CHAPTER 11

Implementing Web Services and the UDDI

THERE HAS BEEN a great deal of hype in the business world concerning Web services. You have probably heard all about them recently, you may have read an article about them. The truth is that I have heard as many different explanations for what a Web service is as people I have heard try to explain what it is. This chapter gives you a solid definition (at least, my version) of what a Web service is and a practical overview of how to implement one. This chapter covers how to create a Web service and how to incorporate the code that you already created for the rest of the NorthwindTraders application. It also covers how to access that Web service from another Windows forms application. You will also see you how to create classes based on a Web service using the Web Service Description Language (WSDL) utility that ships with .NET. This chapter does not cover Web service security, which is an extremely large topic, or the many competing standards and organizations pushing for those standards.

NOTE You can find several resources in Appendix B that will point you in the direction of the different standards bodies working toward unified Web service standards.

The second half of this chapter looks at the UDDI that ships with .NET Enterprise Server 2003. It covers setting up the UDDI, adding Web services to the UDDI, and programmatically accessing the UDDI Server.

What Are Web Services?

In its simplest form, a Web service is a Remote Procedure Call (RPC). In other words, it is a method invocation across a process boundary. That is the extent of the similarities between a Web service and a standard RPC. What differentiates Web services is that the call is made using the Hypertext Transfer Protocol (HTTP) and the request is made and received using the Simple Object Access Protocol.
(SOAP) format. This format is essentially an Extensible Markup Language (XML)—like document format, as shown in Listing 11-1.

**Listing 11-1. A Web Service Request in SOAP Format**

```xml
POST /webservice1/service1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/GetAllEmployees"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
>
  <soap:Body>
    <GetAllEmployees xmlns="http://tempuri.org/"/>
  </soap:Body>
</soap:Envelope>
```

This is a sample of what the SOAP request would look like if you needed to create the SOAP data to send to the Web service yourself.

---

**NOTE** What makes XML Web services so powerful is that anybody who can write a properly formatted SOAP message can call a Web service. You do not need to use a powerful language such as C#, Visual Basic, or Java. This is what makes Web services a "universal" way of sending and receiving data. Anything that can process text can send and receive SOAP messages.

---

The part of the SOAP message to note in Listing 11-1 is the SOAP body tag, which contains the actual call to the GetAllEmployees method. You will walk through creating this Web service later in this chapter (in the “Creating the GetAllEmployees Web Service” section). Listing 11-2 shows an example of the SOAP response format that will be returned by the method call.