

# *Service Oriented Architectures*

mi

Xm1

**SOAP**

- *"What is SOAP if not basically a more object-oriented, somewhat buzzword-compliant upgrade to CGI?"*
  - Anon., Internet
- *"...I think that this is what the grown up web was meant to be, combining the best that HTTP has to offer and great server operating systems, and the best that the desktop has to offer in tools and user interfaces."*
  - Dave Winer, CEO, UserLand

- Microsoft is falling firmly behind SOAP
  - *“The purpose of SOAP is to enable rich and automated Web services based on a shared and open Web infrastructure.”*
    - Microsoft talks about *“the programmable web”*
  - SOAP will underlie much of .NET
    - ASP.NET; Visual Basic.NET, etc.
    - BizTalk server
    - Visual Studio.NET will make it possible to produce SOAP services and clients

*“To be a player in  
the .NET  
development world  
you must  
understand XML  
and SOAP.”*

# Visual Studio.NET and C#

The screenshot shows the Visual Studio.NET IDE with the following components:

- Menu Bar:** File, Edit, View, Project, Build, Debug, Tools, Window, Help.
- Toolbox:** Contains various ASP.NET controls and web services.
- Server Explorer:** Displays a tree view of servers and services, including Data Connections, Servers, Crystal Services, Event Logs, Loaded Modules, Management Data, Message Queues, Performance Counter, Processes, Services, SQL Server Databases, and Web Services.
- Config.web\* (XML View):**

```

<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system>
    <compilation debugmode="true" />
    <!-- CUSTOM ERROR MESSAGES
    Set mode enable="on"
    <error> tags for each error -->
    <customerrors mode="Off"/>
    <!-- SECURITY
    This section sets the security mode to "Passport" and "None" -->
    <security>
      <authentication mode="None" />
    </security>
    <!-- APPLICATION-LEVEL TRACE
    Application-level tracing
    Set trace enabled="true"
    trace information will be sent to the
    application trace listener at the
    root. -->
    <trace enabled="false" requestLimit="65536" />
    <!-- SESSION STATE SETTINGS
    By default ASP.NET uses session state
    If cookies are not available,
    To disable cookies,
    <sessionstate inproc="true" />
    <!-- GLOBALIZATION
    This section sets the globalization
  </system>
</configuration>

```
- WebService1.cs (C# View):**

```

using System;
using System.Web;
using System.Web.Services;
using System.Data;
using System.Data.SqlClient;

public class WebService1 : System.Web.Services.WebService
{
    private System.Data.SqlClient.SqlDataAdapter sqlDataAdapter1;

    public WebService1()
    {
        // Required method for Designer support -
        // the contents of this method will not be
        // used by the compiler.
        private void InitializeComponent()
        {
            sqlDataAdapter1 = new System.Data.SqlClient.SqlDataAdapter();
        }

        protected void sqlDataAdapter1_RowUpdated(object sender,
            System.Data.SqlClient.SqlRowUpdatedEventArgs e)
        {
            // Clean up any resources being used.
        }

        public override void Dispose()
        {
            // Clean up any resources being used.
        }

        //WEB SERVICE EXAMPLE
        //The HelloWorld() example service returns
        //To build, uncomment the following line
        //To test, right-click the Web Service
        //
        //[[WebMethod]]
        //public string HelloWorld()
        //{
        //    return "Hello World";
        //}
    }
}

```
- Solution Explorer:** Shows the project structure for 'WebService' (1 project), including 'References', 'AssemblyInfo.cs', 'Config.web', 'Global.asax', 'WebService.disco', and 'WebService1.aspx'.
- Status Bar:** Ready, Ln 28 Col 25 Ch 25 INS.

```
package au.com.transentia;

import java.io.*;
import java.util.*;
import java.net.*;
import org.apache.soap.*;
import org.apache.soap.rpc.*;

public class GreetingClient
{
    private static PrintWriter
        out = new PrintWriter (System.out, true),
        err = new PrintWriter (System.err, true);

    public static void main (String [] args) throws Exception
    {
        if (args.length != 2)
        {
            err.println ("Usage: java " + GreetingClient.class.getName () +
                " SOAP-router-URL name-to-greet");
            System.exit (1);
        }

        // Build the SOAP RPC call.
        Call call = new Call ();
        call.setTargetObjectURI ("urn:GreetingService");
        call.setEncodingStyleURI (Constants.NS_URI_SOAP_ENC);
        call.setMethodName ("getGreeting");
        Vector params = new Vector ();
        params.addElement (new Parameter("who", String.class, args [1], null));
        call.setParams (params);

        Response resp = call.invoke (new URL (args [0]), "");

        if (resp.generatedFault ())
        {
            Fault fault = resp.getFault ();
            err.println (" Fault Code   = " + fault.getFaultCode ());
            err.println (" Fault String = " + fault.getFaultString ());
            System.exit (2);
        }

        Parameter result = resp.getReturnValue ();
        out.println (result.getValue ());
    }
}
```

```
package au.com.transentia;

public class GreetingService
{
    public String getGreeting (String who)
    {
        return "[" + new java.util.Date () +
            "]" hello " + who);
    }
}
```

- **Performance**

- size

<double> 3.141592653589793E+000 </double>

- *In general is about 10 times the size of binary representations*
      - E.g. sending this 8-byte double in XML, requires 40 bytes of data. If Unicode is used, this doubles!
    - *IBM estimates an average message size of 60K!*

- Speed

- *Serializing Java objects takes...approximately ten times more memory than the binary representation....Serialization and deserialization speeds...are approximately 100 times slower and their throughputs are also a 100 times lower*

- **Security**

- Several commentators have expressed unease at SOAP: they worry about permitting arbitrary communication across a firewall
    - First we close all ports except 80, then we allow arbitrary objects to pass through port 80...
      - Note, however, that we are not passing the actual executable code. This may help us salvage our security...text/xml is easier to grok than binary executables

# Generic SOAP Client

Generic SOAP Client - Microsoft Internet Explorer

File Edit View Favorites Tools Help

## Generic SOAP Client

### Enter SOAP Request

Either choose one of the pre-configured test endpoints or manually enter the SOAP request data below.

**Test-Endpoints:** [show descriptions]  
<http://soap.develop.com/soapdemo/page.asp> (Translate)

**Endpoint:**  
<http://soap.develop.com/soapdemo/page.asp>

**Interface:**  
 soap:cdl:com.develop.soapdemo.VBSoapSrv\_VBSoapTest

**Method:**  
 Translate

**Payload (XML):**

```
<Translate>
  <p type="soap:cdl:com.develop.soapdemo.VBSoapSrv.point">
    <x type="i4">200</x>
    <y type="i4">500</y>
  </p>
</Translate>
```

Generic SOAP Client - Microsoft Internet Explorer

File Edit View Favorites Tools Help

## SOAP Results

### Original SOAP Request

**Request Headers**

```
POST http://soap.develop.com/soapdemo/page.asp HTTP/1.1

Method: Translate
InterfaceName:
soap:cdl:com.develop.soapdemo.VBSoapSrv_VBSoapTest
MessageType: Call
Content-Type: text/xml-SOAP

[some HTTP headers left out for simplicity]
```

**Request Payload:**

```
<Translate>
  <p type="soap:cdl:com.develop.soapdemo.VBSoapSrv.point">
    <x type="i4">200</x>
    <y type="i4">500</y>
  </p>
</Translate>
```

Done Internet

Generic SOAP Client - Microsoft Internet Explorer

File Edit View Favorites Tools Help

## SOAP Response

**Response Headers**

```
Status: 200 OK

Server: Microsoft-IIS/5.0
Date: Mon, 01 Nov 1999 17:24:53 GMT
MessageType: CallResponse
Content-Length: 532
Content-Type: text/xml-SOAP
Set-Cookie: ASPSESSIONIDQGGQJEU=EFBCFLHCCIHHEGHNPNBLPLPO; path=/
Cache-control: private
```

**Response Payload:**

```
<?xml version="1.0" ?>
<soap:SerializedStream xmlns:soap="http://v3.org/Schemas/SOAP/kv"
  soap:headers="#x1" soap:main="#x2" soap:serializationPattern="urn:schemas-
  microsoft-com:soap:v1">
  <soap:headers soap:id="x1">
    <soap:InterfaceName soap:type="string">
      soap:cdl:com.develop.soapdemo.VBSoapSrv_VBSoapTest</soap:InterfaceName>
    </soap:headers>
    <TranslateResponse soap:id="x2">
      <p soap:type="soap:cdl:com.develop.soapdemo.VBSoapSrv.point">
        <x soap:type="i4">202</x>
        <y soap:type="i4">503</y>
      </p>
      </TranslateResponse>
    </soap:SerializedStream>
```

Aaron Skonnard <http://www.skonnard.com>

Done Internet



**Web Services**

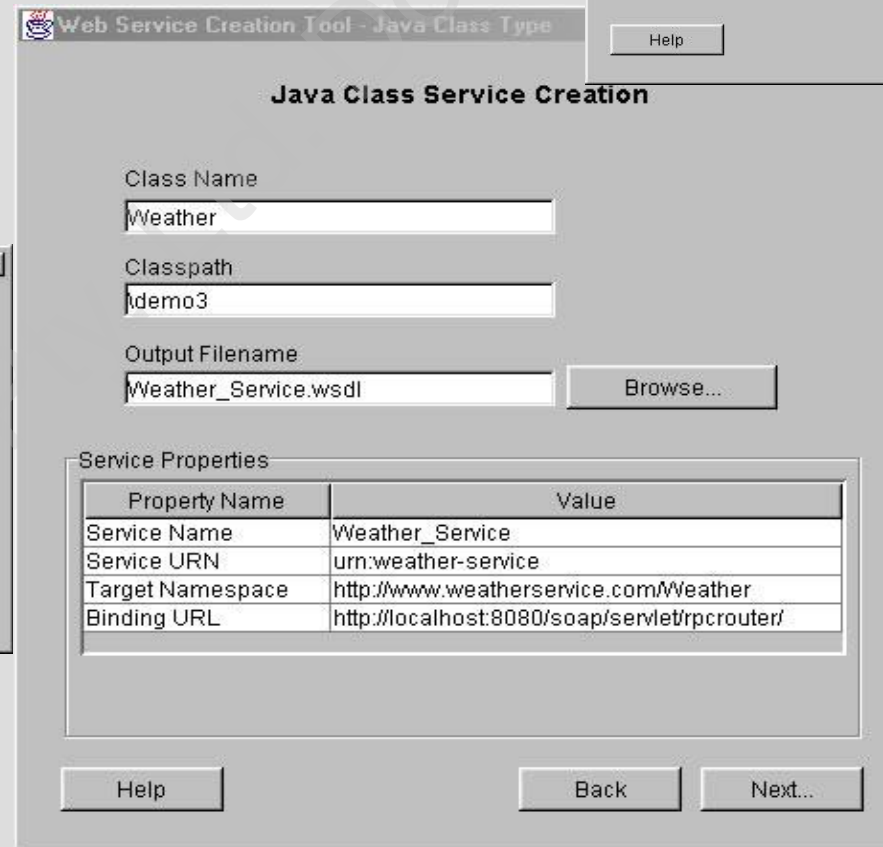
- **WebServices**

- *“A Web Service is a unit of application logic providing data and services to other applications. Applications access Web Services via ubiquitous Web protocols and data formats such as HTTP, XML, and SOAP, with no need to worry about how each Web Service is implemented. Web Services combine the best aspects of component-based development and the Web...”*
  - *A way of specifying an addressable entity: what it does, where it is, how to invoke it, etc...*
- *“Webservices are pretty much guaranteed to be at the heart of the next generation of distributed systems”*
  - **Graham Glass, The Mind Electric**
- *“Four years ago, we all saw the rush to develop Web sites on the Internet. Today, we will witness the rush to set up Web Services over the Internet.”*

- **WebServices Definition Language**

- The XML equivalent of a resumé
  - *Describes what a service can do, where it is, and how to invoke it*
- Since WSDL fully defines a service, it can be used to auto-generate RPC stub/marshalling/proxying code
  - *Most toolkits allow for this*
- *“WSDL can be seen as a complement to SOAP, as it facilitates interoperability between Web services. Like IDL (Interface Definition Language), which acts as a service describer with CORBA, WSDL (Web Service Description Language) is an XML syntax to describe Web services. The specifications for WSDL come from a joint initiative by Microsoft, IBM and Ariba. More and more SOAP implementations support this description language; with WSDL, applications that use SOAP can self-configure exchanges between Web services, while hiding most of the low-level technical details.”*
  - *ALAIN LEFEBVRE, Vice President, Groupe SQLI (TechMetrix)*

- Various
  - IBM (shown)
    - *IBM WSDL Toolkit*
    - *IBM WSTK*
  - Visual Studio.NET



- **Brokers**
  - Xmethods
  - Biztalk
  - ebXML
  - RosettaNET
  - Etc.
  - “the nice thing about standards is that there are so many to choose from...”

The screenshot shows the XMethods website in a Microsoft Internet Explorer browser. The page title is "XMethods - Web Service Listings". The address bar shows "http://www.xmethods.com".

The main content area features a "SOAP Service List" table with columns: Owner, Status, Service Name, Description, and Server. The table lists various services such as "Exchange via SOAP", "Pressure Converter", "Volume Converter", "Weight Converter", "Distance Converter", "Temperature Converter", "Text Messaging", "DOVESPA Stock Quotes", "Alan Bush Compositions", "Web Service Search", "Email", "WebStorageSystem", and "MailToTheFuture".

Below the table, there is a detailed view of the "CurrencyExchange" service. The "Service Owner" is "xmethods". The "Contact Email" is "support@xmethods.net". The "Description" states: "Exchanges rate between any two currencies. Takes in country1 and country2, and returns the exchange rate between currencies (Returns value of 1 unit of country1's currency converted into country2's unit currency)".

The "SOAP Endpoint URL" is "http://services.xmethods.net/Exchange". The "SOAP Action" is "None Needed". The "Method Namespace URI" is "urn:xmethods-CurrencyExchange". The "Method Name(s)" is "getRate". The "WSDL URL" is "http://services.xmethods.net/soap/xmethods-CurrencyExchange.wsdl".

The "Request/Response" section shows the parameter schemas for the "getRate" method. The request parameters are "country1" (type="string") and "country2" (type="string"). The response parameter is "rate" (type="float").

The "Encoding Style" is "http://schemas.xmlsoap.org/soap/encoding".

The "Sample Request Envelope" is shown as an XML document:

```
<?xml version='1.0' encoding='utf-8'>
<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'>
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <xsi:type='xsd:string'>urn:xmethods-CurrencyExchange SOAP-ENV:encodingStyle='http://schemas.xmlsoap.org/soap/encoding'>
      <country1 xsi:type='xsd:string'>England</country1>
      <country2 xsi:type='xsd:string'>Japan</country2>
    </xsi:type>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The "Sample Response Envelope" is also shown as an XML document:

```
<?xml version='1.0' encoding='utf-8'>
<SOAP-ENV:Envelope xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'>
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <xsi:type='xsd:float'>100.0</xsi:type>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

# Example: WSDL File

```
<?xml version = "1.0"?>
<definitions name = "CurrencyExchangeService"
    targetNamespace = "http://www.xmethods.net/sd/CurrencyExchangeService.wsdl"
    xmlns:xsd = "http://www.w3.org/1999/XMLSchema"
    xmlns:soap = "http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns = "http://schemas.xmlsoap.org/wsdl/">
    <message name = "getRateRequest">
        <part name = "country1" type = "xsd:string"/>
        <part name = "country2" type = "xsd:string"/>
    </message>
    <message name = "getRateResponse">
        <part name = "return" type = "xsd:float"/>
    </message>
    <portType name = "CurrencyExchangePortType">
        <operation name = "getRate">
            <input message = "getRateRequest" name = "getRate"/>
            <output message = "getRateResponse" name = "getRateResponse"/>
        </operation>
    </portType>
    <binding name = "CurrencyExchangeBinding" type = "CurrencyExchangePortType">
        <soap:binding style = "rpc" transport = "http://schemas.xmlsoap.org/soap/http"/>
        <operation name = "getRate">
            <soap:operation soapAction=""/>
            <input>
                <soap:body use = "encoded" namespace = "urn:xmethods-CurrencyExchange"
                    encodingStyle = "http://schemas.xmlsoap.org/soap/encoding"/>
            </input>
            <output>
                <soap:body use = "encoded" namespace = "urn:xmethods-CurrencyExchange"
                    encodingStyle = "http://schemas.xmlsoap.org/soap/encoding"/>
            </output>
        </operation>
    </binding>
    <service name = "CurrencyExchangeService">
        <documentation>Returns the exchange rate between the two currencies</documentation>
        <port name = "CurrencyExchangePort" binding = "CurrencyExchangeBinding">
            <soap:address location = "http://services.xmethods.net:80/soap"/>
        </port>
    </service>
</definitions>
```

**xm1**

**UDDI**

Transentia Pty Ltd. DonationWare

# • Universal Description, Discovery and Integration

- *“A common set of SOAP APIs that enable the implementation of a service broker”*

Interop Stack	Universal Service Interop Protocols (these layers are not defined yet)
	Universal Discovery, Discription, Integration (UDDI)
	Simple Open Access Protocol (SOAP)
	Extended Markup Language (XML)
	Common Internet Protocols (HTTP, TCP/IP)

- *“The power of UDDI is the power of ad-hoc discovery of new business partners and processes ....UDDI lets companies do three things:*
  - Discover each other;
  - Define how they can interact via the Internet; and
  - Share all this information via an open, global registry.”
- *‘UDDI is a “next layer” in an emerging stack enabling rich Web Services. UDDI uses standards-based technologies such as TCP/IP, HTTP, XML and SOAP to create a uniform service description format and service discovery protocol.’*
  - *UDDI.org Technical white paper*
- *“What will restrict the success of a directory like UDDI is the same thing that handicapped marketplaces: the real needs of enterprises. It's more about working well with partners you know, rather than finding new ones.”*
- *“UDDI’s Problem: Technology Cannot Replace Relationships”*
  - *KEN VOLLMER , InternetWeek*



# Simple Java Find Example

```
import com.ibm.uddi.*;
import com.ibm.uddi.datatype.business.*;
import com.ibm.uddi.response.*;
import com.ibm.uddi.client.*;
import org.w3c.dom.Element;
import java.util.Vector;
import java.util.Properties;

public class FindBusinessExample {
    public static void main (String args[]) {
        System.setProperty("java.protocol.handler.pkgs", "com.sun.net.ssl.internal.www.protocol");
        java.security.Security.addProvider(new com.sun.net.ssl.internal.ssl.Provider());
        new FindBusinessExample().run();
        System.exit(0);
    }
    public void run() {
        // Construct a UDDIProxy object. Enables SSL support
        UDDIProxy proxy = new UDDIProxy();
        try {
            proxy.setInquiryURL("http://www-3.ibm.com/services/uddi/testregistry/inquiryapi");
            // Find businesses who's name starts with S
            BusinessList bl = proxy.find_business("S", null, 0);
            Vector businessInfoVector = bl.getBusinessInfos().getBusinessInfoVector();
            for (int i = 0; i < businessInfoVector.size(); i++) {
                BusinessInfo businessInfo = (BusinessInfo)businessInfoVector.elementAt(i);
                System.out.println(businessInfo.getNameString());
            }
            // Handle possible errors
        } catch (UDDIException e) {
            DispositionReport dr = e.getDispositionReport();
            if (dr!=null) {
                System.err.println("UDDIException faultCode:" + e.getFaultCode() + "\n operator:" +
                    dr.getOperator() +
                    "\n generic:" + dr.getGeneric() + "\n errno:" + dr.getErrno() + "\n errCode:" +
                    dr.getErrCode() +
                    "\n errInfoText:" + dr.getErrInfoText());
            }
            e.printStackTrace(System.err);
            // Catch any other exception that may occur
        } catch (Exception e) {
            e.printStackTrace(System.err);
        }
    }
}
```