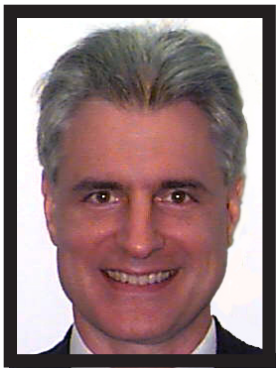
This Goes To Eleven - A Web Services Demonstration

2008

An Oracle "How To" Paper

Presented By:

William Indest



Yogesh Sharma

Geoff Duckwall

ORACLE | SIEBEL

Nigel Tufnel: The numbers all go to eleven. Look, right across the board, eleven, eleven, eleven and...

Marty DiBergi: Oh, I see. And most amps go up to ten?

Nigel Tufnel: Exactly.

Marty DiBergi: Does that mean it's louder? Is it any louder?

Nigel Tufnel: Well, it's one louder, isn't it? It's not ten. You see, most blokes, you know, will be playing at ten. You're on ten here, all the way up, all the way up, all the way up, you're on ten on your guitar. Where can you go from there? Where?

Marty DiBergi: I don't know.

Nigel Tufnel: Nowhere. Exactly. What we do is, if we need that extra push over the cliff, you know what we do?

Marty DiBergi: Put it up to eleven.

Nigel Tufnel: Eleven. Exactly. One louder.

Marty DiBergi: Why don't you just make ten louder and make ten be the top number and make that a little louder?

Nigel Tufnel: [pause, blank look and snapping chewing gum] These go to eleven.

This Is Spinal Tap

Introduction

Some people see things that are and ask, Why? Some people dream of things that never were and ask, Why not? Some people have to go to work and don't have time for all that ...

George Carlin

We've created a Flash-based web services demonstration for you to show how easy it is to invoke Siebel web services, so like, this goes to eleven, you know? In this document we will show you how to enable web services on the Siebel Server, create a WSDL file and where to put it and how to demonstrate how easy it is to call a web service. Simple, easy and elegant.

What is a Web Service?

Here is the Web Services definition from the W3C Working Group Note 11 February 2004:

Definition: A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

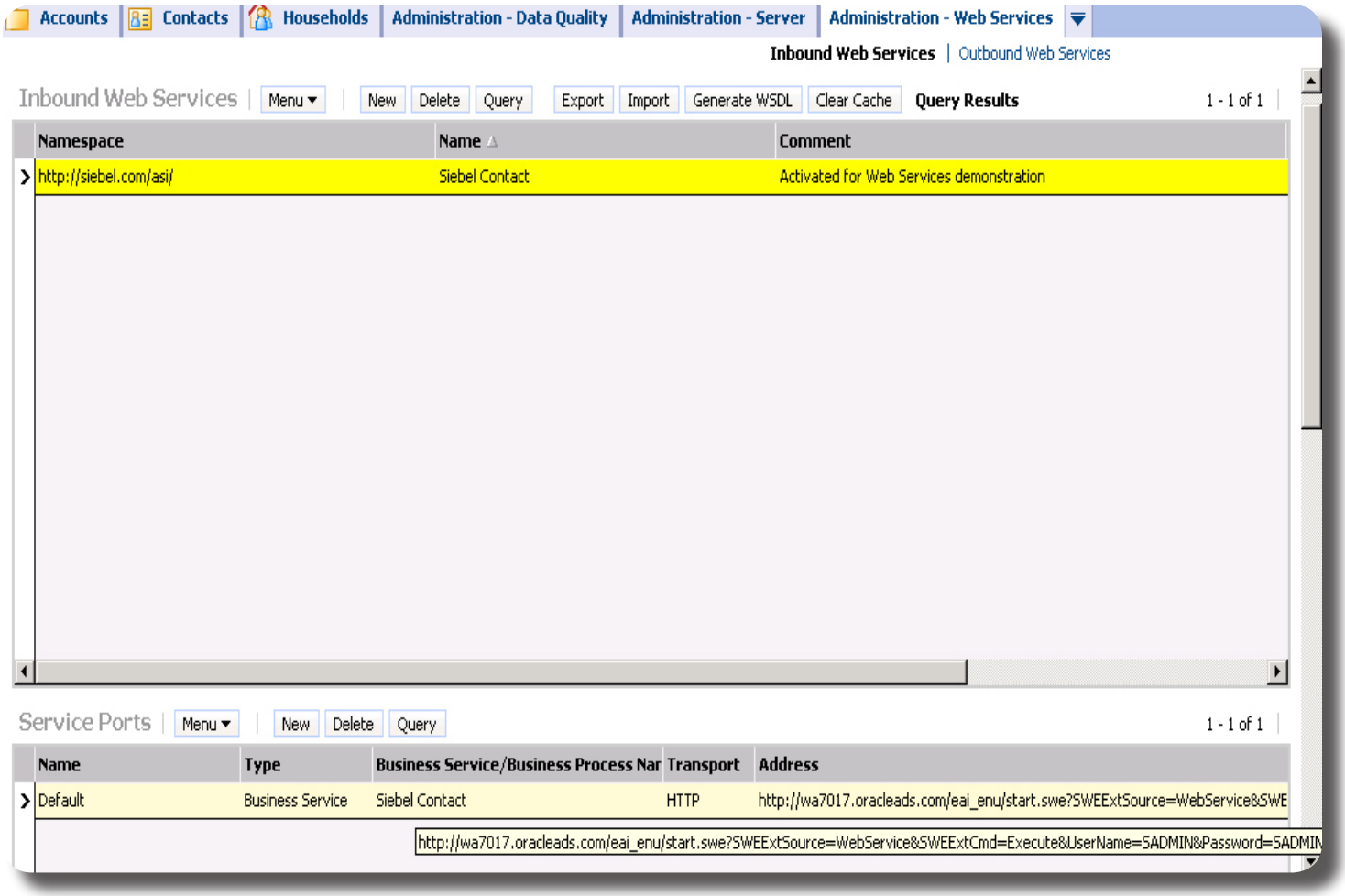
There is no need to write any code for this demo. Your goal during the web service demo is to demonstrate simplicity and ease of use. To make things easy for you, we will be using a Flash-based application that runs in a web browser. Here is how to setup your demonstration.

ORACLE

Setup

Start your server using the Demo Launcher. The EAI Object Manager under Enterprise Application Integration should be checked if you are using the Demo Launcher. If you prefer to not use the Demo Launcher, check to see if the EAI Object Manager is running by going to Administration – Server Configuration > Enterprises. It should be listed under the Component Definitions (query for EAI*) with an Enable State of Enabled.

Once the Server is started, start a thin client session and login as SADMIN / SADMIN. Navigate to the Site Map and select Administration - Web Services > Inbound Web Services. Search for the out-of-the-box web service named: Siebel Contact. Change the status to Active. In the Service Ports section, modify the address field to something like:
http://wa7017.oracleads.com/eai_enu/start.swe?SWEExtSource=WebService&SWEExtCmd=Execute&UserName=SADMIN&Password=SADMIN
with the understanding that you need to replace the server name, *wa7017.oracleads.com*, with the name of your server. Here is a picture of the screen:



The Binding field should be set to:

SOAP_RPC_ENCODED.

Highlight the Siebel Contact record and click on the Generate WSDL button. Save the WSDL file to a place you will soon forget, like the desktop.

Copy the WSDL file, in our example we called it *DemoContact.wsdl*, to the server's web directory, for example, *D:\Inetpub\wwwroot* (this works on a Macintosh, too). Check the availability of the WSDL by opening a browser and typing the following into the address field:

http://myservername/DemoContact.wsdl

(We know it is obvious but we are trying to be helpful here: **myservername** is the name of the box where the Siebel server is running.) You should see the WSDL file appear in the browser.

Highlight and copy the WSDL string location. Open the file, *WSDLLocation.xml*, in the *bin* directory where you have unzipped the ContactByIdWebServiceDemo package (Check Appendix A for the directory structure). Replace the string with the WSDL string. Save the file. Make sure this file is in the same directory as the *ContactByIdWebServiceDemo.swf* file.

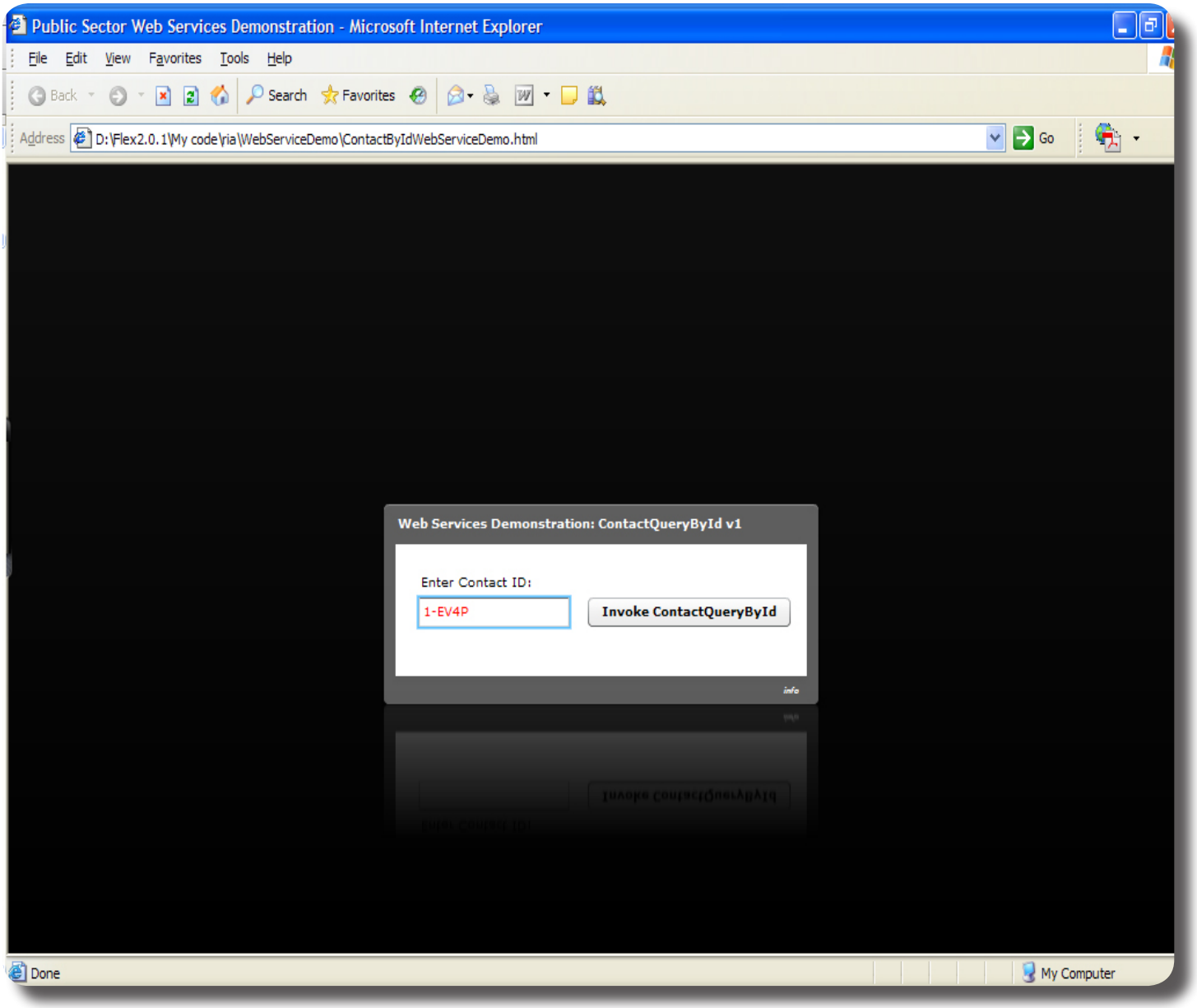
Now you must bounce the EAI Object manager. Navigate to Site Map \ Administration – Server > Enterprises.

In the bottom Applet, highlight the EAI Object Manager row, click on shutdown. Monitor the status of the EAI Object Manager as its shutting down by clickingrepeatedly the Execute Query button in the toolbar. Once the Component is shut down, click the startup button.

Launch the web services demonstration by clicking on *ContactByIdWSDemo.html,* located in the *bin* directory where you unzipped the demonstration package.

You are ready!

Here is the first screen of the web services demonstration application with the Record Id entered, **1-EV4P**.



The next section will show you how to demonstrate this cute little application.

Demo Flow: Ready To Rock?

Siebel supports open standards for Web Services. Web Services support is included in every Siebel Server instance. You can use/customize our out of box web services, or build your own. Let's show you what we mean about web service integration. Let's look at what you can do with our web services out of the box.

We support industry standards SOAP 1.1, and WSDL 1.1. That means that we can talk to applications written in Dot net, Java, and any other web services or integration platform that supports industry standards.

You can make a slide or write these steps on the board. The steps are as follows:

1 – Generate the WSDL in Siebel

2– Make the WSDL available

3 – Invoke the Web Service from an external application.

Launch the Siebel thin-client and find a suitable Contact record to use in your demonstration. You can modify the name to make it unique to the prospect. Click on the help menu item and select About Record and copy the record number.

Say the name of the Contact Name or you can write down the name of the Contact on the board.

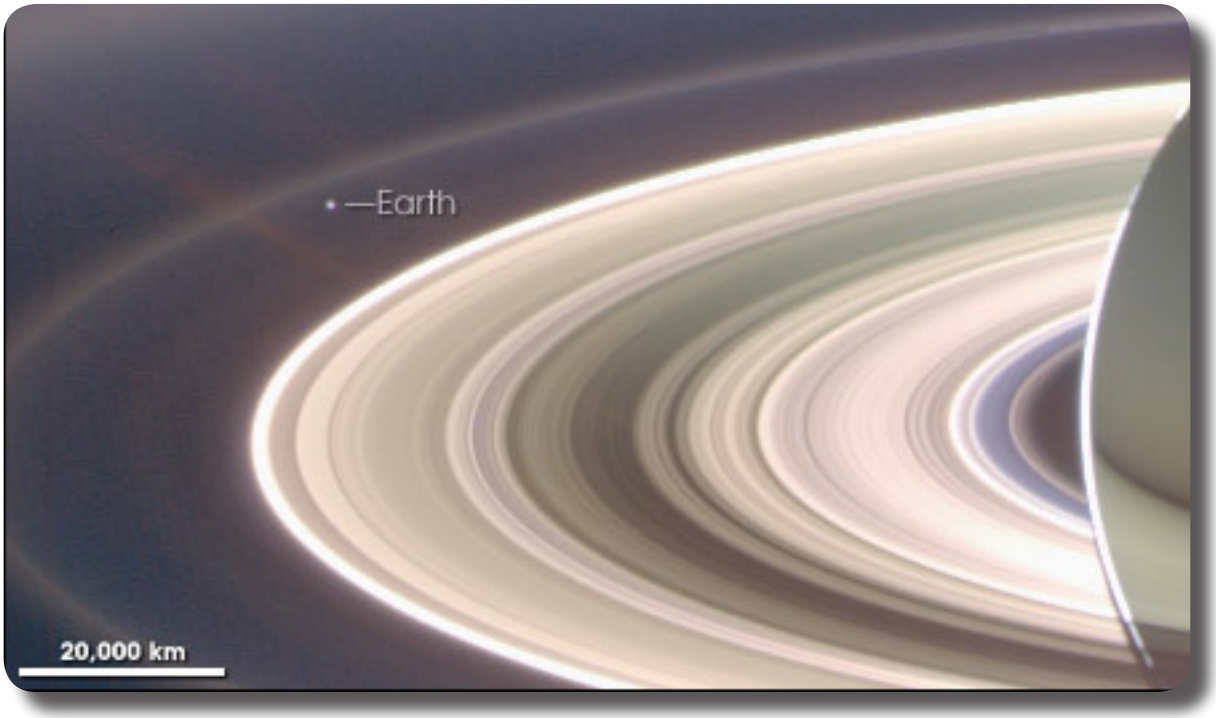
Navigate back to the ContactByIdWSDemo and paste the Row Id (PrimaryRowId argument for the web service) into the Id field. (1-EV4P is a good candidate.)

Click the ContactQueryById button. While the application communicates with the server you can talk about how every object in the Siebel application is web-service enabled and there are several hundred services available out-of-the-box; they just need to be enabled. In a few seconds you will see the response from the Siebel Server.

Change the name and click on the Update button. When the response appears, go back to the Siebel thin-client, refresh the screen and show the updated record. Go back to the ContactWSDemo and click the Return button. Enter the record id again (if necessary) and click on the ContactQueryById button. When you receive the response, click on the show XML button to show your SOAP messages If you are really adventurous you

can right-mouse click in the browser window and select view source and show them how little code was needed to invoke Siebel's web services.

Why does "This go to Eleven"? As you can see from this great photo, we are kinda small and insignificant so we plan to overcome these feelings with something spectacular for your customers to embrace.



Pick a number from one to ten.

Eleven.

Right!

Groucho and Chico Marx in Duck Soup

After your demonstration you should have your customers jumping for joy:

and rushing out to buy a bunch of our software:

We live in a society exquisitely dependent on science and technology, in which hardly anyone knows anything about science and technology.

Carl Sagan

Troubleshooting

If something goes wrong, check the following:

Can you ping the ADS machine?

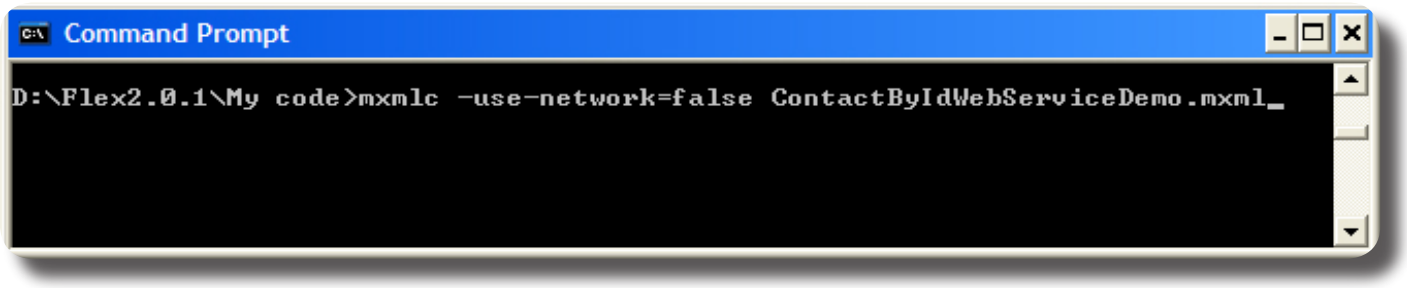
Can you reach the WSDL file using a web browser?

Have you correctly entered the server name into the WSDL-Location.xml file?

Are you using the correct record id – does it exist in the application?

If you still cannot get it to work, launch SoapUI, load the WSDL, open the ContactByIdQuery request, enter in the record id as the PrimaryRecordId and try that. If this works then you can use it to demo web services.

You may get an error which says you do not have permission for cross-domain access (Flash's security model is quite similar to Javascript's security model). Download the free Flex SDK from http://www.adobe.com/flex and run the compiler with the use-network=false switch like this:



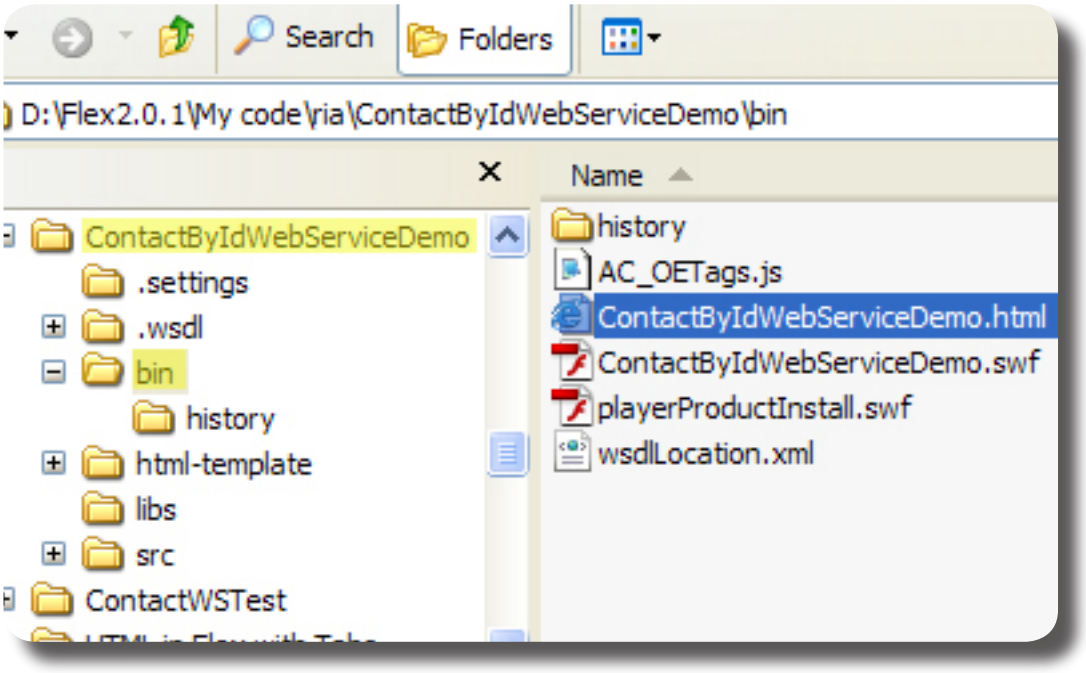
No data – check all of your steps from the beginning. We didn't put a whole lot of error checking in the little app; then it would have been a big app and we would have to change the title to "This goes to twelve" which is not in Wikipedia.

"I don't really think that the end can be assessed as of itself as being the end because what does the end feel like? It's like saying when you try to extrapolate the end of the universe, you say, if the universe is indeed infinite, then how - what does that mean? How far is all the way, and then if it stops, what's stopping it, and what's behind what's stopping it? So, what's the end, you know, is my question to you."

David St. Hubbins, This Is Spinal Tap

Appendix A

Here is the directory structure once you have unzipped the *ContactByIdWebServiceDemo.zip* file in place.



Title: This Goes To Eleven - A Web Services Demonstration
Authors: William Indest, Yogesh Sharma and Geoff Duckwall
Managers: James Perry, Rick Beck and Todd Lesser

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

EOF