

Query Management Facility



# QMF for Windows 入門

第 7 版



Query Management Facility



# QMF for Windows 入門

第 7 版

**請注意！**

使用此資訊及其支援的產品之前，請務必閱讀第137頁的『附錄. 注意事項』的一般資訊。

**第五修訂版（2000 年 9 月）**

除非新修訂版另有指示，否則此修訂版套用到 Query Management Facility for Windows（DB2 Universal Database Server for OS/390，第 7 版 (DB2 UDB for OS/390) 5675-DB2 的一個特性）、Query Management Facility (DATABASE 2 Server for VM 和 VSE，第 7 版 (DB2 for VM 和 VSE) 5697-F42 的一個特性）、Query Management Facility for Windows for AS/400 5697-G24、Query Management Facility for Windows for DB2 Workstation Databases 5697-G22、DB2 Warehouse Manager 5648-D35、DB2 Warehouse Manager for AS/400 5697-G23，以及所有後續版次及修正版。

此修訂版會取代並廢止前一版的 SC26-9582-02。此修訂版的技術變更會在變更的左邊以垂直線指示。非技術性的修訂變更不會加註。

# 目錄

QMF 程式庫 . . . . .	vii	第3章 使用提示的查詢 . . . . .	15
第1章 簡介 . . . . .	1	建置簡單查詢 . . . . .	15
資料庫伺服器 . . . . .	1	開啓新的提示查詢 . . . . .	15
DB2 系列資料庫 . . . . .	1	提示查詢動作按鈕 . . . . .	15
使用者名稱對技術名稱 . . . . .	1	將表格新增至提示查詢 . . . . .	16
設定伺服器名稱 . . . . .	1	執行提示查詢 . . . . .	16
資料庫安全 . . . . .	2	建置複雜查詢 . . . . .	16
登入 . . . . .	2	將直欄新增至提示查詢 . . . . .	17
更正密碼 . . . . .	3	使用排序條件 . . . . .	17
變更密碼 . . . . .	3	新增排序條件 . . . . .	18
指定帳號字串 . . . . .	3	使用列條件 . . . . .	18
支配 . . . . .	4	新增列條件 . . . . .	19
檢視資源限制 . . . . .	4	在提示查詢中使用多個表格 . . . . .	20
設定自己的列限制 . . . . .	5	建立提示查詢合併條件 . . . . .	20
自訂工具列 . . . . .	5	提示查詢和 SQL . . . . .	20
新增按鈕到工具列 . . . . .	5	檢視提示查詢的 SQL . . . . .	20
在工具列上移動按鈕 . . . . .	6	將提示查詢轉換成 SQL . . . . .	21
從工具列移除按鈕 . . . . .	6	在提示查詢中使用替代變數 . . . . .	21
第2章 使用 SQL 查詢 . . . . .	7	儲存提示查詢 . . . . .	21
SQL 查詢 . . . . .	7	將提示查詢儲存到檔案 . . . . .	21
建立新的 SQL 查詢 . . . . .	7	開啓已儲存的提示查詢檔案 . . . . .	21
在資料庫伺服器執行 SQL 查詢 . . . . .	7	在資料庫伺服器儲存提示查詢 . . . . .	22
切換結果檢視畫面與 SQL 檢視畫面 . . . . .	7	在資料庫伺服器開啓已儲存的提示查詢 . . . . .	22
使用字型 . . . . .	8	列印提示查詢 . . . . .	23
選取查詢顯示字型 . . . . .	8	預覽提示查詢 . . . . .	23
多重查詢 . . . . .	8	第4章 使用查詢結果 . . . . .	25
同時顯示多重查詢 . . . . .	8	排序和調整查詢結果的大小 . . . . .	25
產生查詢 . . . . .	9	選取直欄和列 . . . . .	25
建立新的 SQL 查詢 . . . . .	9	重新調整直欄和列的大小 . . . . .	25
SQL 查詢中的替代變數 . . . . .	10	自動符合直欄和列 . . . . .	25
使用替代變數執行 SQL 查詢 . . . . .	10	排序查詢結果 . . . . .	26
儲存和開啓 SQL 查詢 . . . . .	11	重新排序直欄 . . . . .	26
將 SQL 查詢儲存到檔案 . . . . .	11	格式化查詢結果 . . . . .	26
開啓已儲存的 SQL 查詢檔 . . . . .	11	選取查詢結果顯示字型 . . . . .	26
在資料庫伺服器儲存 SQL 查詢 . . . . .	11	格式化數值查詢結果 . . . . .	27
在資料庫伺服器開啓已儲存的 SQL 查詢 . . . . .	12	將查詢結果格式轉換成套表 . . . . .	27
列印 SQL 查詢 . . . . .	12	組合和聚集查詢結果 . . . . .	27
預覽查詢 . . . . .	13	組合查詢結果 . . . . .	27
列印 SQL 查詢 . . . . .	13	彙總查詢結果 . . . . .	27
		儲存查詢結果和格式 . . . . .	27

將查詢結果儲存成表格 . . . . .	28	儲存清單至檔案 . . . . .	43
儲存查詢結果到檔案 . . . . .	28	開啓儲存的清單檔案 . . . . .	43
列印查詢結果 . . . . .	28		
預覽查詢結果 . . . . .	28		
列印查詢結果 . . . . .	28		
<b>第5章 使用報告 . . . . .</b>	<b>31</b>	<b>第8章 使用工作檔案 . . . . .</b>	<b>45</b>
套表 . . . . .	31	工作檔案 . . . . .	45
認識套表 . . . . .	31	建立工作檔案 . . . . .	45
使用套表來產生報告 . . . . .	31	執行工作檔案 . . . . .	45
編輯套表 . . . . .	32	自動符合直欄和列 . . . . .	45
建立套表 . . . . .	32	排序查詢結果 . . . . .	45
步驟 1：建立套表 . . . . .	32	重排直欄 . . . . .	46
步驟 2：變更直欄次序 . . . . .	33	格式化查詢結果 . . . . .	46
步驟 3：變更直欄標題 . . . . .	33	選取查詢結果顯示字型 . . . . .	46
步驟 4：變更直欄格式 . . . . .	33	格式化數值查詢結果 . . . . .	46
步驟 5：新增彙總資訊 . . . . .	33	將查詢結果格式轉換成套表 . . . . .	47
步驟 6：新增頁面標頭和標尾 . . . . .	34	組合及聚集查詢結果 . . . . .	47
儲存套表 . . . . .	34	組合查詢結果 . . . . .	47
將套表儲存至檔案 . . . . .	34	彙總查詢結果 . . . . .	47
從檔案中開啓套表 . . . . .	34	儲存查詢結果與格式 . . . . .	47
在資料庫伺服器上儲存套表 . . . . .	35	將查詢結果儲存為表格 . . . . .	47
開啓資料庫伺服器上儲存的套表 . . . . .	35	儲存查詢結果至檔案 . . . . .	48
列印報告 . . . . .	36	列印查詢結果 . . . . .	48
匯出報告 . . . . .	36	預覽查詢結果 . . . . .	48
		列印查詢結果 . . . . .	48
<b>第6章 使用程序 . . . . .</b>	<b>37</b>	<b>第9章 使用靜態查詢 . . . . .</b>	<b>49</b>
執行程序 . . . . .	37	靜態查詢 . . . . .	49
建立新線性程序 . . . . .	37	建立靜態查詢 . . . . .	49
建立含邏輯的新程序 . . . . .	37	將替代變數取代為主電腦變數 . . . . .	50
在資料庫伺服器上執行程序 . . . . .	37	執行靜態查詢 . . . . .	51
儲存程序 . . . . .	38	<b>第10章 使用表格編輯器 . . . . .</b>	<b>53</b>
儲存程序至檔案 . . . . .	38	表格編輯器 . . . . .	53
開啓儲存的程序檔案 . . . . .	38	使用表格編輯器來搜尋列 . . . . .	53
儲存程序至資料庫伺服器 . . . . .	38	新增列 . . . . .	54
開啓資料庫伺服器上儲存的程序 . . . . .	39	變更列 . . . . .	54
列印程序 . . . . .	39	刪除列 . . . . .	54
預覽程序 . . . . .	40	從查詢結果檢視畫面中編輯表格 . . . . .	55
列印程序 . . . . .	40	從查詢結果檢視畫面中刪除列 . . . . .	55
		從查詢結果檢視畫面中更新直欄 . . . . .	55
<b>第7章 使用清單 . . . . .</b>	<b>41</b>	DB2 套表 . . . . .	55
物件 . . . . .	41	<b>第11章 分送資料 . . . . .</b>	<b>57</b>
清單物件 . . . . .	41	匯出資料 . . . . .	57
清單視窗指令 . . . . .	42	匯出資料至檔案 . . . . .	57
建立清單 . . . . .	42	匯入資料 . . . . .	58
新增物件至清單 . . . . .	42	儲存資料至資料庫伺服器 . . . . .	58
從清單中移除物件 . . . . .	43	使用「傳送至」指令 . . . . .	59

使用 Microsoft Excel 載入程式 . . . . .	60	GetColumnHeader() . . . . .	90
使用範例應用程式 . . . . .	60	GetColumnHeaderEx() . . . . .	91
<b>第12章 使用 QMF Report Center . . . . .</b>	<b>61</b>	GetColumnHeadings() . . . . .	92
QMF Report Center 入門 . . . . .	61	GetColumnValue() . . . . .	92
QMF Report Center 視窗 . . . . .	62	GetColumnValueEx() . . . . .	93
連接伺服器 . . . . .	62	GetDefaultServerName() . . . . .	94
使用報告和物件 . . . . .	63	GetGlobalVariable() . . . . .	94
執行報告 . . . . .	63	GetHostVariableNames() . . . . .	94
使用資料夾與我的最愛 . . . . .	64	GetHostVariableTypes() . . . . .	95
新增報告至我的最愛 . . . . .	65	GetLastErrorString() . . . . .	95
<b>第13章 使用 QMF for Windows API . . . . .</b>	<b>67</b>	GetLastErrorType() . . . . .	96
透過 API 控制 QMF for Windows . . . . .	67	GetLastSQLCode() . . . . .	97
鎖住呼叫 . . . . .	67	GetLastSQLException() . . . . .	98
連接資料庫 . . . . .	68	GetLastSQLState() . . . . .	99
API 參考手冊 . . . . .	68	GetOption() . . . . .	99
AddDecimalHostVariable() . . . . .	69	GetOptionEx() . . . . .	100
AddHostVariable() . . . . .	69	GetProcText() . . . . .	101
BindDecimalHostVariable() . . . . .	70	GetProcVariables() . . . . .	101
BindHostVariable() . . . . .	71	GetQMFObjectInfo() . . . . .	102
BindSection() . . . . .	72	GetQMFObjectInfoEx() . . . . .	104
CancelBind() . . . . .	73	GetQMFObjectList() . . . . .	105
ChangePassword() . . . . .	73	GetQMFObjectListEx() . . . . .	106
ClearList() . . . . .	74	GetQMFPProcText() . . . . .	107
Close() . . . . .	74	GetQMFPQueryText() . . . . .	107
Commit() . . . . .	75	GetQueryText() . . . . .	108
CompleteQuery() . . . . .	75	GetQueryVerb() . . . . .	108
CopyToClipboard() . . . . .	76	GetResourceLimit() . . . . .	109
DeleteQMFObject() . . . . .	76	GetResourceLimitEx() . . . . .	112
EndBind() . . . . .	77	GetRowCount() . . . . .	113
Execute() . . . . .	77	GetServerList() . . . . .	113
ExecuteEx() . . . . .	78	GetServerListEx() . . . . .	114
ExecuteStored Procedure() . . . . .	79	GetStoredProcedureResultSets() . . . . .	114
ExecuteStored ProcedureEx() . . . . .	80	GetVariables() . . . . .	115
Export() . . . . .	81	GetVariablesEx() . . . . .	116
ExportForm() . . . . .	83	InitializeProc() . . . . .	116
ExportReport() . . . . .	84	InitializeQuery() . . . . .	117
FastSaveData() . . . . .	85	InitializeServer() . . . . .	118
FetchNextRow() . . . . .	86	InitializeStaticQuery() . . . . .	119
FetchNextRowEx() . . . . .	87	IsStatic() . . . . .	119
FetchNextRows() . . . . .	87	Open() . . . . .	119
FetchNextRowsEx() . . . . .	89	Prepare() . . . . .	120
FlushQMFCache() . . . . .	89	PrintReport() . . . . .	121
GetColumnCount() . . . . .	90	ReinitializeServer() . . . . .	121
GetColumnDataValue() . . . . .	90	Rollback() . . . . .	121
		RunProc() . . . . .	122
		SaveData() . . . . .	122

SaveQMFProc()	124
SaveQMFQuery()	125
SetBindOption()	126
SetBindOwner()	128
SetBusyWindowButton()	128
SetBusyWindowMessage()	129
SetBusyWindowMode()	129
SetBusyWindowTitle()	130
SetGlobalVariable()	131
SetHostVariable()	131
SetOption()	132





SetParent()	133
SetProcVariable()	133
SetVariable()	134
ShowBusyWindow()	134
StartBind()	135

附錄. 注意事項	<b>137</b>
商標	139
索引	<b>141</b>



## QMF 程式庫

請逕向 IBM 業務代表訂購手冊，若在「美國」境內請撥 1-800-879-2755 訂購專線。

評估	Introducing QMF  GC27-0714			
安裝、規劃、管理以及診斷	Installing and Managing QMF on OS/390 GC27-0719	Installing and Managing QMF on VM/ESA GC27-0720	Installing and Managing QMF on VSE/ESA GC27-0721	安裝及管理 QMF for Windows GC40-0530
	QMF Messages and Codes GC27-0717	QMF High Performance Option User's Guide for OS/390 SC27-0724		
使用	Using QMF SC27-0716	QMF Reference SC27-0715	QMF for Windows 入門 SC40-0531	
應用程式設計	Developing QMF Applications SC27-0718			
線上程式庫	 SK2T-0730 OS/390, VM, & VSE	 僅適用 SK2T-6700 OS/390	 僅適用 SK2T-2067 VM	 僅適用 SK2T-0060 VSE



---

## 第1章 簡介

本章提供 QMF for Windows 總覽以及說明 QMF for Windows 入門的一些基本工作。

---

### 資料庫伺服器

在資料庫伺服器執行和儲存查詢、套表、程序和表格。

#### DB2 系列資料庫

QMF for Windows 可連接各類 DB2 資料庫。

- DB2 UDB for OS/390、DB2 for OS/390 及 DB2 for MVS
- DB2 Server for VSE & VM 及 SQL/DS
- DB2 Universal Database 及 DB2 Common Server
- DB2 Parallel Edition
- DataJoiner

QMF for Windows 軟體使用權決定您可以在 QMF for Windows 安裝及連接哪些 DB2 系列產品。

#### 使用者名稱對技術名稱

不同版本和類型的 DB2 會依 RDB 名稱、位置名稱或其他技術名稱來參照資料庫。

透過 QMF for Windows，管理者可指定一個容易記住的名稱給資料庫名稱，例如 Purchasing Database，而不是 DB2P\_01\_PURCH。

QMF for Windows 稱資料庫伺服器或 DB2 資料庫為「伺服器」。

#### 設定伺服器名稱

查詢資料庫之前，QMF for Windows 必須知道儲存資料庫的位置。

1. 從檔案功能表，選取**新增 SQL 查詢**。開啓新的 SQL 查詢文件。

2. 從查詢功能表，選取**設定伺服器**。開啓「設定伺服器」對話框。



3. 從可用的伺服器清單，選取想要查詢的伺服器並按一下**確定**。下一次 QMF for Windows 階段作業開始時，QMF for Windows 會自動重新連接相同的伺服器。

---

## 資料庫安全

您必須提供使用者 ID 和密碼之後才能連接伺服器。

### 登入

您必須指定有效的使用者 ID 和密碼，才能存取伺服器。資料庫伺服器的使用者 ID 和密碼不必與本端或網路的使用者 ID 和密碼相同。

若執行 Windows，那麼您可選擇在 QMF for Windows 階段作業期間記住伺服器密碼。若目前已登入 Windows，「設定使用者資訊」對話框會顯示一個額外的勾選框，叫作**記住這個密碼？**。如選擇此勾選框，那麼存取伺服器的密碼就會儲存在 Windows 密碼清單裡。往後每次登入 Windows 時，QMF for Windows 即自動擷取該密碼，提示您輸入密碼的訊息就不會再出現。若在執行 QMF for Windows 時沒有登入，或以另一位使用者的身份登入，QMF for Windows 就會提示您輸入使用者 ID 和密碼。

**註：**若選擇儲存一個密碼，那麼可登入您的 Windows 帳號的人就能以您（伺服器）使用者 ID 和密碼存取您的資料庫伺服器。

1. 從查詢功能表，選取**設定使用者資訊**。開啓「設定使用者資訊」對話框。



2. 在適當欄位輸入使用者 ID 和密碼。

**註：**使用者 ID 和密碼有區分大小寫。例如，若使用者 ID 或密碼是大寫字體，就必須以大寫字體輸入它。某些類型的資料庫伺服器會區分使用者 ID 和密碼的大小寫；其他則不會。

3. 若要儲存使用者 ID 和密碼，請勾選**記住這個密碼？**。
4. 按一下**確定**。QMF for Windows 就會儲存此資訊來準備存取伺服器。

## 更正密碼

若輸入錯誤密碼，您可以重新開啓「設定使用者資訊」對話框來更正錯誤。

1. 從查詢功能表，按一下**設定使用者資訊**。開啓「設定使用者資訊」對話框。
2. 重新鍵入密碼並按一下**確定**。更正密碼。

## 變更密碼

您可以在資料庫伺服器上從 QMF for Windows 變更密碼。目前僅 DB2 for OS/390 5 版和更新的版本才支援這個功能。

1. 從查詢功能表，選取**設定使用者資訊**。開啓「設定使用者資訊」對話框。
2. 按一下**變更**。顯示新密碼和確認新密碼欄位。
3. 在**新密碼**和**確認新密碼**欄位鍵入新密碼，然後按一下**確定**。變更資料庫伺服器密碼。

## 指定帳號字串

資料庫伺服器使用帳號字串追蹤系統使用情況。詢問資料庫管理者以瞭解您的系統是否使用帳號字串。

1. 從查詢功能表，選取**設定使用者資訊**。開啓「設定使用者資訊」對話框。

2. 在**帳號字串**欄位，鍵入要使用的帳號字串並按一下**確定**。QMF for Windows 就會儲存此資訊來準備存取伺服器。

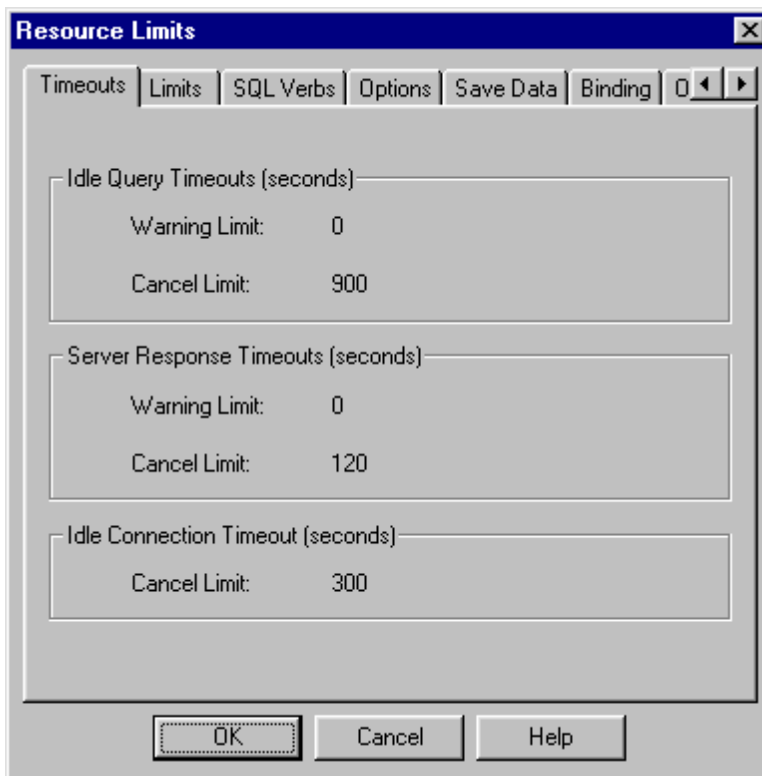
---

## 支配

QMF for Windows 支配程式一律在背景執行，負責監視資料庫和系統資源的使用情況。此支配程式也限制可執行的查詢類型和大小。

### 檢視資源限制

從**檢視功能表**，選取**資源限制**。開啓「資源限制」對話框。「資源限制」對話框中的資訊都是唯讀。而這些限制是由系統管理者來設定。



有效的限制和控制類型如下：

- 逾時
- 限制
- SQL Verb
- 選項

- 儲存資料
- 連結
- 物件追蹤

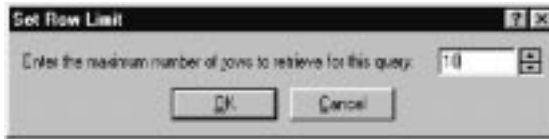
## 設定自己的列限制

您可以指定查詢所能擷取的最大列數。達到這個極限之後，QMF for Windows 會取消查詢。在資源限制群組指定的最大授權列限制優於這個參數。

在這個欄位輸入 0 表示沒有限制。

QMF for Windows 已擷取但超出本限制的列被保留而且可供檢視。

1. 從查詢功能表，按一下**設定列限制**。開啓「設定列限制」對話框。



2. 輸入指定查詢傳回的最大列數並按一下**確定**。下次執行查詢時會套用列限制。

---

## 自訂工具列

您可自訂工具列，只顯示您要查看的按鈕。

### 新增按鈕到工具列

您可選擇將按鈕新增至現有的 QMF for Windows 工具列。不是所有使用者都需要這些按鈕代表的功能，但是使用者可以將這些功能加入工具列。

1. 按兩下工具列周圍的灰色區域。開啓「自訂工具列」對話框。



2. 從可用的**按鈕直欄**，選取要新增的按鈕並按一下**新增**。按鈕會新增至工具列。

3. 新增按鈕之後，請按一下**關閉**。對話框會關閉，而且新的按鈕會新增至工具列。

## 在工具列上移動按鈕

您可選擇在 QMF for Windows 工具列重新排列按鈕。

1. 按兩下工具列周圍的灰色區域。開啓「自訂工具列」對話框。
2. 從**可用的按鈕**直欄，選取要移動的按鈕。
3. 使用**上移**和**下移**按鈕移動工具列內的按鈕。
4. 移動按鈕之後，請按一下**關閉**。對話框會關閉，且按鈕即出現在它們的新位置。

## 從工具列移除按鈕

您可選擇從 QMF for Windows 工具列移除按鈕。

1. 按兩下工具列周圍的灰色區域。開啓「自訂工具列」對話框。
2. 從**可用的按鈕**直欄，選取要移除的按鈕並按一下**移除**。從工具列移除此按鈕。
3. 移除按鈕之後，請按一下**關閉**。對話框會關閉，按鈕即從工具列移除。



---

## 第2章 使用 SQL 查詢

「結構化查詢語言」(SQL) 是使用者與資料庫之間的最基本介面。查詢是以 SQL 撰寫，由資料庫處理。使用者可以使用 SQL 撰寫 QMF for Windows 查詢，或使用「點按方式」方法建立查詢。

---

### SQL 查詢

使用「結構化查詢語言」查詢需要瞭解 SQL 的指令和語法。不熟悉 SQL 的使用者應該嘗試建立提示的查詢。

#### 建立新的 SQL 查詢

在工具列按一下**新增 SQL 查詢**按鈕。



開啓新的查詢文件。

#### 在資料庫伺服器執行 SQL 查詢

1. 開啓新查詢文件並鍵入查詢，或開啓現存的查詢檔，或從資料庫開啓查詢。
2. 在工具列按一下**執行查詢**按鈕。



執行查詢並顯示結果。

#### 切換結果檢視畫面與 SQL 檢視畫面

您可查看查詢結果或 SQL 陳述式本身。

從已執行的查詢的 SQL 檢視畫面，按一下工具列上的**檢視結果**按鈕。



顯示查詢結果。

- 或 -

從查詢的「結果」檢視畫面，按一下**檢視 SQL** 按鈕。



顯示 SQL 陳述式。

---

## 使用字型

您可變更有來顯示查詢的字型。字型選項根據您在電腦上安裝的字型而有不同。關於新增字型的詳細資訊，請參閱作業系統的說明機能。

**註：**若選取新的查詢顯示字型之後儲存該查詢，就一定會使用新字型顯示該查詢。

### 選取查詢顯示字型

1. 從 SQL 檢視畫面，在**查詢**功能表按一下**設定字型**。開啓「字型」對話框。
2. 選取顯示查詢文字的字型，然後按一下**確定**。以新字型重新顯示查詢。

**註：**按一下**設為預設值**使用選取的字型作為新查詢的預設字型。

---

## 多重查詢

您可以同時開啓一個以上的查詢文件。也可以同時執行一個以上的查詢。您可使用這個特性以產生多重報告，或從一個查詢剪下 SQL 文字並貼到另一個查詢。

### 同時顯示多重查詢

1. 開啓至少兩個查詢文件。
2. 從**視窗**功能表，選取下列其中一個指令：

指令	結果
階式排列	以交錯系列顯示查詢。
水平並排	垂直堆疊顯示查詢視窗。
垂直並排	並列堆疊顯示查詢視窗。

根據選取的選項排列查詢視窗。

## 產生查詢

使用「產生查詢」指令建立新的 SQL 查詢文件。您指定一個或多個想要的表格名稱和 SQL 文件類型，QMF for Windows 會自動建立一個 SQL 陳述式，該陳述式參照表格的直欄名稱和資料類型。

### 建立新的 SQL 查詢

1. 從檔案功能表，按一下**產生查詢**。開啓「產生查詢」對話框。



2. 選取要建立的查詢類型：

查詢類型	結果
選取	從一個或多個表格擷取列。
更新	變更表格中的資訊。
插入	將新列新增至表格。

3. 輸入要查詢的表格的擁有者和名稱。

**註：** 您可使用型樣從符合的表格清單選取表格名稱。

- 使用百分比字元 (%)，使它符合含有任何字元的任何長度之字串。例如，若要列示含有以字母 A 為名稱開頭的表格，請輸入 A%。
- 使用底線字元 (\_)，使它符合單一字元。例如，若要列示含有在第二個位置有字母 A 的擁有者的表格，請輸入 \_A%。

輸入一個型樣之後，請按一下**從清單新增**，然後從結果清單中選取表格。

4. 輸入表格的唯一識別碼。
5. 按一下**新增**。表格新增至查詢。
6. 新增想要查詢的表格之後，請按一下**確定**。建立及顯示所選取表格的 SQL 查詢。

---

## SQL 查詢中的替代變數

有了替代變數之後，如在每次執行查詢時提供不同的值，即可使用相同查詢以擷取不同資訊。若要擷取另一個資料集，您不必重寫查詢。只要在執行查詢時在查詢中提供不同值給替代變數即可。

替代變數是您併入查詢的文字。它必須以 `&` 符號字元開頭，最多可包含 18 個字元，可以用字母、數字或下列特殊字元：`^ ! $ % & ' { } ? @ # % \` 或 `_`。例如，下列值是有效的替代變數：

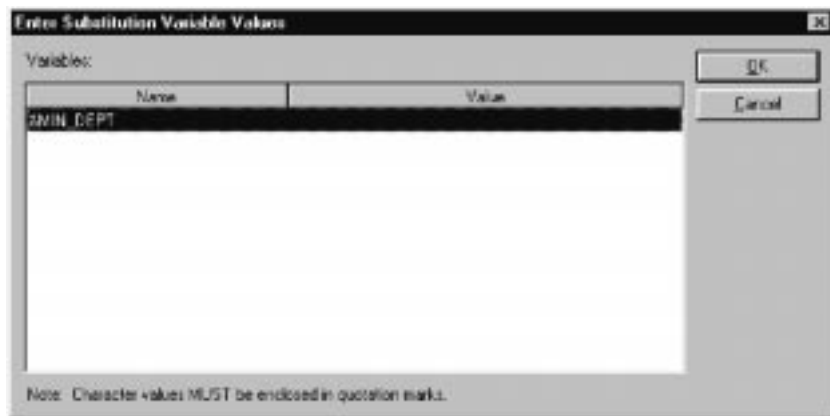
```
&VARIABLE1  
&DEPARTMENT_NUMBER
```

替代變數可出現在查詢中的任何位置，而且它的值是您可以在查詢中寫入的值（除了註解以外）。例如，您可使用替代變數代替直欄名稱、搜尋條件、子查詢或任何特定值。

### 使用替代變數執行 SQL 查詢

1. 開啓新的查詢文件並輸入這個 SQL 陳述式：  

```
SELECT * FROM Q.STAFF WHERE DEPT >= &MIN_DEPT
```
2. 執行查詢。開啓「輸入替代變數值」對話框。



3. 在**值**欄位輸入值 `50` 並按一下**確定**。執行查詢並顯示查詢結果。

嘗試以替代變數取代 `SELECT` 和 `FROM` 子句中的值。看看使用不同輸入之後查詢傳回什麼結果。

---

## 儲存和開啓 SQL 查詢

您可將查詢儲存在 PC、檔案伺服器或資料庫伺服器。

### 將 SQL 查詢儲存到檔案

1. 從開啓的查詢，按一下工具列上的**儲存檔案**按鈕。



若該查詢已儲存，會再儲存一次。若先前沒有儲存過，就會開啓「另存新檔」對話框。

2. 輸入要儲存查詢的檔案名稱並按一下**確定**。儲存查詢。

### 開啓已儲存的 SQL 查詢檔

1. 按一下工具列上的**開啓舊檔**按鈕。



開啓「開啓舊檔」對話框。

2. 選取要開啓的檔案並按一下**確定**。以新的查詢文件開啓選取的查詢。

### 在資料庫伺服器儲存 SQL 查詢

將查詢儲存在伺服器上便可供其他使用者存取。若要與其它使用者共用查詢，請將它們儲存在資料庫伺服器。

1. 從開啓的查詢，按一下工具列上的**儲存在伺服器**按鈕。



開啓「儲存查詢」對話框。



2. 輸入擁有者、名稱，選擇是否要與其它使用者共用已儲存的查詢，並按一下**確定**。查詢即儲存在伺服器。

若同名的查詢已存在，就會提示您改寫先前的查詢。

## 在資料庫伺服器開啓已儲存的 SQL 查詢

您可以開啓已儲存在資料庫伺服器的查詢。

1. 按一下工具列上的**從伺服器開啓**按鈕。



開啓「從伺服器開啓」對話框。



2. 輸入伺服器、擁有者和名稱，然後按一下**確定**。開啓 SQL 查詢。

---

## 列印 SQL 查詢

您可預覽和列印 SQL 查詢。

## 預覽查詢

1. 開啓查詢並啓動 SQL 檢視畫面。顯示 SQL 陳述式。
2. 從**檔案**功能表，按一下**頁面設定**。開啓「頁面設定」對話框。
3. 變更頁面佈置後，按一下**確定**。
4. 按一下工具列上的**列印預覽**按鈕。



顯示列印的查詢預覽。

## 列印 SQL 查詢

1. 開啓查詢並啓動 SQL 檢視畫面。顯示 SQL 陳述式。
2. 從**檔案**功能表，按一下**頁面設定**。開啓「頁面設定」對話框。
3. 變更頁面佈置後，按一下**確定**。
4. 按一下工具列上的**列印**按鈕。



列印查詢。





## 第3章 使用提示的查詢

提示查詢是讓您從功能表和清單選取選項以建立查詢的簡單方法。建立提示查詢之後，您可儲存它或把它轉換為 SQL 查詢。

### 建置簡單查詢

您可使用提示查詢介面輕易建置簡單查詢。

#### 開啓新的提示查詢

- 從檔案功能表，按一下**新的提示查詢**。開啓新的提示查詢文件。



#### 提示查詢動作按鈕

使用查詢動作按鈕編輯提示查詢。該組按鈕控制的區段上面就會出現一組按鈕。

提示查詢動作按鈕	外觀	結果
新增		按一下，新增項目至提示查詢。
編輯		按一下，編輯查詢中強調顯示的項目。
刪除		按一下，刪除選取的項目。
上移和下移		按一下，在提示查詢中上下移動選取的項目。

## 將表格新增至提示查詢

1. 在提示查詢文件的「表格」區段，按一下**新增**按鈕。



開啓「表格」對話框。



2. 鍵入要新增的表格的擁有者和名稱並按一下**新增**。表格新增至查詢。

**註：** 您可使用型樣從符合的物件清單選取物件。

- 使用百分比字元 (%)，使它符合含有任何字元的任何長度之字串。例如，若要列示含有以字母 A 為名稱開頭的表格，請輸入 A%。
- 使用底線字元 (\_)，使它符合單一字元。例如，若要列示含有在第二個位置有字母 A 的擁有者的表格，請輸入 \_A%。

輸入一個型樣之後，請按一下**從清單新增**，然後從結果清單選取表格。

3. 將任何其它的表格條件新增至查詢並按一下**關閉**。顯示提示查詢文件以及已列示的新表格。

## 執行提示查詢

執行提示查詢和執行 SQL 查詢的方法一樣。按一下工具列上的「執行查詢」按鈕。



執行提示查詢。

---

## 建置複雜查詢

您也可使用提示查詢介面建置更複雜的查詢。

## 將直欄新增至提示查詢

1. 在「提示查詢」文件的「直欄」區段，按一下**新增**按鈕。



開啓「直欄」對話框。



2. 選取要新增的直欄並按一下**新增**。直欄會新增至提示查詢。
3. 將其它直欄都新增至查詢並按一下**關閉**。顯示「提示查詢」文件以及已列示的新直欄。

**註：** 您可在「函數」欄位選取一個彙總函數，將該函數套用到直欄。可用的彙總函數包括：AVERAGE、COUNT、MAXIMUM、MINIMUM 和 SUM。

**註：** 您可以在**新直欄名稱**欄位鍵入新的直欄名稱，重新命名查詢中的直欄。

## 使用排序條件

排序條件用於指定在查詢中要如何排序列。您可以採取升冪 (A-Z) 或降冪 (Z-A) 順序排序列。

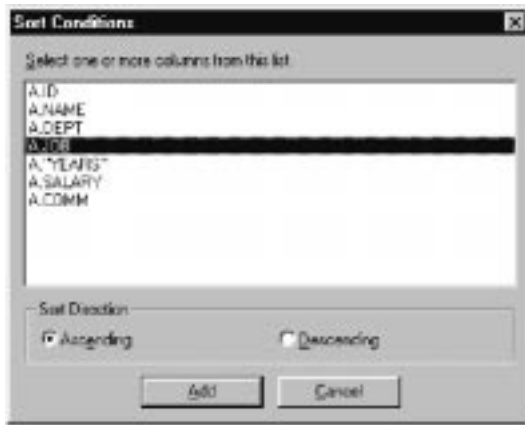
若根據一個以上的直欄排序列，首先會排序第一個直欄，然後在第一個直欄次序內排序第二個直欄，依此類推。

## 新增排序條件

1. 在「提示查詢」文件的「排序條件」區段，按一下**新增**按鈕。



開啓「排序條件」對話框。



2. 選取要作為排序依據的直欄、排序的方向，然後按一下**新增**。排序條件新增至提示查詢。
3. 將任何其他的排序條件新增至查詢並按一下**關閉**。顯示「提示查詢」文件以及已列示的新排序條件。

## 使用列條件

很多時候，您也許只想要檢視表格的某些列。欲選取特定列來檢視，請新增列條件。若不使用列條件，就會顯示表格的全部列。

可使用下列列條件：

- Equal to
- Less than
- Less than or equal to
- Greater than
- Greater than or equal to
- Between
- Starting with
- Ending with

- Containing
- NULL

下列運算子控制列條件：

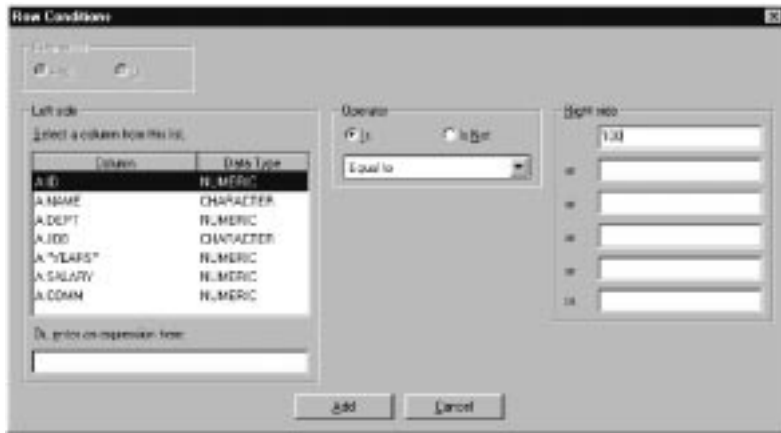
- Is
- Is Not

## 新增列條件

1. 在「提示查詢」文件的「列條件」區段，按一下**新增**按鈕。



開啓「列條件」對話框。



2. 選取條件陳述式的一部份並按一下**新增**。

列條件的一部份	函數
左側	選取要查驗的直欄。
運算子	決定列的左側與右側之間的關聯。
右側	輸入要檢查的條件。

列條件會新增至提示查詢。

3. 將任何其他列條件新增至查詢並按一下**關閉**。顯示「提示查詢」文件以及已列示的新列條件。

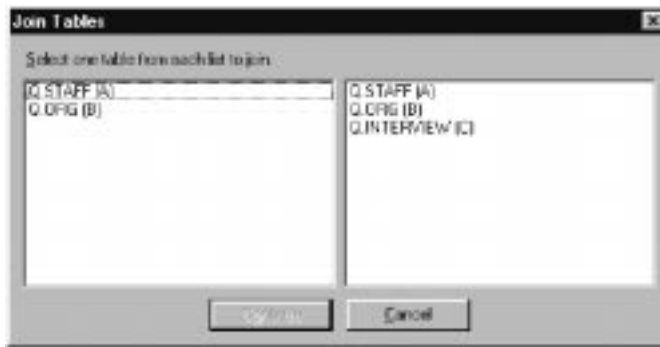
## 在提示查詢中使用多個表格

在提示查詢中您可以納入來自一個以上的表格的資訊。

您必須使兩個表格產生關聯，方法是在這兩個表格中指定一個或多個合併條件。有相等的合併直欄的表格中的列才會被併入結果中。合併條件中，每一個直欄的資料類型必須相符。指定兩個直欄之間的關聯之後，QMF for Windows 就會記住此關聯，並在以後的查詢中加以提醒，使建立後續查詢變得更簡單和有效。

### 建立提示查詢合併條件

1. 在「提示查詢視窗」的「表格」區段，按一下**新增**按鈕，新增至少兩個表格。若之前不曾過合併表格，就會開啓「合併表格」對話框。若之前有合併，QMF for Windows 會建議先前使用的合併條件。



2. 從每一個表格選取一個具有相同資料類型的直欄並按一下**新增**。新的合併條件出現在提示查詢。

---

## 提示查詢和 SQL

您可使用提示查詢介面學習 SQL。

### 檢視提示查詢的 SQL

從提示查詢檢視畫面，按一下工具列上的**檢視 SQL** 按鈕。



顯示提示查詢的相等 SQL 陳述式。您無法從這個檢視畫面修改 SQL 陳述式。

## 將提示查詢轉換成 SQL

提示查詢可轉換成新的 SQL 查詢文件。您可以修改、儲存、列印和執行新的 SQL 查詢。從查詢功能表，按一下**轉換成 SQL**。查詢轉換成新的 SQL 查詢文件。

---

### 在提示查詢中使用替代變數

您可採取在 SQL 查詢中使用替代變數的方法，在提示查詢中使用替代變數。請參閱「SQL 查詢中的替代變數」。

例如，替代變數可用於：

- 列條件  
DEPT Is Greater Than Or Equal To &MinDept
- 直欄規格  
&InputNum

---

### 儲存提示查詢

提示查詢可以儲存在 PC 或伺服器資料庫上的檔案裡。

#### 將提示查詢儲存到檔案

1. 從開啓的提示開啓，按一下工具列上的**儲存檔案**按鈕。



**註：**若該查詢已儲存，會再儲存一次。若先前沒有儲存過，就會開啓「另存新檔」對話框。

2. 輸入要儲存提示查詢的檔名並按一下**確定**。儲存查詢。

#### 開啓已儲存的提示查詢檔案

1. 按一下工具列上的**開啓舊檔**按鈕。



開啓「開啓舊檔」對話框。

2. 選取要開啓的檔案並按一下**確定**。以新的查詢文件開啓選取的提示查詢。

## 在資料庫伺服器儲存提示查詢

1. 從開啓的提示查詢，按一下工具列上的**儲存在伺服器**按鈕。



開啓「儲存查詢」對話框。



2. 輸入擁有者、名稱，選擇是否要與其它使用者共用已儲存的查詢，並按一下**確定**。查詢即儲存在伺服器。

若同名的查詢已存在，就會提示您改寫現存的查詢。

## 在資料庫伺服器開啓已儲存的提示查詢

您可開啓已儲存在資料庫伺服器的提示查詢。

1. 按一下工具列上的**從伺服器開啓**按鈕。



開啓「從伺服器開啓」對話框。



2. 輸入伺服器、擁有者和名稱，然後按一下**確定**。開啓提示查詢。



---

## 列印提示查詢

您可列印提示查詢，也可列印提示查詢的 SQL 文字。請參閱第12頁的『列印 SQL 查詢』。

### 預覽提示查詢

在列印前，您可預覽提示查詢的結果或文字。

1. 開啓查詢並啓動提示檢視畫面。顯示查詢。
2. 從**檔案**功能表，按一下**頁面設定**。開啓「頁面設定」對話框。
3. 變更頁面佈置後，按一下**確定**。
4. 按一下工具列上的**列印預覽**按鈕。



顯示列印的查詢預覽。



## 第4章 使用查詢結果

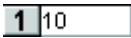



您可以直接在查詢結果中套用格式、組合和聚集。此格式可以隨查詢一起儲存，或匯出成爲套表。

### 排序和調整查詢結果的大小

使用者可從查詢選取、重新調整大小、重排次序以及排序資料結果。

#### 選取直欄和列

執行查詢之後，您就可以使用「結果」檢視畫面中的控制項編輯和選取資訊。

直欄和列選取元	外觀	函數
列選取元		按一下，選取一列中的全部資料。
直欄選取元		按一下，選取一個直欄中的全部資料。
資料格		直接按一下，選取資料格。
捲動至底端按鈕或捲動至頂端按鈕		按一下，捲動至一組查詢結果的頂端或底端。

#### 重新調整直欄和列的大小

您可重新調整一組查詢結果的直欄和列的大小，來變更該組查詢結果的外觀。

1. 使用滑鼠選取兩個直欄或兩個列之間的黑色分隔線。
2. 左右或上下拖曳分隔線重新調整直欄或列的大小。

**註：**若在重新調整查詢的列或直欄大小後儲存查詢，就一定會使用新格式顯示。

#### 自動符合直欄和列

您可自動調整直欄和列的大小以符合它們包含的資料。

使用滑鼠時，選取整個直欄或列，並按兩下它與相鄰物件之間的黑色分隔線。自動重新調整直欄或列的大小以符合它的資料。

**註:** 若重新調整查詢的列或直欄之後儲存查詢，就一定會使用新格式顯示查詢。

## 排序查詢結果

執行查詢之後，您就可以依照直欄按字母順序排序結果。

從查詢的「結果」檢視畫面，選取一個直欄並從**結果**功能表選取**升冪排序**。

依升冪次序排序查詢結果。

- 或 -

從查詢的「結果」檢視畫面選取一個直欄，然後從**結果**功能表選取選取**降冪排序**。

依降冪次序排序查詢結果。

**註:** 若要對選取的直欄套用更複雜的排序，請從**結果**功能表選取**排序**。

## 重新排序直欄

您可以在查詢結果中變更直欄次序。

從查詢的「結果」檢視畫面，選取一個直欄並將它拖曳到新位置。

依新次序顯示直欄。

---

## 格式化查詢結果

您可變更加用於顯示查詢和查詢結果的字型。字型選項根據您在電腦上安裝的字型而有不同。關於新增字型的詳細資訊，請參閱作業系統的說明機能。

**註:** 若在選取新的查詢結果顯示字型之後儲存查詢，就一定會使用新字型顯示那些結果。

## 選取查詢結果顯示字型

1. 從「結果」檢視畫面，從**結果**功能表選取**設定字型**。開啓「字型」對話框。
2. 選取此字型並鍵入顯示查詢結果的大小，然後按一下**確定**。以指定的格式顯示查詢結果。

**註:** 按一下**設為預設值**，使用選取的字型作為查詢結果的預設字型。

## 格式化數值查詢結果

1. 從「結果」檢視畫面選取一個含有數值的直欄，然後從**結果**功能表選取**格式**。開啓「格式」對話框。
2. 指定要套用的格式並按一下**確定**。根據您的選擇格式化值。

**註：**按一下**設為預設值**，使用選取的字型作為查詢結果的預設字型。

## 將查詢結果格式轉換成套表

您可將查詢結果格式轉換成套表。

1. 從**結果**功能表，選取**顯示報告**。開啓「選取套表」對話框。
2. 選取**從查詢**並按一下**確定**。查詢結果格式轉換成套表，並以新的套表視窗開啓它。

---

## 組合和聚集查詢結果

您可對查詢結果套用組合、聚集和彙總格式。

### 組合查詢結果

您可以選擇使用或不使用彙總資訊組合查詢結果。

1. 選取要組合的直欄。
2. 從**結果**功能表，選取要套用的組合類型。根據您的選擇組合直欄。

### 彙總查詢結果

您可依直欄彙總查詢結果。

1. 選取要組合的直欄。
2. 從**結果**功能表，選取要套用的彙總類型。根據您的選擇彙總直欄。

---

## 儲存查詢結果和格式

您可將查詢結果和格式儲存成套表。

## 將查詢結果儲存成表格

您可在資料庫伺服器將查詢結果儲存成表格。

1. 從**結果**功能表，選取**儲存到資料庫**。  
開啓「儲存資料」對話框。
2. 輸入擁有者和表格名稱並按一下**確定**。  
在資料庫上將查詢結果儲存成表格。

## 儲存查詢結果到檔案

您可將查詢結果儲存到 PC 或檔案伺服器上的檔案。

1. 從**結果**功能表，選取**儲存到檔案**。  
開啓「匯出資料」對話框。
2. 指定要儲存檔案的位置、匯出選項，然後按一下**確定**。  
查詢結果儲存到檔案。

---

## 列印查詢結果

您可預覽和列印查詢結果。

### 預覽查詢結果

1. 開啓和執行查詢。顯示查詢結果。
2. 從**檔案**功能表，選取**頁面設定**。開啓「頁面設定」對話框。
3. 變更頁面佈置後，按一下**確定**。
4. 按一下工具列上的**列印預覽**按鈕。



顯示列印的查詢結果預覽。

### 列印查詢結果

1. 開啓查詢並啓動「結果」檢視畫面。顯示查詢結果。
2. 從**檔案**功能表，選取**頁面設定**。開啓「頁面設定」對話框。
3. 變更頁面佈置後，按一下**確定**。

4. 按一下工具列上的**列印**按鈕。



列印查詢結果。





---

## 第5章 使用報告

報告是由套表格式與查詢結果的組合建立。

---

### 套表

套表為用來建立、顯示及列印報告的格式化指令集。

#### 認識套表

套表由一些元件組合而成。這些元件皆可以套表文件來編輯。

**主要** 套表的主要元件包括標頭、表尾及中斷。

**中斷** 報告中最多 6 個小計行的性質、內容及位置。

**計算** 報告計算表示式。

**註：** 您的機器上必須安裝 IBM 的 ObjectREXX 才能使用套表計算。

**直欄** 報告中直欄的格式與外觀。可定義的性質包括直欄次序、格式、用法、內縮及寬度。

**條件** 條件式格式限制。例如，您可以設定套表不顯示與某些性質不符的列。

#### 詳細資料

報告的詳細標題及主體文字。您可以在此以開放式版面文字結合或取代列表格式資料，以建立格式信件或位址標籤。

**最終** 報告的最終文字之內容與位置。例如，您可選擇在報告尾端併入最終文字和摘要資料。

**HTML** HTML 報告中 HTML 標記與格式的內含及位置。

**選項** 報告的雜項外觀選項。

**頁面** 報告上頁面標頭與標尾的內含及位置。

#### 使用套表來產生報告

報告是由套表內含的格式選項及查詢結果的組合建立。您可以重複此程序，從單一查詢結果集產生多重報告。

1. 在查詢結果檢視畫面中，按一下**顯示報告**按鈕。



開啓「選取套表」對話框。



2. 根據「選取套表」對話框中選取的套表類型，會要求您提供額外的資訊。請適當指定檔案位置、擁有者及名稱，或文件標題，然後按一下**確定**。報告即以選定的套表及現行查詢結果產生。

## 編輯套表

「套表」視窗提供許多用來編輯及格式化套表的選項。

從開啓的套表中，顯示「套表」功能表。「套表」功能表列示供您編輯及格式化套表的全部選項。您亦可按一下工具列上對應的按鈕來編輯這些元件。

---

## 建立套表

這些步驟皆包括表格 Q.STAFF 中的範例資料。請嘗試用不同的設定來建立您的自訂套表。

### 步驟 1：建立套表

1. 執行下列 SQL 查詢來擷取報告中顯示的資料：

```
SELECT * FROM Q.STAFF ORDER BY DEPT, NAME
```

顯示查詢結果。

2. 按一下工具列上的**顯示報告**按鈕。即開啓「選取套表」對話框。
3. 指定預設套表，然後按一下**確定**。QMF for Windows 顯示預設報告。若要變更預設格式，請按一下工具列上其中一個套表元件。套表工具列上會顯示每一個套表元件的按鈕。

## 步驟 2：變更直欄次序

我們要將 NAME 當作報告中第一個直欄，將 ID 當作第二個直欄。直欄次序可在套表的「直欄」元件中指定。

1. 按一下**套表**功能表的**直欄 ...**，以顯示「套表」對話框的「直欄」標籤。
2. 鍵入新的次序值來覆蓋現存的次序值，以變更直欄次序。要使 NAME 成為報告的第一個直欄，請將其序號（清單中標示為 Seq 的直欄）變更為 1。
3. 要使 ID 成為報告的第二個直欄，請將其序號變更為 2，然後按一下**確定**。QMF for Windows 會在「套表」視窗中以新直欄次序顯示報告。

## 步驟 3：變更直欄標題

我們要使用 EMPLOYEE 作為第一個直欄標題，使用 COMMISSION 作為第二個直欄標題。直欄標題文字可在套表的「直欄」元件中指定。

1. 按一下**套表**功能表的**直欄 ...**，以顯示「套表」對話框的「直欄」標籤。
2. 鍵入新的直欄標題文字來覆蓋現存的直欄標題文字，以變更直欄標題。將第一個直欄標題變更為 EMPLOYEE，最後一個直欄標題變更為 COMMISSION，然後按一下**確定**。QMF for Windows 會在「套表」視窗中以新直欄標題顯示報告。

## 步驟 4：變更直欄格式

我們要以適當的貨幣符號顯示 SALARY 直欄。直欄格式取決於其編輯碼，可在套表的「直欄」元件中指定。

1. 按一下**套表**功能表的**直欄 ...**，顯示「套表」對話框的「直欄」標籤。
2. 鍵入新的編輯碼來覆蓋現存的編輯碼，將 SALARY 直欄編輯碼變更為 D2，然後按一下**確定**。QMF for Windows 會在「套表」視窗中以使用適當貨幣符號的 SALARY 直欄顯示報告。

## 步驟 5：新增彙總資訊

我們要針對各部門將報告分成不同的區段。此外，我們要在每一區段的尾端查看各部門的 SALARY 及 COMMISSION 總計。若要這樣做，則必須指定報告中每一個直欄的用法。直欄用法取決於其語法碼，可在套表的「直欄」元件中指定。

1. 按一下**套表**功能表的**直欄 ...**，顯示「套表」對話框的「直欄」標籤。
2. 若要依據 DEPT 將報告分成若干區段，請將語法碼 DEPT 變更為 BREAK1。開頭為 BREAK 這個字的語法碼會對指定的直欄產生一個區段中斷。BREAK 後面的數字可決定中斷層次；一個報告最多支援 6 個中斷層次。
3. 若要對每一個 DEPT 併入 SALARY 與 COMMISSION 總計，請將 SALARY 及 COMMISSION 語法碼變更為 SUM。

4. 如果也在每一個區段中斷尾端併入說明資訊，則報告更容易瞭解。若要這樣做，請按一下**套表功能表的中斷 ...**。
5. 您可以在「套表」對話框的「中斷」標籤上指定中斷標尾文字。將第一個中斷標尾行設定為「部門總計」，然後按一下**確定**。QMF for Windows 會顯示「套表」視窗。

## 步驟 6：新增頁面標頭和標尾

我們要新增頁面標頭和標尾到報告中。頁面標頭和標尾可在套表的「頁面」元件中指定。

1. 按一下**套表功能表的頁面 ...**，顯示「套表」對話框的「頁面」標籤。
2. 此對話框上半部用來指定頁面標頭。將頁面標頭第一行設定為「部門報告」，將第二行設定為「薪資與佣金總計」。選擇標頭的對齊方式。
3. 此對話框下半部用來指定頁面標尾。將頁面標尾第一行設定為「頁面結尾」。選擇標尾的對齊方式，然後按一下**確定**。QMF for Windows 會顯示「套表」視窗。

---

## 儲存套表

您可以將套表儲存到 PC、檔案伺服器或資料庫伺服器上。

### 將套表儲存至檔案

1. 從開啓的套表中，按一下**儲存**按鈕。
2. 如果此套表已儲存過，請選取**儲存**。如果此套表先前未儲存過，則會開啓「另存新檔」對話框。
3. 輸入用來儲存套表的檔名，然後按一下**確定**。套表已儲存。

### 從檔案中開啓套表

1. 按一下工具列上的**開啓舊檔**按鈕。



開啓「開啓舊檔」對話框。

2. 選取要開啓的檔案並按一下**確定**。選取的套表會在新的套表文件中開啓。

## 在資料庫伺服器上儲存套表

儲存於伺服器的套表可供其他使用者存取。如果要與其他使用者共用您的套表，請將這些套表儲存在資料庫伺服器上。

1.

從開啓的套表中，按一下工具列上的**儲存於伺服器**按鈕。



開啓「儲存套表」對話框。



2. 輸入擁有者、名稱，選擇要不要與其他使用者共用儲存的套表，然後按一下**確定**。套表即儲存到伺服器上。

如果指定的套表名稱已存在，則會提示您改寫現存的套表。

## 開啓資料庫伺服器上儲存的套表

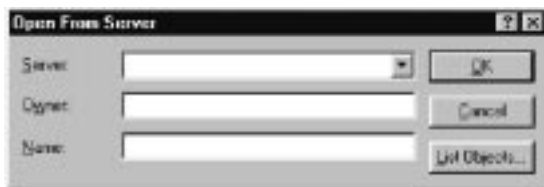
您可以開啓儲存於資料庫伺服器上的套表。

1.

按一下工具列上的**從伺服器開啓**按鈕。



開啓「從伺服器開啓」對話框。



2. 輸入伺服器、擁有者及名稱，然後按一下**確定**。開啓套表。

---

## 列印報告

您可以產生列印的報告。

1. 開啓套表，然後按一下**頁面設定**。
2. 變更頁面佈置後，按一下**確定**。
3. 按一下**檔案功能表**的**列印報告**。  
開始列印報告。

---

## 匯出報告

您可以將報告匯出至檔案。

1. 開啓套表，然後按一下**頁面設定**。
2. 變更頁面佈置後，按一下**確定**。
3. 按一下**檔案功能表**的**匯出報告**。開啓「匯出報告」對話框。



4. 輸入用來儲存報告的檔名，然後按一下**確定**。匯出報告。

---

## 第6章 使用程序

線性程序可讓您執行單一指令來執行查詢、產生報告、編輯資料及執行其它函數。關於 QMF for Windows 支援的所有程序指令之完整清單，請參閱線上說明機能。

含邏輯的程序或 REXX 程序類似線性程序，但有包含 IBM 的 Object REXX 程式設計語言及程序指令。Object REXX 必須在本端安裝才能執行含邏輯的程序。

---

### 執行程序

程序是用來以一個指令執行多重函數。

#### 建立新線性程序

從檔案功能表中，選取**新程序**。

開啓一個新程序文件。

#### 建立含邏輯的新程序

1. 從檔案功能表中，選取**新程序**。  
開啓一個新程序文件。
2. 鍵入 REXX 註解行作為程序的第一行。REXX 註解行的開頭為 /\*，結尾為 \*/
3. 在程序中鍵入任何您要的 QMF 程序指令。QMF 指令必須以大寫輸入，而且要用引號括住。
4. 在程序中鍵入任何您要的 REXX 指令。

**註:** REXX 指令要在本端執行，不能在資料庫伺服器上執行。Object REXX 必須安裝在本端。

#### 在資料庫伺服器上執行程序

1. 開啓一個新程序文件並鍵入一組指令，或從檔案或資料庫伺服器來開啓現存的程序。
2. 按一下工具列上的**執行程序**按鈕。



程序開始執行。

---

## 儲存程序

您可以將程序儲存到 PC、檔案伺服器或資料庫伺服器上。

### 儲存程序至檔案

1. 從開啓的程序中，按一下工具列上的**儲存**按鈕。



如果程序已儲存過，則會直接儲存。如果先前未儲存過，則會開啓「另存新檔」對話框。

2. 輸入用來儲存程序的檔名，然後按一下**確定**。即儲存程序。

### 開啓儲存的程序檔案

1. 按一下工具列上的**開啓舊檔**按鈕。



出現「開啓舊檔」對話框。

2. 選取要開啓的檔案並按一下**確定**。選取的程序會在新程序文件中開啓。

### 儲存程序至資料庫伺服器

1. 從開啓的程序中，按一下工具列上的**儲存於伺服器**按鈕。





開啟「儲存程序」對話框。



2. 輸入擁有者、名稱，選取是否與其他使用者共用儲存的程序，然後按一下**確定**。程序即儲存到伺服器上。

如果指定的程序名稱已存在，則會提示您改寫現存的程序。

## 開啟資料庫伺服器上儲存的程序

您可以開啟儲存於資料庫伺服器上的程序。

- 1.

按一下工具列上的**從伺服器開啟**按鈕。



開啟「從伺服器開啟」對話框。



2. 輸入伺服器、擁有者及名稱，然後按一下**確定**。開啟程序。

---

## 列印程序

您可以列印程序文字。

## 預覽程序

1. 開啓一個程序。出現程序指令。
2. 從**檔案**功能表中，按一下**頁面設定**。開啓「頁面設定」對話框。
3. 變更頁面佈置後，按一下**確定**。
4. 按一下工具列上的**預覽列印**按鈕：



出現列印的程序預覽。

## 列印程序

1. 開啓一個程序。出現程序指令。
2. 從**檔案**功能表中，按一下**頁面設定**。開啓「頁面設定」對話框。
3. 變更頁面佈置後，按一下**確定**。
4. 按一下工具列上的**列印**按鈕：



列印程序。

## 第7章 使用清單

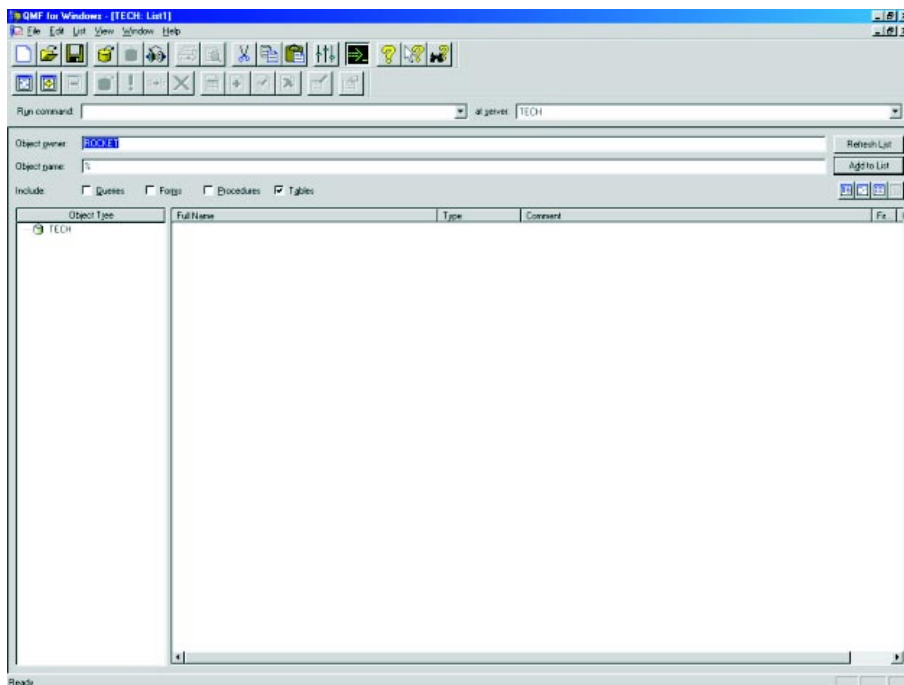
清單提供您簡單的方法來檢視 QMF 物件的集成。

### 物件

QMF for Windows 可辨識四種物件類型：查詢、套表、程序及表格。您可以使用「清單」視窗，根據物件名稱、擁有者及類型來檢視物件。

#### 清單物件

1. 從檔案功能表中，選取新清單。  
開啓「清單」視窗。



2. 指定擁有者和名稱。

**註：**您可以使用型樣，從符合的物件清單中選取物件。

- 使用百分比字元 (%)，使它符合內含任何字元的任何長度之字串。例如，要列示名稱開頭為字母 A 的所有表格，請輸入 A%。

- 使用底線字元 ( )，使它符合單一字元。例如，要列示擁有者名稱第二部份含字母 A 的所有表格，請輸入 \_A%。
3. 選取要搜尋的物件類型。
  4. 按一下**重新整理清單**。顯示資料庫伺服器上所儲存且符合的物件清單。

---

## 清單視窗指令

使用滑鼠右鍵按一下「清單」視窗中的物件，會顯示與「清單」視窗中指令相同的指令清單。

### 顯示物件

開啓選取的物件來檢視。適用查詢、套表、程序及表格。

### 執行物件

執行選取的物件。適用查詢與程序。

### 產生物件

依據選取的表格來建立查詢。您可以選擇產生 SQL SELECT 查詢、SQL UPDATE 查詢、SQL INSERT 查詢或提示查詢。適用表格。

### 編輯物件

開啓選取的物件來編輯。適用表格。

### 內容

顯示所選取物件的內容，包括註解、屬性及歷程用法資訊。適用查詢、套表、程序及表格。

---

## 建立清單

您可以建立清單來作為物件集成之用。例如，您可以建立所有庫存相關的查詢、套表、程序及表格的清單，在同一個位置保存您的工作。建立此清單後，您可以從清單新增及移除物件，以及儲存清單供未來使用。

### 新增物件至清單

您可以新增物件至清單。

從開啓的清單中，指定所要新增物件的擁有者及名稱資訊，然後按一下工具列上的**新增至清單**按鈕。



符合擁有者及名稱的物件即新增至清單。

## 從清單中移除物件

您可以從清單中移除不相關的物件。

從開啓的清單中，按一下工具列上的**移除**按鈕。



物件會從清單中移除，但不會被刪除。

## 儲存清單至檔案

1. 從開啓的清單中，按一下工具列上的**儲存**按鈕。



如果清單已儲存過，則會直接儲存。如果先前未儲存過，則會開啓「另存新檔」對話框。

2. 輸入用來儲存清單的檔名，然後按一下**確定**。儲存清單。

## 開啓儲存的清單檔案

1. 按一下工具列上的**開啓舊檔**按鈕。



開啓「開啓舊檔」對話框。

2. 選取要開啓的檔案並按一下**確定**。選取的清單會在清單文件中開啓。



---

## 第8章 使用工作檔案

您可以使用工作檔案來排程及執程序。工作檔案使用 Windows 排程器，依據預設的時間和日期來執程序。

---

### 工作檔案

您可以建立工作檔案，然後將它儲存在本端或資料庫伺服器上。

#### 建立工作檔案

1. 從**檔案**功能表中，選取**新工作**。  
開啓新工作文件。

#### 執行工作檔案

您可以執行儲存於本端的工作檔案。

1. 開啓工作檔案。
2. 按一下工具列上的**執行工作**按鈕。



3. 左右或上下拖曳分隔子欄線來調整直欄或列大小。

**註：**若在重新調整查詢的列或直欄大小後儲存查詢，就一定會使用新格式顯示。

#### 自動符合直欄和列

您可以自動調整直欄和列的大小，來符合它們所包含的資料。

使用滑鼠來選取整個直欄或列，然後按兩下直欄或列與相鄰物件間的黑色分隔線。直欄或列會自動調整大小以符合其資料。

**註：**若在重新調整查詢的列或直欄大小後儲存查詢，就一定會使用新格式顯示。

#### 排序查詢結果

執行查詢後，您可以按直欄字母順序來排序結果。

從查詢的「結果」檢視畫面中，選取一個直欄，然後從**結果**功能表中選取**依升冪排序**。

查詢結果即以升冪排序。

- 或 -

從查詢的「結果」檢視畫面中，選取一個直欄，然後從**結果**功能表中選取 **降冪排序**。

查詢結果即以降冪排序。

**註:** 若要對選取的直欄套用更複雜的排序，請從**結果**功能表中選取**排序**。

## 重排直欄

您可以變更查詢結果中直欄的次序。

從查詢的「結果」檢視畫面中，選取一個直欄並將它拖曳到新位置。

該直欄以新次序出現。

---

## 格式化查詢結果

您可以變更有來顯示查詢及查詢結果的字型。字型選項根據您在電腦上安裝的字型而有不同。關於新增字型的詳細資訊，請參閱作業系統的說明機能。

**註:** 如果您在選取新查詢結果顯示字型後儲存查詢，就一定會以新字型顯示。

## 選取查詢結果顯示字型

1. 從「結果」檢視畫面中，選取**結果**功能表中的**設定字型**。即開啓「字型」對話框。
2. 選取用來顯示查詢結果的字型及大小，然後按一下**確定**。查詢結果即以您指定的格式顯示。

**註:** 按一下**設為預設值**，使用選取的字型作為所有查詢結果的預設字型。

## 格式化數值查詢結果

1. 從「結果」檢視畫面中，選取一個內含數值的直欄，然後從**結果**功能表中選取**格式**。開啓「格式」對話框。
2. 指定您要套用的格式化並按一下**確定**。



**註:** 按一下**設為預設值**，使用選取的字型作為所有查詢結果的預設字型。

## 將查詢結果格式轉換成套表

您可以將查詢結果格式轉換成套表。

1. 從**結果功能表**中，選取**顯示報告**。  
開啓「選取套表」對話框。
2. 選取「套表」查詢並按一下**確定**。  
查詢結果格式即轉換成套表並在新套表視窗中開啓。

---

## 組合及聚集查詢結果

您可以將組合、聚集及彙總格式套用到查詢結果。

### 組合查詢結果

您可以組合包含或不包含彙總資訊的查詢結果。

1. 選取要組合的直欄。
2. 從**結果功能表**中，選取要套用的組合類型。  
直欄即根據您的選項組合。

### 彙總查詢結果

您可依直欄來彙總查詢結果。

1. 選取要組合的直欄。
2. 從**結果功能表**中，選取要套用的彙總類型。  
直欄即根據您的選項彙總。

---

## 儲存查詢結果與格式

您可以將查詢結果和格式儲存為套表。

### 將查詢結果儲存為表格

您可以在資料庫伺服器上將查詢結果儲存為表格。

1. 從**結果功能表**中，選取**儲存至資料庫**。  
開啓「儲存資料」對話框。
2. 輸入擁有者及表格名稱，然後按一下**確定**。  
查詢結果即儲存為資料庫上的表格。

## 儲存查詢結果至檔案

您可以將查詢結果儲存為 PC 或檔案伺服器上的檔案。

1. 從**結果**功能表中，選取**儲存至檔案**。  
開啓「匯出資料」對話框。
2. 指定用來儲存檔案的位置，匯出選項，然後按一下**確定**。  
查詢結果即儲存至檔案。

---

## 列印查詢結果

您可以預覽及列印查詢結果。

### 預覽查詢結果

1. 開啓並執行查詢。顯示查詢結果。
2. 從**檔案**功能表中，選取**頁面設定**。開啓「頁面設定」對話框。
3. 變更頁面佈置後，按一下**確定**。
4. 按一下工具列上的**預覽列印**按鈕：



出現列印的查詢結果預覽。

### 列印查詢結果

1. 開啓查詢並啓動「結果」檢視畫面。顯示查詢結果。
2. 從**檔案**功能表中，選取**頁面設定**。即出現「頁面設定」對話框。
3. 變更頁面佈置後，按一下**確定**。
4. 按一下工具列上的**列印**按鈕。



列印查詢結果。

---

## 第9章 使用靜態查詢

靜態查詢指先前傳遞至資料庫伺服器並連結到套件的一種 SQL 查詢。執行靜態查詢時，資料庫伺服器使用套件中連結的 SQL 文字，而不是目前出現在查詢視窗中的 SQL 文字。靜態查詢的資源效率比動態查詢高，但靜態查詢無法編輯。

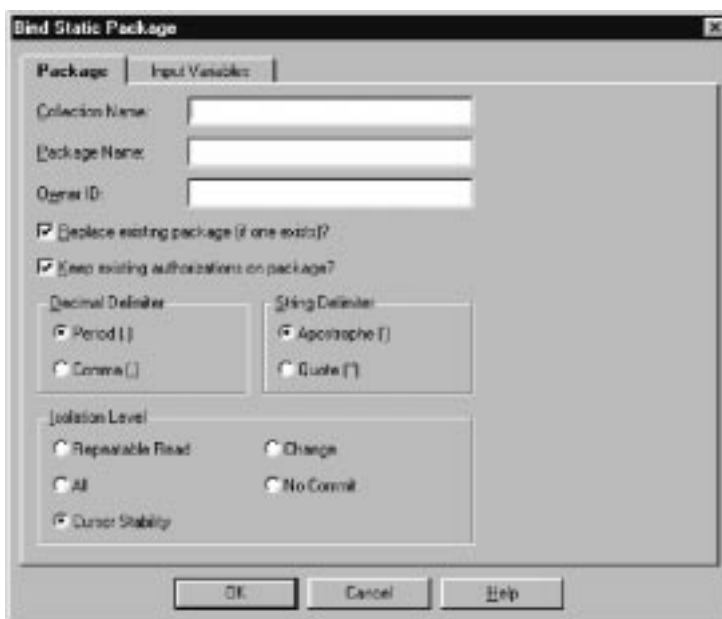
---

### 靜態查詢

靜態查詢是由先前存在的 SQL 查詢及提示查詢所建立。

#### 建立靜態查詢

1. 從查詢功能表中，選取**連結靜態套件**。開啓「連結靜態套件」對話框。



2. 選取「套件」標籤，輸入集成 ID 和套件名稱，然後變更其他想要的選項。
3. 如果查詢包含任何替代變數，請選取「變數」標籤。將任何替代變數取代為主電腦變數。
4. 按一下**確定**。連結靜態查詢。

**註：**連結查詢後，您也要儲存該查詢至檔案或資料庫伺服器。

## 將替代變數取代為主電腦變數

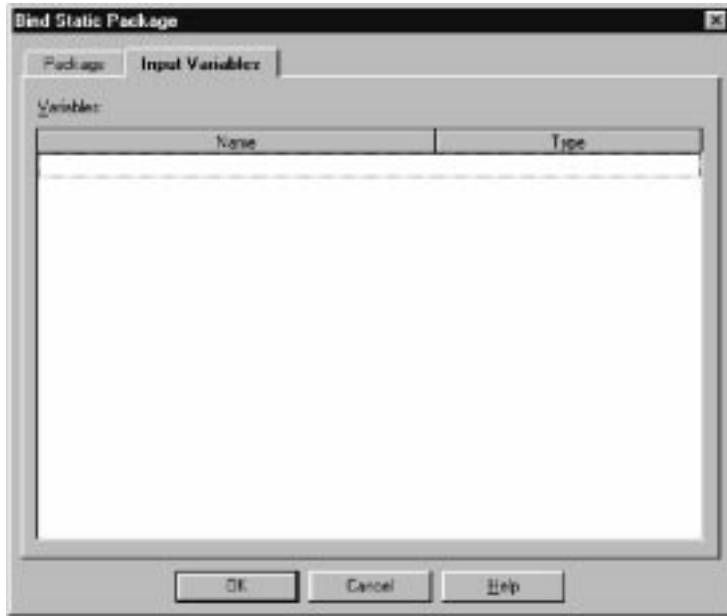
連結套件時，您必須指定以主電腦變數取代 SQL 文字中每一個替代變數。不過，替代變數不能都直接以主電腦變數來取代。在文字傳送到資料庫伺服器前，替代變數在查詢文字中提供直接的文字替代。主電腦變數被當作查詢的一部份傳送到資料庫伺服器。關於主電腦變數使用於查詢的方式及位置之規則，請參閱資料庫伺服器的文件。

指定替代變數與主電腦變數之間的關聯之後，QMF for Windows 會記住此關聯，並在未來的查詢中加以提醒，使連結套件變得更簡單。

主電腦變數的有效資料類型為：

- CHAR(n)
- VARCHAR(n)
- INTEGER
- SMALLINT
- FLOAT
- DECIMAL(p,s)
- DATE
- TIME
- TIMESTAMP

1. 從「連結靜態套件」對話框中，選取「輸入變數」標籤。



2. 輸入每一個主電腦變數的變數類型，然後按一下**確定**。替代變數即轉換成主電腦變數。

## 執行靜態查詢

執行靜態查詢和執行其他查詢的方法相同。請參閱頁面第7頁的『SQL 查詢』上的「SQL 查詢」。



---

## 第10章 使用表格編輯器

您可使用表格編輯器來搜尋、新增、編輯或刪除儲存於表格中的資料，而不必撰寫 SQL 陳述式。

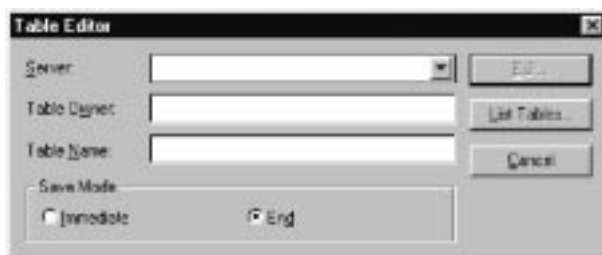
---

### 表格編輯器

表格編輯器讓您在編輯及搜尋資料時有很大的彈性。

#### 使用表格編輯器來搜尋列

1. 從檔案功能表中，選取**表格編輯器**。開啓「表格編輯器」對話框。



2. 指定一個表格。

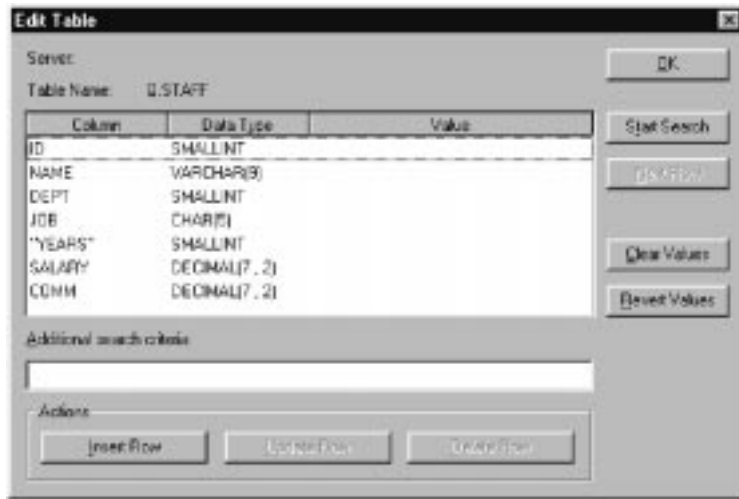
**註:** 您可以使用型樣，從符合的表格清單中選取表格名稱。

- 使用百分比字元 (%)，使它符合內含任何字元的任何長度之字串。例如，要列示名稱開頭為字母 A 的所有表格，請輸入 A%。
- 使用底線字元 (\_)，使它符合單一字元。例如，要列示擁有者名稱第二部份含字母 A 的所有表格，請輸入 \_A%。

輸入型樣後，請按一下**表格清單**，並從結果清單中選取一個表格。

3. 選取一種「儲存模式」。
  - 立即 - 表格每經變更後會立即在資料庫伺服器上更新。

- 結束 - 完成表格的所有變更後才會在資料庫伺服器上更新。當您對表格進行變更時，其他使用者無法對該表格做任何變更。



4. 按一下**編輯**。開啓「編輯表格」對話框。
5. 在「值」直欄中，輸入您要搜尋的值，或在「其它搜尋準則」欄位中鍵入搜尋準則，以指定更複雜的搜尋條件。您可以在「其它搜尋準則」欄位中輸入任何有效的 SQL 述語。
6. 按一下**開始搜尋**。「值」直欄中會顯示第一個符合的列。

## 新增列

1. 在「編輯表格」對話框中，輸入新記錄的資訊。
2. 按一下**插入列**。新列加入表格中。
3. 按一下**確定**。儲存您的變更。

## 變更列

1. 從「編輯表格」對話框中，搜尋要變更的列。
2. 按一下**下一列**，直到出現您要變更的列爲止。
3. 編輯「值」直欄中的資料，然後按一下**更新列**。更新該列。
4. 按一下**確定**。儲存您的變更。

## 刪除列

1. 從「編輯表格」對話框中，搜尋要刪除的列。
2. 按一下**下一列**，直到出現您要刪除的列爲止。



3. 按一下**刪除列**。列即被刪除。
4. 按一下**確定**。儲存您的變更。

---

## 從查詢結果檢視畫面中編輯表格

您可以直接從查詢結果檢視畫面中編輯表格。

### 從查詢結果檢視畫面中刪除列

您可以從查詢結果檢視畫面的表格刪除個別列。

從查詢結果檢視畫面中，選取某列，然後從**編輯**功能表中選取**刪除**。列即被刪除。

### 從查詢結果檢視畫面中更新直欄

您可以在查詢結果檢視畫面中更新個別直欄的內容。

從查詢結果檢視畫面中，按兩下某資料格，輸入新值，然後按 **Enter** 鍵。即更新表格。

---

## DB2 套表

如果您的機器上有安裝「DB2 套表使用者」元件，可以用它來作為不含 LOB 資料表格的表格編輯器。關於 DB2 套表的詳細資訊，請至 **Resource Center for DB2 Forms** 的網站，網址為 [www.rocketsoftware.com/db2forms](http://www.rocketsoftware.com/db2forms)。



## 第11章 分送資料

您可以將資料匯出到其它資料庫及應用程式。

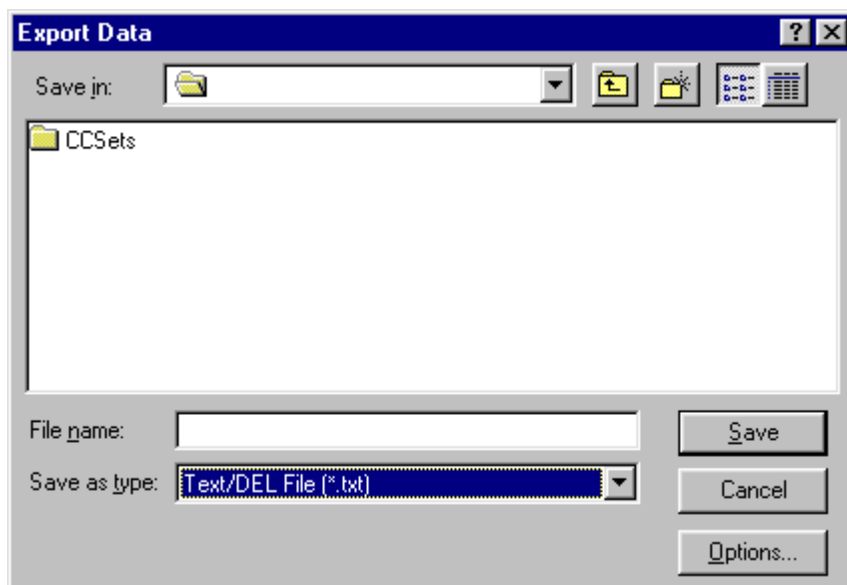
### 匯出資料

您可以利用下列方式將資料從 QMF for Windows 匯出至其它應用程式：

- 將資料匯出成文字、CSV、IXF 或 HTML 檔
- 將查詢結果儲存至表格
- 將查詢結果直接新增至 Microsoft Excel 試算表

### 匯出資料至檔案

1. 檢視查詢結果時，選取**檔案**功能表的**匯出資料**。開啓「匯出資料」對話框。



2. 選取想要的「輸出檔案類型」，然後按一下**選項**按鈕。根據您所選取的輸出檔案類型，會開啓「匯出文字/DEL 選項」對話框、「匯出 HTML 選項」對話框、「匯出 IXF 選項」對話框或「匯出 CSV 選項」對話框。
  - 您可以產生副檔名為 .TXT 的文字檔。此為具有選用性字串及直欄區隔字元的標準 ASCII 檔（指定於「匯出文字/DEL 選項」對話框中）。

- 您可以產生副檔名為 .HTM 的 HTML 檔。此為可供任何 Web 瀏覽器檢視的 HTML 檔案。檔案中會自動產生所有 HTML 標記；它可以發行到 Internet 或企業內部網站。您在「匯出 HTML 選項」對話框所選擇的選項，可控制所匯出資料的外觀。
  - 您可以產生 .IXF 檔。IXF 匯出方式可保留所有資料庫資訊，包括直欄標題及資料類型。它通常用於資料庫之間傳送資訊。
  - 您可以產生 .CSV 檔。CSV 匯出方式與文字匯出非常類似，都是使用逗點作為直欄區隔字元。此格式最常用於試算表應用程式。
3. 針對選定的匯出檔案類型選取一些選項，然後按一下**確定**。關閉「選項」對話框。
  4. 按一下「匯出資料」對話框的**確定**。匯出資料。

## 匯入資料

您可以匯入以 IXF 檔案儲存的資料。匯入資料到查詢視窗後，可以將資料儲存至資料庫伺服器、匯出成新檔案或用於報告。支援 PC/IXF 及字元模式 System/370 IXF 檔案。

1. 從**檔案**功能表中，選取**匯入資料**。開啓「匯入資料」對話框。



2. 選取要匯入的檔案並按一下**確定**。匯入的資料會顯示在新查詢視窗中。

## 儲存資料至資料庫伺服器

您可以將匯入的查詢結果儲存至資料庫表格。

1. 檢視匯入的查詢結果時，選取**檔案**功能表的**儲存資料**。開啓「儲存資料」對話框。



2. 選取資料庫伺服器、輸入表格擁有者及名稱、選擇其它任何想要的選項，然後按一下**確定**。儲存資料。

---

## 使用「傳送至」指令

QMF for Windows 併入了 Send To 指令及基本郵件從屬站。您可以將 Send To 指令與工作檔案一起使用，為查詢排程及分送查詢結果。

1. 從**檔案**功能表中，選取**傳送至**及 **Internet 郵件收件者**。開啓「訊息」對話框。
2. 指定訊息收件者、主題、訊息文字，然後按一下**下一步**。開啓「附件」對話框。
3. 新增或移除訊息的任何附件，然後按一下**下一步**。開啓「傳送訊息」對話框。
4. 指定郵件伺服器名稱並按一下**完成**。送出訊息。

---

## 使用 Microsoft Excel 載入程式

QMF for Windows 併入了 Microsoft Excel 7.0 或更新版本的載入程式。這些載入程式可讓您從 Excel 執行 QMF for Windows，然後直接將查詢結果傳回到試算表中。如果您選擇「典型」或「自訂」安裝選項，並選取「Microsoft Excel 載入程式」選項，則會自動安裝適當的載入程式。

1. 按一下 Excel 工具列的 **QMF for Windows** 按鈕。



開啓 QMF for Windows。

2. 從 QMF for Windows 中，選取查詢並執行。顯示查詢結果。
3. 選取要傳回 Excel 的資料。
4. 從**檔案**功能表中，選取**傳回資料至 Microsoft Excel**。開啓 Excel 並顯示「QMF for Windows 載入程式」對話框。
5. 輸入資料的目的地範圍並按一下**確定**。新增資料至試算表。

---

## 使用範例應用程式

QMF for Windows 提供若干的範例應用程式及整合解決方案。若需要更多資訊，請到 IBM 網站 <http://www.ibm.com/qmf/>。

---

## 第12章 使用 QMF Report Center

QMF Report Center 可讓您利用共用的 QMF 查詢、套表、程序及表格來產生自訂報告。藉由快速存取這些物件，您可以指定資料格式喜好設定，及產生可在各種應用程式中檢視和操作的自訂報告。

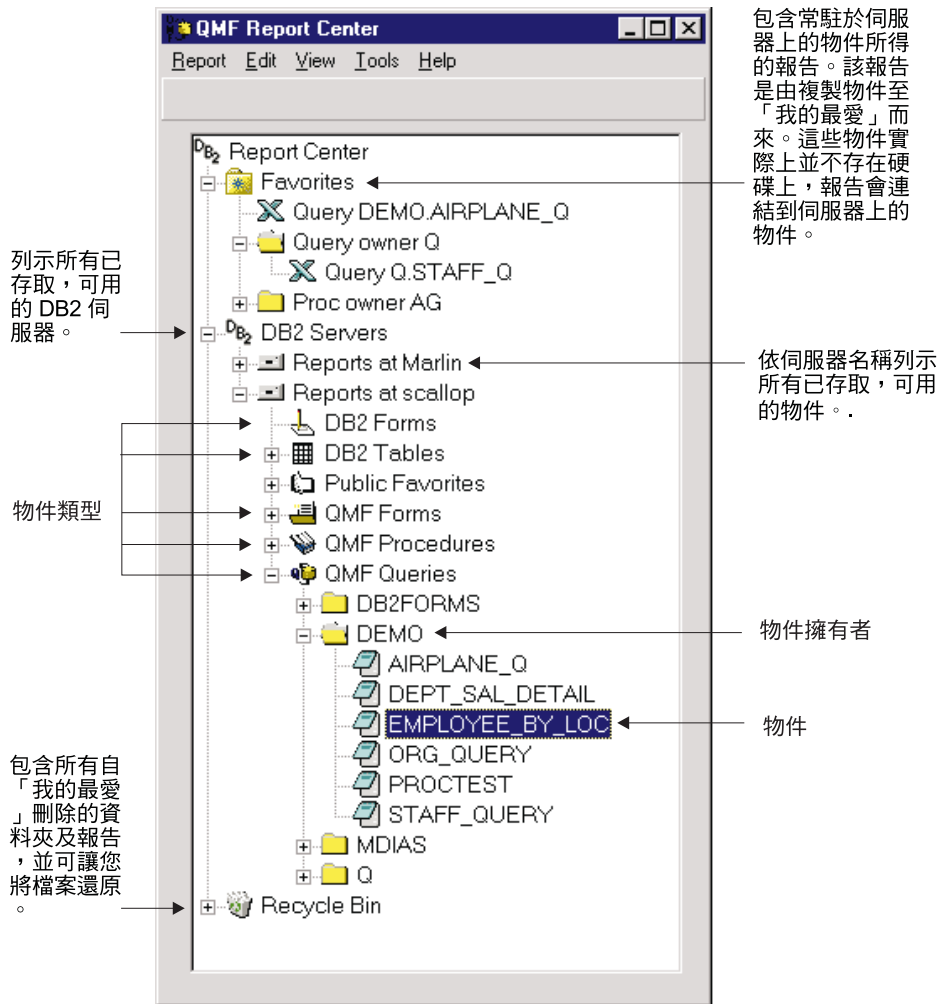
---

### QMF Report Center 入門

- 以滑鼠右鍵按一下任何物件或資料夾，啟動可從工具列功能表取得的相同選項。
- 按一下任何資料夾旁邊的加號 (+) 來開啓第一層內容。按一下加號 (+) 同時按住 SHIFT 鍵，可開啓資料夾下的所有層。

## QMF Report Center 視窗

QMF Report Center 視窗以樹狀結構顯示可用的「我的最愛」、DB2 伺服器、Public Favorites、物件及「資源回收筒」。



此視窗中顯示的物件都有一個圖示, 代表與物件輸出相關的應用程式類型。

## 連接伺服器

1. 如果「DB2 伺服器」下沒有顯示任何伺服器名稱, 請按一下加號 (+)。



- 按一下伺服器旁邊的加號 (+)。開啓「過濾選項」對話框。



- 選取要查看的物件類型，然後按一下**確定**。顯示伺服器上按物件類型組合的可用物件。

---

## 使用報告和物件

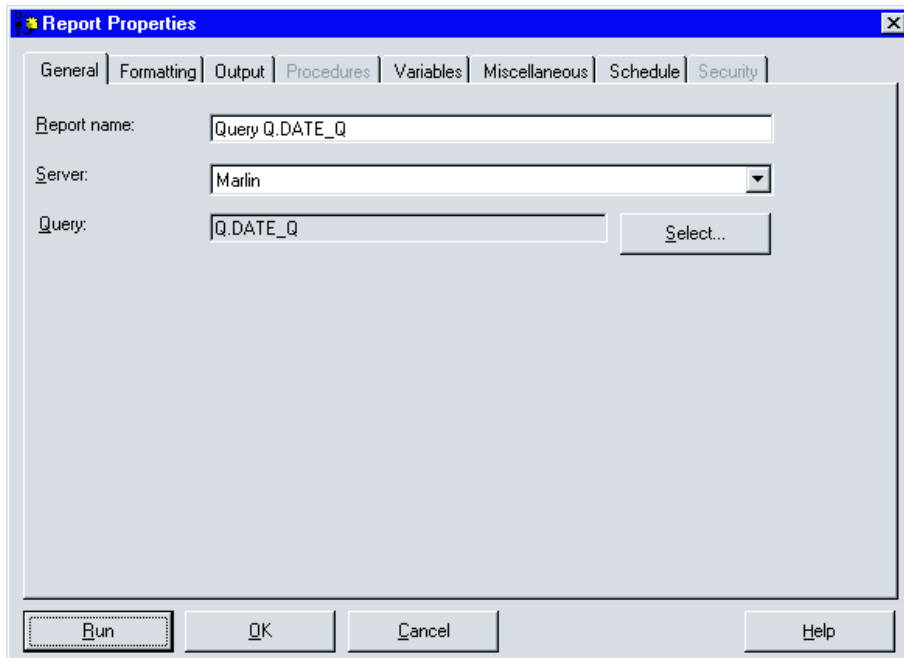
報告視 QMF 物件而定。您個人及 Public Favorites 資料夾內的全部項目被視為報告；您可以操作格式及顯示這些項目的選項。這些「我的最愛」資料夾內含的項目連結到常駐於伺服器上的 QMF 物件。您不是真的修改 QMF 物件，只是修改當作報告來參照的物件之連結。因為報告是根據物件而定，所以物件內容也會套用到報告。

您可以從常駐於伺服器的物件來建立報告；不過，它們不會被儲存到伺服器。此功能可讓您一次快速建立許多報告。不過，從伺服器上的物件建立報告後，您可以選擇儲存報告至「我的最愛」資料夾。

## 執行報告

您可以從「我的最愛」資料夾或位於伺服器上的物件來執行報告。

1. 針對選取的報告或物件，選取「報告」功能表中的內容。開啓「報告內容」對話框。



2. 必要時定義內容。
3. 按一下**執行**按鈕。開始處理報告，而當您在「報告內容輸出」對話框中選取發行後檢視報告選項後，報告會在您指定的應用程式中顯示。

您亦可透過下列任何方法快速執行報告：

- 選取報告，然後選擇「報告」功能表的執行。
- 以滑鼠右鍵按一下報告，然後選取執行。
- 按兩下報告名稱。

---

## 使用資料夾與我的最愛

資料夾用來組合報告與 QMF 物件；資料夾根據物件擁有者名稱來命名。您可以在資料夾執行您對報告所執行的相同作業，例如執行報告及定義報告內容。在某資料夾上執行這些作業時，作業會套用到該資料夾內含的每一個報告。例如，如果您要連續執行某資料夾內含的每一個報告，請選取該資料夾，然後選取「報告」功能表的**執行**。

QMF Report Center 包含供您儲存報告的兩個頂層資料夾。這些資料夾包含指向伺服器物件的報告；這些物件本身不包含在「我的最愛」資料夾內。您的個人「我的最愛」資料夾位於本端（指您的 PC）上，所以您是唯一可存取該資料夾及其內容的使用者。Public Favorites 資料夾位於伺服器上，所有被授權的使用者皆可存取它。根據您的資源限制，您可以存取若干 Public Favorites 資料夾，但每一個伺服器上絕不會多於一個 Public Favorites 資料夾。

複製 QMF 物件至「我的最愛」資料夾時，資料夾會被自動命名來納入物件類型及擁有者名稱。從伺服器（如全部查詢）複製整個物件類型（如相同類型物件的資料夾）時，伺服器名稱也會併入新資料夾名稱中。

## 新增報告至我的最愛

您可以將任何物件或報告從伺服器新增至您個人的「我的最愛」資料夾，或新增至伺服器（您系統管理者授權的伺服器）的 Public Favorites 資料夾。

### 新增報告至個人「我的最愛」：

針對選取的報告或物件，選取「報告」功能表中的新增至我的最愛，或將報告或物件拖曳到您的個人「我的最愛」資料夾。報告以下列命名慣例新增至您個人的「我的最愛」資料夾頂端：ObjecttypeOWNERNAME.OBJECTNAME。

### 新增報告至 Public Favorites：

拖曳 QMF 物件或報告至伺服器上的 Public Favorites 資料夾。您可以從個人「我的最愛」資料夾或任何伺服器來新增報告。

**註：**當新增報告至 Public Favorites 或在 Public Favorites 中修改報告時，您必須在儲存更新至伺服器前，從「報告」功能表中選取儲存變更至 *Public Favorites*。

關於使用 QMF Report Center 的詳細資訊，請參閱線上說明系統。



---

## 第13章 使用 QMF for Windows API

您可以使用 QMF for Windows 應用程式設計介面來建立自訂的應用程式。

---

### 透過 API 控制 QMF for Windows

下列步驟摘要說明您如何使用 API 來控制 QMF for Windows。

1. 建立 QMF for Windows API 物件的案例。如果您使用 Microsoft Visual Basic，請在 QMF for Windows 類型程式庫 (qmfwin.tlb) 新增參考手冊。然後使用 Dim 陳述式：

```
Dim QMFWin As New QMFWin
```

或 CreateObject 陳述式：

```
Dim QMFWin As Object
```

```
Set QMFWin = CreateObject ("QMFWin.Interface")
```

**註：** 如果您使用不同的開發環境，請參閱產品文件來完成此步驟。

2. 選取您要使用的 DB2 伺服器，呼叫 InitializeServer() 來起始設定資料庫的連線。

**註：** 在使用者 ID 和密碼通過 DB2 驗證之前，您無法起始設定伺服器。您可以令 QMF for Windows 提示輸入使用者 ID 和密碼，或在應用程式中提示它們，再傳遞到 InitializeServer() 函數呼叫中當做參數。

3. 使用 InitializeQuery() 來選取您要執行的查詢。如果查詢包含變數，請使用 SetVariable() 函數來設定變數值。
4. 開啓或執行查詢。使用 Open() 函數來開啓 SELECT 陳述式的查詢游標，使用 Execute() 函數來執行非 SELECT 陳述式的 SQL。
5. 如果查詢是 SELECT 陳述式，請重複呼叫 FetchNextRow() 來提取資料列。如果要一次提取一列以上，請使用 FetchNextRows()，或使用 CompleteQuery() 來指示 QMF for Windows 提取所有列。
6. 如果查詢是 SELECT 陳述式，請使用 Close() 函數來關閉查詢。
7. 請使用 Commit() 或 Rollback() 函數來終止工作單元。

### 鎖住呼叫

所有的 QMF for Windows API 函數都是同步作業。這表示函數在所要求的資料庫動作完成之前會鎖住或不傳回。這種實作方式很理想，因為可以簡化從屬站應用

程式的設計。不過，如果您的從屬站應用程式是單執行緒，則它在等待 QMF for Windows API 函數傳回之前，無法回應使用者輸入或重新整理畫面。

QMF for Windows API 一次回應一個來自從屬站的函數呼叫。如果您的從屬站應用程式是多執行緒，您必須：

- 先等待一個函數呼叫完成之後，再發出另一個呼叫，或
- 建立多個 QMF for Windows API 的案例（每一個使用 API 的執行緒各一個案例）。

## 連接資料庫

QMF for Windows API 物件的每一個案例都建立和使用單一連線來連接資料庫，來處理視後續回復或確定而定的全部資料庫活動，包括開啓查詢、提取資料、以及執行 SQL 陳述式。

如果您呼叫 `InitializeQuery()` 兩次以上，在 QMF for Windows API 物件的一個特定案例中建立一個以上的查詢，則所有查詢皆共用一個相同的連線。

QMF for Windows API 一次回應一個來自從屬站的函數呼叫。如果您的從屬站應用程式是多執行緒，您必須：

- `DeleteQMFObject()`
- `GetQMFOBJECTInfo()`
- `GetQMFOBJECTInfoEx()`
- `GetQMFOBJECTList()`
- `GetQMFOBJECTListEx()`
- `GetQMFQueryText()`
- `SaveQMFQuery()`

QMF for Windows 建立和使用第二個連線來連接資料庫，以便處理管理性資料庫活動（例如，擷取 QMF 資訊）。爲了支援從屬站應用程式的回復和確定機制，第二個連線是必要的。

QMF for Windows API 物件會自動處理這些資料庫連線。不過，如果您的系統管理者已限制連線數，請記得 QMF for Windows API 物件的每一個案例可使用兩個連線。

---

## API 參考手冊

此參考手冊列示在使用 QMF for Windows API 建立應用程式時可用的全部指令。

## AddDecimalHostVariable()

short AddDecimalHostVariable(long *QueryID*, short *Type*, short *Precision*, short *Scale*, const VARIANT& *Value*)

### 說明

此函數將 *Value* 的資料套用到 *QueryID* 所起始設定之靜態 SQL 陳述式中的變數。您可以呼叫此函數來起始設定陳述式中的每一個變數。QMF for Windows 不會嘗試比對值和變數，所以您必須按照變數在 SQL 陳述式中的順序來呼叫此函數。

### 參數

名稱	說明
<i>QueryID</i>	從 InitializeStaticQuery() 傳回的查詢 ID。
<i>Type</i>	要傳遞到資料庫伺服器的值之 SQL 資料類型。此值影響 <i>Value</i> 從 VARIANT 資料類型如何轉換成實際傳遞的值。對於 AddDecimalHostVariable() 而言，唯一有效值是 484 (RSDT_DECIMAL)。
<i>Precision</i>	十進位值的精確度。
<i>Scale</i>	十進位值的小數位數。
<i>Value</i>	在陳述式中用來取代變數的資料值。如果要指定空值，variant 的類型應該設為 VT_EMPTY。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值是「非零」，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

## AddHostVariable()

short AddHostVariable(long *QueryID*, short *Type*, const VARIANT& *Value*)

### 說明

此函數將 *Value* 的資料套用到 *QueryID* 所起始設定之靜態 SQL 陳述式中的變數。您必須呼叫此函數來起始設定陳述式中的每一個變數。QMF for Windows 不會嘗試比對值和變數，所以您必須按照變數在 SQL 陳述式中的順序來呼叫此函數。

### 參數

名稱	說明
<i>QueryID</i>	從 InitializeStaticQuery() 傳回的查詢 ID。

<i>Type</i>	要傳遞到資料庫伺服器的值之 SQL 資料類型。此值影響 <i>Value</i> 從 VARIANT 資料類型如何轉換成實際傳遞的值。
<i>Value</i>	在陳述式中用來取代變數的資料值。如果要指定空值，variant 的類型應該設為 VT_EMPTY。

*Type* 的有效值包括：

值	意義
384 (RSDT_DATE)	日期
388 (RSDT_TIME)	時間
392 (RSDT_TIMESTAMP)	時間戳記
448 (RSDT_VARCHAR)	可變長度的字串
452 (RSDT_CHAR)	字串
464 (RSDT_VARGRAPHIC)	可變長度的圖形
468 (RSDT_GRAPHIC)	圖形
480 (RSDT_FLOAT)	浮點數
496 (RSDT_INTEGER)	4 位元組整數
500 (RSDT_SMALLINT)	2 位元組整數

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

## BindDecimalHostVariable()

short BindDecimalHostVariable(BSTR *CollectionName*, BSTR *PackageName*, short *SectionNumber*, short *Number*, BSTR *Name*, short *DataType*, short *Precision*, short *Scale*)

### 說明

此函數在指定的區段中連結一個變數。在 SQL 文字中併入 ":H" 當作主變數的位置保留元。對於 SQL 文字中的每一個十進位主變數，您必須呼叫 BindDecimalHostVariable() 來指定變數的資訊。

### 參數

名稱	說明
<i>CollectionName</i>	您要連結之套件的集成 ID。



<i>PackageName</i>	您要連結之套件的名稱。
<i>SectionNumber</i>	您要連結之集成和套件內的陳述式區段號碼。
<i>Number</i>	您要連結之變數的識別碼。 SQL 陳述式中的第一個變數為變數 0，依此類推。
<i>Name</i>	供資料庫伺服器用來診斷。 QMF for Windows 不會驗證此值，亦不需要此值。
<i>DataType</i>	變數的 SQL 資料類型。對於 BindDecimalHostVariable() 而言，唯一有效值是 484 (RSDT_DECIMAL)。
<i>Precision</i>	十進位值的精確度。
<i>Scale</i>	十進位值的小數位數。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。

## BindHostVariable()

short BindHostVariable(BSTR *CollectionName*, BSTR *PackageName*, short *SectionNumber*, short *Number*, BSTR *Name*, short *DataType*, short *Length*)

### 說明

此函數在指定的區段中連結一個變數。在 SQL 文字中併入 ":H" 當作主變數的位置保留元。對於 SQL 文字中的每一個主變數，您必須呼叫 BindHostVariable() 來指定變數的資訊。

### 參數

名稱	說明
<i>CollectionName</i>	您要連結之套件的集成 ID。
<i>PackageName</i>	您要連結之套件的名稱。
<i>SectionNumber</i>	您要連結之集成和套件內的陳述式區段號碼。
<i>Number</i>	您要連結之變數的識別碼。 SQL 陳述式中的第一個變數為變數 0，依此類推。
<i>Name</i>	供資料庫伺服器用來診斷。 QMF for Windows 不會驗證此值，亦不需要此值。
<i>DataType</i>	變數的 SQL 資料類型。
<i>Length</i>	變數的長度。

*DataType* 的有效值包括：

值	意義
384 (RSDT_DATE)	日期
388 (RSDT_TIME)	時間
392 (RSDT_TIMESTAMP)	時間戳記
448 (RSDT_VARCHAR)	可變長度的字串
452 (RSDT_CHAR)	字串
464 (RSDT_VARGRAPHIC)	可變長度的圖形
468 (RSDT_GRAPHIC)	圖形
480 (RSDT_FLOAT)	浮點數
484 (RSDT_DECIMAL)	十進位
496 (RSDT_INTEGER)	4 位元組整數
500 (RSDT_SMALLINT)	2 位元組整數

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## BindSection()

`short BindSection(BSTR CollectionName, BSTR PackageName, short SectionNumber, BSTR SQLText)`

### 說明

此函數設定在連結期間要於集成和套件的指定區段號碼中使用的 SQL 文字。

### 參數

名稱	說明
<i>CollectionName</i>	您要連結之套件的集成 ID。
<i>PackageName</i>	您要連結之套件的名稱。
<i>SectionNumber</i>	您要連結之集成和套件內的陳述式區段號碼。
<i>SQLText</i>	您要連結之陳述式的 SQL 文字。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## CancelBind()

short `CancelBind(BSTR CollectionName, BSTR PackageName)`

### 說明

此函數取消先前起始設定的連結作業。指定之套件的全部資訊都會釋放。

### 參數

名稱	說明
<code>CollectionName</code>	您要連結之套件的集成 ID。
<code>PackageName</code>	您要連結之套件的名稱。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## ChangePassword()

short `ChangePassword(BSTR NewPassword)`

### 說明

此函數變更先前在 `InitializeServer()` 呼叫上所指定之使用者 ID 的密碼。

**註：**並非全部的資料庫伺服器類型都支援密碼變更。如果 `InitializeServer()` 呼叫上指定的伺服器不支援密碼變更，則傳回錯誤訊息，密碼也不會變更。

### 參數

名稱	說明
<code>NewPassword</code>	新密碼。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## ClearList()

short ClearList(short *Type*)

### 說明

此函數重新起始設定 *Type* 參數所指定的內部清單。

### 參數

名稱	說明
<i>Type</i>	RSL_SERVER 或 RSL_QUERY。

### 回覆值

成功則為「零」，失敗則為 RS\_ERROR\_OUTOFRANGE。

### 相關的主題

Open()

## Close()

short Close(long *QueryID*)

### 說明

此函數可以關閉查詢，使 *QueryID* 無效。如果查詢有一個開啓的游標，則關閉游標，釋放資料庫供其他使用者使用。此函數並不會終止資料庫伺服器的連線。因為連線仍然開啓，所以不會執行回復或確定。

**註：**此函數的名稱與 Microsoft Access 2.0 的關鍵字 Close 發生衝突。如果您使用 MS Access 2.0，請在函數名稱兩旁加上方括弧 [ ]。

### 參數

名稱	說明
<i>QueryID</i>	從 InitializeQuery() 傳回的查詢 ID。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。

## 相關的主題

Execute()

Open()

## Commit()

short Close(long *QueryID*)

### 說明

此函數可以確定您在現行工作單元中的任何變更、結束現行工作單元、關閉開啓的游標、使全部未完成的 Query ID 無效。

**註:** 此函數的名稱與 Microsoft Access 2.0 的關鍵字 Commit 發生衝突。如果您使用 MS Access 2.0，請在函數名稱兩旁加上方括弧 [ ]。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。

## 相關的主題

Rollback()

## CompleteQuery()

short CompleteQuery(long *QueryID*)

### 說明

此函數提取結果集的所有列，並且儲存在 QMF for Windows 內部。如果查詢有一個開啓的游標，則關閉游標，釋放資料庫供其他使用者使用。您可以使用 FetchNextRow() 或 FetchNextRows() 來擷取列。完成此查詢時，請呼叫 Close()。

### 參數

名稱	說明
<i>QueryID</i>	從 InitializeQuery() 傳回的查詢 ID。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。

## CopyToClipboard()

short CopyToClipboard(long *QueryID*, long *FirstRow*, long *FirstCol*, long *LastRow*, long *LastCol*, BOOL *IncludeColHeadings*, [VARIANT *DateTimeFormat*])

### 說明

此函數將指定的欄列範圍複製到剪貼簿。如果您尚未擷取要複製到剪貼簿的全部列資料，請先呼叫 `CompleteQuery()` 再呼叫此函數。如果您嘗試複製尚未從資料庫擷取的列，則傳回錯誤訊息。

### 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
<i>FirstRow</i>	您要複製的第一列。
<i>FirstCol</i>	您要複製的第一直欄。
<i>LastRow</i>	您要複製的最後一列。如果要複製全部列，請指定 <code>-1</code> 。
<i>LastCol</i>	您要複製的最後一欄。如果要複製全部直欄，請指定 <code>-1</code> 。
<i>IncludeColHeadings</i>	指定「非零」可以在第一列中包含直欄標題，指定「零」則不包含。
<i>DateTimeFormat</i>	可選用日期和時間值的格式。有效值包括 0 (ISO 格式)、1 (USA 格式)、2 (EUR 格式)、3 (JIS 格式)、或 4 (Windows 控制台格式)。預設值是 4。

**註：**結果集第一列的值是 0，最後一列的值等於總列數減一。結果集第一直欄的值是 0，最後一欄的值等於總直欄數減一。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()` 或 `GetLastErrorType()` 來取得其他的錯誤資訊。如果結果集是空的，或未從資料庫擷取到任何列，則傳回「非零」，除非 `FirstRow=0` 且 `LastRow=1`。如果是這樣，則會傳回「零」，並且複製一個空字串到剪貼簿。

## DeleteQMFObjct()

short DeleteQMFObjct(BSTR *OwnerAndName*)

### 說明

此函數刪除一個 QMF 物件（查詢、套表、程序或表格）。

## 參數

名稱	說明
<i>OwnerAndName</i>	以句點分隔的字串，含有您要刪除之物件的擁有者和名稱。例如， John.Query2

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## EndBind()

short `EndBind(BSTR CollectionName, BSTR PackageName)`

## 說明

此函數完成靜態 SQL 套件的連結程序。呼叫此函數可以讓 QMF for Windows 將現行套件的完整資訊傳送到資料庫去處理。

## 參數

名稱	說明
<i>CollectionName</i>	先前呼叫 <code>StartBind()</code> 時所用的集成名稱。
<i>PackageName</i>	先前呼叫 <code>StartBind()</code> 時所用的套件名稱。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## Execute()

short `Execute(long QueryID)`

## 說明

此函數可以執行採用 SQL Verb 而非 SELECT 的 SQL 陳述式。如果陳述式沒有傳回結果，請使用 `Execute()`。如果陳述式傳回結果，請使用 `ExecuteEx()`。對於使用 SELECT 動詞的陳述式，請使用 `Open()` 來代替 `Execute()` 或 `ExecuteEx()`。如果要判斷查詢所用的動詞，請呼叫 `GetQueryVerb()`。

**註:** 此函數的名稱與 Microsoft Access 2.0 的關鍵字 `Execute` 發生衝突。如果您使用 MS Access 2.0，請在函數名稱兩旁加上方括弧 [ ]。

### 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

### 相關的主題

`Execute()`

`Open()`

## ExecuteEx()

`short ExecuteEx(long QueryID, VARIANT* Result)`

### 說明

此函數可以執行採用 SQL Verb 而非 SELECT 的 SQL 陳述式。如果陳述式傳回結果，例如 `SELECT INTO` 陳述式，請使用 `ExecuteEx()`。如果陳述式沒有傳回結果，請使用 `Execute()`。對於使用 `SELECT` 動詞的陳述式，請使用 `Open()` 來代替 `Execute()` 或 `ExecuteEx()`。如果要判斷查詢所用的動詞，請呼叫 `GetQueryVerb()`。

### 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。



<i>Result</i>	<p>指向儲存結果的 <code>VARIANT</code> 之指標。結果是一個陣列（<code>variant</code> 類型 <code>VT_ARRAY</code>   <code>VT_VARIANT</code>），其中結果的每一個直欄各有一個值。</p> <p>每一個值都以其原來的資料類型或最接近的 <code>variant</code> 資料類型來指定。支援的傳回類型包括：<code>string</code>（<code>variant</code> 類型 <code>VT_BSTR</code>）、<code>float</code>（<code>variant</code> 類型 <code>VT_R4</code>）、<code>double</code>（<code>variant</code> 類型 <code>VT_R8</code>）、<code>short</code>（<code>variant</code> 類型 <code>VT_I2</code>）、<code>long</code>（<code>variant</code> 類型 <code>VT_I4</code>）、以及 <code>binary</code>（<code>variant</code> 類型 <code>VT_UI1</code>   <code>VT_ARRAY</code>）。</p> <p>在呼叫此函數之前，您必須適當地起始設定 <code>VARIANT</code>。<code>Visual Basic</code> 會自動處理這件工作。<code>Visual C++</code> 程式設計師必須呼叫 <code>VariantInit()</code>。</p>
---------------	--

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## ExecuteStored Procedure()

```
short ExecuteStoredProcedure(long QueryID,[VARIANT vaCommitOK], [VARIANT vaMaxResultSets], [VARIANT vaColumnNames], [VARIANT vaColumnLabels], [VARIANT vaColumnComments])
```

### 說明

此函數可以執行一個採用 `SQL Verb CALL` 的 `SQL` 陳述式，在資料庫伺服器上執行一個儲存程序。如果儲存程序沒有傳回結果（除了結果集以外），請使用 `ExecuteStoredProcedure()`。如果儲存程序傳回結果，請使用 `ExecuteStoredProcedureEx()`。

如果要起始設定 `ExecuteStoredProcedure()` 所執行的儲存程序，請先呼叫 `InitializeQuery()`，指定一個採用 `CALL` 陳述式的 `SQL` 陳述式。在 `CALL` 陳述式中，儲存程序名稱必須指定為文字。`CALL` 陳述式中指定的任何參數（即使是常數）會被忽略。請改用 `AddHostVariable()` 來指定輸入和輸出變數。

### 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。查詢的 <code>SQL</code> 文字應該指定 <code>CALL</code> 陳述式。

<i>vaCommitOK</i>	可選用的 Boolean 值，指定儲存程序是否可以確定工作單元，或是否應該限制此作業。預設值是 True。
<i>vaMaxResultSets</i>	可選用的數值，指定容許儲存程序可以傳回的最大結果集數目。如果您不要讓儲存程序傳回結果集，或資料庫伺服器不支援透過 DRDA 從儲存程序傳回結果集，請指定「零」。
<i>vaColumnNames</i>	可選用的 Boolean 值，對於每一個傳回結果集中的直欄，指定資料庫是否要傳回直欄名稱。
<i>vaColumnLabels</i>	可選用的 Boolean 值，對於每一個傳回結果集中的直欄，指定資料庫是否要傳回直欄標籤。
<i>vaColumnComments</i>	可選用的 Boolean 值，對於每一個傳回結果集中的直欄，指定資料庫是否要傳回直欄註解。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## ExecuteStored ProcedureEx()

```
short ExecuteStoredProcedureEx(long QueryID, VARIANT* Result, [VARIANT vaCommitOK], [VARIANT vaMaxResultSets], [VARIANT vaColumnNames], [VARIANT vaColumnLabels], [VARIANT vaColumnComments])
```

### 說明

此函數可以執行一個採用 SQL Verb CALL 的 SQL 陳述式，在資料庫伺服器上執行一個儲存程序。如果儲存程序傳回結果（除了結果集以外），請使用 `ExecuteStoredProcedureEx()`。對於確實傳回結果的儲存程序，請使用 `ExecuteStoredProcedureEx()`。

如果要起始設定 `ExecuteStoredProcedure()` 所執行的儲存程序，請先呼叫 `InitializeQuery()`，指定一個採用 CALL 陳述式的 SQL 陳述式。在 CALL 陳述式中，儲存程序名稱必須指定為文字。CALL 陳述式中指定的任何參數（即使是常數）會被忽略。請改用 `AddHostVariable()` 來指定輸入和輸出變數。

如果儲存程序會傳回結果集，請呼叫 `GetStoredProcedureResultSets()` 來擷取結果集的查詢 ID。

### 參數

名稱	說明
----	----

<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。查詢的 SQL 文字應該指定 CALL 陳述式。
<i>Result</i>	<p>指向儲存結果的 VARIANT 之指標。結果是一個陣列（variant 類型 VT_ARRAY   VT_VARIANT），其中結果的每一個直欄各有一個值。</p> <p>每一個值都以其原來的資料類型或最接近的 variant 資料類型來指定。支援的傳回類型包括：<code>string</code>（variant 類型 VT_BSTR）、<code>float</code>（variant 類型 VT_R4）、<code>double</code>（variant 類型 VT_R8）、<code>short</code>（variant 類型 VT_I2）、<code>long</code>（variant 類型 VT_I4）、以及 <code>binary</code>（variant 類型 VT_UI1   VT_ARRAY）。</p> <p>在呼叫此函數之前，您必須適當地起始設定 VARIANT。Visual Basic 會自動處理這件工作。Visual C++ 程式設計師必須呼叫 <code>VariantInit()</code>。</p>
<i>vaCommitOK</i>	可選用的 Boolean 值，指定儲存程序是否可以確定工作單元，或是否應該限制此作業。預設值是 <code>true</code> 。
<i>vaMaxResultSets</i>	可選用的數值，指定容許儲存程序可以傳回的最大結果集數目。如果您不要讓儲存程序傳回結果集，或資料庫伺服器不支援透過 DRDA 從儲存程序傳回結果集，請指定「零」。
<i>vaColumnNames</i>	可選用的 Boolean 值，對於每一個傳回結果集中的直欄，指定資料庫是否要傳回直欄名稱。
<i>vaColumnLabels</i>	可選用的 Boolean 值，對於每一個傳回結果集中的直欄，指定資料庫是否要傳回直欄標籤。
<i>vaColumnComments</i>	可選用的 Boolean 值，對於每一個傳回結果集中的直欄，指定資料庫是否要傳回直欄註解。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## Export()

`short Export(long QueryID, long FirstRow, long FirstCol, long LastRow, long LastCol, short Format, short StringDelimiter, short ColumnDelimiter, BOOL IncludeColHeadings, BSTR FileName, [VARIANT DateTimeFormat])`

## 說明

此函數將指定的欄列範圍複製到剪貼簿。如果您尚未擷取要複製到剪貼簿的全部列資料，請先呼叫 `CompleteQuery()` 再呼叫此函數。如果您嘗試複製尚未從資料庫擷取的列，則傳回錯誤訊息。

## 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
<i>FirstRow</i>	您要匯出的第一列。
<i>FirstCol</i>	您要匯出的第一直欄。
<i>LastRow</i>	您要複製的最後一列。如果要複製全部列，請指定 <code>-1</code> 。
<i>LastCol</i>	您要複製的最後一欄。如果要複製全部直欄，請指定 <code>-1</code> 。
<i>IncludeColHeadings</i>	指定「非零」可以在第一列中包含直欄標題，指定「零」則不包含。
<i>Filename</i>	字串，含有您要寫入匯出內容的檔案名稱。
<i>DateTimeFormat</i>	可選用日期和時間值的格式。有效值包括 0 (ISO 格式)、1 (USA 格式)、2 (EUR 格式)、3 (JIS 格式)、或 4 (Windows 控制台格式)。預設值是 4。

**註：**結果集第一列的值是 0，最後一列的值等於總列數減一。結果集第一直欄的值是 0，最後一欄的值等於總直欄數減一。

名稱	說明
<i>Format</i>	指定輸出格式。

值	意義
0 (RSEF_TEXT)	以純文字格式寫入輸出檔案。
1 (RSEF_HTML)	以 HTML 格式寫入輸出檔案，並且以 HTML 表格來組織資料。
2 (RSEF_CSV)	以 CSV (逗號分隔值) 格式寫入輸出檔案。
3 (RSEF_PCIXF)	以 PC/IXF 格式寫入輸出檔案。
4 (RSEF_S370IXF)	以 System/370 IXF 格式寫入輸出檔案。

名稱	說明
字串區隔字元	指定字串區隔字元。如果 <i>Format</i> 是 <code>RSEF_HTML</code> ，則忽略此參數。

值	意義
0 (RSSD_NONE)	不使用字串區隔字元。
1 (RSSD_SINGLEQUOTE)	以單引號字元 (') 來區隔字串。
2 (RSSD_DOUBLEQUOTE)	以雙引號字元 (") 來區隔字串。

名稱	說明
<i>Column Delimiter</i>	指定直欄區隔字元。如果 <i>Format</i> 是 RSEF_HTML，則忽略此參數。

值	意義
0 (RSCD_SPACE)	以一個空白字元 ( ) 來區隔直欄。
1 (RSCD_TAB)	以一個欄標字元 (\t) 來區隔直欄。
2 (RSCD_COMMA)	以一個逗號字元 (,) 來區隔直欄。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()` 或 `GetLastErrorType()` 來取得其他的錯誤資訊。如果結果集是空的，或未從資料庫擷取到任何列，則傳回「非零」，除非 *FirstRow=0* 且 *LastRow=1*。如果是這樣會傳回「零」，並且寫入一個空白檔案。

### 相關的主題

`CopyToClipboard()`

## ExportForm()

short `ExportForm(BSTR OwnerAndName, BSTR FileName)`

### 說明

此函數將指定的 QMF 套表匯出到指定的檔案。

### 參數

名稱	說明
<i>OwnerAndName</i>	以句點分隔的字串，含有您要匯出之套表的擁有者和名稱。例如， John.Query2
<i>FileName</i>	字串，含有您要寫入匯出套表的檔案名稱。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## 相關的主題

`PrintReport()`

## ExportReport()

`short ExportReport(long QueryID, short SourceType, BSTR Source, BSTR OutputFileName, short PageLength, short PageWidth, BOOL IncludeDateTime, BOOL IncludePageNumbers, [VARIANT Format], [VARIANT UseFormPageSetup])`

## 說明

此函數可以為指定的查詢建立一個報告，並且寫入一個檔案。您可以為 QMF 套表中的報告指定格式和佈置。輸出檔是 ASCII 文字檔，每一行以一對回車和換行字元來區隔，每一頁以一個換頁字元來區隔。使用固定字距的字型來檢視輸出檔，效果最好。

## 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
<i>Source</i>	您要使用之套表的名稱（檔案名稱或 <code>Owner.Name</code> ）。
<i>OutputFileName</i>	您要寫入報告的檔案名稱。
<i>PageLength</i>	每一頁報告的行數。 <i>PageLength</i> 為 -1 可以指定連續輸出（除非報告的寬度大於 <i>PageWidth</i> ，否則不會中斷）。
<i>IncludeDateTime</i>	「非零」可以指定在每一頁底端顯示日期和時間。「零」可以指定不顯示日期和時間。
<i>IncludePageNumbers</i>	「非零」可以指定在每一頁底端顯示頁碼。「零」可以指定不顯示頁碼。
<i>Format</i>	可選用指定匯出報告的格式。如果指定「零」，則格式為純文字，表示輸出和套表完全一樣（文字或 HTML，視套表類型而定）。如果指定「非零」，則格式為 HTML，表示輸出為 HTML。對於非 HTML 的套表，輸出的開頭和尾端會加入 HTML 標示來轉換成 HTML。預設值是零。
<i>DateTimeFormat</i>	可選用日期和時間值的格式。有效值包括 0（ISO 格式）、1（USA 格式）、2（EUR 格式）、3（JIS 格式）、或 4（Windows 控制台格式）。預設值是 4。

Format	輸出檔的格式。
UseFormPageSetup	可選用的，如果指定「非零」，表示要忽略 <i>PageLength</i> 、 <i>PageWidth</i> 、 <i>IncludeDateTime</i> 及 <i>IncludePageNumbers</i> 參數，這些參數的值應改採指定套表所儲存的值。預設值是「零」。

值	意義
0 (RSF_DEFAULT)	使用預設的套表。 <i>FormName</i> 要保持空白。
1 (RSF_DATABASE)	使用資料庫的套表。在 <i>FormName</i> 參數中指定套表擁有者和名稱 (Owner.Name)。如果要使用不同資料庫伺服器上的套表，請先使用 <i>ExportForm()</i> 將套表匯到一個檔案，然後指定 <i>SourceType</i> 為 <i>RSF_FILE</i> 。
2 (RSF_FILE)	使用一個檔案中的套表。在 <i>FormName</i> 參數中指定檔案名稱。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 *GetLastErrorString()*、*GetLastErrorType()*、*GetLastSQLCode()*、*GetLastSQLError()* 或 *GetLastSQLState()* 來取得其他的錯誤資訊。

### 相關的主題

*ExportForm()*

## FastSaveData()

short *FastSaveData*(long *QueryID*, BOOL *Replace*, BSTR *Tablename*, BSTR *TableSpaceName*, [VARIANT *Comment*])

### 說明

此函數可以為指定的查詢建立一個報告，並且寫入一個檔案。您可以為 QMF 套表中的報告指定格式和佈置。輸出檔是 ASCII 文字檔，每一行以一對回車和換行字元來區隔，每一頁以一個換頁字元來區隔。使用固定字距的字型來檢視輸出檔，效果最好。

### 參數

名稱	說明
<i>QueryID</i>	從 <i>InitializeQuery()</i> 傳回的查詢 ID。
<i>Replace</i>	如果您要以指定的資料來取代表格中現存的資料，請指定「非零」。如果您要將指定的資料附加到表格中現存的資料，請指定「零」。

TableName	您要用來儲存資料的表格名稱。如果表格不存在，QMF for Windows 會建立表格。
TableSpaceName	用來建立表格或表格所在的表格空間名稱。如果 <i>TableSpaceName</i> 是 NULL 或空字串，QMF for Windows 會使用預設的表格空間。如果您已將 QMF for Windows 配置成一律使用預設的表格空間，此參數會被忽略。請參閱 <code>GetResourceLimit()</code> 說明中的 <code>RSR_SDDIFFERENTTS</code> 。
Comment	可選用的字串，可以為儲存資料的表格指定註解。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## FetchNextRow()

short FetchNextRow(long *QueryID*, VARIANT\* *Row*)

### 說明

此函數從資料庫提取下一列資料。

### 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
Row	指向儲存結果的 VARIANT 之指標。結果是一個陣列 (variant 類型 VT_ARRAY   VT_VARIANT)，其中結果的每一個直欄各有一個值。請呼叫 <code>GetColumnCount()</code> 來決定陣列中值的數目。

每一個值都以其原來的資料類型或最接近的 variant 資料類型來指定。支援的傳回類型包括：`string` (variant 類型 VT\_BSTR)、`float` (variant 類型 VT\_R4)、`double` (variant 類型 VT\_R8)、`short` (variant 類型 VT\_I2)、`long` (variant 類型 VT\_I4)、以及 `binary` (variant 類型 VT\_UI1 | VT\_ARRAY)。

如果已到達結果集的尾端 (沒有其他列可以提取)，或結果集是空的，則結果為空白 (variant 類型 VT\_EMPTY)，不是陣列。

在呼叫此函數之前，您必須適當地起始設定 VARIANT。Visual Basic 會自動處理這件工作。Visual C++ 程式設計師必須呼叫 `VariantInit()`。



**註:** 由於 Microsoft Excel 7.0 和 Microsoft Access 7.0 中的設計瑕疵（也可能發生在其他使用 Visual Basic for Applications 的 32 位元 Microsoft 產品中），從 QMF for Windows 所接收的 Variant 變數的字串資料可能無法從 Unicode（使用於 OLE）轉換成 ANSI（使用於 VBA）。如果發生這種情形，則只顯示字串的第一個字元。為了解決這個問題，請在呼叫 QMF for Windows 函數之前，先將變數設定成空字串。

### 回覆值

成功則為「零」，失敗則為「非零」。當到達結果集的尾端時，回覆值為 -1。如果回覆值為「非零」，您可以呼叫

GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。

### 相關的主題

FetchNextRows()

## FetchNextRowEx()

short FetchNextRowEx(long *QueryID*)

### 說明

此函數從資料庫提取下一列資料。您可以在不支援 VARIANT 陣列的環境中使用此函數，例如在 Microsoft Access 2.0 中。如果要擷取現行列中每一欄的資料，請搭配 GetColumnValue() 來使用此函數。

### 參數

名稱	說明
<i>QueryID</i>	從 InitializeQuery() 傳回的查詢 ID。

### 回覆值

成功則為「零」，失敗則為「非零」。當到達結果集的尾端時，回覆值為 -1。如果回覆值為「非零」，您可以呼叫

GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。

### 相關的主題

FetchNextRowsEx()

## FetchNextRows()

short FetchNextRows(long *QueryID*, VARIANT\* *Rows*, long\* *NumRows*)

## 說明

此函數從資料庫提取下 *NumRows* 列資料。

## 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
Row	<p>指向儲存結果的 <code>VARIANT</code> 之指標。結果是一個二維陣列 (<code>variant</code> 類型 <code>VT_ARRAY</code>   <code>VT_VARIANT</code>)，其中每一列的每一欄各有一個值。請呼叫 <code>GetColumnCount()</code> 來決定陣列中的直欄數。陣列的維度是 [<i>NumRows</i>][<i>ColumnCount</i>]，即使結果集中未提取的列數少於 <i>NumRows</i> (如果是這樣，陣列含有額外尚未使用的項目) 也一樣。</p> <p>每一個值都以其原來的資料類型或最接近的 <code>variant</code> 資料類型來指定。支援的傳回類型包括：<code>string</code> (<code>variant</code> 類型 <code>VT_BSTR</code>)、<code>float</code> (<code>variant</code> 類型 <code>VT_R4</code>)、<code>double</code> (<code>variant</code> 類型 <code>VT_R8</code>)、<code>short</code> (<code>variant</code> 類型 <code>VT_I2</code>)、<code>long</code> (<code>variant</code> 類型 <code>VT_I4</code>)、以及 <code>binary</code> (<code>variant</code> 類型 <code>VT_UI1</code>   <code>VT_ARRAY</code>)。</p> <p>如果已到達結果集的尾端 (沒有其他列可以提取)，或結果集是空的，則結果為空白 (<code>variant</code> 類型 <code>VT_EMPTY</code>)，不是陣列。</p> <p>在呼叫此函數之前，您必須適當地起始設定 <code>VARIANT</code>。<code>Visual Basic</code> 會自動處理這件工作。<code>Visual C++</code> 程式設計師必須呼叫 <code>VariantInit()</code>。</p>
<i>NumRows</i>	指向一個 <code>long</code> 的指標，其中含有要提取的列數。如果結果集中未提取的列數少於 <i>NumRows</i> ， <i>NumRows</i> 會重設為結果中的實際列數。

**註：**由於 Microsoft Excel 7.0 和 Microsoft Access 7.0 中的設計瑕疵 (也可能發生在其他使用 `Visual Basic for Applications` 的 32 位元 Microsoft 產品中)，從 `QMF for Windows` 所接收的 `Variant` 變數的字串資料可能無法從 `Unicode` (使用於 `OLE`) 轉換成 `ANSI` (使用於 `VBA`)。如果發生這種情形，則只顯示字串的第一個字元。為了解決這個問題，請在呼叫 `QMF for Windows` 函數之前，先將變數設定成空字串。

## 回覆值

成功則為「零」，失敗則為「非零」。當到達結果集的尾端時，回覆值為 -1。如果回覆值為「非零」，您可以呼叫

`GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## 相關的主題

FetchNextRow()

## FetchNextRowsEx()

short FetchNextRowsEx(long *QueryID*, long\* *NumRows*)

### 說明

此函數從資料庫提取下 *NumRows* 列資料。您可以在不支援 **VARIANT** 陣列的環境中使用此函數，例如在 Microsoft Access 2.0 中。如果要擷取特定列中每一欄的資料，請搭配 `GetColumnValueEx()` 來使用此函數。

### 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
<i>NumRows</i>	指向一個 <code>long</code> 的指標，其中含有要提取的列數。如果結果集中未提取的列數少於 <i>NumRows</i> ， <i>NumRows</i> 會重設為結果中的實際列數。

### 回覆值

成功則為「零」，失敗則為「非零」。當到達結果集的尾端時，回覆值為 -1。如果回覆值為「非零」，您可以呼叫

`GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## 相關的主題

FetchNextRowEx()

## FlushQMFCache()

void FlushQMFCache()

### 說明

此函數指示 QMF for Windows 去清除 QMF 快取資訊，捨棄其內容。QMF for Windows 下次需要 QMF 資訊時，必須從資料庫中取得。QMF for Windows 通常將從資料庫取得的 QMF 資訊放入快取記憶體中，以減少資料庫資料傳輸，增進效率。在呼叫 `GetQMFOBJECTInfo()`、`GetQMFCacheText()` 或 `GetQMFOBJECTList()` 之前，您可以先呼叫此函數，確保傳回最新的資訊。

### 回覆值

無。

## GetColumnCount()

long GetColumnCount(long *QueryID*)

### 說明

此函數傳回結果集的直欄數。

### 參數

名稱	說明
<i>QueryID</i>	從 InitializeQuery() 傳回的查詢 ID。

### 回覆值

成功則傳回每一列的直欄數。失敗則傳回 0 或 -1。如果回覆值是 0 或 -1，您可以呼叫 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。

## GetColumnDataValue()

short GetColumnDataValue(long *QueryID*, long *Index*)

### 說明

此函數傳回 *Index* 所指定之直欄的現行資料列的資料值。呼叫此函數之後，即可查詢 *Value* 內容來取得回覆值。如果要存取單一資料列的資料，請搭配 FetchNextRowEx() 來使用此函數。

### 參數

名稱	說明
<i>QueryID</i>	從 InitializeQuery() 傳回的查詢 ID。
<i>Index</i>	所要擷取之列資料的零基準索引。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。

## GetColumnHeader()

BSTR GetColumnHeader(long *QueryID*, long *Index*, short\* *Result*)

## 說明

此函數傳回與索引 *Index* 相關聯的直欄標題（直欄名稱）。

## 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
<i>Index</i>	所要擷取之列資料的零基準索引。
<i>Result</i>	成功則為「零」，失敗則為「非零」。如果 <i>Result</i> 為「非零」，您可以呼叫 <code>GetLastErrorString()</code> 、 <code>GetLastErrorType()</code> 、 <code>GetLastSQLCode()</code> 、 <code>GetLastSQLError()</code> 或 <code>GetLastSQLState()</code> 來取得其他的錯誤資訊。

**註：** 直欄標題不適用於靜態 SQL 陳述式。對於 `InitializeStaticQuery()` 所傳回的查詢 ID，`GetColumnHeader` 會傳回 "Coln" 形式的字串，"n" 代表直欄號碼。

## 回覆值

傳回的字串代表 *Index* 參數所指定的直欄名稱。

## GetColumnHeaderEx()

`short GetColumnHeaderEx(long QueryID, long Index)`

## 說明

此函數傳回與索引 *Index* 相關聯的直欄標題（直欄名稱）。呼叫此函數之後，即可查詢 *Value* 內容來取得回覆值。

## 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
<i>Index</i>	所要擷取之列資料的零基準索引。

**註：** 直欄標題不適用於靜態 SQL 陳述式。對於 `InitializeStaticQuery()` 所傳回的查詢 ID，`GetColumnHeader` 會傳回 "Coln" 形式的字串，"n" 代表直欄號碼。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值是「零」，請查詢 *Value* 內容來取得直欄名稱的字串。如果回覆值為「非零」，您可以呼叫

GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。

## GetColumnHeadings()

short GetColumnHeadings(long *QueryID*, VARIANT\* *Headings*)

### 說明

此函數傳回直欄標題（亦稱為直欄名稱）。

### 參數

名稱	說明
<i>QueryID</i>	從 InitializeQuery() 傳回的查詢 ID。
Headings	指向儲存結果的 VARIANT 之指標。結果是一個字串陣列（variant 類型 VT_ARRAY   VT_BSTR），其中每一個直欄標題各有一個字串。  在呼叫此函數之前，您必須適當地起始設定 VARIANT。Visual Basic 會自動處理這件工作。Visual C++ 程式設計師必須呼叫 VariantInit()。

**註：**由於 Microsoft Excel 7.0 和 Microsoft Access 7.0 中的設計瑕疵（也可能發生在其他使用 Visual Basic for Applications 的 32 位元 Microsoft 產品中），從 QMF for Windows 所接收的 Variant 變數的字串資料可能無法從 Unicode（使用於 OLE）轉換成 ANSI（使用於 VBA）。如果發生這種情形，則只顯示字串的第一個字元。為了解決這個問題，請在呼叫 QMF for Windows 函數之前，先將變數設定成空字串。

**註：**直欄標題不適用於靜態 SQL 陳述式。對於 InitializeStaticQuery() 所傳回的查詢 ID，GetColumnHeadings 會傳回 "Col1"、"Col2" 等字串。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。

## GetColumnValue()

short GetColumnValue(long *QueryID*, long *Index*, VARIANT\* *Value*)

## 說明

此函數傳回 *Index* 所指定之直欄的現行資料列的資料值。如果要存取單一資料列的資料，請搭配 `FetchNextRowEx()` 來使用此函數。

## 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
<i>Index</i>	所要擷取之列資料的零基準索引。
<i>Value</i>	指向您要用來儲存結果的 <code>VARIANT</code> 之指標。結果是一個依據 <code>variant</code> 類型的資料值。  在呼叫此函數之前，您必須適當地起始設定 <code>VARIANT</code> 。Visual Basic 會自動處理這件工作。Visual C++ 程式設計師必須呼叫 <code>VariantInit()</code> 。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## GetColumnValueEx()

`short GetColumnValueEx(long QueryID, long RowIndex, long ColIndex, VARIANT* Value)`

## 說明

此函數傳回 *ColIndex* 所指定之直欄和 *RowIndex* 所指定之資料列的資料值。如果要存取單一資料列的資料，請搭配 `FetchNextRowsEx()` 來使用此函數。

## 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
<i>RowIndex</i>	所要擷取之列的零基準索引。
<i>ColIndex</i>	所要擷取之直欄的零基準索引。

值	指向您要用來儲存結果的 <code>VARIANT</code> 之指標。您可以查詢結果的 <code>variant</code> 來瞭解進一步處理的資料類型。  在呼叫此函數之前，您必須適當地起始設定 <code>VARIANT</code> 。Visual Basic 會自動處理這件工作。Visual C++ 程式設計師必須呼叫 <code>VariantInit()</code> 。
---	---

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## GetDefaultServerName()

`BSTR GetDefaultServerName()`

### 說明

此函數傳回一個含有預設伺服器名稱的字串。

### 回覆值

字串，指定預設的伺服器名稱。

## GetGlobalVariable()

`BSTR GetGlobalVariable(BSTR Name)`

### 說明

此函數擷取指定之廣域變數的值。

### 參數

名稱	說明
Name	字串，含有您要設定之變數的名稱。

### 回覆值

字串，含有廣域變數值，如果變數沒有值或發生錯誤，則含 `NULL`。

## GetHostVariableNames()

`short GetHostVariableNames(long QueryID, VARIANT* Names)`



## 說明

此函數傳回一個名稱陣列，代表指定之查詢中所參照的全部主變數。查詢必須是參照主變數的靜態查詢（由 QMF 查詢所儲存，或由 `AddHostVariable()` 所建立）。

## 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
Names	指向您要用來儲存結果陣列的 <code>VARIANT</code> 之指標。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()` 來取得其他的錯誤資訊。

## GetHostVariableTypes()

`short GetHostVariableTypes(long QueryID, VARIANT* Types)`

## 說明

此函數傳回一個資料類型陣列，代表指定之查詢中所參照的全部主變數。查詢必須是靜態查詢參照主變數（由 QMF 查詢所儲存，或由 `AddHostVariable()` 所建立）。關於可傳回的資料類型清單，請參閱 `AddHostVariable()`。

## 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
Types	指向您要用來儲存結果陣列的 <code>VARIANT</code> 之指標。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()` 來取得其他的錯誤資訊。

## GetLastErrorString()

`BSTR GetLastErrorString()`

## 說明

此函數傳回一個字串，其中含有最近錯誤的相關資訊。如果您在一個函數順利執行之後（沒有錯誤）呼叫此函數，此函數會傳回前一個函數呼叫期間發生的最後一個錯誤的相關資訊。為避免混淆，請一律在傳回錯誤的函數呼叫之後立即呼叫此函數。

## 回覆值

含有錯誤資訊的字串。如果自建立 QMF API 物件之後沒有發生錯誤，則傳回 NULL。

## 相關的主題

GetLastErrorType()

GetLastSQLCode()

GetLastSQLError()

GetLastSQLState()

## GetLastErrorType()

short GetLastErrorType()

## 說明

此函數傳回最近錯誤的類型。如果您在一個函數順利執行之後（沒有錯誤）呼叫此函數，此函數會傳回前一個函數呼叫期間發生的最後一個錯誤的相關資訊。為避免混淆，請一律在傳回錯誤的函數呼叫之後立即呼叫此函數。

## 回覆值

回覆值指出錯誤的類型：

值	意義
0 (RS_ERROR_NONE)	自從 QMF for Windows API 物件建立之後沒有發生錯誤。
1 (RS_ERROR_SQL)	發生 SQL 錯誤。如果在以 <i>QueryID</i> 為引數的函數呼叫期間發生錯誤，請呼叫 <i>Close()</i> 來關閉查詢。不會執行回復。雖然可能遭遇其他錯誤，您仍可繼續使用 QMF for Windows API 物件。
2 (RS_ERROR_USER_CANCEL)	使用者取消作業，通常是在作用中視窗上按一下「取消」來達成。這會造成 QMF for Windows 執行隱含的回復（使所有未完成的查詢 ID 無效），並終止資料庫的連線。您必須呼叫 <i>InitializeServer()</i> 或 <i>ReinitializeServer()</i> 才能繼續。

3 (RS_ERROR_FATAL_GOV)	發生嚴重的支配程式錯誤。可能是因為超過最長閒置時間，造成 QMF for Windows API 逾時。這會造成 QMF for Windows 執行隱含的回復（使所有未完成的查詢 ID 無效），並終止資料庫的連線。您必須呼叫 InitializeServer() 或 ReinitializeServer() 才能繼續。
4 (RS_ERROR_NONFATAL_GOV)	發生不嚴重的支配程式錯誤。可能是超過可提取的最大列數，或不容許 SQL Verb。如果在以 QueryID 為引數的函數呼叫期間發生錯誤，請呼叫 Close() 來關閉該查詢。不會執行回復，資料庫連線亦不受影響，所以您可以繼續使用 QMF for Windows API 物件。
5 (RS_ERROR_OTHER)	發生一般錯誤。不會執行回復。雖然可能遭遇其他錯誤，您仍可繼續使用 QMF for Windows API 物件。

### 相關的主題

[GetLastErrorString\(\)](#)  
[GetLastSQLCode\(\)](#)  
[GetLastSQLError\(\)](#)  
[GetLastSQLState\(\)](#)

## GetLastSQLCode()

long GetLastErrorString()

### 說明

此函數傳回最近錯誤的 SQL 程式碼。如果您在一個函數順利執行之後（沒有錯誤）呼叫此函數，此函數會傳回前一個函數呼叫期間發生的最後一個錯誤的相關資訊。為避免混淆，請一律在傳回錯誤的函數呼叫之後立即呼叫此函數。

### 回覆值

最近錯誤的 SQL 程式碼。如果自您建立 QMF for Windows API 物件之後沒有發生錯誤，或最近的錯誤不是 SQL 錯誤，則傳回「零」。

### 相關的主題

[GetLastErrorString\(\)](#)  
[GetLastErrorType\(\)](#)  
[GetLastSQLError\(\)](#)  
[GetLastSQLState\(\)](#)

## GetLastSQLError()

VARIANT GetLastSQLError()

### 說明

此函數傳回最近錯誤的詳細 SQL 錯誤資訊。如果您在一個函數順利執行之後（沒有錯誤）呼叫此函數，此函數會傳回前一個函數呼叫期間發生的最後一個錯誤的相關資訊。為避免混淆，請一律在傳回錯誤的函數呼叫之後立即呼叫此函數。

### 回覆值

一個含有錯誤資訊的陣列（variant 類型 VT\_ARRAY | VT\_VARIANT）。如果您建立 QMF for Windows API 物件之後沒有發生錯誤，或最近的錯誤不是 SQL 錯誤，則傳回空白（variant 類型 VT\_EMPTY）。陣列有下列格式：

元素	類型	內容
0	long (VT_I4)	程式碼
1	字串 (VT_BSTR)	狀態
2	字串 (VT_BSTR)	ErrProc
3	字串 (VT_BSTR)	RDBName
4	long (VT_I4)	ErrD1
5	long (VT_I4)	ErrD2
6	long (VT_I4)	ErrD3
7	long (VT_I4)	ErrD4
8	long (VT_I4)	ErrD5
9	long (VT_I4)	ErrD6
10	字串 (VT_BSTR)	Warn0
11	字串 (VT_BSTR)	Warn1
12	字串 (VT_BSTR)	Warn2
13	字串 (VT_BSTR)	Warn3
14	字串 (VT_BSTR)	Warn4
15	字串 (VT_BSTR)	Warn5
16	字串 (VT_BSTR)	Warn6
17	字串 (VT_BSTR)	Warn7
18	字串 (VT_BSTR)	Warn8
19	字串 (VT_BSTR)	Warn9
20	字串 (VT_BSTR)	WarnA
21	字串 (VT_BSTR)	MessageTokens

## 相關的主題

GetLastErrorString()  
GetLastErrorType()  
GetLastSQLCode()  
GetLastSQLState()

## GetLastSQLState()

BSTR GetLastSQLState()

### 說明

此函數傳回最近錯誤的 SQL 狀態。如果您在一個函數順利執行之後（沒有錯誤）呼叫此函數，此函數會傳回前一個函數呼叫期間發生的最後一個錯誤的相關資訊。為避免混淆，請一律在傳回錯誤的函數呼叫之後立即呼叫此函數。

### 回覆值

字串，含有最近錯誤的 SQL 程式碼。如果自您建立 QMF for Windows API 物件之後沒有發生錯誤，或最近的錯誤不是 SQL 錯誤，則傳回 NULL。

## 相關的主題

GetLastErrorString()  
GetLastErrorType()  
GetLastSQLCode()  
GetLastSQLError()

## GetOption()

short GetOption(short *Option*, VARIANT\* *Value*)

### 說明

取得 QMF for Windows 中指定的選項值。

### 參數

名稱	說明
<i>Option</i>	指定要擷取的選項。

值	意義
0 (RSO_SERVER_DEFINITION_FILE)	伺服器定義檔名稱。
1 (RSO_CPIC_DLL)	CPI-C 提供程式 DLL 檔案名稱。
2 (RSO_CPIC_TIMEOUT_WARNING)	CPI-C 警告逾時值（以秒計）。QMF for Windows API 不使用此限制。

3 (RSO_CPIC_TIMEOUT_CANCEL)	CPI-C 取消逾時值（以秒計）。
4 (RSO_TCP_TIMEOUT_WARNING)	TCP 警告逾時值（以秒計）。QMF for Windows API 不使用此限制。
5 (RSO_TCP_TIMEOUT_CANCEL)	TCP 取消逾時值（以秒計）。
6 (RSO_DISPLAY_NULLS_STRING)	用來顯示空值的字串。
7 (RSO_ENTER_NULLS_STRING)	用來輸入空值的字串。
8 (RSO_ENTER_DEFAULTS_STRING)	用來輸入預設值的字串。
9 (RSO_TRACE_FILE_1)	追蹤檔 1 名稱。
10 (RSO_TRACE_FILE_2)	追蹤檔 2 名稱。
11 (RSO_TCP_TRACE_LEVEL)	TCP 追蹤層次。
12 (RSO_CPIC_TRACE_LEVEL)	CPI-C 追蹤層次。
13 (RSO_DDM_TRACE_LEVEL)	DDM 追蹤層次。

值	指向儲存結果的 VARIANT 之指標。結果是一個陣列（variant 類型 VT_ARRAY   VT_VARIANT），其中結果的每一個直欄各有一個值。請呼叫 GetColumnCount() 來決定陣列中的值數目。在呼叫此函數之前，您必須適當地起始設定 VARIANT。Visual Basic 會自動處理這件工作。Visual C++ 程式設計師必須呼叫 VariantInit()。
---	--

**註：**由於 Microsoft Excel 7.0 和 Microsoft Access 7.0 中的設計瑕疵（也可能發生在其他使用 Visual Basic for Applications 的 32 位元 Microsoft 產品中），從 QMF for Windows 所接收的 Variant 變數的字串資料可能無法從 Unicode（使用於 OLE）轉換成 ANSI（使用於 VBA）。如果發生這種情形，則只顯示字串的第一個字元。為了解決這個問題，請在呼叫 QMF for Windows 函數之前，先將變數設定成空字串。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

### 相關的主題

SetOption()

## GetOptionEx()

short GetOptionEx(short *Option*)

## 說明

取得 QMF for Windows 中指定的選項值。當選項值傳回時，您必須查詢 *Option* 內容來取得資料。

## 參數

名稱	說明
Option	選項值同於 <code>GetOption()</code> 呼叫。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()` 或 `GetLastErrorType()` 來取得其他的錯誤資訊。

## 相關的主題

`GetOption()`

`SetOption()`

## GetProcText()

BSTR `GetProcText(long ProcID)`

## 說明

此函數傳回指定的程序在取代變數之後所執行的文字。您應該在呼叫此函數之前，使用 `SetProcVariable()` 來設定程序中任何變數的值。

## 參數

名稱	說明
ProcID	從 <code>InitializeProc()</code> 傳回的程序 ID。

## 回覆值

如果成功，則傳回一個含有程序文字的字串。如果失敗，則傳回 `NULL`。如果回覆值是 `NULL`，您可以呼叫 `GetLastErrorString()` 或 `GetLastErrorType()` 來取得其他的錯誤資訊。

## GetProcVariables()

short `GetProcVariables(long ProcID, VARIANT* Variables)`

## 說明

取得 QMF for Windows 中指定的選項值。

## 參數

名稱	說明
ProcID	從 <code>InitializeProc()</code> 傳回的程序 ID。
值	指向儲存結果的 <code>VARIANT</code> 之指標。結果是一個字串陣列（ <code>variant</code> 類型 <code>VT_ARRAY   VT_BSTR</code> ），其中每一個字串含有一個變數的名稱。如果程序沒有變數，則結果是空白（ <code>variant</code> 類型 <code>VT_EMPTY</code> ）。在呼叫此函數之前，您必須適當地起始設定 <code>VARIANT</code> 。Visual Basic 會自動處理這件工作。Visual C++ 程式設計師必須呼叫 <code>VariantInit()</code> 。

**註：**由於 Microsoft Excel 7.0 和 Microsoft Access 7.0 中的設計瑕疵（也可能發生在其他使用 Visual Basic for Applications 的 32 位元 Microsoft 產品中），從 QMF for Windows 所接收的 Variant 變數的字串資料可能無法從 Unicode（使用於 OLE）轉換成 ANSI（使用於 VBA）。如果發生這種情形，則只顯示字串的第一個字元。為了解決這個問題，請在呼叫 QMF for Windows 函數之前，先將變數設定成空字串。

## 回覆值

成功則為「零」，失敗則為「非零」。如果程序沒有變數，則回覆值是 `RS_NO_ERROR_NO_DATA` (-1)。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()` 或 `GetLastErrorType()` 來取得其他的錯誤資訊。

## GetQMFOBJECTInfo()

`short GetQMFOBJECTInfo(BSTR OwnerAndName, short Type, short Time, VARIANT* Value)`

## 說明

此函數傳回 QMF 物件（套表或查詢）的相關資訊。傳回的資訊由 `Type` 和 `Time` 參數來指定。

## 參數

名稱	說明
OwnerAndName	以句點分隔的字串，含有您要擷取資訊之物件的擁有者和名稱。例如， <code>John.Query2</code>



值	指向儲存結果的 <code>VARIANT</code> 之指標。如果是 <code>RSI_TIMEUSED</code> 、 <code>RSI_TIMESRUN</code> 、 <code>RSI_TIMESCANCELLED</code> 及 <code>RSI_LEVEL</code> ，則結果是 <code>short</code> (variant 類型 <code>VT_I2</code> )。如果是 <code>RSI_RESTRICTED</code> ，結果是 <code>Boolean</code> (variant 類型 <code>VT_BOOL</code> )。至於其他值，結果是字串 (variant 類型 <code>VT_BSTR</code> )。在呼叫此函數之前，您必須適當地起始設定 <code>VARIANT</code> 。Visual Basic 會自動處理這件工作。Visual C++ 程式設計師必須呼叫 <code>VariantInit()</code> 。
---	--

**註:** 由於 Microsoft Excel 7.0 和 Microsoft Access 7.0 中的設計瑕疵 (也可能發生在其他使用 Visual Basic for Applications 的 32 位元 Microsoft 產品中)，從 QMF for Windows 所接收的 Variant 變數的字串資料可能無法從 Unicode (使用於 OLE) 轉換成 ANSI (使用於 VBA)。如果發生這種情形，則只顯示字串的第一個字元。為了解決這個問題，請在呼叫 QMF for Windows 函數之前，先將變數設定成空字串。

<i>Type</i>	指定要取得的資訊類型。
-------------	-------------

值	意義
0 ( <code>RSI_COMMENT</code> )	註解
1 ( <code>RSI_LEVEL</code> )	層次
2 ( <code>RSI_TYPE</code> )	類型
3 ( <code>RSI_SUBTYPE</code> )	子類型
4 ( <code>RSI_RESTRICTED</code> )	限制
5 ( <code>RSI_MODEL</code> )	模型
6 ( <code>RSI_TIMESUSED</code> )	使用次數。
7 ( <code>RSI_TIMESRUN</code> )	執行次數。
8 ( <code>RSI_TIMESCANCELLED</code> )	取消次數。
9 ( <code>RSI_DATE</code> )	首次使用、上次使用、或上次修改日期。
10 ( <code>RSI_TIME</code> )	首次使用、上次使用、或上次修改時間。
11 ( <code>RSI_USERID</code> )	首次使用、上次使用、或上次修改的使用者 ID。
12 ( <code>RSI_SQLID</code> )	首次使用、上次使用、或上次修改的 SQL ID。
13 ( <code>RSI_ENVIRONMENT</code> )	首次使用、上次使用、或上次修改的環境。
14 ( <code>RSI_MODE</code> )	首次使用、上次使用、或上次修改的模式。
15 ( <code>RSI_COMMAND</code> )	首次使用、上次使用、或上次修改的指令。

<i>Time</i>	指定首次使用、上次使用、或上次修改時間。
-------------	----------------------

值	意義
0 (RST_FIRSTUSED)	首次使用。
1 (RST_LASTUSED)	上次使用。
2 (RST_LASTMODIFIED)	上次修改。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## GetQMFOBJECTInfoEx()

`short GetQMFOBJECTInfoEx(BSTR OwnerAndName, short Type, short Time)`

### 說明

此函數傳回 QMF 物件的相關資訊。傳回的資訊由 *Type* 和 *Time* 參數來指定。呼叫此函數之後，即可查詢 *QMFOBJECTInfo* 內容來取得回覆值。

### 參數

名稱	說明
OwnerAndName	以句點分隔的字串，含有您要擷取資訊之物件的擁有者和名稱。例如， <code>John.Query2</code>
Type	指定要取得的資訊類型。

值	意義
0 (RSI_COMMENT)	註解
1 (RSI_LEVEL)	層次
2 (RSI_TYPE)	類型
3 (RSI_SUBTYPE)	子類型
4 (RSI_RESTRICTED)	限制
5 (RSI_MODEL)	模型
6 (RSI_TIMESUSED)	使用次數。
7 (RSI_TIMESRUN)	執行次數。
8 (RSI_TIMESCANCELLED)	取消次數。
9 (RSI_DATE)	首次使用、上次使用、或上次修改日期。

10 (RSL_TIME)	首次使用、上次使用、或上次修改時間。
11 (RSL_USERID)	首次使用、上次使用、或上次修改的使用者 ID。
12 (RSL_SQLID)	首次使用、上次使用、或上次修改的 SQL ID。
13 (RSL_ENVIRONMENT)	首次使用、上次使用、或上次修改的環境。
14 (RSL_MODE)	首次使用、上次使用、或上次修改的模式。
15 (RSL_COMMAND)	首次使用、上次使用、或上次修改的指令。

*Time* 指定首次使用、上次使用、或上次修改時間。

值	意義
0 (RST_FIRSTUSED)	首次使用。
1 (RST_LASTUSED)	上次使用。
2 (RST_LASTMODIFIED)	上次修改。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## GetQMFOBJECTList()

short GetQMFOBJECTList(BSTR *Owner*, BSTR *Name*, short *Type*, VARIANT\* *List*)

### 說明

此函數傳回一個名稱陣列，代表符合 *Owner* 和 *Name* 參數所指定之型樣的全部 QMF 物件。

### 參數

名稱	說明
<i>Owner</i>	字串，含有您在傳回清單中要包含的物件擁有者。
<i>Name</i>	字串，含有您在傳回清單中要包含的物件名稱。
<i>List</i>	指向儲存結果的 VARIANT 之指標。結果是一個字串陣列（variant 類型 VT_ARRAY   VT_BSTR），每一個字串的格式為 <i>Owner.Name</i> 。如果找不到符合的 QMF for Windows 查詢，則結果是空白（variant 類型 VT_EMPTY）。在呼叫此函數之前，您必須適當地起始設定 VARIANT。Visual Basic 會自動處理這件工作。Visual C++ 程式設計師必須呼叫 <code>VariantInit()</code> 。

**註:** 由於 Microsoft Excel 7.0 和 Microsoft Access 7.0 中的設計瑕疵（也可能發生在其他使用 Visual Basic for Applications 的 32 位元 Microsoft 產品中），從 QMF for Windows 所接收的 Variant 變數的字串資料可能無法從 Unicode（使用於 OLE）轉換成 ANSI（使用於 VBA）。如果發生這種情形，則只顯示字串的第一個字元。為了解決這個問題，請在呼叫 QMF for Windows 函數之前，先將變數設定成空字串。

---

<i>Type</i>	指定您要在清單中包含的 QMF 物件類型。這些值可以加在一起來指定多重物件類型。
-------------	--

---

值	意義
2048 (RSQ_MASK_QUERIES)	在清單中包含 QMF 查詢。
1024 (RSQ_MASK_FORMS)	在清單中包含 QMF 套表。
512 (RSQ_MASK_PROCS)	在清單中包含 QMF 程序。
256 (RSQ_MASK_TABLES)	在清單中包含表格。

### 回覆值

成功則為「零」，失敗則為「非零」。如果找不到相符的 QMF 物件，則回覆值為「零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## GetQMFOBJECTListEx()

short GetQMFOBJECTListEx(BSTR *Owner*, BSTR *Name*, short *Index*)

### 說明

此函數傳回 QMF 物件的名稱，這些物件符合 *Index* 參數所參照之 *Owner* 和 *Name* 參數所指定的型樣。呼叫此函數之後，即可查詢 *Value* 內容來取得回覆值。

### 參數

名稱	說明
<i>Owner</i>	字串，含有您在傳回清單中要包含的物件擁有者。
<i>Name</i>	字串，含有您在傳回清單中要包含的物件名稱。
<i>Index</i>	符合型樣之 QMF 物件清單的索引。

---

<i>Type</i>	指定您要在清單中包含的 QMF 物件類型。這些值可以加在一起來指定多重物件類型。
-------------	--

---

值	意義
2048 (RSQ_MASK_QUERIES)	在清單中包含 QMF 查詢。
1024 (RSQ_MASK_FORMS)	在清單中包含 QMF 套表。
512 (RSQ_MASK_PROCS)	在清單中包含 QMF 程序。
256 (RSQ_MASK_TABLES)	在清單中包含表格。

### 回覆值

成功則為「零」，失敗則為「非零」。如果找不到相符的 QMF 物件，則回覆值為 `RS_ERROR_OUTOFRANGE`。如果回覆值不是「零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## GetQMFProcText()

BSTR GetQMFProcText(BSTR *OwnerAndName*)

### 說明

此函數傳回指定的程序在取代變數之後所執行的文字。您應該在呼叫此函數之前，使用 `SetProcVariable()` 來設定程序中任何變數的值。

### 參數

名稱	說明
<code>OwnerAndName</code>	以句點分隔的字串，含有您要刪除之物件的擁有者和名稱。例如， <code>John.Proc2</code>

### 回覆值

字串，含有被擷取的程序文字，如果無法擷取程序，則傳回 `NULL`。如果回覆值是 `NULL`，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## GetQMFQueryText()

BSTR GetQMFQueryText(BSTR *OwnerAndName*)

### 說明

此函數擷取指定查詢中儲存的 SQL 文字。

## 參數

名稱	說明
OwnerAndName	以句點分隔的字串，含有您要刪除之物件的擁有者和名稱。例如，  John.Query2

## 回覆值

字串，含有被擷取的查詢文字，如果無法擷取查詢，則傳回 NULL。如果回覆值是 NULL，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## GetQueryText()

BSTR GetQueryText(long *QueryID*)

## 說明

此函數傳回指定的查詢在取代變數之後所執行的 SQL 文字。您應該在呼叫此函數之前，使用 `SetVariable()` 來設定查詢中任何變數的值。

## 參數

名稱	說明
QueryID	從 <code>InitializeQuery()</code> 傳回的查詢 ID。

**註：**查詢文字不適用於靜態 SQL 陳述式。對於 `InitializeStaticQuery()` 所傳回的查詢 ID，`GetQueryText()` 會傳回空字串。

## 回覆值

如果成功，則傳回一個含有 SQL 文字的字串。如果失敗，則傳回 NULL。如果回覆值是 NULL，您可以呼叫 `GetLastErrorString()` 或 `GetLastErrorType()` 來取得其他的錯誤資訊。

## GetQueryVerb()

BSTR GetQueryVerb(long *QueryID*)

## 說明

此函數傳回一個字串，其中含有您在查詢中使用的 SQL Verb。

## 參數

名稱	說明
QueryID	從 InitializeQuery() 傳回的查詢 ID。

**註:** 查詢動詞不適用於靜態 SQL 陳述式。對於 InitializeStaticQuery() 所傳回的查詢 ID，GetQueryVerb() 會傳回空字串。

## 回覆值

如果成功，則傳回一個含有 SQL Verb 的字串。如果失敗，則傳回 NULL。如果回覆值是 NULL，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

## GetResourceLimit()

short GetResourceLimit(short *Resource*, long\* *Value*)

### 說明

此函數取得所要求的資源限制。在呼叫此函數之前，您必須先呼叫 InitializeServer()，因為資源限制是按照每一個伺服器來處理的。

### 參數

名稱	說明
Resource	資源值包括：

值	意義
0 (RSR_IDLE_CONNECTION_TIMEOUT)	閒置連線逾時值（以秒計）。
1 (RSR_IDLE_QUERY_TIMEOUT_CANCEL)	閒置查詢逾時值（以秒計）。
2 (RSR_IDLE_QUERY_TIMEOUT_WARNING)	閒置查詢逾時值（以秒計）。此為警告限制；不對 QMF for Windows API 強制施行。
3 (RSR_SERVER_RESPONSE_TIMEOUT_CANCEL)	伺服器逾時值（以秒計）。
4 (RSR_SERVER_RESPONSE_TIMEOUT_WARNING)	伺服器逾時值（以秒計）。此為警告限制；不對 QMF for Windows API 強制施行。
5 (RSR_MAX_ROWS_TO_FETCH_CANCEL)	要提取的最大列數。

6 (RSR_MAX_ROWS_TO_FETCH_WARNING)	要提取的最大列數。此為警告限制；不對 QMF for Windows API 強制施行。
7 (RSR_MAX_BYTES_TO_FETCH_CANCEL)	要提取的最大位元組數。
8 (RSR_MAX_BYTES_TO_FETCH_WARNING)	要提取的最大位元組數。此為警告限制；不對 QMF for Windows API 強制施行。
9 (RSR_MAX_CONNECTIONS)	容許連接至資料庫伺服器的最大連線數。
10 (RSR_ALLOW_SERVER_ACCESS_UI)	是否容許從 QMF for Windows 介面來存取資料庫伺服器？
11 (RSR_ALLOW_SERVER_ACCESS_API)	是否容許從 QMF for Windows API 來存取資料庫伺服器？
12 (RSR_FETCH_ALL_ROWS)	是否自動提取所有列？
13 (RSR_CONFIRM_UPDATES)	是否確定資料庫伺服器更新？此選項不影響 QMF for Windows API；QMF for Windows API 不會確定資料庫更新。
14 (RSR_SUMMARY_TRACKING)	是否啟用摘要物件追蹤？
15 (RSR_DETAILED_TRACKING)	是否啟用明細物件追蹤？
16 (RSR_SQL_TRACKING)	是否啟用 SQL 文字追蹤？
17 (RSR_ADHOC_TRACKING)	是否啟用代用的物件追蹤？
18 (RSR_ALLOW_ACQUIRE)	是否容許 SQL Verb ACQUIRE？
19 (RSR_ALLOW_ALTER)	是否容許 SQL Verb ALTER？
20 (RSR_ALLOW_COMMENT)	是否容許 SQL Verb COMMENT？
21 (RSR_ALLOW_CREATE)	是否容許 SQL Verb CREATE？
22 (RSR_ALLOW_DELETE)	是否容許 SQL Verb DELETE？
23 (RSR_ALLOW_DROP)	是否容許 SQL Verb DROP？
24 (RSR_ALLOW_EXPLAIN)	是否容許 SQL Verb EXPLAIN？



25 (RSR_ALLOW_GRANT)	是否容許 GRANT?	SQL	Verb
26 (RSR_ALLOW_INSERT)	是否容許 INSERT?	SQL	Verb
27 (RSR_ALLOW_LABEL)	是否容許 LABEL?	SQL	Verb
28 (RSR_ALLOW_LOCK)	是否容許 LOCK?	SQL	Verb
29 (RSR_ALLOW_REVOKE)	是否容許 REVOKE?	SQL	Verb
30 (RSR_ALLOW_SELECT)	是否容許 SELECT?	SQL	Verb
31 (RSR_ALLOW_SET)	是否容許 SET?	SQL	Verb
32 (RSR_ALLOW_SIGNAL)	是否容許 SIGNAL?	SQL	Verb
33 (RSR_ALLOW_UPDATE)	是否容許 UPDATE?	SQL	Verb
34 (RSR_ALLOW_CALL)	是否容許 CALL?	SQL	Verb
35 (RSR_ALLOW_SAVE_DATA)	是否容許 Save Data 指令?		
36 (RSR_SAVE_DATA_TABLE_SPACE_NAME)	用來連結套件的預設集成名稱?		
37 (RSR_SAVE_DATA_TABLE_SPACE_NAME_OVERRIDE)	使用者可以置換 Save Data 指令的預設表格空間名稱嗎?		
38 (RSR_ALLOW_BIND_PACKAGE)	是否容許套件連結?		
39 (RSR_DEF_COLLECTION)	用來連結套件的預設集成名稱。		
40 (RSR_DEF_COLLECTION_OVERRIDE)	使用者可以置換套件連結的預設集成名稱嗎?		
41 (RSR_DEF_ISOLATION_LEVEL)	用來連結套件的預設隔離層次。		
42 (RSR_DEF_ISOLATION_LEVEL_OVERRIDE)	使用者可以置換套件連結的預設隔離層次嗎?		
43 (RSR_ALLOW_TABLE_EDIT)	是否容許使用表格編輯器?		
44 (RSR_ALLOW_EXPORT)	是否容許匯出資料?		
45 (RSR_ALLOW_SAVED_QUERIES_ONLY)	是否容許使用者只執行已儲存的查詢?		

46 (RSR_ALLOW_DROP_PACKAGE)	是否容許除去套件？
47 (RSR_QUERY_ISOLATION_LEVEL)	執行查詢時使用的隔離層次。
48 (RSR_ACCOUNT_STRING)	連接資料庫伺服器時要傳遞的帳戶資訊字串。
49 (RSR_ACCOUNT_OVERRIDE)	使用者可以置換在連接資料庫伺服器時所傳遞的帳戶資訊字串嗎？
值	指向儲存結果的 long 之指標。結果是所要求之資源限制的值。如果是 Boolean 值，結果為「非零」表示 true，「零」表示 false。如果是 RSR_SAVE_DATA_TABLE_SPACE_NAME、RSR_DEF_COLLECTION 及 RSR_ACCOUNT_STRING，則傳回 -1，可以查詢 <i>ResourceLimit</i> 內容來取得傳回的字串值。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()` 或 `GetLastErrorType()` 來取得其他的錯誤資訊。

## GetResourceLimitEx()

short GetResourceLimitEx(short *Resource*)

### 說明

此函數取得所要求的資源限制。因為資源限制是按照每一個伺服器來處理的，在呼叫此函數之前，您必須先呼叫 `InitializeServer()`，呼叫此函數之後，請查詢 *ResourceLimit* 內容來取得結果。

### 參數

名稱	說明
Resource	資源值同於 <code>GetResourceLimit()</code> 呼叫。

**註：**查詢動詞不適用於靜態 SQL 陳述式。對於 `InitializeStaticQuery()` 所傳回的查詢 ID，`GetQueryVerb()` 會傳回空字串。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()` 或 `GetLastErrorType()` 來取得其他的錯誤資訊。

## GetRowCount()

long GetRowCount(long *QueryID*)

### 說明

此函數傳回目前在 QMF for Windows 內部緩衝區的列數。此數目可能大於 FetchNextRow() 或 FetchNextRows() 所擷取的列數，因為 QMF for Windows 會將接收自資料庫的資料放入緩衝區。

此函數傳回已從資料庫中擷取的列數。如果您要擷取結果集的總列數，可以：

- 呼叫 CompleteQuery()，使用 FetchNextRow() 或 FetchNextRows() 來提取所有列。
- 呼叫 Open() 時指定 FetchAllRows = TRUE。

### 參數

名稱	說明
QueryID	從 InitializeQuery() 傳回的查詢 ID。

### 回覆值

如果成功，傳回列數（若尚未擷取任何列，則傳回 0），如果失敗，傳回 -1。如果是 1，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

## GetServerList()

short GetServerList(VARIANT\* *List*)

### 說明

此函數傳回一個陣列，其中包含 QMF for Windows 伺服器定義檔 (SDF) 中定義的資料庫伺服器名稱。如果您要使用 QMF for Windows API 來存取資料庫伺服器，您必須在 SDF 檔案中定義該伺服器。

### 參數

名稱	說明
List	指向儲存結果的 VARIANT 之指標。結果是一個字串陣列（variant 類型 VT_ARRAY   VT_BSTR），其中每一個字串含有一個資料庫伺服器的名稱。如果您尚未定義任何資料庫伺服器，則結果是空白（variant 類型 VT_EMPTY）。在呼叫此函數之前，您必須適當地起始設定 VARIANT。Visual Basic 會自動處理這件工作。Visual C++ 程式設計師必須呼叫 VariantInit()。

**註:** 由於 Microsoft Excel 7.0 和 Microsoft Access 7.0 中的設計瑕疵（也可能發生在其他使用 Visual Basic for Applications 的 32 位元 Microsoft 產品中），從 QMF for Windows 所接收的 Variant 變數的字串資料可能無法從 Unicode（使用於 OLE）轉換成 ANSI（使用於 VBA）。如果發生這種情形，則只顯示字串的第一個字元。為了解決這個問題，請在呼叫 QMF for Windows 函數之前，先將變數設定成空字串。

### 回覆值

成功則為「零」，失敗則為「非零」。如果您尚未定義任何資料庫伺服器，則回覆值是「零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

## GetServerListEx()

short GetServerListEx(short *Index*)

### 說明

此函數擷取 *Index* 參數所參照的伺服器名稱。呼叫此函數之後，即可查詢 *Value* 內容來取得回覆值。

### 參數

名稱	說明
<i>Index</i>	伺服器清單中的索引。

### 回覆值

成功則為「零」，索引大於可用的伺服器數目時為 RS\_OUTOFRANGE，失敗則為「非零」。如果您尚未定義任何資料庫伺服器，則回覆值是 RS\_OUTOFRANGE。如果回覆值為「非零」，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

## GetStoredProcedureResultSets()

short GetStoredProcedureResultSets(long *QueryID*, VARIANT\* *ResultSets*)

### 說明

此函數傳回儲存程序（以原始 *QueryID* 來執行）所傳回的結果集之查詢 ID。每一個傳回的查詢 ID 可以使用於 FetchNextRow() 或 FetchNextRows() 來擷取結果集列，到達每一個結果集的尾端時，則使用 Close()。

## 參數

名稱	說明
QueryID	從 <code>InitializeQuery()</code> 傳回的原始查詢 ID。
ResultSets	指向 <code>VARIANT</code> 的指標，其中儲存結果集的查詢 ID。結果是一個 <code>long</code> 整數陣列（ <code>variant</code> 類型 <code>VT_ARRAY   VT_I4</code> ），其中每一個整數是對應之結果集的查詢 ID。如果儲存程序未傳回結果集，則結果為空白（ <code>variant</code> 類型 <code>VT_EMPTY</code> ）。在呼叫此函數之前，您必須適當地起始設定 <code>VARIANT</code> 。Visual Basic 會自動處理這件工作。Visual C++ 程式設計師必須呼叫 <code>VariantInit()</code> 。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## GetVariables()

short `GetVariables(long QueryID, VARIANT* Variables)`

### 說明

此函數傳回一個名稱陣列，代表查詢的 SQL 文字中的變數。在使用 `Open()` 或 `Execute()` 來執行查詢之前，您必須呼叫 `SetVariable()` 來指定這些變數的值。

### 參數

名稱	說明
QueryID	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
Variables	指向儲存結果的 <code>VARIANT</code> 之指標。結果是一個字串陣列（ <code>variant</code> 類型 <code>VT_ARRAY   VT_BSTR</code> ），其中每一個字串含有一個變數的名稱。如果 SQL 陳述式沒有變數，則結果是空白（ <code>variant</code> 類型 <code>VT_EMPTY</code> ）。在呼叫此函數之前，您必須適當地起始設定 <code>VARIANT</code> 。Visual Basic 會自動處理這件工作。Visual C++ 程式設計師必須呼叫 <code>VariantInit()</code> 。

**註:** 由於 Microsoft Excel 7.0 和 Microsoft Access 7.0 中的設計瑕疵（也可能發生在其他使用 Visual Basic for Applications 的 32 位元 Microsoft 產品中），從 QMF for Windows 所接收的 `Variant` 變數的字串資料可能無法從 Unicode（使用於 OLE）轉換成 ANSI（使用於 VBA）。如果發生這種情形，則只顯示字串的第一個字元。為了解決這個問題，請在呼叫 QMF for Windows 函數之前，先將變數設定成空字串。

### 回覆值

成功則為「零」，失敗則為「非零」。如果 SQL 陳述式沒有變數，則回覆值是 RS\_ERROR\_NO\_DATA (-1)。如果回覆值不是零，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

## GetVariablesEx()

short GetVariablesEx(long *QueryID*, short *Index*)

### 說明

此函數傳回 *Index* 參數所參照之查詢的 SQL 文字中的變數名稱。呼叫此函數之後，即可查詢 *Value* 內容來取得回覆值。在使用 Open() 或 Execute() 執行查詢之前，您必須呼叫 SetVariable() 來指定此變數的值。

### 參數

名稱	說明
QueryID	從 InitializeQuery() 傳回的查詢 ID。
Variables	內部變數清單中的索引。請查詢 <i>Value</i> 內容來取得所傳入之索引的相對應字串。如果 SQL 陳述式沒有變數，函數傳回 RS_ERROR_NO_DATA。

### 回覆值

成功則為「零」，失敗則為「非零」。如果 SQL 陳述式沒有變數，則回覆值是 RS\_ERROR\_NO\_DATA (-1)。如果回覆值「非零」，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

## InitializeProc()

long InitializeProc(short *SourceType*, BSTR *Source*)

### 說明

此函數設定您在程序中要使用的文字。您可以將文字當做參數傳遞到此函數、從文字檔中讀取、或從現存的程序中取得。

### 參數

名稱	說明
SourceType	指定程序文字的來源。

值	意義
---	----

0 (RSS_STRING)	文字包含在 <i>Source</i> 參數中。
2 (RSS_FILE)	文字包含在 <i>Source</i> 參數所指定的文字檔中。
3 (RSS_QMFPROC)	文字包含在 <i>Source</i> 參數所指定之擁有者和名稱的 程序中。

<i>Source</i>	字串，含有文字、程序的擁有者和名稱 (Owner.Name)、或含程序文字的檔案名稱。
---------------	---

### 回覆值

如果成功，傳回程序的 ID (ProcID)。如果失敗，傳回 -1。您在所有需要 *ProcID* 參數的介面呼叫中必須使用此值。

## InitializeQuery()

```
long InitializeQuery(short SourceType, BSTR Source)
```

### 說明

此函數設定您在查詢中要使用的文字。您可以將 SQL 文字當做參數傳遞到此函數、從文字檔中讀取、或從現存的查詢中取得。完成查詢時，請呼叫 Close()。

### 參數

名稱	說明
<i>SourceType</i>	指定 SQL 陳述式文字的來源。

值	意義
0 (RSS_STRING)	SQL 文字包含在 <i>Source</i> 參數中。
1 (RSS_QMFQUERY)	SQL 文字包含在 <i>Source</i> 參數所指定之擁有者和名稱的查詢中。
2 (RSS_FILE)	SQL 文字包含在 <i>Source</i> 參數所指定的文字檔中。

### 回覆值

如果成功，傳回查詢的 ID。如果失敗，傳回 -1。您在所有需要 *Query* 參數的介面呼叫中必須使用此值。

## InitializeServer()

short InitializeServer(BSTR *ServerName*, BSTR *UserID*, BSTR *Password*, BOOL *ForceDialog*, [VARIANT *Account*], [VARIANT *SuppressDialog*])

### 說明

此函數起始設定資料庫伺服器的連線。在呼叫 QMF for Windows 的其他函數之前，您必須先呼叫此函數。此函數可以呼叫多次。不過，如果您呼叫此函數，但未呼叫 Commit() 或 Rollback() 來結束，則會造成隱含的回復。

### 參數

名稱	說明
ServerName	字串，含有您要使用的資料庫伺服器名稱。此名稱必須符合 QMF for Windows 伺服器定義檔中定義的其中一個名稱。請呼叫 GetServerList() 來擷取有效的伺服器清單。
UserID	字串，含有您要使用的使用者 ID。如果 UserID 是 NULL 或空字串，QMF for Windows 會嘗試使用最近查詢中的使用者 ID。否則，QMF for Windows 顯示「使用者資訊」對話框來取得使用者 ID 和密碼。
Password	字串，含有指定之使用者 ID 的密碼。如果 Password 是 NULL 或空字串，QMF for Windows 會嘗試使用記得的密碼（需要 Windows for Workgroups）。如果沒有密碼可用，QMF for Windows 顯示「使用者資訊」對話框來取得密碼。
ForceDialog	「非零」指示 QMF for Windows 顯示「使用者資訊」對話框，不論是否已指定使用者 ID 和密碼都一樣。這讓使用者有機會變更資訊。「零」指示 QMF for Windows 只在必要時才顯示「使用者資訊」對話框。
Account	可選用的字串，指定在連接時要傳遞到伺服器的帳戶資訊。伺服器可能在工作帳戶系統中使用此資訊。
SuppressDialog	「非零」指示 QMF for Windows 不顯示「使用者資訊」對話框，即使尚未指定使用者 ID 和密碼也一樣。當執行的環境中沒有使用者可以回應「使用者資訊」對話框時，例如在 web 伺服器上，此選項非常有用。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。



## 相關的主題

SetParent()

## InitializeStaticQuery()

long InitializeStaticQuery(BSTR *CollectionName*, BSTR *PackageName*, BSTR *ConsistencyToken*, short *SectionNumber*)

### 說明

此函數指定您要當做靜態查詢來執行的套件區段。

### 參數

名稱	說明
CollectionName	先前連結的集成名稱。
PackageName	先前連結的套件名稱。
ConsistencyToken	上述指名的集成和套件所使用的記號。
SectionNumber	欲執行之集成和套件內的陳述式區段號碼。

### 回覆值

如果成功，傳回查詢的 ID。如果失敗，傳回 -1。您在所有需要 *QueryID* 參數的介面呼叫中必須使用此值。

## IsStatic()

BOOL IsStatic(long *QueryID*)

### 說明

此函數判斷指定的查詢 ID 是參照靜態查詢或動態查詢。

### 參數

名稱	說明
QueryID	從 InitializeQuery() 或 InitializeStaticQuery() 傳回的查詢 ID。

### 回覆值

如果成功且 *QueryID* 參照一個靜態查詢，傳回「非零」，否則傳回「零」。

## Open()

short Open(long *QueryID*, long *RowLimit*, BOOL *FetchAllRows*)

## 說明

使用此函數可以透過在查詢的資料庫中開啓一個游標，執行一個使用 `SELECT` 動詞的查詢。請使用 `FetchNextRow()` 或 `FetchNextRows()` 來擷取查詢的資料，完成時請呼叫 `Close()`。如果 QMF for Windows 被配置成自動提取所有列（請參閱 `GetResourceLimit()` 說明中的 `RSR_FETCHALLROWS`）或 `FetchAllRows` 參數不是零，則在此呼叫傳回之前，QMF for Windows 會將結果集的所有列提取到內部緩衝區。

**註：** 此函數的名稱與 Microsoft Access 2.0 的關鍵字 `Open` 發生衝突。如果您使用 MS Access 2.0，請在函數名稱兩旁加上方括弧 [ ]。

**註：** 此函數僅適用於含有 SQL Verb `SELECT` 的陳述式。對於含有其他動詞的陳述式，例如 `SET`，請改成呼叫 `Execute()`。如果要判斷查詢所用的動詞，請呼叫 `GetQueryVerb()`。

## 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
<i>RowLimit</i>	數字，表示從資料庫中擷取的最大列數。「零」表示除了 QMF for Windows Administrator 程式所建立的列限制以外，不強制施行任何限制。
<i>FetchAllRows</i>	Boolean 值，指出結果集的所有列是否自動提取到 QMF for Windows 內部緩衝區。如果為「非零」，則自動提取所有列，關閉游標，並且釋放資料庫供其他使用者使用。這和呼叫 <code>CompleteQuery()</code> 一樣。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## Prepare()

`short Prepare(long QueryID)`

## 說明

此函數準備 *QueryID* 所指定的查詢。陳述式由資料庫伺服器來查驗，檢查物件是否存在、所需的權限等等。如果查詢是 `SELECT` 陳述式，則在完成 `Prepare()` 之後，即可取得陳述式所傳回之直欄的相關資訊。

## 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## 相關的主題

`Execute()`

`Open()`

## PrintReport()

`short PrintReport(long QueryID, short SourceType, BSTR Source, BSTR OutputFileName, short PageLength, short PageWidth, BOOL IncludeDateTime, BOOL IncludePageNumbers, [VARIANT Format], [VARIANT UseFormPageSetup])`

## 說明

`PrintReport()` 是 `ExportReport()` 的同義字。

## ReinitializeServer()

`short ReinitializeServer()`

## 說明

此函數重新起始設定資料庫伺服器的連線。通常只有在其中一個 QMF for Windows API 函數傳回錯誤時，您才需要呼叫此函數。呼叫此函數會造成隱含的回復，關閉任何開啓的游標，並且使所有未完成的查詢 ID 無效。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## Rollback()

`short Rollback()`

## 說明

此函數取消現行工作單元中的任何變更、結束現行工作單元、關閉開啓的游標、使全部未完成的查詢 ID 無效。

**註：**此函數的名稱與 Microsoft Access 2.0 的關鍵字 Rollback 發生衝突。如果您使用 MS Access 2.0，請在函數名稱兩旁加上方括弧 [ ]。

**註：**回復僅影響 Open() 或 Execute() 所執行的 SQL 變更。回復不影響其他 QMF for Windows API 函數所做的變更，例如，FastSaveData()、SaveData() 或 DeleteQMFObject()。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。

## 相關的主題

Commit()

## RunProc()

short RunProc(long *ProcID*)

## 說明

此函數執行指定的程序。程序會執行到完成或直到發生錯誤為止。您無法透過此程式設計介面來存取程序的任何結果（例如，查詢所傳回的資料）。不過，在程序執行之後，即可取得任何匯出的檔案或儲存的資料。

## 參數

名稱	說明
<i>ProcID</i>	從 InitializeProc() 傳回的程序 ID。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString()、GetLastErrorType()、GetLastSQLCode()、GetLastSQLError() 或 GetLastSQLState() 來取得其他的錯誤資訊。

## SaveData()

short SaveData(long *QueryID*, long *FirstRow*, long *FirstCol*, long *LastRow*, long *LastCol*, BOOL *Replace*, BSTR *TableName*, BSTR *TableSpaceName*, BSTR

*ServerName*, BSTR *UserID*, BSTR *Password*, BOOL *ForceDialog*, [VARIANT *Account*], [VARIANT *Comment*], [VARIANT *CommitScope*]

## 說明

此函數將指定的欄列範圍儲存到指定之表格空間內的指定表格。如果您尚未擷取所有要儲存在表格中的資料列，則在呼叫此函數之前，您必須先呼叫 `CompleteQuery()`。如果您嘗試儲存尚未從資料庫擷取的列，儲存會失敗。如果表格已存在，則新資料的直欄數目和類型必須與現存的表格相同。

此函數是在單獨的工作單元中作業，不同於其他 API 函數，其結果會自動確定。呼叫 `Commit()` 或 `Rollback()` 不影響您使用此函數所做的變更。

## 參數

名稱	說明
<i>QueryID</i>	從 <code>InitializeQuery()</code> 傳回的查詢 ID。
<i>FirstRow</i>	您要儲存的第一列。結果集第一列的值是 0。
<i>FirstCol</i>	您要儲存的第一欄。結果集第一欄的值是 0。
<i>LastRow</i>	您要儲存的最後一列。如果要儲存全部列，請指定 -1。結果集最後一列的值等於總列數減一。
<i>LastCol</i>	您要儲存的最後一欄。如果要儲存全部直欄，請指定 -1。結果集最後一欄的值等於總直欄數減一。
<i>Replace</i>	「非零」表示指定的資料取代表格中任何現存的資料。「零」表示指定的資料附加到表格中任何現存的資料。
<i>TableName</i>	用來儲存資料的表格名稱。如果表格不存在，則會建立它。
<i>TableSpaceName</i>	有表格存在或在該處建立表格的表格空間名稱。如果 <i>TableSpaceName</i> 是 NULL 或空字串，則使用預設的表格空間。如果您已將 QMF for Windows 配置成一律使用預設的表格空間（請參閱 <code>GetResourceLimit()</code> 說明中的 <code>RSR_SDDIFFERENTTS</code> ），此參數會被忽略。
<i>ServerName</i>	用來儲存表格的資料庫伺服器名稱。如果 <i>ServerName</i> 是 NULL 或空字串，則使用您在 <code>InitializeServer()</code> 呼叫中指定的伺服器， <i>UserID</i> 、 <i>Password</i> 、 <i>ForceDialog</i> 及 <i>Account</i> 皆被忽略。
<i>UserID</i>	如果您在 <i>ServerName</i> 中指定不同的伺服器， <i>UserID</i> 為使用於該伺服器的使用者 ID。如果您不指定使用者 ID，QMF for Windows 使用對此伺服器指定的上一個使用者 ID，如果找不到，則顯示一個對話框。如果 <i>ServerName</i> 是 NULL 或空字串，此參數會被忽略。

Password	如果您在 <i>ServerName</i> 中指定不同的伺服器， <i>Password</i> 為使用於該伺服器的密碼。如果您不指定密碼，QMF for Windows 使用對此伺服器指定的上一個密碼，如果找不到，則顯示一個對話框。如果 <i>ServerName</i> 是 NULL 或空字串，此參數會被忽略。
ForceDialog	如果您在 <i>ServerName</i> 中指定不同的伺服器，則即使已指定或可以取得使用者 ID 和密碼，「非零」會強迫 QMF for Windows 顯示一個對話框來提示輸入登入資訊。「零」表示 QMF for Windows 只在必要時才顯示此對話框。如果 <i>ServerName</i> 是 NULL 或空字串，此參數會被忽略。
Account	可選用的。如果您在 <i>ServerName</i> 中指定不同的伺服器，則一個帳戶資訊字串會在連接時傳遞到伺服器。伺服器可能在工作帳戶系統中使用此資訊。如果 <i>ServerName</i> 是 NULL 或空字串，此參數會被忽略。
Comment	可選用的字串，可以為儲存資料的表格指定註解。
CommitScope	可選用的，指出在確定工作單元之前，要在表格中插入多少列。指定「零」表示在確定之前插入所有列。如果指定 10，表示每插入十列就確定一次。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。如果結果集是空白或未從資料庫中擷取任何列，則傳回「非零」，除非 `FirstRow=0` 且 `LastRow=-1`。如果是這樣會傳回「零」，並且建立一個空白表格。

## SaveQMFProc()

`short SaveQMFProc(BSTR OwnerAndName, BSTR Text, BSTR Comment, BOOL Replace, BOOL Share)`

### 說明

此函數在資料庫伺服器上儲存一個程序。

### 參數

名稱	說明
<i>OwnerAndName</i>	以句點分隔的字串，含有您要儲存之程序的擁有者和名稱。例如， <code>John.Proc2</code>
<i>Text</i>	字串，含有您要在程序中儲存的文字。

Comment	字串，含有您要連同程序一起儲存的任何註解。如果沒有註解，請以空字串或 NULL 來傳遞此參數。
Replace	指定「非零」，表示取代同名的現存程序。指定「零」，表示在發現同名的現存程序時異常中止作業。
Share	指定「非零」，表示與其他使用者共用程序。指定「零」，表示不與其他使用者共用程序。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## SaveQMFQuery()

short SaveQMFQuery(BSTR *OwnerAndName*, BSTR *Text*, BSTR *Comment*, BOOL *Replace*, BOOL *Share*)

### 說明

此函數在資料庫伺服器上儲存一個查詢。

### 參數

名稱	說明
<i>OwnerAndName</i>	以句點分隔的字串，含有您要儲存之查詢的擁有者和名稱。例如， <code>John.Query2</code>
<i>Text</i>	字串，含有您要在查詢中儲存的文字。
<i>Comment</i>	字串，含有您要連同查詢一起儲存的任何註解。如果沒有註解，請以空字串或 NULL 來傳遞此參數。
<i>Replace</i>	指定「非零」，表示取代同名的現存查詢。指定「零」，表示在發現同名的現存查詢時異常中止作業。
<i>Share</i>	指定「非零」，表示與其他使用者共用查詢。指定「零」，表示不與其他使用者共用查詢。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## SetBindOption()

short SetBindOption(BSTR *CollectionName*, BSTR *PackageName*, short *Option*, short *Value*)

### 說明

此函數會在呼叫 `EndBind()` 之前先設定集成和套件的選項。

### 參數

名稱	說明
<code>CollectionName</code>	您要設定選項之套件的集成 ID。
<code>PackageName</code>	您要設定選項的套件名稱。
<code>Option</code>	下列其中一個選項。
值	指定「非零」，表示取代同名的現存查詢。指定「零」，表示在發現同名的現存查詢時異常中止作業。
<code>Share</code>	下列其中一個指定選項的值。

各種選項的意義和值如下：

選項	意義	說明
<code>DDM_PKGRPLOPT(0x211C)</code>	旗號，指定是否以相同集成 ID 和名稱取代現存套件。	<code>DDM_PKGRPLALW (0x241F)</code> 是 <code>DDM_PKGRPLNA (0x2420)</code> 否
<code>DDM_STTDECDEL(0x2121)</code>	區隔字元，用來當做套件中 SQL 陳述式的小數點。	<code>DDM_DECDELPRD (0x243C)</code> 句號 <code>DDM_DECDELCMA (0x243D)</code> 逗號
<code>DDM_STTSTRDEL(0x2120)</code>	區隔字元，用來當做套件中 SQL 陳述式的字串值。	<code>DDM_STRDELAP (0x2426)</code> 單引號 <code>DDM_STRDEL DQ (0x2427)</code> 雙引號



DDM_PKGISOLVL(0x2124)	套件的隔離層次。	DDM_ISOLVLALL (0x2443) 全部 DDM_ISOLVLCHG (0x2441) 變更 DDM_ISOLVLCS (0x2442) 游標穩定性 DDM_ISOLVLNC (0x2445) 沒有確定 DDM_ISOLVLRR (0x2444) 可重複的讀取
DDM_PKGATHOPT(0x211E)	旗號，指定是否保留套件上的現存權限。	DDM_PKGATHKP (0x2425) 保留 DDM_PKGATHRVK (0x2424) 撤銷
DDM_QRYBLKCTL(0x2132)	在套件中提取查詢的資料列時所用的方法。	DDM_FIXROWPRC (0x2418) 一次一列 DDM_LMTBLKPRC (0x2417) 一次一個區塊
DDM_RDBRLSOPT(0x2129)	何時釋放套件執行所取得的資料庫資源。	DDM_RDBRLSCMM (0x2438) 確定 DDM_RDBRLSCNV (0x2439) 交談作業解除配置
DDM_STTDATEFMT(0x2122)	所擷取之日期值的格式。	DDM_ISODATEFMT (0x2429) ISO DDM_USADATEFMT (0x242A) US DDM_EURDATEFMT (0x242B) European DDM_JISDATEFMT (0x242C) Japanese Industrial Standard
DDM_STTTIMEFMT(0x2123)	所擷取之時間值的格式。	DDM_ISOTIMEFMT (0x242E) ISO DDM_USATIMEFMT (0x242F) US DDM_EURTIMEFMT (0x2430) European DDM_JISTIMEFMT (0x2431) Japanese Industrial Standard

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()` 或 `GetLastErrorType()` 來取得其他的錯誤資訊。

## SetBindOwner()

`short SetBindOwner(BSTR CollectionName, BSTR PackageName, BSTR OwnerID)`

### 說明

此函數可讓您為所要連結的套件指定一個不同於原先使用者 ID 的擁有者。如果您的使用者 ID 不具有連結套件所需的權限，但指定的擁有者具有此權限，則需要此函數。

### 參數

名稱	說明
<code>CollectionName</code>	您要指定擁有者之套件的集成 ID。
<code>PackageName</code>	您要指定擁有者的套件名稱。
<code>Comment</code>	字串，含有您要連同查詢一起儲存的任何註解。如果沒有註解，請以空字串或 <code>NULL</code> 來傳遞此參數。
<code>OwnerID</code>	您要連結之套件的擁有者 ID。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## SetBusyWindowButton()

`void SetBusyWindowButton(BSTR Text)`

### 說明

此函數指定作用中視窗的「取消」按鈕上所顯示的文字。

### 參數

名稱	說明
<code>Text</code>	字串，指定作用中視窗的「取消」按鈕上所顯示的文字。預設值是「取消」。指定空字串會隱藏按鈕。不論指定什麼文字，按鈕一定會取消或關閉視窗。

## 回覆值

無。

## 相關的主題

SetBusyWindowMessage()

SetBusyWindowMode()

SetBusyWindowTitle()

ShowBusyWindow()

## SetBusyWindowMessage()

void SetBusyWindowMessage(BSTR *Message*)

### 說明

此函數指定作用中視窗的訊息區中所顯示的文字。

### 參數

名稱	說明
Message	字串，指定作用中視窗的訊息區上所顯示的文字。

## 回覆值

無。

## 相關的主題

SetBusyWindowButton()

SetBusyWindowMode()

SetBusyWindowTitle()

ShowBusyWindow()

## SetBusyWindowMode()

void SetBusyWindowMode(short *Mode*)

### 說明

此函數決定 QMF for Windows 是否要顯示作用中視窗。作用中視窗能夠提供回饋給使用者，並讓使用者取消擱置的資料庫動作。QMF for Windows 下次執行一個會顯示或隱藏作用中視窗的作業時，您所做的變更就會生效。

### 參數

名稱	說明
----	----

Mode	指定 QMF for Windows 何時顯示作用中視窗：
值	意義
0 (RSM_NEVER)	不顯示視窗。此為預設值。
1 (RSM_WHENBUSY)	當 QMF for Windows 忙著與資料庫通信時顯示視窗。QMF for Windows 會適時地自動顯示此視窗。
2 (RSM_CLIENTCONTROLLED)	視窗會在您呼叫 ShowBusyWindow(TRUE) 之後顯示，在呼叫 ShowBusyWindow(FALSE) 之後隱藏。從屬站決定何時顯示視窗。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

### 相關的主題

SetBusyWindowButton()  
SetBusyWindowMessage()  
SetBusyWindowTitle()  
SetParent()  
ShowBusyWindow()

## SetBusyWindowTitle()

```
void SetBusyWindowTitle(BSTR Title)
```

### 說明

此函數指定作用中視窗的標題欄所顯示的文字。

### 參數

名稱	說明
Title	字串，指定作用中視窗的標題欄所顯示的文字。

### 回覆值

無。

## 相關的主題

SetBusyWindowButton()  
SetBusyWindowMode()  
SetBusyWindowMessage()  
ShowBusyWindow()

## SetGlobalVariable()

short SetGlobalVariable(BSTR *Name*, BSTR *Value*)

### 說明

此函數指派一個值給指定的廣域變數。此值可用於查詢、套表及程序。

### 參數

名稱	說明
Name	字串，含有您要設定之變數的名稱。
值	字串，含有您要指派給所指定變數的值。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

## SetHostVariable()

short SetHostVariable(long *QueryID*, VARIANT *Index*, VARIANT *Value*)

### 說明

此函數指派一個值給查詢所參照的主變數。查詢必須是參照主變數的靜態查詢（由 QMF 查詢所儲存，或由 AddHostVariable() 所建立）。*Index* 可以指定主變數的數字索引或名稱。

### 參數

名稱	說明
QueryID	從 InitializeStaticQuery() 傳回的查詢 ID。
Index	數字（variant 類型 VT_I2），指定查詢中主變數的索引；或字串（variant 類型 VT_BSTR），指定主變數的名稱。
值	主變數的值。如果要指定空值，variant 的類型應該設為 VT_EMPTY。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()` 或 `GetLastErrorType()` 來取得其他的錯誤資訊。

## SetOption()

short `SetOption(short Mode, VARIANT Value)`

### 說明

此函數設定 QMF for Windows 中指定的選項值。有些選項要等到 QMF for Windows 重新啟動之後，變更才會生效。正常情形下，要等到您毀損 QMF for Windows API 物件的全部案例之後，才會重新啟動 QMF for Windows。

### 參數

名稱	說明
Option	指定要設定的選項：
值	意義
0 (RSO_SERVER_DEFINITION_FILE)	伺服器定義檔名稱。
1 (RSO_CPIC_DLL)	CPI-C 提供程式 DLL 檔案名稱。
2 (RSO_CPIC_TIMEOUT_WARNING)	CPI-C 警告逾時值（以秒計）。此限制不使用於 QMF for Windows API。
3 (RSO_CPIC_TIMEOUT_CANCEL)	CPI-C 取消逾時值（以秒計）。
4 (RSO_TCP_TIMEOUT_WARNING)	TCP 警告逾時值（以秒計）。此限制不使用於 QMF for Windows API。
5 (RSO_TCP_TIMEOUT_CANCEL)	TCP 取消逾時值（以秒計）。
6 (RSO_DISPLAY_NULLS_STRING)	用來顯示空值的字串。
7 (RSO_ENTER_NULLS_STRING)	用來輸入空值的字串。
8 (RSO_ENTER_DEFAULTS_STRING)	用來輸入預設值的字串。
9 (RSO_TRACE_FILE_1)	追蹤檔 1 名稱。
10 (RSO_TRACE_FILE_2)	追蹤檔 2 名稱。
11 (RSO_TCP_TRACE_LEVEL)	TCP 追蹤層次。
12 (RSO_CPIC_TRACE_LEVEL)	CPI-C 追蹤層次。
13 (RSO_DDM_TRACE_LEVEL)	DDM 追蹤層次。
名稱	說明
值	要設定的選項值。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()` 或 `GetLastErrorType()` 來取得其他的錯誤資訊。

### 相關的主題

`GetOption()`

## SetParent()

`short SetParent(long ParentWnd)`

### 說明

此函數設定對話框的親項視窗。通常當 QMF for Windows 顯示對話框時（在作用中視窗或「使用者資訊」對話框中），對話框會顯示在中央，並且依附於 QMF for Windows 主視窗。此函數可讓您強迫 QMF for Windows 對話框置中顯示，並且依附於您的從屬站應用程式視窗。

### 參數

名稱	說明
ParentWnd	新的親項視窗的 <code>HWND</code> 。指定 <code>NULL</code> 可以使用 QMF for Windows 主視窗當做親項視窗。

### 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()` 或 `GetLastErrorType()` 來取得其他的錯誤資訊。

### 相關的主題

`ShowBusyWindow()`

## SetProcVariable()

`short SetProcVariable(long ProcID, BSTR Name, BSTR Value)`

### 說明

此函數指派一個值給指定的變數。在執行程序之前，此值會取代變數。如果您的程序有一個以上的變數，則在呼叫 `RunProc()` 之前，您必須先呼叫此函數來設定變數值。

## 參數

名稱	說明
ProcID	從 InitializeProc() 傳回的程序 ID。
Name	字串，含有您要設定之變數的名稱。
值	字串，含有您要指派給指定變數的值。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

## SetVariable()

short SetVariable(long *QueryID*, BSTR *Name*, BSTR *Value*)

## 說明

此函數指派一個值給指定的變數。在執行 SQL 陳述式之前，此值會取代變數。如果您的 SQL 陳述式有一個以上的變數，則在呼叫 Open() 或 Execute() 之前，您必須先呼叫此函數來設定變數值。

## 參數

名稱	說明
QueryID	從 InitializeQuery() 傳回的查詢 ID。
Name	字串，含有您要設定之變數的名稱。
值	字串，含有您要指派給指定變數的值。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 GetLastErrorString() 或 GetLastErrorType() 來取得其他的錯誤資訊。

## ShowBusyWindow()

void ShowBusyWindow(BOOL *Show*)

## 說明

此函數指示 QMF for Windows 去顯示或隱藏作用中視窗。作用中視窗能夠提供回饋給使用者，並讓使用者取消擱置的資料庫動作。只有在以 RSM\_CLIENTCONTROLLED 模式來呼叫 SetBusyWindowMode() 時，此函數才有作用。如果您呼叫 SetParent() 來設定親項視窗，則作用中視窗會依附於指定的視窗。



## 參數

名稱	說明
Show	指定「非零」會顯示作用中視窗，指定零會隱藏作用中視窗。如果指定「非零」，則要在您呼叫 <code>ShowBusyWindow()</code> 之後，且 <code>Show</code> 設定為「零」時，作用中視窗才會顯示。

## 回覆值

無。

## StartBind()

short StartBind(BSTR *CollectionName*, BSTR *PackageName*, BSTR *ConsistencyToken*)

## 說明

此函數開始在資料庫中連結套件的程序。

## 參數

名稱	說明
<code>CollectionName</code>	所要指定之套件的集成 ID。
<code>PackageName</code>	所要指定的套件名稱。
<code>ConsistencyToken</code>	16 字元的字串，含有一個八位元組記號的十六進位表示法，用來確保資料庫所連結之套件和使用該套件的應用程式之間的一致性。執行套件中的一個區段時，您必須提供這個相同的值。

## 回覆值

成功則為「零」，失敗則為「非零」。如果回覆值為「非零」，您可以呼叫 `GetLastErrorString()`、`GetLastErrorType()`、`GetLastSQLCode()`、`GetLastSQLError()` 或 `GetLastSQLState()` 來取得其他的錯誤資訊。

## 相關的主題

`EndBind()`

`CancelBind()`



---

## 附錄. 注意事項

本資訊是針對 IBM 在美國所提供之產品與服務開發出來的。而在其他國家中，IBM 不見得有提供本書中所提的各項產品、服務、或功能。要知道在您所在之區是否可用到這些產品與服務時，請向當地的 IBM 服務代表查詢。本書在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，其他非 IBM 產品、程式、或服務在運作上的評價與驗證，其責任屬於使用者。

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 10594-1785  
U.S.A.

若要查詢有關二位元組 (DBCS) 資訊的特許權限事宜，請聯絡您國家的 IBM 智慧財產部門，或者用書面方式寄到：

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**下列段落若與該國之法律條款抵觸，即視為不適用：** IBM 僅以現狀提供本書，而不提供任何明示或默示之保證 (包括但不限於可售性或符合特定效用的保證)。若有些地區在某些交易上並不允許排除上述保證，則該排除無效。

本書中可能會有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。同時，IBM 得隨時改進並 (或) 變動本書中所提及的產品及 (或) 程式。

本書中任何對非 IBM Web 站的敘述僅供參考，以便利 貴客戶之使用。該 Web 站上的資料，並非本 IBM 產品所用資料的一部分，因使用該 Web 站造成之損害，由 貴客戶自行負責。

本程式之獲授權者若希望取得相關資料，以便使用下列資訊者可洽詢 IBM。其下列資訊指的是：(1) 獨立建立的程式與其他程式 (包括此程式) 之間更換資訊的方式 (2) 相互使用已交換之資訊方法若有任何問題請聯絡：

IBM Corporation  
J74/G4  
555 Bailey Avenue  
P.O. Box 49023  
San Jose, CA 95161-9023  
U.S.A.

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於雙方之「IBM 客戶合約」、「IBM 國際程式授權合約」或任何同等合約之條款，提供本書中所說的授權程式與其所有適用的授權資料。

此處包含的效能資料是在控制環境下決定的。因此，在其它作業環境下取得的結果可能大不相同。在開發層次的系統上可能有做過一些測量，但不保證這些測量在目前一般系統上有相同的結果。不僅如此，有些測量是透過推測來預估。實際結果可能不同。本文件的使用者應驗證其特定環境適用的資料。

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他公開管道取得。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、或任何對產品的其他主張是否完全無誤。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。

目前看到的所有 IBM 價格是 IBM 建議的現行零售價，若有變更，恕不另行通知。經銷商價格可能不同。

本資訊僅作規劃用途。在所描述的產品上市之前，此處的資訊可能隨時變更。

本資訊中的範例包含了用於日常商業活動的資料及報告。為了盡可能詳細，範例中涵蓋了個人、公司、品牌及產品的名稱。所有名稱純屬虛構，如有雷同純屬巧合。

著作權授權：

本資訊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。貴客戶得為開發、使用、銷售並散布運用樣本程式之作業平台的應用程式程式

介面所撰寫的應用程式之目的，免費複製、修改並散布這些樣本程式。此些範例並未在所有情況下完整測試。故 IBM 不保證或默示保證此些程式之可靠性、服務性或功能。

如果您檢視資訊軟體，可能不會出現照片和彩色圖例。

---

## 商標

下列專有名詞是 IBM Corporation 在美國或 (以及) 其他國家的商標：

ACF/VTAM	IBMLink
Advanced Peer-to-Peer Networking	IMS
AIX	Language Environment
AIX/6000	MVS/ESA
AS/400	MVS/XA
CICS	OfficeVision/VM
CICS/ESA	OS/2
CICS/MVS	OS/390
CICS/VSE	PL/I
COBOL/370	PROFS
DATABASE 2	QMF
DataJoiner	RACF
DB2	S/390
DB2 Universal Database	SQL/DS
Distributed Relational Database Architecture	Virtual Machine/Enterprise Systems Architecture
DRDA	Visual Basic
DXT	VM/XA
GDDM	VM/ESA
IBM	VSE/ESA
	VTAM

Java 或所有以 Java 為基礎的商標和標誌，以及 Solaris 是 Sun Microsystems, Inc. 在美國，其它國家或兩者的商標。

Lotus 和 1-2-3 是 Lotus Development Corporation 在美國，其它國家或兩者的商標。

Microsoft、Windows 及 Windows NT 是 Microsoft Corporation 的商標或註冊商標。

以兩個星號 (\*\*) 所表示的公司、產品及服務名稱，可能是第三者的商標或服務標示。



# 索引

索引順序以中文字，英文字，及特殊符號之次序排列。

## 〔三劃〕

工作檔案, 建立 45

工具列

自訂 5

移除按鈕 6

移動按鈕 6

新增按鈕 5

## 〔四劃〕

中斷

套表 31

支配 4

## 〔五劃〕

主要

套表 31

主電腦變數

使用於靜態查詢 49

## 〔六劃〕

列印

報告 36

程序 40

SQL 查詢 13

列印查詢結果 28, 48

列印預覽

提示查詢 23

列條件

使用 18

新增 19

between 18

containing 19

ending with 18

equal to 18

列條件 (繼續)

greater than 18

greater than or equal to 18

less than 18

less than or equal to 18

null 19

starting with 18

列條件運算子

Is 19

Is Not 19

列, 選取 25

合併條件

在提示查詢中建立 20

在提示查詢中檢視

提示查詢中的 SQL 20

在資料庫伺服器執行

SQL 查詢 7

在資料庫伺服器開啓

提示查詢 22

SQL 查詢 12

多重查詢

同時顯示 8

多重查詢文件 8

多個表格

在提示查詢中 20

字型

查詢顯示 8

結果顯示 26, 46

## 〔七劃〕

伺服器

設定 1

含邏輯的程序 37

## 〔八劃〕

注意事項 137

物件

清單 41

直欄

套表 31

直欄 (繼續)

新增至提示查詢 17

直欄, 重排 46

直欄, 重新排序 26

直欄, 選取 25

表格

匯出資料至 58

新增至提示查詢 16

表格編輯器 53

搜尋列 53

新增列 54

變更列 54

## 〔九劃〕

建立

靜態查詢 49

建立工作檔案 45

建立線性程序 37

查詢

建置複雜 16

查詢結果, 列印 28, 48

查詢結果, 格式化 27, 46

查詢結果, 排序 26, 45

查詢結果, 組合 27, 47

查詢結果, 彙總 27, 47

查詢結果, 預覽 28, 48

查詢結果, 儲存 28, 47

查詢結果, 儲存至檔案 48

查詢結果, 儲存到檔案 28

計算

套表 31

重排直欄 46

重新排序直欄 26

重新調整直欄和列的大小 25

頁面

套表 31

## 〔十劃〕

### 套表

- 中斷 31
- 主要 31
- 直欄 31
- 計算 31
- 頁面 31
- 條件 31
- 產生報告 31
- 最終 31
- 開啓儲存的檔案 34, 38
- 詳細資料 31
- 選項 31
- 儲存至資料庫伺服器 35, 38
- 儲存至檔案 34, 38
- HTML 31
- 格式化數值查詢結果 27, 46

## 〔十一劃〕

### 動作按鈕

- 提示查詢 15

### 執行

- 提示查詢 16
- 靜態查詢 51

### 執行物件 42

### 密碼

- 更正 3

### 將格式轉換成套表 27, 47

### 帳號字串 3

### 從清單中移除物件 43

### 排序查詢結果 26, 45

### 排序條件

- 使用 17
- 新增 18

### 條件

- 套表 31

### 清單

- 物件 41
- 開啓儲存的檔案 43

### 清單, 移除物件 43

### 清單, 新增物件 42

### 產生物件 42

### 產生查詢

- 建立 9

### 組合查詢結果 27, 47

### 連接資料庫 68

## 〔十二劃〕

### 最終

- 套表 31

### 報告

- 列印 36
- 使用套表來產生報告 31
- 匯出 36
- 預覽 32

### 提示查詢

- 另存新檔 21
- 在 ... 使用 SQL 20
- 使用多個表格 20
- 使用替代變數 21
- 建立 15
- 建立合併條件 20
- 動作按鈕 15
- 執行 16
- 開啓已儲存的檔案 21
- 新增直欄 17
- 新增表格 16
- 儲存到資料庫伺服器 22
- 檢視 SQL 20
- 轉換成 SQL 21

### 替代變數

- 在 SQL 查詢中 10
- 在提示查詢中使用 21
- 使用 ... 執行 SQL 查詢 10
- 使用於靜態查詢 49
- 取代為主電腦變數 49

### 登入 2

### 程序

- 列印 40

### 結果檢視畫面 7

### 開啓

- 已儲存的 SQL 檔 11
- 提示查詢檔案 21
- 資料庫上的程序 39
- 資料庫伺服器上的查詢 22
- 資料庫伺服器上儲存的套表 35

## 〔十三劃〕

### 傳送至 59

### 匯出

- 報告 36

### 匯出資料

- 至其它表格 58
- 至資料庫伺服器 58
- 至檔案 57

### 彙總查詢結果 27, 47

### 搜尋

- 表格編輯器 53

### 新的

- 產生查詢 9

### 新增

- 列條件 19
- 排序條件 18
- 提示查詢 15
- SQL 查詢 7

### 新增列

- 表格編輯器 54

### 新增物件至清單 42

### 詳細資料

- 套表 31

### 資料庫

- 安全 2

### 資料庫伺服器

- 匯出資料至 58

### 載入程式

- Excel 60

### 預覽

- 列印的查詢 13
- 列印的程序 40
- 報告 32
- 預覽查詢結果 28, 48

## 〔十五劃〕

### 範例應用程式 60

### 編輯物件 42

### 複雜查詢

- 建置 16

## 〔十六劃〕

### 選取直欄和列 25

### 選項

- 套表 31

### 靜態查詢

- 使用替代變數 49



靜態查詢 (繼續)

建立 49

執行 51

## 〔十七劃〕

儲存

套表至資料庫伺服器 35, 38

套表至檔案 34, 38

提示查詢成爲檔案 21

提示查詢到資料庫伺服器 22

SQL 查詢到資料庫伺服器 11

SQL 查詢到檔案 11

儲存查詢結果 28, 47

儲存查詢結果至檔案 48

儲存查詢結果到檔案 28

檔案

匯出資料至 57

檢視

結果 7

SQL 7

## 〔十八劃〕

鎖住呼叫 67

## 〔二十三劃〕

變更列

表格編輯器 54

變更密碼 3

顯示物件 42

## A

AddDecimalHostVariable() 69

AddHostVariable() 69

API 參考手冊 68

## B

between (列條件) 18

BindDecimalHostVariable() 70

BindHostVariable() 71

BindSection() 72

## C

CancelBind() 73

ChangePassword() 73

ClearList() 74

Close() 74

Commit() 75

CompleteQuery() 75

containing (列條件) 19

CopyToClipboard() 76

## D

DB2 套表 55

DeleteQMFObject() 76

## E

EndBind() 77

ending with 18

equal to (列條件) 18

Excel

載入程式 60

ExecuteEx() 78

ExecuteStoredProcedureEx() 80

ExecuteStoredProcedure() 79

Execute() 77

ExportForm() 83

ExportReport() 84

Export() 81

## F

FastSaveData() 85

FetchNextRowEx() 87

FetchNextRowsEx() 89

FetchNextRows() 87

FetchNextRow() 86

FlushQMFCache() 89

## G

GetColumnCount() 90

GetColumnDataValue() 90

GetColumnHeaderEx() 91

GetColumnHeader() 90

GetColumnHeadings() 92

GetColumnValueEx() 93

GetColumnValue() 92

GetDefaultServerName() 94

GetGlobalVariable() 94

GetHostVariableNames() 94

GetHostVariableTypes() 95

GetLastErrorString() 95

GetLastErrorType() 96

GetLastSQLCode() 97

GetLastSQLError() 98

GetLastSQLState() 99

GetOptionEx() 100

GetOption() 99

GetProcText() 101

GetProcVariables() 101

GetQMFOBJECTInfoEx() 104

GetQMFOBJECTInfo() 102

GetQMFOBJECTListEx() 106

GetQMFOBJECTList() 105

GetQMFPProcText() 107

GetQMFPQueryText() 107

GetQueryText() 108

GetQueryVerb() 108

GetResourceLimitEx() 112

GetResourceLimit() 109

GetRowCount() 113

GetServerListEx() 114

GetServerList() 113

GetStoredProcedureResultSets() 114

GetVariablesEx() 116

GetVariables() 115

greater than or equal to (列條件)  
18

greater than (列條件) 18

## H

HTML

套表 31

## I

InitializeProc() 116

InitializeQuery() 117

InitializeServer() 118

InitializeStaticQuery() 119

Internet 郵件 59

Is Not (列條件運算子) 19

IsStatic() 119

Is (列條件運算子) 19

## L

less than or equal to (列條件) 18  
less than (列條件) 18

## N

null (列條件) 19

## O

Open() 119

## P

Prepare() 120  
PrintReport() 121

## R

ReinitializeServer() 121  
REXX 程序 37  
Rollback() 121  
RunProc() 122

## S

SaveData() 122  
SaveQMFProc() 124  
SaveQMFQuery() 125  
SetBindOption() 126  
SetBindOwner() 128  
SetBusyWindowButton() 128  
SetBusyWindowMessage() 129  
SetBusyWindowMode() 129  
SetBusyWindowTitle() 130  
SetGlobalVariable() 131  
SetHostVariable() 131  
SetOption() 132  
SetParent() 133  
SetProcVariable() 133  
SetVariable() 134  
ShowBusyWindow() 134

### SQL

在提示查詢中使用 20

### SQL 查詢

列印 13

### SQL 查詢 (繼續)

列印預覽 13  
在資料庫伺服器執行 7  
開啓已儲存的檔案 11  
開啓新文件 7  
儲存至檔案 11  
儲存到資料庫伺服器 11

StartBind() 135

starting with (列條件) 18

# 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。

折疊線

台北市115南港區三重路十九之十一號四棟九樓

臺灣國際商業機器股份有限公司  
軟體國際部 啟

廣告回信
臺灣北區郵政管理局 登記
北台字第
號

免貼郵票

寄件人 姓名：  
地址：

寄

折疊線

讀者意見表





Part Number: CT6NSTC

檔案號碼:

程式號碼: 5675-DB2

5697-F42

5697-G24

5697-G22

5648-D35

5697-G23

Printed in Singapore

SC40-0531-00



(1P) P/N: CT6NSTC

