

Client Side DataSet 與 Web Report

文/黃忠成

一個網站的效能高低通常取決於網頁程式的撰寫方式，如果撰寫網頁程式時過度依賴 Web Server，那麼網站的效能自然就會低落，因此網頁程式通常會使用 Client Side JavaScript 來完成一些資料驗證的工作，藉此減少網路流量及 Web Server 的負擔，進而提升網站的效能。但是除了使用 Client Side JavaScript 來驗證資料外，我們是否還有其它的方式來提升網站的效能呢??

Client Side DataSet

IntraWeb 除了允許設計者結合 DELPHI 原有的 TDataSet 元件來撰寫一般性的資料庫網頁程式之外，同時也允許設計者使用 Client Side DataSet 元件組來撰寫離線模式的資料庫網頁程式，何謂離線模式呢??簡單的說就是使用者操作這個網頁時不需要連結回 Server 端，也不需要 Internet 連線。從 Web Server 的觀點來看，離線模式可以有效的減低網路的流量與 Web Server 的負擔。由使用者觀點來看，使用者可以事先將網頁存在筆記型電腦中，之後只要將 IE 設定為離線模式就能在沒有 Internet 連線情況下操作這個網頁，這種模式適合常常需要在外展示資料，卻又無法時常連上 Internet 的業務員。離線模式的網頁程式雖然有著上述的優點，但也不是每個網頁都適合使用這種方式，其主要原因是 IntraWeb 並未提供太多相容 Client Side DataSet 的 UI 元件，使這類型的網頁無法做出太複雜的使用者介面，另外由於 Client Side DataSet 是將資料先轉換為 JavaScript 的陣列後再傳回 Client 端，這增加了首次連結時的網路流量，也因為 Client Side DataSet 是將資料轉換為 JavaScript 陣列，因此 Client Side DataSet 所能儲存的資料格式也就受到了 JavaScript 語法的限制，這些都是在決定使用 Client Side DataSet 功能前必須考量的因素。

Client Side DataSet 與相關元件

Client Side DataSet 細分為兩個元件，一個是 TIWClientSideDataset，另一個是 TIWClientSideDatasetDBLink，這兩個元件都位於元件盤上的 IW Control 頁，表 1 是其用途的說明。

(表 1)

名稱	說明
TIWClientSideDataset	簡單的 Client Side DataSet，藉由其 Columns 與 Data 屬性來定義其欄位名稱與資料。
TIWClientSideDatasetDBLink	繼承至 TIWClientSideDataset，可由一個 TDataSet 引入資料，例如由 TTable、TQuery 等元件引入資料。

TIWClientSideDatasetDBLink 元件擁有可由另一個 DataSet 元件引入資料的特性，因此使用頻率也較高，相對的 TIWClientSideDataset 元件就較少用到了，多半是用於展現預先定義好的資料，亦或是撰寫特定程式碼由非 DataSet 元件的資料來源中引入資料並顯示，在接下來的範例中會分別介紹這兩個元件的用法。開始撰寫程式之前，我們先檢視 IntraWeb 中有那些元件

支援 Client Side DataSet 操作，這些元件都位於元件盤的 IW Client Side 頁中(見表 2)。

(表 2)

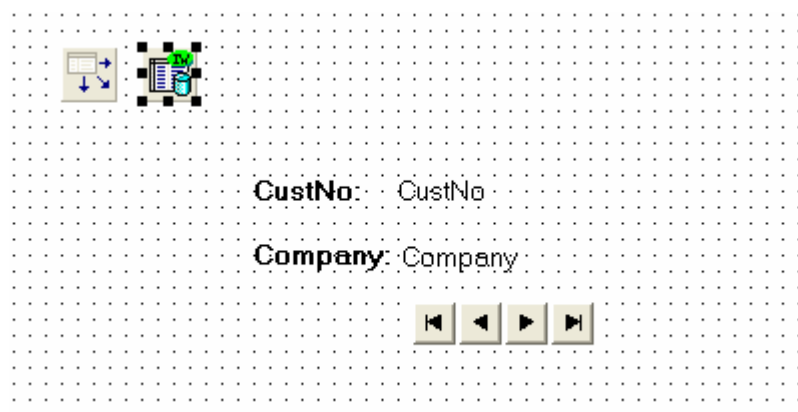
元件	說明
TIWCSLabel	簡單的 Label 元件，可顯示 Client Side DataSet 中的某一欄位資料，類似 TIWDBLabel。
TIWCSNavigator	導覽列，可移動 Client Side DataSet 的記錄指標，類似 TIWDBNavigator。
TIWDynGrid	一個 Grid 元件，可將 Client Side DataSet 中的資料以 Grid 方式顯示。
TIWDynamicChart	可將 Client Side DataSet 中的資料以圖表方式顯示，由於其用途不多，本文不會介紹這個元件。
TIWDynamicChartLegend	TIWDynamicChart 的輔助元件。

由表 2 中可以發現，Client Side DataSet 目前只支援資料的顯示，並不允許使用者改變資料內容，這也限制了其用途。

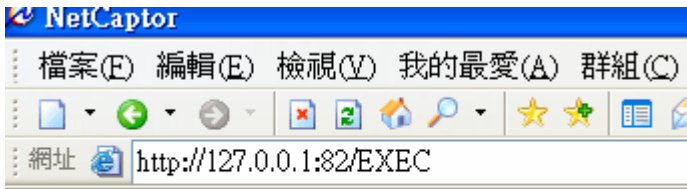
簡單的運用

這一節中的範例程式使用 TIWClientSideDatasetDBLink、TIWCSLabel、TIWCSNavigator 實作出一個可供使用者以導覽列瀏覽資料的網頁，與之前撰寫的資料庫網頁不同的是此網頁並不需要在使用者移動記錄指標時連結回 Server 端取得資料，所有操作完全在 Client 端完成。完成這個範例的步驟相當簡單，首先請放置一個 TIWClientSideDataSetDBLink 元件於 FORM 上，並設定其 DataSource 屬性為欲連結的資料表元件，接著放置用來顯示資料的 TIWCSLabel 元件及移動記錄指標的 TIWCSNavigator 元件，並設定她們的 Data 屬性為該 TIWClientSideDatasetDBLink 元件，完成的畫面如圖 1，圖 2 是執行結果。

(圖 1)



(圖 2)



CustNo: 1221

Company: Kauai Dive Shoppe



在 TIWClientSideDatasetDBLink 元件的幫助下，使用者可藉由導覽列來移動記錄指標，其間不需要再連結回 Server 端取得資料。除了 TIWCSLabel 元件之外，IntraWeb 也提供了另一個 TIWDynamicGrid 元件，可用於將 Client Side Dataset 中的資料以 Grid 模式顯示出來，並且允許使用者排序或是過濾資料。使用這個元件的方法很簡單，只需放置一個 TIWDynamicGrid 於 FORM 上，並設定其 Data 屬性為欲連結的 ClientSideDataset 元件即可，圖 3 是執行的畫面。

(圖 3)

CustNo	Company	Addr1	Addr2	City
All	All	All	All	All
1221	Kauai Dive Shoppe	4-976 Sugarloaf Hwy	Suite 103	Kapaa Kauai
1231	Unisco	PO Box Z-547		Freeport
1351	Sight Diver	1 Neptune Lane		Kato Paphos
1354	Cayman Divers World Unlimited	PO Box 541		Grand Cayman
1356	Tom Sawyer Diving Centre	632-1 Third Frydenhoj		Christiansted
1380	Blue Jack Aqua Center	23-738 Paddington Lane	Suite 310	Waipahu
1384	VIP Divers Club	32 Main St.		Christiansted
1510	Ocean Paradise	PO Box 8745		Kailua-Kona

在 TIWDynamicGrid 中，使用者可藉由 ▲ 按鈕來改變資料的排序，或是藉由位於標題列的下拉盒來選取欲觀看的資料，

例如在 City 下的對話盒中選取 Freeport，就可以使 Grid 只顯示 City=Freeport 的資料列，如圖 4 所示。

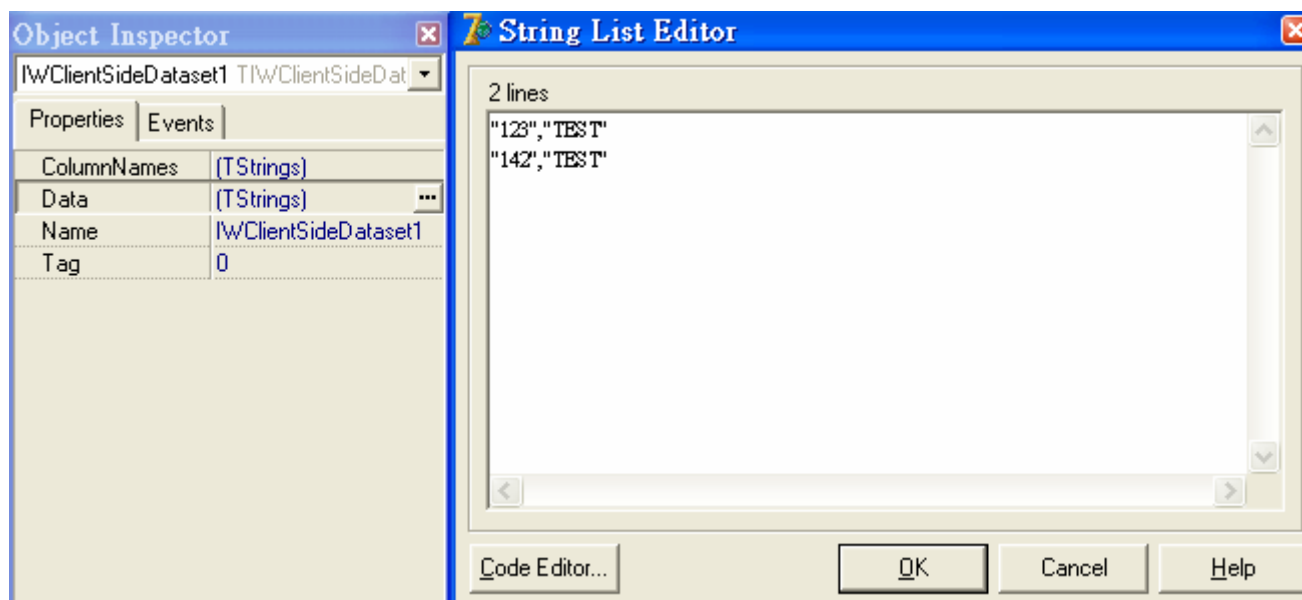
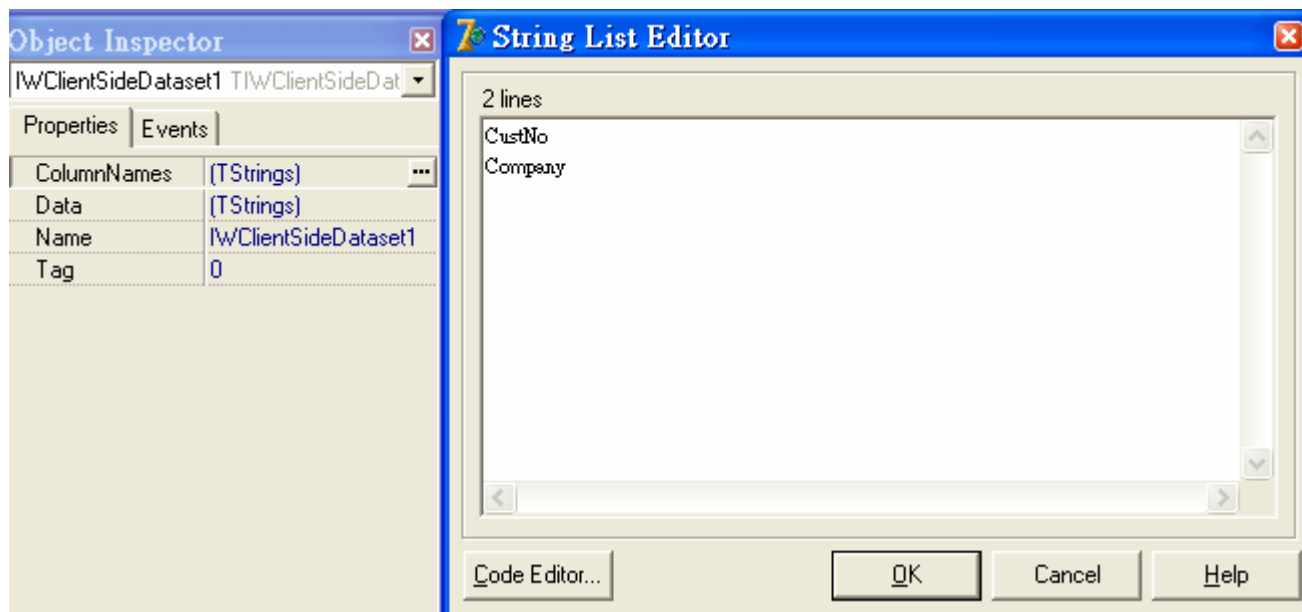
(圖 4)

CustNo	Company	Addr1	Addr2	City
All	All	All	All	Freeport
1231	Unisco	PO Box Z-547		Freeport
2165	Shangri-La Sports Center	PO Box D-5495		Freeport

使用 TIWDynamicGrid 元件與使用其它的 Client Side 元件一樣，所有排序與 Group 的動作都不需要經過 Server，讀者可以嘗試在網頁顯示出來後將 Server 程式關閉，網頁仍然可以正常的運作。

TIWClientSideDataset

上一節中使用了 TIWClientSideDatasetDBLink 元件來顯示資料，這一節中將使用其父類別:TIWClientSideDataset 元件來顯示資料，這個元件的資料必須經由撰寫程式或是使用屬性編輯器來填入，這也賦與程式設計師更高的彈性調整欲填入的資料。如果欲填入的資料屬於不變型態的話，程式設計師可以使用屬性編輯器來編修 TIWClientSideDataset.ColumnNames 及 TIWClientSideDataset.Data 屬性，藉此定義欲顯示的欄位(圖 5)及資料(圖 6)。(圖 5)



在填入資料時有一點必須注意，由於 Data 屬性的資料最後會被轉換為 JavaScript 的陣列，因此當資料內含會破壞 JavaScript 語法的字元時，整個網頁將無法正常顯示，這些字元大多是 JavaScript 的保留字。

如果事先不確定資料的內容，可使用"符號將資料標示為字串，這可以應付大多數的情況(更好的方法是使用 StringReplace 來處理資料後再放入 Data 屬性中)。範例中的 TFormClientSideDataset 展示了這個元件的用法，同時也展示了如何手動的由 DataSet 中讀入資料並填入 Data 屬性中，程式 1 是這個 PrepareData 函式的原始碼，程式並無特別之處，只是單純的由 Query 讀入資料填入 Data 屬性值中。

(程式 1)

```
procedure TFormClientSideDataSet.PrepareData;
begin
  DataModule1.Query1.Open;
  try
    DataModule1.Query1.First;
    while not DataModule1.Query1.EOF do
      begin
        IWClientSideDataset1.Data.Append(DataModule1.Query1.FieldName('CustNo').AsString+','+
          '''+DataModule1.Query1.FieldName('Company').AsString+'''');
        DataModule1.Query1.Next;
      end;
    finally
      DataModule1.Query1.Close;
    end;
  end;
end;
```

Web Report

大概所有撰寫網頁程式的人都會問一個問題，那就是該如何在網頁上呈現報表呢？其實解法不外乎三種，一是將報表輸出成 HTML 格式，直接顯示於瀏覽器上，這種方法的缺點是報表受限於 HTML 語法，無法做出較複雜的報表，另一個問題是當網頁印出時通常會有失真的情況。二是將報表輸出成 PDF 檔案，這個方法的優點是可以允許呈現較複雜的報表，同時也不會發生印出後失真的情況，但是缺點是 Client 端必須安裝可讀取 PDF 的軟體，例如 Arcobat Reader。最後一種是使用 Java Applet，這一種方法是目前最強悍的解法，不但可以顯示複雜的報表，同時也可以列印出自定格式的報表，例如正半張、標籤等等，但這並非是沒有代價的，這類型的元件通常都是高價位的，有多高？唔...2-30 萬不等。這一節中筆者將介紹如何在 IntraWeb 中使用前二種方式來顯示報表，由於筆者慣用的報表程式是 Report Builder，因此範例是使用 Report Builder 寫成的，讀者們可自行挑選其它的報表工具，概念上是相同的。

使用 HTML 顯示報表

Report Builder 的 Server 版本提供將報表轉成 XHTML 格式的元件，假如讀者們使用的是 QuickReport 或是 FastReport，可以至該元件的公司網站查詢是否有將報表轉換為 HTML 格式的元件。程式 2 是將報表輸出成 HTML，並使用另一個視窗來顯示的範例程式。

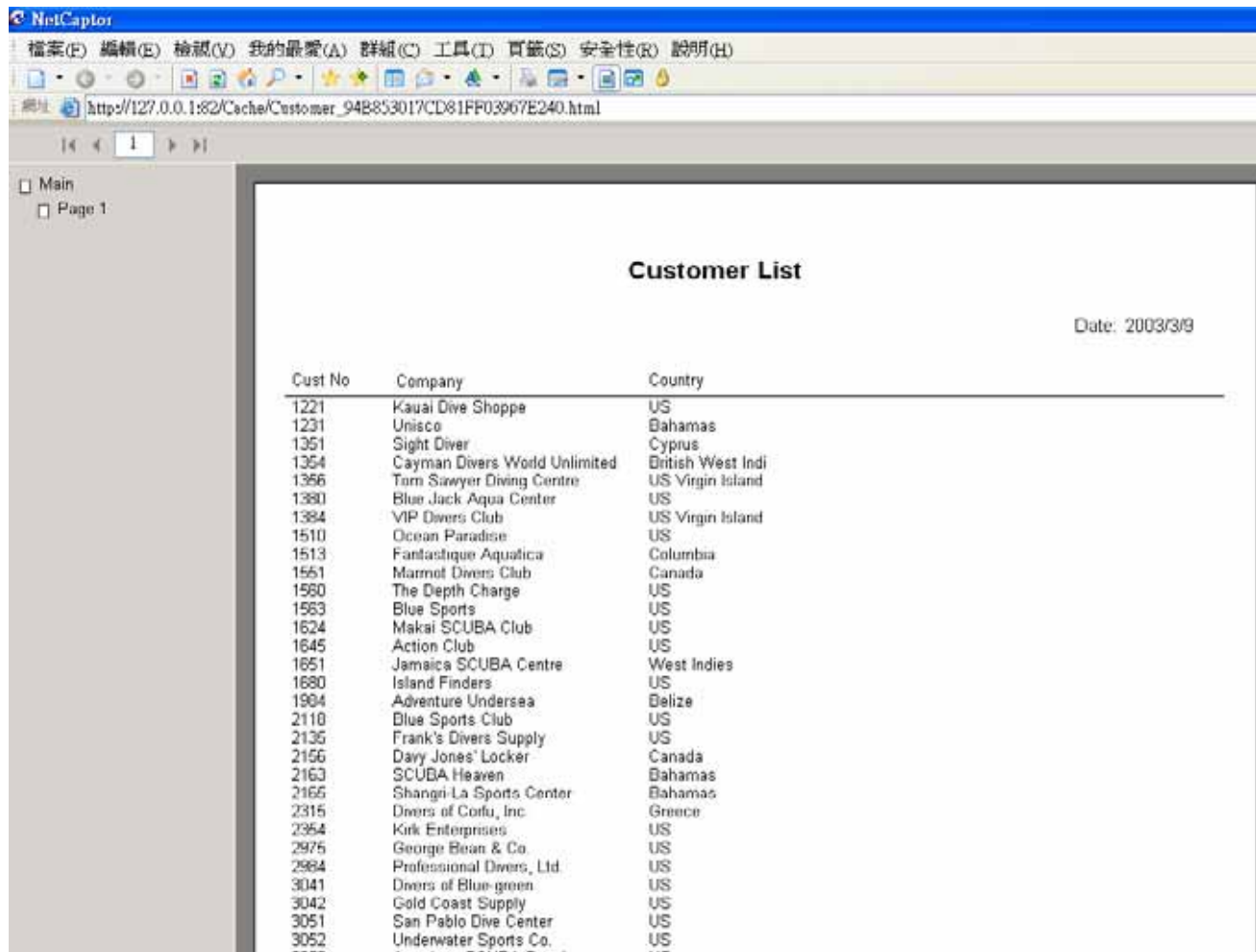
(程式 2)

```
procedure TFormReport.GenerateXHTML;
var
  XHTMLDevice:TrsXHTMLDevice;
  tmpFileName:string;
begin
  XHTMLDevice:=TrsXHTMLDevice.Create(nil);
```

```
try
    tmpFileName:=GServerController.CacheDir+'Customer_'+WebApplication.AppID+'.html';
    if FileExists(tmpFileName) then
        DeleteFile(tmpFileName);
    XHTMLDevice.FileName:=tmpFileName;
    XHTMLDevice.Publisher:=ppReport1.Publisher;
    XHTMLDevice.PageSetting:=psAll;
    ppReport1.PrintToDevices;
    AddToInitProc('NewWindow("'" + GServerController.CacheURL + ExtractFileName(tmpFileName) + "'", "", "");');
finally
    XHTMLDevice.Free;
end;
end;
```

程式 2 利用 AppID 來組成一個唯一的檔案名稱，並指定其儲存目錄為 CacheDir，這個目錄中的檔案會視 ServerController.CacheExpiry 屬性的值決定其生存週期，預設情況下這個目錄中的檔案 2 天會清一次，這可以避免報表檔案不停的增加，圖 6 是此程式執行的結果。

(圖 6)



使用 PDF

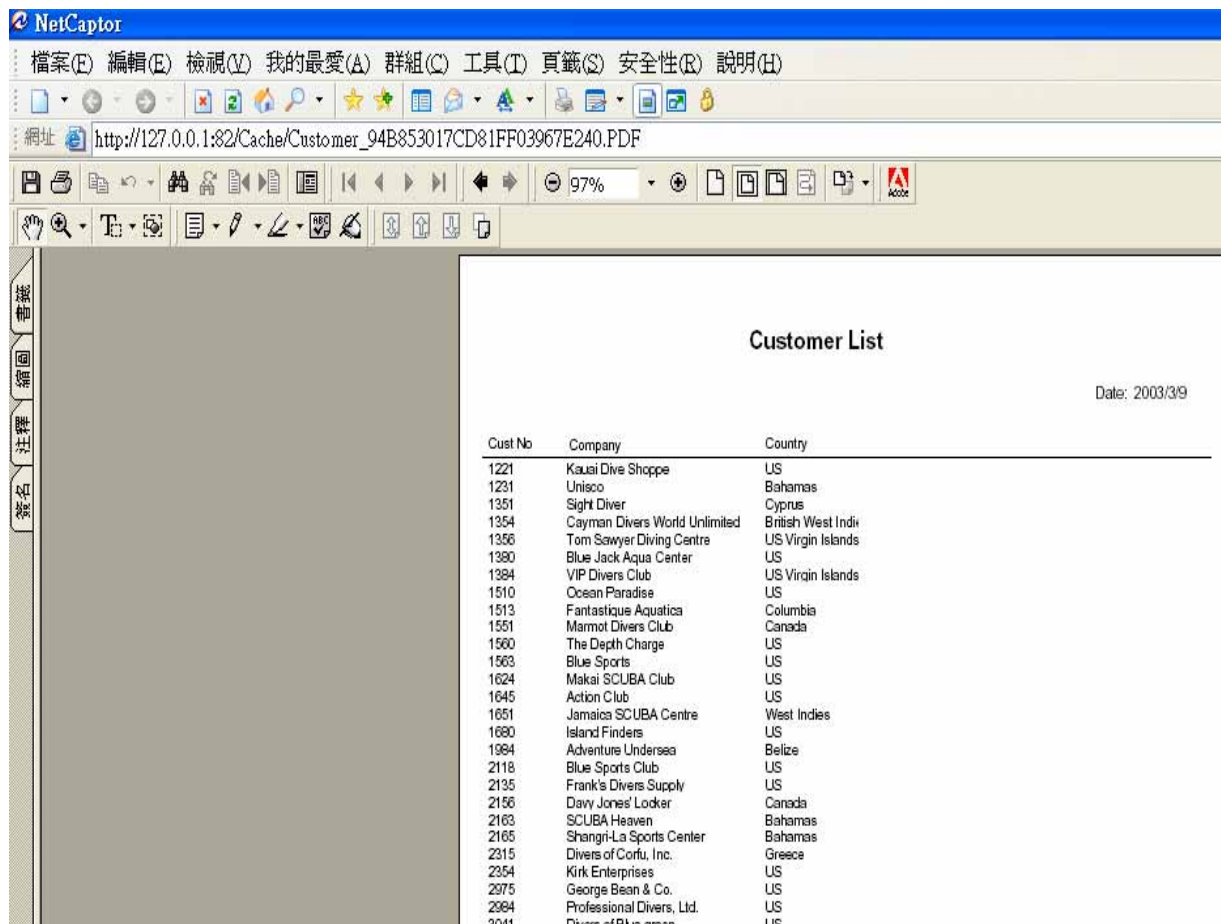
將報表輸出成 PDF 的程式碼與前面的 HTML 範例大致相同，只不過格式換成了 PDF，程式 3 是產生

PDF 的範例程式，圖 7 是執行畫面。

(程式 3)

```
procedure TFormReport.GeneratePDF;
var
  PDFDevice:TIIIPDFDevice;
  tmpFileName:string;
begin
  PDFDevice:=TIIIPDFDevice.Create(nil);
try
  tmpFileName:=GServerController.CacheDir+'Customer_'+WebApplication.AppID+'.PDF';
  if FileExists(tmpFileName) then
    DeleteFile(tmpFileName);
  PDFDevice.FileName:=tmpFileName;
  PDFDevice.Publisher:=ppReport1.Publisher;
  PDFDevice.PageSetting:=psAll;
  ppReport1.PrintToDevices;
  AddToInitProc('NewWindow('' + GServerController.CacheURL + ExtractFileName(tmpFileName) + '', '', '');');
finally
  PDFDevice.Free;
end;
end;
```

(圖 7)



將報表以 PDF 方式顯示有一個相當棘手的問題，那就是 PDF 格式在中文部份必須做一些特殊處理才能正常顯示，

目前多數的報表轉換程式所輸出的 PDF 都會有中文顯示的問題，因此請讀者自行詢問元件的廠商對此問題的解法，就筆者所知，國內有網友已提供了中文 PDF 的解決方案，有需要的讀者可至深度論壇詢問。

後記

日前 IntraWeb 推出了 6.0 版，這個版本中新增了許多的新功能，其大部份是爲了相容日後的 IntraWeb.NET 所做的改變，不幸的是現在的 DELPHI 7 使用者需要付費才能升級至此一版本，讀者可至其網站查看新功能的列表，再決定是否升級至此一版本。這一期文章中所使用的是 IntraWeb 5.1 版，不過大多數的概念可以套用在 5.0 版上，DELPHI 7 的使用者可至 IntraWeb 網站下載 5.1 版本，再使用 Key Request 程式來取得新的 License 更新，這一章也是此次連載的最後一章，日後若有機會，筆者會針對一些特定應用的主題做短篇的介紹，下次再見。