

## Grid Computing

### Globus Toolkit Programming GT4 Tutorial Chapter 6 – Resource Properties

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

## Resource Properties so far

```
<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">
<!-- RESOURCE PROPERTIES -->
<xsd:element name="Value" type="xsd:int"/>
<xsd:element name="LastOp" type="xsd:string"/>
<xsd:element name="MathResourceProperties">
<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 2

## Resource Property Document

- Previously
  - In Java implementation of the resource,
    - our resource class had attributes representing each of the resource properties,
    - we used special Globus classes (ReflectionResourceProperty and ResourcePropertySet) to manage those resource properties.
  - Why are resource properties are declared in XML Schema?
    - can be implemented internally different ways
    - *must* be exchanged with other entities as XML document
  - XML representation is called the *resource property document*

Paul A. Farrell 2006 KENT STATE Grid Computing 3

## Example of Resource Property Document

At a particular time the Resource Property Document could be:

```
<MathResourceProperties
  xmlns:tns="http://www.globus.org/namespaces/examples/core/MathService_instance">
  <tns:Value>50</tns:Value>
  <tns:LastOp>ADDITION</tns:LastOp>
</MathResourceProperties>
```

Paul A. Farrell 2006 KENT STATE Grid Computing 4

## WSDL and Resource Property Document

WSDL:

```
<xsd:complexType> <xsd:sequence>
  <xsd:element ref="tns:Value" minOccurs="1" maxOccurs="unbounded"/>
  <xsd:element ref="tns:LastOp" minOccurs="1" maxOccurs="unbounded"/>
</xsd:sequence> </xsd:complexType>
```

Resource Property Document

```
<MathResourceProperties
xmlns:tns="http://www.globus.org/namespaces/examples/core/MathService_instance">
  <tns:Value>10</tns:Value>
  <tns:Value>30</tns:Value>
  <tns:Value>20</tns:Value>
  <tns:LastOp>ADDITION</tns:LastOp>
  <tns:LastOp>ADDITION</tns:LastOp>
  <tns:LastOp>SUBTRACTION</tns:LastOp>
</MathResourceProperties>
```

Paul A. Farrell 2006 KENT STATE Grid Computing 5

## Standard Interfaces

- An WSRF specification, WS-ResourceProperties, is devoted to RPs, RP documents, and a set of standard portTypes to interact with a service's RPs.
- **GetResourceProperty**
  - accesses the value of a resource property given its QName
- **GetMultipleResourceProperties**
  - accesses the value of several resource properties at once
- **SetResourceProperties**
  - Requests one or several modifications on a service's RPs.
  - Operations:
    - Update: Change the value of a RP with a new value.
    - Insert: Add a new RP with a given value.
    - Delete: Eliminate all occurrences of a certain RP.
  - The SetResourceProperties portType has a single operation (not three separate ones). It uses parameters to specify the action
- **QueryResourceProperties**
  - performs complex queries on the RP document.
    - Currently, the query language used is XPath.

Paul A. Farrell 2006 KENT STATE Grid Computing 6

## Using WS-ResourceProperty portTypes

- Example is variant of Example 1 (Chapter 3) for simplicity
- Now we *do* have a new WSDL file because we want to extend from new portTypes, which necessarily changes our service's interface
- Changes are minimal though
  - eliminate the GetValueRP operation and corresponding messages
  - extend from the four WS-ResourceProperties portTypes
    - Strictly only need the ones we use

Paul A. Farrell 2006 KENT STATE Grid Computing 7

## WSDL root element

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="MathService"
  targetNamespace="http://www.globus.org/namespaces/examples/core/MathService_instance_rp"
  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:wspw="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">
  <wsp:import
    namespace="
      http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ResourceProperties-1.2-draft-01.wsdl"
    location="../wsrf/properties/WS-ResourceProperties.wsdl" />
```

Paul A. Farrell 2006 KENT STATE Grid Computing 8

## b. Write the PortType

```
<portType name="MathPortType"
  wsdlpp:extends="wsrpw:GetResourceProperty
  wsrpw:GetMultipleResourceProperties
  wsrpw:SetResourceProperties
  wsrpw:QueryResourceProperties"
  wsrp:ResourceProperties="tns:MathResourceProperties">
  <operation name="add">
    <input message="tns:AddInputMessage"/>
    <output message="tns:AddOutputMessage"/>
  </operation>
  <operation name="subtract">
    <input message="tns:SubtractInputMessage"/>
    <output message="tns:SubtractOutputMessage"/>
  </operation>
</portType>
```

Paul A. Farrell 2006 KENT STATE Grid Computing 9

## Namespace Mappings

New namespace so need new mappings from WSDL namespaces to Java packages

```
http://www.globus.org/namespaces/examples/core/MathService_instance_rp=
  org.globus.examples.stubs.MathService_instance_rp
http://www.globus.org/namespaces/examples/core/MathService_instance_rp/bindin
  gs= org.globus.examples.stubs.MathService_instance_rp.bindings
http://www.globus.org/namespaces/examples/core/MathService_instance_rp/servic
  e= org.globus.examples.stubs.MathService_instance_rp.service
```

Paul A. Farrell 2006 KENT STATE Grid Computing 10

## Java Files

- Minimal changes
  - no longer need to implement the getValueRP operation.
  - In general, using the WS-ResourceProperties portTypes doesn't require that we add any extra code to our Java files

Paul A. Farrell 2006 KENT STATE Grid Computing 11

## Deployment Files

- There is no change to the JNDI file.
- To use the WS-ResourceProperties portTypes
  - we need to modify our WSDD file so our service uses the Globus-supplied operation providers for those portTypes

Paul A. Farrell 2006 KENT STATE Grid Computing 12

---

```
<?xml version="1.0" encoding="UTF-8"?>
<deployment name="defaultServerConfig"
  xmlns="http://xml.apache.org/axis/wsdd/"
  xmlns:java="http://xml.apache.org/axis/wsdd/providers/java"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <service name="examples/core/rp/MathService" provider="Handler"
    use="literal" style="document">
    <parameter name="className"
      value="org.globus.examples.services.core.rp.impl.MathService"/>
    <wsdlFile>share/schema/examples/MathService_instance_rp/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="GetRPPProvider GetMRPPProvider
      SetRPPProvider QueryRPPProvider"/>
    <parameter name="loadOnStartup" value="true"/>
  </service> </deployment>
```

---

Paul A. Farrell 2006 KENT STATE Grid Computing 13

## Operation Providers

---

- GT4 Core uses a design pattern called *operation providers*
  - An operation provider is a Java class, providing a set of operations, that we can easily plug into our service
  - Recall the WSDL file used previously included:  

```
<portType name="MathPortType"
  wsdlpp:extends="wsrpw:GetResourceProperty"
  wsrp:ResourceProperties="tns:MathResourceProperties">
```
  - `wsdlpp:extends` specifies that our service also implements a standard WSRF portType: the `GetResourceProperty` portType
  - We rely on the operation providers included with GT4 to *provide* an implementation of all the WSRF portTypes
- 

Paul A. Farrell 2006 KENT STATE Grid Computing 14

## Operation Providers (ctd.)

---

- To specify we wanted to use an operation provider in our service, we simply added the following to our WSDD file:  

```
<parameter name="providers" value="GetRPPProvider"/>
```
  - In future, when we want a service to provide standard functionality specified in the WSRF specs, we simply make our service extend from a standard WSRF portType and 'plug in' a Globus operation provider that implements that portType
- 

Paul A. Farrell 2006 KENT STATE Grid Computing 15

## Build and deploy

---

- As before
  - Build the service:  

```
./globus-build-service.sh rp
```
  - And deploy it:  

```
globus-deploy-gar \
  $EXAMPLES_DIR/org_globus_examples_services_core_rp.gar
```
- 

Paul A. Farrell 2006 KENT STATE Grid Computing 16

## Client Code

- We will initially design a client to
  - Get resource properties and print
  - Add 10
  - Get resource properties and print again
- Sample Output  
Value RP: 0  
LastOp RP: NONE  
Value RP: 10  
LastOp RP: ADDITION

- All the getResourceProperty code is placed inside a printResourceProperties method, so main program has:

```
printResourceProperties(math);  
math.add(10);  
printResourceProperties(math);
```

## printResourceProperties method

```
/* This method prints out MathService's resource properties by using the  
   GetResourceProperty operation. */  
private void printResourceProperties(MathPortType portType extends standard  
   GetResourceProperty portType  
   Exception {  
   Parameter is QName of the RP  
   GetResourcePropertyResponse response  
   Return is of type GetResourcePropertyResponse, a  
   String value, lastOp  
   Globus-supplied stub class  
  
   valueRP = math.getResourceProperty(portType, portType, parameter);  
   lastOpRP = math.getResourceProperty(portType, portType, parameter);  
  
   value = valueRP.get_any().get_value();  
   lastOp = lastOpRP.get_any()[0].get_value();  
   System.out.println("Value RP: " + value);  
   System.out.println("LastOp RP: " + lastOp); }
```

## Resource Properties and Resource Responses

- The GetResourcePropertyResponse object will contain zero, one, or many RPs in XML format (i.e. the same way they are represented in the Resource Property document).
- To access these RPs, we need to use the get\_any method, which returns an array of *elements* (in the XML sense of the word).
- In our case, the GetResourcePropertyResponse from requesting the Value RP will contain the following:  

```
<ns1:Value  
  xmlns:ns1="http://www.globus.org/namespaces/examples/core/MathService_instance  
  _rp">0</ns1:Value>
```
- To obtain the value 0 contained in the element, we simply need to access the first position of the array of elements (get\_value()[0]) and get its value (get\_value()).

## SetResourceProperties

- The second block of code updates the value of the Value RP using the SetResourceProperties operation and requesting an Update action.
- All the update code is placed inside a updateRP method, for example:

```
updateRP(endpoint, MathQNames.RP_VALUE, "100");  
printResourceProperties(math);
```

- Output:

Value RP: 100

LastOp RP: ADDITION

## UpdateRP

```
/* * This method updates resource property "rpQName" in the WS-  
Resource * pointed at by the endpoint reference "epr" with the new  
value "value". */  
private void updateRP(EndpointReference epr, String rpQName, String value) throws Exception {  
    WSRResourcePropertiesServiceAddressingLocator locator = new  
    WSRResourcePropertiesServiceAddressingLocator(epr);  
    SetResourceProperties_PortType port = locator.  
    .getSetResourcePropertiesPort(epr);  
    UpdateType update = new UpdateType();  
    MessageElement msg = new MessageElement(rpQName, value);  
    update.set_any(new MessageElement[] { msg });  
    SetResourceProperties_Element request = new  
    SetResourceProperties_Element();  
    request.setUpdate(update);  
    port.setResourceProperties(request);  
}
```

## UpdateRP notes

1. Obtain a reference to a generic SetResourceProperties portType
  - Done to show how to access generic WSRF operations.
  - Alternatively we could have used MathPortType.

```
WSRResourcePropertiesServiceAddressingLocator locator = new  
WSRResourcePropertiesServiceAddressingLocator();  
SetResourceProperties_PortType port = locator  
.getSetResourcePropertiesPort(epr);
```

## UpdateRP notes

2. Create an UpdateType object where we specify update to carry out. May have multiple update requests.
  2. Encapsulate each of these requests inside a MessageElement object.
  3. Then, create an array of MessageElements
  4. Include that array in UpdateType object (using the set\_any method).

```
UpdateType update = new UpdateType();  
MessageElement msg = new  
MessageElement(rpQName, value);  
update.set_any(new MessageElement[] { msg });
```

## UpdateRP notes

3. Create a `SetResourceProperties_Element` object which will represent our `SetResourceProperties` request

- Then, in our case, we add the recently created `UpdateType` object to the request using the `setUpdate` method.

```
SetResourceProperties_Element request = new  
SetResourceProperties_Element();  
request.setUpdate(update);
```

4. Finally, we invoke `SetResourceProperties`.  
`port.setResourceProperties(request);`

## GetMultipleResourceProperties

- The third, and last, block of code prints out the values of the `Value` and `LastOp` RP's using the `GetMultipleResourceProperties` operation.
- All the `GetMultipleResourceProperties` code is placed inside a `printMultipleResourceProperties` method invoked as:

```
printMultipleResourceProperties(math);
```

## printMultipleResourceProperties

```
/* This method prints out MathService's resource properties by  
using the * GetMultipleResourceProperties operation. */  
private void printMultipleResourceProperties(Math math)  
throws Exception {  
    GetMultipleResourceProperties_Element request =  
    GetMultipleResourcePropertiesResponse response;  
    QName[] resourceProperties = new QName[] {  
        MathQNames.RP_VALUE, MathQNames.RP_LASTOP };  
    request = new  
    GetMultipleResourceProperties_Element(resourceProperties);  
    response = math.getMultipleResourceProperties(request);  
    for(int i=0; i<response.get_any().length;i++) {  
        String name = response.get_any()[i].getLocalName();  
        String value = response.get_any()[i].getValue();  
        System.out.println(name +": " + value); }  
}
```

## printMultipleResourceProperties notes

1. Create a `GetMultipleResourceProperties_Element` object that represents the request to `getMultipleResourceProperties`.

- The constructor expects an array of `QNames`.
- We specify the `QNames` for the `Value` and `LastOp` RPs

```
QName[] resourceProperties = new QName[] {  
    MathQNames.RP_VALUE, MathQNames.RP_LASTOP };  
request = new  
    GetMultipleResourceProperties_Element(resourceProperties);
```

## printMultipleResourceProperties notes

---

2. Invoke the `getMultipleResourceProperties`.
  - Notice how the return value is of type `GetMultipleResourcePropertiesResponse`.

```
response = math.getMultipleResourceProperties(request);
```

## printMultipleResourceProperties notes

---

3. Return of `getMultipleResourceProperties` encapsulates 0, 1, or many RPs in XML format.
  - In this case, the `GetMultipleResourcePropertiesResponse` will contain the following:
  - `<ns1:Value xmlns:ns1="http://www.globus.org/namespaces/examples/core/MathService_instance_rp">100</ns1:Value>`
  - `<ns2:LastOp xmlns:ns2="http://www.globus.org/namespaces/examples/core/MathService_instance_rp">ADDITION</ns2:LastOp>`

## printMultipleResourceProperties notes

---

- To extract the value of the RPs, we once again rely on the `get_any` method, which returns an array of elements.
  - We iterate through this array, and write the value of each element using the `getValue` method.
  - Here we are also printing out the name of the property using the `getLocalName` method.

```
for(int i=0; i<response.get_any().length;i++) {  
    String name = response.get_any()[i].getLocalName();  
    String value = response.get_any()[i].getValue();  
    System.out.println(name +": " + value); }
```