 `$1`，指定您的來源檔的名稱。這是唯一必要的參數，也是沒有內嵌式 SQL DB2 API 程式唯一需要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 `$2`，指定您要連接的資料庫名稱；第三個參數 `$3` 指定資料庫的使用者 ID，而 `$4` 指定通行碼。

對於內嵌式 SQL 程式，`bldapp` 會傳送參數以前置編譯及鏈結 Script `embprep`。如果未提供任何資料庫名稱，將使用預設 `sample` 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

下列範例告訴您如何建置及執行 DB2 API 及內嵌式 SQL 應用程式。

若要從來源檔 `cli_info.C` 建置非內嵌式 SQL 範例程式 `cli_info`，請輸入：

```
bldapp cli_info
```

結果是可執行檔 `cli_info`。您可以輸入下面的命令，對 `Sample` 資料庫執行可執行檔：

```
cli_info
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 `tbmod.sqC` 建置內嵌式 SQL 應用程式 `tbmod`：

1. 如果連接同一案例上的 `sample` 資料庫，請輸入：

```
bldapp tbmod
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp tbmod database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp tbmod database userid password
```

結果是可執行檔 `tbmod`。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 `sample` 資料庫，則只要輸入可執行檔檔名：

```
tbmod
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
tbmod database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
tbmod database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』
- 第 258 頁的『Solaris C++ 應用程式編譯及鏈結選項』

相關範例:

- 『`bldapp -- Builds Solaris C++ applications`』
- 『`cli_info.C -- Set and get information at the client level (C++)`』
- 『`tbmod.sqlC -- How to modify table data (C++)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

C++ 應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldapp
# Builds Solaris C++ applications
# Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# To compile 64 bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    CFLAG_ARCH=v9
else
    CFLAG_ARCH=v8plusa
fi

# To specify a runtime path for shared libraries, uncomment the
```

```

# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-R$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f "$1".sqc ]
then
    ./embprep $1 $2 $3 $4
    # Compile the utilemb.C error-checking utility.
    CC -xarch=$CFLAG_ARCH -I$DB2PATH/include -c utilemb.C
else
    # Compile the utilapi.C error-checking utility.
    CC -xarch=$CFLAG_ARCH -I$DB2PATH/include -c utilapi.C
fi

# Compile the program.
CC -xarch=$CFLAG_ARCH -I$DB2PATH/include -c $1.C

if [ -f "$1".sqc ]
then
    # Link the program with utilemb.o
    CC -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilemb.o $EXTRA_LFLAG \
        -L$DB2PATH/lib -ldb2
else
    # Link the program with utilapi.o
    CC -xarch=$CFLAG_ARCH -mt -o $1 $1.o utilapi.o $EXTRA_LFLAG \
        -L$DB2PATH/lib -ldb2
fi

```

Solaris C++ 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Forte C++ 編譯器建置 C++ 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp 建置 Script 中所示範一般。

bldapp 的編譯及鏈結選項	
編譯選項：	
CC	C++ 編譯器。
-xarch=\$CFLAG_ARCH	這個選項確定當編譯器與 libdb2.so 鏈結時，將產生有效的可執行檔。 \$CFLAG_ARCH 將設為 "v8plusa" (32 位元)，或設為 "v9" (64 位元)。
-I\$DB2PATH/include	指定 DB2 併入檔的位置。例如：\$HOME/sql1lib/include
-c	僅執行編譯；沒有任何鏈結。此 Script 檔有不同的編譯及鏈結步驟。

bldapp 的編譯及鏈結選項

鏈結選項：

CC 使用編譯器作為鏈結器的前端。

-xarch=\$CFLAG_ARCH

這個選項確定當編譯器與 libdb2.so 鏈結時，將產生有效的可執行檔。
\$CFLAG_ARCH 將設為 "v8plusa" (32 位元)，或設為 "v9" (64 位元)。

-mt 在多緒支援中鏈結。與 libdb2 鏈結時，它是必需的。

註： 如果使用 POSIX 執行緒，則不管 DB2 應用程式是否為執行緒，亦須利用 -lpthread 來鏈結。

-o \$1 指定可執行檔。

\$1.o 包括程式目的檔。

utilemb.o

如果是內嵌式 SQL 程式，則有內嵌式 SQL 公用程式的目的檔，可執行錯誤檢查。

utilapi.o

如果是非內嵌式 SQL 程式，則有 DB2 API 公用程式目的檔，可執行錯誤檢查。

\$EXTRA_LFLAG

如果 'RUNTIME=true' 未加上註解，將含有值 "-R\$DB2PATH/lib"；不然，沒有任何值。

-R\$DB2PATH/lib

指定在執行時 DB2 共用檔案庫的位置。例如：\$HOME/sql1lib/lib。

-L\$DB2PATH/lib

指定在鏈結時 DB2 共用檔案庫的位置。例如：\$HOME/sql1lib/lib。如果您未指定 -L 選項，將使用 /usr/lib:/lib。

-ldb2 與 DB2 檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 256 頁的『在 Solaris 上建置 C++ 應用程式』

相關範例：

- 『bldapp -- Builds Solaris C++ applications』

在 Solaris 上建置 C++ 常式

DB2 會提供建置 Script，來編譯及鏈結 C 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/c` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script `bldrtn` 含有要建置常式 (儲存程序或使用者定義的函數) 的命令。Script 會編譯常式，並將它放入可被從屬站應用程式呼叫的共用檔案庫。

第一個參數 `$1`，指定您的來源檔的名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。

資料庫參數是選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。因此，既然儲存程序必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

程序:

下列範例告訴您如何利用下列建置常式共用檔案庫：

- 儲存程序
- 非內嵌式 SQL 使用者定義的函數 (UDF)
- 內嵌式 SQL 使用者定義的函數 (UDF)

儲存程序共用檔案庫

若要从來源檔 `spserver.sqC` 建置範例程式 `spserver`：

1. 如果連接到 `sample` 資料庫，請輸入建置檔名稱及程式名稱：

```
bldrtn spserver
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn spserver database
```

Script 檔會將共用檔案庫複製到伺服器上的 `sqllib/function` 目錄。

2. 接著，在伺服器上執行 `spcat` Script，將常式編目：

```
spcat
```

這個 Script 將連接到範例資料庫、將取消常式編目 (若之前已呼叫 `spdrop.db2` 將它們編目的話)，然後呼叫 `spcreate.db2` 來將它們編目，最後切斷資料庫連線。您也可以個別地呼叫 `spdrop.db2` 及 `spcreate.db2` Script。

3. 然後，停止資料庫後再重新啟動，以辨識新的共用檔案庫。

一旦建置共用檔案庫 `spserver` 完成，您即可建置存取共用檔案庫的從屬站應用程式 `spclient`。您可以使用 Script `bldapp` 建置 `spclient`。

欲呼叫共用檔案庫中的儲存程序，請輸入下列命令以執行範例從屬站應用程式：

```
spclient database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式可存取共用檔案庫 `spserver`，並在伺服器資料庫中執行許多儲存程序函數。輸出會傳回從屬站應用程式。

非內嵌式 SQL UDF 共用檔案庫

若要從來源檔 `udfsrv.C` 建置使用者定義的函數程式 `udfsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfsrv
```

Script 檔會將 UDF 複製到 `sqllib/function` 目錄。

一旦您建置了 `udfsrv`，您就可以建置呼叫它的從屬站應用程式 `udfcli`。您可以使用 Script 檔 `bldapp` 從來源檔 `udfcli.sqC` 建置 `udfcli`。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfcli database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

內嵌式 SQL UDF 共用檔案庫

當連接到 `sample` 資料庫時，若要從來源檔 `udfemsrv.sql` 建置內嵌式 SQL 使用者定義的函數程式 `udfemsrv`，請輸入建置 Script 名稱及程式名稱：

```
bldrtn udfemsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn udfemsrv database
```

Script 檔會將 UDF 複製到 `sqllib/function` 目錄。

必要時，請設定 UDF 的檔案模式，以便從屬站應用程式資料庫管理程式可以存取它。

一旦您建置了 `udfemsrv`，您就可以建置呼叫它的從屬站應用程式 `udfemcli`。您可以使用 Script 檔 `bldapp` 從來源檔 `udfemcli.sql` 建置 `udfemcli`。

若要呼叫共用檔案庫中的 UDF，請輸入下列命令以執行從屬站應用程式：

```
udfemcli database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式會存取共用檔案庫 `udfemsrv`，並在伺服器資料庫上執行使用者定義的函數。輸出會傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 65 頁的『C/C++ 範例』
- 第 264 頁的『Solaris C++ 常式編譯及鏈結選項』

相關範例:

- 『bldrtn -- Builds Solaris C++ routines (stored procedures or UDFs)』
- 『spclient.sqC -- Call various stored procedures (C++)』
- 『spserver.sqC -- A variety of types of stored procedures (C++)』
- 『udfcli.sqC -- Call a variety of types of user-defined functions (C++)』
- 『udfsrv.C -- Call a variety of types of user-defined functions (C++)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqC』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqC』

C++ 常式的建置 Script

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds Solaris C++ routines (stored procedures or UDFs)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# To compile 64-bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    CFLAG_ARCH=v9
else
    CFLAG_ARCH=v8plusa
fi

# Set the runtime path since routines run as setuid.
EXTRA_LFLAG="-R$DB2PATH/lib"

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqC" ]
then
    ./embprep $1 $2
fi

# Compile the program.
CC -xarch=$CFLAG_ARCH -mt -DUSE_UI_THREADS -Kpic \
    -I$DB2PATH/include -c $1.C

# Link the program and create a shared library
```

```

CC -xarch=$CFLAG_ARCH -mt -G -o $1 $1.o $EXTRA_LFLAG \
-L$DB2PATH/lib -ldb2

# Copy the shared library to the sqllib/function subdirectory.
# Note: the user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function

```

Solaris C++ 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Forte C++ 編譯器建置 C++ 常式（儲存程序及使用者定義的函數），如同 bldrtn 建置 Script 中所示範一般。

bldrtn 的編譯及鏈結選項	
編譯選項：	
CC	C++ 編譯器。
-xarch=\$CFLAG_ARCH	這個選項確定當編譯器與 libdb2.so 鏈結時，將產生有效的可執行檔。 \$CFLAG_ARCH 將設為 "v8plusa" (32 位元)，或設為 "v9" (64 位元)。
-mt	需要容許多緒支援，因為常式可能在其他常式 (THREADSAFE) 執行所在的同一程序中執行，或是在引擎本身 (NOT FENCED) 中執行。
-DUSE_UI_THREADS	容許 Sun 的 "UNIX International" 執行緒 API。
-Kpic	為共用檔案庫產生與位置無關的程式碼。
-I\$DB2PATH/include	指定 DB2 併入檔的位置。
-c	僅執行編譯；沒有任何鏈結。此 Script 檔有不同的編譯及鏈結步驟。

bldrtn 的編譯及鏈結選項

鏈結選項：

CC 使用編譯器作為鏈結器的前端。

-xarch=\$CFLAG_ARCH

這個選項確定當編譯器與 libdb2.so 鏈結時，將產生有效的可執行檔。
\$CFLAG_ARCH 將設為 "v8plusa" (32 位元)，或設為 "v9" (64 位元)。

-mt 這是必要的，因為 DB2 檔案庫是利用 -mt 來鏈結。

-G 建立共用檔案庫。

-o \$1 指定可執行檔。

\$1.o 包括程式目的檔。

\$EXTRA_LFLAG

含有值 "-R\$DB2PATH/lib" 以設定自從常式當作 setuid 執行後的執行時間路徑。

-R\$DB2PATH/lib

指定在執行時 DB2 共用檔案庫的位置。例如：\$HOME/sql1lib/lib。

-L\$DB2PATH/lib

指定在鏈結時 DB2 共用檔案庫的位置。例如：\$HOME/sql1lib/lib。如果您未指定 -L 選項，將使用 /usr/lib:/lib。

-ldb2 與 DB2 檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 260 頁的『在 Solaris 上建置 C++ 常式』

相關範例：

- 『bldrtn -- Builds Solaris C++ routines (stored procedures or UDFs)』

在 Solaris 上建置 C++ 多緒應用程式

DB2 支援使用 SUN 及 POSIX 執行緒檔案庫的多緒應用程式。預設值是 Sun 執行緒。在 Solaris 上使用 Forte C++ 的多緒應用程式，必須使用 -mt 來編譯及鏈結。如此可傳送 -D_REENTRANT 到前置處理器，以及傳送 -lthread 到鏈結器。您也需要指定編譯定義 -DUSE_UI_THREADS，以使用 Sun 的 "Unix International" 執行緒 API。

註：如果您想要使用 POSIX 執行緒，則必須新增編譯器選項

-D_POSIX_PTHREAD_SEMANTICS，這個選項將容許函數的 POSIX 變式，如

`getpwnam_r()`，而且也會新增鏈結選項 `-lpthread`。如果您要使用提供的 `bldmt Script`，則亦須刪除 `-DUSE_UI_THREADS` 定義。

DB2 會提供建置 `Script`，來編譯及鏈結 C++ 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/cpp` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

`Script bldmt` 含有要建置多緒應用程式的命令。它最多使用四個參數。

第一個參數 `$1`，指定您的來源檔的名稱。第二個參數 `$2`，指定您想要與其連接的資料庫的名稱。第三個參數 `$3`，指定資料庫的使用者 ID，而 `$4` 則指定通行碼。只需要第一個參數 (來源檔名稱)。資料庫名稱、使用者 ID 及通行碼均為可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

程序:

若要從來源檔 `dbthrd.sqC` 建置範例程式 `dbthrds`，請輸入：

```
bldmt dbthrds
```

結果是可執行檔 `dbthrds`。若要對 `sample` 資料庫執行可執行檔，請輸入下列命令：

```
dbthrds
```

註：對於擁有許多連線的多緒程式，核心程式參數 `semsys:seminfo_semume` 及 `shmsys:shminfo_shmseg` 可能要設定成預設值以外的值。請在 `db2osconf` 公用程式上參閱底下的相關鏈結，以取得要設定給這些參數的建議值。

相關概念:

- 第 90 頁的『建置檔』
- *Command Reference* 中的『`db2osconf - Utility for Kernel Parameter Values Command`』

相關參照:

- 第 65 頁的『C/C++ 範例』

相關範例:

- 『`bldmt -- Builds Solaris C++ multi-threaded applications`』
- 『`dbthrd.sqC -- How to use multiple context APIs on UNIX (C++)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

C++ 多緒應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldmt
# Builds Solaris C++ multi-threaded applications
# Usage: bldmt <prog_name> [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# To compile 64-bit programs, uncomment the following line.
# BUILD_64BIT=true

if [ "$BUILD_64BIT" != "" ]
then
    CFLAG_ARCH=v9
else
    CFLAG_ARCH=v8plusa
fi

# To specify a runtime path for shared libraries, uncomment the
# following line (usually only needed for setuid applications).
# RUNTIME=true

if [ "$RUNTIME" != "" ]
then
    EXTRA_LFLAG="-R$DB2PATH/lib"
else
    EXTRA_LFLAG=""
fi

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqc" ]
then
    ./embprep $1 $2 $3 $4
fi

# Compile the program.
CC -xarch=$CFLAG_ARCH -mt -DUSE_UI_THREADS -I$DB2PATH/include -c $1.C

# Link the program.
CC -xarch=$CFLAG_ARCH -mt -o $1 $1.o $EXTRA_LFLAG -L$DB2PATH/lib -ldb2
```

Micro Focus COBOL

在 Solaris 上架構 Micro Focus COBOL 編譯器

若開發含有內嵌式 SQL 和 DB2 API 呼叫的應用程式，而且使用 Micro Focus COBOL 編譯器，以下是您必須記住的幾點：

程序：

- 當您在使用命令行處理器命令 `db2 prep` 前置編譯您的應用程式時，請使用 `target mfcob` 選項。
- 您必須在 Micro Focus COBOL 環境變數 `COBCPY` 中，併入 DB2 COBOL COPY 檔案目錄。`COBCPY` 環境變數會指定 `COPY` 檔的位置。Micro Focus COBOL 的 DB2 COPY 檔常駐在資料庫案例目錄下的 `sqllib/include/cobol_mf` 中。

若要包括目錄，請輸入：

- 在 `bash` 或 Korn shell：

```
export COBCPY=$COBCPY:$HOME/sqllib/include/cobol_mf
```

- 在 C shell：

```
setenv COBCPY $COBCPY:$HOME/sqllib/include/cobol_mf
```

註：您可能想要在 `.profile` 檔中設定 `COBCPY`。

相關作業：

- 第 271 頁的『在 Solaris 上建置 Micro Focus COBOL 常式』
- 第 268 頁的『在 Solaris 上建置 Micro Focus COBOL 應用程式』

在 Solaris 上建置 Micro Focus COBOL 應用程式

DB2 會提供建置 Script，來編譯及鏈結 Micro Focus COBOL 內嵌式 SQL 及 DB2 API 程式。這些都位於 `sqllib/samples/cobol_mf` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

建置 Script `bldapp` 含有要建置 DB2 應用程式的命令。它最多使用四個參數。

第一個參數 `$1`，指定您的來源檔的名稱。對沒有內嵌式 SQL 的程式而言，這是唯一必要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 `$2`，指定您要連接的資料庫名稱；第三個參數 `$3` 指定資料庫的使用者 ID，而 `$4` 指定通行碼。

對於內嵌式 SQL 程式，`bldapp` 會傳送參數以前置編譯及連結 Script `embprep`。如果未提供任何資料庫名稱，將使用預設 `sample` 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序：

若要從來源檔 `client.cbl` 建置非內嵌式 SQL 範例程式 `client`，請輸入：

```
bldapp client
```


結果產生可執行檔 `client`。您可以輸入下面的命令，對 `Sample` 資料庫執行可執行檔：

```
client
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 `updat.sqb` 建置內嵌式 SQL 應用程式 `updat`：

1. 如果連接同一案例上的 `sample` 資料庫，請輸入：

```
bldapp updat
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp updat database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp updat database userid password
```

結果產生是可執行檔 `updat`。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 `sample` 資料庫，則只要輸入可執行檔檔名：

```
updat
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
updat database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
updat database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 270 頁的『Solaris Micro Focus COBOL 應用程式編譯及鏈結選項』
- 第 76 頁的『COBOL 範例』

相關範例:

- 『`bldapp -- Builds Solaris Micro Focus COBOL applications`』
- 『`client.cbl -- How to set and query a client (MF COBOL)`』
- 『`updat.sqb -- How to update, delete and insert table data (MF COBOL)`』
- 『`embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs`』

Micro Focus COBOL 應用程式的建置 Script

```
#!/bin/sh
# SCRIPT: bldapp
# Builds Solaris Micro Focus COBOL applications
# Usage: bldapp [ <db_name> [ <userid> <password> ]]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sql1lib

# If an embedded SQL program, precompile and bind it.
if [ -f "$1".sqb ]
then
    ./embprep $1 $2 $3 $4
fi

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Compile the checkerr.cbl error-checking utility.
cob -cx checkerr.cbl

# Compile the program.
cob -cx $1.cbl

# Link the program.
cob -x $1.o checkerr.o -L$DB2PATH/lib -ldb2 -ldb2gmf
```

Solaris Micro Focus COBOL 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Micro Focus COBOL 編譯器在 Solaris 上建置 COBOL 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp 建置 Script 中所示範一般。

bldapp 的編譯及鏈結選項	
編譯選項：	
cob	Micro Focus COBOL 編譯器。
-cx	編譯至物件模組。

bldapp 的編譯及鏈結選項

鏈結選項：

cob 使用編譯器作為鏈結器的前端。

-x 指定可執行程式。

\$1.o 包括程式目的檔。

checkerr.o

包括公用程式目的檔以便檢查錯誤。

-L\$DB2PATH/lib

指定在鏈結時 DB2 固定及共用檔案庫的位置。例如：\$HOME/sql1lib/lib。

-ldb2 與 DB2 檔案庫鏈結。

-ldb2gmf

鏈結 Micro Focus COBOL 的 DB2 例外處理程式檔案庫。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 268 頁的『在 Solaris 上建置 Micro Focus COBOL 應用程式』

相關範例：

- 『bldapp -- Builds Solaris Micro Focus COBOL applications』

在 Solaris 上建置 Micro Focus COBOL 常式

DB2 會提供建置 Script，來編譯及鏈結 Micro Focus COBOL 內嵌式 SQL 及 DB2 API 程式。這些都位於 sql1lib/samples/cobol_mf 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

Script bldrtn 含有要建置常式 (儲存程序) 的命令。Script 檔會編譯常式，並將它放入可被從屬站應用程式呼叫的共用檔案庫。

第一個參數 \$1，指定您的來源檔的名稱。Script 檔會使用來源檔名稱 \$1 作為共用檔案庫名稱。第二個參數 \$2，指定您想要與其連接的資料庫的名稱。既然常式必須建置在與資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

只有第一個參數是必要的，即來源檔名稱。資料庫名稱是可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 sample。

程序：

在 Solaris 上建置 Micro Focus 常式前，請執行下列命令：

```
db2stop
db2set DB2LIBPATH=$LD_LIBRARY_PATH
db2set DB2ENVLIST="COBDIR LD_LIBRARY_PATH"
db2set
db2start
```

請確定 db2stop 會停止資料庫。最後發出的 db2set 命令是爲了檢查您的設定值；請確定 DB2LIBPATH 及 DB2ENVLIST 的設定正確。

當連接到範例資料庫時，若要從來源檔 `outsrv.sqb` 建置範例程式 `outsrv`，請輸入：

```
bldsrv outsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldsrv outsrv database
```

Script 檔會將共用檔案庫複製到伺服器的路徑 `sqllib/function` 中。

一旦建置了共用檔案庫 `outsrv`，您就可以建置從屬站應用程式 `outcli`，呼叫其中同名的常式。您可以使用 Script 檔 `bldapp` 建置 `outcli`。

若要呼叫儲存程序，請輸入下列命令來執行範例從屬站應用程式：

```
outcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample` 或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式將存取共用檔案庫 `outsrv`，以在伺服器資料庫中執行名稱相同的常式，然後將輸出傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 273 頁的『Solaris Micro Focus COBOL 常式編譯及鏈結選項』
- 第 76 頁的『COBOL 範例』

相關範例:

- 『bldrtn -- Builds Solaris Micro Focus COBOL routines (stored procedures)』
- 『outcli.sqb -- Call stored procedures using the SQLDA structure (MF COBOL)』
- 『outsrv.sqb -- Demonstrates stored procedures using the SQLDA structure (MF COBOL)』
- 『embprep -- To prep and bind C/C++ and Micro Focus COBOL embedded SQL programs』

Micro Focus COBOL 常式的建置 Script

```
#!/bin/sh
# SCRIPT: bldrtn
# Builds Solaris Micro Focus COBOL routines (stored procedures)
# Usage: bldrtn <prog_name> [ <db_name> ]

# Set DB2PATH to where DB2 will be accessed.
# The default is the standard instance path.
DB2PATH=$HOME/sqllib

# If an embedded SQL program, precompile and bind it.
if [ -f $1".sqb" ]
then
    ./embprep $1 $2
fi

# Set COBCPY to include the DB2 COPY files directory.
COBCPY=$COBCPY:$DB2PATH/include/cobol_mf

# Compile the program.
cob -cx $1.cbl

# Link the program.
cob -x -o $1 $1.o -Q -G -L$DB2PATH/lib -ldb2 -ldb2gmf

# Copy the shared library to the sqllib/function subdirectory.
# The user must have write permission to this directory.
rm -f $DB2PATH/function/$1
cp $1 $DB2PATH/function
```

Solaris Micro Focus COBOL 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Micro Focus COBOL 編譯器在 Solaris 上建置 COBOL 常式 (儲存程序)，如同 bldrtn 建置 Script 中所示範一般。

bldrtn 的編譯及鏈結選項

編譯選項：

- cob** COBOL 編譯器。
- cx** 編譯至物件模組。

鏈結選項：

- cob** 使用編譯器作為鏈結器的前端。
- x** 與 **-G** 選項一起使用時，將產生共用檔案庫。
- o \$1** 指定可執行檔程式。
- \$1.o** 指定程式目的檔。
- Q -G** 建立共用檔案庫。
- L\$DB2PATH/lib**
指定 DB2 執行程式共用檔案庫的位置。例如：`$HOME/sql1ib/lib`。如果您未指定 **-L** 選項，則編譯器將採用下列路徑：`/usr/lib:/lib`。
- ldb2** 鏈結 DB2 檔案庫。
- ldb2gmf**
鏈結 Micro Focus COBOL 的 DB2 例外處理程式檔案庫。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 271 頁的『在 Solaris 上建置 Micro Focus COBOL 常式』

相關範例：

- 『`bldrtn -- Builds Solaris Micro Focus COBOL routines (stored procedures)`』

第 11 章 Windows 作業系統

WCHARTYPE CONVERT 前置編譯選項 . . .	275	Windows IBM COBOL 應用程式編譯及鏈	
物件連結和內嵌資料庫 (OLE DB) 表格函數	276	結選項	299
Windows Management Instrumentation (WMI)	277	在 Windows 上建置 IBM COBOL 常式	299
Microsoft Visual Basic	277	IBM COBOL 常式的批次檔	301
利用 Visual Basic 建置 ADO 應用程式	277	Windows IBM COBOL 常式編譯及鏈結選	
利用 Visual Basic 建置 RDO 應用程式	280	項	301
利用 Visual Basic 的物件連結和內嵌		Micro Focus COBOL	302
(OLE) 自動化	282	在 Windows 上架構 Micro Focus COBOL	
Microsoft Visual C++.	283	編譯器	302
利用 Visual C++ 建置 ADO 應用程式 . . .	283	在 Windows 上建置 Micro Focus COBOL	
利用 Visual C++ 的物件連結和內嵌 (OLE)		應用程式	303
自動化	285	Micro Focus COBOL 應用程式的批次檔	305
在 Windows 上建置 C/C++ 應用程式 . . .	286	Windows Micro COBOL 應用程式編譯及	
C/C++ 應用程式的批次檔	287	鏈結選項	305
Windows C/C++ 應用程式編譯及鏈結選項	288	在 Windows 上建置 Micro Focus COBOL	
在 Windows 上建置 C/C++ 常式	290	常式	306
C/C++ 常式的批次檔	293	Micro Focus COBOL 常式的批次檔 . . .	307
Windows C/C++ 常式編譯及鏈結選項 . . .	294	Windows Micro COBOL 常式編譯及鏈結	
IBM VisualAge COBOL	296	選項	308
在 Windows 上架構 IBM COBOL 編譯器	296	物件 REXX	308
在 Windows 上建置 IBM COBOL 應用程		在 Windows 上建置 Object REXX 應用程	
式	297	式	308
IBM COBOL 應用程式的批次檔	298		

本章提供在 Windows 作業系統上建置應用程式的詳細資訊。關於適用於 Windows 的最新 DB2 應用程式開發更新組件，請探訪下列網頁：

<http://www.ibm.com/software/data/db2/udb/ad>

WCHARTYPE CONVERT 前置編譯選項

WCHARTYPE 前置編譯選項使用 `wchar_t` 資料類型，處理多位元組格式或寬字元格式的圖形資料。

就 DB2[®] for Windows[®] 作業系統而言，WCHARTYPE CONVERT 選項支援使用 Microsoft[®] Visual C++ 編譯器編譯的應用程式。不過若應用程式在與資料庫字碼頁不同的字碼頁中將資料插入 DB2 資料庫，請勿對此編譯器使用 CONVERT 選項。DB2 通常會在此狀況下執行字碼頁轉換；不過 Microsoft C 執行環境不處理某些雙位元組字元的字元替代。這樣會導致執行轉換錯誤。

WCHARTYPE 的預設選項是 NOCONVERT。透過 NOCONVERT 選項，在應用程式與資料庫管理程式之間就不會發生隱含字元轉換。從資料庫管理程式收送資料圖形主變數中的資料作為不變的「雙位元組字集」(DBCS) 字元。

若需要將圖形資料從多位元組格式轉換成寬字元格式，請使用 `wcstombs()` 函數。例如：

```
wchar_t widechar[200];  
wchar_t mb[200];  
wcstombs((char *)mb,widechar,200);  
  
EXEC SQL INSERT INTO TABLENAME VALUES(:mb);
```

同樣地，您可使用 `mbstowcs()` 函數從多位元組格式轉換成寬字元格式。

若應用程式靜態連結 C 執行檔案庫，請不要從應用程式發出 `setlocale()` 呼叫，因為這樣會導致 C 執行轉換錯誤。若應用程式動態連結 C 執行檔案庫，那麼就可以使用 `setlocale()`。對於常式（儲存程序及使用者定義的函數）來說，情況也是一樣。

相關參照:

- *Command Reference* 中的『PRECOMPILE Command』

物件連結和內嵌資料庫 (OLE DB) 表格函數

DB2® 支援 OLE DB 表格函數。就這些函數而言，除了建立 CREATE FUNCTION DDL 以外不需要建置應用程式。DB2 在 `sql1lib\samples\oledb` 目錄中提供 OLE DB 表格函數範例檔。這些是「命令行處理器」(CLP) 檔。可執行下列步驟建置它們：

1. `db2 connect to database_name`
2. `db2 -t -v -f file_name.db2`
3. `db2 terminate`

其中 `database_name` 是正在連接的資料庫，`file_name` 是副檔名為 `.db2` 的 CLP 檔之名稱。

這些命令必須在「DB2 命令視窗」中進行。

相關參照:

- 第 85 頁的『物件連結和內嵌資料庫 (OLE DB) 表格函數範例』

Windows Management Instrumentation (WMI)

Windows[®] Management Instrumentation (WMI) 是 Microsoft Windows 管理服務的關鍵元件。WMI 會對應用程式及系統的架構、狀態及作業層面提供一致且說明豐富的模型。

DB2[®] WMI 提供者容許 WMI 應用程式監督 DB2 伺服器服務、列舉並建立資料庫、架構作業設定值，以及執行資料庫備份、復置及 Roll-Forward 作業。

DB2 會提供適用於 Visual Basic Scripting 語言 (位於 sql1lib\samples\wmi 目錄) 的 WMI 範例檔。在執行範例程式前，請確定已執行下列命令登錄了「DB2 WMI 提供者」：

```
mofcomp %DB2PATH%\bin\db2wmi.mof
regsvr32 %DB2PATH%\bin\db2wmi.dll
```

其中 %DB2PATH% 是安裝 DB2 的路徑。

使用 cscript 命令以執行 Visual Basic Script 範例。例如，若要執行 listsvr 範例 Script，請輸入：

```
cscript listsvr.vbs
```

相關參照:

- 第 84 頁的『Windows Management Instrumentation 範例』

Microsoft Visual Basic

利用 Visual Basic 建置 ADO 應用程式

「ActiveX 資料物件 (ADO)」可讓您透過 OLE DB 提供者，撰寫應用程式來存取及操作資料庫伺服器內的資料。ADO 的主要優點是快速、容易使用、記憶體超負荷量低以及少用磁碟。

Visual Basic ADO 範例程式位於 sql1lib\samples\VB\ADO 目錄。

註: 若要執行 DB2 ADO 範例，建議使用下列元件的這些版本或更新版本：

1. Visual Basic 6.0 Professional Edition
2. Microsoft Data Access 2.7 SDK (可選擇隨 DB2 版本 8 一起安裝)
3. Visual Basic Service Pack 5，可自 <http://msdn.microsoft.com/vstudio/sp/vs6sp5/vbfixes.asp> 取得。

4. 最新的 Visual Studio Service Pack，可自 <http://msdn.microsoft.com/vstudio/> 取得。

程序:

您可以使用這兩個遵守 ODBC 提供者之一：

- DB2 的 IBM OLE DB 提供者
- ODBC 的 Microsoft OLE DB 提供者

使用 DB2 的 IBM OLE DB 提供者

Windows 作業系統上的 DB2 版本 8.1 從屬站將選用性地安裝 IBMDADB2 (DB2 的遵守 IBM OLE DB 2.0 提供者)。提供者會提供介面，供想要存取 DB2 資料庫中資料的消費者使用。DB2 的 IBM OLE DB 提供者支援下列 ADO 應用程式類型：

- Microsoft Active Server Pages (ASP)
- Microsoft Visual Studio C++ 及 Visual Basic 應用程式
- Microsoft Visual Interdev

若需有關這些應用程式類型的詳細資訊，請參照 ADO 文件。

若要用 DB2 的 IBM OLE DB 提供者存取 DB2 伺服器，Visual Basic 應用程式應在 ADO 連線字串中指定 PROVIDER 關鍵字，如下所示：

```
Dim c1 As ADODB.Connection
Dim c1str As String
c1str = "Provider=ibmdadb2; DSN=db2alias; UID=userid; PWD=password"
c1.Open c1str
...
```

其中 db2alias 是在 DB2 資料庫目錄中編目的 DB2 資料庫別名。

註：使用 DB2 的 IBM OLE DB 提供者時，您不需要執行資料來源的 ODBC 編目步驟。當使用 ODBC 的 OLE DB 提供者時，才需要這個步驟。

使用 ODBC 的 Microsoft OLE DB 提供者

若要使用 ADO 與 Microsoft OLE DB 提供者及 Visual Basic，您必須建立 ADO 類型檔案庫的參照。請執行下列各項：

1. 從「專案」功能表選取「參照」
2. 勾選「Microsoft ActiveX Data Objects <version_number> Library」方框
3. 按一下「確定」。

其中 <version_number> 是 ADO 檔案庫的現行版本。

完成此參照後，可以透過「VBA 物件瀏覽器」和「IDE 編輯器」存取 ADO 物件、方法和內容。

建立連線：

```
Dim db As Connection
Set db = New Connection
```

設定本端游標檔案庫提供的從屬站端游標：

```
db.CursorLocation = adUseClient
```

並設定提供者，以便 ADO 使用 Microsoft ODBC 驅動程式。

使用 ADO 存取範例資料庫

完整 Visual Basic 程式包含表格和其它圖形元素，您必須在 Visual Basic 環境中才能檢視它。以下是一些 Visual Basic 命令，這些命令是在您使用 IBM OLE DB 提供者或 Microsoft OLE DB 提供者連接資料庫後 (如前面所述)，作為程式的一部份以存取 DB2 sample 資料庫。

開啓 sample 資料庫而不指定使用者 ID 或通行碼；亦即，使用現行的使用者：

```
db.Open "SAMPLE"
```

建立記錄集：

```
Set adoPrimaryRS = New Recordset
```

使用 select 陳述式以填入記錄集：

```
adoPrimaryRS.Open "select EMPNO, LASTNAME, FIRSTNAME, MIDINIT, EDLEVEL, JOB
from EMPLOYEE Order by EMPNO", db
```

程式設計師可從這裡使用 ADO 方法存取資料，例如移至下一個記錄集：

```
adoPrimaryRS.MoveNext
```

刪除記錄集中的現行記錄：

```
adoPrimaryRS.Delete
```

同時程式設計師可執行下列命令存取個別欄位：

```
Dim Text1 as String
Text1 = adoPrimaryRS!LASTNAME
```

相關概念：

- *Application Development Guide: Programming Client Applications* 中的『Purpose of the IBM OLE DB Provider for DB2』

- *Application Development Guide: Programming Client Applications* 中的『Application Types Supported by the IBM OLE DB Provider for DB2』
- *Application Development Guide: Programming Client Applications* 中的『Connections to Data Sources with Visual Basic ADO Applications』
- *Application Development Guide: Programming Client Applications* 中的『OLE DB Services Automatically Enabled by IBM OLE DB Provider』
- *Application Development Guide: Programming Client Applications* 中的『Large Object Manipulation with the IBM OLE DB Provider』
- *Application Development Guide: Programming Client Applications* 中的『MTS and COM+ Distributed Transaction Support and the IBM OLE DB Provider』
- *Application Development Guide: Programming Client Applications* 中的『IBM OLE DB Provider Restrictions』

相關參照:

- *Application Development Guide: Programming Client Applications* 中的『IBM OLE DB Provider Support for OLE DB Components and Interfaces』
- *Application Development Guide: Programming Client Applications* 中的『IBM OLE DB Provider Support for OLE DB Properties』
- *Application Development Guide: Programming Client Applications* 中的『IBM OLE DB Provider Support for ADO Methods and Properties』
- 第 82 頁的『Visual Basic 範例』

利用 Visual Basic 建置 RDO 應用程式

「遠端資料物件 (RDO)」提供透過 ODBC 存取遠端資料來源的資訊模式。RDO 提供一組物件，使容易連接資料庫、執行查詢和儲存程序、操作結果，以及確定伺服器變更。它專門用來存取遠端 ODBC 關聯式資料原始檔，使您更容易使用 ODBC 而不需要複雜應用程式碼，而且是存取一個隨 ODBC 驅動程式出現的關聯式資料庫的主要方法。RDO 透過 Open Database Connectivity (ODBC) API 和驅動程式管理員執行一個薄的程式碼層 (thin code layer)，驅動程式管理員會使用最少的工作站資源來建立連線、建立結果集及游標，並執行複雜的程序。

DB2 在 `sqllib\samples\VB` 目錄中提供 Visual Basic RDO 範例程式。

程序:

若要對 Microsoft Visual Basic 使用 RDO，您必須建立 Visual Basic 專案的參照。請執行下列各項：

1. 從「專案」功能表選取「參照」

2. 勾選「Microsoft Remote Data Object <Version Number>」方框
3. 按一下「確定」。

其中 <version_number> 是現行 RDO 版本。

完整 Visual Basic 程式包含表格和其它圖形元素，您必須在 Visual Basic 環境中才能檢視它。以下是一些 Visual Basic 命令，這些命令作為 DB2 程式的一部份，該程式連接至 sample 資料庫，開啓一個記錄集（選取 EMPLOYEE 表格的全部直欄），然後在訊息視窗逐一顯示員工姓名：

```
Dim rdoEn As rdoEngine
Dim rdoEv As rdoEnvironment
Dim rdoCn As rdoConnection
Dim Cnct$
Dim rdoRS As rdoResultset
Dim SQLQueryDB As String
```

指定連線字串：

```
Cnct$ = "DSN=SAMPLE;UID=;PWD=;"
```

設定 RDO 環境：

```
Set rdoEn = rdoEngine
Set rdoEv = rdoEn.rdoEnvironments(0)
```

連接至資料庫：

```
Set rdoCn = rdoEv.OpenConnection("", , , Cnct$)
```

指定記錄集的 SELECT 陳述式：

```
SQLQueryDB = "SELECT * FROM EMPLOYEE"
```

開啓記錄集並執行查詢：

```
Set rdoRS = rdoCn.OpenResultset(SQLQueryDB)
```

當位置不在記錄集結尾時，顯示表格中 LASTNAME、FIRSTNAME 的「訊息框」，一次顯示一位員工：

```
While Not rdoRS.EOF
MsgBox rdoRS!LASTNAME &" , " &rdoRS!FIRSTNAME
```

移至記錄集的下一列：

```
rdoRS.MoveNext
Wend
```

結束程式：

```
rdoRS.Close  
rdoCn.Close  
rdoEv.Close
```

相關參照:

- 第 82 頁的『Visual Basic 範例』

利用 Visual Basic 的物件連結和內嵌 (OLE) 自動化

您可使用任何語言執行 OLE 自動化 UDF 和儲存程序 (因為 OLE 與語言無關)。做法為顯示 OLE 自動化伺服器的方法，然後以 UDF 向 DB2 登錄這些方法。支援開發 OLE 自動化伺服器的應用程式開發環境包含下列項目的某些版本：Microsoft® Visual Basic、Microsoft Visual C++、Microsoft Visual J++、Microsoft FoxPro、Borland Delphi、Powersoft PowerBuilder 和 Micro Focus COBOL。另外，可以透過 OLE 自動化來存取對於 OLE 而言是適當包裝的 Java™ bean 物件 (例如 Microsoft Visual J++)。

您必須參考適當應用程式開發環境的文件，以取得關於開發 OLE 自動化伺服器的進一步資訊。

OLE 自動化 UDF 及儲存程序

Microsoft Visual Basic 支援建立 OLE 自動化伺服器。透過將某類別模組新增至 Visual Basic 專案，在 Visual Basic 中建立新型物件。將公用次程序新增至此類別模組來建立方法。可以將這些公用程序登錄到 DB2® 為 OLE 自動化 UDF 及儲存程序。如需關於建立和建置 OLE 伺服器的進一步資訊，請參閱 Microsoft Visual Basic 手冊 *Creating OLE Servers, Microsoft Corporation, 1995*，以及參閱 Microsoft Visual Basic 提供的 OLE 範例。

DB2 在 Microsoft Visual Basic 中提供 OLE 自動化 UDF 及儲存程序自我包含的範例，這些範例位於目錄 `sqllib\samples\ole\msvb` 中。若需建置及執行 OLE 自動化 UDF 及儲存程序範例的相關資訊，請參閱 `sqllib\samples\ole` 中的 README 檔。

相關參照:

- 第 84 頁的『物件連結和內嵌 (OLE) 範例』

Microsoft Visual C++

本章節討論如何使用 ActiveX Data Object (ADO)、物件連結和內嵌 (OLE) 以及內嵌式 SQL 與 DB2 API 建置應用程式。

關於 DB2 CLI 應用程式及常式的建置資訊，請參閱 *CLI Guide and Reference*。

利用 Visual C++ 建置 ADO 應用程式

「ActiveX 資料物件 (ADO)」可讓您透過 OLE DB 提供者，撰寫應用程式來存取及操作資料庫伺服器內的資料。ADO 的主要優點是快速、容易使用、記憶體超負荷量低以及少用磁碟。

DB2 在 `sql1lib\samples\VC` 目錄中提供 Visual C++ ADO 範例程式。

程序:

您可以使用這兩個遵守 ODBC 提供者之一：

- DB2 的 IBM OLE DB 提供者
- ODBC 的 Microsoft OLE DB 提供者

使用 DB2 的 IBM OLE DB 提供者

Windows 作業系統上的 DB2 版本 8.1 從屬站將選用性地安裝 IBMDADB2 (DB2 的遵守 IBM OLE DB 2.0 提供者)。提供者會提供介面，供想要存取 DB2 資料庫中資料的消費者使用。DB2 的 IBM OLE DB 提供者支援下列 ADO 應用程式類型：

- Microsoft Active Server Pages (ASP)
- Microsoft Visual Studio C++ 及 Visual Basic 應用程式
- Microsoft Visual Interdev

若需有關這些應用程式類型的詳細資訊，請參照 ADO 文件。

使用 ODBC 的 Microsoft OLE DB 提供者

執行下列變更後，可以像一般 C++ 程式一樣地編譯那些使用 Microsoft OLE DB 提供者及 Visual C++ 的 DB2 ADO 程式。

若要使 C++ 原始程式當作 ADO 程式來執行，您可將下列匯入陳述式置於原始程式檔頂端：

```
#import "C:\program files\common files\system\ado\msado<VERSION NUMBER>.dll" \  
no_namespace \  
rename( "EOF", "adoEOF")
```

其中 <VERSION NUMBER> 是 ADO 檔案庫的版本號碼。

編譯此程式後，使用者必須驗證 msado<VERSION NUMBER>.dll 是否在指定的路徑。另一個驗證方法是將 C:\program files\common files\system\ado 新增至環境變數 LIBPATH，然後在來源檔中使用這個較短的匯入陳述式：

```
#import <msado<VERSION NUMBER>.dll> \  
no_namespace \  
rename( "EOF", "adoEOF")
```

這是用於 DB2 範例程式 BLOBAccess.dsp 的方法。

有了這個 IMPORT 陳述式，DB2 程式就具有 ADO 檔案庫的存取權。您現在可以像編譯其它程式一樣地編譯 Visual C++ 程式。若您也使用另一個程式設計介面，例如 DB2 API 或 DB2 CLI，請參閱本章適當的節次以瞭解關於建置程式的其它資訊。

相關概念:

- *Application Development Guide: Programming Client Applications* 中的『Purpose of the IBM OLE DB Provider for DB2』
- *Application Development Guide: Programming Client Applications* 中的『Application Types Supported by the IBM OLE DB Provider for DB2』
- *Application Development Guide: Programming Client Applications* 中的『Compilation and Linking of C/C++ Applications and the IBM OLE DB Provider』
- *Application Development Guide: Programming Client Applications* 中的『Connections to Data Sources in C/C++ Applications using the IBM OLE DB Provider』
- *Application Development Guide: Programming Client Applications* 中的『OLE DB Services Automatically Enabled by IBM OLE DB Provider』
- *Application Development Guide: Programming Client Applications* 中的『Large Object Manipulation with the IBM OLE DB Provider』
- *Application Development Guide: Programming Client Applications* 中的『IBM OLE DB Provider Restrictions』

相關參照:

- *Application Development Guide: Programming Client Applications* 中的『Data Type Mappings between DB2 and OLE DB』
- *Application Development Guide: Programming Client Applications* 中的『Data Conversion for Setting Data from OLE DB Types to DB2 Types』
- *Application Development Guide: Programming Client Applications* 中的『Data Conversion for Setting Data from DB2 Types to OLE DB Types』

- *Application Development Guide: Programming Client Applications* 中的『IBM OLE DB Provider Support for OLE DB Components and Interfaces』
- *Application Development Guide: Programming Client Applications* 中的『IBM OLE DB Provider Support for OLE DB Properties』
- *Application Development Guide: Programming Client Applications* 中的『IBM OLE DB Provider Support for ADO Methods and Properties』
- 第 83 頁的『Visual C++ 範例』

利用 Visual C++ 的物件連結和內嵌 (OLE) 自動化

您可使用任何語言執行 OLE 自動化 UDF 和儲存程序 (因為 OLE 與語言無關)。做法為顯示 OLE 自動化伺服器的方法，然後以 UDF 向 DB2 登錄這些方法。支援開發 OLE 自動化伺服器的應用程式開發環境包含下列項目的某些版本：Microsoft® Visual Basic、Microsoft Visual C++、Microsoft Visual J++、Microsoft FoxPro、Borland Delphi、Powersoft PowerBuilder 和 Micro Focus COBOL。另外，可以透過 OLE 自動化來存取對於 OLE 而言是適當包裝的 Java™ bean 物件 (例如 Microsoft Visual J++)。

您必須參考適當應用程式開發環境的文件，以取得關於開發 OLE 自動化伺服器的進一步資訊。

OLE 自動化 UDF 及儲存程序

Microsoft Visual C++ 支援建立 OLE 自動化伺服器。可以使用 Microsoft Foundation Classes 和 Microsoft Foundation Class 應用程式精靈，伺服器或執行為 Win32 應用程式。伺服器可以是 DLL 或 EXE。關於進一步資訊，請參閱 Microsoft Visual C++ 文件和 Microsoft Visual C++ 提供的 OLE 範例。

DB2® 提供在 Microsoft Visual C++ 中 OLE 自動化 UDF 及儲存程序的自我包含的範例，這些範例是位於目錄 `sqllib\samples\ole\msvc` 中。若需建置及執行 OLE 自動化 UDF 及儲存程序範例的相關資訊，請參閱 `sqllib\samples\ole` 中的 README 檔。

相關參照:

- 第 84 頁的『物件連結和內嵌 (OLE) 範例』

在 Windows 上建置 C/C++ 應用程式

DB2 提供批次檔，來編譯及鏈結 DB2 API 及內嵌式 SQL C/C++ 程式。這些都位於 `sqllib\samples\c` 及 `sqllib\samples\cpp` 目錄，而且這些目錄中還有可利用這些檔案來建置的範例程式。

批次檔 `bldapp.bat` 含有要建置 DB2 API 及內嵌式 SQL 程式的命令。它最多使用 4 個參數，在批次檔內以變數 `%1`、`%2`、`%3` 及 `%4` 代表。

第一個參數 `%1`，指定您的來源檔的名稱。對沒有內嵌式 SQL 的程式而言，這是唯一必要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個額外參數：第二個參數 `%2`，指定您要連接的資料庫名稱；第三個參數 `%3` 指定資料庫的使用者 ID，而 `%4` 指定通行碼。

對於內嵌式 SQL 程式，`bldapp` 會傳送參數到前置編譯及連結檔案 `embprep.bat`。如果未提供任何資料庫名稱，將使用預設 `sample` 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

下列範例告訴您如何建置及執行 DB2 API 及內嵌式 SQL 應用程式。

若要從來源檔 `cli_info.c` (位於 `sqllib\samples\c`)，或是從來源檔 `cli_info.cxx` (位於 `sqllib\samples\cpp`) 建置 DB2 API 非內嵌式 SQL 範例程式 `cli_info`，請輸入：

```
bldapp cli_info
```

結果是可執行檔 `cli_info.exe`。您可在命令行輸入下列可執行檔名稱（無副檔名）來執行可執行檔：

```
cli_info
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從 C 來源檔 `tbmod.sqc` (位於 `sqllib\samples\c`) 或從 C++ 來源檔 `tbmod.sqx` (位於 `sqllib\samples\cpp`) 建置內嵌式 SQL 應用程式 `tbmod`：

1. 如果連接同一案例上的 `sample` 資料庫，請輸入：

```
bldapp tbmod
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp tbmod database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp tbmod database userid password
```

結果是可執行檔 `tbmod.exe`。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 `sample` 資料庫，則只要輸入可執行檔檔名：

```
tbmod
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
tbmod database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
tbmod database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 288 頁的『Windows C/C++ 應用程式編譯及鏈結選項』
- 第 65 頁的『C/C++ 範例』

相關範例:

- 『`bldapp.bat` -- Builds C applications on Windows』
- 『`cli_info.c` -- Set and get information at the client level (C)』
- 『`embprep.bat` -- Prep and binds a C/C++ or Micro Focus COBOL embedded SQL program on Windows』
- 『`tbmod.sqc` -- How to modify table data (C)』
- 『`bldapp.bat` -- Builds C++ applications on Windows』
- 『`cli_info.C` -- Set and get information at the client level (C++)』
- 『`tbmod.sqC` -- How to modify table data (C++)』

C/C++ 應用程式的批次檔

```
@echo off
rem BATCH FILE: bldapp.bat
rem Builds Windows Microsoft Visual C++ applications
rem Usage: bldapp prog_name [ db_name [ userid password ] ]

rem Default compiler is set to Microsoft Visual C++
rem To use a different compiler, comment out "set BLDCOMP=c1"
rem and uncomment "set BLDCOMP=ic1" or "set BLDCOMP=ec1"
rem Microsoft C/C++ Compiler
set BLDCOMP=c1

rem Intel C++ Compiler for 32-bit applications
rem set BLDCOMP=ic1
```

```

rem Intel C++ Compiler for Itanium 64-bit applications
rem set BLDCOMP=ec1

if exist "%1.sqx" goto embedded
if exist "%1.sqc" goto embedded
goto non_embedded

:embedded
rem Precompile and bind the program.
call embprep %1 %2 %3 %4
rem Compile the program.
if exist "%1.cxx" goto cpp_emb
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.c utilemb.c
goto link_embedded
:cpp_emb
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.cxx utilemb.cxx
rem Link the program.
:link_embedded
link -debug -out:%1.exe %1.obj utilemb.obj db2api.lib
goto exit

:non_embedded
rem Compile the program.
if exist "%1.cxx" goto cpp_non
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.c utilapi.c
goto link_non_embedded
:cpp_non
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.cxx utilapi.cxx
rem Link the program.
:link_non_embedded
link -debug -out:%1.exe %1.obj utilapi.obj db2api.lib
:exit
@echo on

```

Windows C/C++ 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Microsoft Visual C++ 編譯器建置 C/C++ 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp.bat 批次檔中所示範一般。

bldapp 的編譯及鏈結選項

編譯選項：

%BLDCOMP%

編譯器的變數。預設值是 `cl` (Microsoft Visual C++ 編譯器)。它也可以設為 `icl` (32 位元應用程式的 Intel C++ 編譯器) 或 `ec1` (Itanium 64 位元應用程式的 Intel C++ Compiler)。

-Zi 啟用除錯資訊

-Od 停用最佳化。使用一個關閉最佳化的除錯器比較容易。

-c 僅執行編譯；沒有任何鏈結。批次檔有不同的編譯及鏈結步驟。

-W2 輸出警告、錯誤以及嚴重和無法復原的錯誤訊息。

-DWIN32

Windows 作業系統需要的編譯器選項。

鏈結選項：

link 使用鏈結器來鏈結。

-debug 包括除錯資訊。

-out:%1.exe
指定檔名。

%1.obj 包括目的檔

utilemb.obj

如果是內嵌式 SQL 程式，則有內嵌式 SQL 公用程式的目的檔，可執行錯誤檢查。

utilapi.obj

如果不是內嵌式 SQL 程式，則有 DB2 API 公用程式目的檔，可執行錯誤檢查。

db2api.lib

與 DB2 檔案庫鏈結。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 286 頁的『在 Windows 上建置 C/C++ 應用程式』

相關範例：

- 『bldapp.bat -- Builds C applications on Windows』
- 『bldapp.bat -- Builds C++ applications on Windows』

在 Windows 上建置 C/C++ 常式

DB2 提供批次檔，以 C 及 C++ 來編譯及鏈結 DB2 API 及內嵌式 SQL 程式。這些都位於 `sqllib\samples\c` 及 `sqllib\samples\cpp` 目錄，而且這些目錄中還有可利用這些檔案來建置的範例程式。

批次檔 `bldrtn.bat` 含有要建置內嵌式 SQL 常式 (儲存程序及使用者定義的函數) 的命令。批次檔會在伺服器上建置 DLL。它最多使用 2 個參數，在批次檔內以變數 `%1` 及 `%2` 代表。

第一個參數 `%1`，指定您的來源檔的名稱。批次檔會使用來源檔名稱作為 DLL 名稱。第二個參數 `%2`，指定您想要與其連接的資料庫的名稱。既然 DLL 必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

只需要第一個參數 (來源檔名稱)。資料庫名稱是可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

程序:

下列範例告訴您如何利用下列建置常式 DLL：

- 儲存程序
- 非內嵌式 SQL 使用者定義的函數 (UDF)
- 內嵌式 SQL 使用者定義的函數 (UDF)

儲存程序 DLL

若要從 C 來源檔 `spserver.sqc` 或 C++ 來源檔 `spserver.sqx` 建置 `spserver` DLL，請：

1. 輸入批次檔名稱及程式名稱：

```
bldrtn spserver
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn spserver database
```

批次檔會使用與範例程式在相同目錄中的模組定義檔 `spserver.def` 來建置 DLL。批次檔會將 DLL `spserver.dll` 複製到路徑 `sqllib\function` 中的伺服器。

2. 接著，在伺服器上執行 `spcat` Script，將常式編目：

```
spcat
```

這個 Script 將連接到範例資料庫、將取消常式編目 (若之前已呼叫 `spdrop.db2` 將它們編目的話)，然後呼叫 `spcreate.db2` 來將它們編目，最後切斷資料庫連線。您也可以個別地呼叫 `spdrop.db2` 及 `spcreate.db2` Script。

3. 然後，停止資料庫後再重新啟動，以辨識新的 DLL。必要時，請設定 DLL 的檔案模式，以便 DB2 案例可以存取它。

一旦您建置了 DLL `spserver`，您就可以建置呼叫它的從屬站應用程式 `spclient`。

您可以使用批次檔 `bldapp.bat` 建置 `spclient`。

若要呼叫 DLL，請輸入下列命令來執行範例從屬站應用程式：

```
spclient database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample`，或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式可在伺服器資料庫上存取 DLL `spserver`，並執行許多常式。輸出會傳回從屬站應用程式。

非內嵌式 SQL UDF DLL

若要從來源檔 `udfsrv.c` 建置使用者定義的函數 `udfsrv`，請輸入：

```
bldrtn udfsrv
```

批次檔會使用與範例程式檔案位在相同目錄中的模組定義檔 `udfsrv.def`，建置使用者定義的函數 DLL。批次檔會將使用者定義的函數 DLL `udfsrv.dll` 複製到路徑 `sqllib\function` 中的伺服器。

一旦您建置了 `udfsrv`，您就可以建置呼叫它的從屬站應用程式 `udfcli`。提供 DB2 CLI 以及本程式的內嵌式 SQL C 以及 C++ 版本。

您可以使用批次檔 `bldapp`，從 `udfcli.c` 來源檔 (位於 `sqllib\samples\cli`) 建置 DB2 CLI `udfcli` 程式。

您可以使用批次檔 `bldapp`，從 `udfcli.sqc` 來源檔 (位於 `sqllib\samples\c`) 建置內嵌式 SQL C `udfcli` 程式。

您可以使用批次檔 `bldapp`，從 `udfcli.sqx` 來源檔 (位於 `sqllib\samples\cpp`) 建置內嵌式 SQL C++ `udfcli` 程式。

若要執行 UDF，請輸入下列命令：

```
udfcli
```

呼叫應用程式會從 `udfsrv` DLL 呼叫 `ScalarUDF` 函數。

內嵌式 SQL UDF DLL

若要從 C 來源檔 `udfemsrv.sqc` (位於 `sqllib\samples\c`)，或是從 C++ 來源檔 `udfemsrv.sqx` (位於 `sqllib\samples\cpp`)，建置內嵌式 SQL 使用者定義的函數檔案庫 `udfemsrv`，請輸入：

```
bldrtn udfemsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn udfemsrv database
```

批次檔會使用與範例程式位在相同目錄中的模組定義檔 `udfemsrv.def`，建置使用者定義的函數 DLL。批次檔會將使用者定義的函數 DLL `udfemsrv.dll` 複製到路徑 `sqllib\function` 中的伺服器。

一旦您建置了 `udfemsrv`，您就可以建置呼叫它的從屬站應用程式 `udfemcli`。您可以使用批次檔 `bldapp`，從 C 來源檔 `udfemcli.sqc` (位於 `sqllib\samples\c`)，或是從 C++ 來源檔 `udfemcli.sqx` (位於 `sqllib\samples\cpp`) 建置 `udfemcli`。

若要執行 UDF，請輸入下列命令：

```
udfemcli
```

呼叫應用程式會呼叫 `udfemsrv` DLL 中的 UDF。

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 294 頁的『Windows C/C++ 常式編譯及鏈結選項』
- 第 65 頁的『C/C++ 範例』

相關範例:

- 『bldrtn.bat -- Builds C routines (stored procedures and UDFs) on Windows』
- 『embprep.bat -- Prep and binds a C/C++ or Micro Focus COBOL embedded SQL program on Windows』
- 『spcat -- To catalog C stored procedures on UNIX』
- 『spclient.sqc -- Call various stored procedures (C)』
- 『spcreate.db2 -- How to catalog the stored procedures contained in spserver.sqc』
- 『spdrop.db2 -- How to uncatalog the stored procedures contained in spserver.sqc』
- 『spserver.sqc -- A variety of types of stored procedures (C)』
- 『udfcli.sqc -- Call a variety of types of user-defined functions (C)』
- 『udfemcli.sqc -- Call a variety of types of embedded SQL user-defined functions. (C)』
- 『udfemsrv.sqc -- Call a variety of types of embedded SQL user-defined functions. (C)』
- 『udfsrv.c -- Call a variety of types of user-defined functions (C)』
- 『bldrtn.bat -- Builds C++ routines (stored procedures and UDFs) on Windows』
- 『spclient.sqC -- Call various stored procedures (C++)』
- 『spserver.sqC -- A variety of types of stored procedures (C++)』
- 『udfcli.sqC -- Call a variety of types of user-defined functions (C++)』
- 『udfemcli.sqC -- Call a variety of types of embedded SQL user-defined functions. (C++)』
- 『udfemsrv.sqC -- Call a variety of types of embedded SQL user-defined functions. (C++)』
- 『udfsrv.C -- Call a variety of types of user-defined functions (C++)』

C/C++ 常式的批次檔

```

@echo off
rem BATCH FILE: bldrtn.bat
rem Builds Windows Microsoft Visual C++ routines (stored procedures and UDFs)
rem Usage: bldrtn prog_name [ db_name ]

rem Default compiler is set to Microsoft Visual C++
rem To use a different compiler, comment out "set BLDCOMP=c1"
rem and uncomment "set BLDCOMP=ic1" or "set BLDCOMP=ec1"
rem Microsoft C/C++ Compiler
set BLDCOMP=c1

rem Intel C++ Compiler for 32-bit applications

```

```

rem set BLDCOMP=icl

rem Intel C++ Compiler for Itanium 64-bit applications
rem set BLDCOMP=ec1

if exist "%1.sqc" goto embedded
if exist "%1.sqx" goto embedded
goto compile

:embedded
rem Precompile and bind the program.
call embprep %1 %2

:compile
rem Compile the program.
if exist "%1.cxx" goto cpp
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.c
goto link_step
:cpp
%BLDCOMP% -Zi -Od -c -W2 -DWIN32 %1.cxx

:link_step
rem Link the program.
link -debug -out:%1.dll -dll %1.obj db2api.lib -def:%1.def

rem Copy the routine DLL to the 'function' directory
copy %1.dll "%DB2PATH%\function"
@echo on

```

Windows C/C++ 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Microsoft Visual C++ 編譯器建置 C/C++ 常式 (儲存程序及使用者定義的函數)，如同 bldrtn.bat 批次檔中所示範一般。

bldrtn 的編譯及鏈結選項

編譯選項：

%BLDCOMP%

編譯器的變數。預設值是 `cl` (Microsoft Visual C++ 編譯器)。它也可以設為 `icl` (32 位元應用程式的 Intel C++ 編譯器) 或 `ec1` (Intel C++ Compiler for Itanium 64 位元應用程式)。

-Zi 啓用除錯資訊

-Od 停用最佳化。

-c 僅執行編譯；沒有任何鏈結。本書假定編譯及鏈結是分開的步驟。

-W2 輸出警告、錯誤以及嚴重和無法復原的錯誤訊息。

-DWIN32

Windows 作業系統需要的編譯器選項。

鏈結選項：

link 使用鏈結器來鏈結。

-debug 包括除錯資訊。

-out:%1.dll

建置 .DLL 檔。

%1.obj 包括目的檔。

db2api.lib

與 DB2 檔案庫鏈結。

-def:%1.def

模組定義檔。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 290 頁的『在 Windows 上建置 C/C++ 常式』

相關範例：

- 『bldrtn.bat -- Builds C routines (stored procedures and UDFs) on Windows』
- 『bldrtn.bat -- Builds C++ routines (stored procedures and UDFs) on Windows』

在 Windows 上架構 IBM COBOL 編譯器

若開發含有內嵌式 SQL 和 DB2 API 呼叫的應用程式，而且使用 IBM VisualAge COBOL 編譯器，請記住以下幾點：

程序：

- 當您使用命令行處理器命令 `db2 prep`，來前置編譯您的應用程式時，請使用 `target ibmcob` 選項。
- 請勿在您的來源檔中使用 Tab 鍵字元。
- 在來源檔中使用 `PROCESS` 和 `CBL` 關鍵字以設定編譯選項。只在直欄 8 到 72 中加入這些關鍵字。
- 如果您的應用程式僅內嵌式 SQL，且沒有任何 DB2 API 呼叫，則您不需要使用 `pgmname(mixed)` 編譯選項。如果您使用 DB2 API 呼叫，則您必須使用 `pgmname(mixed)` 編譯選項。
- 如果您使用的是 IBM VisualAge COBOL 編譯器的「System/390 主電腦資料類型支援」特性，則您應用程式的 DB2 併入檔會位於下列目錄中：

```
%DB2PATH%\include\cobol_i
```

如果您是使用隨附的批次檔來建置 DB2 範例程式，則在批次檔中所指定的併入檔路徑必須變更為指向 `cobol_i` 目錄，而非指向 `cobol_a` 目錄。

如果您「不」是使用 IBM VisualAge COBOL 編譯器的「System/390 主電腦資料類型支援」特性，或您使用的是此編譯器的舊版本，則您應用程式的 DB2 併入檔會位於下列目錄中：

```
%DB2PATH%\include\cobol_a
```

`cobol_a` 目錄是預設值。

- 指定 `COPY` 檔名，來併入 `.cbl` 副檔名，方式如下：

```
COPY "sql.cbl".
```

相關作業：

- 第 297 頁的『在 Windows 上建置 IBM COBOL 應用程式』
- 第 299 頁的『在 Windows 上建置 IBM COBOL 常式』

在 Windows 上建置 IBM COBOL 應用程式

DB2 提供批次檔，來編譯及鏈結 DB2 API 及內嵌式 SQL 程式。這些都位於 `sqllib\samples\cobol` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

批次檔 `bldapp.bat` 含有要建置 DB2 應用程式的命令。它最多使用 4 個參數，在批次檔內以變數 `%1`、`%2`、`%3` 及 `%4` 代表。

第一個參數 `%1`，指定您的來源檔的名稱。對沒有內嵌式 SQL 的程式而言，這是唯一必要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 `%2`，指定您要連接的資料庫名稱；第三個參數 `%3` 指定資料庫的使用者 ID，而 `%4` 指定通行碼。

對於內嵌式 SQL 程式，`bldapp` 會傳送參數到前置編譯及連結檔案 `embprep.bat`。如果未提供任何資料庫名稱，將使用預設 `sample` 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

下列範例告訴您如何建置及執行 DB2 API 及內嵌式 SQL 應用程式。

若要從來源檔 `client.cbl` 建置非內嵌式 SQL 範例程式 `client`，請輸入：

```
bldapp client
```

結果產生可執行檔 `client.exe`。您可以輸入下列可執行檔名稱 (不含副檔名)，對 `sample` 資料庫執行可執行檔：

```
client
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 `updat.sqb` 建置內嵌式 SQL 應用程式 `updat`：

1. 如果連接同一案例上的 `sample` 資料庫，請輸入：

```
bldapp updat
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp updat database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp updat database userid password
```

結果產生是可執行檔 `updat`。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例上的 `sample` 資料庫，則只要輸入可執行檔檔名：
`updat`
2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：
`updat database`
3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：
`updat database userid password`

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 299 頁的『Windows IBM COBOL 應用程式編譯及鏈結選項』
- 第 76 頁的『COBOL 範例』

相關範例:

- 『`bldapp.bat` -- Builds Windows VisualAge COBOL applications』
- 『`client.cbl` -- How to set and query a client (IBM COBOL)』
- 『`embprep.bat` -- To prep and bind a COBOL embedded SQL program on Windows』
- 『`updat.sqb` -- How to update, delete and insert table data (IBM COBOL)』

IBM COBOL 應用程式的批次檔

```
@echo off
rem BATCH FILE: bldapp.bat
rem Builds Windows VisualAge COBOL applications
rem Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

rem If an embedded SQL program, precompile and bind it.
if not exist "%1.sqb" goto compile_step
call embprep %1 %2 %3 %4

:compile_step
rem Compile the error-checking utility.
cob2 -qpgmname(mixed) -c -qlib -I"%DB2PATH%\include\cobol_a" checkerr.cbl

rem Compile the program.
cob2 -qpgmname(mixed) -c -qlib -I"%DB2PATH%\include\cobol_a" %1.cbl

rem Link the program.
cob2 %1.obj checkerr.obj db2api.lib
@echo on
```

Windows IBM COBOL 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 IBM VisualAge COBOL 編譯器在 Windows 上建置 COBOL 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp.bat 批次檔中所示範一般。

bldapp 的編譯及鏈結選項	
編譯選項：	
cob2	IBM VisualAge COBOL 編譯器。
-pgmname(mixed)	指示編譯器允許 CALL 具有混合字體名稱的檔案庫進入點。
-c	僅執行編譯；沒有任何鏈結。本書假定編譯及鏈結是分開的步驟。
-qlib	指示編譯器處理 COPY 陳述式。
-I路徑	指定 DB2 併入檔的位置。例如：-I"%DB2PATH%\include\cobol_a"。
checkerr.cb1	編譯錯誤檢查公用程式。
鏈結選項：	
cob2	使用編譯器作為鏈結器的前端
%1.obj	包括程式目的檔。
checkerr.obj	包括錯誤檢查公用程式目的檔。
db2api.lib	與 DB2 檔案庫鏈結。
請參閱您的編譯器文件，取得其他編譯器選項。	

相關作業：

- 第 297 頁的『在 Windows 上建置 IBM COBOL 應用程式』

相關範例：

- 『bldapp.bat -- Builds Windows VisualAge COBOL applications』

在 Windows 上建置 IBM COBOL 常式

DB2 提供批次檔，以 IBM COBOL 來編譯及鏈結 DB2 API 及內嵌式 SQL 程式。這些都位於 sqllib\samples\cobol 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

批次檔 bldrtn.bat 含有要建置內嵌式 SQL 常式 (儲存程序) 的命令。批次檔將常式編譯成伺服器上的 DLL。它最多使用 2 個參數，在批次檔內以變數 %1 及 %2 代表。

第一個參數 %1，指定您的來源檔的名稱。批次檔會使用來源檔名稱 %1 作為 DLL 名稱。第二個參數 %2，指定您想要與其連接的資料庫的名稱。既然儲存程序必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

只有第一個參數是必要的，即來源檔名稱。資料庫名稱是可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

程序:

當連接到範例資料庫時，若要從來源檔 `outsrv.sqb` 建置範例程式 `outsrv`，請輸入：

```
bldrtn outsrv
```

如果連接的是另一個資料庫，亦請併入資料庫名稱：

```
bldrtn outsrv database
```

批次檔會將 DLL 複製到路徑 `sqllib/function` 中的伺服器。

一旦建置了 DLL `outsrv`，您就可以建置從屬站應用程式 `outcli`，呼叫 DLL 內的常式 (與 DLL 同名)。您可以使用批次檔 `bldapp.bat`，建置 `outcli`。

若要呼叫 `outsrv` 常式，請輸入下列命令以執行範例從屬站應用程式：

```
outcli database userid password
```

其中

database

為您想要與其連接的資料庫的名稱。此名稱可以是 `sample` 或它的遠端別名或其它名稱。

userid 為有效的使用者 ID。

password

為使用者 ID 的有效通行碼。

從屬站應用程式將存取 DLL `outsrv`，並在伺服器資料庫上執行名稱相同的常式，然後將輸出傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 301 頁的『Windows IBM COBOL 常式編譯及鏈結選項』
- 第 76 頁的『COBOL 範例』

相關範例:

- 『bldrtn.bat -- Builds Windows VisualAge COBOL routines (stored procedures)』
- 『embprep.bat -- To prep and bind a COBOL embedded SQL program on Windows』
- 『outcli.sqb -- Call stored procedures using the SQLDA structure (IBM COBOL)』
- 『outsrv.sqb -- Demonstrates stored procedures using the SQLDA structure (IBM COBOL)』

IBM COBOL 常式的批次檔

```
@echo off
rem BATCH FILE: bldrtn.bat
rem Builds Windows VisualAge COBOL routines (stored procedures)
rem Usage: bldrtn <prog_name> [ <db_name> ]

rem Precompile and bind the program.
call embprep %1 %2

rem Compile the stored procedure.
cob2 -qpgmname(mixed) -c -qlib -I"%DB2PATH%\include\cobol_a" %1.cb1

rem Link the stored procedure and create a shared library.
ilib /nol /gi:%1 %1.obj
ilink /free /nol /dll db2api.lib %1.exp %1.obj iwzrwin3.obj

rem Copy stored procedure to the %DB2PATH%\function directory.
copy %1.dll "%DB2PATH%\function"
@echo on
```

Windows IBM COBOL 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 IBM VisualAge COBOL 編譯器建置 COBOL 常式 (儲存程序及使用者定義的函數)，如同 bldrtn.bat 批次檔中所示範一般。

bldrtn 的編譯及鏈結選項	
編譯選項：	
cob2	IBM VisualAge COBOL 編譯器。
-qpgmname(mixed)	指示編譯器允許 CALL 具有混合字體名稱的檔案庫進入點。
-c	僅執行編譯；沒有任何鏈結。此批次檔有不同的編譯及鏈結步驟。
-qlib	指示編譯器處理 COPY 陳述式。
-I路徑	指定 DB2 併入檔的位置。例如：-I"%DB2PATH%\include\cobol_a".

bldrtn 的編譯及鏈結選項

鏈結選項：

ilink 使用 IBM VisualAge COBOL 鏈結器。

/free 自由格式。

/no1 沒有商標。

/dll 使用原始程式名稱來建立 DLL。

db2api.lib

與 DB2 檔案庫鏈結。

%1.exp 併入匯出檔。

%1.obj 包括程式目的檔。

iwzrwin3.obj

併入 IBM VisualAge COBOL 提供的目的檔。

請參閱您的編譯器文件，取得其他編譯器選項。

相關作業：

- 第 299 頁的『在 Windows 上建置 IBM COBOL 常式』

相關範例：

- 『bldrtn.bat -- Builds Windows VisualAge COBOL routines (stored procedures)』

Micro Focus COBOL

在 Windows 上架構 Micro Focus COBOL 編譯器

若開發含有內嵌式 SQL 和 DB2 API 呼叫的應用程式，而且使用 Micro Focus 編譯器，請記住以下幾點：

程序：

- 當您在使用命令行處理器命令 `db2 prep` 前置編譯您的應用程式時，請使用 `target mfcob` 選項。
- 確定 LIB 環境變數指向 `%DB2PATH%\lib`，例如：

```
set LIB="%DB2PATH%\lib;%LIB%"
```
- Micro Focus COBOL 的 DB2 COPY 位於 `%DB2PATH%\include\cobol_mf`。設定 COBCPY 環境變數來包括目錄，例如：

```
set COBCPY="%DB2PATH%\include\cobol_mf;%COBCPY%"
```

您必須使用呼叫慣例 74，對所有 DB2 應用程式設計介面進行呼叫。DB2 COBOL 前置編譯器會自動將 CALL-CONVENTION 子句插入 SPECIAL-NAMES 段落中。若 SPECIAL-NAMES 段落不存在，DB2 COBOL 會另外建立，如下所示：

```
Identification Division
Program-ID. "static".
special-names.
    call-convention 74 is DB2API.
```

同時，若有呼叫 DB2 API，前置編譯器會在 "call" 關鍵字後面自動加入符號 DB2 API，用來識別呼叫慣例。每當前置編譯器從內嵌式 SQL 陳述式中建立 DB2 API 執行呼叫，就會進行這種動作。

若 DB2 API 呼叫出現在未經過前置編譯的應用程式中，您應該在應用程式中自行建立一個如同上述的 SPECIAL-NAMES 段落。若直接呼叫 DB2 API，則需要自行在 "call" 關鍵字後面加上 DB2API 符號。

相關作業:

- 第 303 頁的『在 Windows 上建置 Micro Focus COBOL 應用程式』
- 第 306 頁的『在 Windows 上建置 Micro Focus COBOL 常式』

在 Windows 上建置 Micro Focus COBOL 應用程式

DB2 提供批次檔，來編譯及鏈結 DB2 API 及內嵌式 SQL 程式。這些都位於 sqllib\samples\cobol_mf 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

批次檔 bldapp.bat 含有要建置 DB2 應用程式的命令。它最多使用 4 個參數，在批次檔內以變數 %1、%2、%3 及 %4 代表。

第一個參數 %1，指定您的來源檔的名稱。對沒有內嵌式 SQL 的程式而言，這是唯一必要的參數。建置內嵌式 SQL 程式需要連線到資料庫，所以也提供您三個選用參數：第二個參數 %2，指定您要連接的資料庫名稱；第三個參數 %3 指定資料庫的使用者 ID，而 %4 指定通行碼。

對於內嵌式 SQL 程式，bldapp 會傳遞參數到前置編譯及連結批次檔 embprep.bat。如果未提供任何資料庫名稱，將使用預設 sample 資料庫。如果建置程式的案例與資料庫所在位置的案例不同時，才需要使用者 ID 及通行碼參數。

程序:

下列範例告訴您如何建置及執行 DB2 API 及內嵌式 SQL 應用程式。

若要從來源檔 client.cbl 建置非內嵌式 SQL 範例程式 client，請輸入：

```
bldapp client
```

結果產生可執行檔 `client.exe`。您可以輸入下列可執行檔名稱 (不含副檔名)，對 `sample` 資料庫執行可執行檔：

```
client
```

建置及執行內嵌式 SQL 應用程式

有三種方式可以用來從來源檔 `updat.sqb` 建置內嵌式 SQL 應用程式 `updat`：

1. 如果連接同一案例上的 `sample` 資料庫，請輸入：

```
bldapp updat
```

2. 如果連接同一案例上的不同資料庫，亦請輸入資料庫名稱：

```
bldapp updat database
```

3. 如果連接另一案例上的資料庫，亦請輸入資料庫案例的使用者 ID 及通行碼：

```
bldapp updat database userid password
```

結果產生可執行檔 `updat.exe`。

有三種方式可以執行此內嵌式 SQL 應用程式：

1. 如果存取同一案例中的 `sample` 資料庫，則只要輸入可執行檔名稱 (不需要副檔名)：

```
updat
```

2. 如果存取同一案例上的另一個資料庫，請輸入可執行檔名稱及資料庫名稱：

```
updat database
```

3. 如果存取另一案例上的資料庫，請輸入可執行檔名稱、資料庫名稱及資料庫案例的使用者 ID 及通行碼：

```
updat database userid password
```

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 305 頁的『Windows Micro COBOL 應用程式編譯及鏈結選項』
- 第 76 頁的『COBOL 範例』

相關範例:

- 『`bldapp.bat` -- Builds Windows Micro Focus Cobol applications』
- 『`client.cbl` -- How to set and query a client (MF COBOL)』
- 『`updat.sqb` -- How to update, delete and insert table data (MF COBOL)』

- 『embprep.bat -- Prep and binds a C/C++ or Micro Focus COBOL embedded SQL program on Windows』

Micro Focus COBOL 應用程式的批次檔

```

@echo off
rem BATCH FILE: bldapp.bat
rem Builds Windows Micro Focus Cobol applications
rem Usage: bldapp <prog_name> [ <db_name> [ <userid> <password> ]]

rem If an embedded SQL program, precompile and bind it.
if not exist "%1.sqb" goto compile_step
call embprep %1 %2 %3 %4

:compile_step
rem Compile the error-checking utility.
cobol checkerr.cbl;

rem Compile the program.
cobol %1.cbl;

rem Link the program.
cbllink -l %1.obj checkerr.obj db2api.lib
@echo on

```

Windows Micro COBOL 應用程式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Micro Focus COBOL 編譯器在 Windows 上建置 COBOL 內嵌式 SQL 及 DB2 API 應用程式，如同 bldapp.bat 批次檔中所示範一般。

bldapp 的編譯及鏈結選項	
編譯選項：	
cobol	Micro Focus COBOL 編譯器。
鏈結選項：	
cbllink	使用鏈結器來鏈結編輯。
-l	鏈結 lcobol 檔案庫。
checkerr.obj	鏈結錯誤檢查公用程式目的檔。
db2api.lib	鏈結 DB2 API 檔案庫。
請參閱您的編譯器文件，取得其他編譯器選項。	

相關作業:

- 第 303 頁的『在 Windows 上建置 Micro Focus COBOL 應用程式』

相關範例:

- 『bldapp.bat -- Builds Windows Micro Focus Cobol applications』

在 Windows 上建置 Micro Focus COBOL 常式

DB2 提供批次檔，以 Micro Focus COBOL 來編譯及鏈結 DB2 API 及內嵌式 SQL 程式。這些都位於 `sqllib\samples\cobol_mf` 目錄，而且目錄中還有可利用這些檔案來建置的範例程式。

批次檔 `bldrtn.bat` 含有要建置內嵌式 SQL 常式 (儲存程序) 的命令。批次檔將常式編譯成伺服器上的 DLL。批次檔最多使用 2 個參數，在批次檔內以變數 `%1` 及 `%2` 代表。

第一個參數 `%1`，指定您的來源檔的名稱。批次檔會使用來源檔名稱 `%1` 作為 DLL 名稱。第二個參數 `%2`，指定您想要與其連接的資料庫的名稱。既然儲存程序必須建置在資料庫常駐的同一案例中，就不需要任何使用者 ID 及通行碼的參數。

只有第一個參數是必要的，即來源檔名稱。資料庫名稱是可選用的。如果未提供任何資料庫名稱，則程式將使用預設的資料庫 `sample`。

程序:

當連接到範例資料庫時，若要從來源檔 `outsrv.sqb` 建置範例程式 `outsrv`，請輸入：

```
bldrtn outsrv
```

如果連接其它的資料庫，亦請輸入資料庫名稱：

```
bldrtn outsrv database
```

Script 檔會將 DLL 複製到路徑 `sqllib/function` 中的伺服器5。

一旦建置了 DLL `outsrv`，您就可以建置從屬站應用程式 `outcli`，呼叫 DLL 內的常式 (與 DLL 同名)。您可以使用批次檔 `bldapp.bat`，建置 `outcli`。

若要呼叫 `outsrv` 常式，請輸入下列命令以執行範例從屬站應用程式：

```
outcli database userid password
```

其中

database

爲您想要與其連接的資料庫的名稱。此名稱可以是 `sample` 或它的遠端別名或其它名稱。

userid 爲有效的使用者 ID。

password

爲使用者 ID 的有效通行碼。

從屬站應用程式將存取 DLL `outsrv`，以在伺服器資料庫上執行名稱相同的常式。然後將輸出傳回從屬站應用程式。

相關概念:

- 第 90 頁的『建置檔』

相關參照:

- 第 308 頁的『Windows Micro COBOL 常式編譯及鏈結選項』
- 第 76 頁的『COBOL 範例』

相關範例:

- 『`bldrtn.bat` -- Builds Windows Micro Focus Cobol routines (stored procedures)』
- 『`outcli.sqb` -- Call stored procedures using the SQLDA structure (MF COBOL)』
- 『`outsrv.sqb` -- Demonstrates stored procedures using the SQLDA structure (MF COBOL)』
- 『`embprep.bat` -- Prep and binds a C/C++ or Micro Focus COBOL embedded SQL program on Windows』

Micro Focus COBOL 常式的批次檔

```
@echo off
rem BATCH FILE: bldrtn.bat
rem Builds Windows Micro Focus Cobol routines (stored procedures)
rem Usage: bldsrv <prog_name> [ <db_name> ]

rem Precompile and bind the program.
call embprep %1 %2

rem Compile the stored procedure.
cobol %1.cbl /case;

rem Link the stored procedure and create a shared library.
cbllink /d %1.obj db2api.lib

rem Copy the stored procedure to the %DB2PATH%\function directory.
copy %1.dll "%DB2PATH%\function"
@echo on
```

Windows Micro COBOL 常式編譯及鏈結選項

下列是 DB2 建議的編譯及鏈結選項，用以利用 Micro Focus COBOL 編譯器建置 COBOL 常式 (儲存程序及使用者定義的函數)，如同 bldrtn.bat 批次檔中所示範一般。

bldrtn 的編譯及鏈結選項	
編譯選項：	
cobol	Micro Focus COBOL 編譯器。
/case	防止外部符號被轉換成大寫字體。
鏈結選項：	
cbllink	使用 Micro Focus COBOL 鏈結器來鏈結編輯。
/d	建立 .dll 檔案。
db2api.lib	鏈結 DB2 API 檔案庫。
請參閱您的編譯器文件，取得其他編譯器選項。	

相關作業：

- 第 306 頁的『在 Windows 上建置 Micro Focus COBOL 常式』

相關範例：

- 『bldrtn.bat -- Builds Windows Micro Focus Cobol routines (stored procedures)』

物件 REXX

在 Windows 上建置 Object REXX 應用程式

Object REXX 是 REXX 語言的物件導向版本。物件導向功能已加入傳統的 REXX 中，但未改變其原有的函數及命令。Object REXX 直譯器已經過進一步改良，提供下列額外的支援：

- 類別、物件及方法
- 傳訊及多型性
- 單一及多重繼承

Object REXX 與傳統的 REXX 完全相容。本節中提及 REXX 時，表示所有 REXX 版本，包括 Object REXX。

您不必前置編譯或連結 REXX 程式。

在 Windows 上，REXX 程式的開頭不需要有註解。不過，基於可移轉性的理由，每一個 REXX 程式在第一行的第一欄中，最好還是以註解做開頭。如此可分辨程式與其它平台上的批次命令：

```
/* Any comment will do. */
```

REXX 範例程式位於 `sqllib\samples\rexx` 目錄中。

程序:

若要執行範例 REXX 程式 `updat`，請輸入：

```
rexx updat.cmd
```

相關參照:

- 第 86 頁的『REXX 範例』

第 4 篇 附錄與後記

附錄 A. DB2 Universal Database 技術資訊

DB2 Universal Database 技術資訊概觀

您可以取得下列格式的 DB2 Universal Database 技術資訊：

- 書籍 (PDF 及印刷書籍格式)
- 主題樹狀結構 (HTML 格式)
- DB2 工具說明 (HTML 格式)
- 範例程式 (HTML 格式)
- 命令行說明
- 教學指導

本節描述所提供之技術資訊的概觀，以及如何存取該資訊的方法。

DB2 文件的 FixPak

IBM 可能會定期地提供 FixPak 文件。有最新的資訊時，FixPak 文件可讓您更新從 *DB2 HTML 文件 CD* 所安裝的資訊。

註：若您有安裝 FixPak 文件，則您的 HTML 文件將包含比 DB2 印刷書籍或線上 PDF 手冊更新的資訊。

DB2 技術資訊的種類

DB2 技術資訊根據下列標題進行分類：

- DB2 核心資訊
- 管理資訊
- 應用程式開發資訊
- Business Intelligence 資訊
- DB2 Connect 資訊
- 入門資訊
- 教學指導資訊
- 選用性元件資訊
- 版本注意事項

下列表格說明了對 DB2 圖書庫中每本書籍在訂購印刷書籍、列印或檢視 PDF，或尋找此書籍的 HTML 目錄時所需的資訊。您可以在 www.ibm.com/shop/publications/order 的「IBM 出版品中心 (IBM Publications Center)」中，找到有關 DB2 圖書庫中各書籍的可用完整說明。

每個資訊種類的 HTML 文件 CD 都有其不同的安裝目錄：

`htmlcdpath/doc/htmlcd/%L/category`

其中：

- `htmlcdpath` 是 HTML CD 的安裝目錄。
- `%L` 是語言識別字。例如，`en_US`。
- `category` 是種類識別字。例如，`core` 是 DB2 核心資訊。

在下列表格的 PDF 檔名直欄中，檔名中第六個字指出書籍的語言版本。例如，檔名 `db2d1e80` 代表英文版的 *Administration Guide: Planning*，而檔名 `db2d1g80` 則代表同一本書的德文版。下列字母會用在檔名的第六個字上，以表示語言版本：

語言	識別字
阿拉伯文	w
巴西葡萄牙文	b
保加利亞文	u
克羅埃西亞文	9
捷克文	x
丹麥文	d
荷蘭文	q
英文	e
芬蘭文	y
法文	f
德文	g
希臘文	a
匈牙利文	h
義大利文	i
日文	j
韓文	k
挪威文	n
波蘭文	p
葡萄牙文	v
羅馬尼亞文	8
俄文	r
簡體中文	c
斯洛伐克文	7
斯洛維尼亞文	l
西班牙文	z

瑞典文	s
繁體中文	t
土耳其文	m

無書號表示該書籍僅有線上版本，而沒有印刷版。

DB2 核心資訊

本類資訊涵蓋了對所有 DB2 使用者而言為基本概念的 DB2 主題。不論您是程式設計師、資料庫管理員，或是使用 DB2 Connect、DB2 Warehouse Manager 或其他的 DB2 產品，您都可以在此種類中找到非常有用的資訊。

此種類的安裝目錄是 `doc/htmlcd/%L/core`。

表 27. DB2 核心資訊

名稱	書號	PDF 檔名
<i>IBM DB2 Universal Database Command Reference</i>	SC09-4828	db2n0x80
<i>IBM DB2 Universal Database 名詞解釋</i>	沒有書號	db2t0t80
<i>IBM DB2 Universal Database Master Index</i>	SC09-4839	db2w0x80
<i>IBM DB2 Universal Database 訊息參考手冊，第一冊</i>	GC40-0735	db2m1t80
<i>IBM DB2 Universal Database 訊息參考手冊，第二冊</i>	GC40-0736	db2m2t80
<i>IBM DB2 Universal Database 新特性介紹</i>	SC40-0734	db2q0t80

管理資訊

本類資訊涵蓋了有效的設計、實施，以及維護 DB2 資料庫、資料倉儲和聯合系統所需的主题。

本類的安裝目錄為 `doc/htmlcd/%L/admin`。

表 28. 管理資訊

名稱	書號	PDF 檔名
<i>IBM DB2 Universal Database Administration Guide: Planning</i>	SC09-4822	db2d1x80

表 28. 管理資訊 (繼續)

名稱	書號	PDF 檔名
<i>IBM DB2 Universal Database Administration Guide: Implementation</i>	SC09-4820	db2d2x80
<i>IBM DB2 Universal Database Administration Guide: Performance</i>	SC09-4821	db2d3x80
<i>IBM DB2 Universal Database Administrative API Reference</i>	SC09-4824	db2b0x80
<i>IBM DB2 Universal Database Data Movement Utilities Guide and Reference</i>	SC09-4830	db2dmx80
<i>IBM DB2 Universal Database Data Recovery and High Availability Guide and Reference</i>	SC09-4831	db2hax80
<i>IBM DB2 Universal Database 資料倉儲中心管理手冊</i>	SC40-0750	db2ddt80
<i>IBM DB2 Universal Database Federated Systems Guide</i>	GC27-1224	db2fpx80
<i>IBM DB2 Universal Database 管理與開發 GUI 工具指南</i>	SC40-0747	db2att80
<i>IBM DB2 Universal Database 抄寫指南與參考手冊</i>	SC40-0751	db2e0t80
<i>IBM DB2 安裝與管理衛星環境</i>	GC40-0776	db2dst80
<i>IBM DB2 Universal Database SQL Reference, Volume 1</i>	SC09-4844	db2s1x80
<i>IBM DB2 Universal Database SQL Reference, Volume 2</i>	SC09-4845	db2s2x80
<i>IBM DB2 Universal Database System Monitor Guide and Reference</i>	SC09-4847	db2f0x80

應用程式開發資訊

本類資訊是提供給對使用 DB2 應用程式來進行開發或設計程式有特殊興趣的人。您可以在這裡找到關於支援語言及編譯器的資訊，以及使用不同支援程式設計介

面 (如內含的 SQL、ODBC、JDBC、SQLj 和 CLI) 存取 DB2 時的必要文件。如果您是在線上檢視該資訊的 HTML 格式，您還可以存取到一組 DB2 範例程式 HTML 格式。

本類的安裝目錄為 doc/htmlcd/%L/ad。

表 29. 應用程式開發資訊

名稱	書號	PDF 檔名
<i>IBM DB2 Universal Database</i> 應用程式開發手冊：建置及執行應用程式	SC40-0726	db2axt80
<i>IBM DB2 Universal Database</i> <i>Application Development Guide: Programming Client Applications</i>	SC09-4826	db2a1x80
<i>IBM DB2 Universal Database</i> <i>Application Development Guide: Programming Server Applications</i>	SC09-4827	db2a2x80
<i>IBM DB2 Universal Database</i> <i>Call Level Interface Guide and Reference, Volume 1</i>	SC09-4849	db211x80
<i>IBM DB2 Universal Database</i> <i>Call Level Interface Guide and Reference, Volume 2</i>	SC09-4850	db212x80
<i>IBM DB2 Universal Database</i> <i>Data Warehouse Center Application Integration Guide</i>	SC27-1124	db2adx80
<i>IBM DB2 XML Extender</i> <i>Administration and Programming</i>	SC27-1234	db2sxx80

Business Intelligence 資訊

本類資訊說明該如何使用可增強資料倉儲並分析 DB2 Universal Database 功能的元件。

本類的安裝目錄為 doc/htmlcd/%L/wareh。

表 30. Business Intelligence 資訊

名稱	書號	PDF 檔名
<i>IBM DB2 Warehouse Manager</i> 資訊型錄中心管理手冊	SC40-0748	db2dit80

表 30. *Business Intelligence* 資訊 (繼續)

名稱	書號	PDF 檔名
<i>IBM DB2 Warehouse Manager</i> 安裝手冊	GC40-0749	db2idt80

DB2 Connect 資訊

本類資訊說明該如何使用 DB2 Connect Enterprise Edition 或 DB2 Connect Personal Edition 來存取主電腦或 iSeries 資料。

本類的安裝目錄為 `doc/htmlcd/%L/conn`。

表 31. *DB2 Connect* 資訊

名稱	書號	PDF 檔名
<i>APPC, CPI-C, and SNA Sense Codes</i>	沒有書號	db2apx80
<i>IBM 連接環境補充資料</i>	沒有書號	db2h1t80
<i>DB2 Connect Enterprise Edition</i> 快速入門	GC40-0729	db2c6t80
<i>DB2 Connect Personal Edition</i> 快速入門	GC40-0730	db2c1t80
<i>IBM DB2 Connect User's Guide</i>	SC09-4835	db2c0x80

入門資訊

當您安裝及架構伺服器、從屬站和其它 DB2 產品時，本類資訊是非常有用的。

本類的安裝目錄為 `doc/htmlcd/%L/start`。

表 32. 入門資訊

名稱	書號	PDF 檔名
<i>IBM DB2 Universal Database</i> <i>DB2 從屬站快速入門</i>	GC40-0728	db2itt80
<i>IBM DB2 Universal Database</i> <i>DB2 伺服器快速入門</i>	GC40-0731	db2ist80
<i>IBM DB2 Universal Database</i> <i>DB2 Personal Edition 快速入門</i>	GC40-0733	db2i1t80
<i>IBM DB2 Universal Database</i> 安裝與架構補充資料	GC40-0732	db2iyt80

表 32. 入門資訊 (繼續)

名稱	書號	PDF 檔名
IBM DB2 Universal Database DB2 Data Links Manager 快速 入門	GC40-0727	db2z6t80

教學指導資訊

教學指導資訊介紹 DB2 特性，並教導您如何執行各種作業。

本類的安裝目錄為 doc/htmlcd/%L/tutr。

表 33. 教學指導資訊

名稱	書號	PDF 檔名
<i>Business Intelligence</i> 教學指 導：資料倉儲中心簡介	沒有書號	db2tut80
<i>Business Intelligence</i> 教學指 導：資料倉儲進階課程	沒有書號	db2tat80
開發中心教學指導 - <i>Video</i> <i>Online</i> (使用 <i>Microsoft Visual</i> <i>Basic</i>)	沒有書號	db2tdt80
資訊型錄中心教學指導	沒有書號	db2ait80
<i>Video Central for e-business</i> <i>Tutorial</i>	沒有書號	db2twx80
<i>Visual Explain</i> 教學指導	沒有書號	db2tvt80

選用性元件資訊

本類資訊說明如何使用選用性 DB2 元件。

本類的安裝目錄為 doc/htmlcd/%L/opt。

表 34. 選用性元件資訊

名稱	書號	PDF 檔名
IBM DB2 Life Sciences Data Connect Planning, Installation, and Configuration Guide	GC27-1235	db2lsx80
IBM DB2 Spatial Extender 使 用與參考手冊	SC40-0753	db2sbt80

表 34. 選用性元件資訊 (繼續)

名稱	書號	PDF 檔名
<i>IBM DB2 Universal Database Data Links Manager Administration Guide and Reference</i>	SC27-1221	db2z0x80
<i>DB2 UDB Net Search Extender 管理與程式設計手冊</i>	SH40-0356	無

註: 本文件的 HTML 格式並非從 HTML 文件 CD 中安裝。

版本注意事項

版本注意事項提供您產品版本及 FixPak 版次的特殊附加資訊。並且也提供各版次及 FixPak 所納入的文件更新摘要。

表 35. 版本注意事項

名稱	書號	PDF 檔名
<i>DB2 版本注意事項</i>	請參閱備註。	請參閱備註。
<i>DB2 安裝注意事項</i>	僅附於產品 CD-ROM 中。	僅附於產品 CD-ROM 中。

註: 您可以從「資訊中心」及產品 CD-ROM 中, 取得可用的版本注意事項的 HTML 版本。若要在 UNIX 平台上檢視 ASCII, 請參閱 Release.Notes 檔案。此檔案是位在 DB2DIR/Readme/%L 目錄中, 其中 %L 代表語言環境名稱, 而 DB2DIR 代表:

- /usr/opt/db2_08_01 (在 AIX 上)
- /opt/IBM/db2/V8.1 (在所有其它 UNIX 作業系統上)

相關作業:

- 第 321 頁的『從 PDF 檔案列印 DB2 書籍』
- 第 321 頁的『訂購 DB2 印刷書籍』
- 第 322 頁的『存取線上說明』
- 第 325 頁的『從管理工具存取「DB2 資訊中心」以尋找產品資訊』
- 第 327 頁的『直接從「DB2 HTML 文件 CD」檢視線上技術文件』

從 PDF 檔案列印 DB2 書籍

您可以從 *DB2 PDF* 文件 CD 上的 PDF 檔列印 DB2 書籍。利用 Adobe Acrobat Reader，您可以列印整本書或只列印特定範圍的頁數。

先決條件:

請確定您已安裝 Adobe Acrobat Reader。可於 Adobe 網站 www.adobe.com 上取得。

程序:

若要從 PDF 檔案來列印 DB2 書籍：

1. 插入 *DB2 PDF* 文件 CD。在 UNIX 作業系統中，請裝載「DB2 PDF 文件」CD。請參照快速入門，以取得如何在 UNIX 作業系統中裝載 CD 的詳細說明。
2. 啟動 Adobe Acrobat Reader。
3. 從下列其中一個位置開啓 PDF 檔：
 - 在 Windows 作業系統，請：
`x:\doc\language` 目錄，其中 *x* 代表光碟機字母，而 *language* 表示兩個字元的專屬區字碼，代表您所使用的語言 (例如，EN 代表英文)。
 - 在 UNIX 作業系統，請：
CD-ROM 中的 `/cdrom/doc/%L` 目錄，其中 `/cdrom` 代表 CD-ROM 的裝載點，且 `%L` 代表想要的語言環境名稱。

相關作業:

- 第 321 頁的『訂購 DB2 印刷書籍』
- 第 325 頁的『從管理工具存取「DB2 資訊中心」以尋找產品資訊』
- 第 327 頁的『直接從「DB2 HTML 文件 CD」檢視線上技術文件』

相關參照:

- 第 313 頁的『DB2 Universal Database 技術資訊概觀』

訂購 DB2 印刷書籍

程序:

若要訂購印刷書籍：

- 請聯絡您的 IBM 授權經銷商或業務代表。若要尋找當地的 IBM 業務代表，請查閱位於 www.ibm.com/planetwide 的「IBM 全球聯絡站名錄 (IBM Worldwide Directory of Contacts)」。

- 在美國，請電洽 1-800-879-2755；在加拿大，請電洽 1-800-IBM-4YOU。
- 請造訪位於 www.ibm.com/shop/publications/order 的「IBM 出版品中心 (IBM Publications Center)」。

您還可以從 IBM 經銷商訂購 DB2 產品的 Doc Pack，以取得 DB2 印刷書籍。Doc Pack 是選自 DB2 圖書庫中各手冊的子集，協助您開始使用所購買的 DB2 產品。Doc Pack 中的手冊與 *DB2 PDF 文件 CD* 中 PDF 格式的手冊相同，且內容與 *DB2 HTML 文件 CD* 中的手冊內容一致。

相關作業:

- 第 321 頁的『從 PDF 檔案列印 DB2 書籍』
- 第 324 頁的『從瀏覽器存取「DB2 資訊中心」以尋找主題』
- 第 327 頁的『直接從「DB2 HTML 文件 CD」檢視線上技術文件』

相關參照:

- 第 313 頁的『DB2 Universal Database 技術資訊概觀』

存取線上說明

所有 DB2 元件隨附的線上說明有三種可用的類型：

- 視窗及筆記本說明
- 命令行說明
- SQL 陳述式說明

視窗及筆記本說明，會說明您可以在某視窗及及筆記本中執行的作業，並說明其控制項。該說明有兩種類型：

- 可從說明按鈕存取的說明
- 欄位及控制項的說明

說明按鈕提供您存取概觀及先決條件資訊的權限。欄位及控制項的說明，則說明了視窗或筆記本中的控制項。您可以從具有使用者介面的 DB2 中心及元件取得可用的視窗及筆記本說明。

命令行說明，包括「命令」說明及「訊息」說明。「命令」說明是說明命令行處理器中的命令語法。「訊息」說明是描述錯誤訊息的原因，及回應錯誤應採用的動作。

SQL 陳述式說明包括 SQL 說明及 SQLSTATE 說明。DB2 會傳回條件的 SQLSTATE 值，該值可為 SQL 陳述式的結果。SQLSTATE 說明，解譯 SQL 陳述式 (SQL 狀態及類別碼) 的語法。

註: UNIX 作業系統中沒有可用的 SQL 說明。

程序:

若要存取線上說明:

- 對於視窗及筆記本說明,請按一下說明或按一下控制項,然後再按一下 **F1**。如果選取了工具設定之一般事項頁面的自動顯示欄位及控制項的說明勾選框,則只要將滑鼠游標保持在控制項上,您也可以看見特定控制項之欄位及控制項的說明。

- 對於命令行說明,請開啓命令行處理器並輸入:

– 「命令」說明:

`? command`

其中 *command* 代表某一關鍵字或整個命令。

例如, `? catalog` 將顯示所有 CATALOG 命令的說明,至於 `? catalog database` 則會顯示 CATALOG DATABASE 命令的說明。

- 「訊息」說明:

`? XXXnnnnn`

其中, *XXXnnnnn* 代表有效的訊息識別字。

例如, `? SQL30081` 會顯示關於 SQL30081 訊息的說明。

- 對於 SQL 陳述式說明,請開啓命令行處理器並輸入:

`? sqlstate` 或 `? class code`

其中 *sqlstate* 代表有效的五位數 SQL 狀態,且 *class code* 代表 SQL 狀態的前兩位數。

例如, `? 08003` 將顯示 08003 SQL 陳述式的說明,至於 `? 08` 則將顯示 08 類別碼的說明

相關作業:

- 第 324 頁的『從瀏覽器存取「DB2 資訊中心」以尋找主題』
- 第 327 頁的『直接從「DB2 HTML 文件 CD」檢視線上技術文件』

從瀏覽器存取「DB2 資訊中心」以尋找主題

從瀏覽器存取的「DB2 資訊中心」，可讓您存取充分利用 DB2 Universal Database 及 DB2 Connect 時所需的資訊。「DB2 資訊中心」也記錄了主要的 DB2 特性及元件，包括抄寫、資料倉儲、中間資料和 DB2 擴充程式。

從瀏覽器存取的「DB2 資訊中心」是由下列主要元素組成：

導覽樹 導覽樹位於瀏覽器視窗的左框中。導覽樹會展開及收合，以顯示及隱藏「DB2 資訊中心」的主題、名詞解釋和主要索引。

導覽工具列

導覽工具列位於瀏覽器視窗頂端的右框中。導覽工具列包含了多個按鈕，這些按鈕可讓您搜尋「DB2 資訊中心」、隱藏導覽樹，並在導覽樹尋找目前顯示的主題。

內容框 內容框位於瀏覽器視窗底端的右框中。內容框會顯示「DB2 資訊中心」的主題，其方法是按一下導覽樹的鏈結、按一下搜尋結果，或者遵循另一個索引或主要索引的鏈結。

先決條件:

若要從瀏覽器存取「DB2 資訊中心」，您必須使用下列其中一個瀏覽器：

- Microsoft Explorer，版本 5 或更新版本
- Netscape Navigator，版本 6.1 或更新版本

限制:

「DB2 資訊中心」僅包含那些您選擇從 *DB2 HTML 文件 CD* 安裝的主題集。如果在您嘗試遵循鏈結到主題時 Web 瀏覽器傳回找不到檔案的錯誤，則必須從 *DB2 HTML 文件 CD* 安裝一個或多個附加主題集。

程序:

若要藉由搜尋關鍵字來尋找主題：

1. 在導覽工具列中，按一下**搜尋**。
2. 在「搜尋」視窗的頂端文字輸入欄位中，輸入一個或多個與關注區域相關的術語，然後再按一下**搜尋**。根據精確度評定等級的主題清單，即會顯示在**結果欄**位中。在命中項旁的數字等級是相符強度的指示（數字越大表示相符度越高）。當減少由查詢傳回的主題數量時，請輸入更多的術語以增加您的查詢的精準度。
3. 在**結果欄**位中，按一下您要讀取的主題標題。該主題即會顯示在內容框。

若要在導覽樹中尋找主題：

1. 在導覽樹按一下與關注區域相關之主題種類的書籍圖示。子種類清單即會顯示在圖示下方。
2. 繼續按一下書籍圖示，直到找到包含您所關注之主題的種類為止。當您將游標移至種類標題上面時，鏈結至主題的種類會以畫有底線的鏈結顯示種類標題。導覽樹會以頁面圖示來識別主題。
3. 按一下主題鏈結。該主題即會顯示在內容框。

若要在主要索引中尋找主題或術語：

1. 在導覽樹中，按一下『索引』種類。種類即會展開，顯示導覽樹中按字母順序排列的鏈結清單。
2. 在導覽樹中，按一下相對應於術語第一個字元的鏈結 (該術語與所關注的主題相關)。包含起始字元的術語清單即會顯示在內容框中。會使用書籍圖示來識別具有多個索引登錄的術語。
3. 按一下相對應於關注之術語的書籍圖示。子術語及主題清單即會顯示在您所按的術語下方。會使用畫有底線的標題的頁面圖示來識別主題。
4. 按一下符合您需求的主題標題。該主題即會顯示在內容框。

相關概念:

- 第 332 頁的『協助工具』
- 第 334 頁的『從瀏覽器存取的「DB2 資訊中心」』

相關作業:

- 第 325 頁的『從管理工具存取「DB2 資訊中心」以尋找產品資訊』
- 第 327 頁的『更新已安裝於您的機器上的 HTML 文件』
- 第 329 頁的『使用 Netscape 4.x 來搜尋 DB2 文件的疑難排解』
- 第 330 頁的『搜尋 DB2 文件』

相關參照:

- 第 313 頁的『DB2 Universal Database 技術資訊概觀』

從管理工具存取「DB2 資訊中心」以尋找產品資訊

「DB2 資訊中心」可讓您快速地存取 DB2 產品資訊，且該方式適用於所有可用 DB2 管理工具的作業系統中。

從工具存取的「DB2 資訊中心」提供六種資訊類型。

作業 列出您可以 DB2 執行的作業。

概念 DB2 的關鍵概念。

參照 DB2 參考資料、如關鍵字、命令及 API。

疑難排解

協助您處理一般 DB2 問題的錯誤訊息及資訊。

範例 鏈結至 DB2 所提供之範例程式的 HTML 報表。

教學指導

教學輔助設計，可協助您學習 DB2 特性。

先決條件:

「DB2 資訊中心」的部份鏈結會指向網際網路中的網站。若要顯示這些鏈結的內容，首先您必須連接至網際網路。

程序:

若要藉由從工具存取「DB2 資訊中心」來尋找產品資訊：

1. 以下列其中一種方式啓動「DB2 資訊中心」：
 - 從圖形式管理工具中，按一下工具列中的**資訊中心**圖示。您也可以從**說明功能表**中選取它。
 - 在命令行中，輸入 **db2ic**。
2. 按一下與您嘗試尋找之資訊相關的資訊類型標籤。
3. 導覽樹狀結構，並按一下所關注的主題。接著，「資訊中心」將會啓動一個 Web 瀏覽器以顯示資訊。
4. 若要尋找資訊但不瀏覽清單，請按一下清單右側的**搜尋**圖示。
一旦「資訊中心」啓動瀏覽器以顯示資訊，即可以執行全文搜尋，其方法是按一下導覽工具列中的**搜尋**圖示。

相關概念:

- 第 332 頁的『協助工具』
- 第 334 頁的『從瀏覽器存取的「DB2 資訊中心」』

相關作業:

- 第 324 頁的『從瀏覽器存取「DB2 資訊中心」以尋找主題』
- 第 330 頁的『搜尋 DB2 文件』

直接從「DB2 HTML 文件 CD」檢視線上技術文件

您也可以直接從 CD 讀取所有可自 *DB2 HTML 文件 CD* 安裝的 HTML 主題。因此，您可以只檢視文件，不一定要安裝。

限制:

因為「工具」說明是從 DB2 產品 CD 安裝而非從 *DB2 HTML 文件 CD* 安裝，所以您必須安裝 DB2 產品才能檢視說明。

程序:

1. 插入 *DB2 HTML 文件 CD*。在 UNIX 作業系統中，裝載 *DB2 HTML 文件 CD*。請參照快速入門一書，以取得如何在 UNIX 作業系統中裝載 CD 的詳細說明。

2. 啟動您的 HTML 瀏覽器，並開啓適當檔案：

- 對於 Windows 作業系統：

```
e:\program files\IBM\SQLLIB\doc\htmlcd\%L\index.htm
```

其中，*e* 代表光碟機；%L 則是您想要使用的文件語言環境，例如，英文版為 **en_US**。

- 對於 UNIX 作業系統：

```
/cdrom/program files/IBM/SQLLIB/doc/htmlcd/%L/index.htm
```

其中，*/cdrom/* 代表 CD 的裝載位置，而 %L 則是您想要使用的文件語言環境，例如，英文版為 **en_US**。

相關作業:

- 第 324 頁的『從瀏覽器存取「DB2 資訊中心」以尋找主題』
- 第 329 頁的『將檔案從 DB2 HTML 文件 CD 複製到 Web 伺服器』

相關參照:

- 第 313 頁的『DB2 Universal Database 技術資訊概觀』

更新已安裝於您的機器上的 HTML 文件

現在，只要 IBM 備妥可用的更新資訊，您即可從 *DB2 HTML 文件 CD* 來更新已安裝的 HTML。您可以用下列兩種方式的其中一種方式來完成該作業：

- 使用「資訊中心」(如果您已安裝 DB2 管理 GUI 工具)。
- 下載及引用 DB2 HTML 文件 FixPak。

註: 如此並「不會」更新 DB2 程式碼；它僅會更新安裝自 *DB2 HTML 文件 CD* 的 HTML 文件。

程序:

若要用「資訊中心」來更新您的本端文件：

1. 以下列其中一種方式啓動「DB2 資訊中心」：
 - 從圖形式管理工具中，按一下工具列中的**資訊中心**圖示。您也可以從**說明功能表**中選取它。
 - 在命令行中，輸入 **db2ic**。
2. 確定您的機器已存取外部網際網路；則更新程式將自 IBM 伺服器中下載最新的文件 **FixPak** (如果必要的話)。
3. 在功能表中，選取**資訊中心** —> **更新本端文件**，以開始更新。
4. 提供您的虛擬資訊 (如果必要的話)，以連接至外部網際網路。

如此即會下載「資訊中心」，並引用最新的文件 **FixPak** (如果有可用的)。

若要手動下載並引用文件 **FixPak**：

1. 確定您的機器已連接至網際網路。
2. 在 Web 瀏覽器中開啓位於 www.ibm.com/software/data/db2/udb/winos2unix/support 的 **DB2 支援**頁面。
3. 遵循版本 8 的鏈結，並尋找「文件 **FixPak**」鏈結。
4. 比較文件 **FixPak** 層次與您所安裝之文件層次，以確定您的本端文件版本是否過期。您機器中目前的文件是下列層次：**DB2 v8.1 GA**。
5. 如果有更新的可用文件版本，那麼請下載適用於您的作業系統的 **FixPak**。有一個適用於所有 Windows 平台的 **FixPak**，及一個適用於所有 UNIX 平台的 **FixPak**。
6. 引用 **FixPak**：
 - 就 Windows 作業系統：文件 **FixPak** 是自解壓縮檔。將下載的文件 **FixPak** 放在空的目錄中，然後執行它。它將會建立一個 **setup** 命令，您可以執行該命令來安裝文件 **FixPak**。
 - 就 UNIX 作業系統：文件 **FixPak** 是壓縮的 tar.Z 檔案。解壓縮並 **untar** 此檔案。它會建立一個名為 **delta_install** 的目錄，並具有一個名為 **installdocfix** 的 Script。執行該 Script 以安裝文件 **FixPak**。

相關作業:

- 第 329 頁的『將檔案從 DB2 HTML 文件 CD 複製到 Web 伺服器』

相關參照:

- 第 313 頁的『DB2 Universal Database 技術資訊概觀』

將檔案從 DB2 HTML 文件 CD 複製到 Web 伺服器

DB2 HTML 文件 CD 會將整個 DB2 資訊檔案庫遞送給您，且可能會將檔案庫安裝在 Web 伺服器上以方便您存取。只要將您想要的語言版本文件複製到您的 Web 伺服器即可。

註：若您是透過低速連線從 Web 伺服器存取 HTML 文件，效能可能會較慢。

程序：

若要將檔案從 DB2 HTML 文件 CD 複製到 Web 伺服器上，請使用適當的來源路徑：

- 對於 Windows 作業系統：

```
E:\program files\IBM\SQLLIB\doc\htmlcd\%L\*.*
```

其中，*E* 代表光碟機，而 *%L* 代表語言識別字。

- 對於 UNIX 作業系統：

```
/cdrom/program files/IBM/SQLLIB/doc/htmlcd/%L/*.*
```

其中，*cdrom* 代表光碟機的裝載點，而 *%L* 代表語言識別字。

相關作業：

- 第 330 頁的『搜尋 DB2 文件』

相關參照：

- DB2 伺服器快速入門 中的『支援的 DB2 介面語言、語言環境及字碼頁』
- 第 313 頁的『DB2 Universal Database 技術資訊概觀』

使用 Netscape 4.x 來搜尋 DB2 文件的疑難排解

大部份的搜尋問題皆與 Web 瀏覽器所提供的 Java 支援相關。該作業說明了可能的可行方案。

程序：

Netscape 4.x 的一般問題是和遺失或錯位的安全等級有關。請嘗試下列解決方案，特別是在您於瀏覽器 Java 主控台中看見下列字行時：

```
Cannot find class java/security/InvalidParameterException
```

- 對於 Windows 作業系統，請：

從 *DB2 HTML* 文件 *CD*，將提供的 `x:program files\IBM\SQLLIB\doc\htmlcd\locale\InvalidParameterException.class` 檔案複製到與 Netscape 瀏覽器安裝相對的 `java\classes\java\security\` 目錄中；其中，*x* 代表光碟機字母，而 *locale* 則代表所需的語言環境名稱。

註：您可能需要建立 `java\security\` 次目錄結構。

- 對於 UNIX 作業系統，請：

從 *DB2 HTML* 文件 *CD*，將提供的 `/cdrom/program files/IBM/SQLLIB/doc/htmlcd/locale/InvalidParameterException.class` 檔案複製到與 Netscape 瀏覽器安裝相對的 `java/classes/java/security/` 目錄中；其中，*cdrom* 代表 CD-ROM 的裝載點，而 *locale* 則代表所需的語言環境名稱。

註：您可能需要建立 `java/security/` 次目錄結構。

如果您的 Netscape 瀏覽器仍然無法顯示搜尋輸入視窗，請嘗試下列方法：

- 停止 Netscape 瀏覽器的所有案例，確定機器上沒有正在執行的 Netscape 程式碼。接著，開啓新的 Netscape 瀏覽器案例，並重新嘗試啓動搜尋。
- 清除瀏覽器的快取。
- 嘗試不同版本的 Netscape，或是不同的瀏覽器。

相關作業：

- 第 330 頁的『搜尋 DB2 文件』

搜尋 DB2 文件

您可以搜尋 DB2 文件的檔案庫以尋找您需要的資訊。在您從瀏覽器存取的「DB2 資訊中心」中，按一下其導覽工具列的搜尋圖示，即會開啓蹦現搜尋視窗。搜尋可能需要一些時間，視您電腦及網路的速度而定。

先決條件：

您需要 Netscape 6.1 或更新版本，或 Microsoft 的 Internet Explorer 5 或更新版本。請確定已啓用您瀏覽器的 Java 支援。

限制：

當您使用文件搜尋時，即適用下列限制：

- 搜尋是不會區分大小寫的。
- 不支援 Boolean 搜尋。

- 不支援萬用字元及部份搜尋。java* (或 java) 的搜尋僅會尋找字串 java* (或 java) ，而不會尋找如 javadoc 的字串。

程序:

若要搜尋 DB2 文件：

1. 在導覽工具列中，按一下 **搜尋** 圖示。
2. 在「搜尋」視窗的頂端的文字輸入欄位中，輸入一個或多個 (以空白鍵分隔) 與關注區域相關的術語，然後再按一下 **搜尋**。根據精確度評定等級的主題清單，即會顯示在**結果**欄位中。在命中項旁的數字等級是相符強度的指示 (數字越大表示相符度越高)。
當減少由查詢傳回的主題數量時，請輸入更多的術語以增加您的查詢的精準度。
3. 在**結果**清單中，按一下您要讀取的主題標題。該主題即會顯示在「DB2 資訊中心」內容框中。

註: 當您執行搜尋時，第一個 (最高順位) 結果即會自動載入到您的瀏覽器框格中。若要檢視其它搜尋結果的內容，請按一下結果清單中的結果。

相關作業:

- 第 329 頁的『使用 Netscape 4.x 來搜尋 DB2 文件的疑難排解』

線上 DB2 疑難排解資訊

DB2® UDB 版本 8 發貨後，將不會再有 *Troubleshooting Guide*。曾經涵蓋在該手冊中的疑難排解資訊，會被整合至 DB2 出版品。藉由該動作，我們將可以傳送最新的資訊。若要在 DB2 的疑難排解公用程式和功能上尋找資訊，請從任一工具存取「DB2 資訊中心」。

如果您遭遇問題且需要協助尋找可能的原因和解決方案，請參照「DB2 線上支援 (DB2 Online Support)」網站。此支援網站包含了大量且不斷更新的 DB2 出版品資料庫、TechNotes、APAR (產品問題) 記錄、FixPak 和其它資源。您可以使用此支援網站來搜尋該知識庫，並為您的問題找到可能的解決方案。

存取位於 www.ibm.com/software/data/db2/udb/winos2unix/support 的「線上支援」網站，或者按一下「DB2 資訊中心」的**線上支援**按鈕。現在，您也可以從該網站取得一直在更新的資訊，如：內部 DB2 錯誤碼的清單。

相關概念:

- 第 334 頁的『從瀏覽器存取的「DB2 資訊中心」』

相關作業:

- 第 325 頁的『從管理工具存取「DB2 資訊中心」以尋找產品資訊』

協助工具

「協助工具」特性會協助行動不便或視覺障礙等身體有缺陷的使用者順利地使用軟體產品。這些即是 DB2® Universal Database 版本 8 的主要協助工具特性：

- DB2 可讓您用鍵盤取代滑鼠，來操作所有的特性。請參閱『鍵盤輸入及導引』。
- DB2 可讓您自訂字型的大小及顏色。請參閱『容易使用的顯示』。
- DB2 可讓您接收圖像或音效警戒提示。請參閱『替代的警戒提示』。
- DB2 可使用 Java™ Accessibility API 來支援協助工具應用程式。請參閱第 333 頁的『與輔助技術相容』。
- DB2 隨附提供容易使用的格式的文件。請參閱第 333 頁的『容易使用的文件』。

鍵盤輸入及導引

鍵盤輸入

您可以僅使用鍵盤來操作「DB2 工具」。您可以使用按鍵或按鍵組合來執行大部份的作業 (這些作業也可使用滑鼠執行)。

鍵盤焦點

在 UNIX 型系統中，會高亮度顯示鍵盤焦點位置，指出視窗中的作用區域，及按鍵生效的位置。

容易使用的顯示

「DB2 工具」具有許多特性，可增強使用者介面並改進視力不良的使用者的協助工具。這些協助工具加強功能包括了可自訂的字型內容支援。

字型設定

「DB2 工具」可讓您使用「工具設定」筆記本，選取功能表和對話視窗的文字顏色、大小及字型。

不依靠顏色

您不需要區分顏色，即可使用本產品的任一功能。

替代的警戒提示

您可以使用「工具設定」筆記本，來指定是否要透過音效或圖像提示來接收警戒。

與輔助技術相容

「DB2 工具」介面支援 Java Accessibility API，可讓殘障人士能夠使用螢幕讀取器及其它的輔助技術。

容易使用的文件

DB2 系列產品文件可用 HTML 格式來取得。如此可讓您根據瀏覽器的顯示喜好設定來檢視文件。它也可讓您使用螢幕讀取器及其它輔助技術。

DB2 教學指導

DB2® 教學指導會協助您學習到各方面的 DB2 Universal Database。此教學指導提供逐步指示課程，讓您使用 DB2 來開發應用程式、調整 SQL 查詢效能、使用資料倉儲、管理中間資料，及開發 Web 服務等。

開始之前:

在您可以使用下列鏈結存取這些教學指導之前，您必須從 *DB2 HTML* 文件 CD 中安裝教學指導。

如果您不想安裝教學指導，您可以直接從 *DB2 HTML* 文件 CD 檢視教學指導的 HTML 版本。您也可以從 *DB2 PDF* 文件 CD 中，取得這些教學指導的 PDF 版本。

部份教學指導課程會使用範例資料或程式碼。請參閱各教學指導說明，以取得其特定作業的所有先決條件說明。

DB2 Universal Database 教學指導:

如果您已從 *DB2 HTML* 文件 CD 中安裝了教學指導，即可以在下列清單中按一下教學指導標題，以檢視該教學指導。

Business Intelligence 教學指導：資料倉儲中心簡介
使用「資料倉儲中心」，以執行初步的資料倉儲作業。

Business Intelligence 教學指導：資料倉儲進階課程
使用「資料倉儲中心」，以執行進階的資料倉儲作業。

開發中心教學指導 - *Video Online* (使用 Microsoft® Visual Basic)
使用 Development Center Add-in for Microsoft Visual Basic，以建置應用程式的各種元件。

資訊型錄中心教學指導
建立及管理資訊型錄，以使用「資訊型錄中心」來尋找和使用中間資料。

Video Central for e-business Tutorial

使用 WebSphere® 產品，以開發及交付進階「DB2 Web 服務」應用程式。

Visual Explain 教學指導

使用 Visual Explain，以分析、最佳化和調整 SQL 陳述式進而取得較佳的效能。

從瀏覽器存取的「DB2 資訊中心」

「DB2® 資訊中心」可讓您存取在企業中充分利用 DB2 Universal Database™ 及 DB2 Connect™ 時所需的資訊。「DB2 資訊中心」也記錄了主要的 DB2 特性及元件，包括抄寫、資料倉儲、「資訊型錄中心」、Life Sciences Data Connect，和 DB2 擴充程式。

若您使用 Netscape Navigator 6.1 或更新版本，或是 Microsoft Internet Explorer 5 或更新版本來檢視「DB2 資訊中心」，則從瀏覽器存取的「DB2 資訊中心」會具有下列特性。某些特性則需要您啓用 Java 或 JavaScript 的支援：

定期更新的文件

下載更新的 HTML，使主題保持在最新狀態。

搜尋 按一下導覽工具列中的**搜尋**，以搜尋所有已安裝在您的工作站中的主題。

整合的導覽樹

從單一導覽樹，尋找任何位於 DB2 檔案庫中的任何主題。導覽樹是由下列資訊類型所組成：

- 作業會提供逐步指示，告訴您如何完成目標。
- 概念會提供主旨的概觀。
- 參照主題會提供與主旨相關的詳細資訊，包括陳述式及命令語法、訊息說明、需求。

主要索引

從主要索引存取從 DB2 HTML 文件 CD 所安裝的資訊。索引是按字母順序 (根據索引術語的排序方式) 而組成。

主要名詞解釋

主要名詞解釋會定義用於「DB2 資訊中心」的術語。名詞解釋是按名詞解釋術語的字母順序來組成的。

相關作業:

- 第 324 頁的『從瀏覽器存取「DB2 資訊中心」以尋找主題』
- 第 325 頁的『從管理工具存取「DB2 資訊中心」以尋找產品資訊』

- 第 327 頁的『更新已安裝於您的機器上的 HTML 文件』

附錄 B. 注意事項

在其它國家中，IBM 可能不會提供本書中所提的各項產品、服務或功能。要知道您所在區域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，請聯絡您國家或地區的 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

下列段落若與該國或地區之法律條款抵觸，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
8200 Warden Avenue
Markham, Ontario
L6G 1C7
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，可能會有很大的差異。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所不同。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、廠牌和產品的名稱。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

下列術語是 IBM 公司在美國及 (或) 其它國家的商標，且最少用於 DB2 UDB 文件圖書庫的其中一份文件中。

ACF/VTAM	LAN Distance
AISPO	MVS
AIX	MVS/ESA
AIXwindows	MVS/XA
AnyNet	Net.Data
APPN	NetView
AS/400	OS/390
BookManager	OS/400
C Set++	PowerPC
C/370	pSeries
CICS	QBIC
Database 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extenders	SQL/DS
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	Tivoli
eServer	VisualAge
Extended Services	VM/ESA
FFST	VSE/ESA
First Failure Support Technology	VTAM
IBM	WebExplorer
IMS	WebSphere
IMS/ESA	WIN-OS/2z/OS
iSeries	zSeries

下列術語是其它公司的商標或註冊商標，且最少用於 DB2 UDB 文件圖書庫的其中一份文件中：

Microsoft、Windows、Windows NT 以及 Windows 商標是 Microsoft Corporation 在美國及 (或) 其它國家的商標。

Intel 及 Pentium 商標是 Intel Corporation 在美國及 (或) 其它國家的商標。

Java 以及所有與 Java 有關的商標是 Sun Microsystems, Inc. 在美國及 (或) 其它國家的商標。

UNIX 是 The Open Group 在美國及其它國家的註冊商標。

其它公司、產品或服務名稱可能是其它公司的商標或服務標誌。

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〕

內嵌式 SQL

建置檔 90

錯誤檢查公用程式檔案 96

AIX IBM COBOL 常式 175

AIX IBM COBOL 應用程式 172

AIX Micro Focus COBOL 常式
183

AIX Micro Focus COBOL 應
用程
式 179

AIX 上的 VisualAge C++ 架
構檔
168

AIX 上的多緒 C 應用程式 153

AIX 上的多緒 C++ 應
用程
式
164

C 常式，在 AIX 上建置 148

C 常式，在 HP-UX 上建置 193

C 常式，在 Linux 上建置 225

C 常式，在 Solaris 作業環
境上
建
置 248

C 應用程式，在 AIX 上建置 144

C 應用程式，在 HP-UX 上
建
置
189

C 應用程式，在 Linux 上
建
置
221

C 應用程式，在 Solaris 作
業環
境
上
建
置 243

COBOL 範例程式檔 76

C++ 常式，在 AIX 上建置 159

C++ 常式，在 HP-UX 上
建
置
205

C++ 常式，在 Linux 上
建
置 235

C++ 常式，在 Solaris 作
業環
境
上
建
置 260

C++ 應用程式，在 AIX 上
建
置
155

內嵌式 SQL (繼續)

C++ 應用程式，在 HP-UX 上
建
置
201

C++ 應用程式，在 Linux 上
建
置
232

C++ 應用程式，在 Solaris 作
業環
境
上
建
置 256

C/C++ 常式，在 Windows 上
建
置
290

C/C++ 範例程式檔 65

C/C++ 應用程式，在 Windows 上
建
置 286

DB2 AD Client 中的支援 3

HP-UX Micro Focus COBOL 常
式
217

HP-UX Micro Focus COBOL 應
用
程
式 214

HP-UX 上的多緒 C 應
用
程
式
199

HP-UX 上的多緒 C++ 應
用
程
式
211

Linux 上的多緒 C 應
用
程
式 230

Linux 上的多緒 C++ 應
用
程
式
241

Solaris 上的多緒 C++ 應
用
程
式
265

Solaris 作業環境 Micro Focus
COBOL 常式 271

Solaris 作業環境 Micro Focus
COBOL 應用程式 268

Solaris 作業環境上的多緒 C 應
用
程
式 253

Windows IBM COBOL 常式 299

Windows IBM COBOL 應
用
程
式
297

Windows Micro Focus COBOL 常
式 306

Windows Micro Focus COBOL 應
用
程
式 303

公用程式

錯誤檢查檔案 96

日誌管理使用者跳出程式範例檔案
88

〔五劃〕

主電腦系統

支援的伺服器 7

建立範例資料庫 41

〔六劃〕

印刷書籍，訂購 321

多緒應用程式

建置檔適用於 90

多緒應用程式，建置

以 AIX C 153

以 AIX C++ 164

以 HP-UX C 199

以 HP-UX C++ 211

以 Linux C 230

以 Linux C++ 241

以 Solaris C 253

以 Solaris C++ 265

〔七劃〕

作業系統

AIX 版本支援 9

DB2 安裝路徑 44

DB2 所支援的 8

HP-UX 版本支援 10

Linux 版本支援 11

Solaris 版本支援 13

Windows 版本支援 14

批次檔

說明 90

〔八劃〕

使用者定義的函數 (UDF)
以 Visual Basic 進行 OLE 自動化 282
以 Visual C++ 進行 OLE 自動化 285
建置檔適用於 90
AIX 上的 VisualAge C++ 架構檔 170
C
AIX 148
HP-UX 193
Linux 225
Solaris 作業環境 248
COBOL 範例程式檔 76
C++
AIX 159
HP-UX 205
Linux 235
Solaris 作業環境 260
C/C++
範例程式檔 65
Windows 290
JDBC 107
JDBC 範例程式檔 70
SQLj 119
SQLj 範例程式檔 72
協助工具
特性 332
命令行處理器 (CLP)
範例檔案 86
DB2 AD Client 3
物件連結和內嵌
自動化
以 Visual Basic 282
以 Visual C++ 285
資料庫表格函數
說明 276
範例檔案 85
範例程式檔 84
DB2 AD Client 中的支援 3
表格函數
OLE DB 276

〔九劃〕

建置檔
說明 90
適用於 AIX C 多緒應用程式 153
適用於 AIX C++ 多緒應用程式 164
適用於 AIX IBM COBOL 常式 175
適用於 AIX IBM COBOL 應用程式 172
適用於 AIX Micro Focus COBOL 常式 183
適用於 AIX Micro Focus COBOL 應用程式 179
適用於 AIX 上的 C 常式 148
適用於 AIX 上的 C 應用程式 144
適用於 AIX 上的 C++ 常式 159
適用於 AIX 上的 C++ 應用程式 155
適用於 HP-UX C 多緒應用程式 199
適用於 HP-UX C++ 多緒應用程式 211
適用於 HP-UX Micro Focus COBOL 常式 217
適用於 HP-UX Micro Focus COBOL 應用程式 214
適用於 HP-UX 上的 C 常式 193
適用於 HP-UX 上的 C++ 常式 205
適用於 Linux C 多緒應用程式 230
適用於 Linux C++ 多緒應用程式 241
適用於 Linux 上的 C 常式 225
適用於 Linux 上的 C 應用程式 221
適用於 Linux 上的 C++ 常式 235
適用於 Solaris C 多緒應用程式 253
適用於 Solaris C++ 多緒應用程式 265

建置檔 (繼續)

適用於 Solaris Micro Focus COBOL 常式 271
適用於 Solaris Micro Focus COBOL 應用程式 268
適用於 Solaris 上的 C 常式 248
適用於 Solaris 上的 C++ 常式 260
適用於 Solaris 上的 C++ 應用程式 256
適用於 Windows IBM COBOL 常式 299
適用於 Windows IBM COBOL 應用程式 297
適用於 Windows Micro Focus COBOL 常式 306
適用於 Windows Micro Focus COBOL 應用程式 303
適用於 Windows 上的 C/C++ 常式 290
適用於 Windows 上的 C/C++ 應用程式 286
HP-UX 上的 C 應用程式 189
HP-UX 上的 C++ 應用程式 201
Linux 上的 C++ 應用程式 232
Solaris 上的 C 應用程式 243
架構檔
適用於 AIX 上的 VisualAge C++ 166
訂購 DB2 書籍 321

〔十劃〕

案例
資料庫管理程式 5

〔十一劃〕

常式
在 AIX 上建置 C 148
在 AIX 上建置 C++ 159
在 AIX 上載入 COBOL 共用檔案庫 143
在 HP-UX 上建置 C 193
在 HP-UX 上建置 C++ 205
在 Linux 上建置 C 225

常式 (繼續)

- 在 Linux 上建置 C++ 235
- 在 Solaris 作業環境上建置 C 248
- 在 Solaris 作業環境上建置 C++ 260
- 在 Windows 建置 C/C++ 290
- 建置 AIX IBM COBOL 175
- 建置 AIX Micro Focus COBOL 183
- 建置 HP-UX Micro Focus COBOL 217
- 建置 JDBC 107
- 建置 Solaris Micro Focus COBOL 271
- 建置 SQLJ 119
- 建置 Windows IBM COBOL 299
- 建置 Windows Micro Focus COBOL 306
- 建置檔 90
- AIX 上的 CREATE 陳述式 142
- AIX 進入點適用於 142
- COBOL 範例程式檔 76
- C/C++ 範例程式檔 65
- JDBC 範例程式檔 70
- SQL 程序範例程式檔 80
- SQLJ 範例程式檔 72
- 常式的進入點, AIX 142
- 教學指導 333
- 移轉
 - 應用程式 44

〔十二劃〕

殘障 332

〔十四劃〕

- 旗號控制, SQL 92 及 MVS Conformance 3
- 疑難排解
 - 線上資訊 331
 - DB2 文件搜尋 329
- 「遠端資料物件」範例程式檔 82

〔十五劃〕

- 範例
 - 程式
 - 支援的語言 53
 - 目錄 53
 - 副檔名 53
 - 設計 58
 - Java 範例目錄適用於 101
 - 範例資料庫
 - 在主電腦系統上建立 41
 - 建立 39
 - 設置 39
 - 連結 43
 - 編目 42
 - 編譯器
 - 使用 AIX IBM COBOL 171
 - 使用 AIX Micro Focus COBOL 179
 - 使用 HP-UX Micro Focus COBOL 213
 - 使用 Solaris Micro Focus COBOL 267
 - 使用 Windows IBM COBOL 296
 - 使用 Windows Micro Focus COBOL 302
 - 建置檔適用於 90
 - AIX 版本支援 9
 - HP-UX 版本支援 10
 - Linux 版本支援 11
 - make 檔適用於 93
 - Solaris 版本支援 13
 - Windows 版本支援 14
 - 線上
 - 說明, 存取 322

〔十七劃〕

- 儲存程序
 - 以 Visual Basic 進行 OLE 自動化 282
 - 以 Visual C++ 進行 OLE 自動化 285
 - 建置 C
 - AIX 148
 - HP-UX 193

儲存程序 (繼續)

- 建置 C (繼續)
 - Linux 225
 - Solaris 作業環境 248
 - Windows 290
- 建置 COBOL
 - AIX IBM 175
 - AIX Micro Focus 183
 - HP-UX Micro Focus 217
 - Solaris Micro Focus 271
 - Windows IBM 299
 - Windows Micro Focus 306
- 建置 C++
 - AIX 159
 - HP-UX 205
 - Linux 235
 - Solaris 作業環境 260
 - Windows 290
- 建置 JDBC 107
- 建置 SQLJ 119
- 建置檔適用於 90
- AIX 上的 VisualAge C++ 架構檔 169
- COBOL 範例程式檔 76
- C/C++ 範例程式檔 65
- JDBC 範例程式檔 70
- SQLJ 範例程式檔 72

A

- ActiveX 資料物件
 - 以 Visual Basic 277
 - 以 Visual C++ 283
 - 範例程式設計 58
- DB2 AD Client 中的支援 3
- Visual Basic 範例程式檔 82
- Visual C++ 範例程式檔 83
- API
 - AIX IBM COBOL 應用程式 172
 - AIX Micro Focus COBOL 應用程式 179
 - AIX 上的 C 應用程式 144
 - AIX 上的 C++ 應用程式 155
 - AIX 上的 VisualAge C++ 架構檔 167
 - COBOL 範例程式檔 76

API (繼續)

- C++ 應用程式 Solaris 作業環境 256
- C/C++ 範例程式檔 65
- HP-UX Micro Focus COBOL 應用程式 214
- HP-UX 上的 C 應用程式 189
- HP-UX 上的 C++ 應用程式 201
- Linux 上的 C 應用程式 221
- Linux 上的 C++ 應用程式 232
- Solaris 作業環境 Micro Focus COBOL 應用程式 268
- Solaris 作業環境上的 C 應用程式 243
- Windows IBM COBOL 應用程式 297
- Windows Micro Focus COBOL 應用程式 303
- Windows 上的 C/C++ 應用程式 286

applet

- 建置 JDBC 105
- 建置 SQLJ 111
- 雜項秘訣 103
- JDBC 範例程式檔 70
- SQLJ 範例程式檔 72

C

C

- 多緒應用程式
 - AIX 153
 - HP-UX 199
 - Linux 230
 - Solaris 作業環境 253
- 版本支援
 - AIX 9
 - HP-UX 10
 - Linux 11
 - Solaris 作業環境 13
 - Windows 14
- 建置檔 90
- 常式
 - 在 AIX 上建置 148
 - 在 HP-UX 上建置 193
 - 在 Linux 上建置 225

C (繼續)

- 常式 (繼續)
 - 在 Solaris 作業環境上建置 248
 - 在 Windows 上建置 290
- 範例程式設計 58
- 範例程式檔 65
- 錯誤檢查公用程式檔案 96
- 應用程式
 - 在 AIX 上建置 144
 - 在 HP-UX 上建置 189
 - 在 Linux 上建置 221
 - 在 Solaris 作業環境上建置 243
 - 在 Windows 上建置 286
- make 檔 93
- CALL 陳述式及儲存程序 137
- CLI 範例程式檔 68
- COBOL 語言
 - 在 AIX 上安裝及執行 143
 - 使用 HP-UX Micro Focus 編譯器 213
 - 建置檔 90
 - 範例程式檔 76
 - 錯誤檢查公用程式檔案 96
 - AIX IBM 常式 175
 - AIX IBM 編譯器 171
 - AIX IBM 應用程式 172
 - AIX Micro Focus 常式 183
 - AIX Micro Focus 編譯器 179
 - AIX Micro Focus 應用程式 179
 - AIX 版本支援 9
 - HP-UX Micro Focus 常式 217
 - HP-UX Micro Focus 應用程式 214
 - HP-UX 版本支援 10
 - make 檔 93
 - Solaris Micro Focus 常式 271
 - Solaris Micro Focus 編譯器 267
 - Solaris Micro Focus 應用程式 268
 - 「Solaris 作業環境」版本支援 13
 - Windows IBM 常式 299
 - Windows IBM 編譯器 296
 - Windows IBM 應用程式 297

COBOL 語言 (繼續)

- Windows Micro Focus 常式 306
- Windows Micro Focus 編譯器 302
- Windows Micro Focus 應用程式 303
- Windows 版本支援 14
- CREATE 陳述式及 AIX 常式 142
- CREATE 陳述式的 EXTERNAL NAME 子句 142
- C++
 - 以 Visual C++ 進行 OLE 自動化 285
 - 多緒應用程式
 - AIX 164
 - HP-UX 211
 - Linux 241
 - Solaris 作業環境 265
 - 安裝有 Visual C++ 的 ADO 應用程式 283
 - 建置檔 90
 - 常式
 - 在 AIX 上建置 159
 - 在 HP-UX 上建置 205
 - 在 Linux 上建置 235
 - 在 Solaris 作業環境上建置 260
 - 在 Windows 上建置 290
 - 範例程式設計 58
 - 範例程式檔 65
 - 錯誤檢查公用程式檔案 96
 - 應用程式
 - 在 AIX 上建置 155
 - 在 HP-UX 上建置 201
 - 在 Linux 上建置 232
 - 在 Solaris 作業環境上建置 256
 - 在 Windows 上建置 286
 - AIX 上的 VisualAge 架構檔 166
 - AIX 版本支援 9
 - HP-UX 版本支援 10
 - Linux 版本支援 11
 - make 檔 93
 - 「Solaris 作業環境」版本支援 13
 - Windows 版本支援 14

D

- DB2 CLI 範例程式檔 68
- DB2 文件搜尋
 - 使用 Netscape 4.x 329
- DB2 的 OLE DB 提供者
 - 用於安裝有 Visual Basic 的 DB2 277
 - 用於安裝有 Visual Basic 的 ODBC 277
 - 安裝有 Visual C++ 的 DB2 283
- DB2 教學指導 333
- DB2 資訊中心 334
- DB2INSTANCE 環境變數 39
- DB2INSTPROF 環境變數 5
- DB2PATH 環境變數 5
- DB2_SQLROUTINE_COMPILE_COMMAND 環境變數 21
- DB2_SQLROUTINE_COMPILER_PATH 環境變數 21

F

- FORTRAN 語言
 - DB2 支援 8

G

- GET ROUTINE CLP 命令 134
- GET_ROUTINE_SAR 內建儲存程序 134

J

- Java
 - 支援, DB2 AD Client 3
 - 建置
 - JDBC applet 105
 - JDBC 常式 107
 - JDBC 應用程式 106
 - SQLj applet 111
 - SQLj 常式 119
 - SQLj 應用程式 110, 113
 - 範例
 - 目錄 101
 - 程式設計 58

Java (繼續)

- 環境設定 20
- AIX JDK 版本支援 9
- AIX 環境設定 25
- applet
 - 使用的重點 103
- HP-UX JDK 版本支援 10
- HP-UX 環境設定 26
- JDBC
 - 範例程式檔 70
- Linux
 - 環境設定 26
 - JDK 版本支援 11
- make 檔 93
- plugin 範例檔 75
- Solaris 作業環境
 - 設定 28
 - JDK 版本支援 13
- SQLj (適用於 Java 的內嵌式 SQL)
 - 範例程式檔 72
- UNIX 環境設定 24
- WebSphere 範例檔 74
- Windows
 - 環境設定 35
 - JDK 版本支援 14
- JDBC
 - 建置 applet 105
 - 建置常式 107
 - 建置應用程式 106
 - 範例程式檔 70
 - applet, 使用的重點 103
 - DB2 AD Client 中的支援 3
- JDK_PATH, 「資料庫管理程式」架構關鍵字 19

K

- KEEPFENCED, 「資料庫管理程式」架構關鍵字 19

M

- make 檔
 - 命令 93
 - 說明 93

N

- NOCONVERT 選項 275

O

- OLE DB 提供者
 - 安裝有 Visual C++ 的 ODBC 283

P

- Perl
 - DB2 支援 8
- PHP, DB2 支援於 8
- PUT ROUTINE CLP 命令 134
- PUT_ROUTINE_SAR 儲存程序 134

R

- RDO 應用程式, 以 Visual Basic 建置 280
- REXX 語言
 - 建置 AIX 應用程式 186
 - 建置 Windows 應用程式 308
 - AIX 版本支援 9
 - DB2 AD Client 中的支援 3
 - DB2 支援 8
 - Windows 版本支援 14

S

- SQL 程序
 - 分送編譯 134
 - 自訂前置編譯及連結選項 132
 - 保持中間檔 131
 - 建立 127
 - 備份及復置 133
 - 與 UNIX 從屬站應用程式一起呼叫 128
 - 與 Windows 從屬站應用程式一起呼叫 129
 - 範例程式檔 80
 - 環境設定 21
 - CALL 陳述式 137
 - UNIX 環境設定 28

SQL 程序 (繼續)

Windows 環境設定 37

SQLJ (適用於 Java 的內嵌式 SQL)

建置檔 90

常式, 建置 119

程式

建置 110

範例程式檔 72

應用程式

建置 113

applet

使用 103

建置 111

DB2 AD Client 中的支援 3

V

Visual Basic

建置 ADO 應用程式 277

建置 RDO 應用程式 280

範例程式設計 58

範例程式檔 82

OLE 自動化 282

Windows 版本支援 14

Visual C++ 範例程式檔 83

W

WCHARTYPE CONVERT 前置編譯

選項 275

wchar_t 資料類型

轉換前置編譯選項 275

Windows Management Instrumentation

(WMI)

範例 277

Windows 上的物件 REXX 308

洽詢 IBM

在美國，請撥以下其中一個電話來連絡 IBM：

- 1-800-237-5511，客戶支援中心
- 1-888-426-4343，取得可用服務選項的資訊
- 1-800-IBM-4YOU (426-4968)，DB2 市場銷售

在加拿大，請撥以下其中一個電話來連絡 IBM：

- 1-800-IBM-SERV (1-800-426-7378)，客戶支援中心
- 1-800-465-9600，取得可用服務選項的資訊
- 1-800-IBM-4YOU (1-800-426-4968)，DB2 市場銷售

若要尋找您所在之國家或地區的 IBM 辦事處，請查閱位於 www.ibm.com/planetwide 網站的「IBM 全球聯絡站名錄 (IBM Directory of Worldwide Contacts)」。

產品資訊

您可以藉由電話或透過全球資訊網 (WWW) www.ibm.com/software/data/db2/udb，取得 DB2 Universal Database 產品的相關資訊。

該網站包含了技術圖書庫、訂購書籍、從屬站下載、新聞群組、修正檔案、新聞及 Web 資源鏈結等最新資訊。

如果您住在美國當地，請撥下列一組電話號碼：

- 1-800-IBM-CALL (1-800-426-2255)，訂購產品或取得一般資訊。
- 1-800-879-2755，訂購出版品。

關於如何聯絡美國以外地區的 IBM 的資訊，請跳至 www.ibm.com/planetwide IBM Worldwide 頁面。

讀者意見表

為使本書盡善盡美，本公司極需您寶貴的意見；懇請您使用過後，撥冗填寫下表，惠予指教。

請於下表適當空格內，填入記號（✓）；我們會在下一版中，作適當修訂，謝謝您的合作！

評估項目	評估意見	備註
正確性	內容說明與實際程序是否符合	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	參考書目是否正確	<input type="checkbox"/> 是 <input type="checkbox"/> 否
一致性	文句用語及風格，前後是否一致	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	實際畫面訊息與本書所提之畫面訊息是否一致	<input type="checkbox"/> 是 <input type="checkbox"/> 否
完整性	是否遺漏您想知道的項目	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	字句、章節是否有遺漏	<input type="checkbox"/> 是 <input type="checkbox"/> 否
術語使用	術語之使用是否恰當	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	術語之使用，前後是否一致	<input type="checkbox"/> 是 <input type="checkbox"/> 否
可讀性	文句用語是否通順	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	有否不知所云之處	<input type="checkbox"/> 是 <input type="checkbox"/> 否
內容說明	內容說明是否詳盡	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	例題說明是否詳盡	<input type="checkbox"/> 是 <input type="checkbox"/> 否
排版方式	本書的形狀大小，版面安排是否方便使用	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	字體大小，顏色編排，是否有助於閱讀	<input type="checkbox"/> 是 <input type="checkbox"/> 否
目錄索引	目錄內容之編排，是否便於查考	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	索引語錄之排定，是否便於查考	<input type="checkbox"/> 是 <input type="checkbox"/> 否
	※評估意見為 "否" 者，請於備註欄說明。	

其他：（篇幅不夠時，請另紙說明。）

上述改正意見，一經採用，本公司有合法之使用及發佈權利，特此聲明。

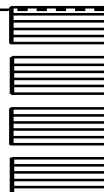
IBM DB2 Universal Database

SC40-0726-00

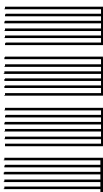
應用程式開發手冊：建置及執行應用程式
版本 8

折疊線

110 台北市基隆路一段二百零六號



臺灣國際商業機器股份有限公司
大中華研發中心 軟體國際部 啟



廣告回信
台灣北區郵政管理局
登記
北台字第 0587 號

(免貼郵票)

寄件人 姓名：
地址：

寄

折疊線



Part Number: CT17WTC

Printed in Singapore

SC40-0726-00



(1P) P/N: CT17WTC

