



如何擴充 dBASE 功能

廣州省圖書館

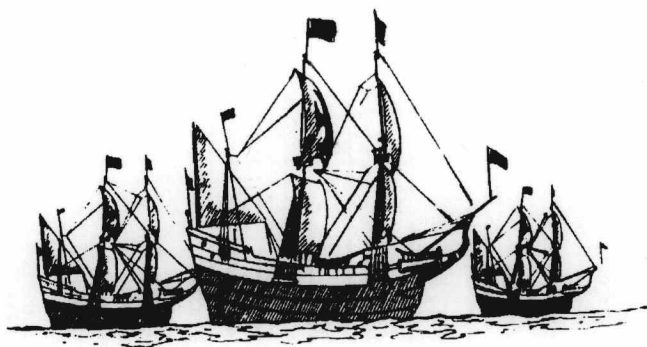


SoftTip 尖端電腦

華園電腦軟體研究開發部

NO.39210

如何擴充 dBASE PLUS 功能



瑩圃祝您：每天都有更大的成就！

[序]

應用軟體的優劣主要在於其功能 (Power) 及彈性 (flexibility), 而事實上後者又影響了前者, 因為富有彈性的軟體可以擴充及變化功能, 使適應性更加擴大。

本書為略具 dBASE 使用經驗的讀者而編, 是 dBASE PLUS 使用者的進階指引, 著眼於三個方向:

其一是如何發揮 dBASE III PLUS 的潛能, 加強運用其功能及彈性。

其二是如何將 dBASE III PLUS 與其他應用軟體 (如: Wordstar、Lotus 1-2-3、Sidekick 等) 結合, 以其特點擴充 dBASE。

其三、實際問題的指導, 例如除錯、財務計算、常式與副程式的撰寫、螢幕畫面的設計、繪圖等等。

善加運用軟體的功能與彈性, 將電腦的效率提高, 是電腦使用者、軟體人員的重要課題。本書的論點與範例完全符合實際應用狀況, 讀者精讀之後除能巧妙地運用本書所提供的工具之外, 更能藉本書指點的技巧, 創造更多更好的程式軟體。

瑩園 R&D

77年11月

目 錄

序言

目錄

第一章 處理郵遞清單的工具

改變大小寫	1-3
處理 "—"	1-13
轉換所有的欄位	1-15
轉換程式的極限	1-21
特殊的索引問題	1-22
輸入郵遞區號	1-24
名和姓	1-29
重覆	1-36
重覆資料的刪除	1-41
移開所有重覆的記錄	1-42
以複合的索引搜尋資料	1-51
結論:.....	1-57
列出記錄:.....	1-58

第二章 產生固定格式信函的工具

一般性的概念	2-3
資料庫檔案結構	2-3
ASCII 與 BINARY.....	2-4
DBASE 至 ASCII.....	2-5
ASCII 檔案型式	2-5
DELIMITED 格式中的欄位型態	2-11
次序和選擇	2-14

索引和排序	2-15
選擇記錄	2-16
選擇欄位	2-19
選擇記錄和欄位	2-21
資料轉移和其他問題	2-22
磁碟儲存的注意事項:	2-26
格式信件	2-28
WORDSTAR, WORDSTAR 2000 和 WORD 產生的信件	2-29
WORDSTAR	2-31
在磁碟上驗證	2-35
刪除空白行	2-36
WORDSTAR 2000 的格式信件	2-37
MICROSOFT WORD	2-43
刪除空白行	2-47
非標準的 ASCII檔案	2-49
WORDPERFECT	2-49
轉換檔案	2-50
Word Perfect中的格式信件	2-56
XY WRITE	2-59
結論	2-64
COPY TO	2-64
Alternate Files	2-65
文書處理程式 :	2-66

第三章 擴展 dBASE 系統

配置 dBASE III	3-2
改變 CONFIG.DB檔案	3-8
加入文書處理程式	3-11
有關 DOS的一些觀念	3-13
WordStar上的問題	3-16
執行文書處理程式	3-17

WordStar.....	3-20
WordStar 2000.....	3-21
WordPerfect.....	3-22
Word.....	3-23
XY Write.....	3-24
RUN 指令.....	3-24
在 dBASE中整合 DOS指令.....	3-26
檔案管理.....	3-30
抓取 DOS的訊息.....	3-31
RUN 指令的優點.....	3-46
MACROS (巨集指令).....	3-47
History.....	3-49
會學習的程式.....	3-53
總結.....	3-61

第四章 記憶體常駐程式

何謂記憶體常駐程式?.....	4-2
SIDEKICK.....	4-4
SIDEKICK 的編輯功能.....	4-4
輸入和輸出.....	4-5
SIDEKICK 的撥號器 (Dailer).....	4-9
SUPERKEY.....	4-14
巨集的發展.....	4-17
選擇性的處理.....	4-25
與巨集的整合.....	4-25
檔案及畫圖.....	4-26
顯示巨集 (DISPLAY MACROS).....	4-30
顯示巨集.....	4-32
螢幕強化功能.....	4-33
SIDEKICK, SUPERKEY AND DBASE.....	4-40
加上撥號功能.....	4-40
總結.....	4-43

第五章 螢幕顯示之設計

工作表設計	5-2
特殊按鍵的反應	5-3
INKEY()做些什麼?	5-4
螢幕上的移動	5-12
利用箭號鍵移動	5-15
以高亮度做選擇	5-20
次功能表的建立	5-34
"跳出"選擇項目	5-39
結論	5-51

第六章 除錯

除錯用的SET指令	6-2
抓取歷程	6-7
顯示狀態	6-11
記憶體變數	6-16
分析程式	6-19
程式結構	6-22
設計--分析程式	6-26
副程式	6-34
儲存結果	6-37
追蹤程式邏輯	6-45
捕捉錯誤	6-58
總結	6-70

第七章 財務計算

年金和貸款	7-4
估價	7-30
回歸分析	7-39
結語	7-45

第八章 試算表與繪圖

dBASE III PLUS (中) 之 (繪) 圖	8-2
顯示一個條形圖	8-3
列印圖形	8-11
傳送資料給 SPREAD SHEET 程式	8-22
轉換一個 Lotus 檔案	8-23
1-2-3 Release 2.0	8-24
dBASE III PLUS 轉換成 1-2-3 Release 2	8-33
ASCII 交換	8-34
SDF 格式資料	8-34
LOTUS PRN Files to dBASE III	8-35
改變格式	8-36
解剖一個 Lotus PRN 檔案	8-42
增加 dBASE III 資料到 SPREADSHEETS.	8-44
其他的 SPREADSHEET 程式	8-46
Framework	8-52
結論	8-55
繪圖	8-55
Spreadsheet	8-56

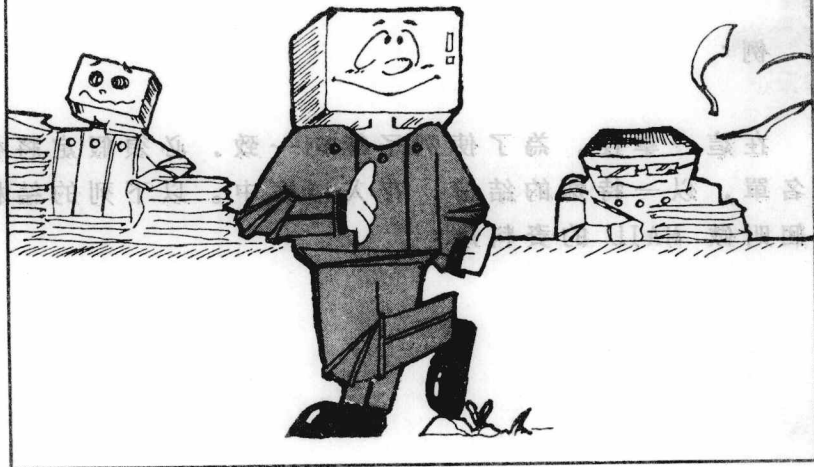
第九章 通訊

電傳通信	9-2
在電傳通信上通用的名詞	9-3
波特率	9-4
同位核對	9-7
非同步通信	9-8
為何要使用一個調變解調器程式	9-9
電子郵件	9-11
格式信件	9-26
股票市場	9-34
結論	9-46

第一章

Mailing Lists Power Tools 處理郵遞清單的工具

像我，用 dBASE III PLUS 的強大功能，
可以省掉在資料排序和校對的工作！



使用資料庫管理程式，最普通的原因，就是把它用來維護表列資料，例如：郵寄名單。從表面上來看，郵寄名單，並不複雜，許多人總是很肯定地說他們不需要使用像 dBASE III plus，功能這麼強的程式來操作郵寄名單。然而，在處理郵寄名單時所出現的問題，只能用 dBASE III plus 來解決。如果你選擇較不具威力的程式，你必須用人工來修正大寫資料的排列、重覆的輸入和龐大的校對工作。

如果你使用 dBASE III plus 來維護你的郵寄名單，你將會發現，這個程式中，含有許多功能和指令，可讓你很容易的解決實際上的問題。在這一章中，你將會學習到，如何使用這些功能和指令。記住，許多用來處理郵寄名單的技巧，同樣可以用在任何種類的資料庫檔案上。這些技巧都是用來維護已經存在的資料庫。處理郵寄名單是在使用 dBASE III plus 時，各種技巧中最普通的例子，因此在這一章中，我們以郵寄名單為例子，來討論這些技巧。

例：

在這一章裡，為了使例子能夠一致，必須假定將你的郵寄名單，以一特殊的結構，存入檔案中。以下列的結構建立一個叫做 MAIL 的資料庫：

Field	Field Name	Type	Width
1	COMPANY	Character	35
2	NAME	Character	30
3	ADDRESS	Character	30
4	CITY	Character	15
5	STATE	Character	2
6	ZIP	Character	5
7	LASTSENT	Date	8

你可以將書上範例中的資料或是你自己的資料，填入檔案中。跟著書上的步驟，在本章結束的時候，你將會發現，一序列的記錄，在資料庫中被使用者。如果，你使用你自己的記錄或是現存的郵寄名單，在你自己的螢幕上所出現的說明和影像，將和書上的有所不同。

改變大小寫

在郵寄名單的資料庫中，不一致的資料輸入或不一致的大小寫，是經常會遇到的問題。看看底下從 MAIL 中所列出的一個記錄：

COMPANY	MORRISON & FOERSTER
NAME	ROBERT MATTSON JR.
ADDRESS	200 PRINGLE AVE. #550
CITY	WLANUT CREEK
STATE	CA
ZIP	94596

注意，資料都以大寫輸入。如同上述的記錄，在許多大寫的字母中，夾雜著少數小寫字母的事情，是經常發生的。除了 STATE 這個欄位外，其餘的均可接受，因為在這個欄位中，字母必須是大寫的（註：美國各州名稱的縮寫，必須用大寫，如 California 的縮寫是 CA 而不是 Ca, 或 ca）。

如果你沒有使用 dBASE III plus，你或許要親自動手來修改這項錯誤。但是 dBASE III plus，有能力產生一個程式，來解這個問題。

將 STATE 這個欄位，轉換成大寫字母，對 dBASE III plus 而言，是一項非常簡單的工作，因為 dBASE III plus 提供了一個叫 UPPER() 的特殊函數。在 REPLACE 的指令中，加入 UPPER 這個函數，使得只需要一個指令，就可以改變全部的資料。

舉個例子，下列的指令，將會把資料庫中，每一筆記錄裡，STATE 欄位中的資料，轉換成大寫：

```
REPLACE ALL state WITH UPPER (state) <return>
```

把一個所有字母都是大寫的字，轉換成第一個字母是大寫，其他的字母均為小寫的字，是一件較為困難的工件。為了解決這個問題，你必須寫一個 dBASE III plus 的程式。程式中所使用最基本的工具，必須具備兩個函數，來轉換一個或一群字母。必備的函數如下：

UPPER() 將字母轉變成大寫

LOWER() 將字母轉變成小寫

用一個簡單的原則來設計這個程式：在每一個欄位中，跟在空白後的字母，均將之轉換為大寫。目前我們所要做的是寫一個程式，讓它來尋找欄位中的每一個空白，並將緊跟在空白後的每一個字母，轉換成大寫，再將其他的字母，轉換成小寫。

一開始，我們必須建立一個叫做 CONVCase 的程式。

鍵入 MODIFYCOMMAND convcase <return>

以下是程式的第一段，這些指令的作用，是將交談式功能關掉和將需要轉換的資料檔案打開，並將指標，指向此檔案中的第一筆記錄：

鍵入 SET TALK OFF
CLEAR
USE MAIL

下一段程式，是從一個處理檔案中，每一筆記錄的迴路圈 (LOOP)開始。

鍵入 DO WHILE NOT EOF()

接下來的四行程式，是用來建立在執行轉換時，所需用到的變數。第一個變數 LENFI（第一個欄位的長度），用來決定在這個欄位中，有多少個字母被轉換，函數 TRIM()，是用來省略在欄位後的空白，這個函數，可以省去許多程式分析這些空白的時間。

變數 LETTER，指出在這個欄位中，正被分析的字母的位置，每當 LETTER 中的值為 1 時，就表示從欄位中的第一個字母開始。變數 MAKEUPPER（轉換成大寫），被定成邏輯數值。 .T. 在 dBASE III plus 中代表 "真" 的值。（"假" 的值是 .F.）這個變數的值，用來讓程式決定是否要轉換，如果 MAKEUPPER 的值是 "真"，程式便將字母轉換成大寫，若是 "假"，則轉換成小寫字母。

CONVTEXT（轉換本文）是一文字變數，注意，以鍵入 " " 來將之定義成空的文字串。使用空白文字串的目的，在於建立一個文字變數，但不是馬上將任何文字放入。CONVTEXT 將會把欲轉換的公司名稱保留住。

```

鍵入  Lenf 1 = LEN (TRIM (company))
      Letter  = 1
      maheupper = .T.
      convtext = " "

```

下兩行程式用來顯示，目前正在處理的記錄編號和存在欄位 COMPANY 中的資料，這部分程式，對轉換文字的工作，並無任何特殊的功用，只是用來告訴程式的使用者，目前程

式正在做什麼。通常這是很好的主意，藉著顯示記錄編號，使用者可以知道，在檔案中有多少筆記錄，已被轉換過或尚未轉換。

```
鍵入  @ 9,10 SAY "RECORD #" + STR (RECNO(),5)
      @ 10,10 SAY company
```

下一段程式在處理真正的轉換工作，進行過程是從一個可以處理欄位中，每一個字母的迴路圈（LOOP）開始。在這迴路圈中，測試的工作，是由 `Letter <= lenf 1` 來執行，這表示迴路圈會不斷的執行，直到 `LETTER` 中的值，大於存在 `LENF 1` 中的值。

```
鍵入  DO WHILE letter <= lenf 1
```

在程式中，下一行是一個 `IF` 陳述（statement），用來測試 `LETTER` 所指的字母是否為空白，在 `SUBSTR ()` 函數中，使用 `LETTER` 來挑出欄位中的一個字母，只有一個字母會被 `SUBSTR()` 函數選出。

```
鍵入  IF SUBSTR (company,letter,1) # SPACE (1)
```

如果這個字母不是空白，那麼你就必須進行下一個測試。如果被 `LETTER` 指到的字母不是空白，則有兩種可能性，第一：這個字母是一個字的開頭，因此必須將它轉換成大寫。第二：這個字母是一個字中的一部分，因此必須為小寫。為了要知道答案是那一個，`dBASE III plus` 必須提供一些資訊來

告訴你，在這個字母之前的字母，是否為空白。這就是為什麼我們要使用變數 MAKEUPPER。你必須測試出 MAKEUPPER 的值是 "真" 或 "假"，因為若 MAKEUPPER 的值是 .T. ("真")，則每次通過迴路圈時，就會將字母轉換成大寫。

鍵入 IF makeupper

注意，因為 MAKEUPPER 是一個邏輯數值 (logical value)，而不是文字數值 (character value)，因此只要簡單的鍵入 IF makeupper，就可以告訴 dBASE III plus，你要測試變數 MAKEUPPER 是真是假。如此簡單的測試型態，就是使用邏輯變數最主要之便利之處。

下一行程式中，將真正的來執行轉換的工作，並將轉換後的字母，存入一個變數中。UPPER 函數的功能，是用來將字母轉換成大寫，整個指令告訴 dBASE III plus，將轉換過後的新字母，加入並存在 CONVTEXT 中。注意，CONVTEXT = CONVTEXT + new character 是循環執行的，如果你只使用 CONVTEXT = new character，先前在 CONVTEXT 中所儲存的值，便會被新的值所蓋掉。

鍵入 convtext = convtext + UPPER (SUBSTR
(company, letter, 1))

在你繼續寫下列程式之前，請記住，在字母被轉換成大寫之後，你必須將 MAKEUPPER 的值改變成 "假"，因為只能有一個字母被轉換成大寫。

鍵入 `makeupper = .F.`

下一段程式，用來處理 MAKEUPPER 的值為“假”的狀況，這狀況表示目前所測試到的字母，不是空白，而其前一個字母也不是空白，因此這字母，必須被轉換成小寫。

鍵入 `ELSE`

```
convtext = convtext + LOWER  
          (SUBSTR(company,letter,1))
```

現在請鍵入 `ENDIF`

注意，ENDIF 是 IF 陳述 (statement) 的一部分，緊跟在 IF 後面，在程式中並無任何功能上的意義。在 dBASE III plus 中，我們可以將註解或說明，放在 ELSE 及 ENDIF 之後。以一般的規則而言，在 ENDIF 後，加入說明，將會幫助你了解，你正結束掉那一個 IF 陳述。

鍵入 `ENDIF makeupper`

下一個部分的程式，是跟第一個測試（這個字母不是空白？）的 ELSE 條件之後。

鍵入 `ELSE`

這個部分的程式，是在處理，如果這個字母是空白的狀況。共有兩個步驟，第一：將一個空白加入 CONVTEXT 中。