WS-Discovery

WS-Discovery

WS-Discovery is a protocol to enable dynamic discovery of services available on the local network. By default, WS-Discovery uses a UDP based multicast transport to announce new services and probe for existing services. However, it also supports a managed mode where a discovery proxy is used to reduce the amount of multicast traffic required.

Enabling WS-Discovery announcements for services

To enable CXF to send "Hello" announcements when services and endpoint are started, the cxf-services-ws-discovery-service and cxf-services-ws-discovery-api jars need to be available on the classpath. The cxf-services-ws-discovery-service jar will register a ServerLifecyleListener that will automatically publish the "Hello" messages. It will also respond to any Probe requests that match the services it has published.

By default, the WS-Discovery service will startup in ad-hoc mode. However, if you set a Bus property of "org.apache.cxf.service.ws-discovery.address" to a URL address of a WS-Discovery Proxy, the service will change to managed mode and send the Hello/Bye notices directly to that proxy. It will also not respond to Probes.

Using WS-Discovery client to probe for services

CXF also provides an API to probe the network or WS-Discovery proxy. The org.apache.cxf.ws.discovery.WSDiscoveryClient class provides several methods for probing the network.

```
//Use WS-Discovery to find references to services that implement the Greeter portType
WSDiscoveryClient client = new WSDiscoveryClient();
   // or: new WSDiscoveryClient("soap.udp://proxyhost:3702");
List<EndpointReference> references = client.probe(new QName("http://cxf.apache.org/hello_world/discovery",
"Greeter"));
client.close();

GreeterService service = new GreeterService();
//loop through all of them and have them greet me.
for (EndpointReference ref : references) {
    Greeter g = service.getPort(ref, Greeter.class);
    System.out.println(g.greetMe("World"));
}
```