

Macromedia 解決方案

拉攏 ASP、PHP、JSP 族群，擁抱 ColdFusion MX（上）

本篇將說明 ASP、PHP、JSP 與 CFML（ColdFusion Markup Language）語法上差異性外，另外也說明利用 ColdFusion 來開發網頁應用程式的優勢。

文：劉仲濱

對於開發動態網頁的人來說，不管您是使用 ASP、PHP、JSP 甚至是 ASP.NET 等之伺服器端語言，不可否定的是，其不管是價格、資源性、支援性、佔有率等等，或是您早已習慣於某一項語言，若要再請您轉換使用其他語言來開發，時常會感到沒這樣的迫切需要，甚至是浪費時間。

然而今天所介紹的 Macromedia ColdFusion MX，主要優點是能夠替您減少並大大縮短開發時間，易於維護、易於學習、系統支援性高、移植性亦高；最重要的是，對於現已持有某項伺服器端語言技術的人來說，您會發覺您更容易踏進這塊園地，因為 ColdFusion 可以說是比前述的語言還更容易學，不僅能在極短的時間內將您學過的技巧帶進到 ColdFusion，更能使您原本要花幾個星期才能開發完的系統，轉眼間於幾個小時內就開發完成，讓您有更多餘的時間去處理系統其他的工作。

本篇將說明 ASP、PHP、JSP 與 CFML（ColdFusion Markup Language）語法上差異性外，另外也說明利用 ColdFusion 來開發網頁應用程式的優勢。

簡介

ColdFusion 原是由美國一家叫 ALLAIRE 公司所開發出來的，該公司是在 1995 年由 J.J. Allaire 與 Jeremy Allaire 二位天才型兄弟所創辦，當時可說是最成功最年青的典型網際網路公司創辦人，專為企業公司解決網站上的動態網頁開發工具，並且於 2001 年 1 月與 Macromedia 公司合併，雖然公司名稱以「Macromedia」為主，但是 Jeremy Allaire 仍舊擔任 Macromedia 的技術執行長(CTO)。

合併時曾有提到未來打算，第一將 ColdFusion 發展成 J2EE 標準平台，並且加入工業標準 XML、JSP 等相容技術，使其讓業界更為親近，其二，開發一連串的網路型服務，並強調元件重複使用與程式邏輯之特性，運行於 J2EE 與 .NET 平台。

現今真的實現了，ColdFusion MX6.1 版後已將原先由 C++ 撰寫的架構，改為以 Java Base 為基礎核心，成為真正的業界 J2EE 平台之標準，不祇是效率有效提高了，也是支援各項跨平台之應用伺服器（如 WebSphere）。

從 ASP 移轉到 ColdFusion

雖然說二者間的語法差異很大，但是若你是 ASP 開發者，你的技術早已具備轉型到 ColdFusion 的條件了，非但如此，更能使你的產能提高，效果加倍。

ASP 主要是以 VBScript 與 JScript 二個語言為主，配合 ASP 內建的五大物件，但是當您在使用其物件時，通常需要先建立再使用其物件，最後還要關閉他，以免佔住系統資源，造成資源浪費，甚至當掉其應用程式。以 ColdFusion 來說，他是使用標籤 Tag 為主，從 ASP 所做的建立物件到關閉等功能，ColdFusion Server 全部自動幫您處理，讓您不需在程式碼作這些動作，我們以 ASP 建立 Recordset 為例子，如下：

```
<%  
set rsGetOrders = Server.CreateObject("ADODB.Recordset")  
rsGetOrders.ActiveConnection = "northwind"  
rsGetOrders.Source = "SELECT CustomerID, OrderDate from Orders"  
rsGetOrders.CursorLocation = 2  
rsGetOrders.LockType = 3  
rsGetOrders.Open()  
rsGetOrders_numRows = 0  
%>
```

再以迴圈的方式將 Recordset 內的資料一筆一筆抽出來顯示在網頁上（以 VBScript 為例）：

```
<table>  
<%  
While (NOT rsGetOrders.EOF)  
%>
```

```

<tr>
<td><%=rsGetOrders.Fields.Item("CustomerID").Value)%></td>
<td><%=rsGetOrders.Fields.Item("OrderDate").Value)%></td>
</tr>
<%
rsGetOrders.MoveNext()
Wend
%>
</table>

```

還沒結束，你要再將此 rsGetOrders 物件關閉，否則會造成放置在記憶體上的空間浪費。

```

<%
rsGetOrders.Close()
Set rsGetOrders = Nothing
%>

```

同樣的功能若寫為 CFML 的程式碼則不同，您會發現變得相當簡潔，以 cfquery 標籤來建立像 ASP 的物件，如下：

```

<cfquery name= "rsGetOrders" datasource= "northwind">
SELECT CustomerID, OrderDate FROM Orders
</cfquery>

```

再將此 Recordset 顯示於網頁上。

```

<table>
<cfoutput query="rsGetOrders">
<tr>
<td>#rsGetOrders.CustomerID#</td>
<td>#rsGetOrders.OrderDate#</td>
</tr>
</cfoutput>
</table>

```

說明：利用 cfquery 建立 rsGetOrders 查詢物件，並指定資料來源為 northwind，標籤內再輸入 SQL 查詢語法，最後以 cfoutput 為輸出標籤，指出 Recordset 為 reGetOrders，再指示要輸出的欄位為 reGetOrders.CustomerID 與

reGetOrders.OrderDate，程式碼是不是少了很多又很簡單呢？

另外，此 Recordset 您可以繼續在此頁面下使用，但是 ASP 卻要再用 rsGetOrder.MoveFirst()來指引出下一筆同樣的操作。

以寄信功能來說，ColdFusion 標籤已具備寄信功能，如下範例：

```
<cfmail to="cf@macromedia.com"  
from="bing@sjl.edu"  
subject="寄信標題">
```

本文內容：用 CFML 撰寫 mail 真是超簡便。

```
</cfmail>
```

(Mail Server 位址另需在 ColdFusion Administrator 內設定)

若您用 ASP 寫過 Mail 功能的就知道，除了要寫比上述程式碼還多之外，還要在此機器上註冊具有 Mail 功能的物件，如 CDO、JMail、ASPMail 等。

ASP Object

ASP 有所謂的 COM、DCOM 等物件，雖然說 ColdFusion 已經有內建相關多的標籤功能，但是若您有現成的 COM、DCOM 要使用，同樣可以使用在 ColdFusion 身上，不管是利用 CFObject 或是在 cfscripts 使用 CreateObject()，就可以呼叫您現有的 COM、DCOM，節省您另外再由 ColdFusion 開發原本 COM 功能的時間。

從 PHP 轉移到 ColdFusion

近年來 PHP 挾帶著效率高、免費的開發性語言，但就學習其語法孰不知與 CFML 仍有相當大的差距。

以 ColdFusion 來說就因為本身為標籤物件之特性，所以當您在使用其標籤語法時，一反應到 ColdFusion 後，所有標籤上的語法它都幫您完成；然而在 PHP 之中，您必須先建立起該物件的實體才能使用其物件，最後不需使用者也要關閉其物件才行。我們同樣舉建立一個 Recordset 物件來說明，您必須先建立與資料庫連線，再行建立 Recordset，如下：

```
<?php  
$connMysqlbookstore = mysql_pconnect("bing", "bing", "mypass") or  
die(mysql_error());  
mysql_select_db("bookstore", $connMysqlbookstore);  
$query_rsGetOrders = "SELECT CustomerID, OrderDate FROM Orders";  
$rsGetOrders = mysql_query($query_rsGetOrders, $connMysqlbookstore)  
or die(mysql_error());
```

```
$row_rsGetOrders = mysql_fetch_assoc($rsGetOrders);
$totalRows_rsGetOrders = mysql_num_rows($rsGetOrders);
?>
```

與 ASP 同樣，必須再使用手動迴圈方式來將 Recordset 內的資料一筆一筆抽出顯示於網頁上。

```
<table>
<?php
do {
?>
<tr>
<td><?php echo $row_rsGetOrders['CustomerID']; ?></td>
<td><?php echo $row_rsGetOrders['OrderDate']; ?></td>
</tr>
<?php
} while ($row_rsGetOrders = mysql_fetch_assoc($rsGetOrders));
?>
</table>
```

最後，您還需要將此 Recordset 從記憶體中釋放，再關閉其資料庫連結。

```
<?php
mysql_free_result($rsGetOrders);
mysql_close($connMysqlbookstore);
?>
```

相同的功能在 ColdFusion 如上述程式碼將不再列出，您亦可了解到此二者間程式碼的差異性與複雜性，值得附加說明的是，CFML 內的 DataSource 已經在 ColdFusion Administrator 中設定完成，非但如此，不管你是什麼樣的資料庫如 Access、MSSQL、Sybase、MySQL、Oracle 等都有支援。

上述同樣的 ColdFusion Recordset 您可以反覆地在網頁中使用，不需另外建立，而 PHP 還需利用 mysql_data_seek() 或其他方式才能達成。

我們再以寄信功能來說，雖然 PHP 已內建寄信函數，但是相較起來開發程式還是太多於所謂的動詞，範例如下：

```

<?php
$to = $_GET ["txt_tofield"];
$subject = "標題主題";
$subject .= $_SESSION_["order_number"];
$message = join("",file("message.txt"));
$from = "From: chongbing <bing@sjl.edu>";
$replyto = "reply-to: bing@mmug.com.tw";
$header = $from;
$header .= "\n";
$header .= $replyto;
mail($to, $subject, $message, $header); // 開始寄信
?>

```

從JSP轉移到ColdFusion

以ColdFusion運作的機制來講，當第一次執行CFM檔並傳到ColdFusion Server時，內部就先將程式碼轉換成servlets，過程中與JSP很類似，因此再仔細研究您將會發現二者的引擎是非常相似的。

對於JSP開發者而言，您可以繼續在ColdFusion MX Server Enterprise Edition 與 ColdFusion MX for J2EE版本上使用cfimport標籤來使用JSP自訂標籤程式庫，並且 ColdFusion MX for J2EE版本可以在Windows、Linux、Solaris、IBM WebSphere 4.03或5、Sun ONE Web Server 6.0或7.0、BEA WebLogic 6或7與Macromedia Jrun 4.0上執行，另外您也不用特別限定配合哪個Web Server上的設定，等於說一家設定天下行。

語言特性：

ColdFusion大部份都是以<cf為開頭，這也是為了一致性與程式碼間的易讀性，而您也可以使用標籤式或Script方式撰寫，JSP也是一樣，請下以下範例：

CFML標籤式語法：

```

<cfset myVar = 1>
<cfif myVar EQ 1>
<!--cfoutput wrapping #myVar# --->
<cfoutput>
One is <b>#myVar#</b>
</cfoutput>
<cfelse>
<b>Other</b>

```

```
</cfif>
```

JSP標籤式語法：

```
<%! int myVar = 2;%>
<% if (myVar == 1){ %>
One is <b><%=myVar%></b>
<% } else { %>
<b>Other</b>
<% } %>
<P></P>
```

CFML Script式語法

```
<cfscript>
if (myVar EQ 1){
writeOutput("One is <b>#myVar#</b>");
}else{
writeOutput("<b>Other</b>");
}
</cfscript>
```

JSP Script式語法

```
<%
if (myVar == 1){
out.println("One is <b>"+
myVar
+"</b>");
}else {
out.print("<b>Other</b>");
}
%>
```

ColdFusion在語法上較沒有像JAVA般嚴謹，例如變數名稱區分大小寫，資料型別亦沒有明確指定，也就是說您可以隨時指定任何變數給現在的字串型別或是其他物件，ColdFusion Server也不會丟出一個錯誤給您，除非您呼叫一個不帶給他資料型別的函數，或是試著要輸出一個無法預料的複雜數值。

另外，您也可以宣告一個變數參數，同時給指定他的資料型別，如：`<cfparam name="myValue" default="1" type="boolean">`，即使如此，您同樣可以在別的

CFM頁面上改變他的型別，但是您會收到一個訊息告訴您與原來的型別不同。

ColdFusion MX與JSP另外的不同點在於變數範例的不同，Request與Response在ColdFusion MX幾乎沒有相同的函數；當您在JSP必須使用`request.getParameter("title")`，但是在ColdFusion中您應該使用變數的範圍以利增加其執行效率，所以應該是用`url.title`或是`title`，如果變數從表單傳來的話，應該是`form.title`或是`title`。需要強調的是，變數名稱若相同，不管是在`url`或是`form`，若您沒指出其範圍為何的話，將會以`url`為主。

ColdFusion MX另外一個使用上的好處是，您可以利用`Application.cfm`與`OnRequestEnd.cfm`來協助您開發上的便利，`Application.cfm`是在程式執行時會先去檢查該程式目錄或子目錄下是否有這檔案，若有則會先去執行此檔，您可以將一般程式會用到的變數等公用設定或程式寫在`Application.cfm`，而`OnRequestEnd.cfm`正好相反，為程式最後執行完時會去執行的檔案，這樣的機制能幫助您解決一些系統開發上的問題。

ColdFusion MX 6.1來說增強了許多新的標籤及使用，下面就列出最MX新版的標籤內容如下：

- 1、CFCOMPONENT：利用函數與標籤建立起自訂ColdFusion元件，可供給ColdFusion網頁或是Web Services使用。
- 2、CFCHART：能將資料庫內的資料分析成表圖顯示，格式可以是SWF或是JPG、PNG等格式。
- 3、CFXML：支援XML與XSLT，可在不同來源的資料做轉換或存取。
- 4、CFDUMP：可以傾印所有的變數、結構等內容於網頁上，這對於在除錯與測試來講都非常有用。
- 5、CFDIRECTORY：提供您伺服端的檔案目錄建立、存取、刪除、更名、顯示等。
- 6、CFFILE：提供您伺服端的檔案建立、存取、刪除、更名、顯示等。
- 7、CFREGISTRY：讀取或是寫入Windows的登錄檔。
- 8、CFEXECUTE：執行程式或伺服端上的相關執行檔。
- 9、CFPOP：管理E-mail上的收取功能。
- 10、CFX：您可以建立這些C++或是Java的標籤，前提是您必在ColdFusion Administrator上註冊，使用CFX可以提高效能，並且使用到其他語言上的資源。
- 11、CFOBJECT：不管是否是在ColdFusion環境外，此標籤能讓您存取COM、CORBA或Java/EJB物件，並有重複使用的特性。

成功案例：

國內知道要用ColdFusion的人還不多，但他的潛力實在非常看好，各位讀者有興趣的人可以看如國內104人力銀行、富邦銀行、還有MSN交友網站等，加上Macromedia台灣使用者俱樂部也是用ColdFusion來開發，有興趣的朋友可以上網查詢。

就平台來講，Macromedia並沒有自己的平台，但官方並不認為此為他們的不利之處，而是像ColdFusion就能運行於J2EE與.NET二平台上，這點就是他們認為有利的優勢。

ASP、PHP、JSP各有其優越之處，但ColdFusion極短的程式碼證明了有效降低學習門檻，除了是專為動態網頁開發的語言外，更整合其他平台，與使用XML、Web services、Flash Remoting上等功能，重要的是不管您是否為老練或新手都能夠快速將開發的生產力提高，加上配合的Dreamweaver快速發展工具，在日後您在撰寫任何Web上的系統，都是一個很好的選擇。

作者：劉仲濱

E-mil：bing@sjj.edu

簡介：從事資訊相關行業多年，致力於 macromedia 產品解決方案，專長為 Dreamweaver、ColdFusion、Fireworks。目前為 Macromedia 台灣使用者俱樂部站長(<http://www.mmug.com.tw>)，有興趣的人可以與他連絡