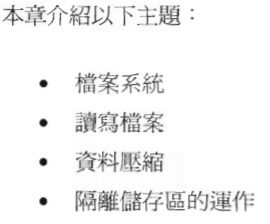
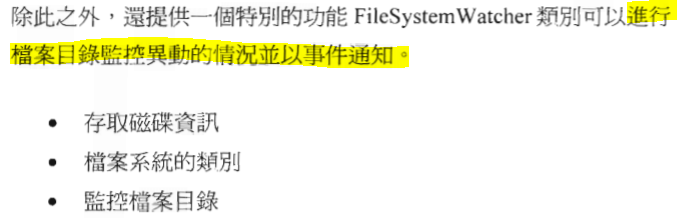
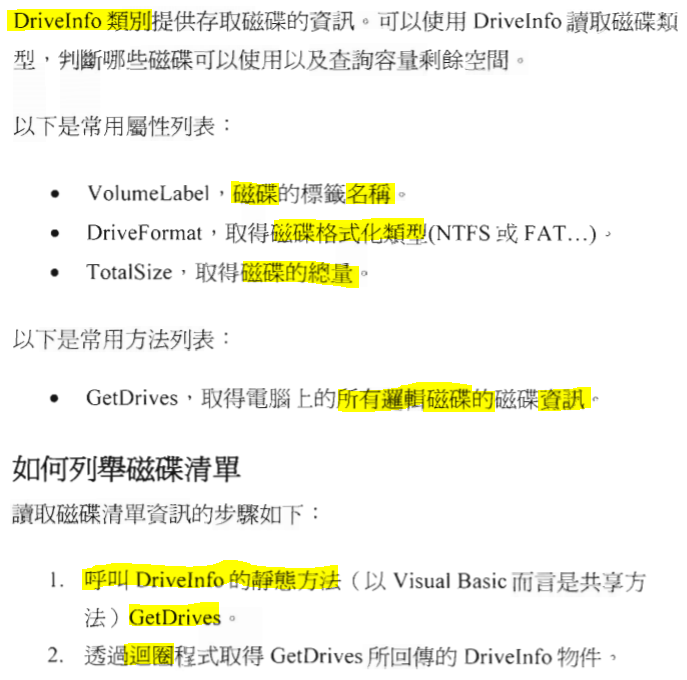
P 61

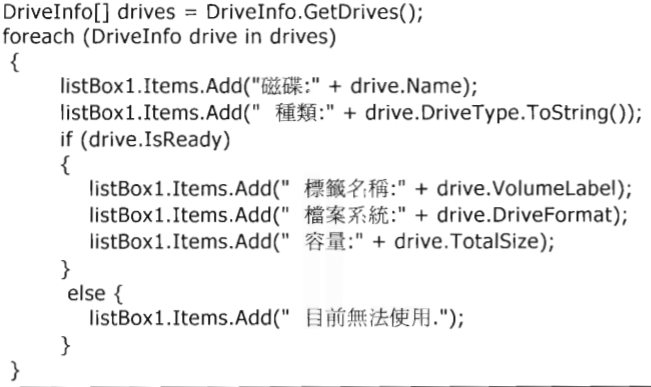


P 64





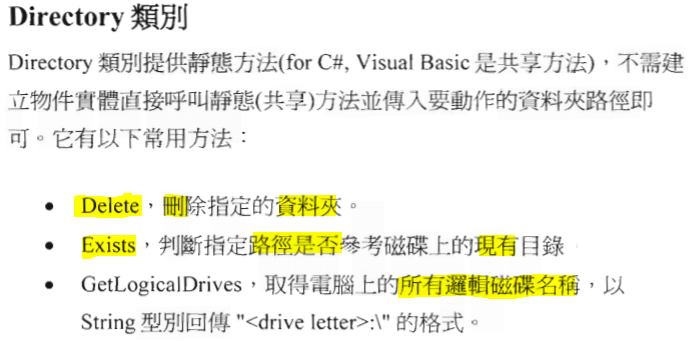
下頁為練習︰



在FileManage/

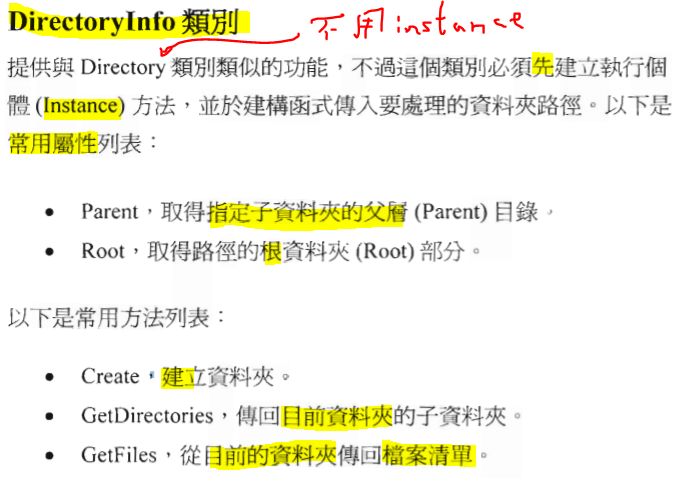
DriveInfo\_ex.aspx

P 67



和 Drive.Name

相似



如何實作？

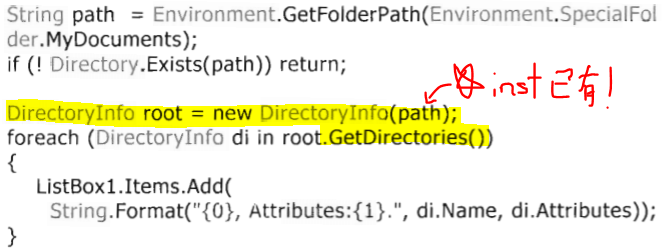
要指定路徑後再

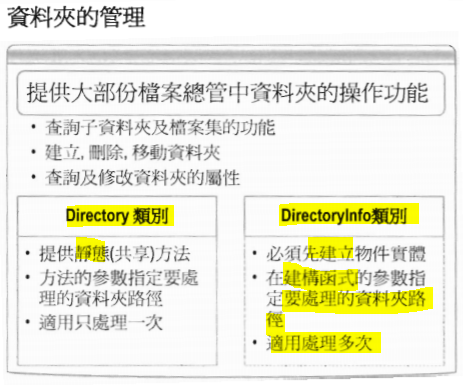
利用這兩個屬性？

見下方，可就這些提到的東東，做更詳細實作。

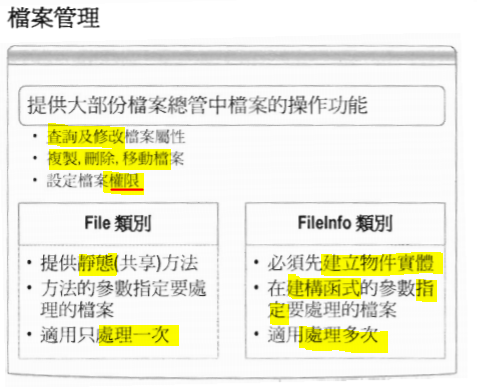
(但我認為沒有實務上的用途？所以略)

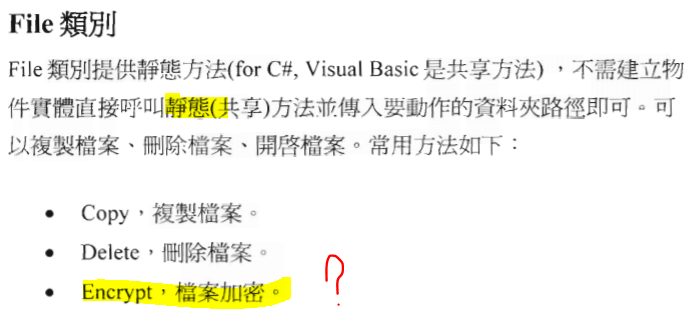
有在 FileSystemWatcher\_ex.aspx 上稍為實作了一下。

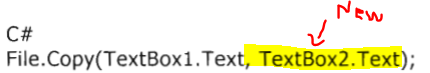


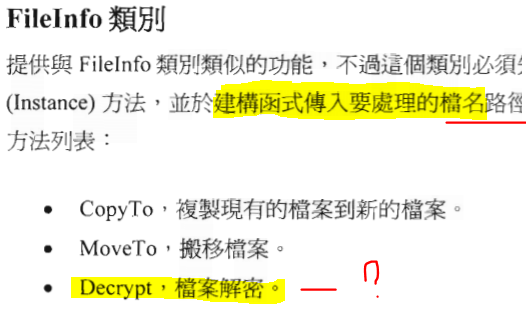
 兩者比較

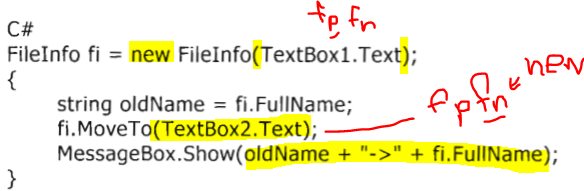
P 69





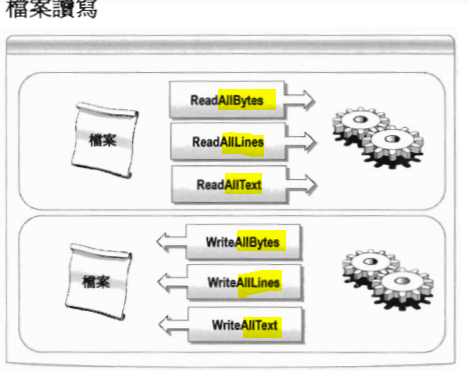






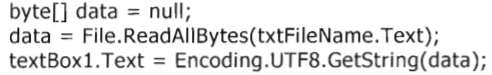
fi.FullName會前後不同！

P 71









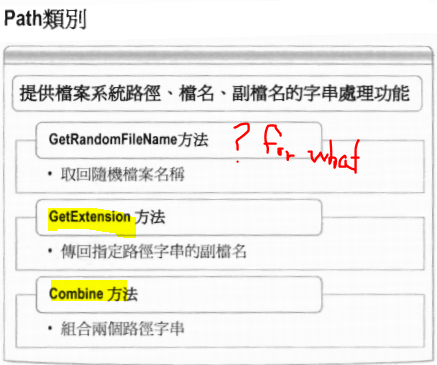


(是 覆蓋)

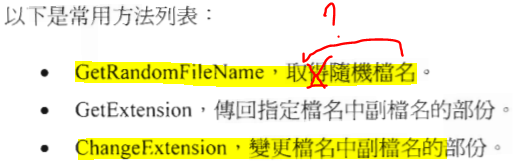


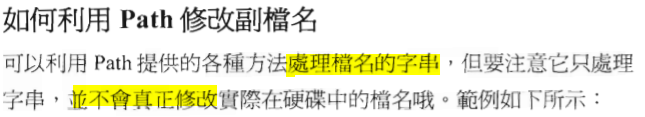
沒有Line 的例子，可能是更不常用吧… (以上我沒有實作。)

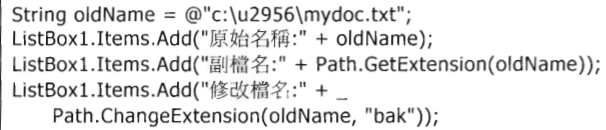
P 73 重點︰只是處理(路徑相關的)字串<- 沒有改真的 路徑/檔名



感想︰莫名奇妙的GetRandomFileName，沒有實用，不打算去查清。







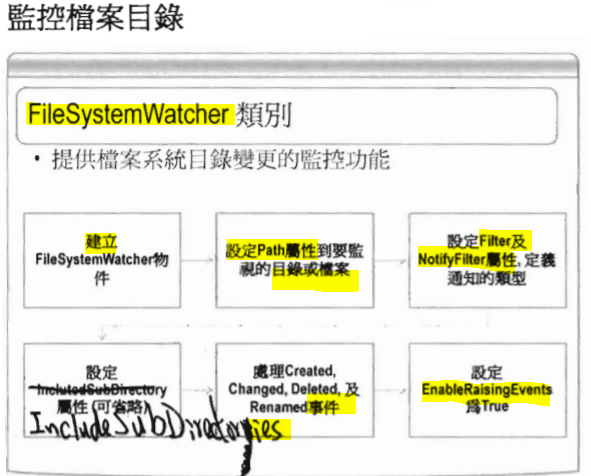
oldName = mydoc.txt

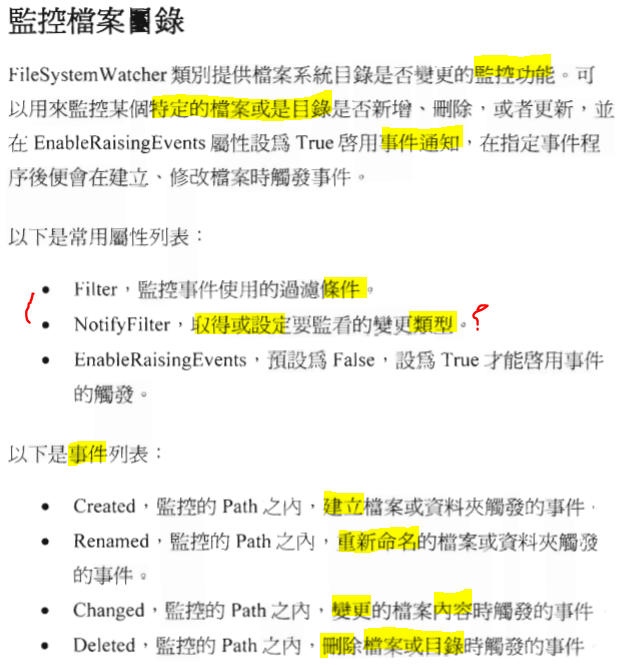
Path.ChangeExtension 只是會感應到檔名字串的 . 後把後方的取代。



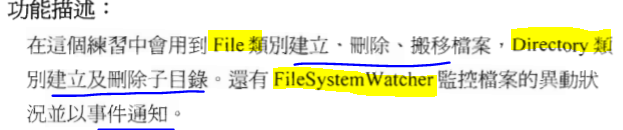
P 75 File練習 File\_ex.aspx

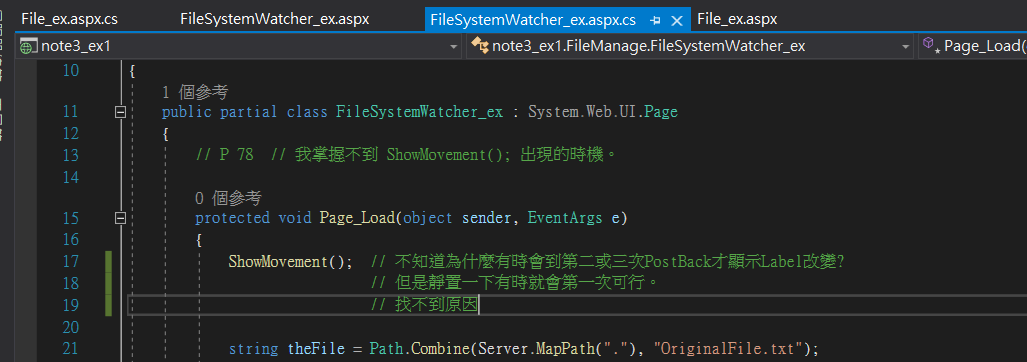
P 77





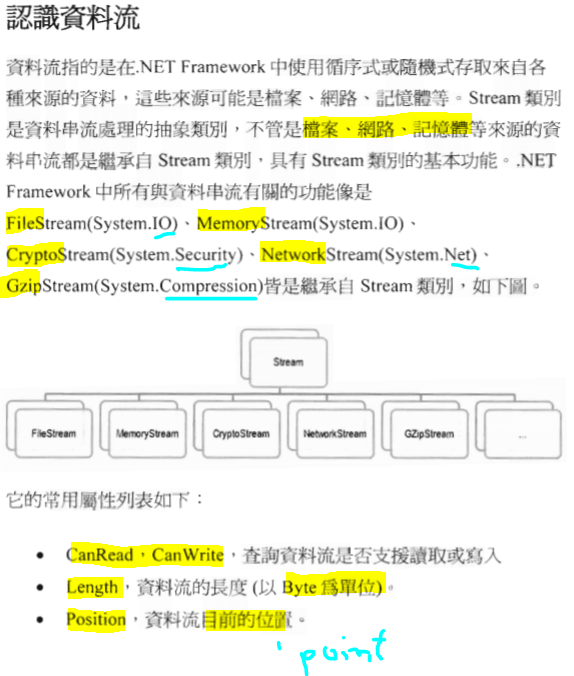
P 78 練習 File , Directory , FileSystemWatcher

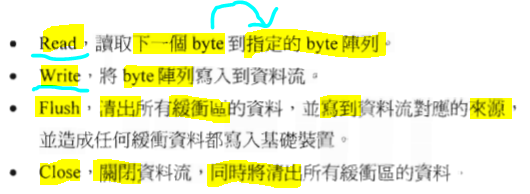


實作結果︰

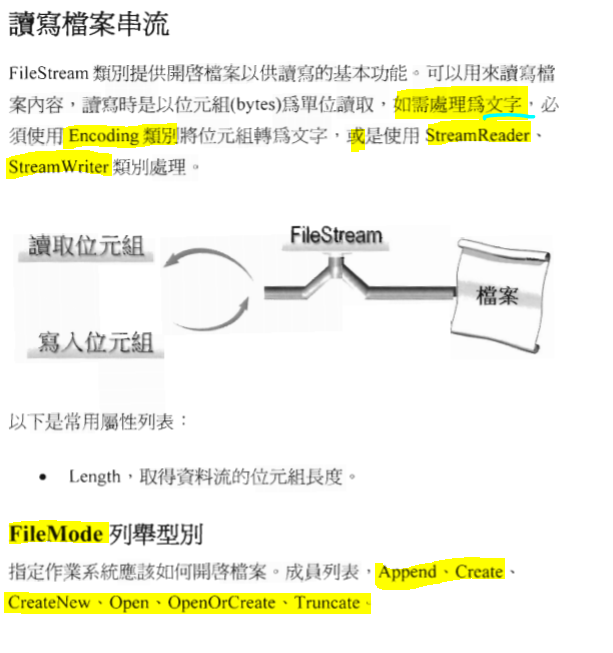
P 82

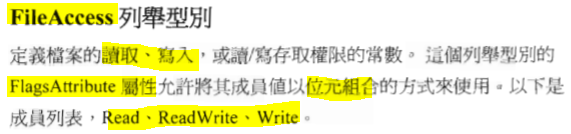




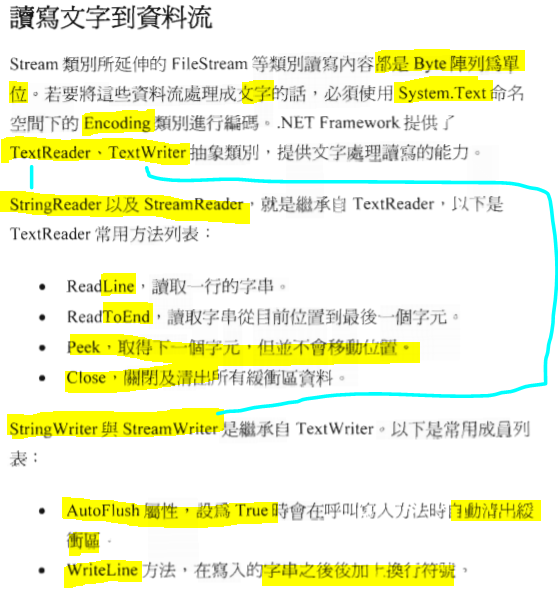


P 85 File Stream





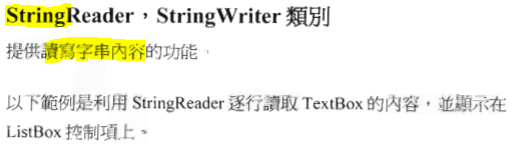
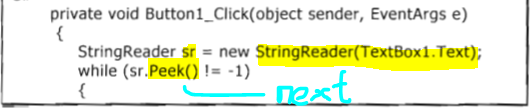
P 87

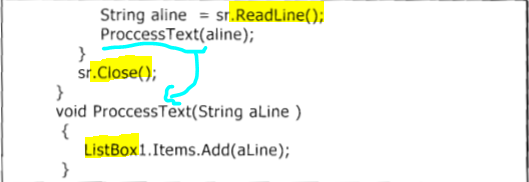


String 和 Stream 要小心看清楚

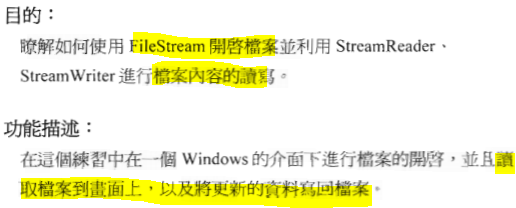
P 88 **Stream / String Read / Write**







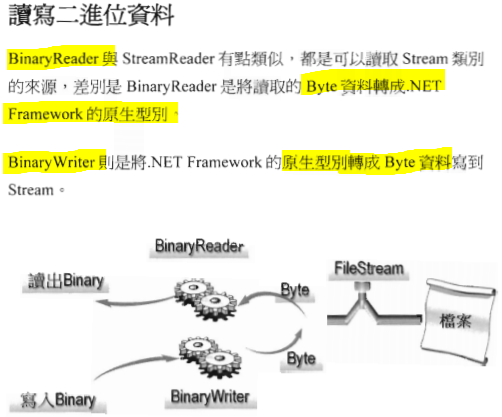
P 90 練習 Read\_Write\_File.aspx







P 94



<https://docs.microsoft.com/zh-tw/dotnet/api/system.io.binaryreader?view=net-5.0>

public static void WriteDefaultValues()

{ using (BinaryWriter writer = new BinaryWriter(File.Open(fileName, FileMode.Create)))

{ writer.Write(1.250F);

writer.Write(@"c:\Temp");

writer.Write(10);

writer.Write(true); } }

public static void DisplayValues()

{ float aspectRatio; string tempDirectory; int autoSaveTime; bool showStatusBar;

if (File.Exists(fileName))

{ using (BinaryReader reader = new BinaryReader(File.Open(fileName, FileMode.Open)))

{ aspectRatio = reader.ReadSingle();

tempDirectory = reader.ReadString();

autoSaveTime = reader.ReadInt32();

showStatusBar = reader.ReadBoolean(); }

Console.WriteLine("Aspect ratio set to: " + aspectRatio);

Console.WriteLine("Temp directory is: " + tempDirectory);

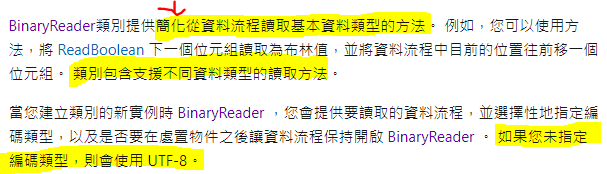
Console.WriteLine("Auto save time set to: " + autoSaveTime);

Console.WriteLine("Show status bar: " + showStatusBar); } }

但是這樣的讀取，不是必須要先知道內容的型別是什麼嗎？

怪怪的。

它的作用，就只是用來**省下內容的 型別轉化 時間**。



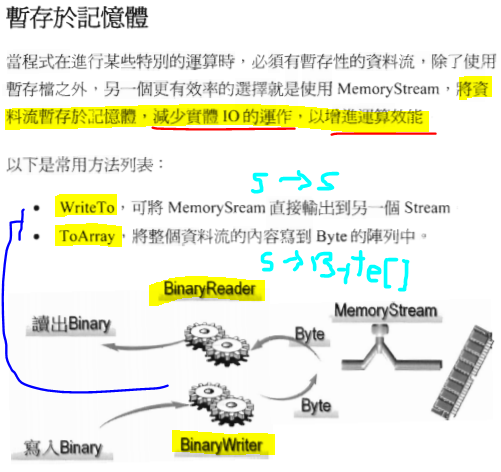
PDF中的例子，是似乎知道那個檔案就是一個Object的化身。

所以這樣的(每次相同順序的)寫進和讀取就可以用於迴圈

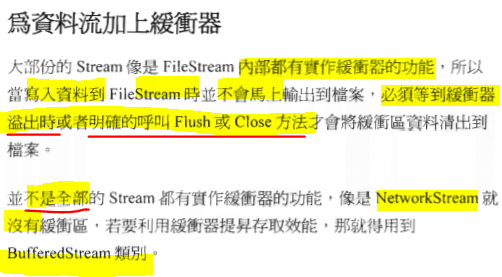


上頁那個BinaryReader( FileOpen( …) ) 是個 Stream

P 97 MemoryStream



P 98



暫存是為了減少「進出」的次數，加快效能。

出發的條件是︰ 滿 / 呼叫 flush / 呼叫 Close

未有「預設」這個功能的 Stream 要自己利用 BufferStream 類別



我認為newStream是MemoryStream ? 不，是Stream

寫到buffer後，是再從 buffer 寫到NetworkStream中？ X , 見下頁實作

P 99 實作 BufferStream\_ex.aspx



實作後感覺︰

只是因為 WeRequest 和 Response 都要求一個Stream來裝內容，

所以它先向Memory 申請出一個Buffer，

把資料在裡面放好，再送出 / 回收處理。

P 103

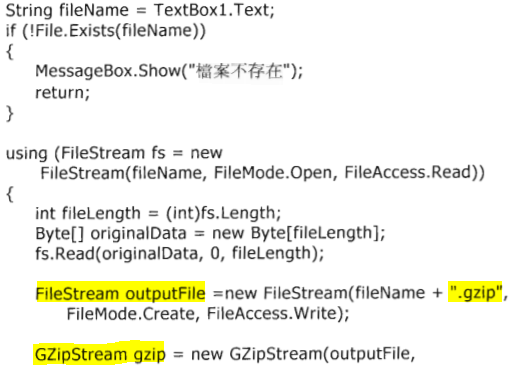


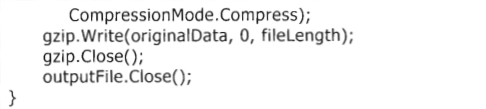
資料量大而用於網路傳輸

不能大於 4 GB

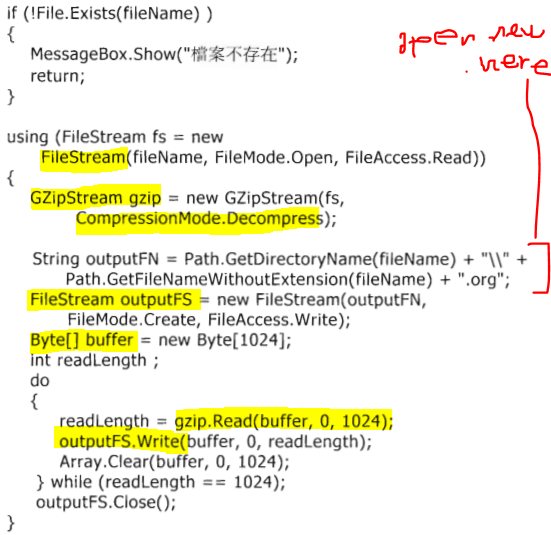


存成一個壓縮檔





壓縮為新檔



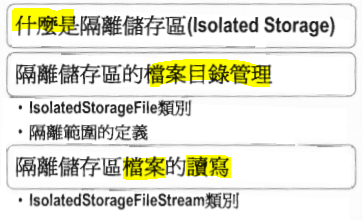
上一種只看不做，DeflateStream就實作 : P 107 見 DeflateStream\_ex.aspx

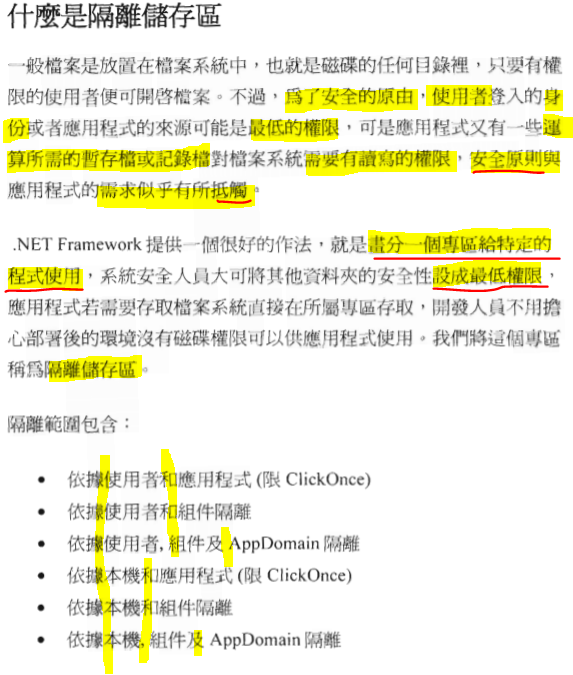
小結︰

StringReader/Writer 只能用作進階文字編輯工具

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stream   |  | | --- | | CanRead  CanWrite  Length  Position  -----  Read  Write  Flush  Close | | System.IO  FileStream   |  | | --- | | FileMode :  Append  CreateNew  Create  Open  OpenOrCreate  Turncate  FileAccess :  Read  ReadWrite  Write | | 文字  System.  Text.  Encoding | TextReader   |  | | --- | | ReadLine  ReadToEnd  Peek  Close | | StringReader  (文字->文字)  StreamReader  (byte->文字) | 所以檔案管理是用 |
| TextWriter   |  | | --- | | AutoFlush  WriteLine | | StringWriter  (文字->文字)  StreamWriter  (文字->byte) |
| 原生資料型別 | BinaryReader | ReadSingle  ReadString  ReadInt32  ReadBoolean |  |
| BinaryWriter | Write |  |
| System.IO  MemoryStream |  |  | WriteTo  ToArray |  |
| System.IO  BufferStream |  |  | Write  Close  Read |  |
| System.Compression  GzipStream |  |  |  |  |
| DeflateStream |  |  |  |  |
| System.Security  CrytoStream |  |  |  |  |
| System.Net  NetworkStream |  |  |  |  |

P 112



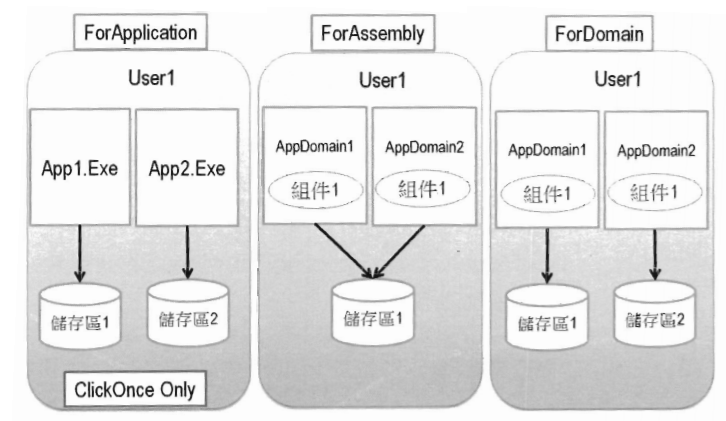


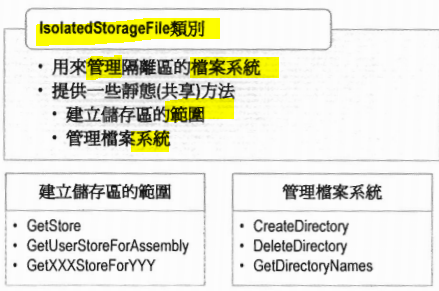


依據使用者/本機？

PDF所提的系統已過時。現在的路徑？

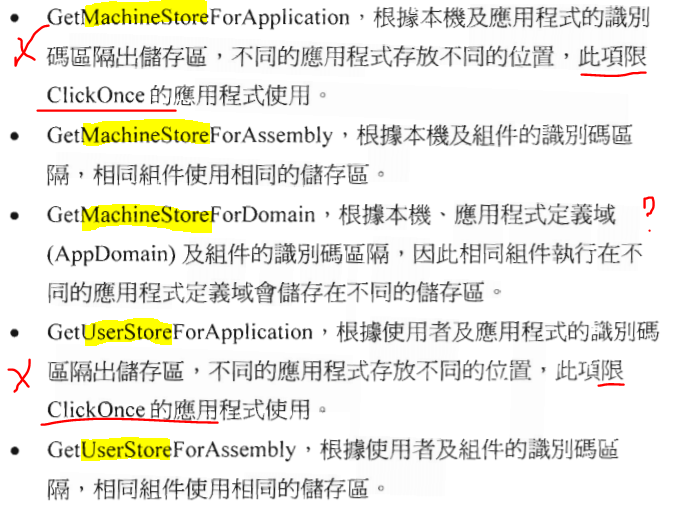
以下是隔離方式︰ P 114

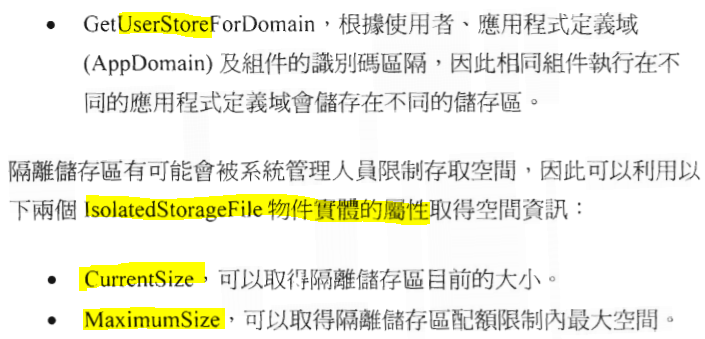


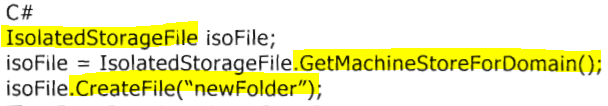
P 116

其實不太清楚…

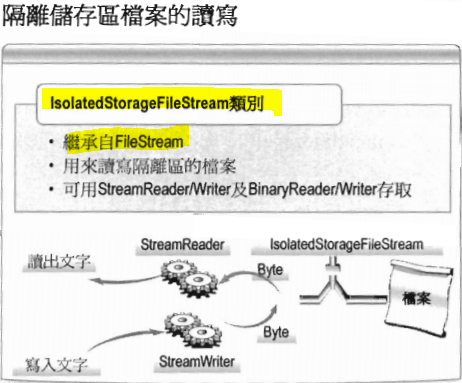
在說什麼？

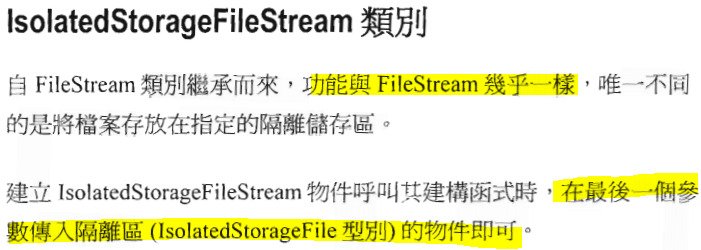


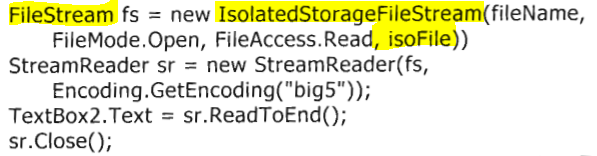




P 118







P 119 練習

<https://docs.microsoft.com/zh-tw/dotnet/standard/io/how-to-read-and-write-to-files-in-isolated-storage>看官網的例子

<https://dotblogs.com.tw/yc421206/2009/01/18/6873>這裡不止增改，還有刪 :

private void button3\_Click(object sender, EventArgs e)

{//為目前使用者建立一個隔離區

IsolatedStorageFile userStorage = IsolatedStorageFile.GetUserStoreForAssembly();

//判斷檔案是否存在

string[] files = userStorage.GetFileNames("Profile");

if (files.Length == 0)

{Console.WriteLine("找不到檔案");}

else{

//刪除隔離區的檔案

userStorage.DeleteFile("Profile");

//釋放資源

userStorage.Dispose();}}

我不知道是不是一定要.Dispose但是我實作的例子和官網都沒有提到這一句。

我去找隔離區(在我的電腦)的實體地址︰

C:\ProgramData\IsolatedStorage

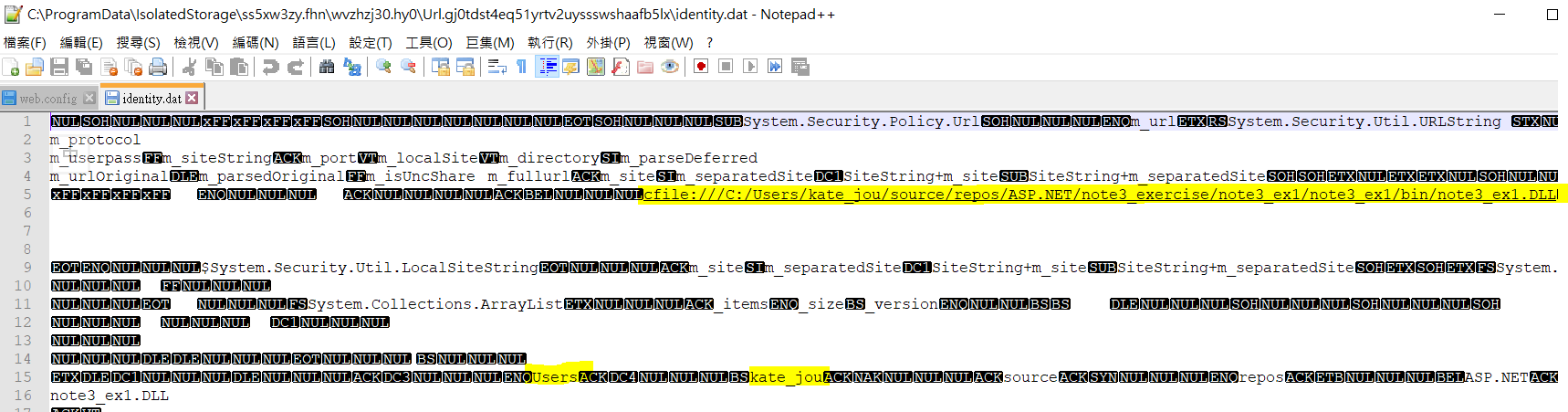
從這裡之後就是一堆亂碼的資料夾和檔案

C:\ProgramData\IsolatedStorage\

ss5xw3zy.fhn\wvzhzj30.hy0\

Url.gj0tdst4eq51yrtv2uyssswshaafb5lx

其中好像有些關鍵字，但是我找不到的程式建立的那個檔案︰



設定(細)權限的方法？(如by Application / by XXX 見上面的很多分類)

筆記沒有再提，我目前也不想去深究。(因為不知道會不會用得到…)

小結

StringReader/Writer 只能用作進階文字編輯工具

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Stream   |  | | --- | | CanRead  CanWrite  Length  Position  -----  Read  Write  Flush  Close | | System.IO  FileStream   |  | | --- | | FileMode :  Append  CreateNew  Create  Open  OpenOrCreate  Turncate  FileAccess :  Read  ReadWrite  Write | | 文字  System.  Text.  Encoding | TextReader   |  | | --- | | ReadLine  ReadToEnd  Peek  Close | | StringReader  (文字->文字)  StreamReader  (byte->文字) | 所以檔案管理是用 |
| TextWriter   |  | | --- | | AutoFlush  WriteLine | | StringWriter  (文字->文字)  StreamWriter  (文字->byte) |
| 原生資料型別 | BinaryReader | ReadSingle  ReadString  ReadInt32  ReadBoolean |  |
| BinaryWriter | Write |  |
| System.IO  MemoryStream |  |  | WriteTo  ToArray |  |
| System.IO  BufferStream |  |  | Write  Close  Read |  |
| System.Compression  GzipStream |  |  | 只有圖 |  |
| DeflateStream |  |  | 有實作 |  |
| System.Security  CrytoStream |  |  | 沒細說 |  |
| System.Net  NetworkStream |  |  | 有在Buffer時說一下 |  |

P 123 完結