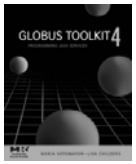
Grid Computing

Globus Toolkit Programming
GT4 Tutorial
Chapter 4

Paul A. Farrell
Fall 2006



Globus Toolkit 4: Programming Java Services

Borja Sotomayor and Lisa Childers

Morgan Kaufmann Publishers / Elsevier

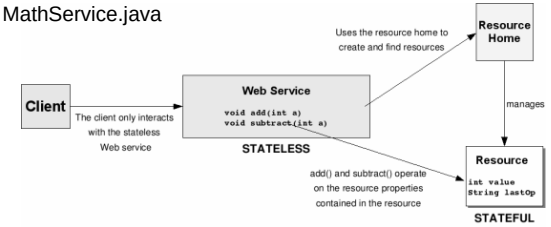
http://gdp.globus.org/gt4-tutorial/

Chapter 4: Singleton Resources

- Splitting the Resource, Home, and Service
- Divide code into 3 files in

$EXAMPLES_DIR/org/globus/examples/services/core/singleton/impl/

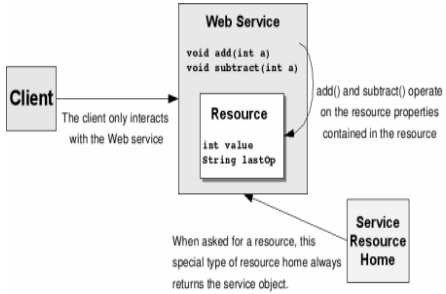
  - MathResource.java
  - MathResourceHome.java
  - MathService.java



How Chapter 3 example worked

<resource name="home" type="org.globus.wsrf.impl.ServiceResourceHome">

special type of resource home that always returns the service object



Five Steps

1. **Define the service's interface**. This is done with *WSDL*
   - Interface is the same, so no change
2. **Implement the service**. This is done with *Java*
   - Most changes are here
3. **Define the deployment parameters**. This is done with *WSDD and JNDI*
   - WSDD: Only need to change the service name, and service class name
   - Need to talk about JNDI
4. **Compile everything and generate GAR file**. This is done with *Ant*
   - As before
5. **Deploy service**. This is done with a GT4 tool

Step 2: Implement the Service

Implement the service using

MathQNames.java and

MathResource.java

MathResourceHome.java

MathService.java

and place in:

$TUTORIAL_DIR/org/globus/examples/services/core/singleton/impl/

QNames for MathService

```
package org.globus.examples.services.core.singleton.impl;
import javax.xml.namespace.QName;
public interface MathQNames {
public static final String NS =
    "http://www.globus.org/namespaces/examples/core/MathService
    _instance";
public static final QName RP_VALUE = new QName(NS, "Value");
public static final QName RP_LASTOP = new QName(NS,
    "LastOp");
public static final QName RESOURCE_PROPERTIES = new
    QName(NS, "MathResourceProperties"); }
```


First Part – Resource Resource

```
package org.globus.examples.services.core.singleton.impl;
/* import java.rmi.RemoteException; */
import org.globus.wsrf.Resource;
import org.globus.wsrf.ResourceProperties;
import org.globus.wsrf.ResourceProperty;
import org.globus.wsrf.ResourcePropertySet;
import org.globus.wsrf.impl.ReflectionResourceProperty;
import org.globus.wsrf.impl.SimpleResourcePropertySet;
/* import org.globus.examples.stubs.MathService_instance.AddResponse;
import
    org.globus.examples.stubs.MathService_instance.SubtractResponse;
import org.globus.examples.stubs.MathService_instance.GetValueRP; */
```


Next – Two Resource Properties

```
public class MathResource  implements Resource, ResourceProperties {

    /* Resource Property set */
    private ResourcePropertySet propSet;

    /* Resource properties */
    private int value;
    private String lastOp;

    /* Get/Setters for the RPs */
    public int getValue() {
            return value;
    }
    public void setValue(int value) {
            this.value = value;
    }
    public String getLastOp() {
            return lastOp;
    }
    public void setLastOp(String lastOp) {
            this.lastOp = lastOp;
    }
```

Should be same as resource properties in WSDL but with lower case first letter

Should be same as resource properties in WSDL

RP Initialization – Not in constructor any more

```
/* Initializes RPs */                                   Use QName defined earlier
public void initialize() throws Exception {
this.propSet = new SimpleResourcePropertySet(MathQNames.RESOURCE_PROPERTIES);

try {
ResourceProperty valueRP = new ReflectionResourceProperty(
        MathQNames.RP_VALUE, "Value", this);
this.propSet.add(valueRP);
setValue(0);

ResourceProperty lastOpRP = new ReflectionResourceProperty(
        MathQNames.RP_LASTOP, "LastOp", this);
this.propSet.add(lastOpRP);
setLastOp("NONE");
} catch (Exception e) {
throw new RuntimeException(e.getMessage());
}
}
/* Required by interface ResourceProperties */
public ResourcePropertySet getResourcePropertySet() {
        return this.propSet;
}
```

Should be same as resource properties in WSDL but with lower case first letter

Service Implementation

```
package org.globus.examples.services.core.singleton.impl;
import java.rmi.RemoteException;
import
    org.globus.examples.services.core.singleton.impl.MathResource;
import org.globus.wsrf.ResourceContext;
import
    org.globus.examples.stubs.MathService_instance.AddResponse;
import
    org.globus.examples.stubs.MathService_instance.SubtractRespo
    nse;
import
    org.globus.examples.stubs.MathService_instance.GetValueRP;
public class MathService {
```


Service Implementation

- Stateful information no longer in this class
  - Have to use MathResource object

```
public AddResponse add(int a) throws RemoteException {
    MathResource mathResource = getResource();
    mathResource.setValue(mathResource.getValue() + a);
    mathResource.setLastOp("ADDITION");

    return new AddResponse();
  }
```


Service Implementation - getResource

- Uses ResourceContext a Globus class to get resource

```
/*
 * Private method that gets a reference to the resource specified in the
 * endpoint reference.
 */
private MathResource getResource() throws RemoteException {
Object resource = null;
try {
resource = ResourceContext.getResourceContext().getResource();
} catch (Exception e) {
throw new RemoteException("Unable to access resource.", e);
}

MathResource mathResource = (MathResource) resource;
return mathResource;
}
```

Resource Home

- Extends Globus SingletonResourceHome
- Implements findSingleton method, creates object, initializes, returns it

```
package org.globus.examples.services.core.singleton.impl;

import org.globus.wsrf.Resource;
import org.globus.wsrf.impl.SingletonResourceHome;
public class MathResourceHome extends SingletonResourceHome {
public Resource findSingleton() {
try {
// Create a resource and initialize it.
MathResource mathResource = new MathResource();
mathResource.initialize();
return mathResource;
} catch (Exception e) {
e.printStackTrace();
return null;
}
}
}
```


Five Steps

1. **Define the service's interface**. This is done with *GWSDL*
2. **Implement the service**. This is done with *Java*
3. **Define the deployment parameters**. This is done with *WSDD (and JNDI)*
4. **Compile everything and generate GAR file**. This is done with *Ant*
5. **Deploy service**. This is done with *a GT4 tool*

WSDD deployment descriptor

```
<?xml version="1.0" encoding="UTF-8"?>
<deployment name="defaultServerConfig"
   xmlns="http://xml.apache.org/axis/wsdd/"
   xmlns:java="http://xml.apache.org/axis/wsdd/providers/java"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <service name="examples/core/singleton/MathService" provider="Handler" use="literal"
      style="document">
      <parameter name="className"
      value="org.globus.examples.services.core.singleton.impl.MathService"/>
      <wsdlFile>share/schema/examples/MathService_instance/Math_service.wsdl</wsdlFile>
      <parameter name="allowedMethods" value="*"/>
      <parameter name="handlerClass" value="org.globus.axis.providers.RPCProvider"/>
      <parameter name="scope" value="Application"/>
      <parameter name="providers" value="GetRPProvider"/>
      <parameter name="loadOnStartup" value="true"/>
   </service>
</deployment>
```

URI = base addr of Web Services container + name e.g. http://localhost:8080/wsrf/services/examples/core/first/MathService

Load service when container is started

JNDI deployment file

- Specifies resource home, and parameters related to how the resource home manages those resources
- Contains service element for each service being configured

```
<?xml version="1.0" encoding="UTF-8"?>
<jndiConfig xmlns="http://wsrf.globus.org/jndi/config">

<service name="examples/core/singleton/MathService">
<resource name="home" type="org.globus.examples.services.core.singleton.impl.MathResourceHome ">
<resourceParams>

<parameter>
<name>factory</name>
<value>org.globus.wsrf.jndi.BeanFactory</value>
</parameter>

</resourceParams>

</resource>
</service>

</jndiConfig>
```

Five Steps

1. **Define the service's interface**. This is done with *GWSDL*
2. **Implement the service**. This is done with *Java*
3. **Define the deployment parameters**. This is done with *WSDD and JNDI*
4. **Compile everything and generate GAR file**. This is done with *Ant*
   - ./globus-build-service.sh singleton
5. **Deploy service**. This is done with *a GT4 tool*
   - *globus-deploy-gar $EXAMPLES_DIR/org_globus_examples_services_core_singleton.gar*

Execute the client

**java **

**-classpath ./build/stubs/classes/:$CLASSPATH **

**org.globus.examples.clients.MathService_instance.Client **

http://127.0.0.1:8080/wsrf/services/examples/core/singleton/MathService

If all goes well, you should see the following:

Current value: 15