ASP.NET 第 5章

*第五章 : 追蹤與除錯 P 157*

(學到第十二章的我回來補充︰

我想這章的功能，是在不能用F9中斷點的時候，才會用得上。

比如說，是掛到iis上之後，或者程式寫到了前端，而不是後端的場合。

有那些iis Express (F5)和掛到iis之間不同的地方，做成的bug。

雖然很少見，但不是沒有。)

**應用程式中常見錯誤**

Complie Time Error 編譯時期錯誤 (語法錯)

Runtime Error 執行時期錯誤(try catch exception 可找到的錯 )

Logidcal Error 邏輯錯誤 (🡨 因為自已的算式產生錯誤結果)

第三種就需要這個 Debug , 這是除了F9 以外，看到 變數值的方法。

**Visual Studio 除錯種類**

* 伺服器端程式除錯

1. Complie Time Error

Error List 視窗, 點項目兩下可跳到那個位置

(檢視>錯誤清單)

2. Runtime Error

產生錯誤的網頁, 同時表明該 原因 以及 地點

可配合利用中斷點 和 Trace? 物件的撰寫 (3-4頁交界)

3. Logidcal Error

> Output Window

顯示執行狀態 / Debug.Write的指令輸出

> Locals Window

顯示目前中斷點的所有區域變數資料

> Watch Window

顯示目前所監看的物件/變數狀態

(需事先加入要監看的物件/變數) ??? How ???

> Call Stack Window

顯示目前記憶體堆疊的配置狀態

> BreakPoints Window

可以管理網站所有的中斷點設定

*這些窗戶在哪裡？ :*

*工具 > 選項 > 偵錯 > 一般 > 勾 偵錯時啟用診斷工具*

*視窗 > 重設視窗配置 或 偵錯 > 視窗*

*在程式中隨意設下一個中斷點 F9，那麼在F5 時，就可以用F10觀看。*

*P 166 實作中斷點 ASP.NET(EX5.1)*

* **用戶端指令碼程式除錯**

和上則不同的，只是針對前端 .aspx 檔下中斷點

IE要啟用指令碼偵錯功能, Chrome不需要做任何東西

IE步驟︰

IE > 工具 (右上角的齒輪)

網際網路選項 >

進階 > 瀏覽 > 不勾︰

停用指令碼偵錯(Internet Explorer)

停用指令碼偵錯(其他)

### 要重新開機才生效 ###

共同步驟︰

1. 在 .aspx 檔決定中斷點
2. 在Solution Explorer 的視窗中，找到要除錯的那頁︰

右鍵 > Set As Start Page

1. F5

*\* 可用 EX3.4測試*

* 應用程式追踨除錯 P171

利用應用程式，追踨執行過程的所有狀態記錄。

見下方的兩個大標題︰

*應用程式追踨除錯 1 .NET Framework 應用程式追蹤*

適用於所有.NET開發出來的程式

包括ASP.NET。

內建輸出目標如︰Output Window(預設), EventLog,

也可以自訂輸出到什麼地方。

**Debug物件**

指令︰Write, WriteLine , WriteLineIf(bool,string)

*P175 使用 Debug 物件 實作︰ ASP.NET(EX5.2)*

工具列中，沒有「Solution Configuration下拉選單」?

解決的方法︰

書簽頁的上方有一個下拉的選單，可選︰

Debug / Release / 組態管理員( = Solution Configuration)

選 Debug (🡨預設，不用動。) 或

在建置>方案組態管理員>使用中的方穼組態: Debug + 專案內容:組態:Debug

按F5 > 跑出頁面。

Visual Studio 上方選單︰ 檢視 > 輸出 :

下方會跳出一個窗戶，把它拉高。

網頁中輸入什麼，再點按鈕，Visual Studio輸出的頁面會出︰

Data = 自己輸入的東東。

**Debug物件 指定 Listener**

暫略:

<https://docs.microsoft.com/zh-tw/dotnet/api/system.diagnostics.debug.listeners?view=netframework-4.8>

<https://iter01.com/103973.html>

<https://dotblogs.com.tw/yc421206/2010/10/15/18379>

<https://docs.microsoft.com/zh-tw/visualstudio/debugger/watch-and-quickwatch-windows?view=vs-2019>

*應用程式追踨除錯2 網頁追蹤 Trace物件 和 層級 P177*

應用程式追踨除錯的一種。

預設輸出對像只有到網頁上。

但較專門和方便，不需自行選寫程式。

* Trace物件

指令︰Write = 輸出黑字 和 Warn = 輸出紅字

例子︰( 第一個參數的是自設分類 , 第二個是自設內容。 )

Trace.Write(“Test message”,”Trace.Write”);

Trace.Warn(“Test message”,”Trace.Warn”);

* 層級

網頁(Page)﹑網站(Web.Config)﹑自訂輸出。

**(頁面層級 = )網頁層級追蹤**

追蹤單一頁

方法︰

<%@ Page Trace = “True” %> 加入Trace 屬性，預設FALSE。

再按F5

結果會自動的出現在網頁的下方，資料分類詳見 P 180 - 181

(應用程式追踨 🡨 不打算實作。

也可以以 Request.XXX (如 Server Variables) 取得資料分類資訊。)

可以設定在網頁的下方輸出時用 資料分類 的排序方式︰

<%@ Page Trace = “True” TraceMode="SortByCategory" %>

SortByTime = 以時間排列 , SortByCategory = 以類別排列

(我實作時，好像看不出有什麼差別，可能是我看不懂吧…)

*實作 ASP.NET(EX5.3)*

**(應用程式 = ) 網站層級 P181**

輸出目標可以是網頁的內容下方或是網站的Cache等

需針對Web.Config來設定

在每個網頁的Web.config檔中加入藍色的東東︰

<configuration>

<system.web>

<trace enabled="true"

pageOutput="true" 是否在網頁的內容下方顯示(不然…見下方)

localOnly="true" 只能在本機瀏覽，不可以給客戶端看見

requestLimit="10" 報告的儲存數量

traceMode="SortByCategory" 排序方法

mostRecent="true" 以最新版本為主

writeToDiagnosticsTrace="true"/> 輸出一份到？(見下方)

</system.web>

<configuration>

( 和Trace = “True”的效果一樣。)

* pageOutput 如false :

報告就不會出現在網頁下方，而在客戶端RAM中。

只要在客戶端網址列，改最後一節的檔名為︰Trace.axd

即可查看。

在Trace.axd 中，可點 View Details 來瀏覽每一份報告。

Remaining = requestLimit 的餘數, 可點clear current trace 清除

* writeToDiagnosticsTrace 的報告寫到哪裡去︰

<http://hk.uwenku.com/question/p-slkdngdy-bnn.html>

*實作 ASP.NET(EX5.4)*

*P183 網頁層級 + Trace物件 + 網站層級 實作 ASP.NET(EX5.5)*

詳情寫在該實作中了。

**自訂輸出 P186**

1. 自行訂閱 Trace 物件的TraceFinished事件

取TraceContextEventArgs 參數內的TraceRecord 集合成員。

1. 自行決定Trace輸出的方式
2. 需搭配網頁/站層級的設定(page/web.config)

最好是web.config的trace, 屬性 :

pageOutput = false , mostRecent = true

有用的總結和參考︰

<https://dotblogs.azurewebsites.net/yc421206/2009/04/27/8195>

*自創實作︰ ASP.NET (EX5.6)*

詳見實作中的說明。