

---

# Writing Simple UDDI Clients

---

## Accessing WebServices Directories

Developed by:



Unit 1, 40 Skew St.  
Sherwood  
Queensland 4075  
Australia

Mobile: 0416 078 813  
Phone/FAX: (07) 3278 5890  
Email: [contact@transentia.com.au](mailto:contact@transentia.com.au)  
WWW: [www.transentia.com.au](http://www.transentia.com.au)

ACN: 091-142-880  
ABN: 680-911-428-80

---

## Writing Simple UDDI Clients

### Accessing WebServices Directories

#### Introduction

UDDI is being positioned by several major companies as an indispensable business tool. In these sections you will access entries already in business registries.

#### Simple Microsoft VBScript Client

This example shows how to use Microsoft's Windows Scripting Host functionality to access a simple UDDI entry.

#### Install the UDDI Toolkit

In this session you will need:

- Microsoft UDDI SDK v1.5.2

Your instructor will provide you with the appropriate installer.

Installing UDDI is a two-stage process: first the installer will uncompress a number of files and then will present you with a web page so that you can choose which actual UDDI package you require. You should ensure that you install the **Visual Studio 6** version of the UDDI toolkit.

#### Create the Client

Execute the following sequence of commands within a Command Prompt window:

```
> mkdir C:\UDDI\vb  
> cd /d C:\UDDI\vb
```

Create the following in the file **C:\UDDI\vb\uddi.vbs**:

```

Set req = CreateObject("UDDIEnv.RequestManager")
Set env = CreateObject("UDDIEnv.Envelope")
Set findBiz = CreateObject("UDDI10.find_business")
Set env.Plugin = findBiz
findBiz.name = "Java"
Set resp = req.UDDIRequest(env)
Set bizList= CreateObject("UDDI10.businessList")
Set resp.Plugin = bizList
If req.UDDIErrno = 0 Then
    For Each r In bizList.businessInfos
        Set e = CreateObject("UDDIEnv.Envelope")
        Set getBizDetails = CreateObject("UDDI10.get_businessDetail")
        Set e.Plugin = getBizDetails
        getBizDetails.AddbusinessKey().businessKey = r.businessKey
        Set resp = req.UDDIRequest(e)
        If req.UDDIErrno = 0 Then
            Set bizDetail = CreateObject("UDDI10.businessDetail")
            Set resp.Plugin = bizDetail
            With bizDetail.businessEntity(1)
                WScript.Echo "****"
                WScript.Echo "  Name: " & .name
                WScript.Echo "  URL(1): " & .discoveryURLs(1).discoveryURL
                WScript.Echo "  Descr(1): " & .description(1).description
            End With
        End If
    Next
End If

```

## Run the Client

Execute this command within the same Command Prompt window that you used previously:

```
> cscript /nologo uddi.vbs
```

You should be rewarded with a list of all the UDDI entries for businesses whose name begin with the string "Java." The information is taken from Microsoft's UDDI registry at <http://uddi.microsoft.com>.

You can verify the returned information by browsing to the above URL and doing a search for Java.

## A Simple Modification

You can see the XML-based request and response packets that are passed between client and server.

Add the following lines **before line 6** in the above VBScript application:

```

req.Trace = -1
req.TracePath = "."

```

Execute the application as before. Use notepad to examine the two files **find\_business.xml** and **businessList.xml** that were created as a result of the execution. You can also open them directly using Microsoft IE 5+ (if this is installed on your system).

## Using IBM's UDDI4J

IBM supplies a simplified API for accessing data that is being maintained in a UDDI registry. This API is becoming the de facto standard for Java-based UDDI-related development.

### Install UDDI4J

In this portion of the session, you will require the following:

- Sun JDK 1.3.1
- Uddi4j
- Soap 2.2
- Java Activation Framework 1.0.1
- Javamail 1.2
- Xerces 1.4.3

You have been given these on your CD-ROM.

### Configuration

A bit of housekeeping is needed before development can begin.

### Install the JDK

You should ensure that you have installed a version of the Sun Java Development Kit (this should be provided to you on your course CD-ROM). Note where it is installed.

### Create and Populate the UDDI Session Directory

A (mostly) pre-installed and configured set of libraries and auxiliary files has been provided for you on your CD-ROM. There remains a small bit of work, however.

Open a new Command Prompt window and issue the following command to copy the directory Z:\Exercises\2 UDDI\framework to C:\UDDI\java:

```
> xcopy /ieq "Z:\Exercises\2 UDDI\framework" C:\UDDI\java
```

Notes:

- This assumes that your CD-ROM device is Z:, you should use your real device letter instead
- Do not simply drag&drop from the CD-ROM; if you do, you will end up with a read-only directory structure and this will complicate things later on

You should end up with the following directory structure on your workstation's hard disk:

```
C:\UDDI\java
  bin
  bsf-2_2
  jaf-1.0.1
  javamail-1.2
  soap-2_2
  xerces-1_4_3
```

## Edit Utility Script

You have been provided with some utility scripts within C:\SOAP\bin. Using a text editor such as notepad, edit the supplied setpath.bat file as follows: (for *path-to-jdk* use the place where you earlier installed the Java Development Kit):

```
@echo off
set JAVA_HOME=path-to-jdk
set PATH=%PATH%;%JAVA_HOME%\bin
```

This will make your life a lot easier later on.

## Create the Client

Create this Java client in the file C:\UDDI\java\UDDIClient.java:

```
import com.ibm.uddi.*;
import com.ibm.uddi.datatype.business.*;
import com.ibm.uddi.response.*;
import com.ibm.uddi.client.*;
import java.util.*;
import java.io.*;

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

    public static void main (String args[])
    {
        try
        {
            UDDIProxy proxy = new UDDIProxy();
            proxy.setInquiryURL
                ("http://www-3.ibm.com/services/uddi/testregistry/inquiryapi");
```

```

final BusinessList
    java = proxy.find_business("Java", null, 0);

final Vector
    businessInfos = java.getBusinessInfos().
        getBusinessInfoVector();
for (int i = 0; i < businessInfos.size(); i++)
{
    final BusinessInfo
        busInfo = (BusinessInfo) businessInfos.elementAt(i);

    out.println("Business Name : " + busInfo.getNameString());
    out.println("Business Description : " +
        busInfo.getDefaultDescriptionString());
    out.println("Business Key : " + busInfo.getBusinessKey());
}
}
catch (UDDIException ue)
{
    final DispositionReport
        dr = ue.getDispositionReport();
    if (dr!=null)
        err.println("UDDIException faultCode:" +
            ue.getFaultCode() +
            "\n operator:" + dr.getOperator() +
            "\n generic:" + dr.getGeneric() +
            "\n errno:" + dr.getErrno() +
            "\n errCode:" + dr.getErrCode() +
            "\n errInfoText:" + dr.getErrInfoText());
    ue.printStackTrace(err);
}
catch (Exception e)
{
    e.printStackTrace(err);
}
}
}

```

Note that once one has retrieved a business key, it is a fairly simple matter to retrieve an associated WSDL file and thus to start making SOAP calls to the advertised services.

## Compile and Execute the Client

To compile, execute the following sequence of commands in the same command window that you have been using all along:

```

> cd /d C:\UDDI\java
> ..\bin\setpath.bat
> ..\bin\setclasspath.bat
> javac UDDIClient.java

```

To run, use the following:

```
> java UDDIClient
```

Congratulations! You have successfully executed a 'live' query against IBM's UDDI business registry.