

# The Oracle Story

Bob Brown

**t r a n s e n t i a**

[bob@transentia.com.au](mailto:bob@transentia.com.au)

<http://www.transentia.com.au>

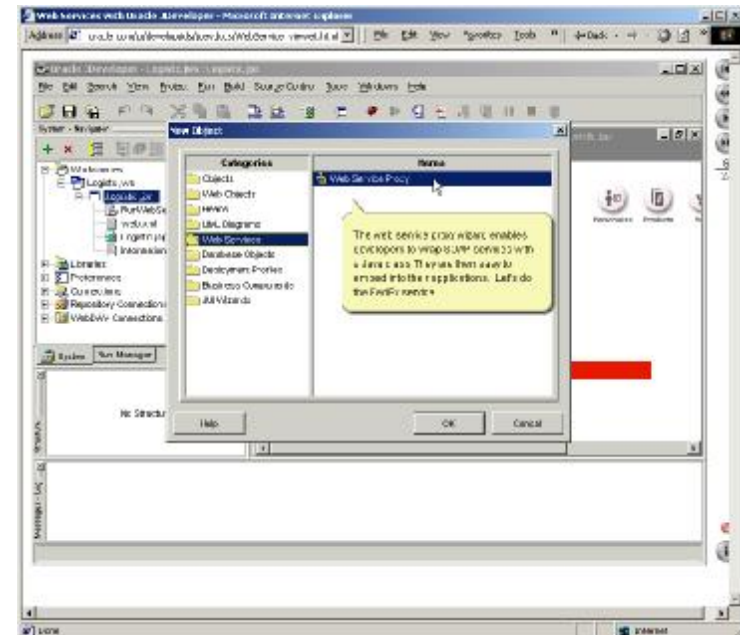
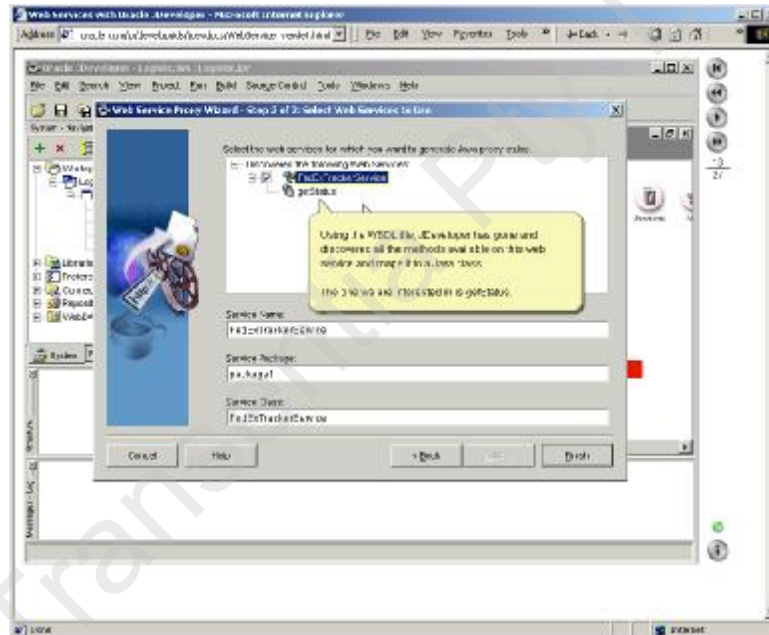
## Strongest “Out of Box” Solution?

- Historically, Oracle offered its “Dynamic Services” technologies
  - With 8.1.x and above
  - Initial work predates SOAP/WSDL/UDDI
  - May have been the best pre-packaged solution
    - “.NOW Web Services Software”
  - Needed to go its own way to get there
- Is now rolling in support for the developing standards as they come up to scratch
  - There is a logical correspondence between Oracle’s initial technology and the current standards activities and technologies
    - Support flows through from old to new easily
- Now treats WebServices as simply another access technology
  - Supported transparently in OC4J/9iAS
  - A common stance

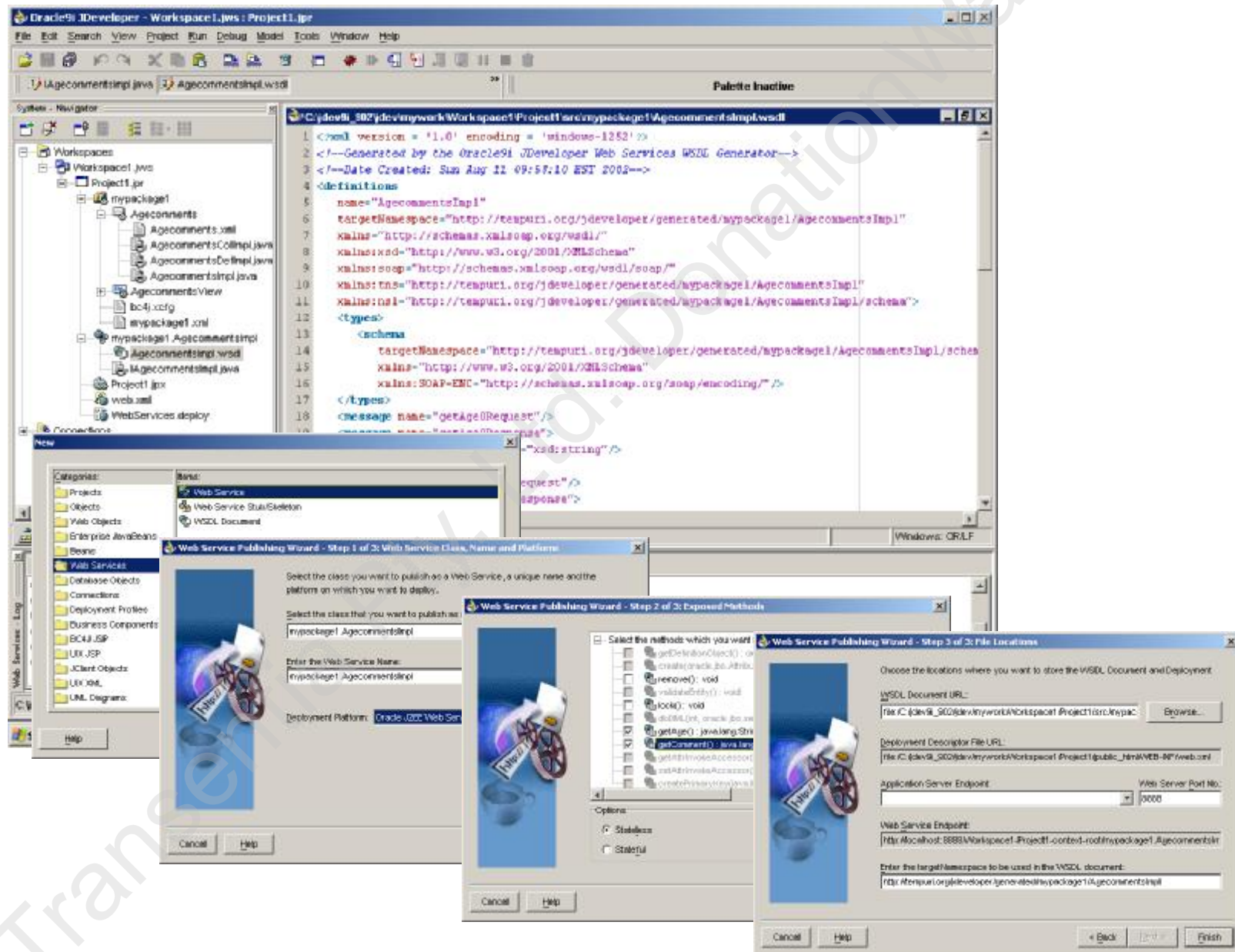
## Three Major Aspects

- JDeveloper 9i
  - Oracle's "flagship IDE"
- OC4J
  - Oracle9i AS Containers for J2EE (OC4J)
    - Provides Apache SOAP 2.2
- Portal
  - Integration/personalisation/e-commerce infrastructure

- Allows almost anything to be turned into a WebService
    - POJO, BC4J component, Servlet, EJB (session, message), PL/SQL procedure
  - Supports all Web Services standards
    - SOAP, WSDL and UDDI
- [http://www.oracle.com/ip/develop/ids/jdevdocs/WebService\\_viewlet.html](http://www.oracle.com/ip/develop/ids/jdevdocs/WebService_viewlet.html)



## • Wizard wonderland...



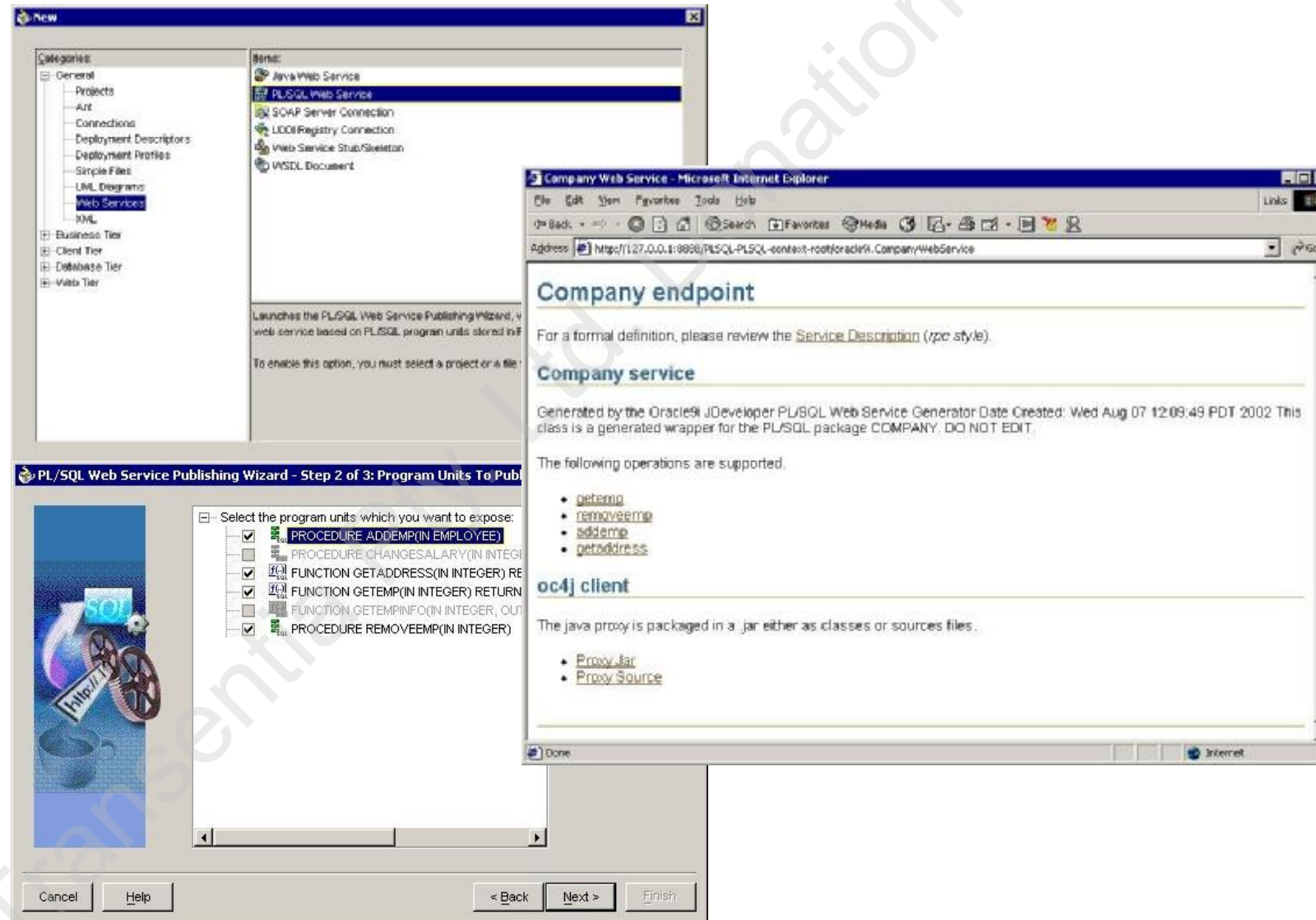


- Example JSP: check an item on ebay

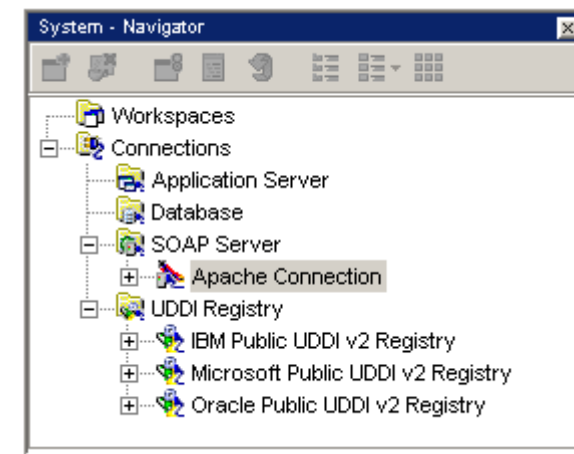
```
<%@ page contentType="text/html"%>
<%@ taglib uri="http://xmlns.oracle.com/j2ee/jsp/tld/ws/wstaglib.tld"
      prefix="ws" %>

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; ">
</HEAD>
<BODY>
<% String itemID = request.getParameter("itemID"); %>
<ws:webservice id="ebay"
  wsdlUrl="http://www.xmethods.net/sd/2001/EBayWatcherService.wsdl"
  binding="eBayWatcherBinding"
  soapLocation="http://services.xmethods.net:80/soap/servlet/rpcrouter"
  scope="page">
  <ws:property name="http.proxyHost" value="www-proxy.us.oracle.com"/>
  <ws:property name="http.proxyPort" value="80"/>
</ws:webservice>
<ws:invoke id="price" webservice="ebay" operation="getCurrentPrice">
  <ws:part name="auction_id" value="<%=itemID%>"/>
</ws:invoke>
<B>Action price for eBay Item # <%=itemID%> is :</B>
<P>
$<%= price%>@<%=new java.util.Date()%>
</P>
</BODY>
</HTML>
```

- JDeveloper makes it easy to expose PL/SQL as a WebService



- Many useful features for WebService development
  - Knows about UDDI registries
  - Knows how to deploy into various SOAP servers
    - Apache/OC4J/etc...
  - Permits debugging of WebServices
    - Local/remote debugging possible
  - Includes a packet monitor
    - Filter/replay possible





- Can publish a WebService as a portlet
  - Portlet environment provides RPCWebServiceRenderer and DocWebServiceRenderer to enable integration of the two styles of messages into portal pages

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<?providerDefinition version="3.1"?>
<provider class="oracle.portal.provider.v2.DefaultProviderDefinition">
<session>>false</session>

<portlet class="oracle.portal.provider.v2.DefaultPortletDefinition">
<id>1</id>
<name>WhoIs</name>
<showPage class="oracle.portal.provider.v2.webservice.DocWebServiceRenderer">
  <contentType>text/html</contentType>
  <logging>>false</logging>
  <endpointURL>http://www.esynaps.com/WebServices/WhoIsService.asmx</endpointURL>
  <soapAction>http://tempuri.org/WhoIs</soapAction>
  <literal class="oracle.portal.provider.v2.webservice.LiteralXML"
    handler="oracle.portal.provider.v2.webservice.LiteralXML$Handler">
    <element name="WhoIs">
      <attribute name="xmlns">http://tempuri.org/</attribute>
      <element name="DomainName"
        bind="urlParams/domainName"
        default="oracle.com"
        prompt="Please enter an Internet Domain name"/>
    </element>
  </literal>
  <escapeOutput>>false</escapeOutput>
  <responseXSL>whoIsResponse.xsl</responseXSL>
</showPage>
</renderer>
</portlet>
```