

第二章 Hello World!

就讓我們從 Hello World! 這個簡單的程式開始學習吧！本章將帶領您進入 Visual LISP 與 VBA 的開發環境中，並為您示範如何寫出第一個簡單的程式範例。本章主要內容將包含：

●開發環境（即撰寫程式的主操作視窗）

●Hello World 程式



2-1 Visual LISP 的門面

首先，我們將先為您介紹撰寫 Visual LISP（以下簡稱 VLISP）的主操作視窗環境。請進入 AutoCAD 中，並點取「工具(T)」下拉式功能表下，「AutoLISP (S)」項目後的「Visual LISP 編輯器(V)」選項。或直接在指令提示區中鍵入：

指令: vlisp <Enter> 或

指令: vlide <Enter>

均將出現如下視窗：

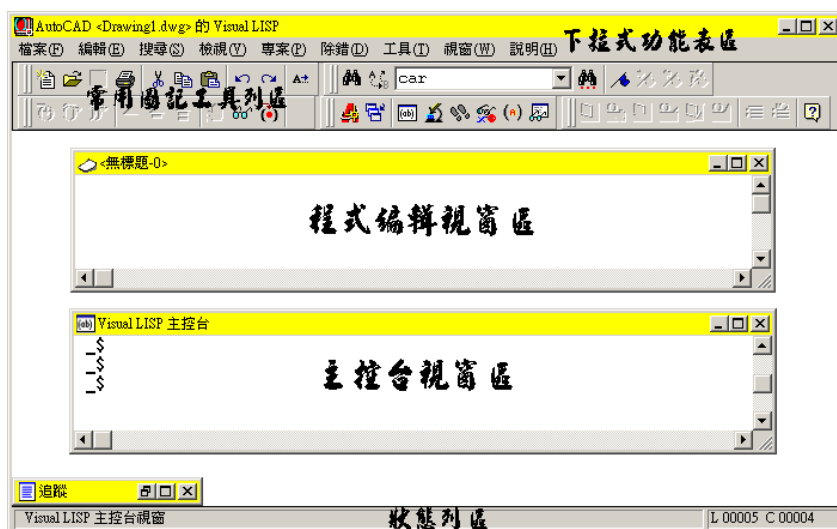


圖 2-1 LISP/VLISP/DCL 的程式撰寫視窗

說明：

1. 下拉式功能表區：由此發出 VLISP 編輯指令。
2. 常用圖示工具列區：可快速發出常用的 VLISP 編輯指令。
3. 程式編輯視窗區：於此編輯 LISP、VLISP、DCL 等程式原始碼檔案。
4. 主控台視窗區：在此可執行許多 VLISP 語法指令。
5. 狀態列區：顯示環境目前的狀態。

接下來，我們將以下示主題來為您詳述此視窗中重點元件的功能：

● 「檔案」下拉式功能表



圖 2-2 「檔案」下拉式功能表的內容

● 「編輯」下拉式功能表

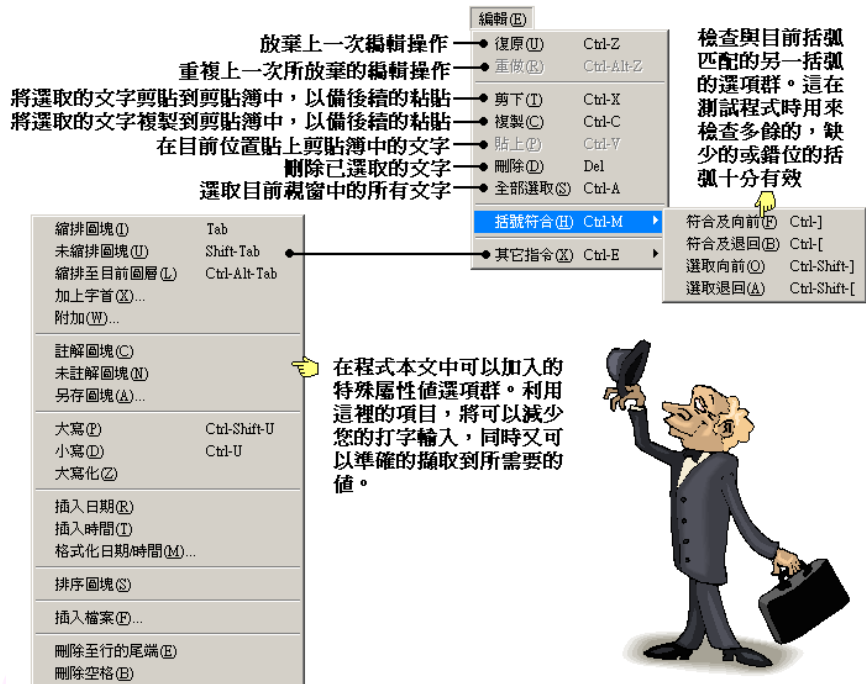


圖 2-3 「編輯」下拉式功能表的內容

● 「搜尋」下拉式功能表

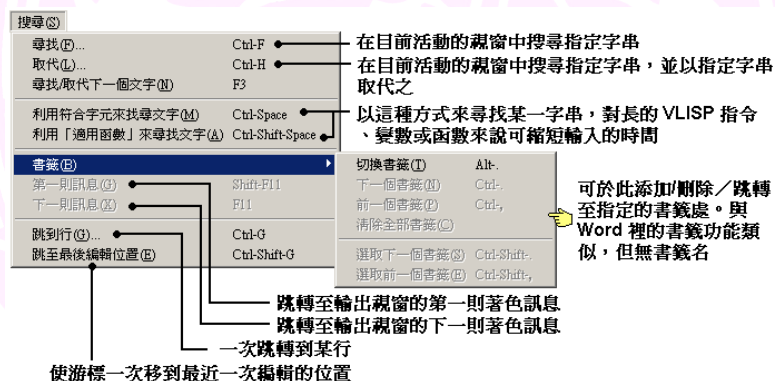


圖 2-4 「搜尋」下拉式功能表的內容

● 「檢視」下拉式功能表

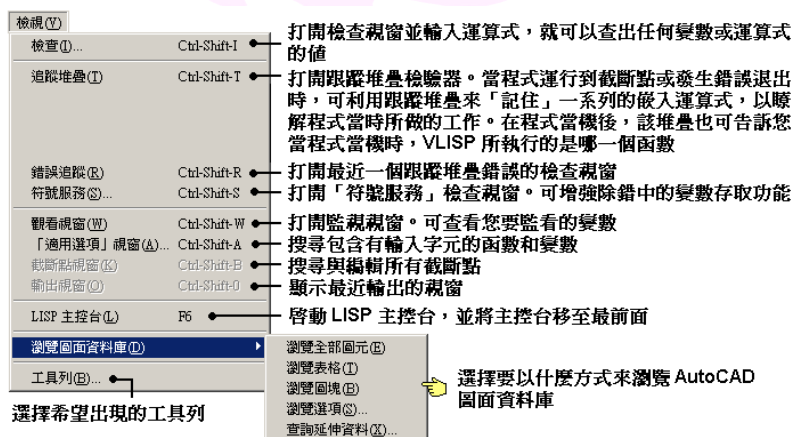


圖 2-5 「檢視」下拉式功能表的內容

●「專案」下拉式功能表

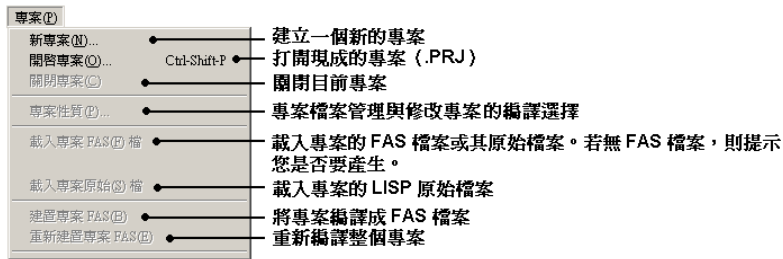


圖 2-6 「專案」下拉式功能表的內容

●「除錯」下拉式功能表

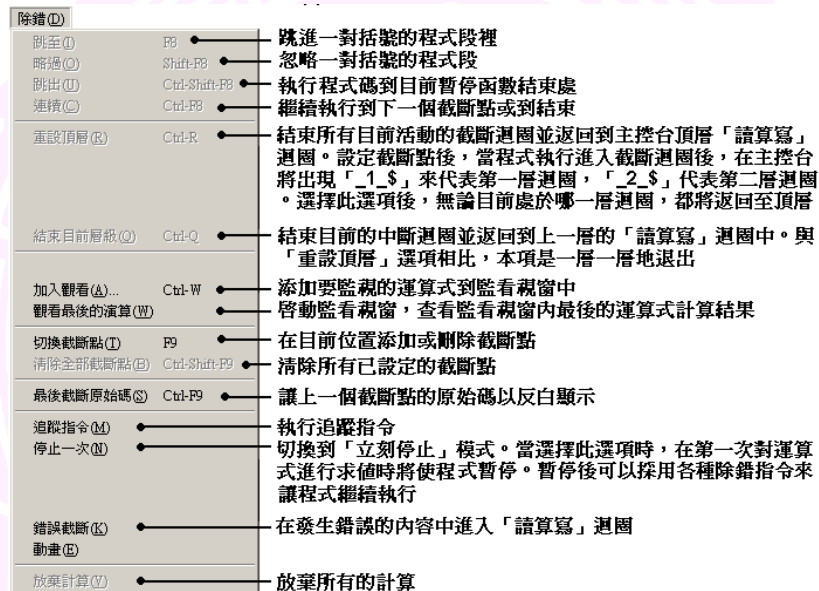


圖 2-7 「除錯」下拉式功能表的內容

●「工具」下拉式功能表

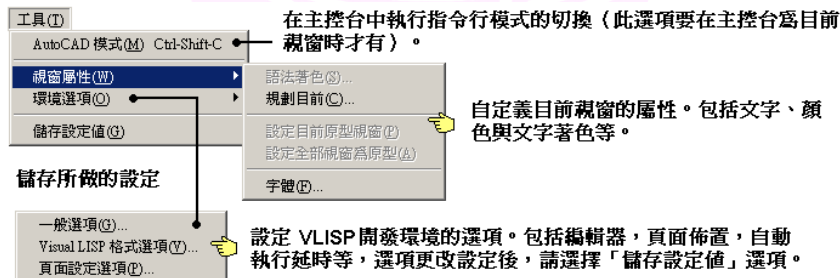
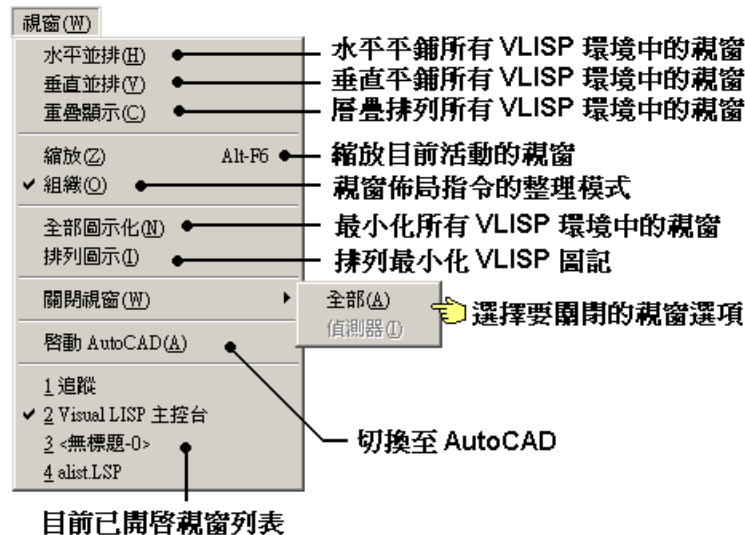


圖 2-8 「工具」下拉式功能表的內容

●「視窗」下拉式功能表



本貼圖內的文字

水平平鋪所有 VLISP 環境中的視窗

垂直平鋪所有 VLISP 環境中的視窗

層疊排列所有 VLISP 環境中的視窗

縮放目前活動的視窗

視窗佈局指令的整理模式

最小化所有 VLISP 環境中的視窗

排列最小化 VLISP 圖示

關閉部分視窗

切換至 AutoCAD

圖 2-9 「視窗」下拉式功能表的內容

●常用圖示工具列區

常用圖示工具列區將由「標準」、「搜尋」、「工具」、「除錯」、「檢視」等 5 部分所組成。雖然其內的功能都可以在下拉式功能表找到，但是多利用它將可以縮短我們點取指令的時間。

●快速鍵部分

當我們在鍵盤上編輯程式時，快速鍵也是一種快速發出指令的工具。例如，在 Windows 下我們常以按下 <Ctrl> + <C> 來進行複製，以按下 <Ctrl> + <V> 二鍵來執行貼上，以按下 <Ctrl> + <X> 執行剪切。下示列表就是一些在 Visual LISP 開發過程中常用到的快速鍵：

表 2-1 撰寫 VLISP 時常用的快速鍵列表

快速鍵	操作環境	作用
<Tab>	主控台視窗	取出剛輸入的前一條指令
<Shift> + <Tab>	主控台視窗	反轉取出指令的方向
<Esc>	主控台視窗	清除主控台目前指令行中的文字
<Shift> + <Esc>	主控台視窗	保留目前指令行文字並打開一新指令行
<Ctrl> + <[>	程式編輯視窗	到對應的左括弧
<Ctrl> + <]>	程式編輯視窗	到對應的右括弧
<Ctrl> + <E>	程式編輯視窗	叫出編輯需要的快顯功能表
<Ctrl> + <Shift> + <Space>	程式編輯視窗	使用「適用函數」來尋找指定文字
<Ctrl> + <Alt> + <E>	程式編輯視窗	載入編輯視窗內的文字
<Ctrl> + <Alt> + <F>	程式編輯視窗	格式化編輯視窗內程式碼
<F9>	程式編輯視窗	在目前位置添加/刪除截斷點
<F8>	除錯時	跳到下一括號運算式內
<Shift> + <F8>	除錯時	略過一括號運算式
<F6>	任意	顯示主控台
<Ctrl> + <W>	任意	將變數添加當監看視窗
<Ctrl> + <Shift> + <W>	任意	顯示監看視窗

●主控台視窗部分

除了 VLISP 主操作視窗上的下拉式功能表、工具列與快速鍵以外，在主控台視窗中輸入指令也一樣可以發出 VLISP 指令。

Visual LISP 的主控台視窗可說是一個類似 AutoCAD 指令提示區的工，不過，它可提供比指令提示區還要更多的功能。在 VLISP 程式的開發過程中，有效地使用主控台視窗將可提高您的開發效率。

在主控台視窗中，您可以直接於此計算輸入的運算式。如果您希望有一臨時性的變數，那麼您也可以於此直接將變數值納入目前程式本文中，當然，這個變數是暫時性的，當程式本文關閉就結束了。

此外，您還可儲存於主控台視窗中所輸入的本文與本文執行後的輸出結果。例如，您可以將程式原始碼分段拷貝在主控台中執行，查看它的結果。如此，在程式的偵錯效率上會比較好。以下是一些使用主控台視窗的注意事項：

1. 雖然 VLISP 主控台視窗和 AutoCAD 指令提示區所提供的功能相近，但完成同一功能的操作可能不同。例如，爲了顯示 AutoLISP 變數的目前值，在 VLISP 主控台視窗中，您只需在視窗中鍵入變數名稱並按下 <Enter> 鍵即可；但是在 AutoCAD 指令提示區中，您必須在變數名稱前加一個驚歎號（！）。
2. 在 AutoCAD 指令提示區中，按空白鍵將對運算式求值。而 VLISP 主控台視窗就不同了，它只有在按下 <Enter> 鍵後，才會處理在主控台提示處所輸入的運算式，如此，您這可以在主控台視窗中撰寫一行程式碼，而毋須擔心未寫完就被執行。
3. 在需要換行處按 <Ctrl> + <Enter> 即可在下一行繼續程式本文的運算式。
4. 在按 <Enter> 鍵之前可以輸入多個運算式。如：_\$(setq a 1)(setq b 2)。
5. 用 <Tab> 或 <Shift> + <Tab> 二鍵則可重覆前一條指令。

2-2 VBA 的門面

首先，我們將先爲您介紹撰寫 VBA 的主操作視窗環境。要撰寫 VBA 程式，您可以點取「工具(T)」下拉式功能表下，「巨集(M)」項目後，一個名爲「Visual Basic 編輯器(B)」的選項，或直接在指令提示區中鍵入：

指令: vbaide <Enter>

來進入。進入後的視窗畫面請參考圖 2-11。但是，因爲在 AutoCAD 裡，執行 VBA 與撰寫 VBA 的動作息息相關，所以，我們在此介紹的是以進入「VBA 管理員」的方式來操作。請進入 AutoCAD 中，並點取「工具(T)」下拉式功能表下，「巨集(M)」項目後的「VBA 管理員(V)...」選項，或直接在指令提示區中鍵入：

指令: vbaman <Enter>

均將出現如下視窗：



本貼圖內的文字

在目前圖形中加入一已選取的專案

新建一個專案

將已選取的專案另存一專案名稱

載入一個已存在的專案

卸下一個已選取的專案

打開巨集交談式視窗，可在此可執行巨集、建立巨集、刪除巨集..等動作。

進入 VBA 的程式撰寫視窗

圖 2-10 VBA 的管理員視窗

說明：現在，我們針對上圖裡相關的專有名稱，再作進一步的說明。

●「巨集」(Macro)

就是一個可以完成某特定的功能的指令集。在 VBA 中也稱之為「過程」，其語法型式為：

```
Public Sub HelloWorld()  
... (程式原始碼)  
... (程式原始碼)  
End Sub
```

在 VBA 中通常我們將巨集包裝在模組 (Module) 中。

●模組 (Module)

就是由一個或多個巨集所組成的程式段。模組的主要作用是用於建立

執行巨集的，程式中的原始碼，通常都存放在一個或幾個模組中。

●表單 (Form)

就是視窗資源。

●專案 (Project)

包含一個或多個的模組、巨集與表單，並將這些巨集與資源整合而構成一應用程式模組。

當於圖 2-10 中點取「Visual Basic 編輯器(V)」鈕時，將出現類似如下的程式撰寫視窗畫面：



圖 2-11 VBA 的程式撰寫視窗

說明：

1. 下拉式功能表區：由此發出 VBA 編輯指令。
2. 常用圖示工具列區：可快速發出常用的 VBA 編輯指令。
3. 程式本文區：於此編輯 VBA 程式原始碼檔案。
4. 專案總管區：列表顯示專案中的模組、表單等資源。
5. 屬性區：顯示於「專案總管區」內，所選物件的屬性。
6. 即時運算視窗區：類似 VLISP 的主控台，可於此直接計算運算式。
7. 區域變數視窗區：顯示區域變數的內容。
8. 運算式監看視窗區：顯示運算式內容。

注意：

1. 上述第 6、7、8 項的視窗區必須分別點取「檢適(V)」下拉式功能表下的「即時運算視窗(I)」、「區域變數視窗(S)」、「監看視窗(H)」等選項來叫出。
2. 「即時運算視窗區」將提供類似 VLISP 主控台的功能，但與主控台有不同之處。「即時運算視窗」可顯示來自程式碼中偵錯語法 `debug.print` 的資訊，或直接鍵入運算式後所產生的結果。在程式中斷時，您可使用「? + 變數名」可查看變數值。我們可以在程式未執行前先執行一些 VB 指令。如如 2-12 所示的實例：

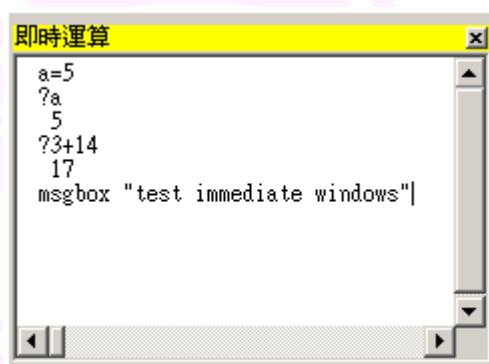


圖 2-12 「即時運算視窗區」的操作實例

當於圖 2-12 中執行 `msgbox "test immediate windows"` 時，將出現圖 2-13 所示的訊息框：

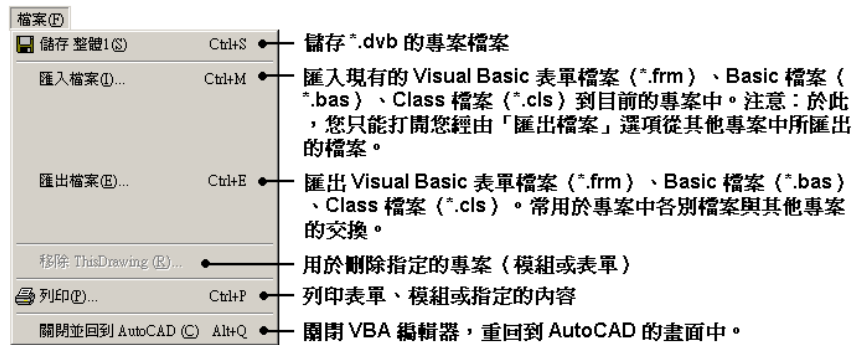


圖 2-13 執行 "test immediate windows" 運算式的結果

在中斷程式的情況下，「即時運算視窗區」、「區域變數視窗區」與「運算式監看視窗區」都是程式偵錯時的好工具，建議您多學習這類偵錯工具的使用。

接下來，照例的，我們將以下主題來為您詳述此視窗中重點元件的功能：

● 「檔案」下拉式功能表



本貼圖內的文字

儲存 *.dwb 的專案檔案

匯入現有的 Visual Basic 表單檔案 (*.frm)、Basic 檔案 (*.bas)、Class 檔案 (*.cls) 到目前的專案中。注意：於此，您只能打開您經由「匯出檔案」選項從其他專案中所匯出的檔案。

匯出 Visual Basic 表單檔案 (*.frm)、Basic 檔案 (*.bas)、Class 檔案 (*.cls)。常用於專案中各別檔案與其他專案的交換。

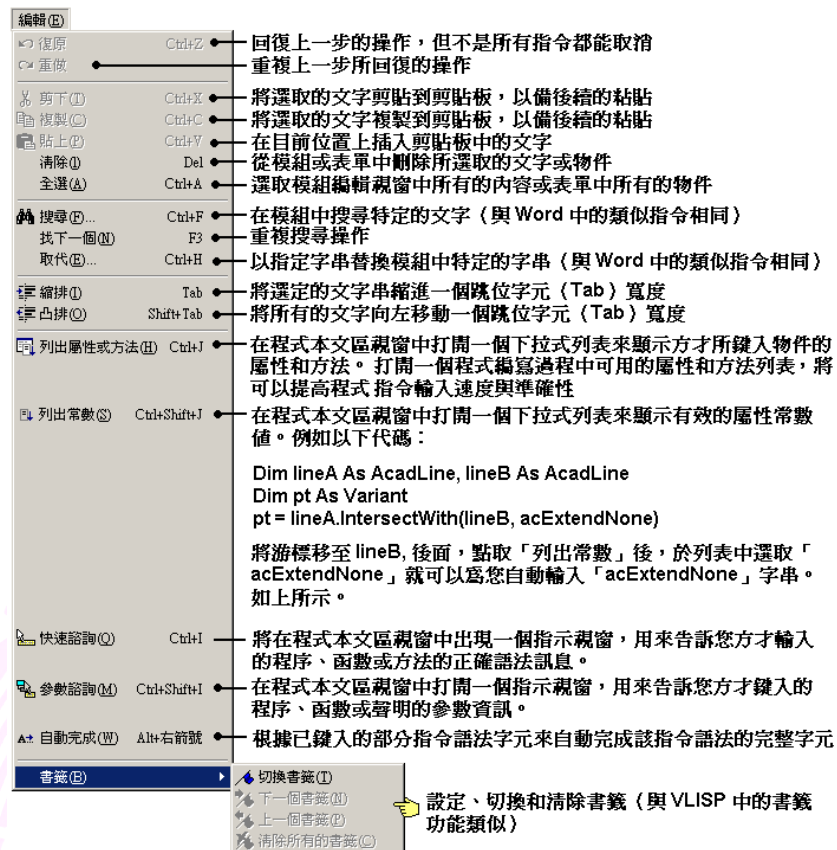
用於刪除指定的專案（模組或表單）

列印表單、模組或指定的內容

關閉 VBA 編輯器，重回到 AutoCAD 的畫面中。

圖 2-14 「檔案」下拉式功能表的內容

● 「編輯」下拉式功能表



本貼圖內的文字

回復上一步的操作，但不是所有指令都能取消
重複上一步所回復的操作

將選取的文字剪貼到剪貼板，以備後續的粘貼

將選取的文字複製到剪貼板，以備後續的粘貼

在目前位置上插入剪貼板中的文字

從模組或表單中刪除所選取的文字或物件

選取模組編輯視窗中所有的內容或表單中所有的物件

在模組中搜尋特定的文字（與 Word 中的類似指令相同）

重複搜尋操作

以指定字串替換模組中特定的字串（與 Word 中的類似指令相同）

將選定的文字串縮進一個跳位字元（Tab）寬度

將所有的文字向左移動一個跳位字元（Tab）寬度

在程式本文區視窗中打開一個下拉式列表來顯示方才所鍵入物件的屬性和方法。打開一個程式編寫過程中可用的屬性和方法列表，將可以提高程式指令輸入速度與準確性

在程式本文區視窗中打開一個下拉式列表來顯示有效的屬性常數值。例如以下代碼：

Dim lineA As AcadLine, lineB As AcadLine

Dim pt As Variant

pt = lineA.IntersectWith(lineB, acExtendNone)

將游標移至 lineB, 後面，點取「列出常數」後，於列表中選取「acExtendNone」就可以為您自動輸入「acExtendNone」字串。如上所示。

將在程式本文區視窗中出現一個指示視窗，用來告訴您方才輸入的程序、函數或方法的正確語法訊息。

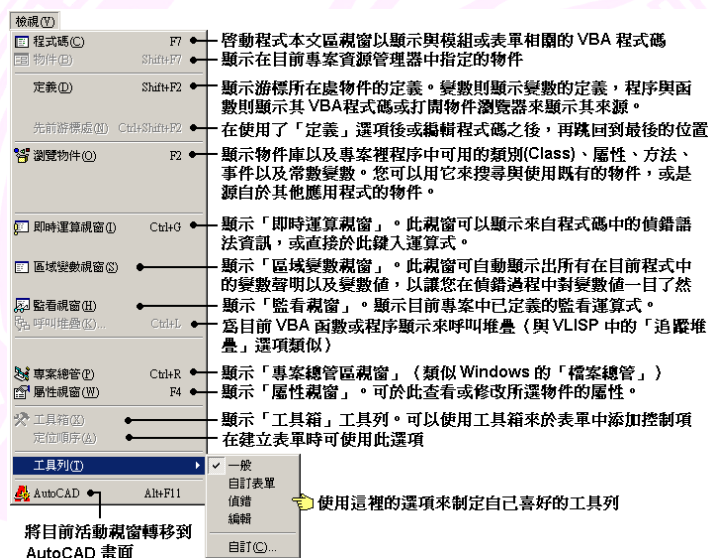
在程式本文區視窗中打開一個指示視窗，用來告訴您方才鍵入的程序、函數或宣告的參數資訊。

根據已鍵入的部分指令語法字元來自動完成該指令語法的完整字元

設定、切換和清除書籤（與 VLISP 中的書籤功能類似）

圖 2-15 「編輯」下拉式功能表的內容

● 「檢視」下拉式功能表



本貼圖內的文字

啟動程式本文區視窗以顯示與模組或表單相關的 VBA 程式碼

顯示在目前專案資源管理器中指定的物件

顯示游標所在處物件的定義。變數則顯示變數的定義，程序與函數則顯示其 VBA 程式碼或打開物件瀏覽器來顯示其來源。

在使用了「定義」選項後或編輯程式碼之後，再跳回到最後的位置

顯示物件庫以及專案裡程序中可用的類別(Class)、屬性、方法、事件以及常數變數。您可以用它來搜尋與使用既有的物件，或是源自於其他應用程式的物件。

顯示「即時運算視窗」。此視窗可以顯示來自程式碼中的偵錯語法資訊，或直接於此鍵入運算式。

顯示「區域變數視窗」。此視窗可自動顯示出所有在目前程式中的變數宣告以及

變數值，以讓您在偵錯過程中對變數值一目了然。

顯示「監看視窗」。顯示目前專案中已定義的監看運算式。

為目前 VBA 函數或程序顯示來呼叫堆疊（與 VLISP 中的「追蹤堆疊」選項類似）

顯示「專案總管區視窗」（類似 Windows 的「檔案總管」）

顯示「屬性視窗」。可於此查看或修改所選物件的屬性。

顯示「工具箱」工具列。可以使用工具箱來於表單中添加控制項

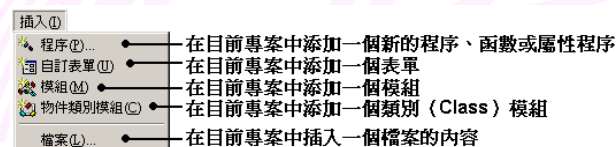
在建立表單時可使用此選項

使用這裡的選項來制定自己喜好的工具列

將目前活動視窗轉移到 AutoCAD 畫面

圖 2-16 「檢視」下拉式功能表的內容

●「插入」下拉式功能表



本貼圖內的文字

在目前專案中添加一個新的程序、函數或屬性程序

在目前專案中添加一個表單

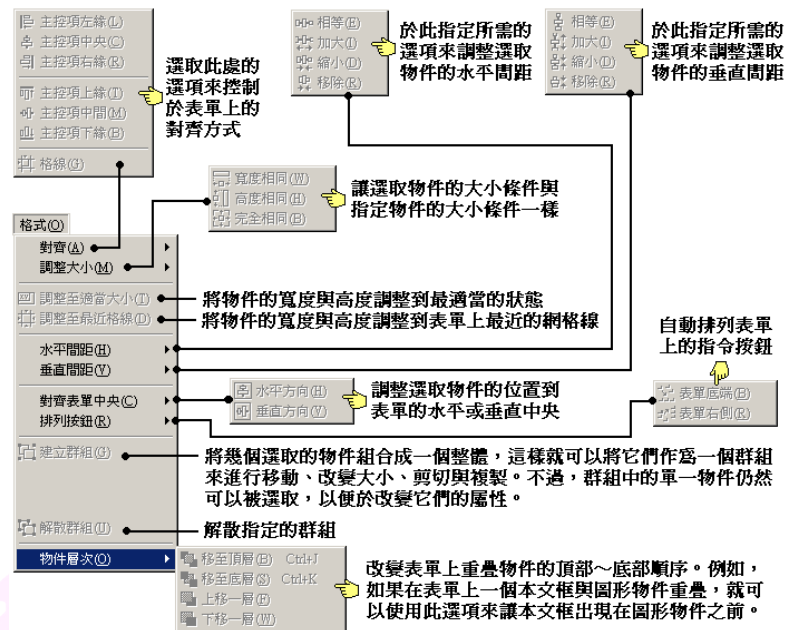
在目前專案中添加一個模組

在目前專案中添加一個類別（Class）模組

在目前專案中插入一個檔案的內容

圖 2-17 「插入」下拉式功能表的內容

●「格式」下拉式功能表



本貼圖內的文字

選取此處的選項來控制於表單上的對齊方式

讓選取物件的大小條件與指定物件的大小條件一樣

將物件的寬度與高度調整到最適當的狀態

將物件的寬度與高度調整到表單上最近的網格線

於此指定所需的選項來調整選取物件的水平間距

於此指定所需的選項來調整選取物件的垂直間距

調整選取物件的位置到表單的水平或垂直中央

自動排列表單上的指令按鈕

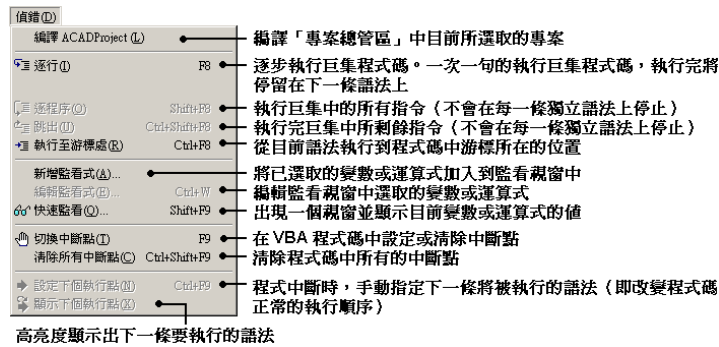
將幾個選取的物件組合成一個整體，這樣就可以將它們作為一個群組來進行移動、改變大小、剪切與複製。不過，群組中的單一物件仍然可以被選取，以便於改變它們的屬性。

解散指定的群組

改變表單上重疊物件的頂部～底部順序。例如，如果在表單上一個本文框與圖形物件重疊，就可以使用此選項來讓本文框出現在圖形物件之前。

圖 2-18 「格式」下拉式功能表的內容

● 「偵錯」下拉式功能表



本貼圖內的文字

編譯「專案總管區」中目前所選取的專案

逐步執行巨集程式碼。一次一句的執行巨集程式碼，執行完將停留在下一條語法

上執行巨集中的所有指令（不會在每一條獨立語法上停止）

執行完巨集中所剩餘指令（不會在每一條獨立語法上停止）

從目前語法執行到程式碼中游標所在的位置

將已選取的變數或運算式加入到監看視窗中

編輯監看視窗中選取的變數或運算式

出現一個視窗並顯示目前變數或運算式的值

在 VBA 程式碼中設定或清除中斷點

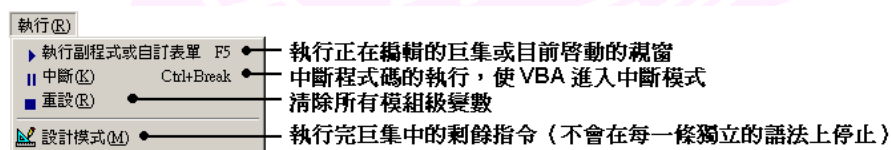
清除程式碼中所有的中斷點

程式中斷時，手動指定下一條將被執行的語法（即改變程式碼正常的執行順序）

高亮度顯示出下一條要執行的語法

圖 2-19 「偵錯」下拉式功能表的內容

● 「執行」下拉式功能表



本貼圖內的文字

執行正在編輯的巨集或目前啟動的視窗

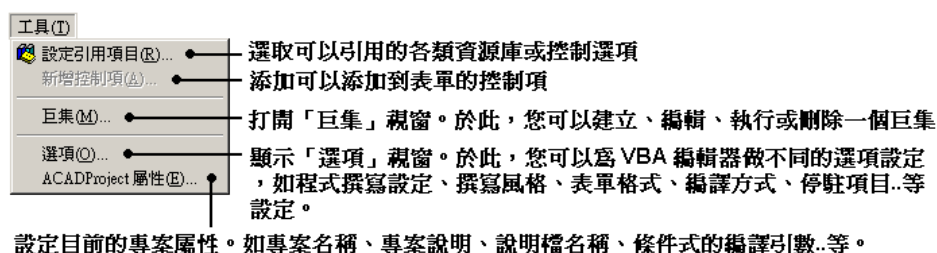
中斷程式碼的執行，使 VBA 進入中斷模式

清除所有模組級變數

執行完巨集中的剩餘指令（不會在每一條獨立的語法上停止）

圖 2-20 「執行」下拉式功能表的內容

●「工具」下拉式功能表



本貼圖內的文字

選取可以引用的各類資源庫或控制選項

添加可以添加到表單的控制項

打開「巨集」視窗。於此，您可以建立、編輯、執行或刪除一個巨集。

顯示「選項」視窗。於此，您可以為 VBA 編輯器做不同的選項設定，如程式撰寫設定、撰寫風格、表單格式、編譯方式、停駐項目..等設定。

設定目前的專案屬性。如專案名稱、專案說明、說明檔名稱、條件式的編譯引數..等。

圖 2-21 「工具」下拉式功能表的內容

●常用圖示工具列區

常用圖示工具列區將由「一般」、「編輯」、「自訂表單」與「偵錯」等 4 部分所組成。同時，您也可以選取「檢視(V)」下拉式功能表下，「工具列」項目後的「自訂(C)...」選項來自訂工具列。雖然圖示工具列內的功能都可以在下拉式功能表中找到，但是多利用它將可以縮短我們點取指令的時間。

●快速鍵部分

當我們在鍵盤上編輯程式時，快速鍵也是一種快速發出指令的工具。下列表就是一些在 VBA 開發過程中常用到的快速鍵：

表 2-2 撰寫 VBA 時常用的快速鍵列表

快速鍵	操作環境	作用
<Ctrl> + <J>	程式編輯視窗	打開一個下拉式列表，以顯示方才鍵入物件的屬性和方法
<Ctrl> + <Shift> + <J>	程式編輯視窗	打開一個列表以顯示有效的屬性常

		數值
<Ctrl> + <I>	程式編輯視窗	打開一個跳出式的說明視窗來顯示方才所輸入的程序、函數或方法的正確語法訊息
<Ctrl> + <Shift> + <I>	程式編輯視窗	在程式碼視窗中打開一個跳出式的說明視窗，以顯示方才鍵入的程序、函數或宣告可用的參數資訊
<F7>	任意	啟動程式碼視窗
<F2>	任意	打開物件瀏覽器視窗
<F8>	任意	逐一執行語法程式
<F5>	任意	執行副程式/自訂表單

2-3 Hello World! 程式初步(AutoLISP 與 Visual LISP)

在瞭解了 LISP/VLISP 與 VBA 的程式開發環境後，現在，我們就以一個簡單的範例來為您示範 LISP/VLISP 程式的語法撰寫。在這個範例裡，我們希望程式能以 (100,100,0) 座標點為圓心，50 為半徑，自動畫出一個圓，並在指令提示區中顯示 "Hello World!" 字樣。

2.3.1 AutoLISP/VLISP 程式的撰寫與執行部分

● AutoLISP 程式部分

請依下步驟來執行整個 AutoLISP 程式的撰寫與執行：

1. 請進入 AutoCAD 2000 中，並點取「工具(T)」下拉式功能表下，「AutoLISP (S)」項目後的「Visual LISP 編輯器(V)」選項。
2. 然後在「程式編輯視窗區」中輸入以下的程式本文（注意：下示每條程式前黑圓裡的數字，如 (1)，(2)...等，是為方便稍後的語法說明，請勿輸入）：

(1) ;;Hello World AutoLISP Program-----firstlisp.lsp

(2) ;;function:draw a circle and print "Hello World"

(3)

```
(4) (defun c:firstlisp(/ centerpt radius)
(5) (setvar "cmdecho" 0)
(6) (setq centerpt (list 100 100 0))
(7) (setq radius 50)
(8) (command "circle" centerpt radius)
(9) (prompt "Hello World!")
(10) (princ)
(11) )
```

分析：首先，程式的第 (1)，(2) 行是註解。意即以分號 “;” 開頭的就表示此行為註解。註解行一般是不執行的，它是程式設計師用來提醒或說明該程式的設計重點或設計內容的。第 (4) 行是指令的定義語法，後面括弧內，斜線後的變數名稱是表示該變數是暫時性的變數。第(5)行用來設定 “cmdecho” 系統變數為 0，“cmdecho” 系統變數將用來控制當 AutoLISP 的 (command) 函數執行時，AutoCAD 是否顯示提示與輸入。

第 (6)，(7) 行的語法是表示要將圓的圓心與半徑的值分別分派給變數 centerpt 與 radius。第 (8) 行則是呼叫 AutoCAD 的 circle 指令，並根據 centerpt 與 radius 的值來執行畫圓動作。第 (9) 行則表示要於指令提示區中印出 "Hello World!" 字樣。第 (10) 行是第 (9)行的動作執行指令。第 (11) 行則是第 (4) 行的對稱括號。

資訊 補充站

Command 函數介紹

由於 Command 函數很常用，所以在此先介紹。Command 是一個系統內部函數，它將使 AutoLISP 可以簡單而方便地呼叫所有的 AutoCAD 指令。Command 函數的參數用來表示所呼叫的 AutoCAD 指令與其副指令需要的資料；它將視所執行的 AutoCAD 指令以及其所需要的資料類型而定。例如，我們在 AutoCAD 中使用 LINE 指令，要從點 “0,0” 到 “100,100” 畫出一條直線，其指令流程是：

指令:line <Enter>

指定第一點: 0,0 <Enter>

指定下一點或 [復原(U)]: 100,100 <Enter>

指定下一點或 [復原(U)]: <Enter>

如果要在 LISP/VLISP 中使用 command 函數來執行一樣的動作，則為：

(command "line" "0,0" "100,100" "")

3. 執行時，請依下圖例，先載入，再於 AutoCAD 的指令行下輸入：
firstlisp。程式執行結果也如下圖例所示：

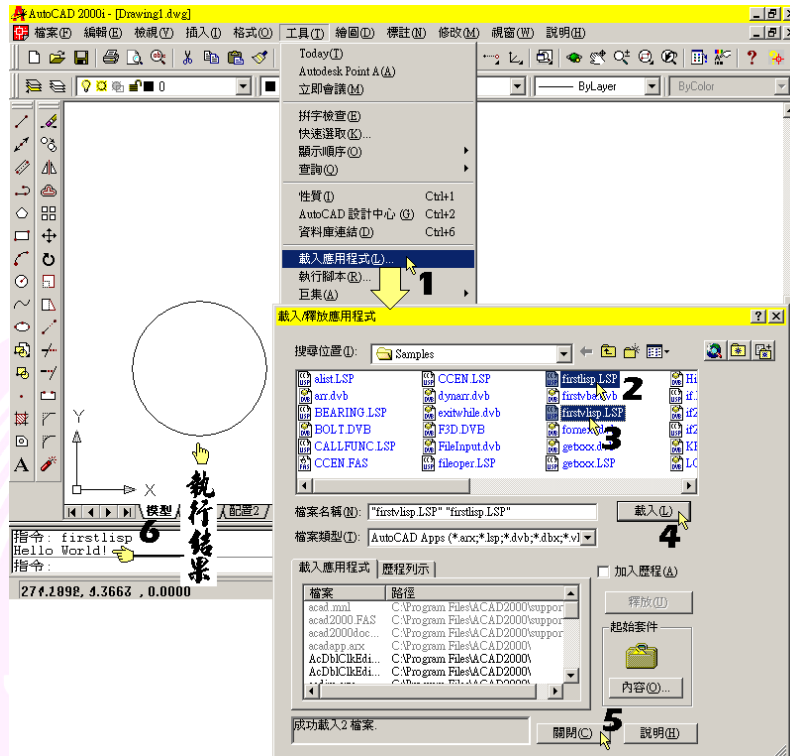


圖 2-22 執行 firstlisp.lsp 的結果

● VLISP 程式部分

請依下步驟來執行整個 VLISP 程式的撰寫與執行：

1. 請進入 AutoCAD 2000/2000i/2002 中，並點取「工具(T)」下拉式功能表下，「AutoLISP(S)」項目後的「Visual LISP 編輯器(V)」選項。
2. 然後在「程式編輯視窗區」中輸入以下的程式本文（注意：下示每條程式前括號裡的數字，如 (1)，(2)...等，是爲了方便稍後的語法說明，請勿輸入）：

- (1) ;;Hello World Visual LISP Program-----firstvlisp.lsp
- (2) ;;function:draw a circle and print "Hello World"
- (3)
- (4) (defun c:firstvlisp (/ centerpt radius)
- (5) (setvar "cmdecho" 0)


```
(6) (vl-load-com)
(7) (setq acadObject (vlax-get-acad-object))
(8) (setq acadDocument (vla-get-ActiveDocument acadObject))
(9) (setq mSpace (vla-get-ModelSpace acadDocument))
(10) (setq pt (list 100 100 0))
(11) (setq centerpt (vlax-3d-point pt))
(12) (setq radius 50)
(13) (vla-addcircle mSpace centerpt radius)
(14) (prompt "Hello World!")
(15) (princ)
(16) )
```

分析：VLISP 的語法是涵蓋 LISP 語法的。所以，我們僅分析其不同處。首先，第 (6) 行的 (vl-load-com) 函數是用來載入 VLISP 所提供的「延伸 AutoLISP 函數」。這是因為 VLISP 必須透過延伸的 AutoLISP 函數來履行對 ActiveX 的支援。第 (7)、(8)、(9) 行將設定一些 vla 系列的變數，其功用我們會在第十一章中詳述。第 (13) 行的函數將在模型空間以 centerpt 為圓心，radius 為半徑畫一個圓。這條指令看來要比 AutoLISP 透過 command 函數的畫圓語法更直接，而且執行起來也比較快！

3. 執行時，如圖 2-22，但於 AutoCAD 的指令行下輸入：firstvlisp。程式的執行結果也如圖 2-22。

2.3.2 Visual LISP 程式的編譯部分

我們都知道：AutoLISP 以前一直只有加密的程式（但是此程式也早已被破解，毫無保護功用），卻無編譯器（Compiler）來將原始程式編譯成一二進位的檔案。現在，這個夢想已經在 Visual LISP 裡實現了。

AutoCAD 的編譯功能可以將一屬文字檔的 Visual LISP 原始檔案 .LSP，編譯成一檔案延伸名為 .FAS 的二進位編譯檔。這樣的 .FAS 的編譯檔就僅能被執行，而不能修改其內容。

請依下述步驟來進行編譯 Visual LISP 原始檔案的操作：

1. 請點取「工具(T)」下拉式功能表下，「AutoLISP(S)」選擇項後的「Visual LISP 編輯器(V)」項目，並將欲編譯且執行已無問題的 Visual LISP 原始檔案載入。

2. 請點取編輯視窗裡，「專案(P)」下拉式功能表下的「新專案(N)...」選擇項。並依圖 2-23 所示操作。

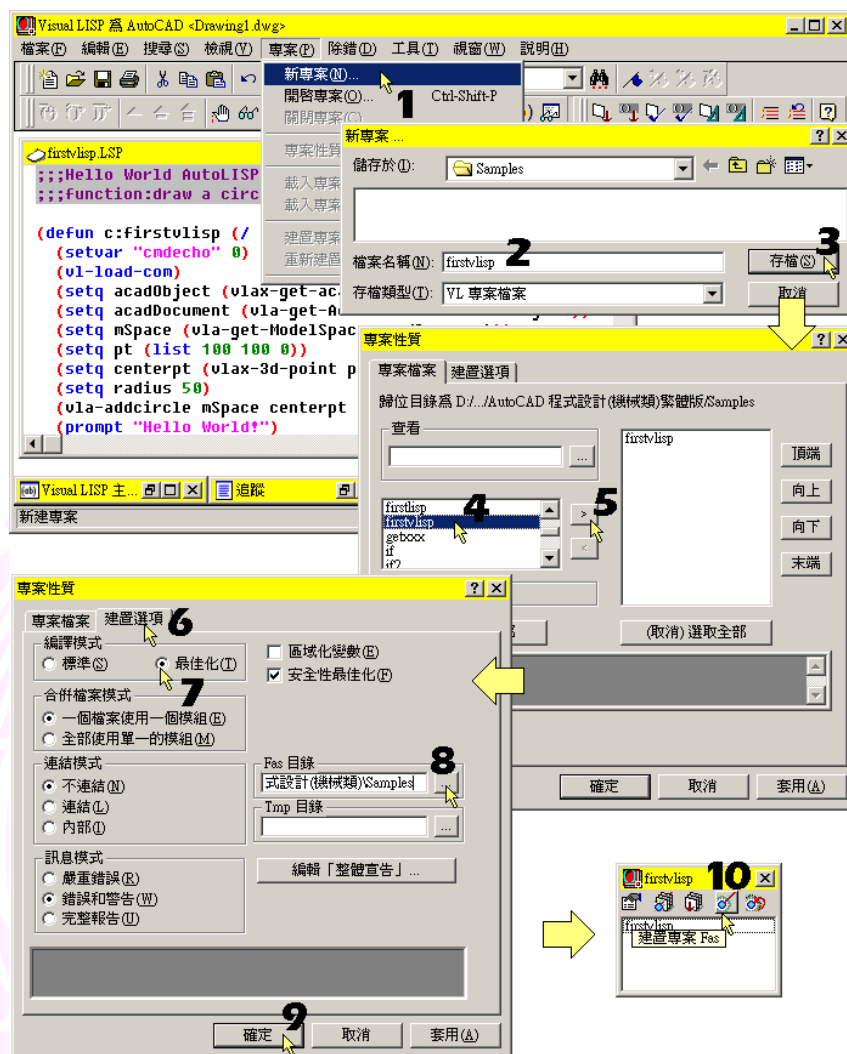
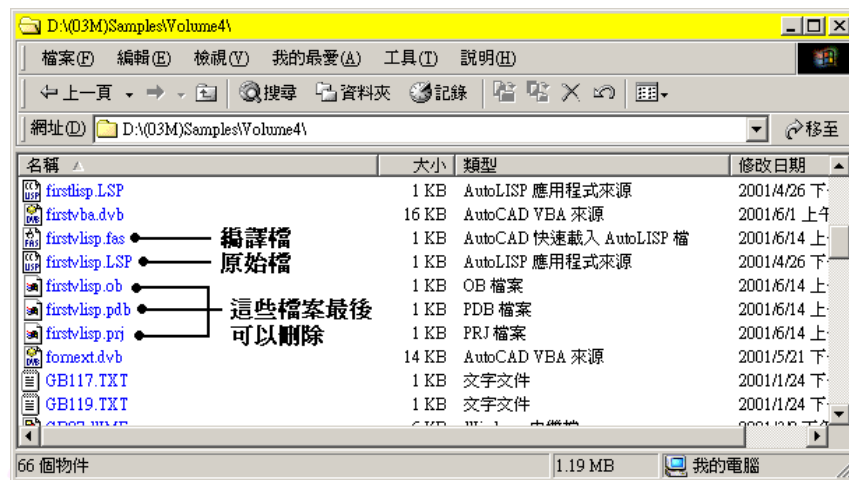


圖 2-23 編譯 firstvlisp.lsp 的操作

圖 2-23 的步驟說明如下：

- 步驟 1：選擇要建立專案的目錄
- 步驟 2：輸入專案名稱
- 步驟 3：點取「存檔(S)」鈕
- 步驟 4：點取要編譯的原始程式檔名，如 V-Xysquare
- 步驟 5：點取「>」鈕，隨即於右邊的框中出現 V-Xysquare 字樣
- 步驟 6：點取「建置選項」標籤鈕
- 步驟 7：選擇「最佳化(T)」選項
- 步驟 8：選擇要存放編譯檔的目錄
- 步驟 9：點取「確定」鈕
- 步驟 10：點取「建置專案 Fas」鈕來開始編譯

3. 編譯完成後，當我們到(03M)Samples\Volume4\的目錄下，就可以看到類似如下畫面：



本貼圖內的文字
編譯檔 原始檔
這些檔案最後可以刪除

圖 2-24 編譯後的結果

如圖 2-24 所示，以後，爲了保護您的智慧財產，您可以將 .LSP 檔另外存起來，而僅留下 .FAS 檔案來載入執行即可。

2-4 Hello World! 程式初步（VBA）

承上節，一樣的這個簡單的畫圓範例，以下我們將以 VBA 的語法來撰寫。

2-4-1 VBA 程式的撰寫與執行部分

請依下步驟來執行整個 VBA 程式的撰寫與執行：

1. 請進入 AutoCAD 中，並點取「工具(T)」下拉式功能表下，「巨集(M)」項目後的「VBA 管理員(V)...」選項。
2. 點取「新建(N)」鈕新建一個專案。
3. 點取「Visual Basic 編輯器(V)」鈕，進入 Visual Basic 編輯視窗中。

4. 點取「插入(I)」下拉式功能表下的「模組(M)」選項。並在「屬性區」視窗中將模組名稱 Module1 修改為 firstvba。
5. 點取「插入(I)」下拉式功能表下的「程序(P)...」選項。加入一個名稱為 hello 的副程式。如圖 2-25 所示：

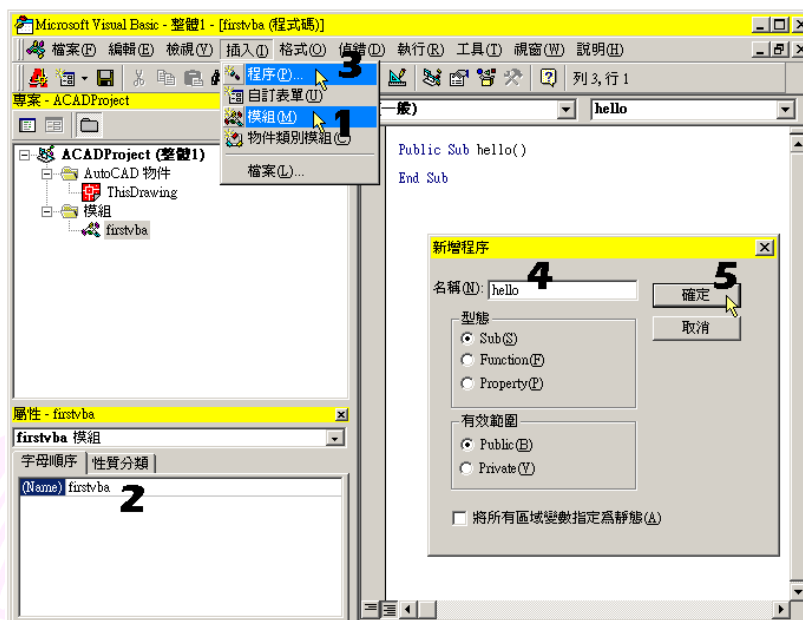


圖 2-25 撰寫 firstvba.dvb 程式前的操作

6. 然後在「程式本文區」中輸入以下的程式本文（注意：下示每條程式前括號裡的數字，如 (1)，(2)...等，是爲了方便稍後的語法說明，請勿輸入）：
 - (1) 'Hello World VBA Program-----firstvba.dvb
 - (2) 'function:draw a circle and print "Hello World"
 - (3)
 - (4) Option Explicit
 - (5) Public Sub hello()
 - (6) Dim centerpt(0 To 2) As Double
 - (7) Dim radius As Double
 - (8) Dim circ As AcadCircle
 - (9) centerpt(0) = 100#
 - (10) centerpt(1) = 100#
 - (11) centerpt(2) = 0#
 - (12) radius = 50#
 - (13) Set circ = ThisDrawing.ModelSpace.AddCircle(centerpt, radius)
 - (14) ThisDrawing.Utility.Prompt "Hello World!"
 - (15) End Sub

分析：首先，第 (1)、(2) 行的註解語法，與 LISP/VLISP 稍有不同。VBA 的註解行語法符號是一單引號 “'”。第(4)行表示要強制對模組中的所有變數進行明確宣告，否則編譯器會認錯。從第(5)行起將開始一個名稱爲 hello 的公用程式（副程式）。第(6)到第(8)行的語法是用來宣告座標點、半徑等變數名稱。第(9)到(12)行的語法則用來賦予點與半徑值給對應的變數名稱。第(13)行是 VBA 執行畫圓的語法。第(14)行則是在 AutoCAD 指令行上列印 "Hello World!" 字樣。第(15)行是結束 hello 公用程式（副程式）的固定語法。

7. 執行時，如果您還在程式撰寫的編輯視窗內，則請按 <F5> 鍵，再點取「執行(R)」鈕，即可執行此程式。如果您已經跳出，則請如先圖 2-22，執行「工具(T)」下拉式功能表下的「載入應用程式(L)...」選項來載入，只是要選取的檔案是 firstvba.dvb。然後，再於指令提示區下如下鍵入：

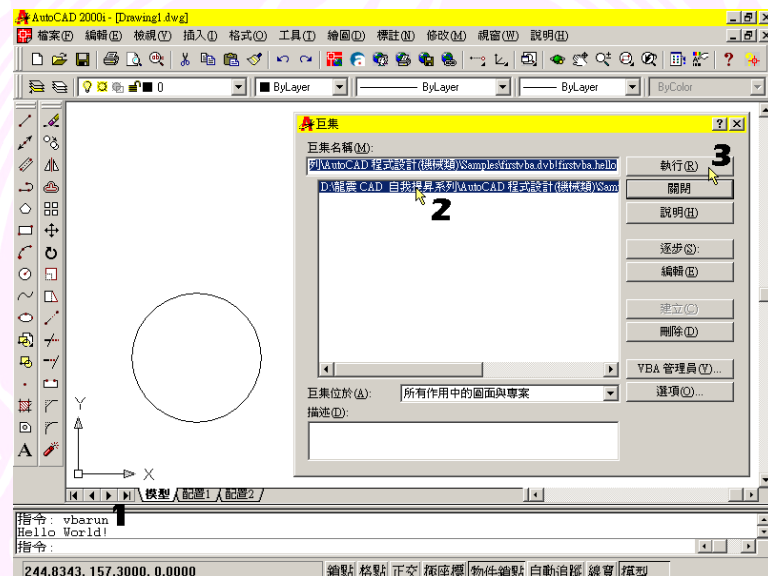


圖 2-26 firstvba.dvb 程式的執行

2-4-2 VBA 程式的加密部分

由於 VBA 程式必須在 AutoCAD 裡進行，所以爲了程式的安全，請使用本節所述的方式來加密保護您的 VBA 程式。承上一節範例，請依下述步驟操作：

1. 請進入 AutoCAD 裡。
2. 點取「工具(T)」下拉式功能表下，「巨集(M)」選擇項後的「VBA 管理員(V) ...」選擇項，隨即出現如下畫面。請依圖 2-27 所示操作。

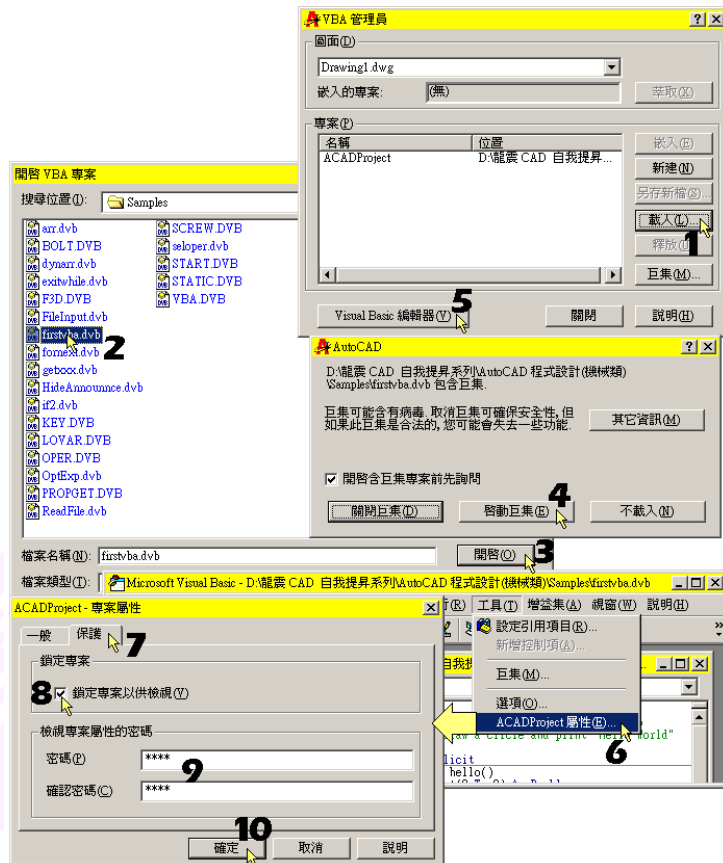


圖 2-27 firstvba.dvb 程式的加密操作

- 如此，當您下一次還要編輯這個 VBA 的原始程式時，就會出現一詢問您密碼的視窗。此時，您必須正確回答密碼後，才能進入編輯視窗內見到程式本文！

2-5 重要的環境設定

好啦！練習過簡單程式的進入、撰寫、執行與後續的編譯或加密等完整的流程後，還要再加入本節所告訴您的重要系統環境設定才算完整。這個設定就是搜尋路徑的設定。

以前在 DOS 版的時代裡，要進入 AutoCAD 以前都要執行批次檔，此批次檔內如容如下所示：

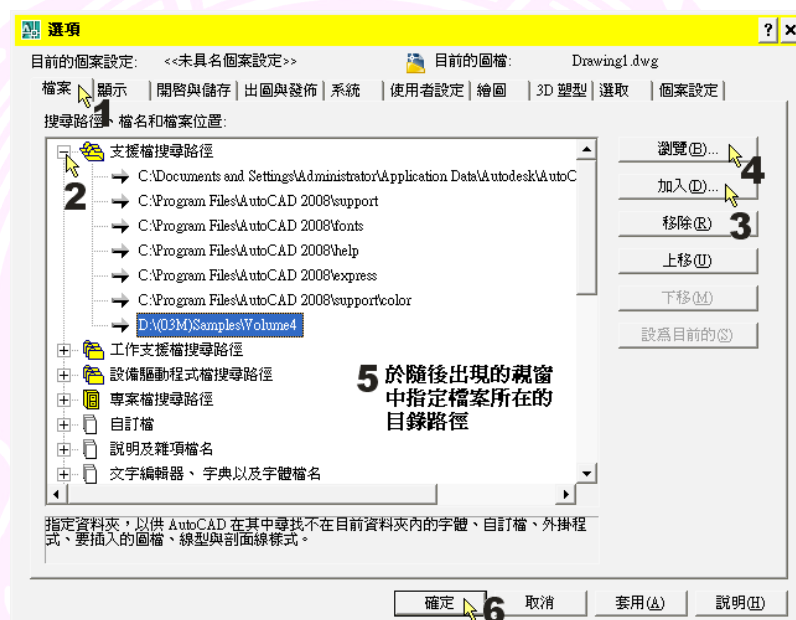
```
SET ACAD=C:\ACAD\...
SET ACADDRV=C:\DRV\....
SET ACADCFG=C:\....
```


等這樣的檔案搜尋設定。到了 Windows 版 AutoCAD 問世後，這些設定仍然是有必要設定的；否則，AutoCAD 怎麼會知道我們所有的 AutoLISP/VLISP/VBA 檔案範例、圖塊檔案範例或功能表檔案都放在那裡呢？不過，檔案搜尋的設定介面變了，請依下述步驟來執行這些檔案搜尋路徑的設定：

1. 請於 AutoCAD 的指令提示號後鍵入：

指令: CONFIG <Enter>

2. 現在，我們要設定的是本電子書範例檔案所在的搜尋路徑。請點取「支援檔案搜尋路徑」名稱前的「+」符號，再按圖 2-28 操作：



本貼圖內的文字

於隨後出現的視窗中指定檔案所在的目錄路徑

圖 2-28 設定本電子書範例所在的搜尋路徑

一開始，您的畫面裡可能只有前面的幾個預設的 AutoCAD 路徑。我們必須將我們放置工作檔案的目錄記錄於此；如此，AutoCAD 才會知道要到這些目錄裡來找。如圖 2-28 所示，要加入的目錄路徑應該是：

\\03M\Samples\Volume4

操作的方法是：點取視窗右邊的「加入(D)...」鈕；然後，再直接鍵入路徑或

點取「瀏覽(B)...」鈕來選取路徑輸入均可。

其他的設定依預設值可也。如有您還有其他的目錄需要設定搜尋路徑，則請自行設定之。

2-6 程式裡的路徑指向問題解決方案

設定檔案搜尋路徑是針對檔案，但是還有一個很重要的問題：即以後在程式裡指向檔案的路徑寫法是否就可以簡化到只要指定檔名即可，而不需大費周章的指定完整路徑？答案是：不行！那怎麼辦呢？因為如果在程式裡就指定固定的路徑，那萬一欲讀取或寫入檔案所在的目錄不是程式裡所指定的那個，那豈不是要逐個修改程式？

是的，這些都是問題。在 AutoLISP/VLISP 方面，我們將執行以下的解決方案步驟來處理這個問題（注意：我們建議您也跟著以下的步驟來做）：

1. 首先，AutoLISP/VLISP 並沒有提供解決這個問題的適當功能，所以我們必須使用其他的方式來處理。於是我們想了一個辦法：於一個文字檔中寫下我們要得取檔案的路徑，然後想辦法讓一個 LISP 程式能夠讀取這個路徑並將此路徑記憶到一個變數中；如此，我們就能將這個字串變數來與要存取的檔案名稱結合，而形成完整路徑。同時，當變換欲存取檔案所在的目錄時，就只要再到此文字檔中改變路徑內容即可。
2. 好！這個文字檔可以隨意命名，但我們將之稱為：dragon.cfg，並將其放在 C 磁碟的根目錄上。這是因為每一台電腦一定會有 C 磁碟。此檔案內容如圖 2-29 所示：

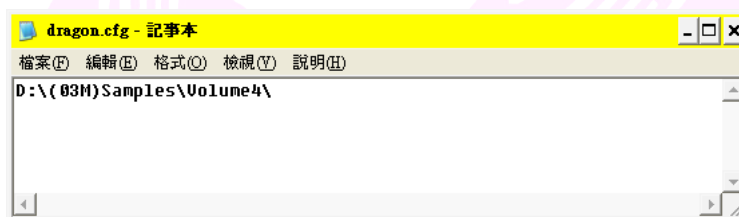
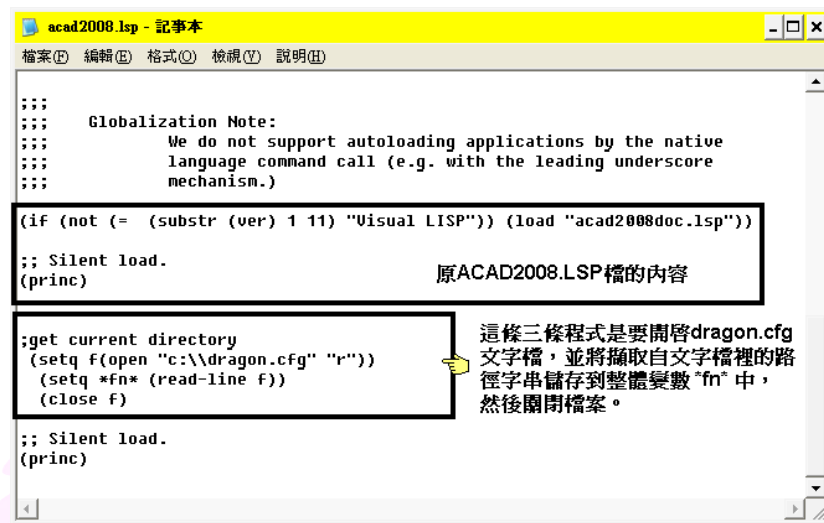


圖 2-29 dragon.cfg 檔的內容

3. 那麼，現在有哪一個 LISP 檔案最適合來讀取這個檔案呢？我們都知道：在 AutoCAD 裡，有一個一進入 AutoCAD 就會自動讀取的 LISP 檔案。該檔在 2000 版以前稱為 **ACAD.LSP** 檔，在 2000 版以後，這個 LISP 檔案就稱為 **ACAD2000.LSP**。您要注意的是：這個檔名會隨 AutoCAD 的版本號而變化。例如，2004 版的此檔就名為 **ACAD2004.LSP**；2008 版的就稱為 **ACAD2008.LSP**；餘此類推。

所以，我們就到 AutoCAD 的系統目錄 — SUPPORT 目錄中找到這個檔案，並在此檔案的末端撰寫圖 2-30 所示的三條程式（本範例以 AutoCAD 2008 版的 ACAD2008.LSP 檔為例）：



```
;;;
;;; Globalization Note:
;;; We do not support autoloading applications by the native
;;; language command call (e.g. with the leading underscore
;;; mechanism.)
;;;
(if (not (= (substr (ver) 1 11) "Visual LISP")) (load "acad2008doc.lsp"))

;; Silent load.
(princ)

;get current directory
(setq f (open "c:\\dragon.cfg" "r"))
(setq *fn* (read-line f))
(close f)

;; Silent load.
(princ)
```

原ACAD2008.LSP檔的內容

這條三條程式是要開啓dragon.cfg 文字檔，並將擷取自文字檔裡的路徑字串儲存到整體變數 *fn* 中，然後關閉檔案。

本貼圖內的文字

原 ACAD2008.LSP 檔的內容

這條三條程式是要開啓 dragon.cfg 文字檔，並將擷取自文字檔裡的路徑字串儲存到整體變數 *fn* 中，然後關閉檔案。

圖 2-30 在 ACAD2008.LSP 檔中增加三條程式

我們在範例光碟\03M\Samples\Volume4\目錄中，將提供您 ACAD2008.LSP 和 ACAD2000.LSP 兩檔案給您選擇應用。使用其他 AutoCAD 版本的，則請按圖 2-30 自行加上。

- 如此，在後續的 AutoLISP/VLISP 程式語法中，就可以使用類似：

```
(setq f (open (strcat *fn* "oilcup.txt") "r"))
```

這樣的語法來結合 *fn* 整體變數的路徑值與欲開啓檔案的檔名，而形成了完整路徑。以後，只要變更 dragon.cfg 文字檔裡的內容，就可以很方便的更改存取的路徑了。

注意：爲了您的方便，dragon.cfg 與 ACAD2008.LSP(或 ACAD2000.LSP) 檔都隨附於本電子書的範例光碟的\03M\Samples\Volume4\目錄中。爲了順利本電子書的運作，請將 dragon.cfg 複製到 C 磁碟機的根目錄上，而將 ACAD2008.LSP(或 ACAD2000.LSP)複製到 AutoCAD 系統目錄下的 SUPPORT 目錄中(如 C:\Program Files\AutoCAD 2008\Support，千萬不要複製錯)。

那 VBA 呢？很遺憾的，在 VBA 中我們找不到有關可以有效解決此問題的語法。上面所建立的 cn 變數，在 VBA 中並無法擷取。所以，通常還是要在程式中指定完整路徑！這在本電子書的 VBA 程式範例中，就可以明顯的看出來！

啓發性習題

一.選擇題(單複選混合)

- 1.() 有關 AutoLISP/VLISP 的程式撰寫的敘述，何者為非？
 - (a) 可直接在文書處理程式上撰寫，不一定使用「Visual LISP 編輯器」來撰寫
 - (b) 它們的副檔名一個是 .LSP，另一個則為 .VSP
 - (c) VLISP 可支援 ActiveX。
 - (d) 以上皆非
- 2.() 我們編寫 LISP 程式時，還是建議多用「Visual LISP 編輯器」是因為：
 - (a) 在此視窗中要偵錯比較容易
 - (b) 可以自動幫我們寫程式
 - (c) 在此有很多的程式輔助撰寫功能可以應用
 - (d) 以上皆是
- 3.() 以下何者是載入 AutoLISP/VLISP 程式的方法：
 - (a) 直接在指令提示號後使用 (load "LISP 檔名") 的語法來載入
 - (b) 點取「工具(T)」下拉式功能表 → 「載入應用程式(L)...」選項來載入
 - (c) 在「Visual LISP 編輯器」內使用「檔案(F)」下拉式功能表 → 「開啓檔案(O)...」選項來載入
 - (d) 以上皆非
- 4.() 以下何者是執行 AutoLISP/VLISP 程式的方法：
 - (a) 直接在指令提示號後鍵入 AutoLISP/VLISP 檔的名稱
 - (b) 直接在指令提示號後鍵入主程式 c: 後所定義的名稱
 - (c) 在「Visual LISP 編輯器」內的「主控台」視窗內執行

(d) 以上皆非

5.() 以下何者是載入 VBA 程式的方法：

- (a) 直接在指令提示號後使用 (load "LISP 檔名") 的語法來載入
- (b) 點取「工具(T)」下拉式功能表 → 「載入應用程式(L)...」選項來載入
- (c) 在「VBA 管理員」內使用「載入(L)...鈕來載入
- (d) 以上皆非

6.() 以下何者是執行 VBA 程式的方法：

- (a) 直接在指令提示號後鍵入 VBA 檔的名稱
- (b) 直接在指令提示號後鍵入主程式所定義的名稱
- (c) 載入後，在「VBA 管理員」內使用「巨集(M)...鈕來執行
- (b) 載入後，點取「工具(T)」下拉式功能表 → 「巨集(M)...」選項來執行

7.() 以下敘述何者為非？

- (a) AutoLISP 程式在 AutoCAD 2000 版以後已經可以編譯
- (b) Visual LISP 的編譯檔案副檔名為 .FAS
- (c) VBA 的原始檔案可以加密
- (d) 以上皆非

8.() 設定系統搜尋路徑的理由是：

- (a) 讓 AutoCAD 可以很快的找到所需的檔案
- (b) 可以幫我們將所找到的檔案自動存檔
- (c) 對程式裡指向檔案的路徑寫法就可以簡化到只要指定檔名即可，而不需大費周章的指定完整路徑
- (d) 以上皆非

二.實作問答題

- 1.請試述在 VLISP 與 VBA 裡偵錯的操作。
- 2.請編譯本電子書中任何一個 VLISP 範例程式。
- 3.請將本電子書任一個現成的 VBA 範例程式加密。
- 4.請實作 2-6 節的內容。

