บทที่ 30 การทำงานกับเว็บเซอร์วิส

เว็บเซอร์วิส (Web Service) คือการให้บริการกลไกการทำงานบางอย่างผ่านทางเว็บไซต์ ซึ่งตัวไฟล์เว็บนั้นทำหน้าที่เป็นคลาส ซึ่งให้แอพพลิเคชั่นของผู้ขอใช้บริการซึ่งอยู่ที่ใดก็ได้ในอินเทอร์เน็ต สร้างอินสแตนซ์ของคลาสในเว็บเซอร์วิสไปใช้งาน โดยแอพพลิเคชั่นของผู้ขอใช้บริการจะเขียนโดยใช้ภาษาโปรแกรมใดก็ได้

ส่วนประกอบของเว็บเซอร์วิส

ในการให้และรับบริการเกี่ยวกับเว็บเซอร์วิสในชีวิตจริงนั้น จะมีส่วนสำคัญหลักๆ อยู่ 3 ส่วนคือ

SOAP (Simple Object Access Protocol)

เป็นโปรโตคอลที่เว็บเซอร์วิสใช้ในการสื่อสาร ซึ่งโปรโตคอลนี้จะรับส่งข้อมูลด้วยรูปแบบ XML ซึ่งทำงานร่วมกับโปรโตคอลอื่นได้หลายอย่างเช่น HTTP, SMTP, FTP เป็นต้น

WSDL (Web Service Description Language)

ในการเผยแพร่เว็บเซอร์วิส การที่ผู้อื่นจะใช้งานเว็บเซอร์วิสของเราเป็นนั้น เราต้องมีเอกสารที่อธิบายรายละเอียดของเว็บเซอร์วิสของเรา ซึ่งเรียกว่า WSDL ซึ่งเขียนในรูปแบบของ XML เอกสาร WSDL นี้จะประกอบด้วยรายละเอียดของเว็บเซอร์วิส เช่น ชื่อเว็บเซอร์วิส, ชื่อเมธอด, ชื่อพารามิเตอร์ เป็นต้น

UDDI (Universal Discovery Description and Integration)

เปรียบเสมือน Search Engine สำหรับรวบรวมและค้นข้อมูลเกี่ยวกับเว็บเซอร์วิส ซึ่งหากเราต้องการเผยแพร่เว็บเซอร์วิส เราต้องไปลงทะเบียนเว็บเซอร์วิสของเรากับผู้ให้บริการ UDDI รายละเอียดเพิ่มเติมเกี่ยวกับ UDDI สามารถค้นได้จาก http:/uddi.org,http://uddi.microsoft.com, http://www-3.ibm.com/services/uddi/testregistry/find

ส่วนประกอบในไฟล์เว็บเซอร์วิส

ไฟล์เว็บเซอร์วิสเราจะตั้งนามสกุลเป็น .asmx และในไฟล์เว็บเซอร์วิสที่เราจะสร้าง ปกติจะต้องประกอบด้วย 4 ส่วน คือ ส่วนประกาศประเภทไฟล์เป็นเว็บเซอร์วิส, นำเข้าเนมสเปซสำหรับสร้างเว็บเซอร์วิส, สร้างคลาสที่จะให้เซอร์วิส, สร้างเมธอดในคลาส

1. ประกาศประเภทไฟล์เป็นเว็บเซอร์วิส

ใช้ไดเร็กทีฟ @WebService ซึ่งมีรูปแบบดังนี้

รูปแบบ

```
<%@WebService Language="language" Class="className"%>
```

โดยที่ *language* หมายถึง ชื่อภาษาที่จะใช้เขียนโค้ดในเอกสารนี้

className หมายถึง ชื่อคลาสที่จะให้บริการเว็บเซอร์วิส

2. นำเข้าเนมสเปซสำหรับสร้างเว็บเซอร์วิส

เนมสเปซที่จำเป็นคือ System.Web.Services (และเราสามารถนำเข้าเนมสเปซอื่นได้อีกถ้าจำเป็น) รูปแบบดังนี้

รูปแบบ

```
Imports System.Web.Services
```

3. สร้างคลาสที่จะให้เซอร์วิส

ในไฟล์เว็บเซอร์วิสจะประกอบด้วยคลาสหลายคลาสก็ได้ แต่คลาสที่จะให้เซอร์วิสมีได้คลาสเดียว คือคลาสที่ชื่อเดียวกับที่กำหนดในแอตทริบิวต์ Class ในไดเร็กทีฟ @WebService ซึ่งคลาสนี้จะต้องสืบทอด (Inherits) จากคลาส WebService ซึ่งรูปแบบการสร้างคลาสเป็นดังนี้

รูปแบบ

```
Public Class ClassName : Inherits WebService
  ..
End Class
```

โดยที่ *ClassName* หมายถึง ชื่อคลาสที่เราตั้งขึ้น

4. สร้างเมธอดในคลาส

การสร้างเมธอดในเว็บเซอร์วิส จะเหมือนกับการสร้างในคลาสทั่วๆ ไป แต่มีคำสั่ง <WebMethod> นำหน้า ซึ่งสามารถสร้างเมธอดได้ทั้งแบบโพรซีเยอร์และแบบฟังก์ชั่น ซึ่งในคลาสหนึ่งจะประกอบด้วยเมธอดมากกว่าหนึ่งก็ได้ โดยมีรูปแบบการสร้างดังนี้

กรณีเมธอดแบบโพรซีเยอร์ (สั่งทำงานบางอย่าง)

รูปแบบ

```
<WebMethod>Public Sub methodName(arguments)
        ...
End Sub
```

กรณีเมธอดแบบฟังก์ชั่น (คืนค่าจากการทำงาน)

รูปแบบ

```
<WebMethod>Public Function methodName(arguments) As DataType
        ...
End Function
```

โดยที่ *methodName* หมายถึง ชื่อโพรซิเยอร์หรือฟังก์ชั่น ซึ่งจะกลายเป็นเมธอดของคลาส

arguments หมายถึง อาร์กิวเมนต์ที่จะส่งเข้าไปในโพรซิเยอร์หรือฟังก์ชั่น สามารถมีได้หลายตัว

DataType หมายถึง ชนิดข้อมูลหรือออบเจกต์ที่ ฟังก์ชั่นคืนค่าออกมา

ชื่อเมธอดต้องนำหน้าด้วย Public เพื่อให้คลาสหรือออบเจกต์อื่นสามารถเรียกใช้เมธอดนี้ได้ มิฉะนั้น ไคลเอนต์จะไม่สามารถเข้าถึงเมธอดของเว็บเซอร์วิสได

ตัวอย่าง MyService.asmx

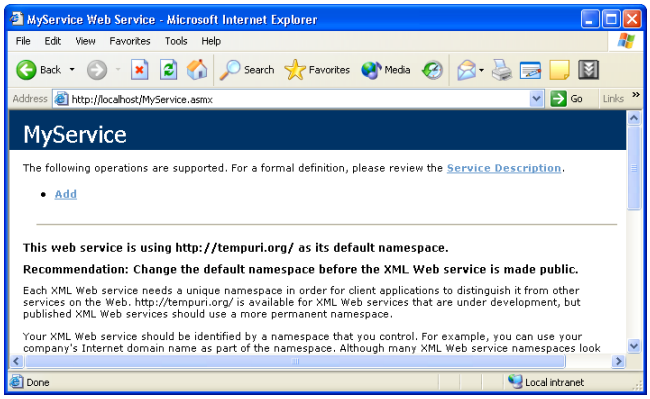
ตัวอย่างนี้เป็นการสร้างคลาสที่ภายในประกอบด้วยฟังก์ชั่นสำหรับบวกเลข และคืนค่าผลลัพธ์ที่ได้

```
1:  <%@ WebService language="VB" class="MyService" %>
2:  Imports System.Web.Services
3:  Public Class MyService : Inherits WebService
4:      <WebMethod> Public Function Add(a As Integer, b As Integer) As Integer
5:              Return a + b
6:      End Function
7:  End Class
```

บรรทัดที่ 4 สร้างฟังก์ชั่นชื่อ Add ที่ต้องการพารามิเตอร์สองตัวคือ a และ b

บรรทัดที่ 5 คืนค่าเป็นผลลัพธ์ของการบวกค่าในพารามิเตอร์ a และ b

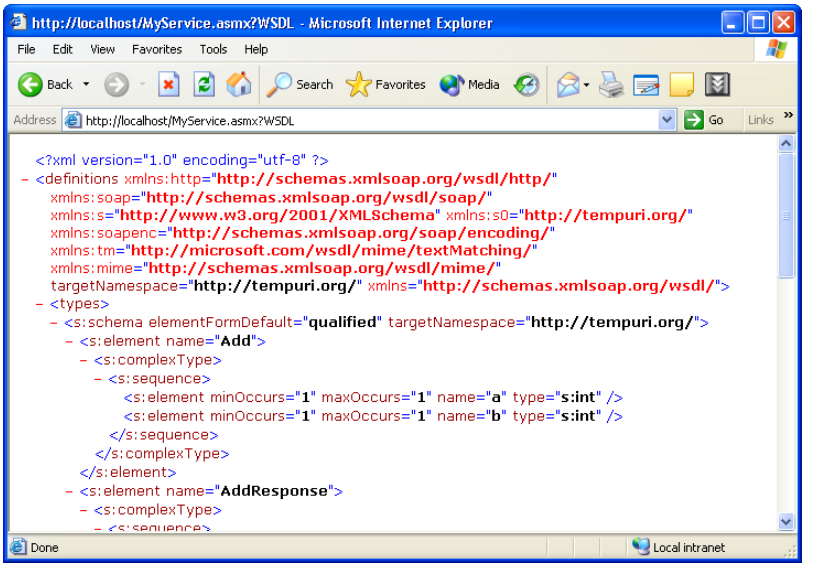
เมื่อพิมพ์โค้ดเรียบร้อยแล้วให้บันทึกไว้ใน C:\Inetpub\wwwroot\ MyService.asmx แล้วทดสอบในบราวเซอร์โดยพิมพ์ URL เป็น http://localhost/MyService.asmx จะได้ผลลัพธ์ดังรูป ซึ่งเป็นเว็บเพจที่แสดงถึงรายละเอียดของเว็บเซอร์วิสที่เราสร้าง



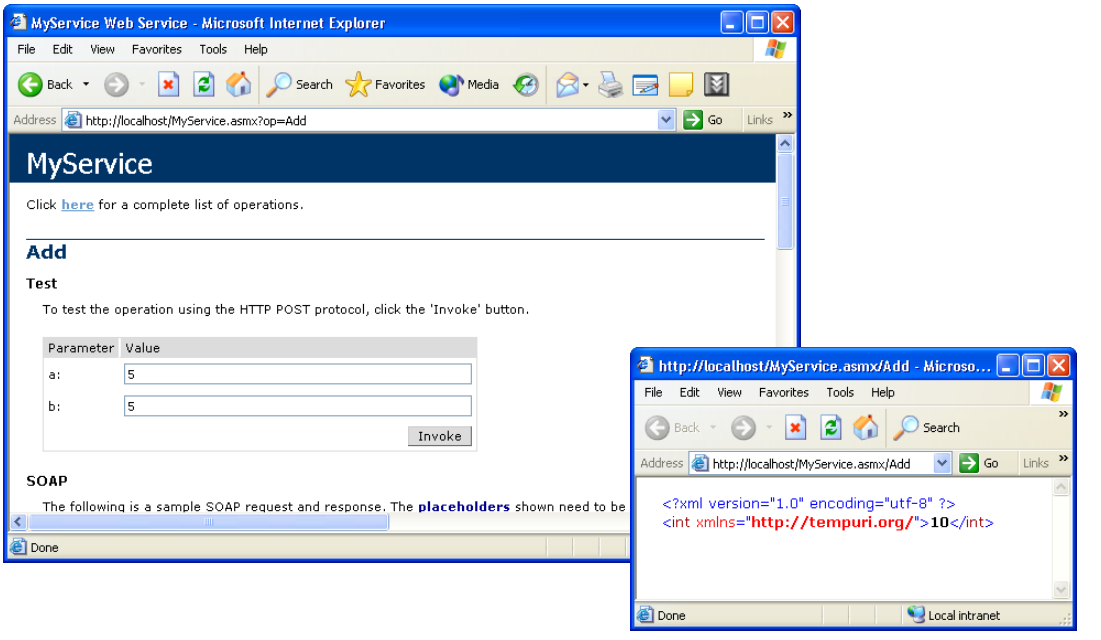
เอกสารแสดงรายละเอียดเว็บเซอร์วิส

ภายในเอกสารที่ได้จากตัวอย่างที่ผ่านมานั้น จะมีลิงค์สำคัญ 2 ลิงค์คือ

1. **Service Description** ซึ่งลิงค์ไปยังเอกสาร WDSL (http://localhost/MyService.asmx?WSDL) ที่แสดงรายละเอียดเกี่ยวกับ WebService ที่เราสร้าง ซึ่งเอกสารหน้านี้เราจะได้ใช้ในขั้นตอนการคอมไพล์ และศึกษาการใช้งานเว็บเซอร์วิส



2. **ลิงค์ชื่อเมธอดในเว็บเซอร์วิสของเรา** (ในที่นี้คือ Add) เมื่อเราคลิ้กที่ลิงค์นี้ จะเข้าสู่หน้าทดสอบใช้งานเว็บเซอร์วิสที่เราสร้าง เราสามารถทดลองกรอกข้อมูลลงในช่องพารามิเตอร์ แล้วคลิ้กปุ่ม Invoke เพื่อดูผลลัพธ์จากเมธอดของเราได้



การคอมไพล์ WebService

ไฟล์ .asmx จะสร้างเอกสาร WSDL ของเว็บเซอร์วิส แต่เรายังไม่สามารถนำไปใช้ได้โดยตรง ต้องทำการคอมไพล์เสียก่อน โดยขั้นแรกต้องแปลงเป็นไฟล์ .vb (หรืออื่นๆ แล้วแต่ภาษาที่ใช้) เสียก่อน จากนั้นนำไฟล์ .vb ที่ได้ ไปคอมไพล์เป็น .dll อีกที จากนั้นเราจึงนำเอาไฟล์ .dll ไปไว้ในไดเร็กทอรี่ bin ของเว็บไซต์เราเพื่อใช้งาน ซึ่งเราเรียกไฟล์ .dll นี้ว่า Proxy DLL

การคอมไฟล์ WSDL เป็น .vb

ผลลัพธ์ของไฟล์ .asmx จะได้เอกสาร WSDL ซึ่งอยู่ในรูปแบบ XML ดังนั้นก่อนเราจะคอมไพล์ได้ เราต้องแปลงให้เป็นเอกสารภาษา VB.NET โดยใช้โปรแกรม wsdl.exe ซึ่งปกติจะอยู่ที่ C:\Program Files\Microsoft.NET\SDK\v1.1\Bin

แต่สามารถก็อบปี้ไปไว้ที่อื่นก็ได้ มีรูปแบบการพิมพ์คำสั่ง ดังนี้

รูปแบบ

```
wsdl /l:vb /o:file.vb http://localhost/file.asmx?wsdl /n:namespace
```

โดยที่	*file.vb*	หมายถึง ชื่อไฟล์ผลลัพธ์
	http://localhost/file.asmx?wsdl	หมายถึง url ชี้ไปยังเอกสาร WSDL ของเว็บเซอร์วิส ซึ่งจะมาจากที่ใดก็ได้
	namespace	หมายถึง ชื่อเนมสเปซที่เราตั้งขึ้น เพื่อจะนำไปใช้งานในไฟล์ที่เรียกใช้เว็บเซอร์วิส

การคอมไฟล์ .vb เป็น .dll

ใช้โปรแกรม vbc.exe ซึ่งปกติจะอยู่ที่ C:\WINDOWS\Microsoft.NET\Framework\v1.1.4322 แต่สามารถก็อบปี้ไปไว้ที่อื่นก็ได้ มีรูปแบบการพิมพ์คำสั่ง ดังนี้

รูปแบบ

```
vbc /out:file.dll /t:library /r:system.web.services.dll,system.xml.dll,system.dll file.vb
```

โดยที่	*file.dll*	หมายถึง ชื่อไฟล์ผลลัพธ์
	file.vb	หมายถึง ชื่อไฟล์ที่ต้องการคอมไพล์

ตัวอย่าง คอมไพล์ไฟล์ MyService.asmx

ตัวอย่างนี้สมมุติว่าเรามีโปรแกรม wsdl.exe และ vbc.exe อยู่ในไดเร็กทอรี่ C:\Inetpub\wwwroot เรียบร้อยแล้ว เพื่อเป็นการสมมุติการใช้งานจริงจากเว็บไซต์ในอินเทอร์เน็ต ผมจึงได้ทำการก็อบปี้ไฟล์ MyService.asmx ไปไว้ในเว็บไซต์จริงในอินเทอร์เน็ตก่อน ที่ http://www.tommysong.com/MyService.asmx จากนั้นทำตามขั้นตอน

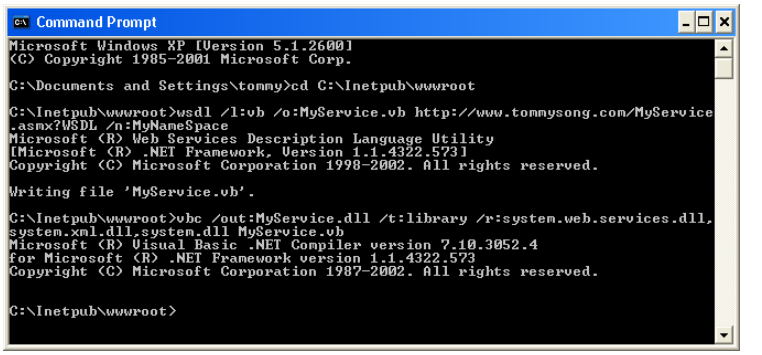
1. เปิดหน้าต่าง Command Prompt โดยคลิ้กที่เมนู Start>All Programs>Accesseries>Command Prompt
2. คราวนี้สมมุติว่าเราต้องการเรียกใช้งานเว็บเซอร์วิสจากเว็บไซต์ http://www.tommysong.com เราควรทำดังนี้

cd C:\inetpub\wwwroot แล้วกดแป้น <Enter>
wsdl /l:vb /o:MyService.vb http://www.tommysong.com/MyService.asmx?wsdl /n:MyNameSpace
แล้วกดแป้น <Enter>
vbc /out:MyService.dll /t:library /r:system.web.services.dll,system.xml.dll,system.dll MyService.vb
แล้วกดแป้น <Enter>

คำสั่งชุดแรก เป็นการย้ายมาทำงานในไดเร็กทอรี่ C:\inetpub\wwwroot

คำสั่งชุดที่สอง เป็นการแปลงเอกสาร WSDL ให้เป็นเอกสารภาษา VB.NET ซึ่งมีการสร้างเนมสเปซให้กับเว็บเซอร์วิสชื่อ MyNameSpace ซึ่งชื่อเนมสเปซนี้จะได้ใช้อ้างอิงในเอกสาร ASP.NET ที่จะเรียกใช้เว็บเซอร์วิสต่อไป

คำสั่งชุดที่สาม เป็นการคอมไพล์เอกสาร VB.NET ให้เป็นเอกสาร dll



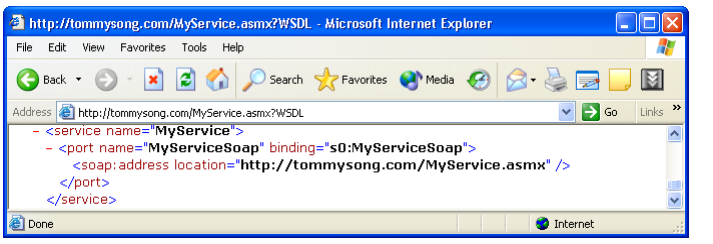
3. เมื่อเรียบร้อยแล้วเราจะได้ไฟล์ผลลัพธ์คือ MyService.vb และ MyService.dll ซึ่งไฟล์ที่เราต้องใช้งานคือ MyService.dll ให้ย้ายไฟล์นี้ไปไว้ในไดเร็กทอรี่ bin ของเว็บไซต์เรา (ในที่นี้ไดเร็กทอรี่C:\Inetpub\wwwroot\bin)

ไฟล์ .dll ที่เราเก็บไว้ในไดเร็กทอรี่ bin ของเรานั้น จะทำหน้าที่ในการเรียกใช้เว็บเซอร์วิสจากเว็บไซต์ผู้ให้บริการอีกที ดังนั้น ขณะที่เราใช้งานเว็บเซอร์วิสทุกครั้งนั้น เว็บไซต์ผู้ให้บริการจะต้องยังคงมีเอกสาร WSDL ที่เราต้องการใช้อยู่เสมอ

การใช้งานเว็บเซอร์วิส

การที่เราจะสามารถใช้งานเว็บเซอร์วิสจากผู้ให้บริการใดได้นั้น เราต้องศึกษาการทำงานจากเอกสารประกอบที่ทางผู้ให้บริการจัดทำขึ้น รวมทั้งสามารถศึกษาได้จากเอกสาร WSDL ของเว็บเซอร์วิสได้ด้วย ซึ่งจากเอกสาร MyService.asmx?WSDL มีจุดที่เราสามารถใช้ประกอบการใช้งานเว็บเซอร์วิสที่สำคัญดังนี้

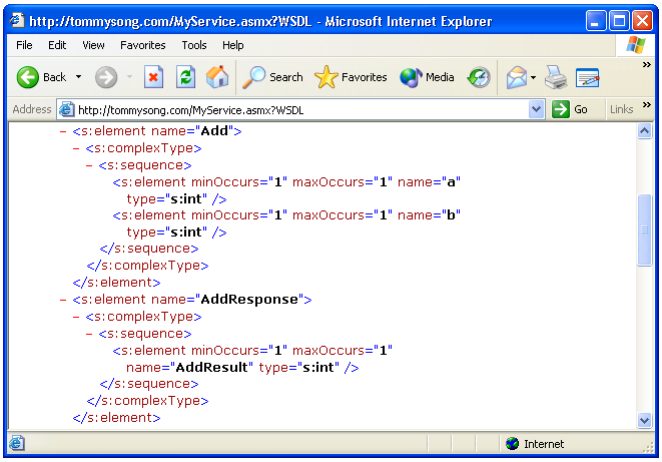
- ส่วน <service name=""> จะบอกชื่อคลาสหรือชื่อเว็บเซอร์วิส ที่เราจะใช้งาน ซึ่งในที่นี้ชื่อ MyService นั่นเอง



- ส่วน <s:element name=""> จะบอกชื่อเมธอดที่สามารถเรียกใช้งานได้ พร้อมทั้งพารามิเตอร์ที่ต้องการ และบอกการคืนค่าของเมธอดด้วย ซึ่งในที่นี้ประกอบด้วย

แท็กแรก **(name="Add")** บอกเมธอดเดียวชื่อ Add ซึ่งต้องการพารามิเตอร์ สองตัว คือ a และ b ซึ่งเป็นชนิด int ทั้งสอง

แท็กที่สอง **(name="AddResponse")** บอกการคืนค่าของเมธอด Add ว่าเป็นชนิด int



หลังจากที่เราทราบรายละเอียดต่างๆ เรียบร้อยแล้ว เราสามารถใช้งานเว็บเซอร์วิสนั้นได้โดยนำเข้าเนมสเปซของเว็บเซอร์วิสที่เราสร้างขึ้น (ตอนทำให้เป็น .vb) และเรียกใช้คลาสของเว็บเซอร์วิสได้ โดยการประกาศอินสแตนซ์เหมือนกับที่เคยใช้กับคลาสทั่วไป

ก่อนจะใช้งานเว็บเซอร์วิสทุกครั้งควรตรวจสอบให้แน่ใจก่อนว่า เว็บเซอร์วิสที่ต้องการใช้งานนั้นยังคงให้บริการอยู่ มิเช่นนั้นจะเกิดข้อผิดพลาด The underlying connection was closed: The remote name could not be resolved.

ตัวอย่าง use_Service.aspx

ตัวอย่างนี้เราจะเรียกใช้งานเมธอด Add จากเว็บเซอร์วิส เพื่อแสดงผลบวกของพารามิเตอร์ที่ส่งเข้าไป

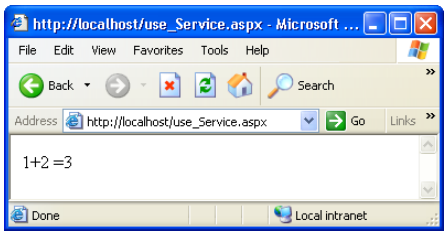
```
1:  <%@ Page Language="VB" %>
2:  <%@Import Namespace="myNameSpace"%>
3:  <script runat="server">
4:      Sub Page_Load(Sender As Object,E As EventArgs)
5:          Dim Service As New MyService
6:          Response.Write("1+2 =")
7:          Response.Write(Service.Add(1,2))
8:      End Sub
9:  </script>
10: <html>
```

บรรทัดที่ 2 นำเข้าเนมสเปซที่เราได้ตั้งไว้ในขั้นตอนการแปลง WDSL เป็นไฟล์ .vb

บรรทัดที่ 5 สร้างอินสแตนซ์จากคลาส MyService ที่อยู่ในเว็บเซอร์วิส

บรรทัดที่ 7 เรียกใช้เมธอด Add ที่อยู่ในเว็บเซอร์วิส โดยส่งผ่านพารามิเตอร์ 2 ตัวเข้าไปในเมธอดด้วย

เมื่อพิมพ์โค้ดเรียบร้อยแล้ว ให้บันทึกไว้ใน C:\Inetpub\wwwroot\use_Service.aspx แล้วทดสอบในบราวเซอร์โดยพิมพ์ URL เป็น http://localhost/use_Service.aspx จะได้ผลลัพธ์ดังรูป

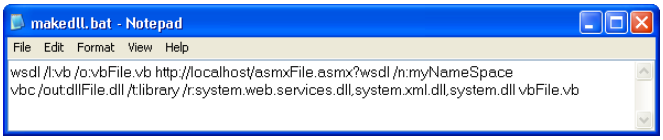


การคืนค่าจากเว็บเซอร์วิส

เนื่องจากเว็บเซอร์วิสทำงานบนพื้นฐานของ XML ดังนั้นเว็บเซอร์วิสจึงสามารถคืนค่าเป็นข้อมูลได้หลายชนิด ไม่ว่าจะเป็น Arrays, Classes, Primitives, XmlNodes หรือแม้แต่ DataSets ซึ่งเป็นตัวแทนของกลุ่มข้อมูลจากฐานข้อมูล ซึ่งต่อไปในบทนี้ จะนำเสนอแนวทางในการคืนค่าเป็นข้อมูลชนิดต่างๆ ที่สำคัญ

เนื่องจากตัวอย่างที่จะนำเสนอต่อไปในบทนี้ จะต้องมีการคอมไพล์ไฟล์ด้วย Command Prompt บ่อยๆ ครั้ง ดังนั้นเพื่อให้สามารถทำงานได้รวดเร็วขึ้น ให้เราสร้างไฟล์ชื่อ makedll.bat ขึ้นมาหนึ่งไฟล์ (ด้วย NotePad ก็ได้) โดยเก็บไฟล์นี้ไว้ใน C:\Inetpub\wwwroot ซึ่งภายในประกอบด้วยคำสั่งดังนี้ (ตัวเอียงคือส่วนที่เราจะเปลี่ยนภายหลัง)

```
wsdl /l:vb /o:vbFile.vb http://localhost/asmxFile.asmx?wsdl /n:myNameSpace
vbc /out:dllFile.dll /t:library /r:system.web.services.dll,system.xml.dll,system.dll vbFile.vb
```



การคืนค่าข้อมูลชนิดอาร์เรย์

ใช้ในกรณีที่เราต้องการคืนค่าเป็นข้อมูลหลายๆ ค่าแต่มีความสัมพันธ์กัน เราสามารถจัดข้อมูลเหล่านั้นให้เป็นหมวดหมู่ด้วยอาร์เรย์ เพื่อให้การเรียกใช้งานภายหลังนั้นสื่อความหมายกว่าการเก็บแยกเป็นหลายๆ ตัวแปร

ตัวอย่าง arrays_service.asmx

ตัวอย่างนี้เป็นการสร้างเมธอดที่รับค่าสองค่า แล้วจะสร้างและคืนค่าอาร์เรย์ที่บรรจุตัวเลขจำนวนเต็มที่อยู่ระหว่างค่าสองค่านั้น (รวมค่าทั้งสองค่าที่ส่งมาด้วย)

```
1:  <%@ WebService language="VB" class="MyService" %>
2:  Imports System.Web.Services
3:  Public Class MyService
4:      Inherits System.Web.Services.WebService

5:      <WebMethod> Public Function GetBetween(min As Integer, max As Integer) As Integer()
6:          Dim i As Integer
7:          Dim arr(max-min) As Integer
```

```
8:              For i = 0 to max-1
9:                      arr(i) = min+i
10:             Next
11:             Return arr
12:     End Function
13: End Class
```

บรรทัดที่ 5 โปรดสังเกตว่าฟังก์ชั่นนี้คืนค่าเป็นอาร์เรย์ของ Integer (As Integer())

บรรทัดที่ 7 ประกาศตัวแปรอารเรย์ของ Integer ซึ่งจะมีสมาชิกมากกว่าผลลัพธ์ของ max-min อยู่ 1 เพราะสมาชิกอาร์เรย์ตัวแรกจะเป็น arr(0) นั่นเอง ทำให้เพียงพอต่อการบรรจุตัวเลขทุกตัว ตั้งแต่ min ถึง max

บรรทัดที่ 8-10 วนรอบนำตัวเลขตั้งแต่ min ถึง max บรรจุลงในสมาชิกอาร์เรย์ตำแหน่งที่ 0 ถึง max-1

หลังจากพิมพ์โค้ดเสร็จแล้ว ให้เข้าไปแก้ไขไฟล์ makedll.bat ดังคำสั่งข้างล่าง แล้วดับเบิ้ลคลิ้กที่ไฟล์ makedll.bat เพื่อรันคำสั่งภายใน จากนั้นให้ย้ายไฟล์ .dll ที่ได้ ไปยังไดเร็กทอรี่ C:\Inetpub\wwwroot\bin

```
wsdl /l:vb /o:arrays_service.vb http://localhost/arrays_service.asmx?wsdl /n:myarrays_service
vbc /out:arrays_service.dll /t:library /r:system.web.services.dll,system.xml.dll,system.dll
arrays_service.vb
```

ตัวอย่าง use_arrays_service.aspx

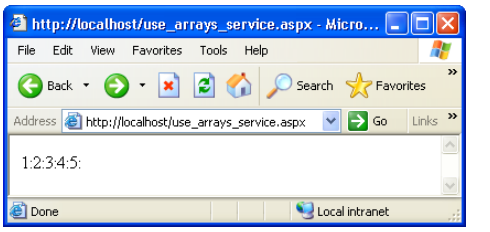
ตัวอย่างนี้จะใช้เว็บเซอร์วิสที่สร้างในตัวอย่างที่แล้ว โดยจะส่งพารามิเตอร์ 1 และ 5 เข้าไปยังเมธอด เพื่อให้แสดงตัวเลข 1 ถึง 5

```
1:  <%@ Page Language="VB" %>
2:  <%@Import Namespace="myarrays_service"%>
3:  <script runat="server">
4:      Sub Page_Load(Sender As Object,E As EventArgs)
5:              Dim Service As New MyService
6:              Dim i As Integer
7:              Dim Result() As Integer = Service.GetBetween(1,5)
8:              For i = 0 To Ubound(Result)
9:                      Response.Write(Result(i)&":")
10:             Next
11:     End Sub
12: </script>
```

บรรทัดที่ 7 ประกาศตัวแปรอาร์เรย์เพื่อรับค่าที่คืนจากเมธอด GetBetween เพราะเมธอดนี้คืนค่าเป็นอาร์เรย์ เราส่งพารามิเตอร์ให้กับเมธอดนี้ 2 ตัว คือ 1 และ 5

บรรทัดที่ 8 วนรอบทำงานตั้งแต่ i มีค่าเป็น 0 ถึงดัชนีของสมาชิกตัวสุดท้ายในอาร์เรย์ (Ubound())

เมื่อพิมพ์โค้ดเรียบร้อยแล้ว ให้บันทึกไว้ใน C:\Inetpub\wwwroot\use_arrays_service.aspx แล้วทดสอบในบราวเซอร์โดยพิมพ์ URL เป็น http://localhost/use_arrays_service.aspx จะได้ผลลัพธ์ดังนี้



การคืนค่าข้อมูลชนิดออบเจกต์

ในกรณีที่เราต้องการคืนค่าข้อมูลที่มีความซับซ้อนมากๆ เพราะควรจะส่งในรูปแบบออบเจกต์ เพราะในออบเจกต์สามารถบรรจุข้อมูลได้หลายชนิด และสามารถเข้าถึงข้อมูลได้จากพรอพเพอร์ตี้ ซึ่งสื่อความหมายในการใช้งานภายหลังมากกว่าอาร์เรย์

ตัวอย่าง object_service.asmx

ตัวอย่างนี้จะมีการสร้างคลาสขึ้นมา 2 คลาสในไฟล์เว็บเซอร์วิสไฟล์เดียว โดยคลาสที่จะเป็นเว็บเซอร์วิสนั้นจะเป็นชื่อ MyService ซึ่งจะคืนค่าเป็นออบเจกต์ของคลาสหนึ่งซึ่งชื่อ MyObject โดยในคลาส MyObject นั้นประกอบด้วยพรอพเพอร์ตี้ 2 ตัว

```
1:  <%@ WebService language="VB" class="MyService" %>
2:  Imports System.Web.Services
3:  Public Class MyObject
4:      Public objProperty1 As Integer
5:      Public objProperty2 As String
6:  End Class
7:  Public Class MyService
8:      Inherits System.Web.Services.WebService
9:      <WebMethod> Public Function GetObject() As MyObject
10:         Dim my_Obj As New MyObject
11:         Return my_Obj
12:     End Function
13: End Class
```

บรรทัดที่ 3-6 สร้างคลาสชื่อ MyObject ซึ่งภายในประกอบด้วยพรอพเพอร์ตี้ 2 ตัวคือ objProperty1 ชนิด Integer และ objProperty2 ชนิด String

บรรทัดที่ 9 จะเป็นว่าเมธอดนี้จะคืนค่าเป็นชนิด MyObject ซึ่งเป็นคลาสที่สร้างไว้ในบรรทัดที่ 3-6

บรรทัดที่ 10-11 สร้างอินสแตนซ์หรือออบเจกต์ของคลาส MyObject แล้วคืนค่าออบเจกต์นี้

หลังจากพิมพ์โค้ดเสร็จแล้ว ให้เข้าไปแก้ไขไฟล์ makedll.bat ดังคำสั่งข้างล่าง แล้วดับเบิ้ลคลิ้กที่ไฟล์ makedll.bat เพื่อรันคำสั่งภายใน จากนั้นให้ย้ายไฟล์ .dll ที่ได้ ไปยังไดเร็กทอรี่ C:\Inetpub\wwwroot\bin

```
wsdl /l:vb /o:object_service.vb http://localhost/object_service.asmx?wsdl /n:myobject_service
vbc /out:object_service.dll /t:library /r:system.web.services.dll,system.xml.dll,system.dll
object_service.vb
```

ตัวอย่าง use_object_service.aspx

ตัวอย่างนี้จะใช้เว็บเซอร์วิสที่สร้างในตัวอย่างที่แล้ว โดยจะมีการกำหนดและอ่านค่าจากพรอพเพอร์ตี้ของออบเจกต์ที่คืนค่ามา

```
1:    <%@ Page Language="VB" %>
2:    <%@Import Namespace="myobject_service"%>
3:    <script runat="server">
4:        Sub Page_Load(Sender As Object,E As EventArgs)
5:            Dim Service As New MyService
6:            Dim objService As MyObject
7:            objService = Service.GetObject()
8:            objService.objProperty1 = 501
9:            objService.objProperty2 = "Hello"
10:           Response.Write(objService.objProperty1)
11:           Response.Write(objService.objProperty2)
12:       End Sub
13:   </script>
```

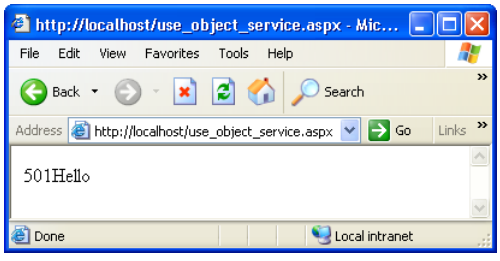
บรรทัดที่ 6 สร้างตัวแปรชนิด MyObject เพราะเมธอดที่เราจะเรียกใช้งานคืนค่าเป็นชนิด MyObject

บรรทัดที่ 7 รับค่าจากเมธอด GetObject() ซึ่งได้ข้อมูลเป็นชนิดออบเจกต์ MyObject

บรรทัดที่ 8-9 กำหนดค่าให้กับพรอพเพอร์ตี้ของออบเจกต์

บรรทัดที่ 10-11 แสดงค่าในพรอพเพอร์ตี้ของออบเจกต์ออกมา

เมื่อพิมพ์โค้ดเรียบร้อยแล้วให้บันทึกไว้ใน C:\Inetpub\wwwroot\ use_object_service.asmx แล้วทดสอบในบราวเซอร์โดยพิมพ์ URL เป็น http://localhost/use_object_service.asmx จะได้ผลลัพธ์ดังนี้



การคืนค่าข้อมูลชนิดอาร์เรย์ของออบเจกต์

ในกรณีที่เราต้องการคืนค่าข้อมูลที่มีความซับซ้อนมากๆ ขึ้นไปอีก เราอาจต้องการคืนค่าเป็นออบเจกต์หลายๆ ตัว ดังนั้นเราสามารถนำเอาออบเจกต์หลายๆ ตัวนั้นบรรจุลงอาร์เรย์ แล้วคืนค่าเป็นอาร์เรย์ของออบเจกต์ก็ได้

ตัวอย่าง array_object_service.asmx

ตัวอย่างนี้เราจะสร้างคลาสขึ้นมา 2 คลาสเหมือนตัวอย่างที่แล้ว แต่เราจะสร้างอินสแตนซ์ของคลาส MyObject หลายๆ ตัว แล้วบรรจุอินสแตนซ์หรือออบเจกต์ของคลาสแต่ละตัวนั้นลงในอาร์เรย์ จากนั้นจึงคืนค่าเป็นอาร์เรย์ที่บรรจุออบเจกต์อีกที

```
1:    <%@ WebService language="VB" class="MyService" %>
2:    Imports System.Web.Services
3:    Public Class MyObject
4:        Public objProperty1 As Integer
5:        Public objProperty2 As String
6:    End Class
7:    Public Class MyService
8:        Inherits System.Web.Services.WebService
9:        <WebMethod> Public Function GetObject() As MyObject()
10:           Dim my_Obj(2) As MyObject
11:           my_Obj(0) = New MyObject
12:           my_Obj(1) = New MyObject
13:           my_Obj(2) = New MyObject
14:           Return my_Obj
15:       End Function
16:   End Class
```

บรรทัดที่ 9 โปรดสังเกตว่าเมธอดนี้คืนค่าเป็นชนิด MyObject() ซึ่งเป็นอาร์เรย์

บรรทัดที่ 10 สร้างอาร์เรย์ชนิด MyObject ขึ้นมาโดยประกอบด้วยสมาชิก 3 ตัว (ตั้งแต่ลำดับที่ 0-2)

บรรทัดที่ 11-13 มอบหมายอินสแตนซ์ของคลาส MyObject ลงในสมาชิกอาร์เรย์

หลังจากพิมพ์โค้ดเสร็จแล้ว ให้เข้าไปแก้ไขไฟล์ makedll.bat ดังคำสั่งข้างล่าง แล้วดับเบิ้ลคลิ้กที่ไฟล์ makedll.bat เพื่อรันคำสั่งภายใน จากนั้นให้ย้ายไฟล์ .dll ที่ได้ ไปยังไดเร็กทอรี่ C:\Inetpub\wwwroot\bin

```
wsdl /l:vb /o:array_object_service.vb http://localhost/array_object_service.asmx?wsdl
/n:myarray_object_service
vbc /out:array_object_service.dll /t:library /r:system.web.services.dll,system.xml.dll,
system.dll array_object_service.vb
```

ตัวอย่าง use_array_object_service.aspx

ตัวอย่างนี้จะใช้เว็บเซอร์วิสที่สร้างในตัวอย่างที่แล้ว โดยจะทำงานวนลูป กำหนดค่าและอ่านค่าพรอพเพอร์ตี้จากออบเจกต์แต่ละตัวในอาร์เรย์

```
1:    <%@ Page Language="VB" %>
2:    <%@Import Namespace="myarray_object_service"%>
3:    <script runat="server">
4:    Sub Page_Load(Sender As Object,E As EventArgs)
5:        Dim Service As New MyService
6:        Dim objService(2) As MyObject
7:        objService = Service.GetObject()
8:        Dim i As Integer
9:        For i = 0 To UBound(objService)
10:           objService(i).objProperty1 = i
11:           objService(i).objProperty2 = "Hello" & i
12:           Response.Write(objService(i).objProperty1 & "<br>")
13:           Response.Write(objService(i).objProperty2 & "<p>")
14:       Next
15:   End Sub
16:   </script>
```

บรรทัดที่ 6 สร้างตัวแปรอาร์เรย์ซึ่งประกอบด้วยสมาชิก 3 ตัว ชนิด MyObject

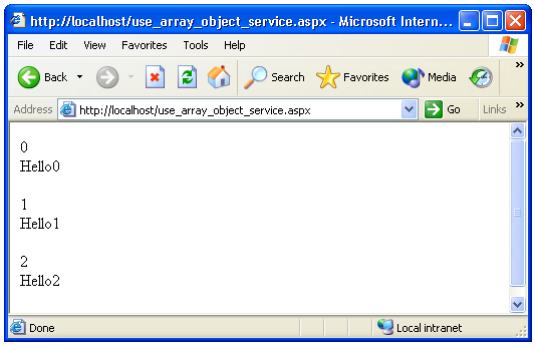
บรรทัดที่ 7 ใช้เมธอด GetObject() ซึ่งจะคืนค่าเป็นอาร์เรย์

บรรทัดที่ 9 ทำงานวนลูปตั้งแต่ค่า i เป็น 0 ถึงเลขดัชนีสมาชิกตัวสุดท้ายในอาร์เรย์ (UBound()) ซึ่งจะทำงานทั้งสิ้น 3 รอบ

บรรทัดที่ 10-11 กำหนดค่าให้กับพรอพเพอร์ตี้ในสมาชิกตัวที่ i ของอาร์เรย์

บรรทัดที่ 12-13 แสดงค่าในพรอพเพอร์ตี้ของสมาชิกตัวที่ i ของอาร์เรย์

เมื่อพิมพ์โค้ดเรียบร้อยแล้ว ให้บันทึกไว้ใน C:\Inetpub\wwwroot\use_array_object_service.aspx แล้วทดสอบในบราวเซอร์โดยพิมพ์ URL เป็น http://localhost/use_array_object_service.aspx จะได้ผลลัพธ์ดังนี้



การคืนค่าข้อมูลชนิด DataSet

เนื่องจากองค์ประกอบของ DataSet นั้นเป็น XML อยู่แล้ว ดังการการส่งผ่านทางเว็บเซอร์วิส ซึ่งเป็น XML เหมือนกันจึงสามารถกระทำได้โดยไม่ต้องสงสัย ซึ่ง DataSet นี้ใช้ในกรณีที่เราต้องการคืนค่าข้อมูลจากฐานข้อมูล

ตัวอย่าง dataset_service.asmx

ตัวอย่างนี้เราจะสร้างเมธอดที่ดึงข้อมูลจากการคิวรี่ฐานข้อมูลชื่อ mydb.mdb (ซึ่งเก็บไว้ในไดเร็กทอรี่เดียวกัน) มาเก็บไว้ใน DataSet จากนั้นคืนค่า DataSet นั้น

```
1:  <%@ WebService language="VB" class="MyService" %>
2:  Imports System.Web.Services
3:  Imports System.Data
4:  Imports System.Data.OleDb
5:  Public Class MyService
6:      Inherits System.Web.Services.WebService
7:      <WebMethod> Public Function GetDataSet() As DataSet
8:          Dim myconn As New OledbConnection _
9:              ("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & _
10:             Server.MapPath("mydb.mdb"))
11:         Dim myda As New OledbDataAdapter _
12:             ("Select * from Books,Authors Where Books.AuthorId = Authors.AuthorId " & _
13:             "Order By Books.BookID",myconn)
14:         Dim ds As New DataSet()
15:         myda.Fill(ds, "tempTable")
16:         Return ds
17:     End Function
18: End Class
```

บรรทัดที่ 3	นำเข้าเนมสเปซที่จำเป็นในการใช้งาน DataSet
บรรทัดที่ 4	นำเข้าเนมสเปซที่จำเป็นในการใช้งานฐานข้อมูล Microsoft Access
บรรทัดที่ 7	สร้างเมธอดที่ไม่ต้องการพารามิเตอร์ และคืนค่าเป็นข้อมูลชนิด DataSet
บรรทัดที่ 8-10	สร้างออบเจกต์ myconn สำหรับเชื่อมต่อกับฐานข้อมูล mydb.mdb ซึ่งอยู่ในไดเร็กทอรี่เกี่ยวกัน
บรรทัดที่ 11-13	สร้างออบเจกต์ myda เพื่อเก็บข้อมูลที่ดึงมาจาก myconn ด้วยประโยค SQL ที่ระบุ
บรรทัดที่ 14	สร้างออบเจกต์ ds ซึ่งเป็นอินสแตนซ์ของคลาส DataSet
บรรทัดที่ 15	สั่งให้นำข้อมูลจากออบเจกต์ myda ไปเก็บไว้ในตารางชื่อ tempTable ในออบเจกต์ ds
บรรทัดที่ 16	คืนค่าออบเจกต์ ds ซึ่งเป็นชนิด DataSet

หลังจากพิมพ์โค้ดเสร็จแล้ว ให้เข้าไปแก้ไขไฟล์ makedll.bat ดังคำสั่งข้างล่าง (โปรดสังเกตด้วยว่ามีการเพิ่ม ,System.data.dll เข้าไปด้วย) แล้วดับเบิ้ลคลิ้กที่ไฟล์ makedll.bat เพื่อรันคำสั่งภายใน จากนั้นให้ย้ายไฟล์ .dll ที่ได้ ไปยังไดเร็กทอรี่ C:\Inetpub\wwwroot\bin

```
wsdl /l:vb /o:dataset_service.vb http://localhost/dataset_service.asmx?wsdl
/n:mydataset_service
vbc /out:dataset_service.dll /t:library /r:system.web.services.dll,system.xml.dll,
system.dll,system.data.dll dataset_service.vb
```

ตัวอย่าง use_dataset_service.aspx

ตัวอย่างนี้จะใช้เว็บเซอร์วิสที่สร้างในตัวอย่างที่แล้ว โดยรับ DataSet ที่คืนค่า มาแสดงลงในคอนโทรล DataGrid (โปรดอย่าลืมว่าก่อนทดสอบตัวอย่างนี้ต้องมีไฟล์ mydb.mdb อยู่ในไดเร็กทอรี่เดียวกันด้วย ถ้าไม่มีให้สร้างขึ้นเองโดยภายในจะประกอบด้วยตารางใดๆ ก็ได้)

```
1:  <%@ Page Language="VB" %>
2:  <%@Import Namespace="mydataset_service"%>
3:  <%@Import Namespace="System.Data"%>
4:  <script runat="server">
5:  Sub Page_Load(Sender As Object,E As EventArgs)
6:      Dim Service As New MyService
7:      Dim objDs As DataSet
8:      objDs = Service.GetDataSet()
9:      mydatagrid.DataSource=objDs.Tables("tempTable")
10:     mydatagrid.DataBind()
11: End Sub
12: </script>
13: <html>
14:     <head><title></title></head>
15:     <body>
16:         <asp:DataGrid id="mydatagrid" runat="server" />
17:     </body>
18: </html>
```

บรรทัดที่ 3 นำเข้าเนมสเปซ System.Data เพื่อการใช้งาน DataSet

บรรทัดที่ 7 ประกาศออบเจกต์ชื่อ objDs ชนิด DataSet เพื่อรอรับค่าที่คืนจากเมธอด

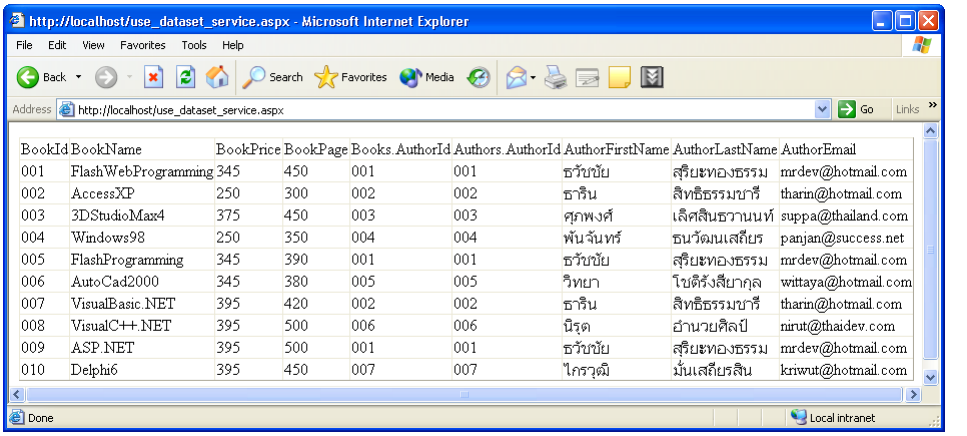
บรรทัดที่ 8 เรียกใช้เมธอด GetDataSet แล้วนำค่าที่คืนมาให้ซึ่งเป็นข้อมูลชนิด DataSet บรรจุลง objDs

บรรทัดที่ 9 กำหนดแหล่งข้อมูลในกับคอนโทรล DataGrid ซึ่งเอามาจากตารางชื่อ tempTable ใน DataSet นั่นเอง

บรรทัดที่ 10 สั่งผูกข้อมูล หรือนำข้อมูลใน DataSet มาแสดงลงใน DataGrid นั่นเอง

บรรทัดที่ 16 สร้างคอนโทรล DataGrid ชื่อ mydatagrid

เมื่อพิมพ์โค้ดเรียบร้อยแล้ว ให้บันทึกไว้ใน C:\Inetpub\wwwroot\use_dataset_service.aspx แล้วทดสอบในบราวเซอร์โดยพิมพ์ URL เป็น http://localhost/use_dataset_service.aspx จะได้ผลลัพธ์ดังนี้



BookId	BookName	BookPrice	BookPage	Books.AuthorId	Authors.AuthorId	AuthorFirstName	AuthorLastName	AuthorEmail
001	FlashWebProgramming	345	450	001	001	ธวัชชัย	สุริยะทองธรรม	mrdev@hotmail.com
002	AccessXP	250	300	002	002	ธาริน	สิทธิธรรมชาวี	tharin@hotmail.com
003	3DStudioMax4	375	450	003	003	ศุภพงศ์	เลิศสินธวานนท์	suppa@thailand.com
004	Windows98	250	350	004	004	พันจันทร์	ธนวัฒนเสถียร	panjan@success.net
005	FlashProgramming	345	390	001	001	ธวัชชัย	สุริยะทองธรรม	mrdev@hotmail.com
006	AutoCad2000	345	380	005	005	วิทยา	โชติรังสียากุล	wittaya@hotmail.com
007	VisualBasic.NET	395	420	002	002	ธาริน	สิทธิธรรมชาวี	tharin@hotmail.com
008	VisualC++.NET	395	500	006	006	นิรุต	อำนวยศิลป์	nirut@thaidev.com
009	ASP.NET	395	500	001	001	ธวัชชัย	สุริยะทองธรรม	mrdev@hotmail.com
010	Delphi6	395	450	007	007	ไกรวุฒิ	มั่นเสถียรสิน	kriwut@hotmail.com

ในบทนี้เราได้เรียนรู้หลักการ,ประโยชน์ และวิธีการพัฒนาเว็บเซอร์วิส และได้ทดลองปฏิบัติสร้างและใช้งานกันมาพอสมควร คงพอทำให้มองเห็นภาพลักษณ์ของการใช้งานและสร้างแอพพลิเคชั่นในยุคเว็บเซอร์วิสได้ดีขึ้นบ้างแล้ว จะเห็นได้ว่าเอกสาร ASP.NET ที่เรียกใช้งานเว็บเซอร์วิสนั้น สามารถลดการทำงานที่ซับซ้อนได้มาก คล้ายกับการเรียกใช้งานคลาสหรือคอนโทรลต่างๆ ใน ASP.NET โดยการทำงานที่ซับซ้อนนั้น จะเป็นภาระของผู้พัฒนาและให้บริการเว็บเซอร์วิส ซึ่งจะเห็นได้ว่าเว็บเซอร์วิสคงจะกลายเป็นอีกช่องทางทำเงินหนึ่งของธุรกิจซอฟต์แวร์ในอนาคตอย่างแน่นอน