

## Tiburon 遊記 4 結合分散式 DataSnap 和 JSON 架構

隨著 Tiburon 推出的日期日益接近，每天的 Beta Build 也變得更密集了，我報上去的許多中文/Unicode 相關的臭蟲也不斷的被修正，Tiburon 的 IDE 也日益更加穩定，看來距離 Embarcadero 正式推出 Delphi/BCB 2009 的日子應該是不遠了，記得在去年初時想到不知 Unicode 的 Delphi/BCB 版本何時才會出來，沒想到日子過的這麼快，支援 Unicode 的 Delphi/BCB 版本即將出現在我們面前，而這個版本對於許多的 Delphi/BCB 開發人員來說是極為重要的一個版本，因為 Delphi/BCB 2009 是一個關鍵的昇級版本，在這個版本之後 Delphi/BCB 將進入 64 位元和多核心的世界，因此把握 Delphi/BCB 2009 昇級舊系統並且使用 Delphi/BCB 2009 開發現在和未來的系統以便相容於未來的 64 位元和多核心趨勢是最為明智的選擇。

在上一篇文章中我介紹了如何使用 Tiburon 建立一個簡易的以 JSON 為基礎的分散式架構，在本篇文章中我將繼續討論如何結合原本 Delphi/BCB 開發人員熟悉的 DataSnap 架構和 JSON 架構以便開發出新一代以 JSON 為基礎的分散式架構，如果您使用過 DataSnap，又閱讀了前面數篇文章，那麼這將非常的簡單，畢竟 RAD 精神就是讓您簡單的就能夠開發出您要的架構，不是嗎？當然如果您想要追根究底的瞭解 DataSnap/JSON 技術，提供了完整原始程式碼的 VCL 框架也可以讓您學習到許多的技術。

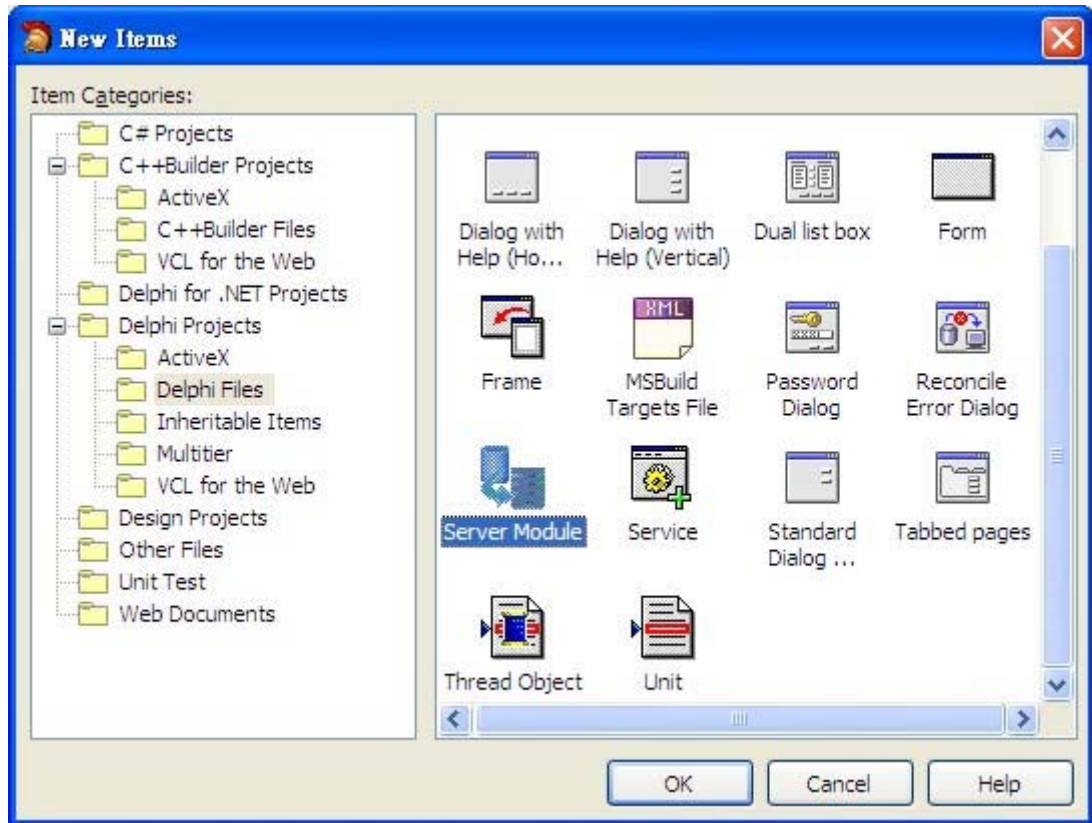
爲了簡單說明起見，我延用『Tiburon 遊記 3 動手建立一個 DataSnap JSON 伺服器吧』這篇文章基本的內容來建立分散式 DataSnap 和 JSON 架構，請您交互參考本篇和『Tiburon 遊記 3 動手建立一個 DataSnap JSON 伺服器吧』文章的內容。

### 步驟 1 – 建立分散式 JSON 伺服器

---

前 5 步驟是一樣的，接著

6. 選擇 File | New | Other，建立 Server Module 如下：

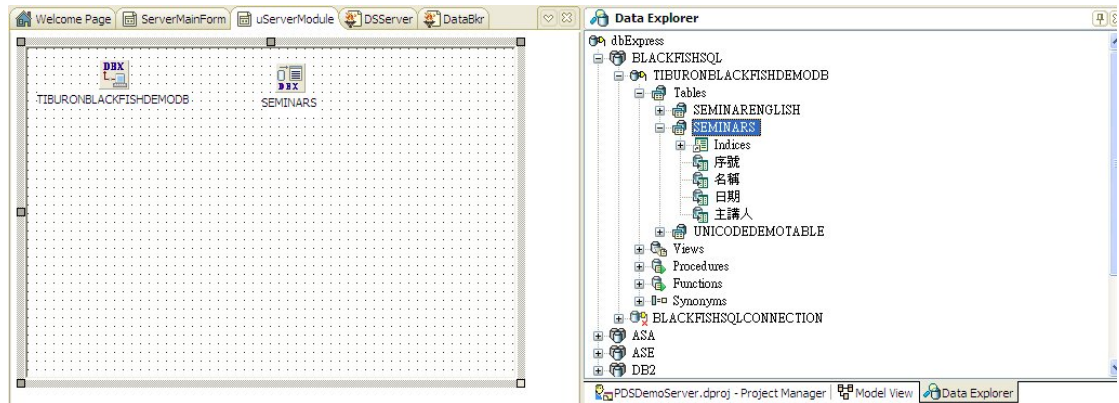


Server Module 的型態是 TDSServerModule，讓我們看看它的定義：

```
{ $MethodInfo ON }  
TDSServerModule = class(TProviderDataModule)  
end;  
{ $MethodInfo OFF }
```

果然，TDSServerModule 即是資料模組，只是以{\$MethodInfo ON}和{\$MethodInfo OFF}包圍宣告，因此它的繼承類中宣告的方法將可輸出到用戶端。

7. 在 Server Module 中拖曳資料庫中的資料表到其中，例如下圖是我把 Blackfish SQL 中的 SEMINARS 資料表拖曳到 Server Module 上：



8. 在 **Server Module** 的類別中，我宣告和實作了一個 **public** 方法 **GetSeminars** 如下：

```
function TDSServerModule1.GetSeminars : TDataSet;  
begin  
    Self.SEMINARS.Open;  
    Result := Self.SEMINARS;  
end;
```

由於在 **DataSnap 4** 中我可以直接把 **TDataSet** 傳遞到用戶端，因此在 **GetSeminars** 中我只需要把 **SEMINARS** 這個 **TSQLDataSet** 物件傳遞回用戶端即可，**DataSnap** 會自動以 **JSON** 封裝 **TDataSet** 的物件。

9. 在 **JSON** 伺服器的 **TDSServerClass** 元件的 **OnGetClass** 事件中輸出 **Server Module** 類別：

```
procedure TForm10.DSServerClass1GetClass (DSServerClass :  
TDSServerClass;  
    var PersistentClass: TPersistentClass);  
begin  
    PersistentClass := TDSServerModule1;  
end;
```

現在編譯並且執行 JSON 伺服器，接著就可以建立分散式用戶端了。

## 步驟 2 – 建立分散式用戶端

---

前 3 步驟是一樣的，只是 TSqlServerMethod 元件的 ServerMethodName 特性連結到 TDSServerModule1.GetSeminars 方法。

4. 在主表單中加入 DataSetProvider 元件連結到 TSqlServerMethod，這是因為 TSqlServerMethod 呼叫的 TDSServerModule1.GetSeminars 方法會回傳一個 TDataSet，因此就可以做為資料提供者，DataSetProvider 元件就可以以這個回傳的 TDataSet 做為資料的來源

5. 放入 TClientDataSet 元件，連結到 DataSetProvider 元件

從步驟 4,5 可以看到這和以前使用 DataSnap 開發用戶端是 樣的，但是現在是連結到 JSON 伺服器。

6. 最後再放資料感知元件和一個 Button 元件，在 Button 元件中撰寫：

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    Self.ClientDataSet1.Active := True;
end;
```

以開啓並且存取資料。

下圖是我同時執行三個分散式用戶端同時存取一個步驟 1 開發的 JSON 伺服器的畫面：



您可以看到用戶端成功的從 JSON 伺服器取得了資料並且顯示在資料感知元件中。

請注意這個應用系統重要的意義，用戶端不需要再載入各種資料庫驅動程式，只需要 DataSnap 的用戶端 DLL 和程式本身就可以從 JSON 伺服器取得資料，因為現在資料都以 JSON 格式傳遞，DataSnap 4 這種用戶端是真正的 thin-client。

使用 Tiburon 開發 JSON 架構的分散式應用系統是不是又簡單，又強大呢？如果您需要更多的處理能力，那麼 VCL 框架中提供了許多相關的 JSON 處理類別可以讓您呼叫使用，這些進階的主題就留待其他文章來討論了。

Have Fun!