

Creating a new web-service in Genesis II

Krasimira Kapitanova

Sal Valente

06/23/2010

Updated: Vanamala Venkataswamy

08/23/2011

1 The XML Documents

Go into the directory `GenesisII/wsd1`. There you should see subdirectories like `rns` and `queue`. Create a new subdirectory called `addition`. In the `addition` directory, create three XML documents:

1. `addition.xsd`. An `xsd` file is an XML Schema Definition file. Here the elements that will be used are specified. In the `addition` example, there is a complex element encapsulating the two integer values to be added, and there is a complex type element for the result.
2. `addition.gwsdl`. This file contains information about the messages that will be exchanged and the port through which they will pass. (An "AdditionRequest" message contains an "AdditionRequest" element. The message is defined in the `gwsdl` file; the element is defined in the `xsd` file.)
3. `addition.build.xml`. This is the build file for the service. Basically, all you need to do to change this file is to replace the name of the old service with the name of your own. Just make sure that the paths are correct.

After the three files are created, you need to change two files:

1. `mappings.properties`. This file is used by the WSDL2Java program. The mapping "a = b" says that if the input file is in a target namespace of "a", then the output file must be in a package of "b". Add a single line to this file for the new service:
`http://vcgr.cs.virginia.edu/genii/addition=edu.virginia.vcgr.genii.addition`
2. `build.xml`. This file needs to be changed so that it will run WSDL2Java on the new service. Find the "generate.ws.source" target. Add a single line to run an "ant" task on "addition.build.xml":
`<ant antfile="${wsdl.dir}/addition/addition.build.xml"/>`

Now, generate the source code using WSDL2Java and related processes. In Eclipse, run the "Build Codegen" external task. (or using command line "ant build")

2 Creating the Container

Go into the directory `GenesisII/src/edu/virginia/vcgr/genii/container`. There you should see subdirectories like `bes` and `rns` and `q2`.

1. Create a new subdirectory called `addition`.

2. Add a java class called `AdditionServiceImpl.java`. This class needs to extend the base container class "GenesisIIBase". Note that this is consistent with how the service was defined in the wsdl file.
3. In this class, implement what the service will actually do.

Go into the directory `GenesisII/deployments/default/configuration`. Change two files:

1. `server-config.xml`. Add the service named "AdditionPortType":


```
<genii:service name="AdditionPortType">
  <genii:resource-provider name="basic-db-resource-provider"/>
  <genii:authz-provider name="gamlacl-authz-provider"/>
</genii:service>
```
2. `client-config.xml`. In the `genii:locator-registry` element, add a new locator element:


```
<locator interface="edu.virginia.vcgr.genii.addition.AdditionPortType"
locator-type="edu.virginia.vcgr.genii.addition.AdditionServiceLocator"/>
```

Note that "AdditionServiceLocator" is a class that was created by WSDL2Java.

Now, compile the entire project and run the container. It should start up without errors. The console should include this line:

```
Deployed edu.virginia.vcgr.genii.container.addition.AdditionServiceImpl.
```

3 Writing the Client

Go into the directory `GenesisII/src/edu/virginia/vcgr/genii/client/cmd/tools`. There you should see tools like `LsTool.java`.

1. `AdditionTool.java`. Create this new file.
2. `client-config.xml`. You edited this file in the previous section. Now edit it again. You need to tell the grid shell that when a user runs the "add" command, it should execute the "AdditionTool" class. In the "genii:tools" element, add a new "mconf:class" element:


```
<mconf:class name="add" base="edu.virginia.vcgr.genii.client.cmd.ITool">
  edu.virginia.vcgr.genii.client.cmd.tools.AdditionTool</mconf:class>
```

Now, compile the project and run the client.

4 Miscellaneous Files

Go into the directory `GenesisII/src/edu/virginia/vcgr/genii/client/cmd/tools`. Add "dadd" in to "description" directory, "add" in to "man" directory and "uls" in to "usage" directory respectively.

5 Running the Service

1. In the grid shell, go to `/containers/BootstrapContainer/Services`. List the directory. It should include "AdditionPortType".

2. Run `ls -e AdditionPortType`. It should show an address something like:
`https://127.0.0.1:18080/axis/services/AdditionPortType?genii-container-id=X`
3. The service can be executed:
`vcgr:$> add 4 5 /containers/BootstrapContainer/Services/AdditionPortType`
`4 + 5 = 9`

6 List of Files

1. add
2. addition.build.xml
3. addition.gwsdl
4. AdditionServiceImpl.java
5. AdditionTool.java
6. addition.xsd
7. dadd
8. uadd