Hướng dẫn tạo MathService tương thích với Globus Toolkit 4.2.x

Guideline to program a MathService is compatible with Globus Toolkit 4.2.x

Mục lục

Pre-requirements:

- You have a Java Globus container of Globus Toolkit version 4.2.x is running successfully.
- You have a '*globus*' account to be able to deploy into the globus container, stop and start the globus container.

Convert from CounterService change into MathService:

1. Copy source code: 'counter' service
2. Change name to 'math'
3. Change some name:

Old name	New name
wsrf_core_samples_counter	wsrf_core_samples_math
CounterService	MathService

4. Copy a schema directory from 'counter' to 'math'

a. cd /home/globus/gt4.2.1-all-source-installer/source-trees/wsrf/schema/core/samples
b. cp –r counter math
c. Result: new path for 'math' schema: /home/globus/gt4.2.1-all-source-installer/source-trees/wsrf/schema/core/samples/math
d. Copy 'math' schema into $GLOBUS_LOCATION/share/schema/core/samples/
cp -r /home/globus/gt4.2.1-all-source-installer/source-trees/wsrf/schema/core/samples/math /usr/grid/globus-4.2.1/share/schema/core/samples/

5. Deploy math.gar into Globus container:

a. Run: /etc/init.d/globus-ws-java-container stop
b. ant clean
c. ant dist
d. ant undeploy
e. ant deploy
f. You should see your MathService is running in your globus container, for example:
[27]: https://172.28.13.11:18443/wsrf/services/MathService
g. globus@netlabserver01:/usr/grid/globus-4.2.1$ bin/math-client -s https://172.28.13.11:18443/wsrf/services/MathService -z none

6. Test MathService:

```
globus@netlabserver01:/usr/grid/globus-4.2.1$ bin/math-create -s  https://172.28.13.11:18443/wsrf/services/MathService -z n
erp
globus@netlabserver01:/usr/grid/globus-4.2.1$ cat /tmp/erp
<ns1:MathReference xsi:type="ns2:EndpointReferenceType" xmlns:ns1="http://math.com" xmlns:xsi="http://www.w3.org/20
XMLSchema-instance" xmlns:ns2="http://www.w3.org/2005/08/addressing"><ns2:Address
xsi:type="ns2:AttributedURI">https://172.28.13.11:18443/wsrf/services/MathService</ns2:Address><ns2:ReferenceParamete
xsi:type="ns2:ReferenceParametersType"><ns1:MathKey>15075988</ns1:MathKey></ns2:ReferenceParameters></ns1:Mat
globus@netlabserver01:/usr/grid/globus-4.2.1$ bin/math-add -e /tmp/erp 2
2
globus@netlabserver01:/usr/grid/globus-4.2.1$ bin/math-add -e /tmp/erp 10
12
globus@netlabserver01:/usr/grid/globus-4.2.1$ bin/math-add -e /tmp/erp 15
27
```

Source code:

Bước 1: Tạo ra các file cho dịch vụ MathService

Yêu cầu:

- Bạn nên đọc hướng dẫn Tutorial để biết 5 bước cần tạo một Grid service.
- Link: http://gdp.globus.org/gt4-tutorial/multiplehtml/pt02.html
- Copy các file trong ví dụ mẫu của dịch vụ 'CounterService' để tạo ra 'MathService'. Thực hiện các bước bên trên để chuyển đổi Counter* -> Math* J

/* File: **Math.java**
*/


```
package org.globus.wsrf.samples.math;

import java.util.Calendar;

import javax.xml.namespace.QName;

import org.globus.wsrf.Resource;
import org.globus.wsrf.ResourceIdentifier;
import org.globus.wsrf.ResourceProperties;
import org.globus.wsrf.ResourceProperty;
import org.globus.wsrf.ResourcePropertySet;
import org.globus.wsrf.Topic;
import org.globus.wsrf.TopicList;
import org.globus.wsrf.TopicListAccessor;
import org.globus.wsrf.ResourceLifetime;
import org.globus.wsrf.WSRFConstants;
import org.globus.wsrf.impl.ReflectionResourceProperty;
import org.globus.wsrf.impl.ResourcePropertyTopic;
import org.globus.wsrf.impl.SimpleResourcePropertySet;
```

```
import org.globus.wsrf.impl.SimpleTopicList;
import org.globus.wsrf.impl.SimpleTopic;
import org.globus.wsrf.impl.SimpleResourceProperty;
import org.globus.wsrf.impl.SimpleResourcePropertyMetaData;

/**
* Resource Implementation
*/
public class Math
    implements Resource,
               ResourceLifetime,
               ResourceIdentifier,
               ResourceProperties,
               TopicListAccessor {

    public static final QName KEY =
        new QName("http://math.com", "MathKey");

    public static final QName RP_SET =
        new QName("http://math.com", "Math");

    public static final QName VALUE =
        new QName("http://math.com", "Value");

    private ResourcePropertySet propSet;
    private TopicList topicList;

    protected Calendar terminationTime = null;
    protected Object key;
    protected ResourceProperty value;

    protected void initialize(Object key) {

        this.key = key;
        this.propSet = new SimpleResourcePropertySet(RP_SET);
        this.topicList = new SimpleTopicList(this);
        ResourceProperty prop = null;

        try {
            this.value = new ResourcePropertyTopic(
                new SimpleResourceProperty(VALUE));
            this.propSet.add(this.value);
            this.topicList.addTopic((Topic) this.value);
```

```
            this.value.add(new Integer(0));

            prop = new
    ReflectionResourceProperty(SimpleResourcePropertyMetaData.TERMINATION_TIME,
    this);
            this.propSet.add(prop);
            this.topicList.addTopic(new SimpleTopic(
                WSRFConstants.TERMINATION_TOPIC));

            prop = new
    ReflectionResourceProperty(SimpleResourcePropertyMetaData.CURRENT_TIME, this);
            this.propSet.add(prop);
        } catch(Exception e) {
            e.printStackTrace();
            throw new RuntimeException(e.getMessage());
        }
    }

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

    public TopicList getTopicList() {
        return this.topicList;
    }

    public int getValue() {
        return ((Integer) this.value.get(0)).intValue();
    }

    public void setValue(int value) {
        this.value.set(0, new Integer(value));
    }

    /**
     * Called when a new Math resource is created.
     *
     * @return the resource key
     */
    public Object create() throws Exception {
        // just an example, might be a file already...
        this.key = new Integer(hashCode());
        initialize(key);
```

```
        return key;
    }

    public Object getID() {
        return this.key;
    }

    public void setTerminationTime(Calendar time) {
        this.terminationTime = time;
    }

    public Calendar getTerminationTime() {
        return this.terminationTime;
    }

    public Calendar getCurrentTime() {
        return Calendar.getInstance();
    }

}
```

```
/* File: MathHome.java
*/
/*
* Copyright 1999-2006 University of Chicago
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
package org.globus.wsrf.samples.math;
```

```
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

import org.globus.wsrf.ResourceKey;
import org.globus.wsrf.impl.ResourceHomeImpl;
import org.globus.wsrf.impl.SimpleResourceKey;

public class MathHome extends ResourceHomeImpl {

    static Log logger =
        LogFactory.getLog(MathHome.class.getName());

    public ResourceKey create() throws Exception {
        Math math = (Math)createNewInstance();
        math.create();
        ResourceKey key = new SimpleResourceKey(keyTypeName,
                                        math.getID());
        add(key, math);
        return key;
    }

}
```

/* File: **PersistentMath.java**
*/


```
package org.globus.wsrf.samples.math;
```

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.File;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.Calendar;

import org.globus.wsrf.ResourceKey;
import org.globus.wsrf.ResourceException;
import org.globus.wsrf.PersistenceCallback;
import org.globus.wsrf.RemoveCallback;
import org.globus.wsrf.NoSuchResourceException;
import org.globus.wsrf.InvalidResourceKeyException;
import org.globus.wsrf.utils.FilePersistenceHelper;
import org.globus.wsrf.utils.SubscriptionPersistenceUtils;

/**
* Persistent Resource Implementation
*/
public class PersistentMath
    extends Math
    implements PersistenceCallback, RemoveCallback {

    private FilePersistenceHelper persistenceHelper;

    public void setValue(int value) {
        super.setValue(value);
        try {
            store();
        } catch (Exception e) {
            throw new RuntimeException(e.getMessage());
        }
    }

    public void setTerminationTime(Calendar time) {
        super.setTerminationTime(time);
        try {
            store();
        } catch (Exception e) {
            throw new RuntimeException(e.getMessage());
        }
    }
```

```
/**
 * User-defined function.
 *
 * @return the resource key
 */
public Object create() throws Exception {
   Object key = super.create();
   store();
   return key;
}

/**
 * Called when activating a Math resource by ResourceHomeImpl
 */
public void load(ResourceKey key) throws ResourceException {
   File file = getKeyAsFile(key.getValue());
   if (!file.exists()) {
      throw new NoSuchResourceException();
   }
   initialize(key.getValue());
   FileInputStream fis = null;
   int value = 0;
   try {
      fis = new FileInputStream(file);
      ObjectInputStream ois = new ObjectInputStream(fis);
      value = ois.readInt();
      this.terminationTime = (Calendar)ois.readObject();
      SubscriptionPersistenceUtils.loadSubscriptionListeners(
         this.getTopicList(), ois);
   } catch (Exception e) {
      throw new ResourceException("Failed to load resource", e);
   } finally {
      if (fis != null) {
         try { fis.close(); } catch (Exception ee) {}
      }
   }
   this.value.set(0, new Integer(value));
}

public synchronized void store() throws ResourceException {
   FileOutputStream fos = null;
   File tmpFile = null;
```

```
        try {
            tmpFile = File.createTempFile(
                "counter", ".tmp",
                getPersistenceHelper().getStorageDirectory());
            fos = new FileOutputStream(tmpFile);
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeInt(((Integer) this.value.get(0)).intValue());
            oos.writeObject(this.terminationTime);
            SubscriptionPersistenceUtils.storeSubscriptionListeners(
                this.getTopicList(), oos);
        } catch (Exception e) {
            if (tmpFile != null) {
                tmpFile.delete();
            }
            throw new ResourceException("Failed to store resource", e);
        } finally {
            if (fos != null) {
                try { fos.close();} catch (Exception ee) {}
            }
        }

        File file = getKeyAsFile(this.key);
        if (file.exists()) {
            file.delete();
        }
        if (!tmpFile.renameTo(file)) {
            tmpFile.delete();
            throw new ResourceException("Failed to store resource");
        }
    }

    private File getKeyAsFile(Object key)
        throws InvalidResourceKeyException {
        if (key instanceof Integer) {
            return getPersistenceHelper().getKeyAsFile(key);
        } else {
            throw new InvalidResourceKeyException();
        }
    }

    public void remove() throws ResourceException {
        getPersistenceHelper().remove(this.key);
```

```
    }

    protected synchronized FilePersistenceHelper getPersistenceHelper() {
        if (this.persistenceHelper == null) {
            try {
                this.persistenceHelper =
                    new FilePersistenceHelper(getClass(), ".ser");
            } catch (Exception e) {
                throw new RuntimeException(e.getMessage());
            }
        }
        return this.persistenceHelper;
    }

}
```

File: **MathService.java**

```
/*
* Copyright 1999-2006 University of Chicago
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
package org.globus.wsrf.samples.math;

import java.util.Set;
import java.util.Iterator;

import org.globus.wsrf.security.SecurityManager;

import java.rmi.RemoteException;

import org.apache.axis.MessageContext;
```

```
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

import org.globus.wsrf.utils.log4j.SystemAdminLogFactory;

import org.globus.axis.message.addressing.EndpointReferenceType;

import org.globus.wsrf.ResourceContext;
import org.globus.wsrf.ResourceKey;
import org.globus.wsrf.utils.AddressingUtils;
import org.globus.wsrf.utils.ContextUtils;
import org.globus.wsrf.utils.log4j.LogObject;

import com.math.CreateMath;
import com.math.CreateMathResponse;

import java.security.cert.X509Certificate;

import javax.security.auth.Subject;

import org.globus.wsrf.impl.security.util.EPRUtil;

/**
* Math service implementation.
*/
public class MathService
{

   private static Log logger =
         LogFactory.getLog(MathService.class.getName());

   private static Log sysLogger =
      SystemAdminLogFactory.getLog(MathService.class.getName());

   public CreateMathResponse createMath(CreateMath request)
      throws RemoteException
   {

      LogObject log = new LogObject("org.samples.math.create.start");
      sysLogger.info(log);

      ResourceContext ctx = null;
```

```
MathHome home = null;
ResourceKey key = null;

try
{
   ctx = ResourceContext.getResourceContext();
   home = (MathHome) ctx.getResourceHome();
   key = home.create();
}
catch(RemoteException e)
{
   sysLogger.error(new LogObject("org.samples.math.create.end",
                        "-1"));
   logger.error("Error creating math", e);
   throw e;
}
catch(Exception e)
{
   sysLogger.error(new LogObject("org.samples.math.create.end",
                        "-1"));
   throw new RemoteException("", e);
}

EndpointReferenceType epr = null;
try
{
   epr = AddressingUtils.createEndpointReference(ctx, key);
}
catch(Exception e)
{
   sysLogger.error(new LogObject("org.samples.math.create.end",
                         "-1"));

   throw new RemoteException("", e);
}

MessageContext msgCtx = MessageContext.getCurrentContext();
String servicePath = ContextUtils.getTargetServicePath(msgCtx);
if (servicePath.equals("SecureMathService")) {
   // try to get service's credential
   SecurityManager manager = SecurityManager.getManager(msgCtx);
   Subject subject = manager.getServiceSubject();
   Set set = subject.getPublicCredentials(X509Certificate[].class);
```

```
        Iterator it = set.iterator();
        while (it.hasNext()) {
            X509Certificate[] certs = (X509Certificate[])it.next();
            EPRUtil.insertCertificates(epr, certs, null);
            break;
        }
    }

    CreateMathResponse response = new CreateMathResponse();
    response.setEndpointReference(epr);

    sysLogger.info(new LogObject("org.samples.math.create.end",
                            "0"));

    return response;
}

public int add(int arg0) throws RemoteException
{

    sysLogger.info(new LogObject("org.samples.math.add.start"));


    Object resource = null;
    try
    {
        resource = ResourceContext.getResourceContext().getResource();
    }
    catch(RemoteException e)
    {
        sysLogger.info(new LogObject("org.samples.math.add.end",
                            "-1"));
        throw e;
    }
    catch(Exception e)
    {
        sysLogger.info(new LogObject("org.samples.math.add.end",
                            "-1"));

        throw new RemoteException("", e);
    }
    Math math = (Math) resource;
    synchronized(math)
```

```
        {
            int result = math.getValue();
            result += arg0;
            math.setValue(result);
            sysLogger.info(new LogObject("org.samples.math.add.end", "0"));
            return result;
        }
    }
}
```

Bước 2: Tạo các file WSDL cho dịch vụ MathService:

File: **MathService.wsdl**

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="Math" targetNamespace="http://math.com/service"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:binding="http://math.com/bindings"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:import namespace="http://math.com/bindings" location="math_bindings.wsdl"/>
  <wsdl:service name="MathService">
    <wsdl:port name="MathPortTypePort" binding="binding:MathPortTypeSOAPBinding">
      <soap:address location="http://localhost:8080/wsrf/services/"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

File: **deploy-server.wsdd**

```
<?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="MathService" provider="Handler"
        use="literal" style="document">
        <parameter name="allowedMethodsClass"
            value="com.math.MathPortType"/>
        <parameter name="handlerClass"
            value="org.globus.axis.providers.RPCProvider"/>
        <parameter name="className"
```

```
        value="org.globus.wsrf.samples.math.MathService"/>
      <wsdlFile>share/schema/core/samples/math/math_service.wsdl</wsdlFile>
      <parameter name="scope" value="Application"/>
      <parameter name="providers" value="
        DestroyProvider SetTerminationTimeProvider GetRPProvider
        QueryRPProvider GetMRPProvider
        SubscribeProvider GetCurrentMessageProvider"/>
  </service>
</deployment>
```

File: **math_bindings.wsdl**

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="Math" targetNamespace="http://math.com/bindings" xmlns:porttype="http://math.com"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
 <wsdl:import namespace="http://math.com" location="math_flattened.wsdl"/>
 <wsdl:binding name="MathPortTypeSOAPBinding" type="porttype:MathPortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="createMath">
   <soap:operation soapAction="http://math.com/MathPortType/createMathRequest"/>
   <wsdl:input>
    <soap:body use="literal"/>
   </wsdl:input>
   <wsdl:output>
    <soap:body use="literal"/>
   </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="add">
   <soap:operation soapAction="http://math.com/MathPortType/addRequest"/>
   <wsdl:input>
    <soap:body use="literal"/>
   </wsdl:input>
   <wsdl:output>
    <soap:body use="literal"/>
   </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetResourceProperty">
   <soap:operation soapAction="http://docs.oasis-open.org/wsrf/rpw-2/GetResourceProperty/
GetResourcePropertyRequest"/>
   <wsdl:input>
    <soap:body use="literal"/>
   </wsdl:input>
   <wsdl:output>
```

```
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="InvalidResourcePropertyQNameFault">
      <soap:fault name="InvalidResourcePropertyQNameFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ResourceUnavailableFault">
      <soap:fault name="ResourceUnavailableFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ResourceUnknownFault">
      <soap:fault name="ResourceUnknownFault" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="QueryResourceProperties">
    <soap:operation soapAction="http://docs.oasis-open.org/wsrf/rpw-2/QueryResourceProperties/
QueryResourcePropertiesRequest"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="InvalidResourcePropertyQNameFault">
      <soap:fault name="InvalidResourcePropertyQNameFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="InvalidQueryExpressionFault">
      <soap:fault name="InvalidQueryExpressionFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="QueryEvaluationErrorFault">
      <soap:fault name="QueryEvaluationErrorFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ResourceUnavailableFault">
      <soap:fault name="ResourceUnavailableFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ResourceUnknownFault">
      <soap:fault name="ResourceUnknownFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="UnknownQueryExpressionDialectFault">
      <soap:fault name="UnknownQueryExpressionDialectFault" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="GetMultipleResourceProperties">
```

```
    <soap:operation soapAction="http://docs.oasis-open.org/wsrf/rpw-2/GetMultipleResourceProperties/
GetMultipleResourcePropertiesRequest"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="InvalidResourcePropertyQNameFault">
      <soap:fault name="InvalidResourcePropertyQNameFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ResourceUnavailableFault">
      <soap:fault name="ResourceUnavailableFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ResourceUnknownFault">
      <soap:fault name="ResourceUnknownFault" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="Subscribe">
    <soap:operation soapAction="http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/SubscribeRequest"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="TopicNotSupportedFault">
      <soap:fault name="TopicNotSupportedFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="InvalidTopicExpressionFault">
      <soap:fault name="InvalidTopicExpressionFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="UnsupportedPolicyRequestFault">
      <soap:fault name="UnsupportedPolicyRequestFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="UnacceptableInitialTerminationTimeFault">
      <soap:fault name="UnacceptableInitialTerminationTimeFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="InvalidMessageContentExpressionFault">
      <soap:fault name="InvalidMessageContentExpressionFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="InvalidProducerPropertiesExpressionFault">
```

```
      <soap:fault name="InvalidProducerPropertiesExpressionFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="SubscribeCreationFailedFault">
      <soap:fault name="SubscribeCreationFailedFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="TopicExpressionDialectUnknownFault">
      <soap:fault name="TopicExpressionDialectUnknownFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="InvalidFilterFault">
      <soap:fault name="InvalidFilterFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ResourceUnknownFault">
      <soap:fault name="ResourceUnknownFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="NotifyMessageNotSupportedFault">
      <soap:fault name="NotifyMessageNotSupportedFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="UnrecognizedPolicyRequestFault">
      <soap:fault name="UnrecognizedPolicyRequestFault" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="GetCurrentMessage">
    <soap:operation soapAction="http://docs.oasis-open.org/wsn/bw-2/NotificationProducer/
GetCurrentMessageRequest"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="MultipleTopicsSpecifiedFault">
      <soap:fault name="MultipleTopicsSpecifiedFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="TopicNotSupportedFault">
      <soap:fault name="TopicNotSupportedFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="InvalidTopicExpressionFault">
      <soap:fault name="InvalidTopicExpressionFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="NoCurrentMessageOnTopicFault">
      <soap:fault name="NoCurrentMessageOnTopicFault" use="literal"/>
    </wsdl:fault>
```

```
    <wsdl:fault name="TopicExpressionDialectUnknownFault">
      <soap:fault name="TopicExpressionDialectUnknownFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ResourceUnknownFault">
      <soap:fault name="ResourceUnknownFault" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="Destroy">
    <soap:operation soapAction="http://docs.oasis-open.org/wsrf/rlw-2/ImmediateResourceTermination/
DestroyRequest"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ResourceNotDestroyedFault">
      <soap:fault name="ResourceNotDestroyedFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ResourceUnavailableFault">
      <soap:fault name="ResourceUnavailableFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ResourceUnknownFault">
      <soap:fault name="ResourceUnknownFault" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="SetTerminationTime">
    <soap:operation soapAction="http://docs.oasis-open.org/wsrf/rlw-2/ScheduledResourceTermination/
SetTerminationTimeRequest"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ResourceUnavailableFault">
      <soap:fault name="ResourceUnavailableFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="UnableToSetTerminationTimeFault">
      <soap:fault name="UnableToSetTerminationTimeFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="ResourceUnknownFault">
```

```
      <soap:fault name="ResourceUnknownFault" use="literal"/>
    </wsdl:fault>
    <wsdl:fault name="TerminationTimeChangeRejectedFault">
      <soap:fault name="TerminationTimeChangeRejectedFault" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
 </wsdl:binding>
</wsdl:definitions>
```

File: **math_flattened.wsdl**

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="Math" targetNamespace="http://math.com" xmlns:wsrp="http://docs.oasis-
open.org/wsrf/rp-2" xmlns:tns="http://math.com" xmlns:wsrpw="http://docs.oasis-open.org/wsrf/rpw-2"
xmlns:wsrlw="http://docs.oasis-open.org/wsrf/rlw-2" xmlns:wsdlpp="http://www.globus.org/namespaces/
2004/10/WSDLPreprocessor" xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl" xmlns:xsd="http://www.w3.org/2001/
XMLSchema" xmlns:wsntw="http://docs.oasis-open.org/wsn/bw-2" xmlns:gtwsdl0="http://docs.oasis-
open.org/wsrf/rw-2" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/
wsdl/">
  <wsdl:import namespace="http://docs.oasis-open.org/wsn/bw-2" location="../../../wsrf/notification/
bw-2.wsdl"/>
  <wsdl:import namespace="http://docs.oasis-open.org/wsrf/rpw-2" location="../../../wsrf/properties/
rpw-2.wsdl"/>
  <wsdl:import namespace="http://docs.oasis-open.org/wsrf/rlw-2" location="../../../wsrf/lifetime/
rlw-2.wsdl"/>
  <wsdl:import namespace="http://docs.oasis-open.org/wsrf/rw-2" location="../../../wsrf/resource/
rw-2.wsdl"/>
  <wsdl:types>
    <xsd:schema elementFormDefault="qualified" targetNamespace="http://math.com"
xmlns:rpns0="http://docs.oasis-open.org/wsrf/rl-2" xmlns:tns="http://math.com"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:import namespace="http://docs.oasis-open.org/wsrf/rl-2" schemaLocation="../../../wsrf/lifetime/
rl-2.xsd"/>

<xsd:import namespace="http://docs.oasis-open.org/wsrf/rp-2" schemaLocation="../../../wsrf/properties/
rp-2.xsd"/>


        <xsd:import namespace="http://www.w3.org/2005/08/addressing" schemaLocation="../../../ws/
addressing/ws-addr.xsd"/>

        <xsd:element name="createMath">
          <xsd:complexType/>
```

```
        </xsd:element>

        <xsd:element name="createMathResponse">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element ref="wsa:EndpointReference"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>

        <xsd:element name="add" type="xsd:int"/>
        <xsd:element name="addResponse" type="xsd:int"/>

        <xsd:element name="Value" type="xsd:int"/>

        <xsd:element name="MathRP">
           <xsd:complexType>
              <xsd:sequence>
                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:Value"/>

        <xsd:element maxOccurs="unbounded" minOccurs="0" ref="wsrp:QueryExpressionDialect"/>
        <xsd:element maxOccurs="1" minOccurs="1" ref="rpns0:TerminationTime"/>
        <xsd:element maxOccurs="1" minOccurs="1" ref="rpns0:CurrentTime"/>
       </xsd:sequence>
           </xsd:complexType>
        </xsd:element>

     </xsd:schema>
  </wsdl:types>
  <wsdl:message name="AddInputMessage">
    <wsdl:part name="parameters" element="tns:add"/>
  </wsdl:message>
  <wsdl:message name="AddOutputMessage">
    <wsdl:part name="parameters" element="tns:addResponse"/>
  </wsdl:message>
  <wsdl:message name="CreateMathResponse">
    <wsdl:part name="response" element="tns:createMathResponse"/>
  </wsdl:message>
  <wsdl:message name="CreateMathRequest">
    <wsdl:part name="request" element="tns:createMath"/>
  </wsdl:message>
  <wsdl:portType name="MathPortType" wsrp:ResourceProperties="tns:MathRP">
    <wsdl:operation name="createMath">
      <wsdl:input message="tns:CreateMathRequest"/>
```

```
    <wsdl:output message="tns:CreateMathResponse"/>
  </wsdl:operation>
  <wsdl:operation name="add">
    <wsdl:input message="tns:AddInputMessage"/>
    <wsdl:output message="tns:AddOutputMessage"/>
  </wsdl:operation>
  <wsdl:operation name="GetResourceProperty">
    <wsdl:input name="GetResourcePropertyRequest" message="wsrpw:GetResourcePropertyRequest"
wsaw:Action="http://docs.oasis-open.org/wsrf/rpw-2/GetResourceProperty/
GetResourcePropertyRequest"/>
    <wsdl:output name="GetResourcePropertyResponse"
message="wsrpw:GetResourcePropertyResponse" wsaw:Action="http://docs.oasis-open.org/wsrf/rpw-2/
GetResourceProperty/GetResourcePropertyResponse"/>
    <wsdl:fault name="InvalidResourcePropertyQNameFault"
message="wsrpw:InvalidResourcePropertyQNameFault" wsaw:Action="http://docs.oasis-open.org/wsrf/
fault"/>
    <wsdl:fault name="ResourceUnavailableFault" message="gtwsdl0:ResourceUnavailableFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/fault"/>
    <wsdl:fault name="ResourceUnknownFault" message="gtwsdl0:ResourceUnknownFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/fault"/>
  </wsdl:operation>
  <wsdl:operation name="QueryResourceProperties">
    <wsdl:input name="QueryResourcePropertiesRequest"
message="wsrpw:QueryResourcePropertiesRequest" wsaw:Action="http://docs.oasis-open.org/wsrf/
rpw-2/QueryResourceProperties/QueryResourcePropertiesRequest"/>
    <wsdl:output name="QueryResourcePropertiesResponse"
message="wsrpw:QueryResourcePropertiesResponse" wsaw:Action="http://docs.oasis-open.org/wsrf/
rpw-2/QueryResourceProperties/QueryResourcePropertiesResponse"/>
    <wsdl:fault name="InvalidResourcePropertyQNameFault"
message="wsrpw:InvalidResourcePropertyQNameFault" wsaw:Action="http://docs.oasis-open.org/wsrf/
fault"/>
    <wsdl:fault name="InvalidQueryExpressionFault" message="wsrpw:InvalidQueryExpressionFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/fault"/>
    <wsdl:fault name="QueryEvaluationErrorFault" message="wsrpw:QueryEvaluationErrorFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/fault"/>
    <wsdl:fault name="ResourceUnavailableFault" message="gtwsdl0:ResourceUnavailableFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/fault"/>
    <wsdl:fault name="ResourceUnknownFault" message="gtwsdl0:ResourceUnknownFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/fault"/>
    <wsdl:fault name="UnknownQueryExpressionDialectFault"
message="wsrpw:UnknownQueryExpressionDialectFault" wsaw:Action="http://docs.oasis-open.org/wsrf/
fault"/>
  </wsdl:operation>
  <wsdl:operation name="GetMultipleResourceProperties">
```

```
    <wsdl:input name="GetMultipleResourcePropertiesRequest"
message="wsrpw:GetMultipleResourcePropertiesRequest" wsaw:Action="http://docs.oasis-open.org/wsrf/
rpw-2/GetMultipleResourceProperties/GetMultipleResourcePropertiesRequest"/>
    <wsdl:output name="GetMultipleResourcePropertiesResponse"
message="wsrpw:GetMultipleResourcePropertiesResponse" wsaw:Action="http://docs.oasis-open.org/
wsrf/rpw-2/GetMultipleResourceProperties/GetMultipleResourcePropertiesResponse"/>
    <wsdl:fault name="InvalidResourcePropertyQNameFault"
message="wsrpw:InvalidResourcePropertyQNameFault" wsaw:Action="http://docs.oasis-open.org/wsrf/
fault"/>
    <wsdl:fault name="ResourceUnavailableFault" message="gtwsdl0:ResourceUnavailableFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/fault"/>
    <wsdl:fault name="ResourceUnknownFault" message="gtwsdl0:ResourceUnknownFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/fault"/>
  </wsdl:operation>
  <wsdl:operation name="Subscribe">
    <wsdl:input message="wsntw:SubscribeRequest" wsaw:Action="http://docs.oasis-open.org/wsn/bw-2/
NotificationProducer/SubscribeRequest"/>
    <wsdl:output message="wsntw:SubscribeResponse" wsaw:Action="http://docs.oasis-open.org/wsn/
bw-2/NotificationProducer/SubscribeResponse"/>
    <wsdl:fault name="TopicNotSupportedFault" message="wsntw:TopicNotSupportedFault"
wsaw:Action="http://docs.oasis-open.org/wsn/fault"/>
    <wsdl:fault name="InvalidTopicExpressionFault" message="wsntw:InvalidTopicExpressionFault"
wsaw:Action="http://docs.oasis-open.org/wsn/fault"/>
    <wsdl:fault name="UnsupportedPolicyRequestFault"
message="wsntw:UnsupportedPolicyRequestFault" wsaw:Action="http://docs.oasis-open.org/wsn/fault"/>
    <wsdl:fault name="UnacceptableInitialTerminationTimeFault"
message="wsntw:UnacceptableInitialTerminationTimeFault" wsaw:Action="http://docs.oasis-open.org/
wsn/fault"/>
    <wsdl:fault name="InvalidMessageContentExpressionFault"
message="wsntw:InvalidMessageContentExpressionFault" wsaw:Action="http://docs.oasis-open.org/wsn/
fault"/>
    <wsdl:fault name="InvalidProducerPropertiesExpressionFault"
message="wsntw:InvalidProducerPropertiesExpressionFault" wsaw:Action="http://docs.oasis-open.org/
wsn/fault"/>
    <wsdl:fault name="SubscribeCreationFailedFault" message="wsntw:SubscribeCreationFailedFault"
wsaw:Action="http://docs.oasis-open.org/wsn/fault"/>
    <wsdl:fault name="TopicExpressionDialectUnknownFault"
message="wsntw:TopicExpressionDialectUnknownFault" wsaw:Action="http://docs.oasis-open.org/wsn/
fault"/>
    <wsdl:fault name="InvalidFilterFault" message="wsntw:InvalidFilterFault"
wsaw:Action="http://docs.oasis-open.org/wsn/fault"/>
    <wsdl:fault name="ResourceUnknownFault" message="gtwsdl0:ResourceUnknownFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/fault"/>
```

```
    <wsdl:fault name="NotifyMessageNotSupportedFault"
message="wsntw:NotifyMessageNotSupportedFault" wsaw:Action="http://docs.oasis-open.org/wsn/
fault"/>
    <wsdl:fault name="UnrecognizedPolicyRequestFault"
message="wsntw:UnrecognizedPolicyRequestFault" wsaw:Action="http://docs.oasis-open.org/wsn/
fault"/>
  </wsdl:operation>
  <wsdl:operation name="GetCurrentMessage">
    <wsdl:input message="wsntw:GetCurrentMessageRequest" wsaw:Action="http://docs.oasis-open.org/
wsn/bw-2/NotificationProducer/GetCurrentMessageRequest"/>
    <wsdl:output message="wsntw:GetCurrentMessageResponse" wsaw:Action="http://docs.oasis-
open.org/wsn/bw-2/NotificationProducer/GetCurrentMessageResponse"/>
    <wsdl:fault name="MultipleTopicsSpecifiedFault" message="wsntw:MultipleTopicsSpecifiedFault"
wsaw:Action="http://docs.oasis-open.org/wsn/fault"/>
    <wsdl:fault name="TopicNotSupportedFault" message="wsntw:TopicNotSupportedFault"
wsaw:Action="http://docs.oasis-open.org/wsn/fault"/>
    <wsdl:fault name="InvalidTopicExpressionFault" message="wsntw:InvalidTopicExpressionFault"
wsaw:Action="http://docs.oasis-open.org/wsn/fault"/>
    <wsdl:fault name="NoCurrentMessageOnTopicFault"
message="wsntw:NoCurrentMessageOnTopicFault" wsaw:Action="http://docs.oasis-open.org/wsn/
fault"/>
    <wsdl:fault name="TopicExpressionDialectUnknownFault"
message="wsntw:TopicExpressionDialectUnknownFault" wsaw:Action="http://docs.oasis-open.org/wsn/
fault"/>
    <wsdl:fault name="ResourceUnknownFault" message="gtwsdl0:ResourceUnknownFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/fault"/>
  </wsdl:operation>
  <wsdl:operation name="Destroy">
    <wsdl:input name="DestroyRequest" message="wsrlw:DestroyRequest"
wsaw:Action="http://docs.oasis-open.org/wsrf/rlw-2/ImmediateResourceTermination/DestroyRequest"/>
    <wsdl:output name="DestroyResponse" message="wsrlw:DestroyResponse"
wsaw:Action="http://docs.oasis-open.org/wsrf/rlw-2/ImmediateResourceTermination/DestroyResponse"/>
    <wsdl:fault name="ResourceNotDestroyedFault" message="wsrlw:ResourceNotDestroyedFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/faults"/>
    <wsdl:fault name="ResourceUnavailableFault" message="gtwsdl0:ResourceUnavailableFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/faults"/>
    <wsdl:fault name="ResourceUnknownFault" message="gtwsdl0:ResourceUnknownFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/faults"/>
  </wsdl:operation>
  <wsdl:operation name="SetTerminationTime">
    <wsdl:input name="SetTerminationTimeRequest" message="wsrlw:SetTerminationTimeRequest"
wsaw:Action="http://docs.oasis-open.org/wsrf/rlw-2/ScheduledResourceTermination/
SetTerminationTimeRequest"/>
```

```
    <wsdl:output name="SetTerminationTimeResponse"
message="wsrlw:SetTerminationTimeResponse" wsaw:Action="http://docs.oasis-open.org/wsrf/rlw-2/
ScheduledResourceTermination/SetTerminationTimeResponse"/>
    <wsdl:fault name="ResourceUnavailableFault" message="gtwsdl0:ResourceUnavailableFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/faults"/>
    <wsdl:fault name="UnableToSetTerminationTimeFault"
message="wsrlw:UnableToSetTerminationTimeFault"/>
    <wsdl:fault name="ResourceUnknownFault" message="gtwsdl0:ResourceUnknownFault"
wsaw:Action="http://docs.oasis-open.org/wsrf/faults"/>
    <wsdl:fault name="TerminationTimeChangeRejectedFault"
message="wsrlw:TerminationTimeChangeRejectedFault" wsaw:Action="http://docs.oasis-open.org/wsrf/
faults"/>
   </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

Bước 3: Tạo các file để chuẩn bị build 'MathService' bằng công cụ Apache Ant:

File: **filelist**

```
/lib/wsrf_core_samples_math.jar
/lib/wsrf_core_samples_math_stubs.jar
/etc/gpt/packages/globus_wsrf_core_samples_math/undeploy.xml
/etc/globus_wsrf_core_samples_math/security-config.xml
/etc/globus_wsrf_core_samples_math/server-config.wsdd
/etc/globus_wsrf_core_samples_math/jndi-config.xml
```

File: **build.xml**

- File build.xml sẽ được công cụ Apache Ant đọc và sử dụng trong quá trình thực thi các chỉ thị theo yêu cầu.
- Những lệnh thường dùng: **ant clean, ant dist, ant undeploy, ant deploy, ant all**

```
<?xml version="1.0"?>

<project default="all" name="globus_wsrf_core_samples" basedir=".">
  <description>
    WSRF Core Math Sample
  </description>
```

```
<!--
Give user a chance to override without editing this file
(and without typing -D each time it compiles it)
-->

<property environment="env"/>

<property file="build.properties"/>
<property file="${user.home}/build.properties"/>

<!--property name="env.GLOBUS_LOCATION" value="../../../../install"/ -->
        <property name="env.GLOBUS_LOCATION" value="/usr/grid/globus-4.2.1"/>

<property name="deploy.dir" location="${env.GLOBUS_LOCATION}"/>
<property name="base.name" value="wsrf_core_samples_math"/>
<property name="package.name" value="globus_${base.name}"/>
<property name="gar.name" value="${package.name}.gar"/>
<property name="jar.name" value="${base.name}.jar"/>
<property name="stubs.jar.name" value="${base.name}_stubs.jar"/>
<property name="build.dir"  location="build"/>
<property name="build.dest" location="build/classes"/>
<property name="build.lib.dir" location="build/lib"/>
<property name="stubs.dir" location="build/stubs"/>
<property name="stubs.src" location="build/stubs/src"/>
<property name="stubs.dest" location="build/stubs/classes"/>
<property name="build.packages" location=
   "${deploy.dir}/share/globus_wsrf_common/build-packages.xml"/>
<property name="build.stubs" location=
   "${deploy.dir}/share/globus_wsrf_tools/build-stubs.xml"/>
<property name="java.debug" value="on"/>

<property name="schema.src" location="${deploy.dir}/share/schema"/>
<property name="schema.local"   location="schema"/>
<property name="schema.dest"  location="${build.dir}/schema"/>

<property name="garjars.id" value="garjars"/>
<fileset dir="${build.lib.dir}" id="garjars"/>

<property name="garschema.id" value="garschema"/>
<fileset dir="${schema.dest}" id="garschema">
   <include name="core/samples/**/*"/>
</fileset>
```

```
<property name="garetc.id" value="garetc"/>
<fileset dir="etc" id="garetc"/>

<target name="init">
   <mkdir dir="${build.dir}"/>
   <mkdir dir="${build.dest}"/>
   <mkdir dir="${build.lib.dir}"/>

   <mkdir dir="${stubs.dir}"/>
   <mkdir dir="${stubs.src}"/>
   <mkdir dir="${stubs.dest}"/>

   <mkdir dir="${schema.dest}"/>
   <copy toDir="${schema.dest}">

      <fileset dir="${schema.src}" casesensitive="yes">
         <include name="wsrf/**/*"/>
         <include name="ws/**/*"/>
         <include name="core/samples/math/*"/>
      </fileset>
   </copy>

   <available property="stubs.present" type="dir"
          file="${stubs.src}/com/math" />
</target>

<target name="stubs" unless="stubs.present" depends="init">
   <ant antfile="${build.stubs}" target="generateStubs">
      <property name="source.stubs.dir"
         location="${schema.dest}/core/samples/math"/>
      <property name="target.stubs.dir" location="${stubs.src}"/>
      <property name="wsdl.file"
         value="math_service.wsdl"/>
   </ant>
</target>

<target name="compileStubs" depends="stubs">
  <javac srcdir="${stubs.src}" destdir="${stubs.dest}"
      debug="${java.debug}">
      <include name="**/*.java"/>
      <classpath>
         <fileset dir="${deploy.dir}/lib">
            <include name="common/*.jar"/>
```

```
            <include name="*.jar"/>
            <exclude name="${stubs.jar.name}"/>
            <exclude name="${jar.name}"/>
          </fileset>
        </classpath>
    </javac>
          <copy todir="${build.dest}" >
              <fileset dir="src" includes="**/*.properties" />
              <fileset dir="src" includes="**/*.xml" />
          </copy>
</target>

<target name="compile" depends="compileStubs">
    <javac srcdir="src" destdir="${build.dest}"
        debug="${java.debug}">
        <include name="**/*.java"/>
        <classpath>
          <pathelement location="${stubs.dest}"/>
          <fileset dir="${deploy.dir}/lib">
            <include name="common/*.jar"/>
            <include name="*.jar"/>
            <exclude name="${stubs.jar.name}"/>
            <exclude name="${jar.name}"/>
          </fileset>
        </classpath>
    </javac>
</target>

<target name="jarStubs" depends="compileStubs">
    <jar destfile="${build.lib.dir}/${stubs.jar.name}"
         basedir="${stubs.dest}"/>
</target>

<target name="jar" depends="compile">
    <jar destfile="${build.lib.dir}/${jar.name}" basedir="${build.dest}"/>
</target>

<target name="dist" depends="jarStubs, jar">
    <ant antfile="${build.packages}" target="makeGar">
        <reference refid="${garjars.id}"/>
        <reference refid="${garschema.id}"/>
        <reference refid="${garetc.id}"/>
    </ant>
```

```
    </target>

    <target name="clean">
        <delete dir="tmp"/>
        <delete dir="${build.dir}"/>
        <delete file="${gar.name}"/>
    </target>

    <target name="deploy" depends="dist">
        <ant antfile="${build.packages}" target="deployGar"/>
    </target>

    <target name="undeploy">
        <ant antfile="${build.packages}" target="undeployGar">
            <property name="gar.id" value="${package.name}"/>
        </ant>
    </target>

    <target name="all" depends="deploy"/>

</project>
```

Bước 4: Viết phần client của dịch vụ MathService

File: **MathClient.java**

```
/*
* Copyright 1999-2006 University of Chicago
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
```

```
package org.globus.wsrf.samples.math.client;

import java.util.List;
import java.util.Vector;
import java.util.Map;
import java.util.HashMap;
import java.security.cert.X509Certificate;

import javax.xml.rpc.Stub;

import org.globus.axis.message.addressing.EndpointReferenceType;

import org.oasis.wsn.FilterType;
import org.oasis.wsn.TopicExpressionType;
import org.oasis.wsn.Subscribe;
import org.oasis.wsn.Unsubscribe;
import org.oasis.wsn.NotificationMessageHolderTypeMessage;
import org.oasis.wsrf.lifetime.Destroy;
import org.oasis.wsrf.properties.ResourcePropertyValueChangeNotificationType;
import org.oasis.wsrf.properties.GetResourcePropertyResponse;

import org.globus.wsrf.encoding.ObjectSerializer;

import org.globus.wsrf.NotificationConsumerManager;
import org.globus.wsrf.NotifyCallback;
import org.globus.wsrf.WSNConstants;
import org.globus.wsrf.core.notification.ResourcePropertyValueChangeNotificationElementType;
import org.globus.wsrf.core.notification.SubscriptionManager;
import org.globus.wsrf.core.notification.service.SubscriptionManagerServiceAddressingLocator;
import org.globus.wsrf.utils.FaultHelper;
import org.globus.axis.util.Util;

import org.globus.wsrf.impl.security.authentication.Constants;
import org.globus.wsrf.security.authorization.client.Authorization;
import org.globus.wsrf.impl.security.authorization.IdentityAuthorization;
import org.globus.wsrf.impl.security.authorization.SelfAuthorization;
import org.globus.wsrf.impl.security.descriptor.GSISecureConvAuthMethod;
import org.globus.wsrf.impl.security.descriptor.GSISecureMsgAuthMethod;
import org.globus.wsrf.impl.security.descriptor.GSITransportAuthMethod;
import org.globus.wsrf.impl.security.descriptor.ResourceSecurityDescriptor;
import org.globus.wsrf.samples.math.Math;

import org.globus.wsrf.client.BaseClient;
```

```
import org.globus.security.gridmap.GridMap;

import com.math.MathPortType;
import com.math.CreateMath;
import com.math.CreateMathResponse;
import com.math.service.MathServiceAddressingLocator;

import org.globus.wsrf.container.ServiceContainer;
import org.globus.wsrf.encoding.DeserializationException;
import org.globus.wsrf.encoding.ObjectDeserializer;

import org.globus.wsrf.impl.security.util.EPRUtil;

import org.apache.axis.message.MessageElement;

public class MathClient extends BaseClient implements NotifyCallback {

   static {
      Util.registerTransport();
   }

   public static void main(String[] args) {

      MathClient client = new MathClient();

      try {
         client.parse(args);
      } catch (Exception e) {
         System.err.println("Error: " + e.getMessage());
         System.exit(COMMAND_LINE_ERROR);
      }

      MathServiceAddressingLocator locator =
         new MathServiceAddressingLocator();

      NotificationConsumerManager consumer = null;

      try {

         // Create counter resource
         MathPortType port =
            locator.getMathPortTypePort(client.getEPR());
```

```
client.setOptions((Stub)port);
CreateMathResponse createResponse =
   port.createMath(new CreateMath());

// Process EPR
EndpointReferenceType counterEPR =
   createResponse.getEndpointReference();

// Use these certs to get identity of server.
X509Certificate[] certs = EPRUtil.extractCertificates(counterEPR);

// Math resource reference
MathPortType counterPort =
   locator.getMathPortTypePort(counterEPR);
client.setOptions((Stub)counterPort);

// Create client side notification consumer
String scheme = client.getEPR().getAddress().getScheme();
if (scheme.equals("https")) {
   Map properties = new HashMap();
   properties
      .put(ServiceContainer.CLASS,
         "org.globus.wsrf.container.GSIServiceContainer");
   consumer = NotificationConsumerManager.getInstance(properties);
} else {
   consumer = NotificationConsumerManager.getInstance();
}
consumer.startListening();

EndpointReferenceType consumerEPR = null;
Authorization authorization =
   (Authorization)((Stub)counterPort)
   ._getProperty(Constants.AUTHORIZATION);
if (authorization != null) {
   ResourceSecurityDescriptor resDesc =
      new ResourceSecurityDescriptor();
   String authz = Authorization.AUTHZ_NONE;
   if (authorization instanceof SelfAuthorization) {
      authz = Authorization.AUTHZ_SELF;
   }
   else if (authorization instanceof IdentityAuthorization) {
      GridMap gridMap = new GridMap();
      gridMap.map(
```

```
            ( (IdentityAuthorization) authorization).getIdentity(),
            "temp");
        resDesc.setDefaultGridMap(gridMap);
        authz = Authorization.AUTHZ_GRIDMAP;
    }
    resDesc.setPDP(authz);

    Vector authMethod = new Vector();
    authMethod.add(GSISecureMsgAuthMethod.BOTH);
    authMethod.add(GSISecureConvAuthMethod.BOTH);
    authMethod.add(GSITransportAuthMethod.BOTH);
    resDesc.setDefaultAuthMethods(authMethod);

    consumerEPR =
        consumer.createNotificationConsumer(client, resDesc);
} else {
    consumerEPR =
        consumer.createNotificationConsumer(client);
}


// Subscribe request
Subscribe request = new Subscribe();
request.setConsumerReference(consumerEPR);
TopicExpressionType topicExpression = new TopicExpressionType();
topicExpression.setDialect(WSNConstants.SIMPLE_TOPIC_DIALECT);
topicExpression.setValue(Math.VALUE);
MessageElement element =
    (MessageElement)ObjectSerializer
    .toSOAPElement(topicExpression, WSNConstants.TOPIC_EXPRESSION);
FilterType filter = new FilterType();
filter.set_any(new MessageElement[] { element });
request.setFilter(filter);

EndpointReferenceType subscriptionEPR =
    counterPort.subscribe(request).getSubscriptionReference();
SubscriptionManagerServiceAddressingLocator
    subscriptionManagerLocator =
    new SubscriptionManagerServiceAddressingLocator();
SubscriptionManager subscriptionPort =
    subscriptionManagerLocator.getSubscriptionManagerPort(
        subscriptionEPR);
```

```
    // Add
    MathPortType addPort =
       locator.getMathPortTypePort(counterEPR);
    client.setOptions((Stub)addPort);

    // Add 3
    addPort.add(3);

    synchronized (client) {
       client.wait(1000 * 30);
       if (!client.called) {
          System.err.println("Did not receive notification in time!");
       }
    }

    // Get the value RP
    GetResourcePropertyResponse getRPResponse =
       counterPort.getResourceProperty(Math.VALUE);
    try {
       Integer value = (Integer)ObjectDeserializer.getFirstAsObject(
                              getRPResponse, Integer.class);
       System.out.println("Math has value: " + value);
    } catch (DeserializationException e) {
       System.err.println("Error deserializing value: " +
                     e.getMessage());
    }

    // same security setting
    // some line
    addPort.add(10);

    // Destroy the subscription
    client.setOptions((Stub)subscriptionPort);
    subscriptionPort.unsubscribe(new Unsubscribe());
    //subscriptionPort.destroy(new Destroy());

    MathPortType destroyPort =
       locator.getMathPortTypePort(counterEPR);
    client.setOptions((Stub)destroyPort);
    // Destroy the counter resource
    destroyPort.destroy(new Destroy());
  } catch(Exception e) {
    if (client.isDebugMode()) {
```

```
            FaultHelper.printStackTrace(e);
        } else {
            System.err.println("Error: " + FaultHelper.getMessage(e));
        }
        System.exit(APPLICATION_ERROR);
    } finally {
        if (consumer != null) {
            try { consumer.stopListening(); } catch (Exception ee) {}
        }
    }

}

boolean called = false;

// Notification callback
public void deliver(List topicPath,
                    EndpointReferenceType producer,
                    Object message) {

    NotificationMessageHolderTypeMessage notifMsg =
            (NotificationMessageHolderTypeMessage)message;
    MessageElement[] msgElem = notifMsg.get_any();
    if (msgElem != null) {
            ResourcePropertyValueChangeNotificationType changeMessage
                    = null;
        try {
            changeMessage = (ResourcePropertyValueChangeNotificationType)ObjectDeserializer.
                    toObject(msgElem[0],
                            ResourcePropertyValueChangeNotificationType.class);
        } catch (Exception e) {
            // FIXME
        }

        if(changeMessage != null) {
            try {
                Integer value = (Integer)ObjectDeserializer.getFirstAsObject(
                        changeMessage.getNewValues(), Integer.class);
                System.out.println("Got notification with value: " + value);
            } catch (DeserializationException e) {
                System.err.println("Error deserializing value: " +
                                e.getMessage());
            }
```

```
            }
            synchronized (this) {
                called = true;
                notify();
            }
        }
    }

}
```

File: **Create.java**

```
/*
* Copyright 1999-2006 University of Chicago
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
package org.globus.wsrf.samples.math.client;

import org.globus.axis.message.addressing.EndpointReferenceType;

import com.math.MathPortType;
import com.math.CreateMathResponse;
import com.math.CreateMath;
import com.math.service.MathServiceAddressingLocator;

import javax.xml.rpc.Stub;

import org.globus.wsrf.encoding.ObjectSerializer;
import org.globus.wsrf.utils.FaultHelper;
import org.globus.axis.util.Util;

import javax.xml.namespace.QName;
```

```
import org.globus.wsrf.client.BaseClient;

import org.globus.wsrf.impl.security.util.EPRUtil;

import java.io.FileWriter;

import org.apache.commons.cli.Option;
import org.apache.commons.cli.CommandLine;
import org.apache.commons.cli.OptionBuilder;

import java.util.Properties;

public class Create extends BaseClient {

    String eprAsStr = null;

    public static final Option EPR_AS_STR =
        OptionBuilder.withArgName("eprAsString" )
        .withDescription("Prints a string representation of the EPR to "
                         + "specified file")
        .withLongOpt("eprAsString").hasArgs()
        .create("b");

    private static final QName NAME =
        new QName("http://math.com", "MathReference");

    static {
        Util.registerTransport();
    }

    public Create() {
        addOption(EPR_AS_STR);
    }

    protected CommandLine parse(String [] args, Properties defaultOptions)
        throws Exception {

        CommandLine line = super.parse(args, defaultOptions);

        if (line.hasOption("b")) {
            eprAsStr = line.getOptionValue("b");
        }
```

```
    return line;
  }

  public String getEPRAsString() {
    return this.eprAsStr;
  }

  public static void main(String[] args) {

    Create client = new Create();

    try {
      client.parse(args);
    } catch (Exception e) {
      System.err.println("Error: " + e.getMessage());
      System.exit(COMMAND_LINE_ERROR);
    }

    String eprAsStrFile = client.getEPRAsString();

    MathServiceAddressingLocator locator =
      new MathServiceAddressingLocator();

    try {
      MathPortType port =
        locator.getMathPortTypePort(client.getEPR());
      client.setOptions((Stub)port);

      CreateMathResponse createResponse =
        port.createMath(new CreateMath());

      EndpointReferenceType epr =
        createResponse.getEndpointReference();

      System.out.println(ObjectSerializer.toString(epr, NAME));

      if (eprAsStrFile != null) {
        String value = EPRUtil.getEPRAsString(epr);
        FileWriter writer = null;
        try {
          writer = new FileWriter(eprAsStrFile);
          writer.write(value);
```

```
            } finally {
                if (writer != null) {
                    writer.close();
                }
            }
        }
    } catch(Exception e) {
        if (client.isDebugMode()) {
            FaultHelper.printStackTrace(e);
        } else {
            System.err.println("Error: " + FaultHelper.getMessage(e));
        }
        System.exit(APPLICATION_ERROR);
    }
  }

}
```

File: **Add.java**

```
/*
* Copyright 1999-2006 University of Chicago
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
package org.globus.wsrf.samples.math.client;

import java.util.List;

import com.math.MathPortType;
import com.math.service.MathServiceAddressingLocator;

import org.globus.wsrf.client.BaseClient;
```

```
import org.globus.wsrf.utils.FaultHelper;
import org.globus.axis.util.Util;

import org.apache.commons.cli.ParseException;
import org.apache.commons.cli.CommandLine;

import javax.xml.rpc.Stub;

public class Add extends BaseClient {

    static {
        Util.registerTransport();
    }

    public static void main(String[] args) {
        Add client = new Add();
        client.setCustomUsage("value");

        int value = 0;

        try {
            CommandLine line = client.parse(args);

            List options = line.getArgList();
            if (options == null || options.isEmpty()) {
                throw new ParseException("Expected value argument");
            }
            value = Integer.parseInt((String)options.get(0));
        } catch (Exception e) {
            System.err.println("Error: " + e.getMessage());
            System.exit(COMMAND_LINE_ERROR);
        }

        MathServiceAddressingLocator locator =
            new MathServiceAddressingLocator();

        try {
            MathPortType port =
                locator.getMathPortTypePort(client.getEPR());
            client.setOptions((Stub)port);
            System.out.println(port.add(value));
        } catch(Exception e) {
            if (client.isDebugMode()) {
```

```
            FaultHelper.printStackTrace(e);
        } else {
            System.err.println("Error: " + FaultHelper.getMessage(e));
        }
        System.exit(APPLICATION_ERROR);
    }
  }

}
```

Bước 5: Thêm tác vụ 'sub' vào dịch vụ MathService

Một số file cần thay đổi: **math_bindings.wsdl, math_flatterned.wsdl, MathService.java, MathClient.java, Sub.java.**

Add 'sub' operation in file: **math_bindings.wsdl**

Search for "add" operation element, then add 'sub' operation element under the 'add' operation.

```
<wsdl:operation name="sub">
    <soap:operation soapAction="http://math.com/MathPortType/subRequest"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
```

Add some elements in file: **math_flatterned.wsdl**

Search for <wsdl:message name="AddInputMessage"> and copy the XML element, then rename to as following:

```
<wsdl:message name="SubInputMessage">
   <wsdl:part name="parameters" element="tns:add"/>
 </wsdl:message>
 <wsdl:message name="SubOutputMessage">
   <wsdl:part name="parameters" element="tns:addResponse"/>
 </wsdl:message>
```

Search for "<xsd:element name="add" type="xsd:int"/>" and add 2 lines below it:

```
<xsd:element name="sub" type="xsd:int"/>
        <xsd:element name="subResponse" type="xsd:int"/>
```

Open file 'MathService.java' and add 'sub' method into the file:

File: **MathService.java**

```
/*
* Copyright 1999-2006 University of Chicago
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
package org.globus.wsrf.samples.math;

import java.util.Set;
import java.util.Iterator;
```

```
import org.globus.wsrf.security.SecurityManager;

import java.rmi.RemoteException;

import org.apache.axis.MessageContext;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

import org.globus.wsrf.utils.log4j.SystemAdminLogFactory;

import org.globus.axis.message.addressing.EndpointReferenceType;

import org.globus.wsrf.ResourceContext;
import org.globus.wsrf.ResourceKey;
import org.globus.wsrf.utils.AddressingUtils;
import org.globus.wsrf.utils.ContextUtils;
import org.globus.wsrf.utils.log4j.LogObject;

import com.math.CreateMath;
import com.math.CreateMathResponse;

import java.security.cert.X509Certificate;

import javax.security.auth.Subject;

import org.globus.wsrf.impl.security.util.EPRUtil;

/**
* Math service implementation.
*/
public class MathService
{

   private static Log logger =
         LogFactory.getLog(MathService.class.getName());

   private static Log sysLogger =
      SystemAdminLogFactory.getLog(MathService.class.getName());

   public CreateMathResponse createMath(CreateMath request)
      throws RemoteException
   {
```

```
LogObject log = new LogObject("org.samples.math.create.start");
sysLogger.info(log);

ResourceContext ctx = null;
MathHome home = null;
ResourceKey key = null;

try
{
   ctx = ResourceContext.getResourceContext();
   home = (MathHome) ctx.getResourceHome();
   key = home.create();
}
catch(RemoteException e)
{
   sysLogger.error(new LogObject("org.samples.math.create.end",
                        "-1"));
   logger.error("Error creating math", e);
   throw e;
}
catch(Exception e)
{
   sysLogger.error(new LogObject("org.samples.math.create.end",
                        "-1"));
   throw new RemoteException("", e);
}

EndpointReferenceType epr = null;
try
{
   epr = AddressingUtils.createEndpointReference(ctx, key);
}
catch(Exception e)
{
   sysLogger.error(new LogObject("org.samples.math.create.end",
                         "-1"));

   throw new RemoteException("", e);
}

MessageContext msgCtx = MessageContext.getCurrentContext();
String servicePath = ContextUtils.getTargetServicePath(msgCtx);
```

```
        if (servicePath.equals("SecureMathService")) {
            // try to get service's credential
            SecurityManager manager = SecurityManager.getManager(msgCtx);
            Subject subject = manager.getServiceSubject();
            Set set = subject.getPublicCredentials(X509Certificate[].class);
            Iterator it = set.iterator();
            while (it.hasNext()) {
                X509Certificate[] certs = (X509Certificate[])it.next();
                EPRUtil.insertCertificates(epr, certs, null);
                break;
            }
        }

        CreateMathResponse response = new CreateMathResponse();
        response.setEndpointReference(epr);

        sysLogger.info(new LogObject("org.samples.math.create.end",
                                     "0"));

        return response;
    }

public int sub(int arg0) throws RemoteException
    {
                            sysLogger.info(new LogObject("org.samples.math.sub.start"));
                            Object resource = null;
        try
        {
            resource = ResourceContext.getResourceContext().getResource();
        }
        catch(RemoteException e)
        {
            sysLogger.info(new LogObject("org.samples.math.sub.end",
                                         "-1"));
            throw e;
        }
        catch(Exception e)
        {
            sysLogger.info(new LogObject("org.samples.math.sub.end",
                                         "-1"));

            throw new RemoteException("", e);
        }
```

```
        Math math = (Math) resource;
        synchronized(math)
        {
            int result = math.getValue();
            result -= arg0;
            math.setValue(result);
            sysLogger.info(new LogObject("org.samples.math.sub.end", "0"));
            return result;
        }
}

    public int add(int arg0) throws RemoteException
    {

        sysLogger.info(new LogObject("org.samples.math.add.start"));


        Object resource = null;
        try
        {
            resource = ResourceContext.getResourceContext().getResource();
        }
        catch(RemoteException e)
        {
            sysLogger.info(new LogObject("org.samples.math.add.end",
                                "-1"));
            throw e;
        }
        catch(Exception e)
        {
            sysLogger.info(new LogObject("org.samples.math.add.end",
                                "-1"));

            throw new RemoteException("", e);
        }
        Math math = (Math) resource;
        synchronized(math)
        {
            int result = math.getValue();
            result += arg0;
            math.setValue(result);
            sysLogger.info(new LogObject("org.samples.math.add.end", "0"));
            return result;
```

```
        }
    }
}
```

Change MathClient.java file to invoke ‘sub’ operation of MathService:

File: **MathClient.java**

```
/*
* Copyright 1999-2006 University of Chicago
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
package org.globus.wsrf.samples.math.client;
import java.util.List;
import java.util.Vector;
import java.util.Map;
import java.util.HashMap;
import java.security.cert.X509Certificate;
```

```
import javax.xml.rpc.Stub;

import org.globus.axis.message.addressing.EndpointReferenceType;

import org.oasis.wsn.FilterType;

import org.oasis.wsn.TopicExpressionType;

import org.oasis.wsn.Subscribe;

import org.oasis.wsn.Unsubscribe;

import org.oasis.wsn.NotificationMessageHolderTypeMessage;

import org.oasis.wsrf.lifetime.Destroy;

import org.oasis.wsrf.properties.ResourcePropertyValueChangeNotificationType;

import org.oasis.wsrf.properties.GetResourcePropertyResponse;

import org.globus.wsrf.encoding.ObjectSerializer;

import org.globus.wsrf.NotificationConsumerManager;

import org.globus.wsrf.NotifyCallback;

import org.globus.wsrf.WSNConstants;

import org.globus.wsrf.core.notification.ResourcePropertyValueChangeNotificationElementType;

import org.globus.wsrf.core.notification.SubscriptionManager;

import org.globus.wsrf.core.notification.service.SubscriptionManagerServiceAddressingLocator;

import org.globus.wsrf.utils.FaultHelper;

import org.globus.axis.util.Util;

import org.globus.wsrf.impl.security.authentication.Constants;

import org.globus.wsrf.security.authorization.client.Authorization;

import org.globus.wsrf.impl.security.authorization.IdentityAuthorization;

import org.globus.wsrf.impl.security.authorization.SelfAuthorization;

import org.globus.wsrf.impl.security.descriptor.GSISecureConvAuthMethod;

import org.globus.wsrf.impl.security.descriptor.GSISecureMsgAuthMethod;
```

```
import org.globus.wsrf.impl.security.descriptor.GSITransportAuthMethod;
import org.globus.wsrf.impl.security.descriptor.ResourceSecurityDescriptor;
import org.globus.wsrf.samples.math.Math;
import org.globus.wsrf.client.BaseClient;
import org.globus.security.gridmap.GridMap;
import com.math.MathPortType;
import com.math.CreateMath;
import com.math.CreateMathResponse;
import com.math.service.MathServiceAddressingLocator;
import org.globus.wsrf.container.ServiceContainer;
import org.globus.wsrf.encoding.DeserializationException;
import org.globus.wsrf.encoding.ObjectDeserializer;
import org.globus.wsrf.impl.security.util.EPRUtil;
import org.apache.axis.message.MessageElement;

public class MathClient extends BaseClient implements NotifyCallback {
   static {
      Util.registerTransport();
   }
   public static void main(String[] args) {
      MathClient client = new MathClient();
      try {
         client.parse(args);
      } catch (Exception e) {
         System.err.println("Error: " + e.getMessage());
```

```
    System.exit(COMMAND_LINE_ERROR);
}
MathServiceAddressingLocator locator =
    new MathServiceAddressingLocator();
NotificationConsumerManager consumer = null;
try {
    // Create counter resource
    MathPortType port =
        locator.getMathPortTypePort(client.getEPR());
    client.setOptions((Stub)port);
    CreateMathResponse createResponse =
        port.createMath(new CreateMath());
    // Process EPR
    EndpointReferenceType mathEPR =
        createResponse.getEndpointReference();

    // Use these certs to get identity of server.
    X509Certificate[] certs = EPRUtil.extractCertificates(mathEPR);

    // Math resource reference
    MathPortType mathPort = locator.getMathPortTypePort(mathEPR);
    client.setOptions((Stub)mathPort);

    // Create client side notification consumer
    String scheme = client.getEPR().getAddress().getScheme();
```

```
if (scheme.equals("https")) {
    Map properties = new HashMap();
    properties
        .put(ServiceContainer.CLASS,
            "org.globus.wsrf.container.GSIServiceContainer");
    consumer = NotificationConsumerManager.getInstance(properties);
} else {
    consumer = NotificationConsumerManager.getInstance();
}
consumer.startListening();


EndpointReferenceType consumerEPR = null;
Authorization authorization =
    (Authorization)((Stub)mathPort)
    ._getProperty(Constants.AUTHORIZATION);
if (authorization != null) {
    ResourceSecurityDescriptor resDesc =
        new ResourceSecurityDescriptor();
    String authz = Authorization.AUTHZ_NONE;
    if (authorization instanceof SelfAuthorization) {
        authz = Authorization.AUTHZ_SELF;
    }
    else if (authorization instanceof IdentityAuthorization) {
        GridMap gridMap = new GridMap();
        gridMap.map(
```

```
            ( (IdentityAuthorization) authorization).getIdentity(),
            "temp");
        resDesc.setDefaultGridMap(gridMap);
        authz = Authorization.AUTHZ_GRIDMAP;
    }
    resDesc.setPDP(authz);

    Vector authMethod = new Vector();
    authMethod.add(GSISecureMsgAuthMethod.BOTH);
    authMethod.add(GSISecureConvAuthMethod.BOTH);
    authMethod.add(GSITransportAuthMethod.BOTH);
    resDesc.setDefaultAuthMethods(authMethod);

    consumerEPR =
        consumer.createNotificationConsumer(client, resDesc);
} else {
    consumerEPR =
        consumer.createNotificationConsumer(client);
}

// Subscribe request
Subscribe request = new Subscribe();
request.setConsumerReference(consumerEPR);
TopicExpressionType topicExpression = new TopicExpressionType();
```

```
topicExpression.setDialect(WSNConstants.SIMPLE_TOPIC_DIALECT);
topicExpression.setValue(Math.VALUE);
MessageElement element =
   (MessageElement)ObjectSerializer
   .toSOAPElement(topicExpression, WSNConstants.TOPIC_EXPRESSION);
FilterType filter = new FilterType();
filter.set_any(new MessageElement[] { element });
request.setFilter(filter);

EndpointReferenceType subscriptionEPR =
   mathPort.subscribe(request).getSubscriptionReference();
SubscriptionManagerServiceAddressingLocator
   subscriptionManagerLocator =
   new SubscriptionManagerServiceAddressingLocator();
SubscriptionManager subscriptionPort =
   subscriptionManagerLocator.getSubscriptionManagerPort(
      subscriptionEPR);

// Add
MathPortType addPort =
   locator.getMathPortTypePort(mathEPR);
client.setOptions((Stub)addPort);

// Add 3
System.out.println("Math.add(3) is invoked: ");
```

```
addPort.add(3);
synchronized (client) {
    client.wait(1000 * 30);
    if (!client.called) {
        System.err.println("Did not receive notification in time!");
    }
}

// Get the value RP
GetResourcePropertyResponse getRPResponse =
    mathPort.getResourceProperty(Math.VALUE);
try {
    Integer value = (Integer)ObjectDeserializer.getFirstAsObject(
                                getRPResponse, Integer.class);
    System.out.println("Math has value: " + value);
} catch (DeserializationException e) {
    System.err.println("Error deserializing value: " +
                       e.getMessage());
}

// same security setting
// some line
System.out.println("Math.add(10) is invoked: ");
addPort.add(10);
```

```
    // some line
    System.out.println("Math.sub(10) is invoked: ");
     addPort.sub(10);

    // Get the value RP
getRPResponse = mathPort.getResourceProperty(Math.VALUE);
try {
   Integer value = (Integer)ObjectDeserializer.getFirstAsObject(
                              getRPResponse, Integer.class);
   System.out.println("Math has value: " + value);
} catch (DeserializationException e) {
   System.err.println("Error deserializing value: " +
                    e.getMessage());
}

// Destroy the subscription
client.setOptions((Stub)subscriptionPort);
subscriptionPort.unsubscribe(new Unsubscribe());
//subscriptionPort.destroy(new Destroy());

MathPortType destroyPort =
   locator.getMathPortTypePort(mathEPR);
client.setOptions((Stub)destroyPort);
// Destroy the counter resource
destroyPort.destroy(new Destroy());
```

```
    } catch(Exception e) {
        if (client.isDebugMode()) {
            FaultHelper.printStackTrace(e);
        } else {
            System.err.println("Error: " + FaultHelper.getMessage(e));
        }
        System.exit(APPLICATION_ERROR);
    } finally {
        if (consumer != null) {
            try { consumer.stopListening(); } catch (Exception ee) {}
        }
    }
}

boolean called = false;
// Notification callback
public void deliver(List topicPath,
                    EndpointReferenceType producer,
                    Object message) {

    NotificationMessageHolderTypeMessage notifMsg =
        (NotificationMessageHolderTypeMessage)message;
    MessageElement[] msgElem = notifMsg.get_any();
    if (msgElem != null) {
        ResourcePropertyValueChangeNotificationType changeMessage = null;
```

```
        try {
            changeMessage = (ResourcePropertyValueChangeNotificationType)ObjectDeserializer.
                toObject(msgElem[0],
                    ResourcePropertyValueChangeNotificationType.class);
        } catch (Exception e) {
            // FIXME
        }

        if(changeMessage != null) {
            try {
                Integer value = (Integer)ObjectDeserializer.getFirstAsObject(
                    changeMessage.getNewValues(), Integer.class);
                System.out.println("Got notification with value: " + value);
            } catch (DeserializationException e) {
                System.err.println("Error deserializing value: " +
                        e.getMessage());
            }
        }
        synchronized (this) {
            called = true;
            notify();
        }
    }
  }
}
```

Copy WSDL files to schema directory and copy these WSDL schemas in the Globus Toolkit's container:

cp math_*.wsdl /home/globus/gt4.2.1-all-source-installer/source-trees/wsrf/schema/core/samples/math/

cp -r /home/globus/gt4.2.1-all-source-installer/source-trees/wsrf/schema/core/samples/math /usr/grid/globus-4.2.1/share/schema/core/samples/

Now, let come to un-deploy, build and re-deploy the **MathService**:

ant undeploy

Buildfile: build.xml

undeploy:

undeployGar:

testContainerRunning:

isContainerRunning:

undeploy:

undeployClient:

[delete] Deleting /usr/grid/globus-4.2.1/etc/globus_wsrf_core_samples_math/security-config.xml

[delete] Deleting /usr/grid/globus-4.2.1/etc/globus_wsrf_core_samples_math/server-config.wsdd

[delete] Deleting /usr/grid/globus-4.2.1/etc/globus_wsrf_core_samples_math/jndi-config.xml

[delete] Deleting directory /usr/grid/globus-4.2.1/etc/globus_wsrf_core_samples_math

[delete] Deleting: /usr/grid/globus-4.2.1/lib/wsrf_core_samples_math.jar

[delete] Deleting: /usr/grid/globus-4.2.1/lib/wsrf_core_samples_math_stubs.jar

[delete] Deleting: /usr/grid/globus-4.2.1/share/schema/core/samples/math/math_bindings.wsdl

[delete] Deleting: /usr/grid/globus-4.2.1/share/schema/core/samples/math/math_flattened.wsdl

[delete] Deleting: /usr/grid/globus-4.2.1/share/schema/core/samples/math/math_service.wsdl

[delete] Deleting directory /usr/grid/globus-4.2.1/etc/globus_packages/globus_wsrf_core_samples_math

BUILD SUCCESSFUL

Total time: 0 seconds

Note that: after you run 'ant undeploy' the MathService, you have to copy the schema directory which contains WSDLs files to the Globus's container's schema again. For example:

cp -r /home/globus/gt4.2.1-all-source-installer/source-trees/wsrf/schema/core/samples/math /usr/grid/globus-4.2.1/share/schema/core/samples/

Now, we run the math-client:

globus@netlabserver01:/usr/grid/globus-4.2.1$ **myproxy-logon -s netlabserver01 -t 100**

Enter MyProxy pass phrase:

A credential has been received for user globus in /tmp/x509up_u5002.

globus@netlabserver01:/usr/grid/globus-4.2.1$ **bin/math-client -s https://172.28.13.11:18443/wsrf/services/MathService**

Math.add(3) is invoked:

Got notification with value: 3

Math has value: 3

Math.add(10) is invoked:

Math.sub(10) is invoked:

Got notification with value: 13

Got notification with value: 3

Math has value: 3

globus@netlabserver01:/usr/grid/globus-4.2.1$ **bin/math-create -s https://172.28.13.11:18443/wsrf/services/MathService -z none > /tmp/epr**

globus@netlabserver01:/usr/grid/globus-4.2.1$ **cat /tmp/epr**

<ns1:MathReference xsi:type="ns2:EndpointReferenceType" xmlns:ns1="http://math.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns2="http://www.w3.org/2005/08/addressing"><ns2:Address xsi:type="ns2:AttributedURI">https://172.28.13.11:18443/wsrf/services/MathService</ns2:Address><ns2:ReferenceParameters xsi:type="ns2:ReferenceParametersType"><ns1:MathKey>25761586</ns1:MathKey></ns2:ReferenceParameters></ns1:MathReferen

globus@netlabserver01:/usr/grid/globus-4.2.1$ **bin/math-add -e /tmp/epr 10**

10

globus@netlabserver01:/usr/grid/globus-4.2.1$ **bin/math-add -e /tmp/epr 20**

30

globus@netlabserver01:/usr/grid/globus-4.2.1$ **bin/math-add -e /tmp/epr 5**

35

globus@netlabserver01:/usr/grid/globus-4.2.1**$ bin/math-sub -e /tmp/epr 10**

Math.sub(10) is invoked .....

25

globus@netlabserver01:/usr/grid/globus-4.2.1**$ bin/math-sub -e /tmp/epr 10**

Math.sub(10) is invoked .....

15

globus@netlabserver01:/usr/grid/globus-4.2.1$ **cp bin/math-add bin/math-sub**

globus@netlabserver01:/usr/grid/globus-4.2.1$ **vi bin/math-sub**

Change: 'Add' classname to 'Sub'

globus@netlabserver01:/usr/grid/globus-4.2.1**$ bin/math-sub -e /tmp/epr 2**

Math.sub(2) is invoked .....

13

globus@netlabserver01:/usr/grid/globus-4.2.1$ **bin/math-sub -e /tmp/epr 13**

Math.sub(13) is invoked .....

0

globus@netlabserver01:/usr/grid/globus-4.2.1$ **bin/math-add -e /tmp/epr 50**

50

globus@netlabserver01:/usr/grid/globus-4.2.1**$ bin/math-sub -e /tmp/epr 50**

Math.sub(50) is invoked .....

0