



Web Services & Your HP e3000

David Thatcher

Advanced Network Systems, LLC

www.advnetsys.com

(908) 638-3330



Our Tools

Open Source Tools

- Axis - beta 3
- WSIF - Web Services Inspection Language
- WSDL4J – Web Services Description language for Java
- UDDI - Universal Description Discovery and Integration for Java

Commercial Tools

- ADBC Developer's Kit – IMAGE/SQL Java API
- Web/iX – J2EE Enterprise Server
- COBOL Subprogram



Our Service

- To begin, we will take a COBOL application that is already being called via a Java class and make it a web Service.
- Second we will take an existing Java program that accesses an IMAGE/SQL database and also make it a Web Service.
- Last, we will write Java clients to access the programs via SOAP calls.

This lesson will concentrate on creating Web Services, not Interfaces between Java and your HP e3000.

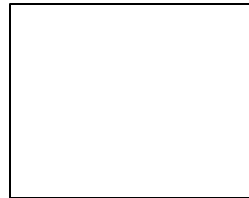


Deploy COBOL to Web Services

(Ethernet)

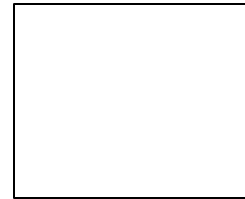
Enterprise System

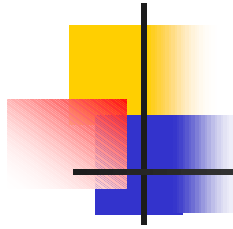
Java Program



HP e3000

COBOL Order Status Check





Deploy COBOL to Web Services

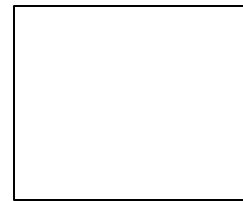
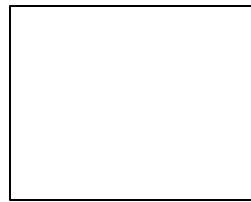
(Ethernet)

Enterprise System

HP e3000

Java Program

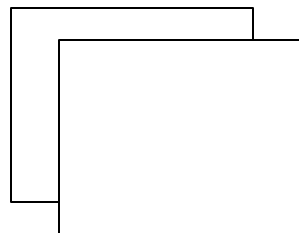
COBOL Order Status Check



SOAP, XML, J2EE...

Customer/Vendors

Web Service Client





Deploy Our Code as Web Service

To Deploy our code as a Web Service under Axis, we need a deployment descriptor:

(Code Example will be displayed at Symposium)



Deploy Our Code as Web Service

The deployment descriptor defines several things:

- The Name of the Service
- What type of service this is
- The name of the java class that provides the service
- The names of the methods supported by the service



Write a Client Application

Regardless of how we write our client application, the application has to know five things:

- The name of the machine hosting the service
- The name of the service
- The name of the method we want to invoke
- The parameters of the method we want to invoke
- What the SOAPAction field should contain



Write a Client Application

Our first application will be a Java client written using the Axis client API. The first steps in using Axis are:

- Create a service object
- Use the service object to create a call object
- Set the properties of the call object
- Use the invoke method of the call object



Write a Client Application

(Demonstrate writing a client application. You must be at the Solution Symposium to see.)

Client will communicate with a service and tell it to call a COBOL subprogram located on the HP e3000. The COBOL program will query a dataset to retrieve the status of a factory order. The order status is sent back to the service and returns the value to the Web Service client.



Write a Client Application

Client can be traced pretty easily using Axis
Tracemon:



Web Services Discovery

WSDL:

- Your client can get all necessary details about the web service.



Web Services Discovery

WSIF:

- IBM recently donated WSIF to the Apache XML project.
- WSIF is designed to isolate developers from the details of invoking the Web Service.



Web Services Discovery

UDDI:

- With UDDI, we store the description of the service in a UDDI registry.
- Our client searches the registry for the appropriate service, finds it's details (WSDL) and invokes the service.



Web Services

Conclusion:

In this presentation, we created a Web Service from a simple piece of code. We deployed the services and had it communicate with your HP e3000.

Your HP e3000 can still be used and add value in your e-service initiatives. All development done using Web Services will move along with you as it follows open standards.



Web Services & Your HP e3000

David Thatcher

Advanced Network Systems, LLC

www.advnetsys.com

(908) 638-3330