補充說明

目前 MATLAB 已發表了 MATLAB R2006a 版本 (MATLAB 7.2)，從 2006 年起，MathWorks 公司每年的 3 月與 9 月，將進行兩次 MATLAB 產品的發表，並分別以 a 與 b 來區分本年度的上下版本，因此 R2006a 版本表示 2006 年上半年度的產品，而每一次的發表中都會包含所有產品的新介面、問題修復以及新產品的推出。目前在 R2006a 中，MATLAB 已對 64 位 Windows 支援，以及對其他十個產品進行更新；另外，更將原本的 MATLAB Builder for COM 整合爲 MATLAB Builder for .net 工具箱，除了可編譯 M 檔案爲 COM 元件外，更可以輕易的將 M 檔案編譯爲.net，就可以透過 C#、VB.net、…等程式語言來進行操作，增加 MATLAB 函數的零活性。

MathWorks 公司於 9 月 1 日正式宣佈 MATLAB R2006b 版本上市，這次改版爲 2006 年起第二次改版 (2006 年起每年實施兩次產品更新的標準改版，分爲 a 版的三月更新與 b 版的九月更新)。這次 MATLAB R2006b 改版有包括了更新 MATLAB 7.3 與 Simulink 6.5，以及新增了 Aerospace Toolbox (航空工具箱，飛行器設計、分析、模擬與軟體開發之工具)、MATLAB Builder for Java (將 M 檔案編譯爲 Java 元件之工具)、SystemTest (系統測試工具箱，開發、管理與編輯測試架構，並利用預先定義的測試元件去測試 MATLAB 演算法及 Simulink 模型之工具)、Link for TASKING (供 MATLAB/Simulink 連結 Altium 所開發的 TASKING 產品，以協助建立、測試與驗證嵌入式程式碼之工具)、SimHydraulics (液壓模擬方塊組，模擬包含液壓元件與機械元件的複雜行爲之工具)與 Simulink HDL Coder (供 Simulink 模型與 Stateflow 圖表事件，產生之獨立且可合成的 Verilog 或 VHDL 這兩種硬體描述語言程式碼之工具) 等 6 套新產品。另外，在 MATLAB 主操作環境部份，加強了快速處理大量運算資料的功能，並明顯改善 MATLAB 環境下記憶體容量的使用率，因此處理大量資料(如：大量陣列資料)時，會有明顯的改善。Distributed Computing Toolbox(分散式計算工具箱)可支援 Windows Compute Cluster Server2003 系統以及分散式陣列與平行式數學運算等功能。

另外，MATLAB R2006a 與 R2006b 版本中，對 MATLAB 的操作上做一小幅度的更新，這裡取與本書相關的內容說明之。首先關於 MATLAB Desktop 介面的更新如下：

Windows 下 MATLAB 安裝目錄結構

R2006a 版本以後，於 Windows 作業系統下安裝 MATLAB，其 MATLAB 的安裝目錄有點小幅度的改變，預設會以 MATLAB 爲主目錄而 R2006a 爲子目錄，因此所有 MATLAB 相關產品都會安裝於 R2006a 目錄中。預設安裝於 C:\Program Files\MATLAB\R2006a，若前一版本是安裝於 C:\Program Files\MATLAB7.1。

錯誤記錄報導

一般執行 MATLAB 運算錯誤時，我們會直接觀察 MATLAB 命令視窗所傳回的錯誤訊息，來進行 MATLAB 執行語句的修正；但如果是本身語法沒有問題或是因爲 MATLAB 本身操作上造成的錯誤時，MATLAB 會產生一個錯誤紀錄，並且在下次使用時，MATLAB 會提示使用者將這個錯誤記錄直接 e-mail 給 MathWorks 公司，一旦寄出後，MATLAB 會回信確定已寄至 MathWorks 公司，日後如果有任何修正或改善此錯誤記錄的方法會立即通知使用者。但請注意這個錯誤記錄報導的功能，如果使用 –r 或 /r 選項來開啓 MATLAB 時是無效的。

64 位元的 Linux 作業系統更新了 JVM 版本

對於 64 位元的 Linux 作業系統而言，Java 虛擬機(JVM)的版本，目前 MATLAB R2006a 是使用 Sun 1.5.0_04。

重新組織 Preferences 選單的內容

MATLAB R2006b 版在 Preferences 選單中多了一個新的 Keyboard 面板，主要用以設定命令視窗或 M 檔案編輯器內按鍵、Tab 鍵功能與符號設定等調整。

M-Lint 使用者偏好設定目前加到 MATLAB 的 Preferences 內

MATLAB R2006b 版在 MATLAB 的 Preferences 內新添加了 M-Lint Panel Preferences，以幫助使用者調整預設的 M-Lint 偏好設定來設定顯示或不顯示 M-Lint 指定的訊息，以便使 M-Lint 在檢查我們的 M 檔案時，會顯示哪些建議訊息來告知使用者應該如何更改 M 檔案的內容。此外，如果有安裝 MATLAB Compiler，則更可以設定顯示或不顯示 MATLAB Compiler 的相關訊息。注意這些偏好設定對於 M 檔案編輯器(Editor/Debugger)內自動產生的 M-Lint 碼以及經由目前目錄瀏覽器的目錄報告或由功能表選取“Tools”→“M-Lint”→“Show M-Lint Report”選項所產生的 M-Lint 報告而言，都是有效的。

強化關閉檔案功能

MATLAB R2006b 版中，當使用者目前有開啓多個檔案於 MATLAB 內時，如於 M 檔案編輯器內開啓多個 M 檔案，我們除了可以由功能表選取“File”→“ Close...”選項 ，將指定的檔案關閉外；還可以直接於狀態列上單擊滑鼠右鍵選取關閉所有的檔案或特定的檔案或除了目前檔案外的所有檔案。

強化 help 命令

目前 R2006a 版本，已經可以允許使用 help 命令去顯示 .mdl 檔 (Simulink 檔) 的相關輔助說明，而這個說明主要是使用者用以告知操作者於 Simulink 模型的相關操作方式或模型中對應方塊的輔助說明，這些可以由 Simulink 的功能表

選取“Model Properties”→“Description”選項來添加說明的。舉例來說，假設我們要顯示 Simulink 的 F-14 範例模型所對應的輔助說明，則可以於 MATLAB 命令視窗中輸入：

```
>>help f14_dap.mdl
Multirate digital pitch loop control for F-14 control design demonstration.
```

則會立即告知這個 f14_dap.mdl 的用途。

Help 介面的更新

MATLAB R2006b 版在 Help 瀏覽器內，新增了關鍵字、萬用字元(*)與布林(Boolean)搜尋功能，這非常類似我們於 Google 中搜尋的方式，不過此功能目前無法使用於日本系統。

- 搜尋欄位永遠顯示：當 Help 瀏覽器一開啓後，就會一直顯示搜尋欄位，並且會將搜尋到的結果清單顯示於 Search Results tab，這樣可以便於使用者觀察與比較。
- 關鍵字搜尋：可以透過關鍵字搜尋，找出更正確的結果。關於這一點可以於搜尋欄位上以雙引號("")括住要搜尋的關鍵字部分來完成，舉例來說，輸入 "plot tools" ，則搜尋的結果只會找到對應於 plot tools 完整字句的結果，這有點類似 Google 進階搜尋內的包含完整的字句搜尋功能，因此不會找到任何單獨由 plot 或 tools 所對應的結果。當然使用者可以定義更進階的搜尋，使 MATLAB 能夠更爲精準與快速的搜尋結果，如於搜尋欄位上輸入"plot tools" "figure pallette"，就可以搜尋出 plot tools 與 figure pallette 共同成立的結果。
- 萬用字元(*)搜尋：搜尋中加入萬用字元(*)來取代要搜尋的關鍵字或字元，舉例來說，使用 plot* 進行搜尋，則就會找到所有以 plot 所組成的關鍵字，如：plot、plots 與 plotting 等，因爲 * 表示萬用字元，它可以是任何字元所組成的，因此只要字句中有出現 plot，搜尋就算成立。爲了更爲彈性搜尋結果，我們可以加入多個萬用字元(*)於搜尋中。
- 布林操作元搜尋：有 NOT、OR 與 AND，使搜尋條件更爲彈性，而其中評估的順序是以 NOT 優先，再來是 OR，最後爲 AND，並且由左自右順序性來搜尋。

注意：任何非 Mathworks 的產品所存在的輔助說明，將無法使用 R2006b 版的搜尋功能，但輔助說明仍然會顯示於 Help 瀏覽器內。如果使用者想要提供的輔助說明能夠應用到 Help 瀏覽器的搜尋功能時，則須要整個更新 info.xm 檔內的 helpsearch.db 登錄爲 helpsearch 目錄，並且準備 R2006b 相容的搜尋資料庫。

關於 M 檔案編輯器的更新如下：

使用 dbstop 函數復原中斷點

使用者可以透過 s=dbstatus 與 save 命令的搭配，儲存中斷點的狀態爲 MAT 檔並且透過 dbstop 額外的選項指定中斷點狀態之相關訊息，以便載入 MAT 檔能復原爲上一次中斷點的狀態，舉例來說，設定好中斷點後，執行 s=dbstatus，就可以將中斷點狀態儲爲 s，而使用 dbstatus 的'completenames'選項，就可以儲存絕對路徑與中斷點函數巢狀序列，如果要將 s 儲存爲名爲 mydebugsession 的 MAT 檔，則必須執行 ***save mydebugsession s***，之後執行 ***load mydebugsession*** 就可以載入中斷點資訊 s 於工作空間中，然後執行 dbstop(s)就可以復原這些中斷點。

使用 dbquit('all')將除錯功能停止

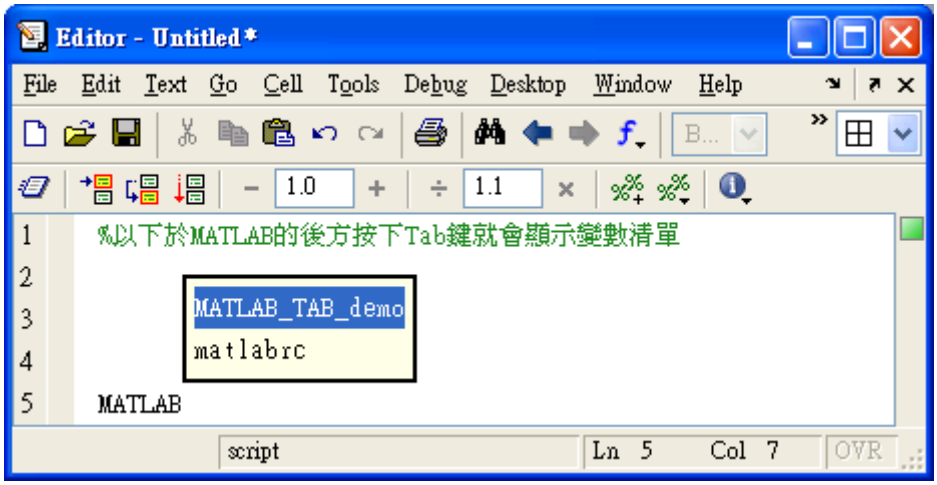
如果使用者一次除錯多個 M 檔案時，則可以執行 dbquit('all') 將所有在執行中的儲錯模式關閉。

Tab 鍵目前可應用於函數或變數名

目前可以於 M 檔案編輯器中，透過 Tab 鍵的功能來搜尋工作空間中的函數或變數名，這與一般 MATLAB 命令視窗中使用 Tab 鍵的功能類似，假設我們僅知道函數或變數某部分的名稱時，於對應的字元後方按下 Tab 鍵後，MATLAB 就會列出可能的函數或變數名清單，並且當使用者選取其中之一後，MATLAB 就會自動將對應的函數或變數名填妥，但須要特別注意於 M 檔案編輯器中是無法使用 Tab 鍵來搜尋檔案或路徑名的。如以下假設於 MATLAB 命令視窗中輸入一變數 MATLAB_TAB_demo：

```
>> MATLAB_TAB_demo=100;
```

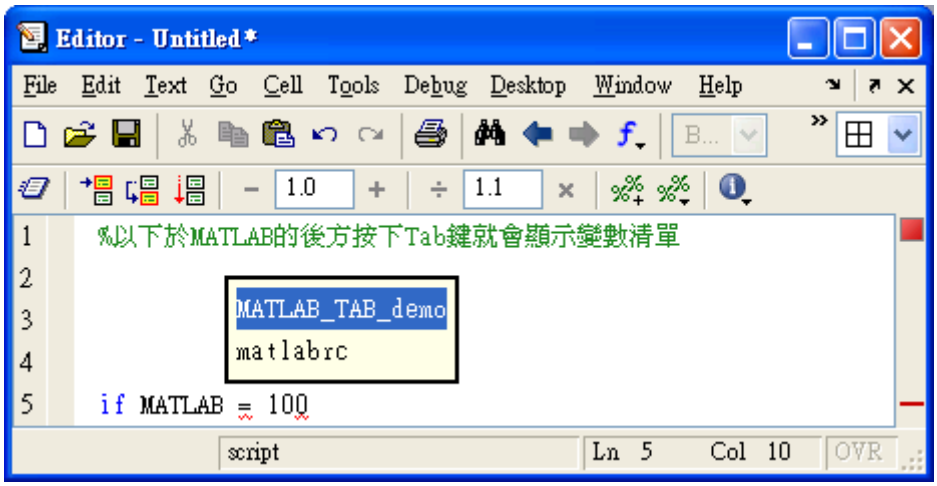
然後於開啓 M 檔案編輯器，輸入 MATLAB 後於 B 字元後方按下 Tab 鍵即會顯示相關的變數清單，如下所示：



接著由鍵盤的上下鍵即可控制清單中的游標，選取 MATLAB_TAB_demo 後按下 Enter 鍵即可，則 MATLAB 就會自動將 MATLAB 補爲 MATLAB_TAB_demo。

同樣的，我們也可以Tab鍵的用法也與MATLAB命令視窗中一樣彈性，只會搜尋字元對應的名稱，因此也可以應用於運算式中的搜尋，如輸入if a = MATLAB運算式後，於MATLAB的B字元後方按下Tab鍵，也會顯示相關的變數清單，如以下所示：

或輸入 if MATLAB = 100 運算式後，於 MATLAB 的 B 字元後方按下 Tab 鍵，也會顯示相關的變數清單，如以下所示：



同樣的，由鍵盤的上下鍵選取清單中的 MATLAB_TAB_demo 後按下 Enter 鍵，則 MATLAB 就會自動將 MATLAB 補爲 MATLAB_TAB_demo，因此我們可以知道 Tab 鍵的功能是非常有彈性的。

M 檔案編輯器中新增了 Go 選單

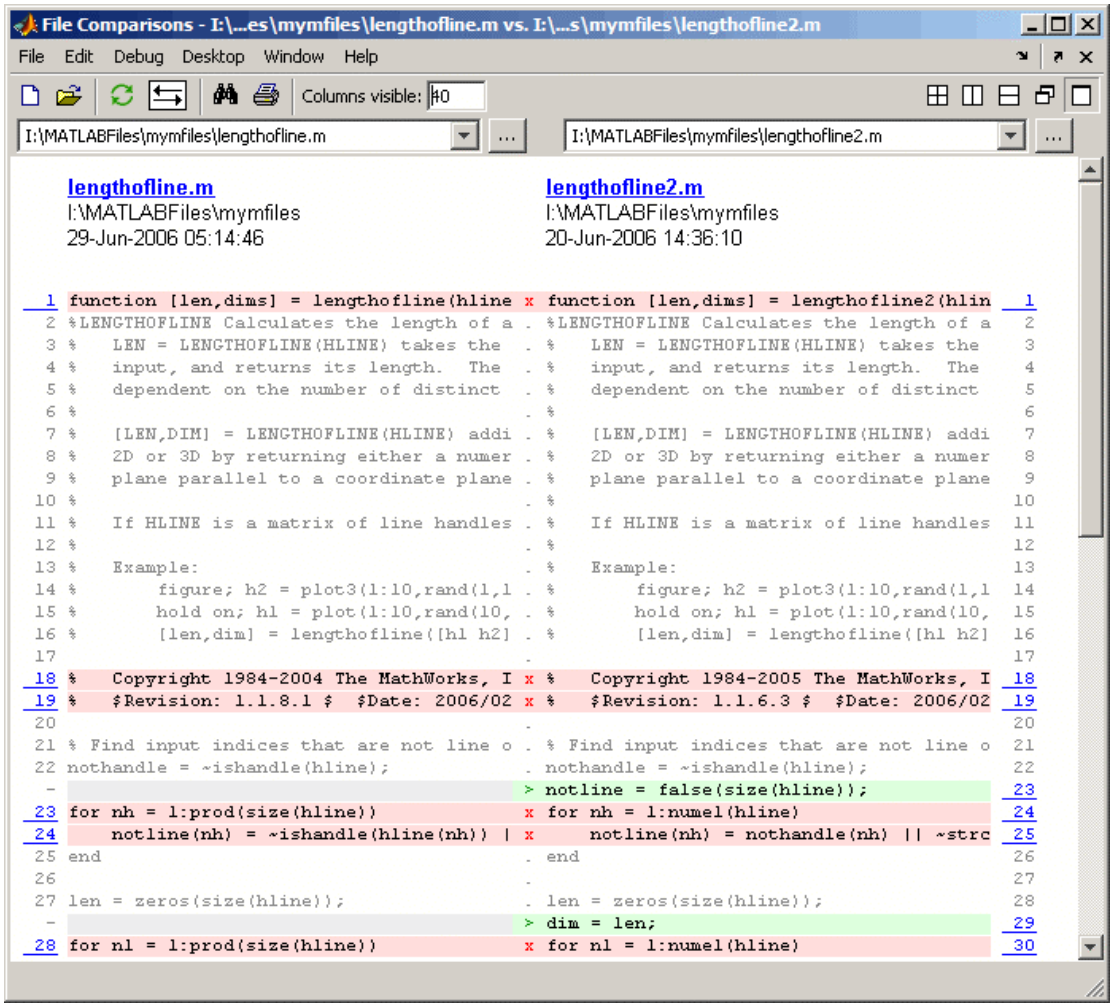
M 檔案編輯器新增了書簽效果，使用者可以於檔案中將特定行建立爲書簽，如此就可以快速的移動到指定的書簽行，這對有大量程式碼的檔案是特別有效

的。若要將特定的程式行設定爲書簽，首先將游標移動到該行首位置，然後由功能表選取“Go”→“Set/Clear Bookmark”選項即可，則書簽圖示就會顯示於該行行首處。若有多個書簽於一個程式中，我們可以由功能表選取“Go”→“Next Bookmark”或“Previous Bookmark”選項，移動到其他的書簽。若要刪除書簽，首先將游標移動到該書簽行任何位置後，由功能表選取“Go”→“Select/Clear Bookmark Next Bookmark”選項即可。注意 M 檔案內所設定的書簽功能，於關閉 M 檔案後就會消失。

另一方面，我們可以透過功能表選取“Go”→“Back”選項或“Go”→“Forward”選項來瀏覽 M 檔案中上一次編輯修改的地方。舉例來說，如果目前開啓一個 M 檔案並且依序更改了檔案中的第 3、9 與 6 行，因此當我們由功能表選取“Go”→“Forward”選項後，游標會由第 1 行移到第 3 行，再選一次“Go”→“Forward”選項，則會由第 3 行移到第 9 行，再選一次“Go”→“Forward”選項，則會由第 9 行移到第 6 行；相反的，如果由功能表選取“Go”→“Back”選項，則會先由第 6 行移到第 9 行，再選一次“Go”→“Back”選項，則會由第 9 行移到第 3 行，再選一次“Go”→“Back”選項，則會由第 3 行移到第 1 行。

新增檔案比較工具

MATLAB R2006b 版中，透過檔案比較工具，會將程式兩檔案程式上不同處以粉紅區域標出，因此我們可以立即了解M檔案之間的不同。首先我們先開啓一個M檔案，然後由功能表選取“Tools”→“Compare Against”選項，選取第二個要開啓的M檔案或直接由目前目錄瀏覽器拖曳M檔案於比較工具中即可。則開啓的兩個檔案會共同顯示於一個視窗中，並且已經由粉紅區域標出程式行不同處，如下圖所示：



M-Lint 內建於 M 檔案編輯器

M-Lint 功能目前內建於 M 檔案編輯器中，以便於使用者可以快速且方便的分析程式碼可能發生的問題，並且建議要如何進行更改。目前 M-Lint 功能具備能夠標出目前程式碼中有問題的程式行並且直接於 M 檔案中連結 M-Lint 訊息報告，以便了解要如何進行更改；以顏色區分這個 M 檔案是否有問題，如果程式碼執行上可能造成錯誤(紅)或警告(橘色)，則加上顏色底線於程式行中，再於右方以顏色於對應程式行標示，並且當滑鼠游標移動至標示處就會顯示對應的訊息，如果整個 M 檔案都沒有問題發生，則右上角會顯示綠色；更改 M 檔案後，不須要儲存 M 檔案或退出 M-Lint 報告，就可以繼續再分析程式碼，並且更新 M-Lint 訊息報告，來觀察更改前後的差異。

若覺得每次進行程式碼分析的速度很慢而要將 M 檔案編輯器內的 M-Lint 功能關閉，則可以由 MATLAB 功能表選取“File”→“Preferences” 選項，然後於開啓的 Preferences 視窗左方選取“Editor/Debugger”→“Language”，於右方的 Language 下拉式選單中選取 M，接著於 Syntax 框架下將 Enable M-Lint messages 不勾選即可，右方的 Underline warnings and errors 可以用以決定程式碼中要標示底線的準則。

強化 M-Lint 功能

MATLAB R2006b 版主要是針對強化隱藏 M-Lint 訊息的功能，以及會顯示 MATLAB 編譯器相關訊息，另外，目前日本系統也可支援 M-Lint 的功能，因此也可以透過產生的訊息來調整程式語句的內容。隱藏不必要或對目前程式來說影響非常小的 M-Lint 訊息是非常有用的，因爲這樣就可以節省不須要的記憶體浪費，並且也可以快速抓住程式修改的重點，以下介紹幾個方法來隱藏 M-Lint 訊息：

- 對一個 M-Lint 訊息而言，直接在其訊息底線上單擊滑鼠右鍵，選取 Ignore this "Terminate statement with semicolon..." ，即可將對應的訊息給隱藏，這是因爲 M-Lint 會加一個%#ok 與訊息 ID 標籤於程式行尾，而這就告訴 MATLAB 要去隱藏一個 M-Lint 訊息了。
- 對一個 M-Lint 訊息而言，直接在其訊息底線上單擊滑鼠右鍵，選取 Disable all "Terminate statement with semicolon..."，即可將所有的訊息給隱藏。
- 由功能表選取“File”→“Preferences” →“M-Lint”選項，直接將偏好設定中已指定的訊息取消即可，使其 M-Lint 運作上不會顯示這些訊息，並且使用者可以將這些設定儲存爲一個 M-Lint 偏好設定檔；另外，在 M 檔案編輯器中，可以由功能表選取“Tools” →“M-Lint”選項，選取 M-Lint 偏好設定檔來載入使用之。

M-Lint 目前可以顯示 MATLAB 編輯器編譯與配置相關的訊息，只要使用者的 MATLAB 中有安裝 MATLAB 編輯器，並且已經於偏好設定中將 Show MATLAB Compiler deployment messages 選取就會驅動這此功能，比如說，cd 函數是不能進行編譯的，因此在程式中就會提醒使用者 MCC does not permit the CD function 訊息，告知不要使用 cd 命令於程式中；另外，使用者可以於 M 檔案編輯器的功能表中選取“Tools” →“M-Lint”選項，來驅動編譯相關的 M-Lint 訊息，並且也可以使用隱藏一般 M-Lint 訊息的方法來隱藏編譯相關的 M-Lint 訊息。

強化 mlint 訊息識別碼與%#ok 語句

MATLAB R2006b 版中，mlint 函數目前多添加了一個輸入參數-id 來告知 M-Lint 訊息識別碼，舉例來說，當我們於 MATLAB 命令視窗下執行 ***mlint('filename.m', '-id')*** ，則 MATLAB R2006a 就會傳回以下的訊息識別碼：
L 22 (C 1-9) 2:AssignmentNotUsed : The value...

如果是 MATLAB R2006b 則會傳回以下的訊息識別碼：
L 22 (C 1-9): NASGU: The value ...

我們可以發現兩個版本的差異在於 R2006a 版本會多添加一個數值標識符，也就是以上的 2，因此必須將它修正爲目前新的格式後，才是正確的訊息識別碼。另外，%#ok 語法主要是用以將 M 檔案中程式行所顯示的 M-Lint 訊息隱藏，而目前的已經強化爲可以去隱藏指定的 M-Lint 訊息，舉例來說，於 M 檔案中寫入 ***data{nd} = getfield(flds,fdata{nd});*** 這行程式，則 M-Lint 所執行的訊息爲以下兩個：

34: 'data' might be growing inside a loop; consider preallocating for speed.
34: Use dynamic fieldnames with structures instead of GETFIELD. Type 'doc struct' for more information.

如果是在 MATLAB R2006b 之前的版本，我們僅可以將整行程式所對應的所有 M-Lint 訊息給隱藏，也就是使用以下的作法：

```
>>data{nd} = getfield(flds,fdata{nd});          %#ok
```

但是因爲 MATLAB R2006b 有提供 M-Lint 訊息識別碼，因此我們就可以依據 M-Lint 訊息識別碼來隱藏對應的訊息，當然也是可以將整行程式所對應的所有 M-Lint 訊息給隱藏，舉例來說，M-Lint 訊息識別碼爲 GFLD 所對應的 M-Lint 訊息，則可以使用以下的方式：

```
>> data{nd} = getfield(flds,fdata{nd}); %#ok<GFLD>
```

若要隱藏多個訊息，則可以一次指定多個 M-Lint 訊息識別碼 ，並且其間以逗號間隔開來，如以下所示，隱藏 M-Lint 訊息識別碼 GFLD 與 AGROW 所對應的訊息：

```
>> data{nd} = getfield(flds,fdata{nd}); %#ok<GFLD,AGROW>
```

如果使用以上這個語句於 MATLAB R2006b 之前的版本中，則 M-Lint 會忽略 M-Lint 訊息識別碼，導致會將整行程式所對應的所有 M-Lint 訊息給隱藏。

儲存 M-Lint 偏好設定

MATLAB R2006b 版中，mlintrpt 新增了一個參數，以便使 M-Lint 相關的偏好設定能夠於指定的檔案中使用。如以下所示，可以將偏好設定檔 fullpath_to_configfile.txt 於指定的檔案或指定目錄中的檔案使用：

```
mlintrpt('fullpath_to_file', 'file', 'fullpath_to_configfile.txt')
mlintrpt('fullpath_to_dir', 'dir', 'fullpath_to_configfile.txt')
```

舉例來說，於功能表選取“File” →“Preferences” →“M-Lint”選項，設定儲存偏好設定的檔案爲 NoSemiSetting.txt，然後再經由偏好設定或 M 檔案編輯

器中的 M-Lint 或是由 mlintrpt 來使用這些設定，這一點的好處是可以依據不同的 M 檔案與偏好設定來彈性化我們的程式，如以下爲 mlintrpt 的使用範例：

```
>> mlintrpt('lengthofline.m', 'file', 'I:\MATLABFiles\NoSemiSettings.txt')
```

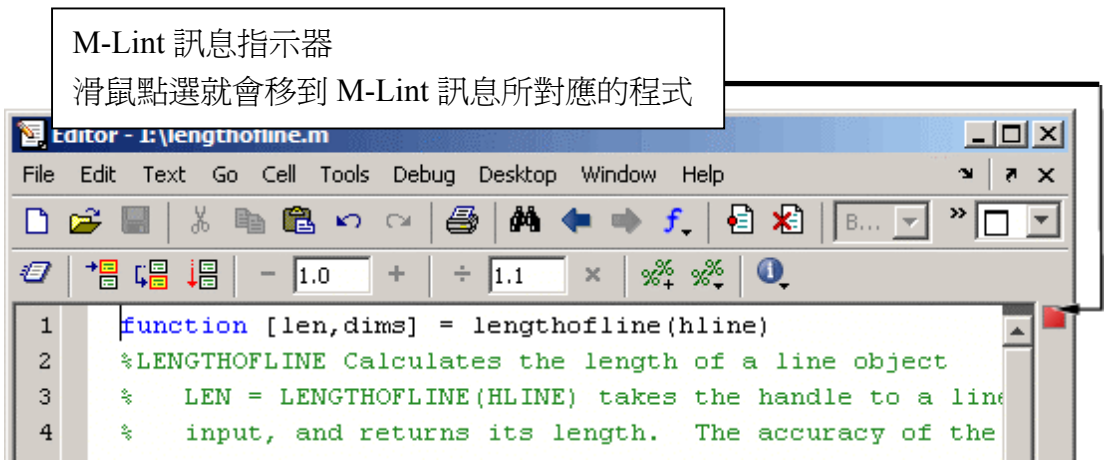
以上程式執行完成後，就會於 MATLAB 預設的網頁瀏覽器中，依據 I:\MATLABFiles\NoSemiSettings.txt 所指定好的 M-Lint 訊息之偏好設定，顯示目前目錄 lengthofline.m 這個 M 檔案中，程式相關的 M-Lint 報告了。以下介紹一個簡單的 M-Lint 操作方式：

(1). 首先由偏好設定中確定我們已經驅動 M-Lint 訊息功能了，因此由功能表選取“File”→“Preferences”→“M-Lint”，將 Enable integrated M-Lint warning and error messages 勾選；另外，爲了便於觀察程式中哪些程式碼是有問題，我們將 Underlining 設爲 Underline warnings and errors，表示它會透過 M-lint 去找出程式中可能會發生警告或錯誤的部份，並且分別以橘色或紅色表出底線，更進一步當我們將滑鼠游標移動到底線的位置處，就會立即顯示 M-Lint 訊息，提醒使用者程式應該如何修改，最後於 Preferences 視窗中按下【OK】按鈕即可。

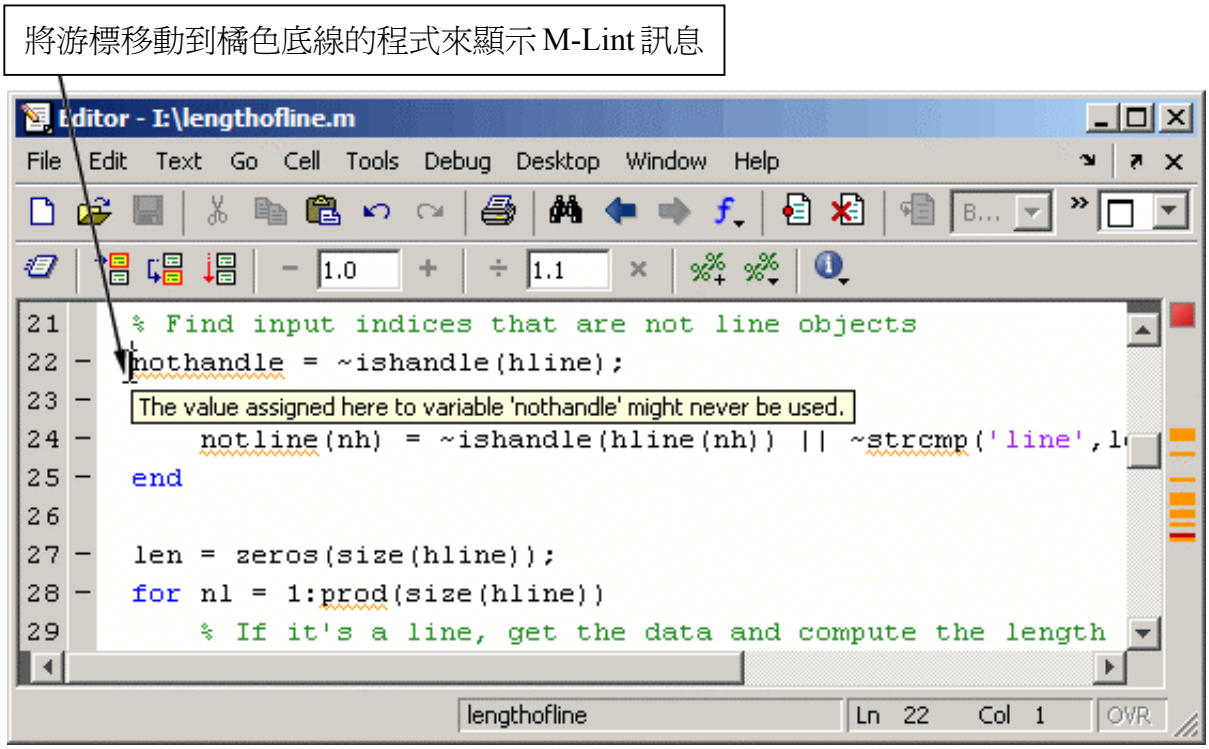
(2). 這裡以MATLAB 範例 lengthofline.m 做爲例子，因此我們可以透過以下的方式來開啓這個檔案：

```
>>open(fullfile(matlabroot,'help','techdoc','matlab_env','examples','lengthofline.m'))
```

M-Lint 訊息指示器會於視窗的右上角顯示，並且以顏色來區分目前 M 檔案程式碼的好壞狀態，其中顯示爲紅色表示程式監測到錯誤；顯示爲橘色表示程式監測到警告或程式碼寫法上是可以進一步改善來增加效能的，注意綠色表示目前沒有任何錯誤產生；如果顯示爲綠色表示程式非常優良，沒有錯誤、警告或須要修改的地方。如下圖顯示紅色，表示程式中有錯誤產生。



(3). 我們直接由滑鼠點選 M-Lint 訊息指示器，就會直接將游標移到 M-Lint 訊息所對應的程式行中，以這個例子來說，一開始點選 M-Lint 訊息指示器，就會將游標移動到第 22 行程式的位置處，因爲這是第一個發生 M-Lint 訊息的位置，並且也是一個橘色的 M-Lint 訊息，當我們把游標移動到這行程式有底線的位置時，就會立即顯示 M-Lint 訊息，如下圖所示。注意底線也會以顏色標出程式 M-Lint 的訊息， 也就是說紅色底線，表示該程式有錯誤，若爲橘色底線，表示該程式有警告或可以修改來增加效能 。



這個訊息說明了，變數 nothandle 雖然有被指定一個值，但因爲在後續的程式行中，都沒有再次使用到 nothandle 這個變數了，因此可以將 22 整行刪除，或者將第 24 行的~ishandle(hline(nh))改爲 nothandle(nh)。當我們一更改程式瑪後，M-Lint 訊息就會立即自動更新爲對應於新程式碼的結果。

(4). 確定第 22 行程式修正完後，我們可以再次點選 M-Lint 訊息指示器，移動游標到下一個程式有問題處，或直接將滑鼠移動到右方 M-Lint 訊息列中的標誌，來顯示該行程式所對應的 M-Lint 訊息，同樣的，紅色標誌表示該行程式監測到錯誤；橘色標誌表示該行程式監測到警告或程式碼寫法上是可以進一步改善來增加效能的，注意綠色表示目前沒有任何錯誤產生。舉例來說，我們將滑鼠移動到右方 M-Lint 訊息列中有紅色標誌出現的位置處，然後點選這個標誌就會將滑鼠游標移動到程式中對應出現錯誤的位置，因此這裡就會移動到 48 行程式 ***temp = diff([data{1}(:) data{2}(:) data{3}(:)]);*** ，這個訊息說明了這行程式的分隔符號可能

設定不當，關於這一點，可以由功能表選取“File”→“Preferences”→“Keyboard”中將 Delimiter Matching 下的 Match on arrow key 勾選後，移動滑鼠游標到每個分隔符號上，觀察分隔符號其前後設定上是否有相配合，該行程式搜尋結果爲優良的，但其實主要的問題是發生在 ***data{3}(;)*** 這個地方，它必須更改爲 ***data{3}(:)***，當我們修正程式後，則本行程式就不再有底線出現了，也就是說，我們修正好這個錯誤了，並且 M-Lint 訊息指示器就會變爲橘色的，指出目前程式只剩下警告或可以修正來增加效能的訊息。最後當所有程式都已修正完畢後，M-Lint 訊息指示器就會顯示爲綠色，表示目前 M 檔案已經沒有問題了。

強化 profile 功能

profile 函數操作上新增了-nohistory 選項，如果先前已使用 profile 函數加上-history 選項操作時，則-nohistory 選項執行後，就不會進一步的記錄相關操作上的歷程，但其他功能還是照常進行。此外，Profiler 提供更精密的計算方式，使程式執行的計時上能夠更爲精準，目前更進一步可以透過 profile 函數或 Profiler 介面統計遞迴函數(Recursive Functions)所花費的時間於 FunctionTable's TotalTime。

另外，目前 FunctionTable 表包含了一個新的 PartialData 值，用以告知執行 profile 的過程中，如果有任何函數被修改，則 PartialData 值爲 1，例如過程中有函數被編輯或清除，因此資料僅會收集至修改前的那一時間點。在前一版本中，FunctionTable 中還有包含 AcceleratorMessages，目前已經將這一功能去除。

工具列的 Refresh 按鈕移除

原本 Profiler 工具列中包含的更新按鈕，已經被移除了，因爲它的功能就與一般 Profiler 報表中的【Refresh】按鈕功能一樣，因此取代了更新按鈕。

publish 函數新增了 catchError 選項

publish 函數於操作上新增加了 catchError 選項，以便決定產生 M 檔案報告時，發生錯誤是否要繼續進行。

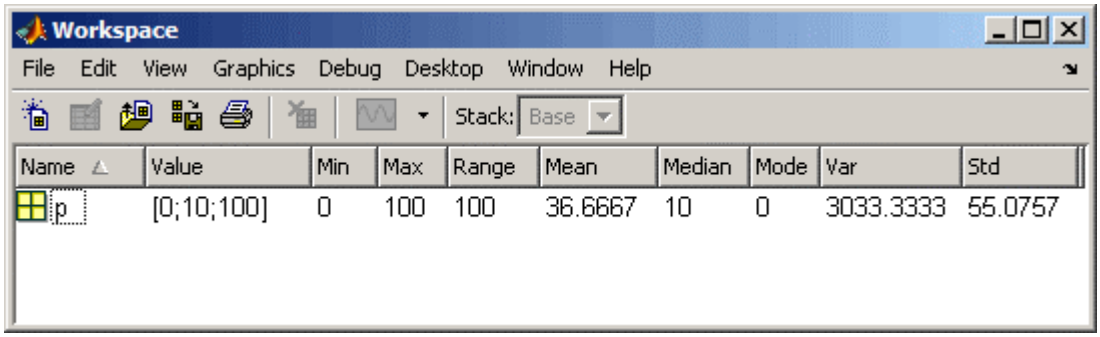
notebook 設定更新

在 MATLAB 7.1 中，notebook 函數已經被強化爲可以自動監測 Microsoft Word 的版本與其他需要的相關訊息，因此當我們使用 ***notebook –setup*** 具有參數 wordversion、wordlocation 與 templatelocation 來設定 Microsoft Word 版本、位置與暫存位置等訊息時，會顯示警告訊息，但在 MATLAB R2006b 中，是會直接顯示一個錯誤訊息，告訴使用者不能去使用這些參數。

關於工作空間瀏覽器與目前目錄視窗的更新如下：

工作空間瀏覽器新增了資料統計運算功能

MATLAB R2006b 版中，工作空間瀏覽器目前包含了幾個新的行欄位來顯示統計計算的資料值，這些包含了：最小值、最大值、範圍值、平均值、中間值、模數(Mode)、變異數與標準差等，我們可以由功能表選取“View”→“Choose Columns”選項，來使指定的資料值於特定行顯示，如以下所示：



可以於陣列編輯器內開啓大的陣列

R2006b 版已經可以允許於陣列編輯器內開啓大的陣列，如果是在之前版本要開啓大的陣列時，會產生錯誤訊息並開啓失敗的。

目前目錄視窗將視覺化目錄的功能移除

目前目錄視窗已經將視覺化目錄(Visual Directory view)功能移除，因爲它已內建於目前目錄視窗，並且大部分的檔案存取操作也都能夠直接由目前目錄視窗來進行。

新增了 toolboxdir 函數

函數 toolboxdir 會傳回指定工具箱的絕對路徑名稱，這對 MATLAB Compiler 來說是非常有用的，因爲工具箱的目錄路徑定義上與 MATLAB 路徑有所差異的，這樣可以避免編譯上的錯誤。

關於數學運算的更新如下：

加強數學運算

MATLAB R2006a 版對於時滯微分方程式(delay differential equations，簡稱 DDEs)而言，目前 MATLAB 提供了一個新的二次求解函數 ddesd，它不同於目前存在的 dde23 函數，因爲 ddesd 函數主要是應用於一般時滯方程式，因此使用者可以提供一個方程式當作 ddesd 函數執行上的輸入參數，則 MATLAB 運算後會傳回對應求解器的時滯向量，相關的說明可由 help ddesd 得知，並且關於 DDEs 方程式求解的初始值問題上，可由 MATLAB Mathematics 相關的輔助說明來了解。另外，也提供新函數 idivide，它的功能非常類似於整數運算時的 A./B，差別處在於當我們指定不同的整數化方式時，如：fix、round、floor 或 ceil，分數的商依據這些準則化整數後，運算上的結果是有所不同的。

而 gallery 函數目前提供一個新的選擇性輸入參數爲 classname，classname 的輸入值必須爲字串'single'或'double'，因此當使用 gallery 時，我們指定好 classname，則 MATLAB 就會產生對應類別的矩陣。另外，當必須使用 condest 函數，運算 1-norm 條件數的稀疏矩陣時，會非常有效率。另外，expm 函數目前使用一個改善的演算法來計算指數矩陣，而這個演算法經常須要少數的矩陣乘法運算。

MATLAB R2006b 版本，提供了以下幾個新的函數：

函數	功能描述
amd	amd 演算法，類似 symamd，但速度快許多
bvpxtend	形成一個推測結構以延伸邊界值求解問題
ldl	滿 ldl 因式分解並求解 Hermitian 矩陣

其他 MATLAB R2006b 相關的更新介紹入下：

max 與 min 函數應用相角於複數求解

對於複數的輸入而言，min 與 max 求解上主要是依據複數的強度來求取極大值與極小值，也就是 min(abs(x)) 與 max(abs(x))，但對於相同強度的元素而言，目前 MATLAB R2006b 是使用相角來求取極大值與極小值，也就是 min(angle(x)) 與 max(angle(x))。因爲在先前的版本中，min 與 max 是依據複數的強度來求取極大與極小值，而對於相同強度的元素，僅使用第一個元素來進行運算，這樣是不符合 sort 函數的作動方式，因此之前計算的結果，請再於 MATLAB R2006b 中運算一次。

lu 函數新增額外的輸出

lu 函數添加了一個額外的輸出參數來幫助使用者改善稀疏 lu 因式分解的數

值穩定度，如以下的使用方式：

[L,U,P] = lu(A)
傳回上三角矩陣 U 與一個具有單位對腳線元素的下三角矩陣 L，以及一個排列矩陣 P，它們之間的關係爲 L*U=P*A，其中 P 主要是用來幫助數值穩定用。假設有一 A 矩陣爲：

```
>>A = [ 1     2     3
        4     5     6
        7     8     0 ];
```

一般使用上若要求取 LU 因式分解，則必須最少指定兩個輸出參數於 lu 函數，來儲存因式分解後的上、下三角矩陣：

```
>>[L1,U] = lu(A)

L1 =
    0.1429    1.0000         0
    0.5714    0.5000    1.0000
    1.0000         0         0

U =
    7.0000    8.0000         0
         0    0.8571    3.0000
         0         0    4.5000
```

其中 L1 是一個下三角排列矩陣，因此如果使用者對調第二列與三列，然後對調第一列與二列，則形成的矩陣就是一個對角線上具有單位元素 1 的下三角矩陣了，同樣的，上三角矩陣也有一樣的特性。另外，我們可以透過 A= L1*U 的結果來驗證因式分解的正確性，因而 *X = inv(A)* 的解等同 *X = inv(U)*inv(L1)* 的解。如果 lu 運算中多添加一個輸出參數，則可以用以儲存排列(Permutation)矩陣的結果 P：

```
>>[L2,U,P] = lu(A)

L2 =
    1.0000         0         0
    0.1429    1.0000         0
    0.5714    0.5000    1.0000

U =
```

```
    7.0000    8.0000         0
         0    0.8571    3.0000
         0         0    4.5000

P =
     0     0     1
     1     0     0
     0     1     0
```

其中 L2 = P*L1：

```
>>P*L1

ans =

    1.0000         0         0
    0.1429    1.0000         0
    0.5714    0.5000    1.0000
```

我們可以使用以下方式來證明它們之間的關係：

```
>>P*A - L2*U

ans =
     0     0     0
     0     0     0
     0     0     0
```

同樣的，*inv(U)*inv(L)* 的結果可以由 *inv(P)*inv(A)* 表示；*d = det(A)* 的結果可以由 *d = det(L)*det(U)* 表示；AX=B 方程式的解 *X = A\B* 的結果可以由 *Y = L\B;X = U\Y* 表示。

提升 chol 與 ldl 函數上三角與下三角運算效能

對於稀疏 chol 而言，計算 chol 與 ldl 上三角與下三角矩陣因子的效能有明顯的增加。

lu、luinc 與 ldl 函數的排列向量功能

對於有排列向量功能的 lu、luinc 與 ldl 函數而言，目前有提供記憶體儲存，以及處理大資料並有明顯增加計算效能之功能；此外，使用者目前可以將排列資訊儲存在一個 1-by-N 的向量，而不是 n-by-n 的矩陣。

lu 與 spparms 函數中提供了一個新的二元素門檻

在 lu 與 spparms 函數中提供了一個新的二元素門檻 (Threshold)，也就是使用*[...] = lu(A,thresh)* 的格式，以便於計算稀疏 lu 與稀疏右除(\)上能更爲彈性。

關於資料處理與分析的更新如下：

目前 MATLAB R2006b 支援 AMD 2.0、COLAMD 2.5、CHOLMOD 1.1、UMFPACK 5.0 等新版本的函式庫。其他更新如下：

支援於 64 位元 Windows XP 系統內進行大資料集運算

MATLAB 目前有支援 64 位元 Windows XP 系統，可是允許使用者處理更大的資料集，目前最高限制是可以允許每個變數有 2 GB 的大小，並且可以於同一時間內儲存多個變數於記憶體中。

64 位元系統的稀疏陣列運算

在 64 位元的系統中，MATLAB R2006b 已經將內部儲存稀疏陣列的方式改變了，這個變化對於一般 MATLAB 的使用者不會有太大的影響，但對於有存取到稀疏陣列的 MEX 檔而言，在 64 位元的系統中，必須透過以下的方式重新由 MATLAB R2006b 編譯 MEX 檔具有新的稀疏陣列格式：

- 修改 MEX 檔內所有稀疏陣列的宣告語句，因爲要能於 64 位元的系統中正確執行，宣告中就必須使用 mwSize 與 mwIndex 類型來取代大部分的資料類型，如：取代 int 或 INTEGER*4。
- 安排一個負數值至一個用來把持稀疏陣列索引值與大小的變數於 MEX 檔中，如：我們可以指定索引值爲一個旗標，如果爲-1 則表示這個索引值並未設定，但必須注意不能使用負整數值於 mwSize 或 mwIndex 中。
- 最後編譯中使用-largeArrayDims 重新編譯 MEX 檔即可。

設定環境變數

MATLAB R2006a 版本目前可直接透過 setenv 函數來設定系統的環境變數。

函數刪除

在 MATLAB R2006b 中執行以下所列的函數或功能時，因爲這些函數或功能已經刪除，因此會操作失敗並給出錯誤訊息，若在 R2006a 版中使用，則會產生警告訊息並且這些函數或功能也可能無法執行成功，這些包含了：

(1) 執行 bvpval、quad8、table1、table8 與 bessela 函數。
(2) sort 函數運算中，以複數整數爲輸入資料來運算的情形。
(3) gt、ge、lt 與 le 函數運算中，以複數整數爲輸入資料來運算的情形。
(4) beta 函數運算中，當使用三個輸入參數時。
(5) gamma 函數運算中，當使用兩個輸入參數時。
(6) 以 eigs 的選擇性結構欄位 opts.cheb 與 opts.stagtol 進行運算時。

以上(1)~(6)項這些情形，未來都不再受 MATLAB 支援，因此使用上都會發生錯誤的。

基本擬合工具與統計分析工具支援產出 M 檔案功能

於 MATLAB 繪圖視窗功能表選取“File”→“Generate M-File”選項，目前可以將繪圖視窗中的基本擬合工具(Basic Fitting Tool)或統計分析工具(Data Statistics Tool)所繪製的結果，產出對應的 M 檔案，並且使用者可以直接於 MATLAB 下使用這個 M 檔案，則會依據輸入的指定資料，來繪製具有相同圖形屬性的圖形，並且也會依據這個資料，去進行擬合或分析的動作；另外，使用 Generate M-File 的好處是，可以透過這個程式碼來了解如何與基本擬合工、統計分析工具互動，這樣也可以幫助使用者日後建立相關的程式。

關於程式使用上的更新如下：

可儲存大於 2 GB 的資料於 MAT 檔

MATLAB R2006b 中，允許使用者使用 HDF5 格式的 MAT 檔，來儲存大於 2GB 的資料，關於這一點，可以於 save 命令的操作過程中使用-v7.3 儲存選項來達成，如以下，將大於 2GB 的變數 v1 與 v2 儲存於 myfile.mat 中，則可以使用：

```
>>save -v7.3 myfile v1 v2
```

因此我們可以知道 MATLAB R2006b 預設是無法直接儲存 HDF5 格式的 MAT 檔，一定要透過-v7.3 儲存選項來達成，而目前 MATLAB R2006b 預設所儲存的標準 MAT 格式，是可以在 MATLAB 7.0、MATLAB 7.0 與 MATLAB R2006a 中使用，使用者也可以透過 save -v7 方式，明確的指定須儲存爲標準格式。不過在未來的版本中，MATLAB 預設的 MAT 檔就是以 HDF5 格式所建立的，這是爲了能夠處理大量的資料，使用者也可以透過 save -v7.3 方式，明確的指定須儲存爲 HDF5 格式；若要儲存爲標準格式的 MAT 檔，就必須使用 save -v7 的方式了。下表爲一個摘要性的介紹，說明了不同儲存選項所造成的結果與可用的版本：

儲存選項	功能描述	可用版本	預設版本
save -v4	儲存爲可以於 MATLAB 4 版中開啓的 MAT 檔	5 或 5 之前的版本	4, 5
save -v6	儲存爲可以於 MATLAB 5 或 6 版中開啓的 MAT 檔	7 或 7 之前的版本	6
save -v7	儲存爲可以於 MATLAB 7 版中開啓的 MAT 檔	7 .3 或 7.3 之前的版本	7.3
save -v7.3	儲存大於 2GB 的資料	7 .3 或 7.3 之前的版本	7.3 之後的版本

另外，如果使用者要在 64 位元的系統中，儲存一個非常大資料的變數爲 MATLAB 7 (或之前版本的)之 MAT 檔時，如果 save 命令未添加-v7.3 儲存選項時(也就是不是透過 save -v7.3 的方式來儲存變數的)，則 MATLAB 會忽略這個大資料的變數於儲存的過程中，並且傳回一個警告訊息；如果是在 32 位元的系統中，嘗試以 load 載入-v7.3 格式的 MAT 檔內之變數，並且這個變數是太大以至於無法擬合 32 位元的系統之記憶體位址空間時，MATLAB 會忽略這個大資料的變數於載入的過程中，並且傳回一個警告訊息。

save 命令的壓縮與字元編碼選項去除

在 MATLAB 7.0 到 MATLAB R2006a 的版本中，使用者可以在 save 命令儲存的過程中，透過–compress、–nocompress、–unicode 與–nounicode 等選項，來

驅動儲存的過程中是否壓縮 MAT 檔或是否使用字元編碼原則來儲存 MAT 檔，不過這些選項在 MATLAB R2006b 中都不再支援了，因爲 MATLAB R2006b 預設就有壓縮 MAT 檔與字元編碼原則的功能，也就是直接透過 ***save -v7 filename*** 即可；當然也可以選則不去使用壓縮 MAT 檔與字元編碼原則這兩個功能，因爲這樣就可以將建立好的 MAT 檔於 MATLAB6 或之前版本中使用了，關於這一點，則必須使用 ***save -v6 filename*** 來建立 MAT 檔，或者我們也可以直接於 MATLAB 的偏好設定中，先設定好所有欲建立 MAT 檔將不具有壓縮與字元編碼原則的功能，也就是先開啓 MATLAB 的 Preferences 對話框，左方樹狀結構選取 General → MAT-Files，然後將右方的 Ensure backward compatibility (-v6)勾選即可。

載入精靈可產出 M 檔案程式碼

目前 MATLAB R2006b 的載入精靈新增了產生 M 檔案程式碼的選項，因此當使用者於 Import Wizard 對話框的右下角處按下【Generate M-code】按鈕，並且一旦使用者按下【Finish】按鈕完成載入的動作後，MATLAB 就會開啓 M 檔案編輯器產生對應的 M 檔案，這個 M 檔案爲 importfile.m (如果這個名稱已經存在 MATLAB 會以 importfileN.m，其中 N 爲一個大於目前存在 importfile.m 檔的數字)，透過 importfile.m 這個 M 檔案就可以用來載入由剛才 Import Wizard 對話框中選定類型的檔案，就好像一個小型的載入精靈功能一樣，以便簡化我們載入檔案的過程。importfile.m 這個 M 檔案具有以下的輸入與輸出參數：

- 輸入參數 fileToRead1：欲載入檔案的檔名。
- 輸出參數 newData1：將載入的資料存至結構 newData1。

此外，importfile.m 這個 M 檔案是可以再經過修改的，以便使我們的載入工作更爲彈性。當使用 importfile.m 這個 M 檔案載入檔案後，就會像載入精靈一樣，於工作空間儲存由這個檔案所產生的資料於指定的變數中。

預留了 MATLAB 關鍵字

MATLAB R2006b 版中，預定兩個新關鍵字 parfor (於分散式計算工具箱中指定一個 for 迴圈) 與 classdef (一個 MCOS 類別開始所定意的訊號) 給 MATLAB 使用，使用者可以透過 iskeyword 函數來查看是否爲關鍵字，因爲關鍵字是不能用於檔名或檔案識別碼的命名上。

強化 whos 命令的顯示功能

MATLAB R2006b 版中，whos 命令所輸出的資料有些微的不同，這些改變如下所示：

- 多添加了一個輸出行 Attributes，來記錄變數的屬性是爲 sparse、complex、global 或 persistent。
- Class 行下所顯示的"array"與"object"這兩個字現在都已被去掉了，舉例來

說，原本表示雙精準度陣列或計時器物件爲 double array 與 timer object，現在都僅會顯示 double 與 timer 而已。

- MATLAB 不再顯示元素的"Grand total"值之相關摘要訊息，如：目前工作空間內有一個 8 個位元組的變數，則就會顯示 Grand total is 1 element using 8 bytes；並且變數佔用的位元組直接顯示於 Bytes 行。
- 新欄位爲'persistent'，如果是永久(persistent)變數則顯示爲 1；反之，則爲 0。

如以下爲在 MATLAB R2006b 版本中執行 whos 的例子，我們可以發現輸出的外觀皆有明顯的不同了：

```
>>whos

Name          Size          Bytes    Class           Attributes

VCell         5x7            2380    cell
vComplex      6x2             192    double          complex
vDouble       6x5x9          2160    double
vFuncHdl      1x1              16    function_handle
vGlobal       0x0               0    double          global
vObject       1x1             248    timer
vPersist      0x0               0    double          persistent
vSparse       15x15           244    double          sparse
vStruct       1x3             804    struct
```

強化 unique 函數的操作

我們知道使用 unique 函數可以用來找出輸入向量中元素值是唯一的元素，而目前的 MATLAB R2006b 版本中，unique 函數具有一個新的 first 選項，可以用以傳回一個具有元素的向量，並且這個向量是用以表示輸入向量的唯一元素之最小索引值，假設我們建立一個 A 向量爲：

```
>>A = [16 7 5 41 2 16 8 2 6 3 16 6 2 5 2 5];
```

若僅單獨使用 ***unique(A)*** 來找出唯一元素，則除了會將唯一元素找出外，還會將輸出結果由小到大排列，如以下所示：

```
>>v = unique(A)

v =
     2     3     5     6     7     8    16    41
```

但是如果我們於 unique 函數中加入 first 或 last 選項來幫助我們搜尋，則傳回的結果，將會以索引值排列的先後順序來記錄，如以下所示：

```
%由左開始找，找到符合的就傳回值與索引值
>>[v, m1] = unique(A, 'first');
>>m1

m1 =
     5     10      3      9      2      7      1      4


%由右開始找，找到符合的就傳回值與索引值
>>[v, m2] = unique(A, 'last');
>>m2

m2 =
    15     10     16     12      2      7     11      4
```

try-catch 所產生的警告

在 MATLAB R2006b 版本中，對於單行輸入 try-catch 的語法上有嚴格的限制，因此當我們使用 try try_statements, catch catch_statements, end 這種格式來處理程式時，是會有以下警告訊息出現的：

```
Warning: This try-catch syntax will continue to work in R2006b, but may be illegal
or may mean something different in future releases of MATLAB.
```

爲了避免這個警告訊息於 MATLAB R2006b 版本中發生，使用者必須插入一個分隔字元(如：逗號、分號或換行)，於 catch 與 catch_statements 間，不過我們還是建議使用以下的方式來建立 try-catch 的語句，這樣會比較直覺與方便：

```
try
    try_statements
catch
    catch_statements
end
```

相同 M 檔案或函數名稱的衝突不再

在 MATLAB R2006b 版本中，對於相同名稱 M 檔案或函數衝突，所造成的警告訊息已經被移除：

"Function call foo invokes /somewhere/on/the/path/foo.m, however, function

/somewhere/ahead/on/the/path/FOO.m that differs only in case precedes it on the path."

在之前版本中，主要要因爲下列的原因使呼叫函數 (如：以上的 foo 函數) ，產生警告訊息的：

- 相同名稱的 MATLAB 檔案儲存於 MATLAB 搜尋路徑，也就是說，目前的搜尋路徑上有多個名爲 foo 的檔案。
- 檔名相同，但大小寫不同的 MATLAB 檔案 (如：以上的 FOO.m 與 foo.m) 儲存於 MATLAB 搜尋路徑。

fprintf(0,...)會產生錯誤

若是在執行 MATLAB R2006b 版本中，執行 fprintf(0, ...) 與 fwrite(0, ...) 是會發生錯誤的，因爲指定檔案識別碼爲 0，但在之前的版本還是可以正常執行。

修復指定非純量的結構陣列欄位給單一變數造成的錯誤

MATLAB R2006b 之前的版本中，指派一個非純量的結構陣列欄位給一個單一變數是會發生錯誤的，舉例來說，以下應該要指定 S.A 的輸出給四個變數，如果僅指定一個變數，則會傳回錯誤訊息：

```
% Create a 1-by-4 structure array S with field A.
>>S(1).A = 1;      S(2).A = 2;      S(3).A = 3;      S(4).A = 4;

% Assigning S(1).A and S(2).A works as expected.
>>[x y] = S.A

x =
        1
y =
        2

% Assigning only S(1).A should work, but does not.
>>x = S.A;
??? Illegal right hand side in assignment. Too many elements.
```

不過以上這個錯誤已經於 MATLAB R2006b 版本中修復好了。

xlsread 函數支援更多的格式

xlsread 函數目前支援 Excel 檔有其他的格式(如：HTML)可以使用，只要系統中的 COM 伺服器是有用的。這些格式我們可以由 xlsfinfo 函數來得知相關的

訊息，也就是由 [A, DESCR, FORMAT] = xlsfinfo('FILENAME') 就可以了解指定的 Excel 檔 FILENAME，所能允許的讀取格式 FORMAT 爲何，就可以應用於 xlsread 函數中了。

函數 sendmail 執行上不再須要 ASCII 訊息

使用 sendmail 函數寄 E-mail 信，目前已經不再嚴格限定一定要使用 ASCII 字元編碼原則，可以允許使用不同格式所寫的資料來進行傳送。

I/O 函數可以進行字元編碼

fopen 函數目前多了一個選擇性參數，一個用以定義字元編碼原則的名稱或別名(alias)字串，用於定義檔案相關的字元編碼原則；如果沒有輸入這個選擇性參數或輸入空字串(' ')，則 MATLAB 會以預設的字元編碼規則來定義之；如果 fopen 的使用上僅輸入檔案識別碼 fid，則 fopen 會傳回一個額外的字串，用以定義檔案相關的字元編碼原則，也就是：

[filename, mode, machineformat, encoding] = fopen(fid)
傳回檔案識別碼 fid 所對應的檔名 filename、存取格式 mode、machineformat 字串與字元編碼原則 encoding，注意這個 encoding 字串所傳回的是標準字元編碼原則名稱，可能會與我們於 fopen 中用來開啓檔案所指定的字元編碼原則有些不同。fopen 函數指定字元編碼原則來開啓檔案的使用方式爲：

[fid, message] = fopen(filename, mode, machineformat, encoding)
如此就可以由定義的存取格式 mode 與 machineformat 來開啓指定的檔案，並且該檔案所相對應的字元存取動作都必須依據 encoding 所定意的字元編碼原則。參數 encoding 必須是空字串(' ')或字元編碼原則的名稱或別名，比較常見的名稱有'UTF-8'、'latin1'、'US-ASCII'、'Shift_JIS'、…等，這些我們可以由以下網址去觀察它的內容：

http://www.iana.org/assignments/character-sets

如果使用者沒有定義 encoding 字串或是定義爲空字串(' ')，則 MATLAB 會以預設的字元編碼原則來開啓檔案。

因此 fread、fscanf、fgetl 與 fgets 這些檔案讀取函數，就會依據 fopen 使用中所定義的檔案相關的字元編碼原則來讀取檔案的字元；相對的，fwrite 與 fprintf 函數也是會依據 fopen 使用中所定義的檔案相關的字元編碼原則來寫入字元於檔案中。但須要特別注意字元編碼原則也以下的限制：

- 不支援代理對(Surrogate pairs)，每一個代理對被讀取，並當作一個代替的字元，這個字元等於 char(26)。
- 不支援 Stateful 字元編碼原則。

- 位元排列標記無法於任何特定的方法中被詮釋，位元碼會被忽略。
- 使用 fscanf 函數讀取的數目前僅支援 ASCII 之 supersets (除 UTF-16 編碼外) 字元編碼原則。

對 Native-Encoded 檔案進行字元編碼轉換

在先前版本中，使用 fread、fgets、fgetl 或 fscanf 函數讀取 native-encoded 編碼格式的檔案時，若要於這個檔案轉換任何字元編碼原則時，必須使用 native2unicode 函數；同樣的，如果須要轉換編碼原則並透過 fwrite 函數寫入 native-encoded 檔時，也須要使用 unicode2native 函數。但在目前版本的 MATLAB 中，使用者可以直接存取 native-encoded 編碼格式的檔案，而不須要再透過 native2unicode 或 unicode2native 函數，因爲在大部分的情形下，MATLAB 會自動進行字元編碼轉換與 Unicode 編碼的應用，因此假設要讀取 native-encoded 檔案時，我們可以使用以下的作法：

- fread 指定讀取格式爲 '*char' 或 'char=>char'。因此在之前版本中讀取 native-encoded 檔，我們必須使用：

  ```
  indata = native2unicode(fread(fid, '*char'));
  indata = native2unicode(fread(fid, 'char=>char'));
  ```

  雖然在 MATLAB R2006a 版中還是可以執行的，但以下列作法來取代會比較好：

  ```
  indata = fread(fid, '*char');
  indata = fread(fid, 'char=>char');
  ```

- fgets 或 fgetl。因此在之前版本中讀取 native-encoded 檔，我們必須使用：

  ```
  indata = native2unicode(fgets(fid));
  indata = native2unicode(fgetl(fid));
  ```

  雖然在 MATLAB R2006a 版中還是可以執行的，但以下列作法來取代會比較好：

  ```
  indata = fgets(fid);
  indata = fgetl(fid);
  ```

- fscanf 指定讀取格式爲 '%s' 或 '%c'。因此在之前版本中讀取 native-encoded 檔，我們必須使用：

  ```
  indata = native2unicode(fscanf(fid, '%s'));
  indata = native2unicode(fscanf(fid, '%c'));
  ```

雖然在 MATLAB R2006a 版中還是可以執行的，但以下列作法來取代會比較好：

```
indata = fscanf(fid, '%s');
indata = fscanf(fid, '%c');
```

- 使用 fwrite 寫入 native-encoded 檔必須指定寫入格式爲'char' 或 'char*1'。因此在之前版本中讀取 native-encoded 檔，我們必須使用：

```
fwrite(fid, unicode2native(outbuff), 'char');
fwrite(fid, unicode2native(outbuff), 'char*1');
```

若是 MATLAB R2006a 版則爲：

```
fwrite(fid, outbuff, 'char');
fwrite(fid, outbuff, 'char*1');
```

支援編譯 ActiveX 物件的許可碼

基本上 MATLAB 所支援的 ActiveX 物件，設計或執行上都必須具備特定的授權，因此當 M 檔案內有包含呼叫 ActiveX 物件的相關語句時，在編譯的過程中，須確保對物件有充分的執行授權。一般來說，MATLAB 就必須崁入一個授權碼(license key)來控制讀取動作，使用上如果授權碼有符合就可以合法的使用。以下說明由 MATLAB 建立一個授權碼的過程：

A. 由 ActiveX 物件獲得對應的授權碼。

使用 actxcontrol 函數時，必須額外的輸入選擇性參數 runtimelicense，也就是使用以下的方式：

```
actxcontrol(progid,'runtimelicense',true)
```

假設 progid 爲 MFCCONTROL2.MFCControl2Ctrl.1，並且將 runtimelicense 的值設定爲 true，則呼叫 MFCCONTROL2 這個 ActiveX 物件時，就必須要有充分的授權。接著建立一個名爲 actxlicense.m 的 M 檔案於目前的工作目錄中，這個 M 檔案是一具有兩個欄位的結構，如下所示：

- progid：MFCCONTROL2 物件所對應的 ProgID。
- lickey：MFCCONTROL2 物件所對應的授權碼。

因此簡單來說，M 檔案內容爲：

```
function lic = actxlicense(progid)
licensecontrol(1).progid ='MFCCONTROL2.MFCControl2Ctrl.1';    %filled by
                                                              %MATLAB
```

```
licensecontrol(1).lickey = 'Copyright (c) 2005 The MathWorks';      %Filled
                                                                    % by MATLAB
```

B. 建立 ActiveX 物件。

建立一個包含驅動 actxlicense 與 ActiveX 物件的 M 檔案，這個 M 檔案主要是透過%#function 語句，將 actxlicense 於編譯的過程中崁入於執行檔中使用，因此 MATLAB 呼叫該 ActiveX 物件時就可以獲得充分的授權。M 檔案內容爲：

```
function buildcontrol
%#function actxlicense
h=actxcontrol('MFCCONTROL2.MFCControl2Ctrl.1',[10 10 200 200]);
```

接著使用 mcc -m buildcontrol 就可以將 buildcontrol.m 進行編譯並且使用這個對應的 ActiveX 物件。

actxcontrol 會檢查合法的 ProgID

MATLAB R2006b 版本中，若要嘗試插入一個 COM 伺服器於 MATLAB 繪圖視窗中，可能會造成不可預期的錯誤，也就是說，使用伺服器 ProgID 於 actxcontrol 中是會造成錯誤的，爲了避免這個錯誤的發生，現在 actxcontrol 會於登錄器的 HKR/CLSID/Control 路徑中，進行相對應的 ProgID 關鍵字搜尋，因此如果這個 ProgID 並不是屬於指定的 ActiveX 控件時，就會傳回錯誤訊息，因此伺服器的 ProgID 不再能用於 actxcontrol 中了。之前的版本中，並不會進行這個檢查功能，因而使一些伺服器 ProgID，仍然可以正確的於 actxcontrol 下執行，這可能是因爲 Microsoft 指出這個伺服器 GUID 是在對應控件的 GUID 下。舉例來說，之前我們可以使用 Mediaplayer 的伺服器 ProgID (即 Mediaplayer.mediaplayer.1) 於 actxcontrol 中，但在 MATLAB R2006b 版本中是會發生錯誤的，因此以下的程式將無法於 MATLAB R2006b 版本執行：

```
>>h = actxcontrol('Mediaplayer.mediaplayer.1');     % 傳回錯誤
```

正確的控件 ProgID 爲 WMPlayer.OCX.7，也就是說，在 MATLAB R2006b 版本中必須更改爲以下的程式才可以正確執行：

```
>>h = actxcontrol('WMPlayer.OCX.7');     % Use control ProgID
```

不過須要特別注意的是，使用控件的 ProgID 所喚出的 ActiveX 物件，對應的方法與屬性皆會有所不同，請務必小心。當然我們也可以使用以下的方式，將 MATLAB R2006b 版本的檢查功能關閉：

```
>>feature('COM_ActxProgidCheck',0)
```

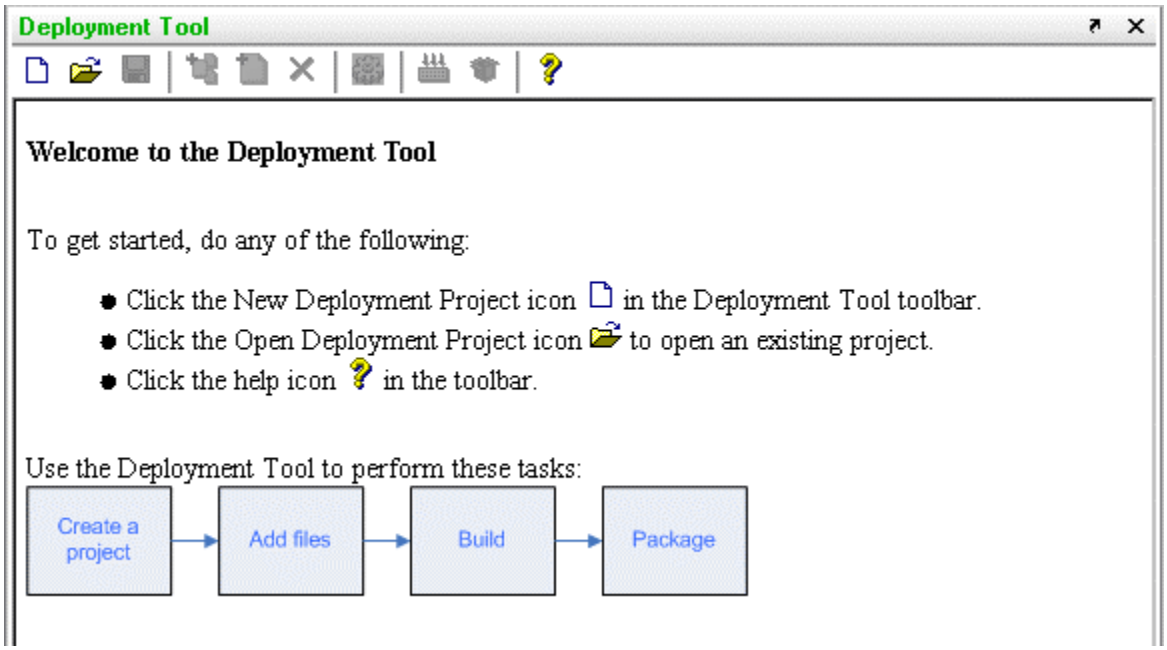
這樣以往的程式又可以正確的執行了，或是透過以下的方式再次將 MATLAB R2006b 版本的檢查功能開啓：

```
>>feature('COM_ActxProgidCheck',1)
```

不過 MATLAB R2006b 版本預設就已經有開啓這個功能了。

關於 MATLAB R2006b 的 MATLAB Builder for Excel 1.2.7 更新如下：

MATLAB R2006b 版本中，MATLAB Builder for Excel 這個介面已經整合至佈署工具 (Deployment Tool) 中，但操作方式還是與之前類似，不過目前的 MATLAB R2006b 版本仍然可以支援使用 mxltool 命令來建立我們的專案於之前的版本中使用，但在未來版本中就不再支援 mxltool 命令了，並且於先前使用 mxltool 命令所建立的 MATLAB Builder 專案，將無法使用於 MATLAB 中。我們可以透過 deploytool 這個命令來開啓 Deployment Tool 介面(不過使用前請先確定已經有使用 ***mbuild –setup*** 指定好相關的編譯器了)：



一開始 Deployment Tool 介面會有一個簡單的介紹告訴使用者要如何操作一個完整的編譯動作，並且如果是第一次執行 deploytool 命令來開啓 Deployment Tool 介面，那麼介面可能是嵌入到 MATLAB Desktop 介面中，如果不想，可以點選 Deployment Tool 介面右上方的 dock 按鈕，將它嵌出。接下來，按下 Deployment Tool 介面工具列上的【New Deployment Project】按鈕(或由 MATLAB 功能表選取“File”→“New Deployment Project”選項也可)，開啓 New Deployment Project 對話框，並選取左方的“MATLAB Builder for Excel”，然後設定專案名稱和位置後按點選【OK】鈕(一般我們會先建立好 M 檔案與專案目錄，則這裡就可以直接點選【Browse】鈕，直接指定我們的專案目錄)，則所產生的元件之元件名稱與類別名稱預設都會跟著這個專案名稱來命名，完成後 MATLAB 會再次顯示 Deployment Tool 介面。

此時，Deployment Tool 介面，會顯示目前專案編譯上所有須要的檔案目錄清單，預設第一個目錄就是我們要將檔案進行編譯的目錄，也就是類別名稱目錄(我

們可以於該目錄上單擊滑鼠右鍵選取“Rename Class”選項來更改類別名稱)，然後點選工具列上的【Add Files】  按鈕後選取欲加入 M 或 MEX 檔案，或直接拖曳位於目前目錄視窗內的 M 或 MEX 檔案至 Deployment Tool 介面的類別名稱目錄中即可；同樣的，這裡要加入 Deployment Tool 介面的所有檔案名稱，不得與函式庫中任何要進行編譯的函數名稱相同。

加入檔案後，就可以開始設定我們的專案相關屬性，可由工具列的【Settings】  按鈕來設定，也就是設定元件名稱、版本號、是否要進行除錯、設定檔案封裝屬性、編譯器的指定、警告訊息的觸發時機…等相關設定，設定完後若無錯誤，最後點選工具列的【Save】  按鈕，將這個專案儲存起來，並點選工具列的【Build】  按鈕，開始建立 Excel Builder 元件，同一時刻，會於 Output 面版顯目前編譯的過程與相關資訊，編譯完後 DLL 會自動登錄於系統中，即可使用這個元件了。其實這個 Deployment Tool 介面的操作上，大部份的步驟都還是與之前的 MATLAB Builder for Excel 類似，只是因爲目前把 MATLAB Compiler、MATLAB Builder for Excel、MATLAB Builder for .Net 與、MATLAB Builder for Java 等介面整合一體，使操作上會有些微的不同而已，但產生的元件或檔案於其他軟體中使用的方式還是一致的。

關於 MATLAB R2006b 的 Symbolic Math Toolbox 與 Extended Symbolic Math Toolbox 3.1.5 更新如下：

在 Symbolic Math Toolbox 與 Extended Symbolic Math Toolbox 3.1.5 中，關於 maple 函數的操作上有些微的改變，目前使用 Maple 的程式碼產生封裝器(Code Generation Package)之方式來呼叫 Maple 函數時，須要明確的說明封裝器的名稱，舉例來說，如透過 codegen 封裝器來計算 x^2-4 的值，則必須將 codegen 這個封裝器輸入：

```
>>maple('codegen[fortran](x^2-4)');
```

使用此種 maple 函數操作的方式，其相關的計算方式與結果，都跟之前的版本是沒有差異的。不過之前版本，在 maple 函數中有使用程式碼產生封裝的情形下，會透過 Maple 的 with 命令，來自動呼叫函數進行運算，所以就不須要指定封裝器於 maple 函數中，如以下所示：

```
>>maple('fortran(x^2-4)');
```

但這樣有時候會造成一個衝突，當指定一個相同於程式碼產生封裝器的函數名稱給 Maple 變數，就會造成錯誤。

關於 MATLAB R2006b 的 Excel Link 2.4 更新如下：

在 Excel Link 2.4 中加了以下幾個新的函數，介紹如下：

- MLGetFigure：載入 MATLAB 繪圖視窗至 Excel 工作表，這樣就可以將圖形結果直接於 Excel 中顯示。
- MLStartDir：定義 MATLAB 啓動的目錄。
- MLUseFullDesktop：定義是否去使用完整的 MATLAB Desktop 介面功能，還是僅使用 MATLAB 命令視窗。

MLShowMatlabErrors：之前版本透過 MLEvalString 來執行 MATLAB 語句時，若發生錯誤可能無法立即或完整的得知，因此目前可以透過 MLShowMatlabErrors 這個函數，來傳回 MLEvalString 來執行 MATLAB 語句時，所發生的 Excel Link 錯誤或 MATLAB 執行上的錯誤。

勘誤表

Chapter 1

P1-4 第 1 行
”機購” 改爲 ”機構”

P1-12 倒數第 2 行
“關建字” 改爲 ”關鍵字”

P1-40 倒數第 3 行
copyfile 的說明去掉，因 P1-39 同表前方已提過了

P1-57 倒數第 11 行
“執行 ex1.m” 改爲 “執行 ex2.m”

P1-62 第 2 行
“行成” 改爲” 形成”

P1-72 第 2 行與第 18 行
“function =fun(a)” 改爲” function p=fun(a)”

P1-75 第 11 行
“試窗” 改爲 ”視窗”

P1-99 第 17 行
“重新儲錯” 改爲 “重新除錯”

P1-99 第 12 行
“遊標” 改爲 “游標”

P1-102 第 1 行
“工能” 改爲 “功能”

Chapter 2

P2-2 第 2 行
“牘” 改爲 ”讀”

P2-6 倒數第 2 行
“訊昔” 改爲 ”訊息”

P2-7 倒數第 6 行
“決大” 改爲 ”絕大”

P2-15 倒數第 1 行
“一大推” 改爲 ”一大堆”

P2-19 第 6 行
“想比” 改爲 ”相比”

P2-29 第 5 行與第 8 行
“%試取-5/3 之餘數” 改爲 “%試取-5/2 之餘數”

P2-33 倒數第 7 行
加法運算”回” 改爲 加法運算”會”

P2-46 倒數第 9 行
“linspac 函數 e” 改爲 “linspace 函數”

P2-53 倒數第 2 行
“應用合” 改爲 ”應用場合”

P2-61 倒數第 10 行
” >>cumsum(A)” 改爲 ” >>A=[1 -1 0]; cumsum(A)”

P2-80 第 11 行
“擴展第三行” 改爲 “擴展第四行”

P2-100 第 2 行
“校多” 改爲 ”較多”

P2-109 倒數第 5 行
“因此若 A 爲” 改爲” 因此若 X 爲”

P2-111 第 8 行, 第 9 行, 第 10 行, 第 14 行, 倒數第 2 行, 倒數第 12 行,
P2-112 第 9 行
“rank([A;B])” 改爲 "rank([A,B])”

P2-112 第 3 行
” x+y+z = 1” 改爲 “x+2y+z = 1”

第 10 行
“最小范數” 改爲 ”最小範數”

P2-154 第 4 行
“必較大” 改爲 ”比較大”

P2-148 第 2 行與 **P2-154** 第 2 行
“決大” 改爲 ”絕大”

P2-155 第 14 行
“決大” 改爲 ”絕大”

P2-160 第 6 行
“決大” 改爲 ”絕大”

P2-161 倒數第 11 行
“負 Lapacian” 改爲 ”負 Laplacian”

P2-164 倒數第 1 行
“D 爲特徵值” 改爲 ”d 爲特徵值”

P2-170 倒數第 7 行
“x2=dy/dx” 改爲 “x2=dy/dt”

Chapter 3

P3-26 第 13 行與第 21 行
“Problem with x and/or y!” 改爲 “Problem with A and/or B!”

P3-27 第 11 行
“將迴圈向量” 改爲 ” 將迴圈向量化”

P3-28 第 6 行
“5~0 間隔 1 的值” 改爲 “0.5~0 間隔 0.1 的值”

P3-28 倒數第 15 行
“則 S 依序對取值” 改爲 “則對 S 依序取值”

P3-34 倒數第 11 行
“非常危及” 改爲 “非常危急”

P3-35 倒數第 13 行
“I>6” 改爲 “i>6”

P3-38 第 10 行與第 15 行
“ex3_11” 改爲 ” ex3_12”

P3-39 ~ P3-42 “成積” 改爲 “成績”

Chapter 4

P4-33 第 15 行
富“利”葉 改爲 富“立”葉

P4-38 倒數第 3 行
“執行 ex4_4.m” 改爲 “執行 ex4_6.m”

P4-42 第 8 行
“若”下 改爲 “落”下

P4-46 第 11 行
“Zi” 改爲 “ZI”

P4-51 第 13 行
“Y1、Y2 與 Y3” 改爲 “XI、YI 與 ZI”

P4-53 倒數第 7 行
“Z 賦來定義” 改爲 “Z 來定義”

P4-66 第 9 行
“決大” 改爲 ”絕大”

Chapter 5

P5-10 倒數第 10 行
“但不爲” 改爲 “但不會”

P5-13 倒數第 9 行
”那是因爲” 後加入 ”strcat 用於水平字串組合而 strvcat 用於垂直字串組合，而”

P5-26 倒數第 13 行
”預設的字元寬且小數點後” 去掉

倒數第 10 行
”3 字元寬且小數點後” 去掉去掉

P5-27 第 12 行
“%5.2f” 改爲 “%5.4f”

P5-35 倒數第 13 行
“S2” 改爲 “s2”

P5-35 倒數第 11 行
“LASTER” 改爲 “laster”

P5-35 倒數第 9 行
“輸出參” 改爲 “輸出參數”

P5-38 倒數第 2 行
“在內函數” 改爲 “在函數”

P5-65 第 5 行 少了

函數	功能描述

表頭

P5-77 第 2 行

"因此僅會取 Sum_of_BC 陣列中的前四個元素，而後四個忽略" 改爲 "因此 reshape 會以行方向重新排列 Sum_of_BC 陣列中的四個元素"

Chapter 6

每頁上方的標題 符號數學工具"書" 改爲 符號數學工具"箱"

P6-4 第 11 行

"兩者定" 前方加入 "因此 sym 與 syms"

P6-11 第 2 行

爲"例" 改爲 爲"何"

P6-13 第 1 行

"X" 改爲 "S"

P6-14 倒數第 7 行

"反求回元" 改爲 "反求回原"

P6-20 倒數第 8 行

"and SYMSUM(s,v,a,b)" 改爲 "與 symsum(s,v,a,b)"

P6-24 第 10 行

"sym(1/2)+1/3" 改爲 "sym(1/2+1/3)"

P6-25 倒數第 9 行

"元素階爲" 改爲 "元素皆爲"

P6-27 第 11 行

"浮點數數" 改爲 "浮點數"

P6-28 倒數第 1~2 行

"s" 皆改爲大寫 S

P6-34 最下方

改爲"d2/dx2y2sinx2"

P6-40 第 1 行
“x/|x|” 改爲 “x/|x|”

P6-41 第 4 行
刪除 “定義的變數)”

P6-42 第 12 行
”一樣的” 後加上 “，而這裡計算 2ax2+2b+c 試試”

倒數第 7 行
“solve(方程式,’變數)” 改爲 “solve(方程式,變數)” 單引號去掉

P6-80 程式碼框外出

Chapter 7

P7-10 第 7 行
“複製” 改爲 “複雜”

P7-11 第 12 行
”賴”開啓 改爲 “來” 開啓

P7-12 第 10 行
“builtin” 後加上 ”：”

Chapter 8

P8-6 第 4 行
“那”一種 改爲 “哪”一種

P8-16 第 2 行
“文檔” 改爲 “文字檔”

P8-19 倒數第 10 行
“Save 檔名變數 1 變數 2…” 改爲
“save 檔名 變數 1 變數 2…” (中間是有空格的)

P8-20 第 11 行
“傳會一包含” 改爲 “傳回一包含”

P8-25 第 5 行
“ 如’ ’” 加上 “也就是 dlmwrite(…,’ ’)”

下方表內的所有 分”格”符號 都改爲 分”隔”符號

P8-25 倒數第 1、3、5 與 7 行
“%5n” 改爲 “%5d”

P8-26 第 2 行
”分格符號” 改爲 ”分隔符號”

P8-38 第 14 行
“textscar”中 改爲 “textscan”中

P8-46 第 5 行
“檔案存在” 改爲 “檔案存取”

P8-56 第 2 行
刪除 “方面可以使用 fread”

第 15 行
”是先” 改爲 ”事先”

P8-63 第 15 行
“使用上 C 語言” 改爲 “使用上與 C 語言”

P8-75 第 4、5 與 6 行
“參考點爲於” 改爲 “參考點位於”

P8-77 倒數第 7 行
“繼續網下” 改爲 “繼續往下”

倒數第 4 行
“叛斷” 改爲 “判斷”

Chapter 9

P9-31 倒數第 7 行
"運成功" 改爲 "運算成功"

P9-33 倒數第 9 行
MATLAB6.5 稱爲 MATLAB Excel Builder

P9-42 第 4 行
"MATLAB Conpiler" 改爲 "MATLAB Compiler"

Chapter 10

P10-18 第 11 行
"在於握把式" 改爲 "再於握把式"

P10-20 倒數第 1 行
"圖中的左" 改爲 "圖 10.12 中的左"

P10-30 第 6 行
"set(…,''屬性值')" 改爲 "set(…,'屬性值')" 屬性值外須爲單引號

P10-55 倒數第 10 行
"獨取" 改爲 "讀取"

P10-58 第 6 行
"滑屬" 改爲 "滑鼠"

P10-61 第 15 行
"軟體會 MATLAB" 改爲 "軟體與 MATLAB"

P10-65 第 2 行
"實數落虛" 改爲 "實數或虛"

第 3 行
"矩陣進行" 改爲 "矩陣無法進行"

第 15 行
“GetFullMatrix” 改爲 “PutFullMatrix”

P10-67 第 8 行
“命令視窗的中” 改爲 “命令視窗中”

P10-85 第 6 行“更家活潑” 改爲 “更加活潑”

Chapter 11

P11-11 第 5 行“這告類別” 改爲 “這個類別”