

# 資訊民國百年問題的因應與思考

## 壹、前言

人過百歲，我們稱他「人瑞」，是件喜事；一個系統要過民國百年，還真引發一些麻煩。

在資訊應用系統의 日期欄位的規劃與設計上，當然是以國人環境因素考量，自然以民國年份顯示。有些較早開發的系統的民國年欄位只設定 2 位數，一旦到了民國 100 年，依照程式文數字右靠的慣例，2 位數的民國年欄位就呈現「00」；如遇有年份計算公式，譬如民國 100 年有人剛好申請退休，其退休年資一經計算（00 年－70 年＝－70 年），再乘以退休基數，依會計常識，負數的金額代表負債；那麼此人非但領不到退休金，還欠款呢！此即所謂的「資訊民國百年問題」。

## 貳、釋疑

民國百年問題只發生在資訊系統或一些可能的辦公室軟體(如 Word)，它不像 Y2K(千禧年危機)涉及硬體設計、操作系統 OS、程式語言、與業務資訊應用系統，甚至發生於嵌入式設備。當年記憶體技術有限，價格昂貴，為節省記憶體空間的使用，也是採 2 位數表示，譬如以 98 取代 1998 年。

百年問題比千禧年的問題簡單許多，衝擊性亦輕，但卻也不容等閒視之。因為民國百年問題只發生在我國，此之所以個人僅稱之「問題」，不稱「危機」的原因。

記得個人在 Y2K 的宣導上及「Y2K 教戰手冊」，亦曾一併呼籲「在修改 Y2K 程式的同時，一併處理民國百年 3 位數，竟其功於一役」；只是，事過境遷，人事更迭，資訊界近來新進青年才俊輩出，或許未逢 2000 年時 Y2K 全世界總動員的盛況，在電腦年序可能的問題上沒有特別留意欄位長度的陷阱，這幾年來系統開發設計時或有可能仍習慣性地使用 2 位數。

民國年欄位資訊最常被利用於資料的儲存(如出生欄位)、計算(如程式中的公式)、顯示(如螢幕)、印表(如證明單、收費單...等)，何況政府資訊具有公文書特質，非正確無誤不可。是以，釜底抽薪之計仍然不可輕忽，應作及早因應。

## 參、因應步驟

資訊技術演進神速，2000 年時，應用系統運作平台仍多屬傳統主機型態，使用的程式語言大多集中於 COBOL、FORTRAN 等，因應方法單純；而今日歷經 Client-Server、Web-Base、Web2.0 等規劃或設計風潮的演化，使用的軟體語言工具很多，各機關必須針對個別軟體環境分類，搜集所使用的各種軟體語言工具的日期年份的習慣性程式寫法，找出關鍵字詞，彙成關鍵字詞庫，以之當 key 進行原始碼的搜尋，找出有民國年的欄位，然後修正、測試和取代舊程式。

其步驟列表如下：

步驟	工作內容
一、清查	1.原碼程式檔分 2 類清查：(1)需編譯為執行檔者、和(2)直譯者 2.按「各資訊系統原始碼清查表」全面清查
二、分類	1.將各資訊系統原始碼依程式語言工具歸類 2.搜集有關民國年的各資訊系統，及該程式語言工具寫法，擬出關鍵字詞(含運算的公式) 3.彙總各資訊系統關鍵字詞成關鍵字詞檔 4.可參考附表一「各程式語言之日期關鍵字詞一覽表」
三、搜尋	1.逐步以一個資訊系統的原碼檔為對象，依關鍵字詞搜尋有無民國年的定義或公式
四、修正	1.執行修正前，務必先作好原始程式碼完整備份及擬妥回復程序；避免修正或測試失敗 2.執行修正與測試的日期宜選在假日；避免失敗而影響正常作業 3.team work 狀況下，宜妥善分配 team member 負責修正之程式群，避免重覆修正，免除版本控制之問題 4.修正原始程式碼，並應記錄 5.若原碼程式檔非直譯語言，需重新編譯執行檔至原執行檔所在之目錄夾
五、測試	1.能有測試套系統最好；(與正式套系統分開)就不必在假日為之 2.檢視相關的儲存、計算、顯示、印表…等功能是否無誤
六、正式上線	1.確認測試成功後，上線至正式系統，完成原始程式碼修正作業

另應用系統中用來儲存資料之資料庫，亦需一併處理民國百年問題，先確認儲存「日期」所採用之資料類型為何，及欄位長度是否足夠，配合擴充欄位長度。

依其資料欄位儲存類型，概述其處理方式列表如下：

資料欄位儲存類型	處理方式
西元年方式	● 無民國百年問題，不需變動
字元字串	● 如為日期欄位，原長度為 6 位，遇民國百年時會遭截斷，需將長度擴充為 7 位以上。 ● 如為年度欄位，原長度為 2 位，同理，需將長度擴充為 3 位以上。
數值	● 資料型別設定為 int，欄位長度不需擴充。 ● 因為 int 值域範圍為 $-2^{31}$ (-2,147,483,648) 到 $2^{31}-1$ (2,147,483,647)，可以容納 7 位長度的值。

因應步驟中，常見較繁瑣或困難的問題如下：

- 一、搜尋難度：目前尚無搜索工具可用，如何有系統地搜尋，端賴資訊同仁的思考。個人建議仿 windows 檔案搜尋方式；先設定 Source 置放之程式目錄夾為尋找範圍，再以 keyword 一一搜尋；或建議寫一簡單程式，以關鍵字詞檔表列搜尋產出對應的原碼檔名和關鍵字，再從原碼檔名逐檔修正。或如附表二使用 UltraEdit 搜尋關鍵字詞。
- 二、版本落差：現行原始碼已經遺失，或經過了多次修改的維護版本已經沒有記錄了；這是最頭痛的風險。或可由儲存、計算、顯示、印表…等功能逆推回去。
- 三、完全委外：本身資訊人員只辦理委外業務，幾乎不接觸程式或內容。建議仿版本落差方式，與業務同仁多加思考可能的民國年出現的資訊作業，與委外廠商確認；或只好再專案委外辦理。此外，與各委外的廠商連繫，瞭解該系統的民國年設計模式，以為奧援。

#### 肆、工作時程規劃原則

從因應步驟中可以體認，因應民國百年問題必須業務與資訊兩個單位協力合作，更需長官支持。個人認為仍應(一)成立團隊，(二)決定清倉重點優先序，(三)成員中最好有熟悉系統、程式和業務者，(四)規劃適當時程，及早因應。

工作時程之規劃，視業務系統之多寡與繁複、業務影響大小等因素，建議時程可按上述因應步驟採日期倒推法規劃。

時程	完成項目
98.01~98.06	組成團隊、宣導
98.07~98.12	全面清查、搜尋完成
99.01~99.06	改正、測試完成
99.07~99.10	取代完成，應付失漏或突發狀況

#### 伍、結論

基於國情，政府機關提供民眾的資訊係採民國記年，所以民國百年問題只發生在國內，國際資訊業者不會關注這個課題；又政府資訊多少具有公文書特質，必須為民眾所信賴，自然必須正確無誤。

是以，必須由政府各機關及早正視，及早動員，及早謀求改正為宜。

#### 陸、附錄

【附表一】 各程式語言之日期關鍵字詞一覽表

程式語言	與日期有關的語法	搜尋之關鍵字
任何語言	常見以「民國年=西元現年-1911」計算公式 (任何程式語言均採用)	搜尋「1911」
JAVA、JSP	<ol style="list-style-type: none"> <li>Date date1 = new Date();</li> <li>SimpleDateFormat dateFormat = new SimpleDateFormat("yyyyMMddHHmmss");</li> <li>Calendar date = Calendar.getInstance(TimeZone.getTimeZone("GMT+8"));</li> <li>GregorianCalendar now = new GregorianCalendar();</li> </ol>	<ol style="list-style-type: none"> <li>Date</li> <li>SimpleDateFormat</li> <li>Calendar</li> <li>GregorianCalendar</li> </ol>
JavaScript	<ol style="list-style-type: none"> <li>var today=new Date();</li> </ol>	<ol style="list-style-type: none"> <li>Date</li> </ol>
VBScript	<ol style="list-style-type: none"> <li>Dim myDateString myDateString = Date() d="April 22, 2001" if IsDate(d) then document.write(CDate(d)) end if</li> <li>document.write(Now)</li> </ol>	<ol style="list-style-type: none"> <li>Date</li> <li>IsDate 或 CDate</li> <li>Now</li> </ol>
ASP	<ol style="list-style-type: none"> <li>tname1 = "pc_res_"+cstr(year(date()-1911-2))</li> <li>writetime = year(now) &amp; "/" &amp; month(now) &amp; "/" &amp; day(now) &amp; " " &amp; hour(now) &amp; ":" &amp; minute(now) &amp; ":" &amp; second(now)</li> <li>CDate(now()+2)</li> <li>IsDate("December 31,1999")</li> </ol>	<ol style="list-style-type: none"> <li>Date</li> <li>now</li> <li>CDate</li> <li>IsDate</li> </ol>
VB	<ol style="list-style-type: none"> <li>Dim dtmTest As Date dtmTest = DateValue(Now)</li> <li>Dim MyDate MyDate = Date</li> <li>Dim YourDate, NoDate, MyCheck YourDate = #2/12/69#: NoDate = "Hello" MyCheck = IsDate(MyDate) ' 傳回 True MyCheck = IsDate(NoDate) ' 傳回 False</li> </ol>	<ol style="list-style-type: none"> <li>DateValue 或 Now</li> <li>Date</li> <li>IsDate</li> </ol>
ASP.NET 之 C#	<ol style="list-style-type: none"> <li>DateTime d1 = DateTime.Now;</li> <li>string date = "01/08/2008"; DateTime dt = Convert.ToDateTime(date);</li> </ol>	<ol style="list-style-type: none"> <li>DateTime 或 Now</li> <li>ToDateTime</li> </ol>
ASP.NET 之 VB.NET	<ol style="list-style-type: none"> <li>Dim d1 As DateTime = DateTime.Now</li> <li>Dim convertedDate As Date convertedDate = Convert.ToDateTime(value)</li> </ol>	<ol style="list-style-type: none"> <li>DateTime 或 Now</li> <li>ToDateTime</li> </ol>
ASP.NET 之 C++	<ol style="list-style-type: none"> <li>DateTime d1 = DateTime::Now;</li> </ol>	<ol style="list-style-type: none"> <li>DateTime 或 Now</li> </ol>
ASP.NET 之 J#	<ol style="list-style-type: none"> <li>DateTime d1 = DateTime.get_Now();</li> </ol>	<ol style="list-style-type: none"> <li>DateTime 或 Now</li> </ol>
Foxpro	<ol style="list-style-type: none"> <li>DATE() 求當前日期 ex: 2009-02-03</li> <li>DATETIME() 求當前日期和時間 ex: 2009-02-03 0 : 22 : 33am</li> </ol>	<ol style="list-style-type: none"> <li>DATE</li> <li>DATETIME</li> </ol>

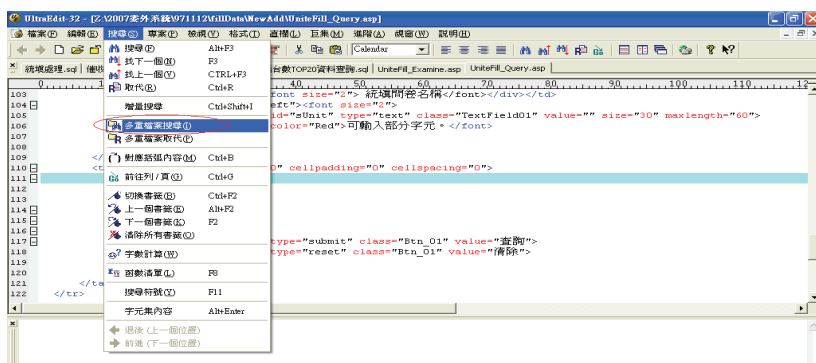
DELPHI	<pre> 1. lSendTime.Caption := DateTimeToStr(Now); 2. slReplyContent.Add(CurUserID + ' + CurUserName + ' 於 ' +    FormatDateTime('yyyy/mm/dd hh:nn:ss', Now) + ' 的回覆內容:'); 3. var d1, d2: TDate;    begin      d1 := StrToDate('09/01/2000');      d2 := StrToDate('09/01/2001');      d3 := round(d2-d1);      Showmessage(IntToStr(d3));    end; 4. var aa,bb,cc:word;    tt:tdatetime;    decodedate(now,aa,bb,cc);    tt:=encodedate(aa,bb,cc); 5. now,date,time : 現在的日期時間    Label1.Caption := 'Today is ' + DateToStr(Date);    Label1.Caption := DateTimeToStr(Now);    Caption := TimeToStr(Time); 6. if Chart1.BottomAxis.IsDateTime then... </pre>	<ol style="list-style-type: none"> <li>1. DateTime 或 Now</li> <li>2. FormatDateTime 或 Now</li> <li>3. TDate 或 StrToDate</li> <li>4. tdatetime 或 decodedate 或 Encodedate</li> <li>5. now 或 date 或 time</li> <li>6. IsDateTime</li> </ol>
--------	---	---

**【附表二】使用 UltraEdit 搜尋關鍵字詞說明**

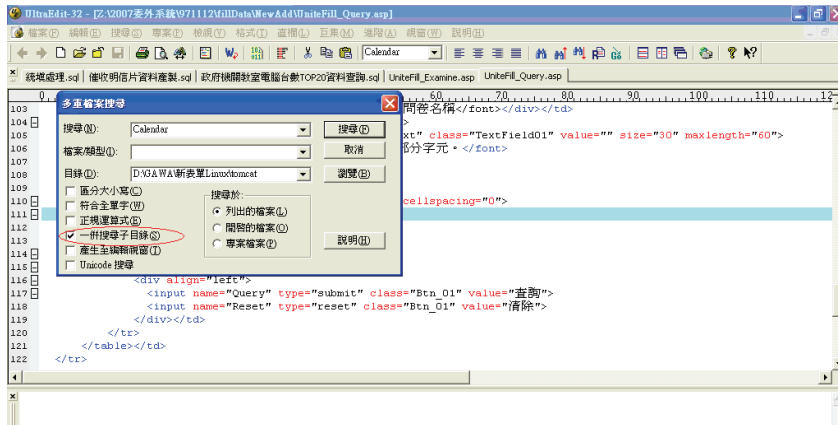
文書編輯器 UltraEdit 簡介：

1. 提供超強的文字檔或任何原始程式檔編輯、預視、列印功能。
2. 提供直接編輯十六進位碼功能。
3. 可編輯 HTML 檔案，以彩色顯示 HTML 標記，方便網頁編輯。
4. 可同時開啓多個檔案編輯。
5. 提供編譯 java 原始程式檔功能。
6. 提供多重檔案搜尋及多重檔案取代，一併搜尋目錄夾下所有子目錄夾。
7. 非免費軟體。

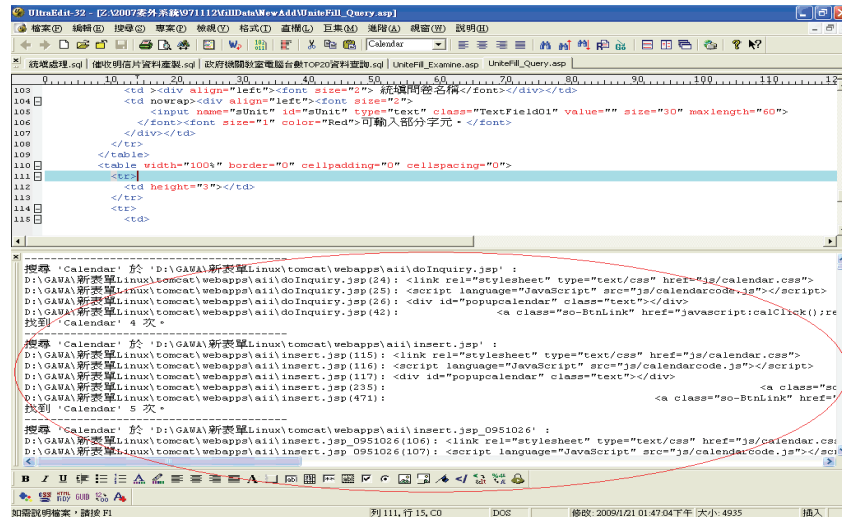
Step1、開啓 UltraEdit 工具，選擇「檔案」－「多重檔案搜尋」。



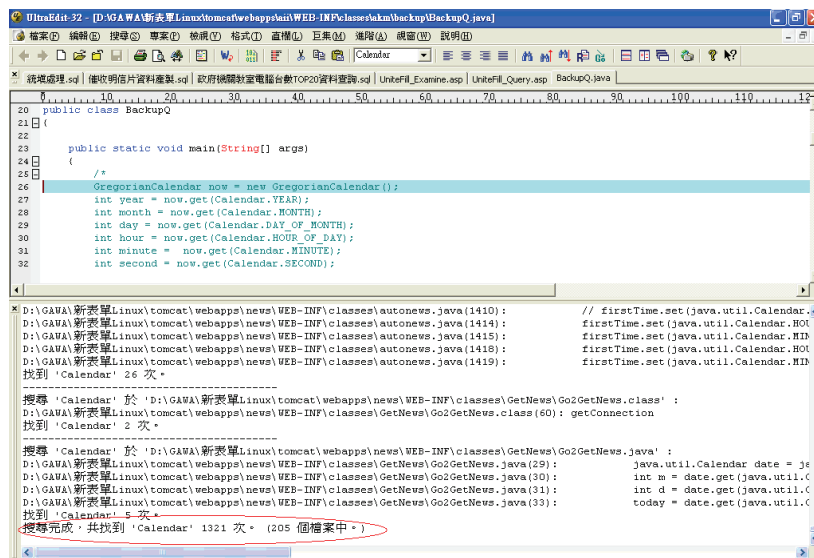
Step2、輸入搜尋關鍵字，如：「Calendar」，選擇 source 置放之程式目錄夾，按「搜尋」按鈕。



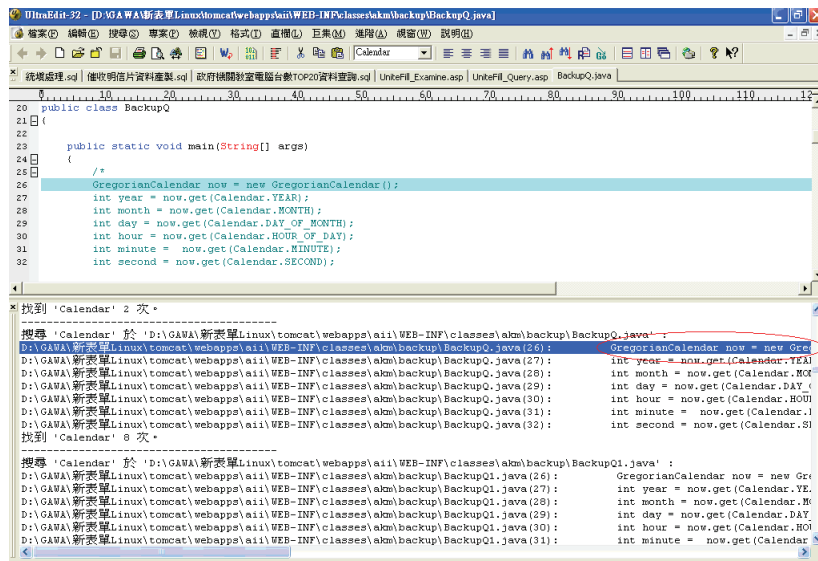
Step3、搜尋完成，搜尋之結果在下方窗格。



Step4、下方窗格右邊之捲軸，拉至最尾端，顯示共找到之次數及檔案個數。



Step5、點選搜尋到「Calendar」之某一程式檔「BackupQ.java」，上方窗格會自動開啓「BackupQ.java」原始程式檔，即可開始修正程式碼。



(本文轉載自主計月刊 98 年 5 月第 641 期)

(本文由行政院主計處電子處理資料中心組長黃芳川、設計師陳嘉華 提供)