

行政院及所屬各機關出國報告  
(出國類別：實習)

## 「實習 web/N\_tiers 應用技術」報告

服務機關：中華電信股份有限公司  
數據通信分公司

出國人：職 稱 姓 名  
助理工程師 施習文

出國地點：美國 加拿大

出國期間：92年10月05日 92年10月18日

報告日期：92年12月15日

H6/  
C09203585

系統識別號:C09203585

公 務 出 國 報 告 提 要

頁數: 34 含附件: 否

報告名稱:

實習web/N\_tiers應用技術

主辦機關:

中華電信數據通信分公司

聯絡人/電話:

/

出國人員:

施習文 中華電信數據通信分公司 資訊處 助理工程師

出國類別: 實習

出國地區: 加拿大 美國

出國期間: 民國 92 年 10 月 05 日 -民國 92 年 10 月 18 日

報告日期: 民國 92 年 12 月 15 日

分類號/目: H6/電信 H6/電信

關鍵詞: Web/N tiers

內容摘要: 依據中華電信公司九十二年度派員出國計畫第155項,奉派自92年10月05日至92年10月18日,共計十四天,赴美國參家Sybase公司於洛杉磯舉辦之「Sybase Developing Web Application with EAserver」課程及加拿大溫哥華舉辦之「Visual Studio.NET Web Application with Crystal Report」。藉由此次實習討論獲得Web應用開發技術、伺服器元件、應用伺服器管理、Web Reporting 應用及「物件導向技術」(Object Oriented Development)等技術在電子商務或應用系統方面的應用,希望對本公司在相關服務系統開發有所助益。

本文電子檔已上傳至出國報告資訊網

## 公務出國報告

報告名稱：研習 Web/N\_tiers 應用技術

主辦單位：中華電信數據通信分公司

出國人員：施習文

出國類別：實習、

出國地區：美國、加拿大

出國期間：2003 年 10 月 05 日至 2003 年 10 月 18 日

報告日期：2003 年 10 月 19 日

內容摘要：依據中華電信公司九十二年度資本支出派員出國計劃第 155 項，奉派自 92 年 10 月 05 日至 92 年 10 月 18 日，共計十四天，赴美國與加拿大參加 Sybase 公司於洛杉磯舉辦之「Sybase Developing Web Application with EAserver」課程及加拿大溫哥華舉辦之「Visual Studio.NET Web Application with Crystal Report 8.5」。

## 摘要

依據中華電信公司九十二年度派員出國計劃第 155 項，奉派自 92 年 10 月 05 日至 92 年 10 月 18 日，共計十四天，赴美國參加 Sybase 公司於洛杉磯舉辦之「Sybase Developing Web Application with EAserver」課程及加拿大溫哥華舉辦之「Visual Studio.NET Web Application with Crystal Report」。

藉由此次實習討論獲得 Web 應用開發技術、伺服器元件、應用伺服器管理、Web Reporting 應用及「物件導向技術」(Object Oriented Development) 等技術在電子商務或應用系統方面的應用，希望對本公司在相關服務系統開發有所助益。

## 目 錄

壹、前言	03
貳、目的	05
參、行程	06
肆、研習心得	07
一、Web/N tiers 應用技術概述	07
二、應用伺服器管理、Web 應用開發技術	11
三、Web Reporting Services 應用	18
四、Web Application 與 Crystal Report	28
伍、心得總結	32
陸、感想與建議	33
參考資料	35

## 壹、前言

網路使用的普遍性，改變了資訊流通的傳統方式，透過 Web 服務方式來取得所需的資訊，只要在任何有提供上網的地點，將可立即到網站去取得資料庫的資訊，無需如以前必需在工作地點或特定辦公室才能對資料庫進行存取作業，可大幅延伸資料存取的即時性與方便性，中華電信內部資訊服務系統也有越來越多採用 Web-Based 的趨勢。

本次訓練課程著重在 Application service(Sybase EA Server) 之 Web service 及、微軟 Visual Studio.NET 之 Reporting Service 與 Web Application 報表的開發技術等。

## 貳、目的

藉由本次實習，了解 Web Service 、Application Server 管理及、網站資料庫存取技術、物件導向系統分析與 Web-Based Reporting 等相關技術在企業網路上應用。期望對本處目前所維運系統與新系統的開發有所助益。

參、行程

日期	主要行程概述
10/5	去程
10/6~10/11	Sybase Developing Web Application with EAserver
10/12	中間行程
10/13~10/16	微軟 Visual Studio.NET Web Application with Crystal Report Server
10/17~10/18	回程



## 肆、研習心得

### 一、Web/N\_tiers 應用技術概述

#### Web 服務與分散式系統

分散式系統的意義表示其應用並不只限於本身系統機器，也可以包括遠端機器上的應用。如同一個遠端的多人連線程式，使用者利用個人電腦主機，透過網路的連結至遊戲伺服器中，與其他分散各地的使用者進行溝通。在現今的環境中，分散式系統的架構已取代傳統大型主機的集中架構。原因是現在的系統愈來愈複雜，且系統發展的模組化已成形，加上設備費用不像以往的高，因此企業再變更系統設計時，寧可將系統建構在新機器上，藉由分散式的架構連結成企業所需的系統，以節省更新舊有系統設備上所花費的人力及時間。分散式系統的應用若是能執行在適當的時機中，藉由適當的元件連結，將能帶給企業作業執行上的效率及系統的最大效能。傳統的應用程式在 Clint-Server 架構上，都只被設計在終端主機上，如此缺乏系統的規模性，若要應用程式有良好的工作表現，大多需要透過分散式的合作，因此必須藉由元件化技術，連結異端不同的電腦主機。

分散式系統的運用，最早出現在主從式(Client/Server)的架構中，而隨著資訊技術的蓬勃發展和成熟的網路建設下，使之更擴充到N階層(N-tiers)的架構中。藉由分散式物件的技術，使小單位的軟體元件隨插即用，而且分

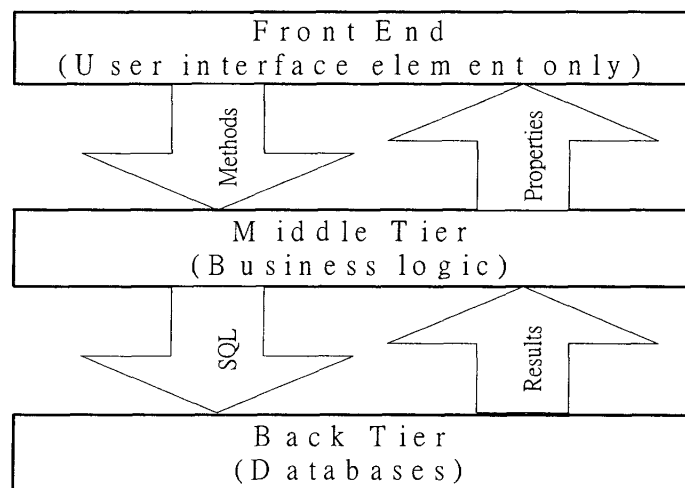
散式物件間可以透過網路相互運作，並在不同的平台上被執行；或是經由物件的包裹，可以和舊有應用系統(Legacy Applications)做結合。分散式系統本身也擁有管理的機能，可以控制資源輸出與輸入的控管。此外，藉由分散式物件所設計的系統，將具有享有物件可重複使用性(Reusability)、可擴充性(Scalability)及可維護性(Maintainability)的特性。

分散式系統在設計的本質上，是以物件導向分析與設計做基礎的。物件導向式的分析與設計方法，能協助系統開發者，迅速地利用物件模型，將企業所面臨的問題建構起來，藉由分析，釐清問題的癥結所在，藉由設計，規劃出解決方案。並且，運用物件技術，可以直接地設計並進行實作，每個物件經設計被包裹成一個元件，透過一致的元件介面，元件才可相互運作，構成一個解決企業問題的系統。值得注意的是，這些物件若是散佈在各地，便是所謂分散式物件。既然是分散式的物件，那這些物件可能是同時並存在同一台機器上，或是在遠端的機器上，要如何找出所要的物件呢？其實使用者無須煩惱這些物件身在何處，或需要以何種作業系統、應用程式去執行物件，因為現今的技術有制定異質的、分散的、元件式的中間溝通標準的架構(Standard Middle-ware)，來提供這些服務。目前市場上，三大分散式物件軟體企業所提出來的架構有：OMT(Object Modeling Technique)提出的CORBA(Common Object Request Broker Architecture)、Sun提出的Java Bean及Microsoft的DCOM(Distributed Component Object Model)。這些分散式

系統解決架構在市場上，都有相當的使用率。

### 多層式系統的架構

傳統我們常見的 Client/Server 架構，就是三層式架構的前身。為了能解決 CS 運用在現今分散的網路環境中，避免 CS 在資料處理上有過多的負載，因此三層式架構便被提出，甚是擴大推廣到多層式的架構(N-tier)中。在三層式的架構中，主要分成三個功能區塊，顯示層、邏輯層及資料作用層，如下圖所示：



圖一. 多層式架構圖

一、顯示層(Presentation)：顯示層主要利用使用者介面，如表格、表單、畫面，將資料或資訊表現出來。

二、邏輯層(Logic)：邏輯層是整個三層式架構中，在重要的一個應用部分，因為它會反映整個系統的邏輯概念。邏輯層所反映出來的結果，是資料經由系統邏輯的判斷而產生的。

三、資料作用層(Processing)：在資料作用層中，資料可以做新增、刪除、編輯的工作，並提供邏輯層使用。

### 多層式架構開發所面臨的問題

對多層式系統而言，是否提供 Web\_Based 介面供使用者操作需取決於實際的需求，並非所有的多層式系統服務都必需具備 Web\_Based 介面。

在開發 N-tiers 系統時經常會面臨以下問題：

- (1) 開發成本較高，系統時程較久，無法彈性擴充。
- (2) 元件不易部署及系統功能切分不易。
- (3) 分散式架構資料交換及一致性不佳。
- (4) 應用程式部署及使用界面難以統一。
- (5) 資料庫更新衝突不易處理及撰寫。
- (6) 系統複雜度高，不易瞭解系統全貌。
- (7) 安全性不足、權限及使用記錄難以掌控。
- (8) 過多工具及技術，系統整合不易。

全球化多國語言及本地需求不需要建置等諸類問題。

## 二、應用伺服器管理、Web 應用開發技術

資訊架構由 **Client/Service**，走向具有更大的使用者負荷量的多階 (**N-tier**) 架構的同時，資訊系統的開發工具也由單純 **Client/Server** 轉而提供多階(**N-tier**)架構，目前市面上較熱門的開發工具有 **Java(J2EE)**、**.Net**、**Delphi** 等，本節將著重於介紹 **Sybase** 公司在多階架構系統開發上所提出的解決方案 **PowerBuilder Web Framework**(簡稱 **PWF**)。

資訊系統架構從 **Mainframe** 到 **Client/Server** 已有一段時間，**Internet** 的出現對資訊世界帶來的最大影響，在於任何地方、凡是有 **IC** 和 **IP** 的設備，隨時都要互相連結得上，這也使得「**Web**化」、**「機動性」**成了資訊系統開發的首要任務。

但面對上述的需求，特別是複雜的企業電子化環境，**Client/Server** 已顯得力有未逮，**N-tier** 的架構也就應運而生。**N-tier** 架構以應用伺服器為核心，來支援透過網際網路瀏覽器與企業資料庫做溝通與聯繫的大量使用者。

傳統的 **Client-Server** 電腦系統與 **N-tier** 系統究竟有什麼不同之處呢？

下表為兩者之間的比較：

**N-Tiers 多階架構與傳統 CS(Client/Server)環境的比較表**

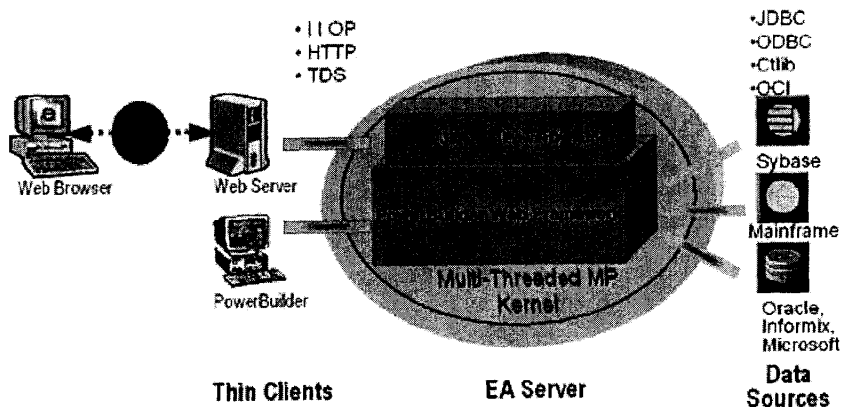
	<b>N-Tier 系統</b>	<b>Client/Server</b>
即時資訊作業能力	標準功能	無法擴充或難以擴充
同步能力	隨時	須仰賴額外的同步作業，或其他通訊工具進行批次作業
執行效率	最佳化  採用智慧型分段存取，回應快，效率高。	較慢。  易與資料量多寡成反比。
安全性	較佳  由中介的應用伺服器進行安全管理，與資料存取，後端資料庫伺服器無安全的顧慮	較低  資料庫伺服器須直接暴露在 <b>INTERNET</b> 的環境中
多使用者負載平衡	可以自動調整	沒有，且無法擴充
網路架構	為因應 <b>INTERNET/INTRANET</b> 應用而設計	為區域網路應用而設計
客戶端/遠端作業點的維護	零組態，易於維護及管理	必須獨立設定，維護管理不易

透過 **N-Tiers** 的架構，資訊人員不需要在使用者端安裝須經常維護的 **Client** 程式（如 **Windows** 應用程式），對於資訊系統的使用與管理，變得更加的輕鬆，在系統未來的規劃與擴充上也更有彈性。

## Sybase N-tiers 的解決方案 PowerBuilder Web Framework(PWF)

PowerBuilder Web Framework 採用完全開放且具擴充性的 CORBA 為架構中心，因此建立在 PowerBuilder Web Framework 之上的元件程式可完全被任何 CORBA 或 COM 前端所存取使用，舉凡 PowerBuilder、Java、C++、Microsoft Visual Basic、Microsoft Active Server Page 等之外，PowerBuilder Web Framework 亦可完全相容於 Enterprise JavaBean 的規格，可輕易整合各式前端介面。

圖二.PowerBuilder Web Framework 簡稱 PWF 的開發環境



PowerBuilder 8、EAServer 4.0、IE5.5 以上、Web Client 使用 JSP 語言開發。

### PWF 系統開發環境建置

本段介紹的 PWF 開發環境主要是建置在 Win2000 server 的平台上，所使用的 **Application Server** 為 **Sybase Enterpriser Application Server (Jaguar server 4.0)**，後端資料庫為 **Sybase ASE 12.0**，開發工具為 **Sybase 的 Power Builder 8**。

在開發環境建置這一個部分，大致上可以分成八個步驟：

#### **Step1. 安裝 Power Builder、EAServer 和 Sybase Open Client**

**Step2. 部署 EAF Package**

**Step3. 建立開發元件的 Server 端環境**

**Step4. 建立 Web Server Client 端網頁的環境**

**Step5. 使用 Native Driver 時的設定**

**Step6. 建立 Connection Cache**

**Step7. 產生 Java Script files**

**Step8. 網頁報表的設定**

八個步驟的概說如下：

**Step1. 安裝 Power Builder、EAServer 和 Sybase Open Client**

建置一個系統的開發環境，首要的任務就在於安裝我們所需要使用到的軟體工具，因此我們分別要安裝程式開發的工具程式—**PowerBuilder**、部署元件用的應用伺服器(**Application Server**)—**EAServer** 及用來連接後端資料庫的 **Sybase open client**。

**Step2. 部署 EAF Package**

**PWF** 這個 **framework** 一共包含了六個資料夾(如下表所示)和一個壓縮檔 **eaf.jar**，首先我們必須在電腦上建立一個新的資料夾(**Ex:PWF**)，用來存放 **PWF Framework** 所提供的相關檔案。

主要檔案如下

<p><i>HelpClassFiles</i>: PWF 所提供將 DataWindow 轉成 html 的 class。</p> <p><i>Javascrpts</i>: 4 個 .js 檔案。</p> <p><i>JSPClient</i>: client 端的 jsp 範例程式。</p> <p><i>PBClient_Sample</i>: 撰寫 PB client 端所需要用到的 PB library。</p> <p><i>ServerObject</i>: 撰寫 server 端元件所需用到的 PB library。</p> <p><i>WebClient</i>: power dynamo 的範例程式。</p>
---



### **Step3. 建立開發元件的 Server 端環境**

在這個步驟我們要執行的工作是要建立一個目錄(Ex: CEMIS) 來放置 Server 端 PB 相關元件(如.pbw、.pbl、.pbt)，在建立好這個目錄後，接著將 PWF 所提供的 **ServerObject** 這個目錄下拷貝到此目錄下，這個目錄下共有七個檔案(如下所示)是 PWF 所提供的 **library**，提供 **server** 端開發相關的 PB 元件和基本功能，可以讓不同的應用程式共用。

```
extension_servermain.pbl  extension_utilities.pbl
extension_serverservices.pbl  ddseaf_servermain.pbl
ddseaf_utilities.pbl      ddseaf_serverproxies.pbl
ddseaf_serverservices.pbl
```

### **Step4. 建立 Web Server Client 端網頁的環境**

這個步驟是建立一個目錄來放置 Client 端的網頁程式。首先在 **Jaguar server** 安裝目錄下，**java/classes/com/DDS** 這個目錄中，新增 **EAF20** 這個子目錄，再將由 PWF 所提供在 **HelpClassFiles** 目錄下的兩個檔案拷貝到 **EAF20** 這個目錄下。接著再把由 PWF 所提供的 **javascripts** 這個目錄拷貝到 **Jaguar server** 安裝目錄下，**html** 這個子目錄下面。

### **Step5. 使用 Native Driver 時的設定**

一般連結資料庫的方式有兩種，一種是透過 **ODBC** 來跟資料庫做連結，另外一種則是利用原廠所提供的 **Native Driver** 來連結後端資料庫。利用原廠所提供的 **driver** 來連結資料庫，一般來說在存取的速度上會優於 **ODBC**，所以建議使用 **Native Driver** 來連結。因為我們的系統是使用 **Sybase** 的資料庫，對資料做存取時，需要額外對 **Jaguar Server** 作一些設定的動作，才能正確的存取資料庫中的資料。

### **Step6. 建立 Connection Cache**

在 **native driver** 的設定完成之後，我們還必須為 **Jaguar Server** 建立一條連接到資料庫的通道，所以我們必須根據我們的需求建立 **Connection Cache**。

### **Step7. 產生 Java Script files**

在之前建立 **Client** 端環境的步驟裡，我們有複製了一個資料夾 **javascripts** 到 **Jaguar Server** 安裝目錄下的 **html** 目錄中，這個資料夾包含了四個 **.js** 檔案，分別是 **dwcomn.js**、**dwdate.js**、**dwnmbr.js** 和 **dwstr.js**，當我們想要將 **DataWindow** 顯示在網頁上時，就需要這四個檔案的幫助。

由於每一個版本的 **Power Builder** 所產生的檔案都有些許的差異，所以當我們開發 **PB** 元件的 **PB** 版本和產生 **.js** 檔案的 **PB** 版本不同時，在執行網頁程式的時候會發生執行期間的 **run time error**，因此必須要注意開發元件的 **PB** 版本，要和產生 **.js** 檔的 **PB** 版本一致。

### **Step8. 網頁報表的設定**

**PWF** 可以將 **DataWindow** 顯示在網頁上，以報表的方式呈現，如果要使用這樣的功能時，需要先將 **DataWindow** 存成 **.psr** 檔案，但是在網頁讀取 **.psr** 檔的時候，沒有辦法辨識，所以需要在 **Jaguar Server** 加入一些設定條件。

另外，如果報表有下載成 **excel** 的功能，為了要讓網頁能開啟 **excel**，必須在 **MIME Type** 中，在加入一項設定，新增一個 **extension** 叫做 **xls**，**MIME Type** 為 **application/xxx**，按下 **OK** 再重新啟動 **Jaguar Server** 即可，這樣網頁就能正確的辨識 **xls** 檔案格式了。

以上是有關 **PWF** 開發環境的建置步驟，按部就班的完成之後，我們就可以開始根據我們的需求來開發建構新系統了。

### **PWF 開發流程**

在介紹了 **PWF** 的基本架構之後，再來要談的就是如何運用 **PWF** 來開發一個多階架構的系統。**PowerBuilder** 最引以為傲的就在於它的 **DataWindow** 技術，可以簡單的將資料庫中的資料欄位呈現出來，而 **PWF** 是繼承這項特點，加入一些特殊的技巧，將 **DateWindow** 由 **Client/Server** 的方式，轉成 **N-tier** 的方式來呈現。**PWF** 的開發流程大致可分為六個步驟：

1. 製作 **DataWindow**。
2. 元件開發 (**Dataset and Business Object**)。
3. 在 **IM (Interface Manager)** 註冊元件。
4. 元件部署。
5. 開發 **JSP** (可透過 **jsp wizard**)。
6. 功能測試。

## 結論

**PWF** 運用既有純熟的 **PowerBuilder** 技巧，如同 **Client/Server** 架構下的應用程式開發方式來發展 **Web** 及分散式應用系統大量降低開發 **Web** 及分散式應用程式所需的學習曲線與風險，也大幅提升開發分散式架構 (**distributed computing**) 應用程式的速度與效率。此外，其完整的開放架構，可輕易整合多種前端介面，包含 **PowerBuilder**、**Java**、**C++**、**Microsoft Visual Basic**、**Microsoft Active Server Page** 等等。

**Sybase** 利用 **PowerBuilder** 的 **DataWindow** 結合 **webframework**，不只可以簡單輕鬆的開發出資料建檔的作業畫面，還可以依照個人需求來產生簡單的報表，是目前少有結合建檔程式開發及報表程式開發的軟體開發工具，不需學習多套軟體就可輕鬆建置一套多階架構系統，有效簡化開發環境、降低系統建置的複雜度，並提高工作效能。但在報表部分的功能尚未十分完善，因此在下一節中將介紹目前在 **web\_based** 環境中最方便開發與佈署的 **Solution**。

### 三、Web Reporting Services 應用

概說:

**Microsoft Reporting Services** 是一全面性以伺服器為基礎的平台，用來建立、管理、與傳送以傳統紙張報表為主的報告，及以互動式網頁為基礎的報告。**Reporting Services** 改善組織的生產力提供企業一個高效率管理化的報告環境。使組織的 IT 專業人員可以從任何資料來源，以熟悉的網頁瀏覽器、**Office** 工具、或內鑲於應用程式的方式來傳送即時資訊，比現有的解決方案成本低廉。搭配 **Analysis Service**，**Reporting Services** 能在任何實際的商業環境中，輕鬆的將正確資訊傳送到正確的使用人員。

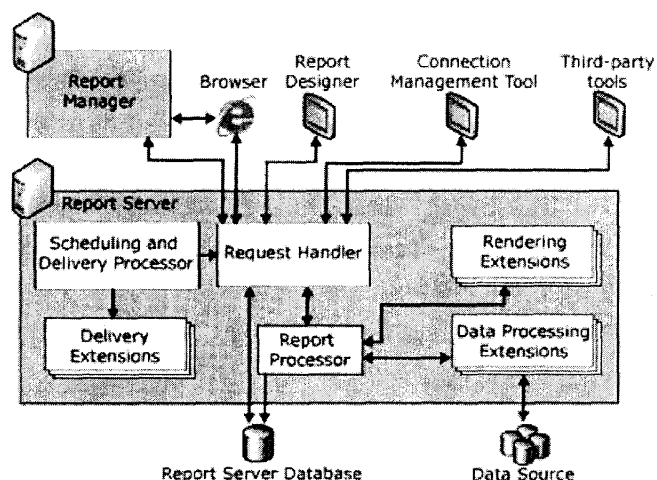
#### a. Web Services 架構

Reporting Services 使用一個 Web Services 架構。此產品包含了一個可服務報表的 Web 架構報表伺服器，和一個 Web 架構的報表管理應用程式。本產品透過 .NET Web Services 環境，在不同的平台之間通訊。

此架構以不同的方式來支援重要的設計需求：

- 此架構在設計上，不僅可延展，也可以擴充，因此滿足了企業級的可靠性和連接性需求。例如，高容量的報表環境可建立報表伺服器叢集，以提供大量的報表給 Web 架構的用戶端。
- Web 架構可減少基礎架構和實作的成本。組織可運用符合成本效益的運算平台來做為伺服器，並可快速地部署 Web 架構的伺服器，而不用建置完整的報表基礎架構。
- 使用者可以隨時/隨地存取所需的資訊，進而改善企業生產力，並在人們和資訊之間提供絕佳的連接能力。

圖三 Reporting Services 架構圖



b. 傳統和互動式報表

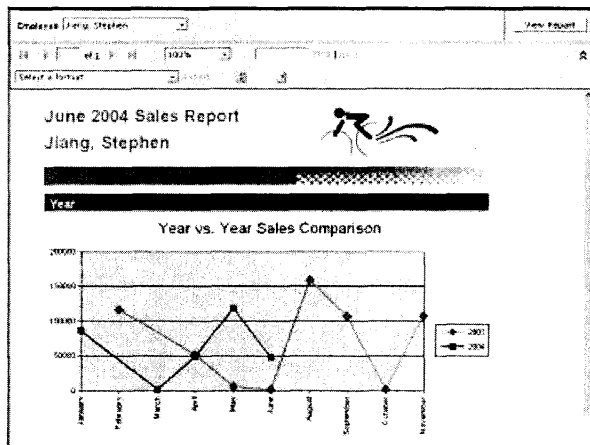
Reporting Services 可支援傳統（紙張導向）和互動式（Web 架構）的報表，因此可符合眾多的潛在報表需求。

支援的報表配置包括如下：

	描述
表格	固定的欄數
矩陣	欄數取決於查詢結果
圖表	以圖形方式來表示資料 (圓形圖、線條圖、長條圖)
清單 (任意形式)	資料隨意地排列在紙張上 - 適用於建立表單

報表的製作者可合併多種配置到一份報表內。報表處理引擎提供了後續查詢篩選、排序、群組和各式彙總，包括了加總、平均、最小值、最大值、計數和執行中的數值。

圖表配置以圖形方式來代表報表資料



### c. 互動式報表

Reporting Services 提供了許多支援使用者互動性的功能，包括：

功能	描述
參數化報表	參數化報表可篩選資料，僅提供適合使用者的資料。
隱藏的 (Drill-down) 報表項目	報表上的隱藏項目可根據使用者的動作切換成顯示或隱藏。
超連結	內嵌於報表內的超連結可引導使用者進入網頁。
鑽研式 (Drillthrough) 連結	鑽研式連結可從報表內開啓另一份報表，例如從摘要報表中開啓明細報表。原本的報表會傳送參數給鑽研式報表。
書籤連結	書籤連結可將使用者引導到目前報表的另一個區域。
文件地圖	在 HTML Viewer 中，文件地圖會顯示成報表旁邊的一個目錄。使用者可按一下目錄中的項目，之後瀏覽器就會跳到報表內的該項目。

為了進一步地增強企業的生產力，並提供使用者真正需要的資訊，

Reporting Services 提供了多個方法來將報表個人化。透過提供使用者

真正相關的報表版本，IT 部門不用再耗時地替特定的部門或個人建立與

維護專門的報表。

個人化的報表: Reporting Services 替個人化的報表提供多個方法，包括：

- 透過資料庫安全性的層級，控制使用者可存取何種報表資料。要如此做，報表必須以使用者的安全性憑證來執行。
- 提供執行時期的查詢參數搭配使用者的相關資訊。此作法會從資料來源中只蒐集必要的資訊。
- 在報表本身之中篩選資料。透過此方法，查詢會傳回所有資料，但只有與使用者相關的資訊才會包含在其報表版本內。

建立連結的報表，這可參考到現有的報表，但使用不同的屬性、權限和（或）

參數。例如，連結的報表可篩選從報表定義中傳回的資料，以便替現有的

報表建立更專門的版本。

#### d. 整合性的開發環境

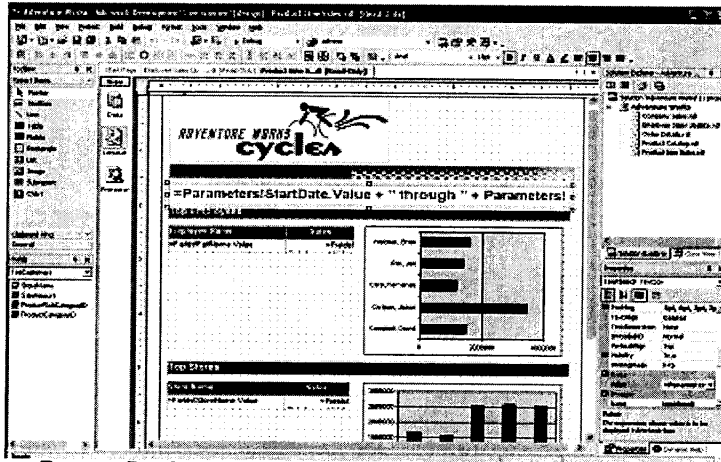
由於是完善的報表解決方案，Reporting Services 包含了建立與傳送報

表所需的一切能力。製作者不需撰寫程式，即可建立報表。Reporting

Services 也可與眾多製作工具和報表建立環境互相整合。

報表製作環境：

Reporting Services 包含了一個名為 Report Designer 的圖形式報表開發環境。Report Designer 使用了 Microsoft Visual Studio® .NET 整合性開發環境（IDE）讓使用者存取所有的報表設計功能。當您運用 Visual Studio .NET 內的豐富 IDE 和專案架構時，並不需要撰寫任何程式或 XML，即可建立報表。



Report Designer 提供了圖形式報表配置的能力

透過 Report Designer，您可定義資料來源和查詢資訊、在報表內加入資料區域和欄位、定義報表配置、替報表設定互動式功能。報表預備好之後，Report Designer 會用 Reporting Services Simple Object Access Protocol (SOAP) API 將它們發行給報表伺服器。

#### XML 報表定義語言：

報表定義一個 XML 為基礎的標準定義。Reporting Services 使用此發行的 Report Definition Language (RDL) 格式來定義報表。

以 XML 為架構架構的報表，報表製作者擁有許多製作報表的選項：

報表製作者可使用 Report Designer 產生 XML。

透過寫入到發行格式，Web 開發人員和其他人可使用文字編輯器來建立報表。

透過支援 RDL 協力廠商的工具，可建立 Reporting Services 所管理與發行的報表。

#### 多元的資料來源：

Reporting Services 可從眾多的來源中建立報表，包括了關聯式、階層式和多維式 (cube) 資料來源。Reporting Services 可支援下列的資料來源：

Microsoft SQL Server 2000 和 SQL Server 7.0 與 Oracle 資料庫 OLE DB 的相容資料來源，包括了 Analysis Services 與 ODBC



e. 全面性的報表管理：

用在伺服器架構的報表解決方案，Reporting Services 包含了企業用以整合與管理整個報表生命週期之報表活動所需的一切工具。Report Manager web 應用程式可管理報表生命週期的各個層面，包括了功能有：

1. 資料來源
2. 報表執行和排程
3. 報表記錄
4. 安全性
5. 訂閱與傳送

圖形式介面和整合管理工具集，可讓 IT 集中化地管理報表功能，同時發佈報表到整個組織和組織之外。

Reporting Services 也包含一個指令碼環境，可讓管理員自動批次化所有在部署伺服器上的操作。例如，您可撰寫指令碼，以套用安全性到伺服器上的多個報表、加入與刪除項目，或從一個伺服器中將報表伺服器項目複製到另一個伺服器內。安全性模型可區別通常由 IT 人員所執行的系統等級管理工作，以及特定報表的內容管理活動。因此，IT 可委派特定報表或報表群組的內容管理工作給報表開發人員或個別的部門，同時又整體地維持報表環境和其安全性的控制。

預定的角色，	支援下列工作的權限：
瀏覽器角色，	執行報表並瀏覽於資料夾結構之間，
內容管理員角色，	定義資料夾結構，以儲存報表和其他項目、在項目層級設定安全性、檢視與管理伺服器所儲存的項目，
發行者角色，	發行內容到報表伺服器，
我的報表角色，	針對個人用途建立報表，或將報表儲存於使用者擁有的資料夾內，
系統管理員角色，	啟用功能與設定預設值、設定整個網站的安全性、定義建立角色的定義、管理工作，
系統使用者角色，	檢視共用排程中的排程資訊，或檢視有關報表伺服器的其他基本資訊，

要滿足多個企業和大型報表客戶的需求，報表解決方案必須平衡使用者的需求，以便即時地存取最新資料，並充分地處理和使用資源。Reporting Services 包含了排程和處理報表的彈性選項。

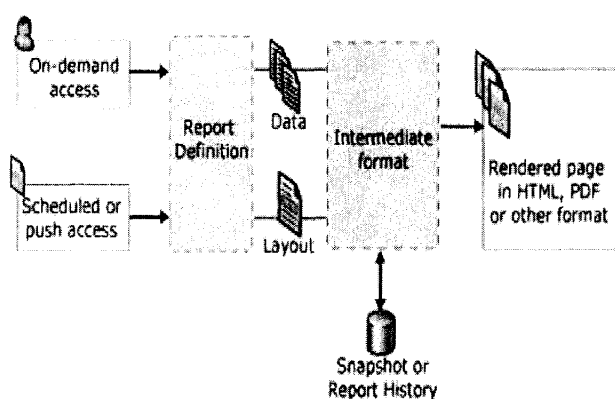
報表的排程

報表管理員可控制報表何時執行；這對於存取或處理大量資料的報表相當重要。報表的執行方式可視需要（當使用者要求它）、依排程或兩者混合。多份報表可共用一個共同的處理排程，例如月底或季末處理。

#### 報表快取

Reporting Services 可將基本報表處理（蒐集和配置資料）和報表呈現（替報表建立最後的格式）分開。在呈現與傳送報表之前，Report Server 可建立中間的報表格式。這可讓管理員將報表處理功能和傳送處理分開，以便更有效率地使用資源。

圖四:Reporting Service Access 方式



將處理和呈現分開，可讓您更有彈性地管理報表的處理時機與方式

#### f. 彈性的報表派送功能：

Reporting Services 可讓使用者隨時隨地存取所需的報表。透過彈性的呈現和傳送格式，Reporting Services 亦可傳送資訊給任何裝置。

Reporting Services 支援兩種報表傳送方式：

#### PULL 或 PUSH 傳送

視需求 (Pull) 傳送：使用者動態地向 Report Server 要求報表。

排程或事件導向 (Push) 的報表傳送：訂閱者可自動於報表資料重新整理時或依指定的排程接收報表。

視需要的報表可透過瀏覽器存取 Report Server。排程的報表傳送是由訂閱所驅動。

訂閱定義了接收報表的使用者，以及呈現和傳送的選項。例如，一個使用者可能

希望看到報表是 HTML 版本，而另一個使用者則透過電子郵件訊息的 Excel 附件來接收報表。

使用者可透過下列三種訂閱方式之一來接收報表：

- 報表管理員將使用者包含於報表的訂閱之中。
- 使用者可動態地訂閱某份報表，並可自訂傳送和呈現選項。
- 報表接收者清單於執行時，從外部的資料來源中動態產生(料導向的訂閱)。

#### 個人訂閱

使用者可建立自己的個人化訂閱，定義接收報表的偏好設定，以及想要接收哪些報表。訂閱將受制於安全性模型，因此使用者只能訂閱他們可存取的報表。個人訂閱可設定參數化報表包含特定參數（例如只檢視西部地區的銷售資料）以及呈現和傳送的偏好設定。

#### 資料導向的訂閱

在許多大型組織中，由於員工的異動或地域的重新指定，報表的接收者可能會經常變動。有些報表的對象可能原本就是動態的群組，例如下個月將要退休的員工。Reporting Services 可支援資料導向的訂閱，不需要 IT 人員手動地維護訂閱清單。在資料導向的訂閱中，系統將於訂閱執行時，使用員工資料庫等的外部資料來源，動態地產生報表接收者的清單。資料來源可包含其他資料（雇用日期、工作職稱），以篩選報表資料或自訂報表。

#### 報表呈現的選項

因為報表呈現已和起初的報表資料處理分開，同樣的一份報表可以不同的格式（HTML、PDF）呈現給不同的使用者。這些不同的格式如下：

呈現擴充功能	描述
HTML	Reporting Services 可支援 HTML 4 瀏覽器，例如，Internet Explorer 和 HTML 3.2 的相容瀏覽器，以便視需要檢視與瀏覽報表。
HTML 搭配 Office web 元件	具有額外功能的 HTML 可使用 Office Web Components (OWC) 來提供額外的用戶端互動性。圖表可透過用戶端的圖表控制功能來呈現，而矩陣可透過 PivotTable 控制功能來呈現。
MHTML	多重部分的 HTML 可讓影像內嵌於輸出檔案內。此格式適用於無法存取報表伺服器或其他位置之資源的 Web 用戶端。
EXCEL	建立包含原本配置和格式的 Microsoft Excel 試算表。
影相	以點陣圖或中繼資料來呈現報表，包括了 GDI+ 支援的任何格式：BMP、EMF、GIF、JPEG、PNG、TIFF 和 WMF。在預設狀態下，影像將以 TIFF 來呈現，它可透過影像檢視器來顯示。

	影像呈現可確保報表在每個用戶端上看起來都一樣。呈現將發生在伺服器上；而報表所使用的字體必須安裝在伺服器上。
PDF	以 Adobe Acrobat 所能開啟和檢視的 Portable Document Format (PDF) 建立報表。

開發人員也可透過開放性 API 支援其他的呈現格式。

#### g. 擴充功能、API 和介面：

Reporting Services 上可滿足不同的報表需求，包括了 ISV 需求，希望將表功能整合到工具和應用程式內。

模組化設計和廣泛的 Application Programming Interfaces (API) 可讓軟體開發人員和企業將報表能力整合至傳統系統或協力廠商的應用程式內。例如，開發人員可使用開放式介面和擴充功能來：

透過 Simple Access Object Protocol (SOAP) 介面，建立應用程式來管理報表伺服器建立應用程式或使用 Web 瀏覽器，並透過 URL 管理報表的輸出使用 Microsoft .NET Framework 建立額外的呈現、傳送和資料處理擴充功能

#### XML Web Services

Reporting Services Web Services 提供了一個進入點，讓您存取報表伺服器的完整功能，並可做為用戶端程式和報表伺服器之間的通訊介面。

Web Services 會在 HTTP 上使用 Simple Object Access Protocol (SOAP)，並讓您替報表生命週期的任何部分（從管理到執行）建立自訂工具。

#### URL 存取

您可透過 URL 要求來存取 Reporting Services 報表伺服器，因此開發人員可內嵌超連結到報表和應用程式的報表伺服器項目內。例如，您可透過 URL 從報表伺服器中存取與要求一份報表，並傳送適當的參數。URL 遵循聯合 World Wide Web Consortium W3C/IETF 草擬標準所建議的 URL 格式方針，並與大多數支援標準 URL 定址的瀏覽器和應用程式相容。

#### WMI 提供者

Reporting Services 包含了服務管理功能的 Windows Management instrumentation (WMI) 提供者。

#### 四、Web Application 與 Crystal Web Reporting

概說：

電子化報表在商業智慧中扮演的角色

商業智慧 (BI) 運用了公司的資訊資產和 IT 投資來提供商業洞察力，進而改善組織的生產力和有效性。微軟長期致力於替各種類型和規模的組織擴展 BI 能力，協助企業使用者制訂更好、更快、更適切的決策。

報表代表了商業智慧的「最後一站」，可將 BI 系統和程序的成果和價值擴及整個組織的使用者、甚至到組織之外。具有深入取得資訊能力的互動式 Web 報表已模糊了報表和互動式分析工具的界限，並提供企業使用者即時資訊以支援例行性作業和決策。

Crystal Web Reporting 將許多元件整合到一個具有彈性、以 Web 為基礎的架構內：

- 一組可建立、管理、檢視和提供報表的完整工具
- 一個可用於主控和處理報表的可延展伺服器架構引擎

開放性介面和 API，以內嵌報表或整合異質 IT 環境內的解決方式。

雖然 Crystal Web Reporting 使用 MS\_SQL Server 作為其報表資料庫，但仍可透過廣泛部署的 ODBC 和 OLE DB 介面，從不同的資料來源中存取資料。此產品幾乎已設計成可部署在任何 IT 架構內，並且完整地內嵌到不同

應用程式之中。

一般使用者報表需求可歸類成幾個一般性類別：

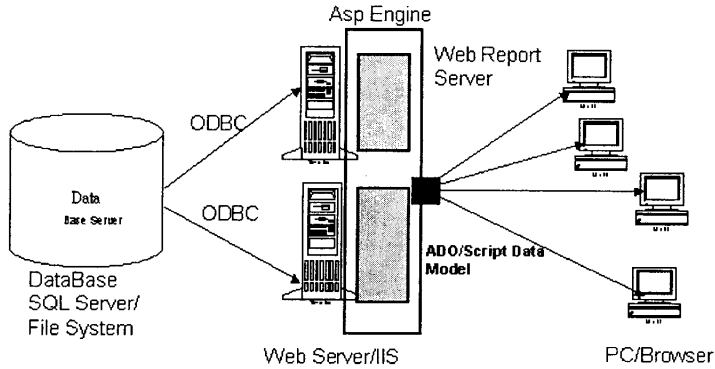
1. 企業報表：大型組織必須擁有能夠提供即時資訊給分佈各地的員工。組織應該可以快速且符合成本效益地部署報表解決方案，並且必須充分地運用現有的系統和專業。這些組織的使用者可以在需要時取得最新資訊，而不用學習新的用戶端工具或必須從 IT 人員取得特殊報表。IT 組織需要企業級管理能力和效能來管理大量的報表訂閱，並確保報表的發佈可符合公司的安全性需求。

2. 內嵌式報表：軟體廠商需要一個可整合到其所有的套裝 BI 或線上商務 (LOB) 解決方案內的報表解決方案，並且完全不會增加終端使用者的成本或複雜度。客戶應可針對套裝報表進行細微的修改，而不需要軟體廠商去提供許多特殊客製化的開發支援。簡言之，軟體廠商希望能將心力投注於經由其解決方案所提供的商業價值上，而非報表功能機制上。

3. B2B 與 B2C 報表：許多組織想要將報表內嵌在 Web 架構應用程式內，以便散發給合作廠商和客戶。這些組織需要將外部使用者與內部資料來源的複雜度互相隔離，同時又維持適當的資料安全性。大型組織需要可高度延展的解決方案，以處理上千名用戶端經由 Web 存取類似報表的狀況。

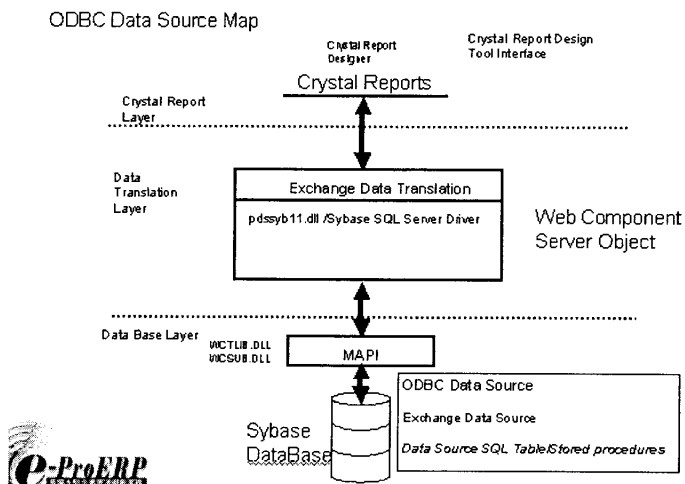
## Crystal Report 的架構

Crystal Report 7.0 具備 thin-client 的架構，使用者不需在 client 端 PC 安裝任何的 plug-in，只要有 IE Browser 即可瀏覽報表！



圖五:Crystal Report 採用 thin-client 架構

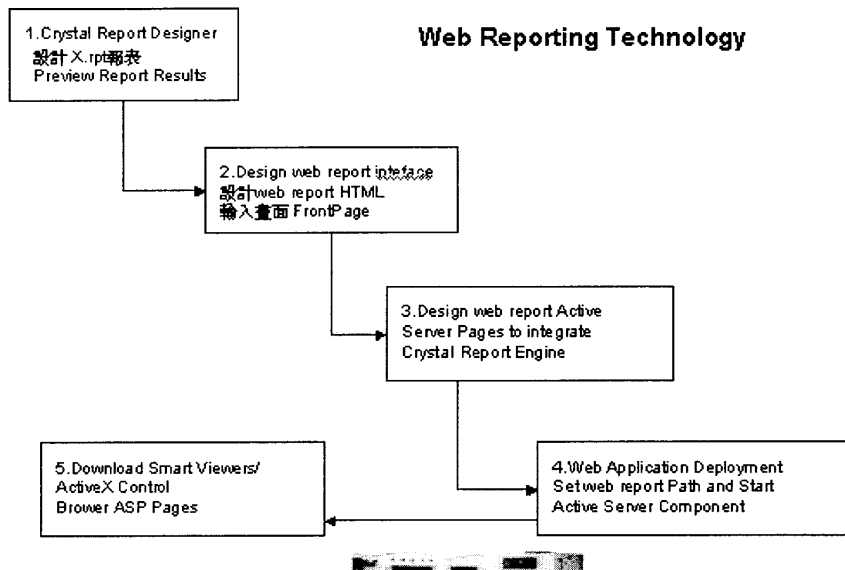
## Crystal Report 的元件組織圖



圖六:Crystal Report 元件組織圖

透過報表元件報表開發人員不需瞭解資料的結構，只需要經由圖型介面將所要的報表欄位加以組織，並使用 Crystal Report 所提供的元件布署工具即可立即將所要的報表展現在 Web 上使用完全不需開發獨立的報表程式介面，開發者亦或可自行設計使用者介面。

## Crystal Report 報表開發流程



圖七:Crystal Report 開發流程圖

Crystal Report Web Reporting 目前支援在 Web Reporting Smart Viewer 上的呈現方式有 Java , Html/XML , Office , Web ActiveX Control , PDF 等。

結論:

**Reporting Service** 在企業大型內部網路中，具備完整的電子報表開發，管理，派送，訂閱等功能，可與不同系統內既有之報表互相整合使用，而 **Crystal Report Web Reporting** 可提供 IT 或 MIS 系統開發人員快速學習使用並可整合其 **Crystal Enterprise** 所提供的伺服器應用元件來達到功能擴展與負載控制的目的。



## 伍、心得總結：

### Web 化報表服務設計的目標

為了滿足在大型組織中報表客戶需求，Web Reporting 需達到下列之目標：

- **最佳化企業生產力：**Web Reporting 透過熟悉的工具，提供即時資訊給整個組織使用者，以支援快速、適切的決策制訂。
- **最小化整體持有成本 (TCO)：**Web Reporting 是設計用來運用現有的資料來源和應用程式基礎架構。因為它包含了全面性的部署和管理報表工具，並可簡化部署與加快其速度。由於與現有的微軟產品互相整合，因此可運用熟悉的工具和現有的專業。
- **提供企業級的可靠性和延展性：**透過共同平台的能力和 Web Reporting 模型，Web Reporting 可輕鬆延展，以處理高容量的報表環境。包含了可管理整個報表生命週期的工具和程序，從報表製作一直到傳送，並支援複雜和高需求的企業環境。
- **連結使用者和資訊：**Web Reporting 可隨時隨地將使用者連結至其所需的資訊。在後端，它會從許多現有的資料來源中蒐集資料。在前端，它可以許多不同的格式，將此資料提供於 Web 上，並且輕鬆地整合到現有的應用程式內。

## 陸、感想與建議：

網際網路應用興起，本公司的資訊系統架構也漸趨複雜，加上各系統 e 化時機不同，過去本公司內部的資訊系統皆是以大型主機 (Mainframe) 模式為主，後來再逐漸轉為 Client/Server 架構，但是隨著網際網路環境成熟，目前主要廠商如微軟、昇陽等開始推廣 Web Services 概念，因此資訊系統逐漸走向多層次分散式架構 (N-Tiers)，讓終端介面逐漸標準化，使其透過網路能直接連結後端伺服器。

因此目前趨勢擴展至現今的三層式架構，分別為 Client/AP Server/DataStore。而多層式架構亦為總公司資訊處所大力倡導中，而目前總公司也正試導入一些整合性平台軟體，如 ERP、CRM、SCM，使得本公司在應用整合 (Enterprise Application Integration, EAI) 市場跟著一起成長。

雖然三階層 (N-Tiers) 技術架構成為當前資訊系統設計的主流之際，但 IT 人員卻普遍面臨技術門檻過高的瓶頸，開發時間過久，系統的維運複雜度增高的不利考量。

在決定系統服務 WEB 化應考慮幾個關鍵點：

### 1. 善用開發輔助工具 (CASE Tool)

良好的開發工具可減少開發人員摸索的時間增強對系統的掌握。

## 2. 開發工具不需趕流行，符合需求最重要

開發工具的選擇，追求流行並非重點，好的工具不等於好的系統，重點在於選擇適合系統特性需求的開發工具。

## 3. 穩定的 AP Server，維持不停頓的系統運作

N-Tiers 系統是較為複雜的運作模式，因此需有個穩固的平台與備援支援環境來運作。

## 4. 具備行動資料庫能力 隨時提供即時資訊

某些資料可在任何地點隨時存取(無線)可使業務人員在與潛在客戶互動中更能增進彼此信任度。

參考資料

1. Sybase EAS Application Server - Installation and Administration Guide  
/Sybase IT Press 2002
2. Web Services with WebSphere Studio Application Developer  
/Ueli Wahli ; Mark Tomlinson ; Olaf Zimmermann ; Wouter Deruyck ;  
Denise Hendrics  
/IBM (International technical Support Organization)
3. The XML Files : Using XML for Business-to-Business and Business-to-Consumer Application  
/Luis Ennser ; Pietro Leo ; Tamas Meszaros ; Eric Valade  
/IBM (International technical Support Organization)
4. Getting Started with Crystal Report Professional  
/Seagate Software 2003
5. Microsoft SQL Server Reporting Service - Web reporting environment solution evaluations  
/Microsoft Press 2003