# *Destroying the WSRF Resource (Scheduled Destruction)*

***Version: 1.0 Date 13/07/05***

This tutorial is the continuation of fifth tutorial "Destroying WSRF Resource (Immediate Destruction)" which can be found at http://tyne.dl.ac.uk/WOSE/Tutorial5.pdf. In that tutorial when we were calling Destroy() operation, the resource was immediately destroyed. In this tutorial we will have scheduled destruction. Scheduled destruction can be used as Lease Time, and at any time you can change the destruction time, extending or cancelling the Lease. The concept between this Scheduled destruction is adding two new resource properties, one for currentTime and other for terminationTime. Globus implementation will keep the checking the terminationTime with respect to currentTime and will destroy the resource appropriate time. At anytime before termination you can change the terminationTime, thus extending or cancelling the lease.

Now we have the following classes

1. HelloQNames1.java; it is same interface with QNames, only Namespace has changed.
2. HelloWorldFactoryService7.java; this is new class with createResource(), which is responsible to create instance of the resource with unique ID and sending the URL of HelloWorldService7 to the client.
3. HelloWorldResourceFactory7.java; represents the resources for our WSRF HelloWorld.
4. HelloWorldResourceHomeFactory7.java; this is home for our resource and resource has to be initialised through this class.
5. HelloWorldService7.java; this is the class which has business methods for our main service, in last tutorials it was HelloWorldService.java.

## Destroying the WSRF with multiple instances

We will start with same WSRF Service, which, you have already seen in the last tutorials and discussing the changes which you need to made in the tutorials. Our simple WSRF *Service*, has already two business methods, echo() and getNameRP(), and one compulsory method GetResourceProperty() and then we had added new business method called create() (in Tutorial 3, but in Tutorial 4 it is createResource()), both in the WSDL file and in java implementation. Client can't interact with any business method of our web service unless first he calls the create()/createResource(), in return operation will create new instance of resource for "WSRF HelloWorld" and will return the unique identifier to the client. Client will use this unique identifier for further interaction with the web service. In Tutorial 5 we added a new method Destroy, which will let the client to destroy the instance of the resource immediately, now for scheduled destruction we have to add new method "SetTerminationTime"; luckily Globus implementation provides the implementation of this operation just like Destroy() operation, so we don't need to implement it but we have to add this method in the WSDL file for HelloWorldService.

## *Modifying WSDL File:*

Just like previous tutorials we are starting with our WSDL file; we have two WSDL files one for HelloWorldService and one for HelloFactoryService. WSDL file for HelloFactoryService is "factory7_port_type.wsdl" and is exactly the same as last tutorial; the only change is the change in the target name space. There are in total 4 changes highlighted by bold blue font.

> These WSDL files are in the directory "schema\tutorial" relative to tutorial "src" directory with the name "helloworld7_port_type.wsdl" and "factory7_port_type.wsdl.

We have to change the WSDL file for HelloWorldService which is helloworld7_port_type.wsdl, other than changing the target name space; we have to add the SetTerminationTime() operation. We are using the operation provided by the Globus implementation, therefore we don't have to add any data type or input/output messages, just have to add the operation in the element portType .

```
<operation name="SetTerminationTime">
  <input message="wsrlw:SetTerminationTimeRequest"
    wsa:Action="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
        ResourceLifetime/SetTerminationTime"/>
  <output message="wsrlw:SetTerminationTimeResponse"
    wsa:Action="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
        ResourceLifetime/SetTerminationTimeResponse"/>
```

```
    <fault message="wsrlw:UnableToSetTerminationTimeFault"
           name="UnableToSetTerminationTimeFault"/>
    <fault message="wsrlw:ResourceUnknownFault"
           name="ResourceUnknownFault"/>
    <fault message="wsrlw:TerminationTimeChangeRejectedFault"
           name="TerminationTimeChangeRejectedFault"/>
  </operation>
```

The complete WSDL file helloworld7_port_type.wsdl is shown below with changes in bold blue font which is similar to helloworld5_port_type.wsdl except the addition of SetTerminationTime Operation:

```
<definitions name="HelloWorld7"
   targetNamespace="http://tutorial7.wsrf.dl.ac.uk/helloworld"
   xmlns:tns="http://tutorial7.wsrf.dl.ac.uk/helloworld"
   xmlns:wsrp="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd"
   xmlns:wsrlw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.wsdl"
   xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor"
   xmlns:gtwsdl1="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.wsdl"
   xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing"
   xmlns:wsntw="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl"
   xmlns:wsrpw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl"
   xmlns="http://schemas.xmlsoap.org/wsdl/">
  <import namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-BaseFaults-1.2-draft-01.wsdl"
   location="../wsrf/faults/WS-BaseFaults.wsdl"/>
  <import namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.wsdl"
   location="../wsrf/lifetime/WS-ResourceLifetime.wsdl"/>
  <import namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl"
   location="../wsrf/properties/WS-ResourceProperties.wsdl"/>
  <import namespace="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl"
   location="../wsrf/notification/WS-BaseN.wsdl"/>
  <import namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.wsdl"
   location="../wsrf/servicegroup/WS-ServiceGroup.wsdl"/>
  <import namespace="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl"
   location="../wsrf/notification/WS-BaseN.wsdl" />

  <types>
   <schema
     targetNamespace="http://tutorial7.wsrf.dl.ac.uk/helloworld"
     xmlns:tns="http://tutorial7.wsrf.dl.ac.uk/helloworld"
     xmlns="http://www.w3.org/2001/XMLSchema"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <import namespace="http://schemas.xmlsoap.org/ws/2004/03/addressing"
     schemaLocation="../ws/addressing/WS-Addressing.xsd"/>
   <import  namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ServiceGroup-1.2-draft-01.xsd"
     schemaLocation="../wsrf/servicegroup/WS-ServiceGroup.xsd"/>

    <element name="name" type="xsd:string"/>

    <element name="getNameRPRequest" >
     <complexType/>
    </element>
    <element name="getNameRPResponse" type="xsd:string"/>

    <element name="HelloWorldResourcePropertiesSet">
     <complexType>
      <sequence>
       <element ref="tns:name"/>
```

```
      </sequence>
     </complexType>
    </element>

    <element name="echoRequest" type="xsd:string" />
    <element name="echoResponse" type="xsd:string" />
  </schema>
</types>

<message name="EchoRequest">
  <part name="EchoRequest"  element="tns:echoRequest" />
</message>
<message name="EchoResponse">
 <part name="EchoResponse" element="tns:echoResponse" />
</message>

<message name="GetNameRPRequest">
  <part name="GetNameRPRequest" element="tns: getNameRPRequest " />
</message>
<message name="GetNameRPResponse">
  <part name="GetNameRPResponse" element="tns:getNameRPResponse" />
</message>

<portType name="HelloWorldPortType"
        wsrp:ResourceProperties="HelloWorldResourcePropertiesSet">

  <operation name="GetResourceProperty">
    <input name="GetResourcePropertyRequest"
        message="wsrpw:GetResourcePropertyRequest"
        wsa:Action="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties/GetResourceProperty"/>
   <output name="GetResourcePropertyResponse"
         message="wsrpw:GetResourcePropertyResponse"
         wsa:Action="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties/GetResourcePropertyResponse"/>
   <fault  name="InvalidResourcePropertyQNameFault"
          message="wsrpw:InvalidResourcePropertyQNameFault"/>
    <fault name="ResourceUnknownFault" message="wsrpw:ResourceUnknownFault"/>
  </operation>

 <operation name="Destroy">
    <input message="wsrlw:DestroyRequest"
         wsa:Action="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime/Destroy"/>
    <output message="wsrlw:DestroyResponse"
         wsa:Action="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime/DestroyResponse"/>
    <fault  message="wsrlw:ResourceNotDestroyedFault" name="ResourceNotDestroyedFault"/>
    <fault message="wsrlw:ResourceUnknownFault"   name="ResourceUnknownFault"/>
  </operation>

  <operation name="SetTerminationTime">
     <input message="wsrlw:SetTerminationTimeRequest"
        wsa:Action="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime/SetTerminationTime"/>
    <output message="wsrlw:SetTerminationTimeResponse"
       wsa:Action="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime/SetTerminationTimeResponse"/>
    <fault message="wsrlw:UnableToSetTerminationTimeFault" name="UnableToSetTerminationTimeFault"/>
   <fault message="wsrlw:ResourceUnknownFault"  name="ResourceUnknownFault"/>
   <fault message="wsrlw:TerminationTimeChangeRejectedFault"
      name="TerminationTimeChangeRejectedFault"/>
```

```
</operation>

   <!--  name in input and output is optional-->
   <operation name="echo">
    <input  name="EchoRequest"  message="tns:EchoRequest" />
    <output name="EchoResponse" message="tns:EchoResponse" />
   </operation>

   <operation name="getNameRP">
    <input  name="GetNameRPRequest"  message="tns:GetNameRPRequest" />
    <output name="GetNameRPResponse" message="tns:GetNameRPResponse" />
   </operation>

  </portType>
</definitions>
```

### *WSDL for Factory Service*

Below is the WSDL file for our Factory Service, which is has only one operation createResource() to create the instance of our main service HelloWorldServie. This WSDL file is in factory7_port_type.wsdl and is exactly the same as factory5_port_type.wsdl, except target name space difference.

```
  <?xml version="1.0" encoding="UTF-8"?>
<definitions name="HelloFactoryService7"
   targetNamespace="http://tutorial7.wsrf.dl.ac.uk/helloworld"
   xmlns:tns="http://tutorial7.wsrf.dl.ac.uk/helloworld"
   xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing"
   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns="http://schemas.xmlsoap.org/wsdl/">

<types>
<xsd:schema targetNamespace="http://tutorial7.wsrf.dl.ac.uk/helloworld"
   xmlns:tns="http://tutorial7.wsrf.dl.ac.uk/helloworld"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">

     <import namespace="http://schemas.xmlsoap.org/ws/2004/03/addressing"
      schemaLocation="../ws/addressing/WS-Addressing.xsd"/>

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

</xsd:schema>
</types>

<message name="CreateResourceRequest">
        <part name="request" element="tns:createResource"/>
</message>
```

```
    <message name="CreateResourceResponse">
        <part name="response" element="tns:createResourceResponse"/>
    </message>

    <portType name="HelloWorldFactoryPortType">
        <operation name="createResource">
            <input message="tns:CreateResourceRequest"/>
            <output message="tns:CreateResourceResponse"/>
        </operation>
    </portType>

</definitions>
```

**Modified Implementation of Web Service:**

This implementation is using the similar interface which we used in last example and changed the NS variable from *http://tutorial6.wsrf.dl.ac.uk/helloworld* to *http://tutorial7.wsrf.dl.ac.uk/helloworld*.

```
public static final String NS = "http://tutorial7.wsrf.dl.ac.uk/helloworld";
```

```
package uk.ac.dl.ws.service;

import javax.xml.namespace.QName;

public interface HelloQNames1 {
   public static final String NS = "http://tutorial7.wsrf.dl.ac.uk/helloworld";
   public static final QName RP_NAME = new QName(NS, "name");
  public static final QName RP_LAST_MODIFIED = new QName(NS,"lastModified");
   public static final QName RESOURCE_PROPERTIES = new QName(NS, "HelloWorldResourcePropertiesSet");
}
```

There are not many changes in most of the implementation classes and they are exactly the same as last tutorial, except the changes in the name, lack of blue font in the table below indicates the extensive similarities in classes. It is only *HelloWorldResourceFactory7* which requires few changes, which are related to adding new resource property for current time and termination time although we don't have these properties in our WSDL file for HelloWorld Service. I have outlined the changes below:

1. Import few new classes related to Life Time of Resource
   ```
   import org.globus.wsrf.ResourceLifetime;
   import org.globus.wsrf.impl.ReflectionResourceProperty;
   import org.globus.wsrf.impl.SimpleResourcePropertyMetaData;
   ```
2. *HelloWorldResourceFactory7* also implements interface *ResourceLifeTime*
3. Add new variable terminationTime of type Calendar
   ```
   private Calendar terminationTime
   ```
4. Create two new ResourceProperty representing currentTime and terminationTime
   ```
   //new resource property which is used for currentTime and terminationTime
   ResourceProperty prop;

   // this property exposes the termination time
    prop = new ReflectionResourceProperty(
   SimpleResourcePropertyMetaData.TERMINATION_TIME, this);
   this.propSet.add(prop);

   // this property exposes current time, as believe by local system
   prop = new ReflectionResourceProperty(
   SimpleResourcePropertyMetaData.CURRENT_TIME, this);
   ```

```
this.propSet.add(prop);
```

5. Add set/get methods for terminationTime

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

6. Add get method for currentTime

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

Tutorial 6

```
package uk.ac.dl.ws.service;

import org.globus.wsrf.Resource;
import org.globus.wsrf.ResourceProperties;
import org.globus.wsrf.ResourceProperty;
import org.globus.wsrf.ResourcePropertySet;
import org.globus.wsrf.ResourceIdentifier;
import org.globus.wsrf.impl.SimpleResourceProperty;
import org.globus.wsrf.impl.SimpleResourcePropertySet;
import javax.xml.namespace.QName;
import java.util.Calendar;

public class HelloWorldResourceFactory6  implements
        Resource, ResourceProperties, ResourceIdentifier {

 /** the identifier of the Resource */
 private Object id;

 /** Stores ResourceProperties for HelloWorldService */
 private ResourcePropertySet propSet;

 /* Variable for Resource property */
 private String name;

 /** Resource property "name". */
 private ResourceProperty nameRP;
 private ResourceProperty lastModifiedRP;

 /** initializes the HelloWorld. */
 public void initialize() throws Exception {
   // choose an ID
   this.id = new Integer(hashCode());

   // create the resource property set
   this.propSet = new SimpleResourcePropertySet
        (HelloQNames1.RESOURCE_PROPERTIES);

   // create resource properties
   this.nameRP = new
       SimpleResourceProperty(HelloQNames1.RP_NAME);
   this.lastModifiedRP = new SimpleResourceProperty(new
     QName("http://tutorial6.wsrf.dl.ac.uk/helloworld","lastModified"));
   //this.lastModifiedRP = new SimpleResourceProperty(
       HelloQNames1.LAST_MODIFIED_RP);

   this.propSet.add(this.nameRP);
   setName("Asif Akram from Tutorial 6");
   this.nameRP.add(name);

   this.propSet.add(this.lastModifiedRP);
```

Tutorial 6

```
package uk.ac.dl.ws.service;

import org.globus.wsrf.Resource;
import org.globus.wsrf.ResourceProperties;
import org.globus.wsrf.ResourceProperty;
import org.globus.wsrf.ResourcePropertySet;
import org.globus.wsrf.ResourceLifetime;
import org.globus.wsrf.ResourceIdentifier;
import org.globus.wsrf.impl.ReflectionResourceProperty;
import org.globus.wsrf.impl.SimpleResourcePropertyMetaData;
import org.globus.wsrf.impl.SimpleResourceProperty;
import org.globus.wsrf.impl.SimpleResourcePropertySet;
import javax.xml.namespace.QName;
import java.util.Calendar;

public class HelloWorldResourceFactory7  implements Resource,
    ResourceProperties, ResourceIdentifier, ResourceLifetime {

 /** the identifier of this sticky note */
 private Object id;

 /** Stores the ResourceProperties for HelloWorldService */
 private ResourcePropertySet propSet;

 /* Variable for Resource property */
 private String name;

 private Calendar terminationTime;

 /** Resource property "name". */
 private ResourceProperty nameRP;
 private ResourceProperty lastModifiedRP;

 /** initializes the HelloWorld. */
 public void initialize() throws Exception {
   // choose an ID
   this.id = new Integer(hashCode());

   // create the resource property set
   this.propSet = new SimpleResourcePropertySet(HelloQNames1.
                                RESOURCE_PROPERTIES);

   // create resource properties
   this.nameRP = new
       SimpleResourceProperty(HelloQNames1.RP_NAME);
   this.lastModifiedRP = new SimpleResourceProperty(new
     QName("http://tutorial7.wsrf.dl.ac.uk/helloworld", "lastModified"));
   //this.lastModifiedRP = new SimpleResourceProperty(
       HelloQNames1.LAST_MODIFIED_RP);
```

```
    this.lastModifiedRP.add(Calendar.getInstance());
  }

  public ResourcePropertySet getResourcePropertySet() {
    return propSet;
  }

  public void setNameRP(String nameValue) {
    setName(nameValue);
    this.nameRP.set(0, name);
    this.lastModifiedRP.set(0,Calendar.getInstance());
  }

  public String getName() {
    return name;
  }

  public void setName(String name) {
    this.name = name;
  }

  public Object getID() {
    return id;
  }

}
```

```
    this.propSet.add(this.nameRP);
    setName("Asif Akram from Tutorial 7");
    this.nameRP.add(name);

    this.propSet.add(this.lastModifiedRP);
    this.lastModifiedRP.add(Calendar.getInstance());

    // these are the RPs necessary for resource lifetime management
    ResourceProperty prop;

    // this property exposes the termination time
    prop = new ReflectionResourceProperty(
        SimpleResourcePropertyMetaData.TERMINATION_TIME, this);
    this.propSet.add(prop);

    // this property exposes current time, as believe by local system
    prop = new ReflectionResourceProperty(
        SimpleResourcePropertyMetaData.CURRENT_TIME, this);
    this.propSet.add(prop);
  }
  public ResourcePropertySet getResourcePropertySet() {
    return propSet;
  }
  public void setNameRP(String nameValue) {
    setName(nameValue);
    this.nameRP.set(0, name);
    this.lastModifiedRP.set(0, Calendar.getInstance());
  }
  public String getName() {
    return name;
  }
  public void setName(String name) {
    this.name = name;
  }
  public Object getID() {
    return id;
  }
  public void setTerminationTime(Calendar time) {
      this.terminationTime = time;
    }
    public Calendar getTerminationTime() {
      return this.terminationTime;
    }
    public Calendar getCurrentTime() {
      return Calendar.getInstance();
    }
}
```

```
    package uk.ac.dl.ws.service;

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

    public class HelloWorldResourceHomeFactory6 extends
                                        ResourceHomeImpl{

      public ResourceKey create() {
        try {
          HelloWorldResourceFactory6 hello =
              (HelloWorldResourceFactory6) createNewInstance();
          hello.initialize();
          ResourceKey key = new
              SimpleResourceKey(keyTypeName, hello.getID());
          this.add(key, hello);
          return key;
        }
        catch (Exception e) {
```

```
    package uk.ac.dl.ws.service;

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

    public class HelloWorldResourceHomeFactory7 extends
                                        ResourceHomeImpl{

      public ResourceKey create() {
        try {
          HelloWorldResourceFactory7 hello =
              (HelloWorldResourceFactory7) createNewInstance();
          hello.initialize();
          ResourceKey key = new
              SimpleResourceKey(keyTypeName, hello.getID());
          this.add(key, hello);
          return key;
        }
        catch (Exception e) {
```

```
            System.out.println("Exception when creating HelloWorld: ");
            e.printStackTrace();
            return null;
          }
        }
      }
```

```
            System.out.println("Exception when creating HelloWorld: ");
            e.printStackTrace();
            return null;
          }
        }
      }
```

```
package uk.ac.dl.ws.service;

import java.rmi.RemoteException;
import org.apache.axis.message.addressing.EndpointReferenceType;
import org.globus.wsrf.ResourceContext;
import org.globus.wsrf.ResourceKey;
import org.globus.wsrf.utils.AddressingUtils;
import uk.ac.dl.wsrf.tutorial6.helloworld.*;

public class HelloWorldService6 {

  public HelloWorldService6() throws RemoteException { }

  public String echo(String name) throws RemoteException {
    HelloWorldResourceFactory6 helloWorldResource = getResource();
    helloWorldResource.setNameRP(name);
    return "Hello " + name + " !";
  }
  public String getNameRP(GetNameRPRequest params) throws
                                                  RemoteException {
    HelloWorldResourceFactory6 helloWorldResource = getResource();
    return helloWorldResource.getName();
  }
  public HelloWorldResourceFactory6 getResource() throws
                                                  RemoteException {
    Object resource = null;
    try {
      resource =
            ResourceContext.getResourceContext().getResource();
    } catch (Exception e) {
      throw new RemoteException("", e);
    }
    HelloWorldResourceFactory6 helloWorldResource =
          (HelloWorldResourceFactory6) resource;
    return helloWorldResource;
  }
}
```

```
package uk.ac.dl.ws.service;

import java.rmi.RemoteException;
import org.apache.axis.message.addressing.EndpointReferenceType;
import org.globus.wsrf.ResourceContext;
import org.globus.wsrf.ResourceKey;
import org.globus.wsrf.utils.AddressingUtils;
import uk.ac.dl.wsrf.tutorial7.helloworld.*;

public class HelloWorldService7 {

  public HelloWorldService7() throws RemoteException { }

  public String echo(String name) throws RemoteException {
    HelloWorldResourceFactory7 helloWorldResource = getResource();
    helloWorldResource.setNameRP(name);
    return "Hello " + name + " !";
  }
  public String getNameRP(GetNameRPRequest params) throws
                                                  RemoteException {
    HelloWorldResourceFactory7 helloWorldResource = getResource();
    return helloWorldResource.getName();
  }
  public HelloWorldResourceFactory7 getResource() throws
                                                  RemoteException {
    Object resource = null;
    try {
      resource =
            ResourceContext.getResourceContext().getResource();
    } catch (Exception e) {
      throw new RemoteException("", e);
    }
    HelloWorldResourceFactory7 helloWorldResource =
          (HelloWorldResourceFactory7) resource;
    return helloWorldResource;
  }
}
```

```
package uk.ac.dl.ws.service;

import java.rmi.RemoteException;
import java.net.URL;

import org.globus.wsrf.ResourceContext;
import org.globus.wsrf.ResourceKey;
import org.apache.axis.message.addressing.EndpointReferenceType;
import org.apache.axis.MessageContext;
import org.globus.wsrf.utils.AddressingUtils;
import org.globus.wsrf.container.ServiceHost;
import uk.ac.dl.wsrf.tutorial6.helloworld.*;

public class HelloWorldFactoryService6{

  /* Implementation of createResource Operation */
  public CreateResourceResponse createResource(CreateResource
                                  request) throws  RemoteException {
    ResourceContext ctx = null;
    HelloWorldResourceHomeFactory6 home = null;
    ResourceKey key = null;

    /* First, we create a new HelloWorldResourceFactory through
            the HelloWorldResourceHomeFactory */
```

```
package uk.ac.dl.ws.service;

import java.rmi.RemoteException;
import java.net.URL;

import org.globus.wsrf.ResourceContext;
import org.globus.wsrf.ResourceKey;
import org.apache.axis.message.addressing.EndpointReferenceType;
import org.apache.axis.MessageContext;
import org.globus.wsrf.utils.AddressingUtils;
import org.globus.wsrf.container.ServiceHost;
import uk.ac.dl.wsrf.tutorial7.helloworld.*;

public class HelloWorldFactoryService7{

  /* Implementation of createResource Operation */
  public CreateResourceResponse createResource(CreateResource
                                  request) throws RemoteException {
    ResourceContext ctx = null;
    HelloWorldResourceHomeFactory7 home = null;
    ResourceKey key = null;

    /* First, we create a new HelloWorldResourceFactory through
            the HelloWorldResourceHomeFactory */
```

```
    try {
      ctx = ResourceContext.getResourceContext();
      home = (HelloWorldResourceHomeFactory6)
                                    ctx.getResourceHome();
      key = home.create();
    }
    catch (Exception e) {
      throw new RemoteException("", e);
    }
    EndpointReferenceType epr = null;

    /* We construct the instance's endpoint reference. The instance's
      service path can be found in the WSDD file as a parameter. */
    try {
      URL baseURL = ServiceHost.getBaseURL();
      String instanceService = (String) MessageContext
          .getCurrentContext().getService().getOption("instance");
      String instanceURI = baseURL.toString() + instanceService;
      // The endpoint reference includes instance's URI and resource key
      epr = AddressingUtils.createEndpointReference(instanceURI, key);
    }
    catch (Exception e) {
      throw new RemoteException("", e);
    }

    /* Finally, return endpoint reference in a CreateResourceResponse */
    CreateResourceResponse response = new
                        CreateResourceResponse();
    response.setEndpointReference(epr);
    return response;
  }
}
```

```
    try {
      ctx = ResourceContext.getResourceContext();
      home = (HelloWorldResourceHomeFactory7)
                                    ctx.getResourceHome();
      key = home.create();
    }
    catch (Exception e) {
      throw new RemoteException("", e);
    }
    EndpointReferenceType epr = null;

    /* We construct the instance's endpoint reference. The instance's
service path can be found in the WSDD file as a parameter. */
    try {
      URL baseURL = ServiceHost.getBaseURL();
      String instanceService = (String) MessageContext
          .getCurrentContext().getService().getOption("instance");
      String instanceURI = baseURL.toString() + instanceService;
      // The endpoint reference includes instance's URI and resource key
      epr = AddressingUtils.createEndpointReference(instanceURI, key);
    }
    catch (Exception e) {
      throw new RemoteException("", e);
    }

    /* Finally, return endpoint reference in a CreateResourceResponse */
    CreateResourceResponse response = new
CreateResourceResponse();
    response.setEndpointReference(epr);
    return response;
  }
}
```

**Modified Deployment Descriptor:**

Deployment Descriptor is also nearly the same as last tutorial, nothing to change except name of Services, WSDL files and name of classes implementing Services. The only significance change is adding "SetTerminationTimeProvider" as one of the value for "providers".

```
<deployment name="defaultServerConfig" xmlns="http://xml.apache.org/axis/wsdd/"
xmlns:aggr="http://mds.globus.org/aggregator/types"
xmlns:java="http://xml.apache.org/axis/wsdd/providers/java"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <service name="HelloWorldService7" provider="Handler" use="literal" style="document">
    <parameter name="providers" value="GetRPProvider DestroyProvider QueryRPProvider
                                                    SetTerminationTimeProvider"/>
    <parameter name="handlerClass" value="org.globus.axis.providers.RPCProvider"/>
    <parameter name="scope" value="Application"/>
    <parameter name="allowedMethods" value="*"/>
    <parameter name="activateOnStartup" value="true"/>
    <parameter name="className" value="uk.ac.dl.ws.service.HelloWorldService7"/>
    <wsdlFile>share/schema/tutorial/HelloWorld7_service.wsdl</wsdlFile>
  </service>

  <! -- Factory service -->
    <service name="HelloFactoryService7" provider="Handler" use="literal" style="document">
        <parameter name="handlerClass" value="org.globus.axis.providers.RPCProvider"/>
       <parameter name="scope" value="Application"/>
       <parameter name="allowedMethods" value="*"/>
       <parameter name="instance" value="HelloWorldService7"/>
```

```
    <parameter name="className" value="uk.ac.dl.ws.service.HelloWorldFactoryService7"/>
    <wsdlFile>share/schema/tutorial/FactoryService7_service.wsdl</wsdlFile>
  </service>

</deployment>
```

| Tutorial 7 | Tutorial 6 |
|---|---|
| `<deployment name="defaultServerConfig"`<br>`xmlns="http://xml.apache.org/axis/wsdd/"`<br>`xmlns:aggr="http://mds.globus.org/aggregator/types"`<br>`xmlns:java="http://xml.apache.org/axis/wsdd/providers/java"`<br>`xmlns:xsd="http://www.w3.org/2001/XMLSchema">`<br><br>`<service name="HelloWorldService7" provider="Handler"`<br>`use="literal" style="document">`<br>`<parameter name="providers" value="GetRPProvider`<br>`DestroyProvider QueryRPProvider SetTerminationTimeProvider"/>`<br>`<parameter name="handlerClass"`<br>`value="org.globus.axis.providers.RPCProvider"/>`<br>`<parameter name="scope" value="Application"/>`<br>`<parameter name="allowedMethods" value="*"/>`<br>`<parameter name="activateOnStartup" value="true"/>`<br>`<parameter name="className"`<br>`value="uk.ac.dl.ws.service.HelloWorldService7"/>`<br>`<wsdlFile>`<br>`share/schema/tutorial/HelloWorld7_service.wsdl`<br>`</wsdlFile>`<br>`</service>`<br><br>`<!-- Factory service -->`<br>`<service name="HelloFactoryService7" provider="Handler"`<br>`use="literal" style="document">`<br>`<parameter name="handlerClass"`<br>`value="org.globus.axis.providers.RPCProvider"/>`<br>`<parameter name="scope" value="Application"/>`<br>`<parameter name="allowedMethods" value="*"/>`<br>`<parameter name="instance" value="HelloWorldService7"/>`<br>`<parameter name="className"`<br>`value="uk.ac.dl.ws.service.HelloWorldFactoryService7"/>`<br>`<wsdlFile>`<br>`share/schema/tutorial/FactoryService7_service.wsdl`<br>`</wsdlFile>`<br>`</service>`<br><br>`</deployment>` | `<deployment name="defaultServerConfig"`<br>`xmlns="http://xml.apache.org/axis/wsdd/"`<br>`xmlns:aggr="http://mds.globus.org/aggregator/types"`<br>`xmlns:java="http://xml.apache.org/axis/wsdd/providers/java"`<br>`xmlns:xsd="http://www.w3.org/2001/XMLSchema">`<br><br>`<service name="HelloWorldService6" provider="Handler"`<br>`use="literal" style="document">`<br>`<parameter name="providers" value="GetRPProvider`<br>`DestroyProvider QueryRPProvider "/>`<br>`<parameter name="handlerClass"`<br>`value="org.globus.axis.providers.RPCProvider"/>`<br>`<parameter name="scope" value="Application"/>`<br>`<parameter name="allowedMethods" value="*"/>`<br>`<parameter name="activateOnStartup" value="true"/>`<br>`<parameter name="className"`<br>`value="uk.ac.dl.ws.service.HelloWorldService6"/>`<br>`<wsdlFile>`<br>`share/schema/tutorial/HelloWorld6_service.wsdl`<br>`</wsdlFile>`<br>`</service>`<br><br>`<!-- Factory service -->`<br>`<service name="HelloFactoryService6" provider="Handler"`<br>`use="literal" style="document">`<br>`<parameter name="handlerClass"`<br>`value="org.globus.axis.providers.RPCProvider"/>`<br>`<parameter name="scope" value="Application"/>`<br>`<parameter name="allowedMethods" value="*"/>`<br>`<parameter name="instance" value="HelloWorldService6"/>`<br>`<parameter name="className"`<br>`value="uk.ac.dl.ws.service.HelloWorldFactoryService6"/>`<br>`<wsdlFile>`<br>`share/schema/tutorial/FactoryService6_service.wsdl`<br>`</wsdlFile>`<br>`</service>`<br><br>`</deployment>` |

**Modified Deployment "jndi-config":**

The "deploy-jndi-config.xml" Deployment Descriptor is also nearly the same as last tutorial, nothing to change except target name space and Service name.

```
<jndiConfig xmlns="http://wsrf.globus.org/jndi/config">

  <service name="HelloWorldService7">
    <resource
      name="home" type="uk.ac.dl.ws.service.HelloWorldResourceHomeFactory7">
      <resourceParams>
        <parameter>
          <name>factory</name>
          <value>org.globus.wsrf.jndi.BeanFactory</value>
        </parameter>
        <parameter>
```

```
            <name>resourceClass</name>
            <value>uk.ac.dl.ws.service.HelloWorldResourceFactory7</value>
          </parameter>
          <parameter>
            <name>resourceKeyName</name>
            <value>{http://tutorial7.wsrf.dl.ac.uk/helloworld}NameKeyFactory7</value>
          </parameter>
          <parameter>
            <name>resourceKeyType</name>
            <value>java.lang.Integer</value>
          </parameter>
        </resourceParams>
      </resource>
    </service>

    <!-- Factory service -->
    <service name="HelloFactoryService7">
      <resourceLink name="home" target="java:comp/env/services/HelloWorldService7/home"/>
  </service>

</jndiConfig>
```

| Tutorial 7 | Tutorial 6 |
|---|---|
| `<jndiConfig xmlns="http://wsrf.globus.org/jndi/config">`<br><br>`<service name="HelloWorldService7">`<br>`<resource name="home"`<br>`type="uk.ac.dl.ws.service.HelloWorldResourceHomeFactory7">`<br>`<resourceParams>`<br>`<parameter>`<br>`<name>factory</name>`<br>`<value>org.globus.wsrf.jndi.BeanFactory</value>`<br>`</parameter>`<br>`<parameter>`<br>`<name>resourceClass</name>`<br>`<value>`<br>`uk.ac.dl.ws.service.HelloWorldResourceFactory7`<br>`</value>`<br>`</parameter>`<br>`<parameter>`<br>`<name>resourceKeyName</name>`<br>`<value>`<br>`{http://tutorial7.wsrf.dl.ac.uk/helloworld}NameKeyFactory7`<br>`</value>`<br>`</parameter>`<br>`<parameter>`<br>`<name>resourceKeyType</name>`<br>`<value>java.lang.Integer</value>`<br>`</parameter>`<br>`</resourceParams>`<br>`</resource>`<br>`</service>`<br><br>`<!-- Factory service -->`<br>`<service name="HelloFactoryService7">`<br>`<resourceLink name="home"`<br>`target="java:comp/env/services/HelloWorldService7/home"/>`<br>`</service>`<br><br>`</jndiConfig>` | `<jndiConfig xmlns="http://wsrf.globus.org/jndi/config">`<br><br>`<service name="HelloWorldService6">`<br>`<resource name="home"`<br>`type="uk.ac.dl.ws.service.HelloWorldResourceHomeFactory6">`<br>`<resourceParams>`<br>`<parameter>`<br>`<name>factory</name>`<br>`<value>org.globus.wsrf.jndi.BeanFactory</value>`<br>`</parameter>`<br>`<parameter>`<br>`<name>resourceClass</name>`<br>`<value>`<br>`uk.ac.dl.ws.service.HelloWorldResourceFactory6`<br>`</value>`<br>`</parameter>`<br>`<parameter>`<br>`<name>resourceKeyName</name>`<br>`<value>`<br>`{http://tutorial6.wsrf.dl.ac.uk/helloworld}NameKeyFactory6`<br>`</value>`<br>`</parameter>`<br>`<parameter>`<br>`<name>resourceKeyType</name>`<br>`<value>java.lang.Integer</value>`<br>`</parameter>`<br>`</resourceParams>`<br>`</resource>`<br>`</service>`<br><br>`<!-- Factory service -->`<br>`<service name="HelloFactoryService6">`<br>`<resourceLink name="home"`<br>`target="java:comp/env/services/HelloWorldService6/home"/>`<br>`</service>`<br><br>`</jndiConfig>` |

**Modified Implementation of our Client:**

Last thing to do is to write Client to test our WSRF Web Service. Below is the complete code of Client7.java followed by discussing important parts, notice that Client5.java is in package "uk.ac.dl.ws.service.*client*" and our service HelloWorld.java is in package "*package* uk.ac.dl.ws.service". This is not the requirement of WSRF or Globus implementation of WSRF, in fact it is restriction due to the Ant "build.xml" file which will be used later to generate stubs, compile service and deploy to the container. You can change "build.xml" to adjust changes made in your package structure. We have to put all java classes and additional files like WSDL and "deploy-jndi-config.xml" in specific directories as required by Ant "build.xml". Ant "build.xml" is so handy and useful that following few restrictions is still worthy, as it hides all dirty work of setting class path, copying files from different locations and above all making all imported files in our WSDL available to our WSDL file.

```
package uk.ac.dl.ws.service.client;

//import org.apache.axis.message.MessageElement;
//import org.apache.axis.message.Text;

import org.oasis.wsrf.properties.WSResourcePropertiesServiceAddressingLocator;
import org.oasis.wsrf.properties.GetResourceProperty;
import org.oasis.wsrf.properties.GetResourcePropertyResponse;
import org.apache.axis.message.addressing.EndpointReferenceType;
import org.oasis.wsrf.properties.QueryResourceProperties_Element;
import org.oasis.wsrf.properties.QueryResourcePropertiesResponse;
import org.oasis.wsrf.properties.QueryResourceProperties_PortType;
import org.oasis.wsrf.properties.QueryExpressionType;

import org.oasis.wsrf.lifetime.SetTerminationTime; // added related to LeaseTime
import org.oasis.wsrf.lifetime.SetTerminationTimeResponse; // added related to LeaseTime

import org.apache.axis.types.URI;

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

import org.globus.wsrf.WSRFConstants;
import org.globus.wsrf.client.BaseClient;
import org.globus.wsrf.encoding.ObjectSerializer;
import org.globus.wsrf.encoding.SerializationException;
import org.globus.wsrf.utils.AnyHelper;
import org.globus.wsrf.utils.FaultHelper;

import org.oasis.wsrf.lifetime.Destroy;

import java.io.FileWriter;
import java.io.IOException;

import java.util.List;
import java.util.*;

//import javax.xml.namespace.QName;
import javax.xml.rpc.Stub;
import uk.ac.dl.ws.service.HelloQNames1;
import uk.ac.dl.wsrf.tutorial7.helloworld.service.*;
import uk.ac.dl.wsrf.tutorial7.helloworld.*;

public class Client7 extends BaseClient {

  final static HelloWorld7ServiceAddressingLocator locator =
     new HelloWorld7ServiceAddressingLocator();

  final static HelloFactoryService7ServiceAddressingLocator factoryLocator =
     new HelloFactoryService7ServiceAddressingLocator();

  public static void main(String[] args) {
```

```
Client7 client = new Client7();

// first, parse the commandline
try {
  CommandLine line = client.parse(args);
}
catch (ParseException e) {
  System.err.println("Parsing failed: " + e.getMessage());
  System.exit(1);
}
catch (Exception e) {
  System.err.println("Error: " + e.getMessage());
  System.exit(1);
}

try {
  EndpointReferenceType instanceEPR;
  HelloWorldPortType port;
  HelloWorldFactoryPortType factoryPort = factoryLocator.
     getHelloWorldFactoryPortTypePort(client.getEPR());
  CreateResourceResponse createResponse = factoryPort
     .createResource(new CreateResource());

  instanceEPR = createResponse.getEndpointReference();
  System.out.println("new resource created...");

  instanceEPR.getAddress().setPort(8082);

  System.out.println("Before Creating instance.");
  // Get instance PortType
  port = locator.getHelloWorldPortTypePort(instanceEPR);
  System.out.println("Created instance.");

  // This is the XPath query that we will use.
  String xpathQuery = "/*";

  QueryExpressionType query = new QueryExpressionType();
  query.setDialect(new URI(WSRFConstants.XPATH_1_DIALECT));
  query.setValue(xpathQuery);
  QueryResourceProperties_Element qrp =
     new QueryResourceProperties_Element(query);
  //client.setOptions((Stub)queryPort);

  QueryResourcePropertiesResponse response1 =
     port.queryResourceProperties(qrp);
  System.out.println(AnyHelper.toSingleString(response1));

  GetResourcePropertyResponse response =
     port.getResourceProperty(HelloQNames1.RP_NAME);
  System.out.println(AnyHelper.toSingleString(response));

  String result = port.echo("Rob Allan First Time Tutorial-7");
  response =
     port.getResourceProperty(HelloQNames1.RP_NAME);
  System.out.println(AnyHelper.toSingleString(response));
  /*
        response =
     port.getResourceProperty(HelloQNames1.LAST_MODIFIED_RP);
        System.out.println(AnyHelper.toSingleString(response));
   */
  response1 = port.queryResourceProperties(qrp);
  System.out.println(AnyHelper.toSingleString(response1));

  result = port.echo("Rob Allan Second Time Tutorial-7");
  response =
     port.getResourceProperty(HelloQNames1.RP_NAME);
```

```
    System.out.println(AnyHelper.toSingleString(response));
    response1 = port.queryResourceProperties(qrp);
    System.out.println(AnyHelper.toSingleString(response1));

    System.out.println(port.getNameRP(new GetNameRPRequest()));
    //port.destroy(new Destroy());
    //System.out.println(port.getNameRP(new GetNameRPRequest()));

    // Stuff Related to LeaseTime
    Calendar termination = Calendar.getInstance();
    termination.add(Calendar.SECOND, 2);
    SetTerminationTime requestLT;
    SetTerminationTimeResponse responseLT;
    requestLT = new SetTerminationTime(termination);
    responseLT = port.setTerminationTime(requestLT);
    boolean terminated = false;
    int seconds = 0;
    while (!terminated) {
      try {
        System.out.println(
           "*****************************************************");
        System.out.println("Second " + seconds);
        response1 = port.queryResourceProperties(qrp);
        System.out.println(AnyHelper.toSingleString(response1));
        Thread.sleep(1000);
        seconds++;
      }
      catch (Exception e) {
        System.out.println("Resource has been destroyed");
        terminated = true;
      }
    }

  }
  catch (Exception e) {
    if (client.isDebugMode()) {
      FaultHelper.printStackTrace(e);
    }
    else {
      System.err.println("Error: " + FaultHelper.getMessage(e));
    }
  }
 }
}
```

Implementation of the Client7.java is very simple to previous, tutorials and there are only few changes.

```
    // Stuff Related to LeaseTime
    Calendar termination = Calendar.getInstance();
    termination.add(Calendar.SECOND, 2);
    SetTerminationTime requestLT;
    SetTerminationTimeResponse responseLT;
    requestLT = new SetTerminationTime(termination);
    responseLT = port.setTerminationTime(requestLT);
```

In the code segment 2 Seconds are added to current time and then used to create setTerminationTime(). To check if resource is terminated there is while loop, with two utility variables, terminated and seconds:

```
    boolean terminated = false;
    int seconds = 0;
    while (!terminated) {
      try {
        System.out.println(
           "*****************************************************");
        System.out.println("Second " + seconds);
        response1 = port.queryResourceProperties(qrp);
        System.out.println(AnyHelper.toSingleString(response1));
        Thread.sleep(1000);
```

```
        seconds++;
      }
      catch (Exception e) {
        System.out.println("Resource has been destroyed");
        terminated = true;
      }
    }
```

This loop keeps in iterating and querying the resource properties until there is error from Service. Technically speaking loop should iterate 2 times, but this is not the case as Globus implementation compares currentTime and terminationTime by default after interval of 60 seconds, therefore there are more chances that loop will iterate 42–47 times. You can change this default value by adding sweeperDelay parameter in the "deploy-jndi-config.xml" for HelloWordService7, below 1000 milliseconds mean 10 seconds.

```
        <parameter>
            <name>sweeperDelay</name>
            <value>1000</value>
        </parameter>
```

In the final code, which you can download it is already set to 10 seconds.

**Modified "post-deploy":**

Now everything is available and we have to build the WSRF Web Service. For build purposes we will use build tool "ANT" and the build.xml provided by the Globus. To make life easy to test web service we need last file "post-deploy.xml" which will create script to run the client, without setting class path, without worrying about the generated stubs and location of compiled stubs. This xml file is generated when everything goes well, below are the contents of file "post-deploy.xml".

```
<project default="all" basedir=".">
  <property environment="env"/>
  <property file="build.properties"/>
  <property file="${user.home}/build.properties"/>
  <property name="env.GLOBUS_LOCATION" value="."/>
  <property name="deploy.dir" location="${env.GLOBUS_LOCATION}"/>
  <property name="abs.deploy.dir" location="${deploy.dir}"/>
  <property name="build.launcher"
    location="${abs.deploy.dir}/share/globus_wsrf_common/build-launcher.xml"/>

  <target name="setup">
    <ant antfile="${build.launcher}"
      target="generateLauncher">
      <property name="launcher-name" value="helloworld7"/>
      <property name="class.name"  value="uk.ac.dl.ws.service.client.Client7"/>
    </ant>
  </target>
</project>
```

This file is quite simple, and most of time remains same for most of clients. There are two adjustments to be made for each client and location of those adjustments is shown in bold. Name of the generated script: ***<property name="launcher-name" value="helloworld5"/>*** and location and name of java client of WSRF Web Service. In our case client is "*Client5.java*" in package "***uk.ac.dl.ws.service.client":***

```
        <property name="class.name"    value="uk.ac.dl.ws.service.client.Client5"/>
```

| Tutorial 7: *"post-deploy.xml"* | Tutorial 6: *"post-deploy.xml"* |
|---|---|
| `<project default="all" basedir=".">`<br>`  <property environment="env"/>`<br>`  <property file="build.properties"/>`<br>`  <property file="${user.home}/build.properties"/>` | `<project default="all" basedir=".">`<br>`  <property environment="env"/>`<br>`  <property file="build.properties"/>`<br>`  <property file="${user.home}/build.properties"/>` |

```
    <property name="env.GLOBUS_LOCATION" value="."/>
    <property name="deploy.dir"
        location="${env.GLOBUS_LOCATION}"/>
    <property name="abs.deploy.dir" location="${deploy.dir}"/>
    <property name="build.launcher"

location="${abs.deploy.dir}/share/globus_wsrf_common/build-
launcher.xml"/>

    <target name="setup">
      <ant antfile="${build.launcher}"
          target="generateLauncher">
        <property name="launcher-name" value="helloworld7"/>
        <property name="class.name"
            value="uk.ac.dl.ws.service.client.Client7"/>
      </ant>
    </target>
</project>
```

```
    <property name="env.GLOBUS_LOCATION" value="."/>
    <property name="deploy.dir"
      location="${env.GLOBUS_LOCATION}"/>
    <property name="abs.deploy.dir" location="${deploy.dir}"/>
    <property name="build.launcher"

location="${abs.deploy.dir}/share/globus_wsrf_common/build-
launcher.xml"/>

    <target name="setup">
      <ant antfile="${build.launcher}"
          target="generateLauncher">
        <property name="launcher-name" value="helloworld6"/>
        <property name="class.name"
            value="uk.ac.dl.ws.service.client.Client6"/>
      </ant>
    </target>
</project>
```

**Modified Ant Script:**

Now we have to use Ant build file, but before that we have to change the build file which are very minimum changes i.e. name changes (which we are doing in all tutorials). The changes are only about changing the name of WSDL files generated by the Globus, in last tutorials we had added another Service thus we have now two Services, HelloWorldService7 and HelloWorldFactoryService7. These changes are the last step required to finish the whole implementation.

Once again I have copied the complete build file and highlighted the changes related to Tutorial 5 in Bold Green Font:

```
<?xml version="1.0" encoding="UTF-8"?>
<project basedir="." default="all" name="globus_tutorial7_ helloworld ">
    <description>
        WSRF MDS Aggregator Implementation
    </description>

     <property environment="env"/>

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

    <property location="../install" name="env.GLOBUS_LOCATION"/>
    <property location="${env.GLOBUS_LOCATION}" name="deploy.dir"/>
    <property name="base.name" value="tutorial7_helloworld"/>
    <property name="package.namç" value="globus_${base.name}"/>
    <property name="jar.name" value="${package.name}.jar"/>
    <property name="gar.name" value="${package.name}.gar"/>
    <property location="build" name="build.dir"/>
    <property location="build/class_'14" name="build.dest"/>
    <property location="build/lib" name="build.lib.xir"/>
    <property location="build/stubs" name="stubs.dir"/>
    <property location="build/stubs/_rc" name="stubs.src"/>
    <property location="build/stubs/classes" name="stubs.dest"/>
    <property name="stubs.jar.name" value="${package.name}_stubs.jar"/>
    <property location="${deploy.dir}/share/globus_wsrf_common/build-packages.xml"
            name="build.packages"/>
    <property location="${deploy.dir}/share/globus_wsrf_tools/build-stubs.xml" name="build.stubs"/>
```

```
<property name="java.debug" value="on"/>

<property name="test-reports.dir" value="test-reports"/>
<property name="junit.haltonfailure" value="true"/>

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

<property name="garjars.id" value="garjars"/>
<fileset dir="${build.lib.d_u114 ?}" id="garjars"/>

<property name="garschema.id" value="garschema"/>
<fileset dir="${schema.dest}" id="garschema" includes="tutorial/*"/>

<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.d_u114 ?}"/>

   <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/**/*"/>
    </fileset>
   </copy>
   <copy toDir="${schema.dest}">
      <fileset dir="schema/" casesensitive="yes">
        <include name="tutorial/*"/>
     </fileset>
   </copy>

   <available file="${stubs.dest}/org.globus.wsrf.mds.aggregator" property="stubs.present" type="dir"/>
</target>

<target name="bindings" depends="init">
<ant antfile="${build.stub_}" target="generateBindin,">
      <property name="source.binding.dir"
        value="${schema.dest}/tutorial"/>
      <property name="target.binding.dir"
        value="${schema.dest}/tutorial"/>
      <property name="binding.root" value="HelloWorld7"/>
      <property name="porttype.wsdl" value="helloworld7_port_type.wsdl"/>
   </ant>
</target>

<target name="bindingsFactory" depends="bindings">
```

```
  <ant antfile="${build.stub_}" target="generateBindin(">
    <property name="source.bin€u105 ?ng.dir"
      value="${schema.dest}/tutorial"/>
    <property name="target.binxing.dir"
      value="${schema.dest}/tutorial"/>
    <property name="binding.root" value="FactoryService7"/>
    <property name="porttype.wsdl" value="factory7_port_type.wsdl"/>
  </ant>
</target>

  <target name="stubs" unless="stubs.present" depends="bindingsFactory">
    <ant antfile="${build.stub_}" target="generateStub_">
      <property location="${schema.dest}/tutorial" name="source.stubs.dir"/>
      <property name="wsdl.file" value="HelloWorld7__service.wsdl"/>
      <property location="${stubs.src}" name="target.stubs.dir"/>
    </ant>
  </target>

   <target name="stubsFactory" unless="stubs.present" depends="stubs">
    <ant antfile="${build.stub_}" target="generateStub_">
      <property location="${schema.dest}/tutorial" name="source.stubs.dir"/>
      <property name="wsdl.file" value="FactoryService7_service.wsdl"/>
      <property location="${stubs.src}" name="target.stubs.dir"/>
    </ant>
  </target>

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

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

  <target depends="compileStubs" name="jarStubs">
    <jar basedir="${stubs.dest}" destfile="${build.lib.d_u114 ?}/${stubs.jar.name}"/>
  </target>
```

```
<target depends="compile" name="jar">
  <jar basedir="${build.dest}" destfile="${build.lib.d_u114 ?}/${jar.name}"/>
</target>

<target depends="jar, jarStub_" name="dist">
  <ant antfile="${build.packages}" target="makeGar">
    <property name="gar.name" value="${build.lib.d_u114 ?}/${gar.name}"/>
    <reference refid="${garjars.id}"/>
    <reference refid="${garschema.id}"/>
    <reference refid="${garetc.i€u125 ?"/>
  </ant>
</target>

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

<target depends="dist" name="deploy">
  <ant antfile="${build.packages}" target="deployGar">
    <property name="gar.name" value="${build.lib.d_u114 ?}/${gar.name}"/>
    <property name="gar.id" value="${package.name}"/>
    <!-- <property name="noSchema" value="true"/> -->
  </ant>
</target>

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

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

<target depends="compile" name="test">
  <mkdir dir="${test-reports.dir}"/>
  <junit fork="yes" haltonfailure="${junit.haltonfailure}" printsummary="yes" timeout="600000">
    <classpath>
      <pathelement location="${build.dest}"/>
      <pathelement location="${deploy.dir}"/>
      <pathelement location="${deploy.dir}/etc/wsrf-bundles.properties"/>
      <fileset dir="${deploy.dir}/lib">
        <include name="*.jar"/>
      </fileset>
    </classpath>
    <formatter type="xml"/>
    <batchtest todir="${test-reports.dir}">
      <fileset dir="${build.dest}">
        <include name="**/*Test.class"/>
      </fileset>
    </batchtest>
  </junit>
</target>
```

```
</project>
```

There are only 8 minor changes in the Ant Script from Tutorial 4; those changes are highlighted in the Bold Green Font. Changes in the script are more related to changing the name of WSDL file created by us for Tutorial 7 i.e. "*helloworld7_port_type.wsdl*" and "*factory7_port_type.wsdl*", and the name of final WSDL which Ant Script will use to generate complete WSDL file, these changes are in target "bindings" and "bindingsFactory"

```
< property name="binding.root" value ="HelloWorld7"/>
< property name="porttype.wsdl" value ="helloworld7_port_type.wsdl"/>

<property name="binding.root" value="FactoryService7"/>
<property name="porttype.wsdl" value="factory7_port_type.wsdl"/>
```

Two changes are related to the name of final WSDL file used to create stubs for both Services, these changes are in target "stubs" and "stubsFactory":

```
<property name="wsdl.file" value="HelloWorld7__service.wsdl"/>

<property name="wsdl.file" value="FactoryService7_service.wsdl"/>
```

Before we can run Ant script we have to put them at proper location where Ant build can locate them. Tutorial-7 is the main directory containing all files are under one umbrella, our Web Service implementation *HelloWorldService7.java* will be in the directory structure according to package statement i.e. "*Tutorial-5\source\src\uk\ac\dl\ws\service*" our *Client7.java* will be the directory "*Tutorial-7\source\src\uk\ac\dl\ws\service\client*" (2). Our WSDL files "*helloworld7_port_type.wsdl*" and "*factory7_port_type.wsdl*" are located at "*Tutorial-7\source\schema\tutorial*" and source directory contains "deploy-server.wsdd", "deploy-jndi-config.xml" and "build.xml". File "post-deploy.xml", which; is used for creating client script is in directory "*Tutorial-7\source\etc*".

**Compiling and Deploying Modified WSRF Web Service:**

I am using Windows Operating System and have installed only WS-Core component of Globus Toolkit. You have to set an environment variable called GLOBUS_LOCATION referencing the installation of Globus Toolkit. In my case it is:

```
set GLOBUS_LOCATION= E:\GlobusWeek\gt-install
```

Open two Command Prompt one for starting Globus Tomcat Server and other for deploying Web Service and testing client. In one window go to directory *E:\GlobusWeek\ Tutorial-7\source* and run build file to create stubs, compile all java files, and deploy Web Service.

```
ant clean
ant deploy
```

If everything goes fine then after 14 seconds you will see the final outcome similar to:

```
generateWindows:
    [echo] Creating Windows launcher script helloworld7
    [copy] Copying 1 file to E:\GlobusWeek\gt-install\bin
  [delete] Deleting: E:\GlobusWeek\Tutorial-7\source\tmp\gar\etc\post-deploy.xml
  [delete] Deleting directory E:\GlobusWeek\Tutorial-7\source\tmp\gar

BUILD SUCCESSFUL
Total time: 14 seconds
```

In second Window go to directory "E:\GlobusWeek\gt-install" and run command:

```
bin\globus-start-container –nosec
```

This will start the container.

**Testing Modified WSRF Web Service:**

If you remember in the file "post-deploy.xml" we have mentioned helloworld7 as name of script to be generated to run the client, which is confirmed by the final outcome of the Ant build file. Now it is time to run the client:

```
%GLOBUS_LOCATION%\bin\helloworld7 -s http://localhost:8082/wsrf/services/HelloFactoryService7
new resource created...
Before Creating instance.
Created instance.
<ns0:HelloWorldResourcePropertiesSet
      xmlns:ns0="http://tutorial7.wsrf.dl.ac.uk/helloworld"
      xmlns:ns1="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.xsd"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ns1:name xmlns:ns1="http://tutorial7.wsrf.dl.ac.uk/helloworld">
     Asif Akram from Tutorial 7
   </ns1:name>
   <ns1:lastModified xmlns:ns1="http://tutorial7.wsrf.dl.ac.uk/helloworld">
      2005-07-13T21:14:38.500Z
   </ns1:lastModified>
   <ns1:TerminationTime xsi:nil="true"/>
   <ns1:CurrentTime>2005-07-13T21:14:39.531Z</ns1:CurrentTime>
 </ns0:HelloWorldResourcePropertiesSet>
  ……..
 ……..
Rob Allan Second Time Tutorial-7

**********************************************************
Second 0
<ns0:HelloWorldResourcePropertiesSet
xmlns:ns0=http://tutorial7.wsrf.dl.ac.uk/helloworld
 xmlns:ns1="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.xsd">
    ……..
  <ns1:TerminationTime>2005-07-13T21:14:43.968Z</ns1:TerminationTime>
  <ns1:CurrentTime>2005-07-13T21:14:42.546Z</ns1:CurrentTime>
</ns0:HelloWorldResourcePropertiesSet>

**********************************************************
Second 1
<ns0:HelloWorldResourcePropertiesSet
xmlns:ns0=http://tutorial7.wsrf.dl.ac.uk/helloworld
 xmlns:ns1="http://docs.oasis-open.org/wsrf/2004/06/wsr
f-WS-ResourceLifetime-1.2-draft-01.xsd">
  ……..
 <ns1:TerminationTime>2005-07-13T21:14:43.968Z</ns1:TerminationTime>
 <ns1:CurrentTime>2005-07-13T21:14:43.859Z</ns1:CurrentTime>
</ns0:HelloWorldResourcePropertiesSet>

**********************************************************
Second 2
Resource has been destroyed
```

"*Asif Akram from Tutorial 7*" is the initial value assigned to our variable "name" and "*Rob Allan First Time Tutorial-7*" is the value assigned to variable "name" from "Client7.java". Client7.java calls "echo" method twice and second time it passes the value "*Rob Allan Second Time Tutorial-7*". After calling echo() second time, we have set the termination time for our resource for 2 seconds, and in the while loop we keep on querying the

ResourcePropertySet. The response to the query is all resource properties including CurrentTime and TerminationTime. Once Resource is destroyed further query results in the error and we exit while loop. SOAP Requests and SOAP Responses are very similar to the last tutorials so I will not repeat all of them and will show only one SOAP Request and SOAP Response where client is setting the termination time.

| SOAP Request | Soap Response |
|---|---|
| <soapenv:Envelope<br>xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"<br>xmlns:xsd="http://www.w3.org/2001/XMLSchema"<br>xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"<br>xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing"><br><soapenv:Header><br><wsa:MessageID soapenv:mustUnderstand="0"><br>uuid:573d3070-f5f9-11d9-905f-e4cc6fb43c7e<br></wsa:MessageID><br><*wsa:To soapenv:mustUnderstand="0"*><br>*http://192.168.123.100:8082/wsrf/services/HelloWorldService7*<br>*</wsa:To>*<br><wsa:Action soapenv:mustUnderstand="0"><br>http://....wsrf-WS-ResourceLifetime/SetTerminationTime<br></wsa:Action><br><wsa:From soapenv:mustUnderstand="0"><br><wsa:Address><br>http://schemas.xmlsoap.org/ws/2004/03/addressing/role/anonymous<br></wsa:Address><br></wsa:From><br><ns1:NameKeyFactory7<br>xmlns:ns1="http://tutorial7.wsrf.dl.ac.uk/helloworld"<br>soapenv:mustUnderstand="0"><br>**8009378**<br></ns1:NameKeyFactory7><br></soapenv:Header><br><soapenv:Body><br><SetTerminationTime<br>xmlns="http://.....wsrf-WS-ResourceLifetime-1.2-draft-01.xsd"><br><RequestedTerminationTime><br>**2005-07-16T12:58:46.328Z**<br></RequestedTerminationTime><br></SetTerminationTime><br></soapenv:Body><br></soapenv:Envelope> | <soapenv:Envelope<br>xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"<br>xmlns:xsd="http://www.w3.org/2001/XMLSchema"<br>xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"<br>xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing"><br><soapenv:Header><br><wsa:MessageID soapenv:mustUnderstand="0"><br>uuid:5765c710-f5f9-11d9-90fd-cf43b2eff3af<br></wsa:MessageID><br><wsa:To soapenv:mustUnderstand="0"><br><br>http://schemas.xmlsoap.org/ws/2004/03/addressing/role/anonymous<br></wsa:To><br><wsa:Action soapenv:mustUnderstand="0"><br>http://.../wsrf-WS-ResourceLifetime/SetTerminationTimeResponse<br></wsa:Action><br><wsa:From soapenv:mustUnderstand="0"<br>xmlns:ns1="http://tutorial7.wsrf.dl.ac.uk/helloworld"><br><wsa:Address><br><br>http://192.168.123.100:8082/wsrf/services/HelloWorldService7<br></wsa:Address><br><wsa:ReferenceProperties><br><ns1:NameKeyFactory7 soapenv:mustUnderstand="0"<br>xmlns:ns1="http://tutorial7.wsrf.dl.ac.uk/helloworld"<br><br>xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"><br>**8009378**<br></ns1:NameKeyFactory7><br></wsa:ReferenceProperties><br></wsa:From><br><wsa:RelatesTo RelationshipType="wsa:Reply"<br>soapenv:mustUnderstand="0"><br>uuid:573d3070-f5f9-11d9-905f-e4cc6fb43c7e<br></wsa:RelatesTo><br></soapenv:Header><br><soapenv:Body><br>**<SetTerminationTimeResponse**<br>**xmlns="http://../wsrf-WS-ResourceLifetime-1.2-draft-01.xsd">**<br>**<NewTerminationTime>**<br>**2005-07-16T12:58:46.328Z**<br>**</NewTerminationTime>**<br>**<CurrentTime>**<br>**2005-07-16T12:58:44.609Z**<br>**</CurrentTime>**<br>**</SetTerminationTimeResponse>**<br></soapenv:Body><br></soapenv:Envelope> |

SOAP Request to setTerminationTime is very simple; Client reads the current System time, adds 2 seconds in it and then sends new time to the HelloWorldService7 as shown below:

*<SetTerminationTime*
*xmlns="http://.....wsrf-WS-ResourceLifetime-1.2-draft-01.xsd">*
*<RequestedTerminationTime>*
***2005-07-16T12:58:46.328Z***
*</RequestedTerminationTime>*
*</SetTerminationTime>*

SOAP Response for setTermincationTime includes CurrentTime and NewTerminationTime as shown in the following SOAP Response fragment, notice current time is *12:58:44.328Z* and new termination time is *12:58:46.328Z*:

```
<SetTerminationTimeResponse
  xmlns="http://../wsrf-WS-ResourceLifetime-1.2-draft-01.xsd">
   <NewTerminationTime>
       2005-07-16T12:58:46.328Z
  </NewTerminationTime>
   <CurrentTime>
       2005-07-16T12:58:44.609Z
   </CurrentTime>
</SetTerminationTimeResponse>
```