

Grid Computing

Globus Toolkit Programming GT4 Tutorial Chapter 3

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/>

Paul A. Farrell 2006 KENT STATE Grid Computing 1

Chapter 3: Writing your first Grid Service

- **Math Service**
 - Will perform two operations, addition and subtraction
- **Before you start**
 - Download the tutorial source files to your account on cohn.cs.kent.edu
 - http://gdp.globus.org/gt4-tutorial/download/progtutorial-examples_0.2.1.tar.gz

Paul A. Farrell 2006 KENT STATE Grid Computing 2

Directory Layout

```
$TUTORIAL_DIR
|
|-- schema/
|   |
|   |-- examples/
|       |-- MathService_instance
|       |-- Math.wsdl
|
|--- org/
|   |
|   |-- globus/
|       |-- examples/
|           |-- services/ ----> Service impl files
|           |-- clients/ ----> Client impl files
```

Paul A. Farrell 2006 KENT STATE Grid Computing 3

Five Steps

1. **Define the service's interface.** This is done with *WSDL*
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*

Paul A. Farrell 2006 KENT STATE Grid Computing 4

Step 1: Define the interface in GWSDL – Two options

- **Writing the WSDL directly.**
 - This is the most versatile option.
 - Total control over the description of our portType.
- **Generating WSDL from a Java interface.**
 - The easiest option, but very complicated interfaces are not always converted correctly to WSDL

The interface in Java

```
public interface Math
{
    public void add(int a);
    public void subtract(int a);
    public int getValueRP();
}
```

Notice there is only one parameter!?!

Steps to WSDL

- Write the root element <definitions>
- Write the <PortType>
- Write an input and output <message> for each operation in the PortType.
- Write the <types>
 - Response and request types , and resource properties

See
\$EXAMPLES_DIR/schema/examples/MathService_instance/Math.wsdl
Read Appendix A on writing WSDL

a. Write the root element

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="MathService"
  targetNamespace="http://www.globus.org/namespaces/examples/core/MathService_instance"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="http://www.globus.org/namespaces/examples/core/MathService_instance"
  xmlns:wsp="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsrp="http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ResourceProperties-1.2-draft-01.xsd"
  xmlns:wsrpw="http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ResourceProperties-1.2-draft-01.wsdl"
  xmlns:wsdpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

All the portTypes and operations will belong to this namespace

Name of WSDL file. Not related to name of portType

<definitions ...>

name: the 'name' of the WSDL file. Not related with the name of the PortType

targetNamespace: The target namespace of the WSDL file. All portTypes will belong to this namespace.

<import ...>

File with definitions we need later:

```
<wsdl:import
  namespace=
  "http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ResourceProperties-
  1.2-draft-01.wsdl"
  location="../wsrf/properties/WS-ResourceProperties.wsdl" />
```

b. Write the PortType

```
<portType name="MathPortType"
  wsdlpp:extends="wsrp:GetResource
  wsrp:ResourceProperties" tns:MathR
  <operation name="add">
    <input message="tns:AddInputMessage" />
    <output message="tns:AddOutputMessage" />
  </operation>
  <operation name="subtract">
    <input message="tns:SubtractInputMessage" />
    <output message="tns:SubtractOutputMessage" />
  </operation>
  <operation name="getValueRP">
    <input message="tns:GetValueRPInputMessage" />
    <output message="tns:GetValueRPOutputMessage" />
  </operation>
</portType>
```

Not a standard part of WSDL. Part of WSDLPreprocessor namespace provided by Globus

What service's resource properties are

Message that should be passed when operation is invoked

portType tag important attributes

- name: name of the PortType
- **wsdlpp:extends**
 - not a standard part of WSDL.
 - part of the WSDLPreprocessor namespace provided by Globus
 - notice how we declared wsdlpp in the namespace declarations
 - In WSDL, only way of including operations and portTypes from WSRF specifications would be to copy and paste from the spec's WSDL file into our own WSDL file
 - Globus provides a WSDL Preprocessor that does that automatically for us
 - wsdlpp:extends attribute, tells WSDL Preprocessor to include the GetResourceProperty portType from the WS-ResourceProperties WSDL file

Inside portType

- An <operation> tag for each method in the PortType
- Operation tag has an input tag, and an output tag

c. Write input and output messages for each port type

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions ... >
<message name="AddInputMessage">
<part name="parameters" element="tns:add"/> </message>
<message name="AddOutputMessage">
<part name="parameters" element="tns:addResponse"/> </message>
<!-- PortType --> </definitions>
```

- Name of each message has to be the same as the one written in the message attribute of the <input> and <output> tags
- Messages are composed of <part>s! Our messages will only have one part

d. Define the XML elements inside the <types> tag

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions ... >
<types>
<xsd:schema
targetNamespace="http://www.globus.org/namespaces/examples/core/MathService_instance"
xmlns:tns="http://www.globus.org/namespaces/examples/core/MathService_instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!-- REQUESTS AND RESPONSES -->
<xsd:element name="add" type="xsd:int"/>
<xsd:element name="addResponse">
<xsd:complexType/>
</xsd:element>
<!-- more type definitions -->
</xsd:schema>
</types>
<!-- Messages -->
<!-- PortType -->
</definitions>
```

Response and targetNamespace of WSDL document

Input parameter of add operation

d. Define the XML elements inside the <types> tag

```
<types> <xsd:schema
targetNamespace="http://www.globus.org/namespaces/examples/core/MathService_instance"
xmlns:tns="http://www.globus.org/namespaces/examples/core/MathService_instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!-- Requests and responses declarations-->
<xsd:element name="Value" type="xsd:int"/>
<xsd:element name="LastOp" type="xsd:string"/>
<xsd:element name="MathResourceProperties" type="tns:MathResourceProperties"/>
<xsd:complexType base="tns:MathResourceProperties">
<xsd:sequence base="tns:MathResourceProperties">
<xsd:element ref="tns:Value" minOccurs="1" maxOccurs="1"/>
<xsd:element ref="tns:LastOp" minOccurs="1" maxOccurs="1"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</types>
```

Resource property targetNamespace of WSDL document

Two resource properties: Value, and Last Op

Recall <portType name="MathPortType" wsdlpp:extends="wsrpw:GetResourceProperties" wsrp:ResourceProperties="tns:MathResourceProperties">

The whole WSDL file

```
<a  
  href=http://www.cs.kent.edu/~farrell/grid06/lectures/P  
  rogtutorial/schema/examples/MathService_instance/  
  Math.wsdl>
```

Paul A. Farrell 2006 KENT STATE Grid Computing 17

WSDL Preprocessor

- The WSDL Preprocessor (the `wsdlpp:extends` attribute in the `portType` element) is provided in GT4 as a *convenience*.
- WSDL files using `wsdlpp:extends` will always be converted to standard WSDL before they are actually used.
 - doesn't affect GT4's interoperability with other WSRF implementations
 - called "flattening" because we take several WSDL files (our file plus any WSRF WSDL files we extend from) and then merge them into a single (flattened) file.
 - GT4 will always publish the *flattened* version of our WSDL file.
- You are not *required* to use `wsdlpp:extends`. If you choose to, you can write the flattened version directly. However, this involves a fair amount of copy-pasting that can be very error-prone.
- Bottom line: GT4 doesn't require that other Web Services implementations or other WSRF implementations use `wsdlpp:extends`
 - It is a purely internal feature of the toolkit.

Paul A. Farrell 2006 KENT STATE Grid Computing 18

Namespace Mappings

- WSDL is *language-neutral*
- Eventually we need to specify the language
- Use a set of *stub classes* generated from the WSDL file using a GT4 tool
- To successfully generate the stub classes, we need to tell it where (i.e. in what Java package) to place the stub classes.
- We do this with a *mappings file*, which maps WSDL namespaces to Java packages

Paul A. Farrell 2006 KENT STATE Grid Computing 19

Namespace Mappings

```
http://www.globus.org/namespace/examples/core/MathService_instance  
  = org.globus.examples.stubs.MathService_instance  
http://www.globus.org/namespace/examples/core/MathService_instance/  
  bindings= org.globus.examples.stubs.MathService_instance.bindings  
http://www.globus.org/namespace/examples/core/MathService_instance/  
  service= org.globus.examples.stubs.MathService_instance.service
```

- First namespace is the target namespace of the WSDL file
- Other two namespaces are automatically generated when a GT4 tool 'completes' the WSDL file
- The example stub classes will be in Java Packages:
 - `org.globus.examples.stubs` e.g.
 - `org.globus.examples.stubs.MathService_instance`

Paul A. Farrell 2006 KENT STATE Grid Computing 20

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*
4. **Compile everything and generate GAR file.** This is done with *Ant*
5. **Deploy service.** This is done with a *GT4 tool*

Paul A. Farrell 2006 KENT STATE Grid Computing 21

Step 2: Implement the Service

Implement the service using
MathQNames.java and
MathService.java
and place in:

[\\$TUTORIAL_DIR/org/globus/examples/services/core/first/impl/](#)

Paul A. Farrell 2006 KENT STATE Grid Computing 22

QNames

- *Qualified name*, or QName for short includes a namespace and a *local name* e.g.
 - {http://www.globus.org/namespaces/examples/core/MathService_instance}Value
- A qualified name is represented in Java using the QName class
- Since we refer to service's qualified names frequently, good practice to put in a separate interface

Paul A. Farrell 2006 KENT STATE Grid Computing 23

QNames for MathService

```
package org.globus.examples.services.core.first.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"); }
```

Paul A. Farrell 2006 KENT STATE Grid Computing 24

Service Implementation

- First simple example, service implementation is Java class with code for both service *and* resource
- Later we will divide these
- Need to implement the Resource interface
 - this interface doesn't require any methods.
 - It is simply a way of tagging a class as being a resource
 - Implementing the ResourceProperties interface indicates our class has a set of resource properties we want to make available.
 - This interface requires that we add the following to our class:


```
private ResourcePropertySet propSet;
public ResourcePropertySet getResourcePropertySet() {
    return this.propSet;
}
```

Paul A. Farrell 2006 KENT STATE Grid Computing 25

First Part – Import

```
package org.globus.examples.services.core.first.impl;
import java.rmi.RemoteException;
import org.globus.wsrfl.Resource;
import org.globus.wsrfl.ResourceProperties;
import org.globus.wsrfl.ResourceProperty;
import org.globus.wsrfl.ResourcePropertySet;
import org.globus.wsrfl.impl.ReflectionResourceProperty;
import org.globus.wsrfl.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;
```

Paul A. Farrell 2006 KENT STATE Grid Computing 26

Next – Two Resource Properties

- Value of type xsd:int and LastOp of type xsd:string
- Need attribute for each resource property along with a get/set method pair for each resource property

```
public class MathService 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

Paul A. Farrell 2006 KENT STATE Grid Computing 27

Recall the XML elements inside the <types> tag

Resource properties

```
<types> <xsd:schema
    targetNamespace="http://www.globus.org/namespaces/examples/core/MathService_instance"
    xmlns:tns="http://www.globus.org/namespaces/examples/core/MathService_instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <!-- Requests and responses declarations-->
    <xsd:element name="Value" type="xsd:int"/>
    <xsd:element name="LastOp" type="xsd:string"/>

    <xsd:element name="MathResourceProperties">
        Two resource properties:
        Value , and Last Op
        <xsd:complexType>
        <xsd:sequence>
        <xsd:element ref="tns:Value" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="tns:LastOp" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
</types>
```

Paul A. Farrell 2006 KENT STATE Grid Computing 28

Constructor & Resource Initialization

```
/* Constructor. Initializes RPs */
public MathService() throws RemoteException {
    /* Create RP set */
    this.propSet = new SimpleResourcePropertySet(
        MathQNames.RESOURCE_PROPERTIES);

    /* Initialize the RP's */
    try {
        ResourceProperty valueRP = new ReflectionResourceProperty(
            MathQNames.RP_VALUE, "Value", this);
        this.propSet.add(valueRP);
        /* should be same as resource properties
           in WSDL but with lower case first letter */
        ResourceProperty lastOpRP = new ReflectionResourceProperty(
            MathQNames.RP_LASTOP, "LastOp", this);
        this.propSet.add(lastOpRP);
        setLastOp("NONE");
    } catch (Exception e) {
        throw new RuntimeException(e.getMessage());
    }
}
```

Use QName defined earlier

Paul A. Farrell 2006 KENT STATE Grid Computing 29

Constructor & Resource Initialization

- Used simplest type of resource creation - ReflectionResourceProperty
 - makes the implementation much simpler,
 - also adds restrictions on our resource implementation
 - such as the need for get/set method
- GT4 includes other classes to deal with more complex resource scenarios e.g.
 - SimpleResourceProperty
 - PersistentResourceProperty

Paul A. Farrell 2006 KENT STATE Grid Computing 30

Implementation of remotely-accessible methods

- Straightforward, except for peculiarities in how the parameters and return types have to be declared
- ```
/* Remotely-accessible operations */
public AddResponse add(int a) throws RemoteException {
 value += a;
 lastOp = "ADDITION";
 return new AddResponse(); }

public int getValueRP(GetValueRP params) throws RemoteException {
 return value; }
```
- add operation
    - Has no return type in the WSDL file,
    - return type is AddResponse here
  - Why? WSRF uses *document/literal bindings*

Paul A. Farrell 2006 KENT STATE Grid Computing 31

## Document/literal bindings

- For operation which is part of WSDL interface
  - parameters and return values will *in some cases* be 'boxed' inside stub classes
  - generated automatically from the WSDL file
  - Only boxed when
    - number of parameters is more than one
    - return type is void or a complex type
  - add and subtract return void, and are 'boxed' so we have to return is AddResponse and SubtractResponse
  - getValueRP method returns an int value, so it is not 'boxed'

Paul A. Farrell 2006 KENT STATE Grid Computing 32

## 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*

Paul A. Farrell 2006 KENT STATE Grid Computing 33

## Step 3: Configure the deployment in WSDD

Saved in files called

\$TUTORIAL\_DIR/org/globus/examples/services/core/first/deploy-server.wsdd

\$TUTORIAL\_DIR/org/globus/examples/services/core/first/deploy-jndi-config.xml

Paul A. Farrell 2006 KENT STATE Grid Computing 34

## 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*
4. **Compile everything and generate GAR file.** This is done with *Ant*
5. **Deploy service.** This is also done with *Ant*

Paul A. Farrell 2006 KENT STATE Grid Computing 35

## 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/
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 URI = base addr of Web Services
 container + name e.g.
 http://localhost:8080/wsrfl/services/examples/
 core/first/MathService
 <service name="examples/core/first/MathService" provider="Handler" use="literal"
 style="document">
 <parameter name="className"
 value="org.globus.examples.services.core.first.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>
```

Load service when container is started

Paul A. Farrell 2006 KENT STATE Grid Computing 36

## Service name

```
<service name="examples/core/first/MathService" provider="Handler" style="wrapped">
```

- Specifies the location where our Grid Service will be found.
- Combined with the base address (URL) of Grid Service container, gives the full URI of the Grid Service.
- For example, if we are using the GT4 standalone container, the base URL will probably be `http://localhost:8080/wsrf/services`.
- Therefore, our service's GSH would be:

```
http://localhost:8080/wsrf/services/progtutorial/core/first/MathService
```

## WSDL file

```
<wsdlFile>share/schema/examples/MathService_instance/Math_service.wsdl</wsdlFile>
```

- Tells the grid service container where the **WSDL** file for this service can be found
- Our WSDL is a (non-standard) extension of WSDL
- It must first be converted to normal WSDL so it can be truly interoperable with existing web services technologies.
- WSDL file (`Math_service.wsdl`) will be generated automatically by a GT4 tool when we compile the service.

## Common parameters

- Three parameters found in every grid service we program

```
<parameter name="allowedMethods" value="**"/>
<parameter name="handlerClass" value="org.globus.axis.providers.RPCProvider"/>
<parameter name="scope" value="Application"/>
```

## JNDI deployment file

- Needed but not really used in this example

```
<?xml version="1.0" encoding="UTF-8"?>
<jndiConfig xmlns="http://wsrf.globus.org/jndi/config">
 <service name="examples/core/first/MathService">
 <resource name="home" type="org.globus.wsrf.impl.ServiceResourceHome">
 <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*
5. **Deploy service.** This is done with a *GT4 tool*

Paul A. Farrell 2006 KENT STATE Grid Computing 41

## Step 4: Create a GAR file with ANT

- Using the files we wrote we generate a *Grid Archive*, or *GAR file*.
- This GAR file is a single file which contains all the files and information the grid services container need to *deploy* our service and make it available to the whole world.

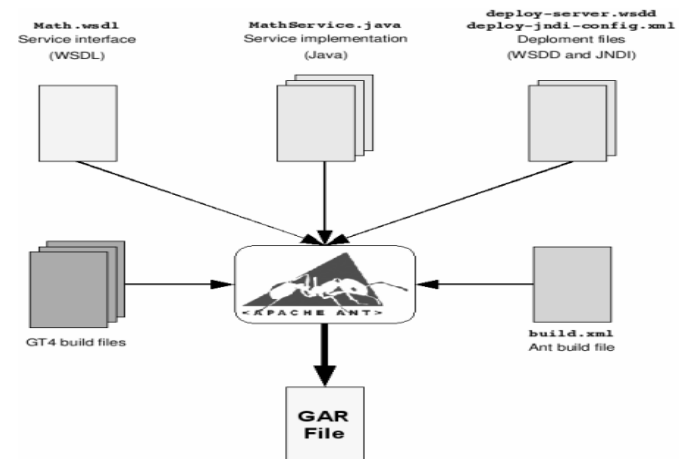
Paul A. Farrell 2006 KENT STATE Grid Computing 42

## Creating a GAR

- Processing the WSDL file to add missing pieces (such as bindings)
- Creating the stub classes from the WSDL
- Compiling the stubs classes
- Compiling the service implementation
- Organize all the files into a very specific directory structure

We'll have a detailed discussion on **ant** soon.

Paul A. Farrell 2006 KENT STATE Grid Computing 43



Paul A. Farrell 2006 KENT STATE Grid Computing 44

## globus-build-service script

---

- **One of the tools developed as part of the Globus Service Build Tools (GSBT) project**
  - <http://gsbt.sourceforge.net/>
  - **Avoids having to modify an Ant buildfile every time**

```
./globus-build-service.sh -d <service base directory> -s
 <service's WSDL file>
```

- Run this from \$TUTORIAL\_DIR
- ```
./globus-build-service.sh -d org/globus/examples/services/core/first/ -s  
  schema/examples/MathService_instance/Math.wsdl
```
- A GAR file will be generated:
\$EXAMPLES_DIR/org_globus_examples_services_core_first.gar

Paul A. Farrell 2006 KENT STATE Grid Computing 45

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*
4. **Compile everything and generate GAR file.** This is done with *Ant*
5. **Deploy service.** This is done with a *GT4 tool*

Paul A. Farrell 2006 KENT STATE Grid Computing 46

Step 5: Deploy the service into a grid service

- GAR file contains all the files and information the web server needs to deploy the Grid Service
- **globus-deploy-gar**
 - a GT4 tool that, using Ant, unpacks the GAR file and copies the files within (WSDL, compiled stubs, compiled implementation, WSDD) into key locations in the GT4 directory tree
- Deployment command must be run with a user that has write permission in \$GLOBUS_LOCATION

```
globus-deploy-gar  
  $EXAMPLES_DIR/org_globus_examples_services_core_first.gar
```

- Also undeploy tool
- ```
globus-undeploy-gar org_globus_examples_services_core_first
```

---

Paul A. Farrell 2006 KENT STATE Grid Computing 47

## A simple client

---

```
package org.globus.examples.clients.MathService_instance;
import org.apache.axis.message.addressing.Address;
import org.apache.axis.message.addressing.EndpointReferenceType;
import org.globus.examples.stubs.MathService_instance.MathPortType;
import org.globus.examples.stubs.MathService_instance.GetValueRP;
import
 org.globus.examples.stubs.MathService_instance.service.MathServiceAddressingLocator;
public class Client {
 public static void main(String[] args) {
 MathServiceAddressingLocator locator = new MathServiceAddressingLocator();
 try {
 String serviceURI = args[0];
```

---

Paul A. Farrell 2006 KENT STATE Grid Computing 48

