Oracle Developer Day

Sponsored by: 

Track # 1: Session #2
Web Services

Speaker
Agenda

- Developing Web services
  - Architecture, development and interoperability
- Quality of service
  - Security, reliability, management
  - QoS deployment options
- Best Practices
  - Web Services and Management

Basic Web Services

UDDI Registry

- Points to description
- Finds Service

WSDL

- Points to service
- Describes Service

Web Service Client (J2EE, .NET, PL/SQL…)

- Invokes with XML Messages

SOAP

Web Service (J2EE, PL/SQL, .NET, C/C++, Legacy…)

Describes Service
Enterprise Web Services

- Standards + QoS + Processes

SOA Tools

- Web Service Distributed Management
- Web Service Orchestration
- Web Service Transactions
- Web Service Policy
- Web Service Security
- Web Service Reliable Messaging

JAX-*: JAX-B, JAX-P, JAX-RPC, ...

WS-I Basic Profile

SOAP, WSDL, UDDI

J2EE 1.4 Standards

<table>
<thead>
<tr>
<th>Java APIs for XML</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAXP</td>
<td>Java API for XML Parsing</td>
</tr>
<tr>
<td>JAXB</td>
<td>Java API for XML Data Binding</td>
</tr>
<tr>
<td>JAX-RPC</td>
<td>Java API for XML Remote Procedure Call</td>
</tr>
<tr>
<td>SAAJ</td>
<td>SOAP API for Attachments in Java</td>
</tr>
<tr>
<td>JAXR</td>
<td>Java API for XML Registries</td>
</tr>
<tr>
<td>EJB 2.1</td>
<td>Stateless Session EJB Endpoint Model</td>
</tr>
<tr>
<td>JSR 109</td>
<td>Web Services Deployment Model</td>
</tr>
</tbody>
</table>
JAX-RPC

- Core WS standard for Java
  - Enables portability across containers (different vendors implemented differently before JAX-RPC for J2EE 1.4)
- Standard class and interface structure for client and server
  - Various hook-points for custom serializers, custom handlers, attachments
- Tools must provide abstractions to handle service as set of classes
  - Navigator structure for quick location
  - Wizard support for declarative definition

Web Services Interoperability

- Web Services Interoperability
  - Develops profiles
  - Suggests best practices
  - Provides testing tools
- Runtime and tools uptake
  - Oracle JDeveloper analysis
  - OracleAS compliance
Beyond J2EE 1.4

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSR-181</td>
</tr>
<tr>
<td>Web Services Metadata for the Java™ Platform</td>
</tr>
<tr>
<td>EJB 3.0</td>
</tr>
<tr>
<td>Simplifying EJB development</td>
</tr>
<tr>
<td>WSIF</td>
</tr>
<tr>
<td>Web Services Invocation Framework</td>
</tr>
</tbody>
</table>

**JSR-181 Annotations**

- Support a code-driven version of the bottom up use case: create web services implicitly by annotating the source code
- Example annotations
  - @WebService
  - @WebMethod
  - @WebParam
  - @WebResult
  - @SOAPBinding
  - ...
EJB 3.0 as Web Service

- Annotate business interface
- Deploys to Java EE container(s) as a Web Service

```java
package buslogic;
import java.rmi.RemoteException;
import java.rmi.Remote;
import javax.jws.WebMethod;
import javax.jws.WebService;

@WebService
public interface CustomerFacade {
    @WebMethod
    public Integer getCreditRating(String ssn);
}
```

Web Services Invocation Framework (WSIF)

- Describe any artifact with WSDL, invoke with native protocols
  - No overhead from SOAP processing
  - Focus on performance and transactionality
- Used extensively by BPEL
Two Development Use Cases
*Top Down Contract Driven And Bottom Up*

**Bottom Up**
- Service Implementation
- WSDL
  - e.g. EJB/Java Class to WSDL

**Top Down**
- WSDL
- Service Implementation
  - e.g. WSDL contract as the interface & message description

**Demonstration**
Publishing with Annotations
Bottom-Up Web Service
Agenda

- Developing Web services
  - Architecture, development and interoperability
- Quality of service
  - Security, reliability, management
  - QoS deployment options
- Best Practices
  - Web Services and Management

Quality of Service Requirements

- Reliability
  - Are my messages actually being delivered?
    - Guaranteed delivery, ordered delivery
  - WS-ReliableExchange standard in OASIS
- Security
  - Are my messages secure?
    - Authentication, encryption and signing
  - WS-Security standard in OASIS
- Management
  - How is my system handling messages?
    - Configuration, logging, monitoring
  - WS for Distributed Management (WSDM) standard in OASIS
Web Services Reliability

- At least once semantics
- At most once semantics
- Guaranteed message ordering
- Exactly once semantics

Web Services Security

- Authentication
  - User name/password token
  - X.509 token
  - SAML token for SSO Applications
- XML Digital Signatures
- XML Encryption
Administration of Web Services

Auditing, Logging, Reliability and WS-Security

Monitoring and Performance Analysis

DEMONSTRATION

Administration of Web Services
Web Services Management

- Managing and deploying operational policies across applications and Web services
  - Layer best-practice security and management across all services
- No modifications to applications or services
  - Support WS-* standards
- Can secure/manage Web services implemented in heterogeneous languages and environments
Policy Enforcement and Visibility: Monitoring

- Provide real-time visibility into Web Service interactions
  - Automatic issue resolution by dynamically updating policies
  - Alerts about anomalies
  - Enforce policies
  - Validate compliance with IT best practices

Deployment Options QoS

- Oracle Application Server
  - Administrator manages policies service by service
  - Configuration for Oracle Application Server services only
  - Designed to be seamless with Oracle Application Server
    - Integration in JDeveloper and Application Server Control
- Independent Web Service Manager
  - Framework manage groups of policies for groups of services
  - Configuration for any service
    - J2EE (Oracle, IBM, BEA …), .NET, C, Perl, PHP …
  - Designed to scale to large scale heterogeneous deployments
Demonstration

Web Services Management

Agenda

- Developing Web services
  - Architecture, development and interoperability
- Quality of service
  - Security, reliability, management
  - QoS deployment options
- Best Practices
  - Web Services and Management
Best Practices – Web Services

- Build coarse grained Web services not fine grained
- Design to an interface, not an implementation
- Interoperability
  - Target conformity to WS-I
  - Avoid language dependent types to maximize interoperability (e.g. Java collections)
  - Provide sample clients to jumpstart clients
- Know when to use Web services
  - Often native protocols are all that is available or may be faster
  - Use WSIF bindings for native protocols and BPEL

Best Practices – Web Services Management

- Development
  - Use native container for development and testing
    - Security, Reliability, Management …
- Deployment
  - If working homogeneously use Oracle Application Server
  - If working in a heterogeneous environment strongly consider a Web Services Management framework
Join Over 3,000,000 Developers!
Free Technical Advice
Free Software Downloads
www.oracle.com/technology/products/jdev
otn.oracle.com/tech/webservices
otn.oracle.com/tech/java
otn.oracle.com/bpel

Learn Oracle From Oracle

• Instructor led training
• Self-Study
• Online learning
• Oracle Certification
• Oracle iLearning
• Oracle Tutor

oracle.com/education
QUESTIONS & ANSWERS

Oracle Developer Day

Sponsored by: Intel