Introduction:

OSGi technology emphasis on principles of modularity, component-orientation and service orientation. It uses the prior mention aspects to provide a dynamic framework that support dynamic service deployment. Hence, the software industry is more interested in providing OSGi compatible components.

In order to be more visible in OSGi market, Axis2 has taken provision to provide clean bundles that could be deploy in any OSGi implementation. This document will focus on the current Axis2 OSGi integration effort and the process involving deploying the bundles in Eclipse Equinox OSGi implementation.

Axis2 bundles:

As the starting point, Axis2 provides a bundle that encapsulate resources from Axis2-kernel and Axis2-adb with the provisions of providing extension to include other Axis2 modules. This bundle utilizes the principles of OSGi extender model and OSGi service model to be efficient in any OSGi implementation.

Users will be able to checkout Axis2 code base from [1], via SVN. There exist a Maven2 module under “modules” directory where all the improvements and research is going on. When this Maven2 module is built, it will generate an OSGi artifact “org.apache.axis2.osgi.jar”, which can be deployed in any OSGi environment.

The prior artifact wouldn't be able to resolved itself. It needs third party bundles which needs to be resolved prior.

org.apache.axis2.osgi bundle dependency:

org.apache.axis2.osgi bundle contains the minimum resources need to be fully operational Axis2 Engine in an OSGi environment. This bundle depends on other bundles. Following lists the dependent bundles.

1. Commons logging
2. Javax servlet
3. Annogen
4. Ant
5. Axiom
6. Commons codec
7. Commons fileuplad
8. Commons httpclient
9. Geronimo Stax
10. Geronimo WS metadata
11. Httpcore

12. Neethi
13. WSDL4J
14. Stax implementation (wstx-asl)
15. XmlSchema
16. Commons Io
17. Xml-apis (Xerces)
18. Woden
19. Geronimo Javamail
20. Geronimo Activation
21. Dom4j
22. Jaxen

The list is long. Unfortunately not every artifact from above list a bundle. Most of them are regular jars and they have to be re-bundle as OSGi bundles.

Hence, in order to demonstrate workings of Axis2 OSGi bundle, I have create a project [2] in Axis2 scratch area to convert these legacy jars into OSGi bundles. When it's time to do the demonstration, lets discuss about this more. Before that lets look at the dependency stats.

#	Artifact	Type	Bundle Available in Maven2 repository	OSGi bundle status
1	Commons logging	jar	no	Equinox bundle org.apache.commons.logging_1.0.4.v200706111724
2	Servlet API	jar	no	Equinox bundle javax.servlet_2.4.0.v200706111738
3	Annogen	jar	no	re-packaged as a bundle annogen_0.1.0
4	Ant	jar	no	re-packaged as a bundle ant_1.7.0
5	Axiom	bundle	no	This bundle wraps the classes of axiom-api, axiom-impl and axiom-dom, reflecting proper version. axiom_1.0.0.SNAPSHOT
6	Commons Codec	jar	no	re-packaged as a bundle commons.codec_1.3.0
7	Commons fileuplad	jar	no	re-packaged as a bundle commons.fileupload_1.2.0
8	Commons httpclient	jar	no	re-packaged as a bundle commons.httpclient_3.1.0
9	Geronimo stax	bundle	yes	org.apache.geronimo.specs.geronimo-stax-

				api_1.0_spec_1.0.1
10	Geronimo ws metadata	bundle	yes	org.apache.geronimo.specs.geronimo-ws-metadata_2.0_spec_1.1.2
11	Httpcore	jar	no	re-packaged as a bundle httpcore_4.0.0.beta1
12	Neethi	bundle	yes	org.apache.neethi_2.0.4
13	WSDL4j	jar	no	re-packaged as a bundle wsdl4j_1.6.2
14	WSTX-ASL	jar	no	re-packaged as a fragment bundle wstx-asl_3.2.4
15	XmlSchema	bundle	yes	org.apache.ws.commons.schema_1.4.2
16	Commons IO	no	no	re-packaged as a bundle commons.io_1.4.0
17	Xml-APIs (Xerces)	no	no	re-packaged as a bundle xml-apis_1.0.0
18	Wooden	bundle	no	repackaged, due to missing constraint exception woden_1.0.0.M8
19	Geronimo Javamail	bundle	yes	org.apache.geronimo.specs.geronimo-javamail_1.4_spec_1.2.0
20	Geronimo Activation	bundle	yes	org.apache.geronimo.specs.geronimo-activation_1.1_spec_1.0.1
21	Dom4j	jar	no	re-packaged as a bundle dom4j_1.6.1
22	Jaxen	jar	no	re-packaged as a bundle jaxen_1.1.1

Note:
Axiom and Woden jars are available as bundles in Maven2 repository. But,

In order to reduce the number of bundles that should be included from Axiom, this project contains a wrapper 3rd party Axiom bundle which export packages from Axiom-API, Axiom-IMPL and Axiom-DOM jars. This bundle handles the version of package and bundle properly and reflect them correspondingly.

Woden has two major bundles. Woden-api-1.0M8 and Woden-impl-dom-1.0M8. These bundles are available in Maven2 repository. When these two bundles try to resolve the dependencies, Woden-impl-dom bundle fails with "org.osgi.framework.BundleException: The bundle could not be resolved. Reason: Missing Constraint: Import-Package: org.apache.woden; version="1.0.0", in Equinox environment. Thus, in order to eliminate this problem a 3rd party wrapper Woden bundle has been created with proper version and package constraints.

Installing and running:

Following steps will allow you to deploy org.apache.axis2.osgi bundle in **Eclipse Equinox OSGi implementation**.

0. Take Axis2 checkout from [1] to any free place in your disk. Please build the project using "mvn clean install". If you are already taken an checkout from [1] and build the latest, you can simply ignore this step.
1. Take Axiom checkout from [4] to any fee place in your disk. Please build the project using **"mvn clean install"**. If you are already taken an checkout from [4] and build the latest, you can simply ignore this step.
2. Download latest Eclipse Equinox binary from [3]. Do download eclipse-equinox-3.3.2.zip. Unzip this and traverse to **eclipse/plugins** directory.
3. Take an checkout from [2]. This is the sample OSGi project, where we used in our demonstration. This project converts the legacy jars to bundles. From root type "**mvn clean install**" to build the project. Then traverse to d**istribution/target** folder and **unzip distribution-1.0.0.zip.** Travers to **distribution-1.0.0/plugins** directory and **copy** all the jars available there to the step 1 **plugins** directory.
4. Go to step 1 **plugins** directory. Then type

```
java  -Dorg.osgi.service.http.port=8080  -jar org.eclipse.osgi_3.3.2.R33x_v20080105.jar -console
```

-Dorg.osgi.service.http.port=8080 is used to start the connectors of underlying HttpService implementation. *org.eclipse.osgi_3.3.2.R33x_v20080105.jar*, provides the OSGi implementation. Once done, you will be able to see the following console when you type the command "*ss*".

```
osgi> ss
Framework is launched.
id      State       Bundle
0       ACTIVE      org.eclipse.osgi_3.3.2.R33x_v20080105

osgi>
```

5. Lets install and start the basic Equinox jars that we need. These bundles will provide commons logging, serlvet-api, HttpService and Equinox controls.

```
osgi> install file:org.eclipse.equinox.common_3.3.0.v20070426.jar
Bundle id is 1

osgi> install file:org.eclipse.osgi.services_3.1.200.v20070605.jar
Bundle id is 2

osgi> install file:org.eclipse.equinox.registry_3.3.1.R33x_v20070802.jar
```

```
Bundle id is 3

osgi> install file:org.apache.commons.logging_1.0.4.v200706111724.jar
Bundle id is 4

osgi> install file:javax.servlet_2.4.0.v200706111738.jar
Bundle id is 5

osgi> install file:org.eclipse.equinox.http.servlet_1.0.1.R33x_v20070816.jar
Bundle id is 6

osgi> install file:org.mortbay.jetty_5.1.11.v200706111724.jar
Bundle id is 7

osgi> install file:org.eclipse.equinox.http.jetty_1.0.1.R33x_v20070816.jar
Bundle id is 8

osgi>
```

Then type “ss”

```
osgi> ss

Framework is launched.

id    State       Bundle
0     ACTIVE      org.eclipse.osgi_3.3.2.R33x_v20080105
1     INSTALLED   org.eclipse.equinox.common_3.3.0.v20070426
2     INSTALLED   org.eclipse.osgi.services_3.1.200.v20070605
3     INSTALLED   org.eclipse.equinox.registry_3.3.1.R33x_v20070802
4     INSTALLED   org.apache.commons.logging_1.0.4.v200706111724
5     INSTALLED   javax.servlet_2.4.0.v200706111738
6     INSTALLED   org.eclipse.equinox.http.servlet_1.0.1.R33x_v20070816
7     INSTALLED   org.mortbay.jetty_5.1.11.v200706111724
8     INSTALLED   org.eclipse.equinox.http.jetty_1.0.1.R33x_v20070816

osgi>
```

As you see bundles are in “installed” phase. You need to start them

```
osgi> start 1 2 3 4 5 6 7 8
```

Then you will observe

```
Jun 21, 2008 9:34:45 PM org.mortbay.http.HttpServer doStart
INFO: Version Jetty/5.1.x
Jun 21, 2008 9:34:45 PM org.mortbay.util.Container start
INFO: Started org.mortbay.jetty.servlet.ServletHandler@15212bc
Jun 21, 2008 9:34:45 PM org.mortbay.util.Container start
INFO: Started HttpContext[/,/]
Jun 21, 2008 9:34:45 PM org.mortbay.http.SocketListener start
INFO: Started SocketListener on 0.0.0.0:8080
Jun 21, 2008 9:34:45 PM org.mortbay.util.Container start
INFO: Started org.mortbay.http.HttpServer@fec107

osgi>
```

Type “ss” again

```
osgi> ss

Framework is launched.

id    State      Bundle
0     ACTIVE     org.eclipse.osgi_3.3.2.R33x_v20080105
1     ACTIVE     org.eclipse.equinox.common_3.3.0.v20070426
2     ACTIVE     org.eclipse.osgi.services_3.1.200.v20070605
3     ACTIVE     org.eclipse.equinox.registry_3.3.1.R33x_v20070802
4     ACTIVE     org.apache.commons.logging_1.0.4.v200706111724
5     ACTIVE     javax.servlet_2.4.0.v200706111738
6     ACTIVE     org.eclipse.equinox.http.servlet_1.0.1.R33x_v20070816
7     ACTIVE     org.mortbay.jetty_5.1.11.v200706111724
8     ACTIVE     org.eclipse.equinox.http.jetty_1.0.1.R33x_v20070816

osgi>
```

Prior bundles are the **ONLY** bundles needed from Equinox. Now you are ready to install the other bundles needed to run “org.apache.axis2.osgi” bundle.

6. Let's install the other bundles . These bundles are the once we created from **step [3]** and copied to “**plugins**” directory. The order in which you are installing these bundles wouldn't matter. Event the starting order wouldn't matter. You just need to install and start them.

```
osgi> install file:annogen_0.1.0.jar
Bundle id is 9

osgi> install file:ant_1.7.0.jar
Bundle id is 10
```

```
osgi> install file:commons.codec_1.3.jar
Bundle id is 11

osgi> install file:commons.fileupload_1.2.jar
Bundle id is 12

osgi> install file:commons.httpclient_3.1.jar
Bundle id is 13

osgi> install file:commons.io_1.4.jar
Bundle id is 14

osgi> install file:geronimo-stax-api_1.0_spec_1.0.1.jar
Bundle id is 15

osgi> install file:geronimo-ws-metadata_2.0_spec_1.1.2.jar
Bundle id is 16

osgi> install file:httpcore_4.0-beta1.jar
Bundle id is 17

osgi> install file:neethi_2.0.4.jar
Bundle id is 18

osgi> install file:woden_1.0M8.jar
Bundle id is 19

osgi> install file:wsdl4j_1.6.2.jar
Bundle id is 20

osgi> install file:xml-apis_1.0.0.jar
Bundle id is 21

osgi> install file:XmlSchema_1.4.2.jar
Bundle id is 22

osgi> install file:wstx-asl_3.2.4.jar
Bundle id is 23

osgi> install file:jaxen_1.1.1.jar
Bundle id is 24

osgi> install file:dom4j_1.6.1.jar
```

```
Bundle id is 25

osgi> install file:geronimo-javamail_1.4_spec_1.2.jar
Bundle id is 26

osgi> install file:geronimo-activation_1.1_spec_1.0.1.jar
Bundle id is 27

osgi> install file:axiom_SNAPSHOT.jar
Bundle id is 28

osgi> install file:org.apache.axis2.osgi_SNAPSHOT.jar
Bundle id is 29

osgi>
```

Type “ss”

```
osgi> ss

Framework is launched.

id    State       Bundle
0     ACTIVE      org.eclipse.osgi_3.3.2.R33x_v20080105
1     ACTIVE      org.eclipse.equinox.common_3.3.0.v20070426
2     ACTIVE      org.eclipse.osgi.services_3.1.200.v20070605
3     ACTIVE      org.eclipse.equinox.registry_3.3.1.R33x_v20070802
4     ACTIVE      org.apache.commons.logging_1.0.4.v200706111724
5     ACTIVE      javax.servlet_2.4.0.v200706111738
6     ACTIVE      org.eclipse.equinox.http.servlet_1.0.1.R33x_v20070816
7     ACTIVE      org.mortbay.jetty_5.1.11.v200706111724
8     ACTIVE      org.eclipse.equinox.http.jetty_1.0.1.R33x_v20070816
9     INSTALLED   annogen_0.1.0
10    INSTALLED   ant_1.7.0
11    INSTALLED   commons.codec_1.3.0
12    INSTALLED   commons.fileupload_1.2.0
13    INSTALLED   commons.httpclient_3.1.0
14    INSTALLED   commons.io_1.4.0
15    INSTALLED   org.apache.geronimo.specs.geronimo-stax-api_1.0_spec_1.0.1
16    INSTALLED   org.apache.geronimo.specs.geronimo-ws-metadata_2.0_spec_1.1.2
17    INSTALLED   httpcore_4.0.0.beta1
18    INSTALLED   org.apache.neethi_2.0.4
19    INSTALLED   woden_1.0.0.M8
20    INSTALLED   wsdl4j_1.6.2
```

```
21     INSTALLED   xml-apis_1.0.0
22     INSTALLED   org.apache.ws.commons.schema_1.4.2
23     INSTALLED   wstx-asl_3.2.4
24     INSTALLED   jaxen_1.1.1
25     INSTALLED   dom4j_1.6.1
26     INSTALLED   org.apache.geronimo.specs.geronimo-javamail_1.4_spec_1.2.0
27     INSTALLED   org.apache.geronimo.specs.geronimo-activation_1.1_spec_1.0.1
28     INSTALLED   axiom_1.0.0.SNAPSHOT
29     INSTALLED   org.apache.axis2.osgi_1.0.0.SNAPSHOT

osgi>
```

Now you have left with starting the bundles. "wstx-asl_3.2.4" is a special bundle. We call it a "Fragment Host". Fragment host bundles cannot be started. They are attached to another bundle. In our case this bundle is attach to "org.apache.geronimo.specs.geronimo-stax-api_1.0_spec_1.0.1" bundle. Fragment host are resolved automatically by the framework. Hence, you just need to start bundles except "wstx-asl_3.2.4". Lets start them sequentially so you get a hang of it. Lets start the bundle from 9 to 15.

```
osgi> start 9 10 11 12 13 14 15
```

Then "ss"

```
osgi> ss

Framework is launched.

id     State       Bundle
0      ACTIVE      org.eclipse.osgi_3.3.2.R33x_v20080105
1      ACTIVE      org.eclipse.equinox.common_3.3.0.v20070426
2      ACTIVE      org.eclipse.osgi.services_3.1.200.v20070605
3      ACTIVE      org.eclipse.equinox.registry_3.3.1.R33x_v20070802
4      ACTIVE      org.apache.commons.logging_1.0.4.v200706111724
5      ACTIVE      javax.servlet_2.4.0.v200706111738
6      ACTIVE      org.eclipse.equinox.http.servlet_1.0.1.R33x_v20070816
7      ACTIVE      org.mortbay.jetty_5.1.11.v200706111724
8      ACTIVE      org.eclipse.equinox.http.jetty_1.0.1.R33x_v20070816
9      ACTIVE      annogen_0.1.0
10     ACTIVE      ant_1.7.0
11     ACTIVE      commons.codec_1.3.0
12     ACTIVE      commons.fileupload_1.2.0
13     ACTIVE      commons.httpclient_3.1.0
14     ACTIVE      commons.io_1.4.0
15     ACTIVE      org.apache.geronimo.specs.geronimo-stax-api_1.0_spec_1.0.1
```

```
            Fragments=23
16    RESOLVED    org.apache.geronimo.specs.geronimo-ws-metadata_2.0_spec_1.1.2
17    RESOLVED    httpcore_4.0.0.beta1
18    RESOLVED    org.apache.neethi_2.0.4
19    RESOLVED    woden_1.0.0.M8
20    RESOLVED    wsdl4j_1.6.2
21    RESOLVED    xml-apis_1.0.0
22    RESOLVED    org.apache.ws.commons.schema_1.4.2
23    RESOLVED    wstx-asl_3.2.4
            Master=15
24    RESOLVED    jaxen_1.1.1
25    RESOLVED    dom4j_1.6.1
26    RESOLVED    org.apache.geronimo.specs.geronimo-javamail_1.4_spec_1.2.0
27    RESOLVED    org.apache.geronimo.specs.geronimo-activation_1.1_spec_1.0.1
28    RESOLVED    axiom_1.0.0.SNAPSHOT
29    RESOLVED    org.apache.axis2.osgi_1.0.0.SNAPSHOT

osgi>
```

As you can see when some of the bundle are started, other bundle might go to “resolved” state. This is the ultimate feature of the framework. When you start a bundle, it will be resolved using the the rest of “installed” bundles and these bundles will go to “resolved” state.

When you look at org.apache.geronimo.specs.geronimo-stax-api_1.0_spec_1.0.1 and wstx-asl_3.2.4 bundle, you would see the master/fragment relationship. Lets start the rest of the bundles. It has to be noted that you cannot start a fragment bundle. Thus, I have left out the bundle 23.

```
osgi> start 16 17 18 19 20 21 22 24 25 26 27 28
```

Then “ss”

```
osgi> ss

Framework is launched.

id    State      Bundle
0     ACTIVE     org.eclipse.osgi_3.3.2.R33x_v20080105
1     ACTIVE     org.eclipse.equinox.common_3.3.0.v20070426
2     ACTIVE     org.eclipse.osgi.services_3.1.200.v20070605
3     ACTIVE     org.eclipse.equinox.registry_3.3.1.R33x_v20070802
4     ACTIVE     org.apache.commons.logging_1.0.4.v200706111724
5     ACTIVE     javax.servlet_2.4.0.v200706111738
```

```
6      ACTIVE     org.eclipse.equinox.http.servlet_1.0.1.R33x_v20070816
7      ACTIVE     org.mortbay.jetty_5.1.11.v200706111724
8      ACTIVE     org.eclipse.equinox.http.jetty_1.0.1.R33x_v20070816
9      ACTIVE     annogen_0.1.0
10     ACTIVE     ant_1.7.0
11     ACTIVE     commons.codec_1.3.0
12     ACTIVE     commons.fileupload_1.2.0
13     ACTIVE     commons.httpclient_3.1.0
14     ACTIVE     commons.io_1.4.0
15     ACTIVE     org.apache.geronimo.specs.geronimo-stax-api_1.0_spec_1.0.1
                Fragments=23
16     ACTIVE     org.apache.geronimo.specs.geronimo-ws-metadata_2.0_spec_1.1.2
17     ACTIVE     httpcore_4.0.0.beta1
18     ACTIVE     org.apache.neethi_2.0.4
19     ACTIVE     woden_1.0.0.M8
20     ACTIVE     wsdl4j_1.6.2
21     ACTIVE     xml-apis_1.0.0
22     ACTIVE     org.apache.ws.commons.schema_1.4.2
23     RESOLVED   wstx-asl_3.2.4
                Master=15
24     ACTIVE     jaxen_1.1.1
25     ACTIVE     dom4j_1.6.1
26     ACTIVE     org.apache.geronimo.specs.geronimo-javamail_1.4_spec_1.2.0
27     ACTIVE     org.apache.geronimo.specs.geronimo-activation_1.1_spec_1.0.1
28     ACTIVE     axiom_1.0.0.SNAPSHOT
29     RESOLVED   org.apache.axis2.osgi_1.0.0.SNAPSHOT

osgi>
```

Finally lets start “org.apache.axis2.osgi” bundle.

```
osgi> start 29
```

Then “ss”

```
Framework is launched.

id     State      Bundle
0      ACTIVE     org.eclipse.osgi_3.3.2.R33x_v20080105
1      ACTIVE     org.eclipse.equinox.common_3.3.0.v20070426
2      ACTIVE     org.eclipse.osgi.services_3.1.200.v20070605
3      ACTIVE     org.eclipse.equinox.registry_3.3.1.R33x_v20070802
4      ACTIVE     org.apache.commons.logging_1.0.4.v200706111724
5      ACTIVE     javax.servlet_2.4.0.v200706111738
```

```
6      ACTIVE     org.eclipse.equinox.http.servlet_1.0.1.R33x_v20070816
7      ACTIVE     org.mortbay.jetty_5.1.11.v200706111724
8      ACTIVE     org.eclipse.equinox.http.jetty_1.0.1.R33x_v20070816
9      ACTIVE     annogen_0.1.0
10     ACTIVE     ant_1.7.0
11     ACTIVE     commons.codec_1.3.0
12     ACTIVE     commons.fileupload_1.2.0
13     ACTIVE     commons.httpclient_3.1.0
14     ACTIVE     commons.io_1.4.0
15     ACTIVE     org.apache.geronimo.specs.geronimo-stax-api_1.0_spec_1.0.1
                  Fragments=23
16     ACTIVE     org.apache.geronimo.specs.geronimo-ws-metadata_2.0_spec_1.1.2
17     ACTIVE     httpcore_4.0.0.beta1
18     ACTIVE     org.apache.neethi_2.0.4
19     ACTIVE     woden_1.0.0.M8
20     ACTIVE     wsdl4j_1.6.2
21     ACTIVE     xml-apis_1.0.0
22     ACTIVE     org.apache.ws.commons.schema_1.4.2
23     RESOLVED   wstx-asl_3.2.4
                  Master=15
24     ACTIVE     jaxen_1.1.1
25     ACTIVE     dom4j_1.6.1
26     ACTIVE     org.apache.geronimo.specs.geronimo-javamail_1.4_spec_1.2.0
27     ACTIVE     org.apache.geronimo.specs.geronimo-activation_1.1_spec_1.0.1
28     ACTIVE     axiom_1.0.0.SNAPSHOT
29     ACTIVE     org.apache.axis2.osgi_1.0.0.SNAPSHOT

osgi>
```

You could start the bundles in one step or the order in which you like. I used the above approach only for this demonstration.

7. Now you have stated the Axis2 Engine. Now you need to install bundles that mimic Axis2 services and modules. I have created two bundles for this. They are “simple.module_1.0.0.jar” and “simple.version_1.0.0.jar” bundles. Code for these two bundles are available in [2]. “simple.version” contains a Version service. In services.xml this “Version” service contains a reference to “simpleModule”, which is available from “simple.module”. If you installed and start “simple.version” bundle first, then it has a unmet reference to “simpleModule”. At this point, “simple.version” bundle will go into “unresolved” dependency list. When such time the “simple.module” bundle is installed and started, the “simple.version” bundle will sprung to life and “resolved” itself. this will demonstrate the ultimate dynamism of Axis2 and OSGi. Lets demonstrate this.

```
osgi> install file:simple.version_1.0.0.jar
Bundle id is 30
```

```
osgi> ss

Framework is launched.

id      State       Bundle
0       ACTIVE      org.eclipse.osgi_3.3.2.R33x_v20080105
1       ACTIVE      org.eclipse.equinox.common_3.3.0.v20070426
2       ACTIVE      org.eclipse.osgi.services_3.1.200.v20070605
3       ACTIVE      org.eclipse.equinox.registry_3.3.1.R33x_v20070802
4       ACTIVE      org.apache.commons.logging_1.0.4.v200706111724
5       ACTIVE      javax.servlet_2.4.0.v200706111738
6       ACTIVE      org.eclipse.equinox.http.servlet_1.0.1.R33x_v20070816
7       ACTIVE      org.mortbay.jetty_5.1.11.v200706111724
8       ACTIVE      org.eclipse.equinox.http.jetty_1.0.1.R33x_v20070816
9       ACTIVE      annogen_0.1.0
10      ACTIVE      ant_1.7.0
11      ACTIVE      commons.codec_1.3.0
12      ACTIVE      commons.fileupload_1.2.0
13      ACTIVE      commons.httpclient_3.1.0
14      ACTIVE      commons.io_1.4.0
15      ACTIVE      org.apache.geronimo.specs.geronimo-stax-api_1.0_spec_1.0.1
                Fragments=23
16      ACTIVE      org.apache.geronimo.specs.geronimo-ws-metadata_2.0_spec_1.1.2
17      ACTIVE      httpcore_4.0.0.beta1
18      ACTIVE      org.apache.neethi_2.0.4
19      ACTIVE      woden_1.0.0.M8
20      ACTIVE      wsdl4j_1.6.2
21      ACTIVE      xml-apis_1.0.0
22      ACTIVE      org.apache.ws.commons.schema_1.4.2
23      RESOLVED    wstx-asl_3.2.4
                Master=15
24      ACTIVE      jaxen_1.1.1
25      ACTIVE      dom4j_1.6.1
26      ACTIVE      org.apache.geronimo.specs.geronimo-javamail_1.4_spec_1.2.0
27      ACTIVE      org.apache.geronimo.specs.geronimo-activation_1.1_spec_1.0.1
28      ACTIVE      axiom_1.0.0.SNAPSHOT
29      ACTIVE      org.apache.axis2.osgi_1.0.0.SNAPSHOT
30      INSTALLED   simple.version_1.0.0

osgi>
```

```
osgi> start 30
osgi> ss

Framework is launched.

id    State       Bundle
0     ACTIVE      org.eclipse.osgi_3.3.2.R33x_v20080105
1     ACTIVE      org.eclipse.equinox.common_3.3.0.v20070426
2     ACTIVE      org.eclipse.osgi.services_3.1.200.v20070605
3     ACTIVE      org.eclipse.equinox.registry_3.3.1.R33x_v20070802
4     ACTIVE      org.apache.commons.logging_1.0.4.v200706111724
5     ACTIVE      javax.servlet_2.4.0.v200706111738
6     ACTIVE      org.eclipse.equinox.http.servlet_1.0.1.R33x_v20070816
7     ACTIVE      org.mortbay.jetty_5.1.11.v200706111724
8     ACTIVE      org.eclipse.equinox.http.jetty_1.0.1.R33x_v20070816
9     ACTIVE      annogen_0.1.0
10    ACTIVE      ant_1.7.0
11    ACTIVE      commons.codec_1.3.0
12    ACTIVE      commons.fileupload_1.2.0
13    ACTIVE      commons.httpclient_3.1.0
14    ACTIVE      commons.io_1.4.0
15    ACTIVE      org.apache.geronimo.specs.geronimo-stax-api_1.0_spec_1.0.1
             Fragments=23
16    ACTIVE      org.apache.geronimo.specs.geronimo-ws-metadata_2.0_spec_1.1.2
17    ACTIVE      httpcore_4.0.0.beta1
18    ACTIVE      org.apache.neethi_2.0.4
19    ACTIVE      woden_1.0.0.M8
20    ACTIVE      wsdl4j_1.6.2
21    ACTIVE      xml-apis_1.0.0
22    ACTIVE      org.apache.ws.commons.schema_1.4.2
23    RESOLVED    wstx-asl_3.2.4
             Master=15
24    ACTIVE      jaxen_1.1.1
25    ACTIVE      dom4j_1.6.1
26    ACTIVE      org.apache.geronimo.specs.geronimo-javamail_1.4_spec_1.2.0
27    ACTIVE      org.apache.geronimo.specs.geronimo-activation_1.1_spec_1.0.1
28    ACTIVE      axiom_1.0.0.SNAPSHOT
29    ACTIVE      org.apache.axis2.osgi_1.0.0.SNAPSHOT
30    ACTIVE      simple.version_1.0.0

osgi>
```

Now install and start “simple.module” bundle.

```
osgi> install file:simple.module_1.0.0.jar
Bundle id is 31


osgi> start 31


osgi> Simple module init
[Axis2/OSGi] Starting any modules in Bundle - simple.module
Simple module engage notify
Simple module engage notify
[Axis2/OSGi] Deployed axis2 service group:Version in Bundle: simple.version



osgi>
```

8. Go to your favorite browser and type following urls to obtain the results.

```
http://localhost:8080/services/Version?wsdl
http://localhost:8080/services/Version?wsdl2
http://localhost:8080/services/Version/getVersion
```

9. On the console you will be able to observe

```
osgi> Handler1 invoked
Handler2 invoked
```

Reference Links:

[1]. https://svn.apache.org/repos/asf/webservices/axis2/trunk/java

[2]. https://svn.apache.org/repos/asf/webservices/axis2/scratch/java/saminda/osgi_test

[3]. http://download.eclipse.org/eclipse/equinox/

[4]. https://svn.apache.org/repos/asf/webservices/commons/trunk/modules/axiom