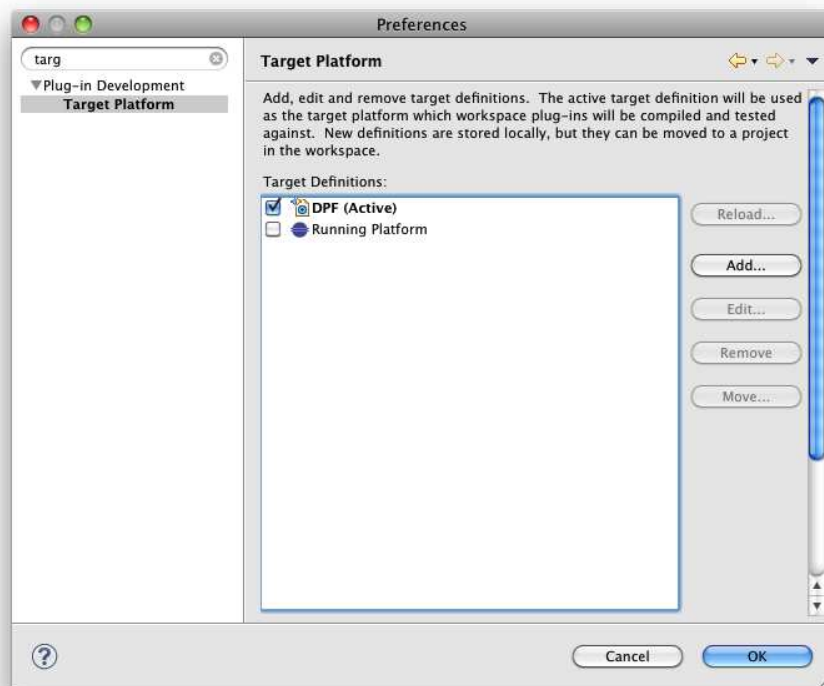


OSGi and Design Patterns

Task 1: The OSGi Framework

Download Eclipse Classic 3.5.1 from <http://www.eclipse.org> and install ObjectTeams (OT) via the update manager of Eclipse. More information on installing OT can be found on the OT download site.¹

Read the tutorial of Lars Vogel² and download the exercise Equinox runtime from the course website. You have to register the target runtime in the preferences as depicted in the following screenshot.



1a)

Now create a new OT Plug-in Project `dpf.osgi` for the OSGi framework Equinox. Make sure you add `org.eclipse.core.runtime` to the plug-in dependencies.

Create the following `plugin.xml` in the plug-in's root folder.

¹<http://objectteams.org/distrib/install.1.3.html>

²<http://www.vogella.de/articles/OSGi/article.html>

```

<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
  <extension
    id="app"
    point="org.eclipse.core.runtime.applications">

    <application
      cardinality="singleton-global"
      thread="any"
      visible="true">
      <run
        class="dpf.osgi.Apparat">
      </run>
    </application>
  </extension>
</plugin>

```

Implement an application using the following code.

```

import org.eclipse.equinox.app.IApplication;
import org.eclipse.equinox.app.IApplicationContext;

public class Apparat implements IApplication {

    @Override
    public Object start(IApplicationContext context) throws Exception {
        Person hans = new Person("Hans");
        Person karl = new Person("Karl");

        hans.identify();
        karl.identify();

        return null;
    }

    @Override
    public void stop() { }
}

```

Now create a second OT Plug-in Project `dpf.osgi.base` without generating an Activator and which only consists of the `Person` class outlined below.

```

public class Person {

    public String name;

    public Person(String name) {
        this.name = name;
    }

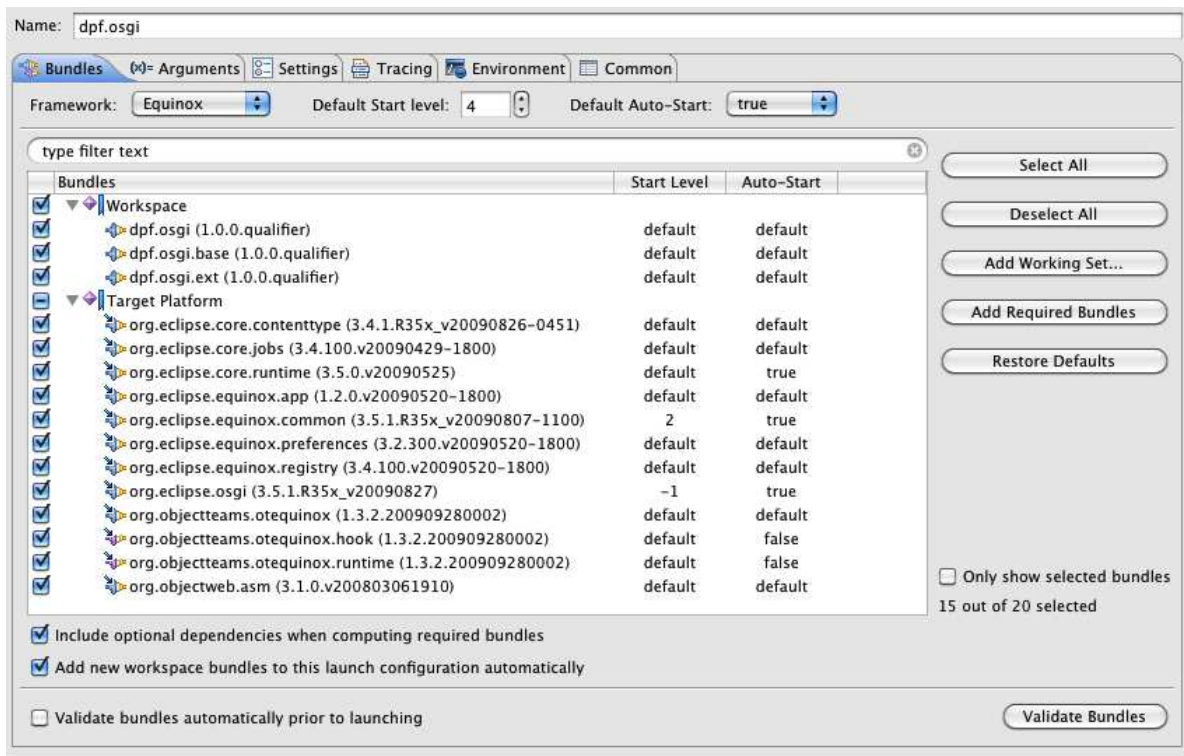
    public String getName() {
        return name;
    }

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

    public void identify() {
        System.out.println(name);
    }
}

```

Now create a new OSGi run configuration, make sure that all bundles are activated and the `Enable OT/Equinox` setting is enabled.



Run the application. The commands `help`, `apps` and `startApp` on the OSGi console should get you started.

Additional hints can be found at ³ and ⁴.

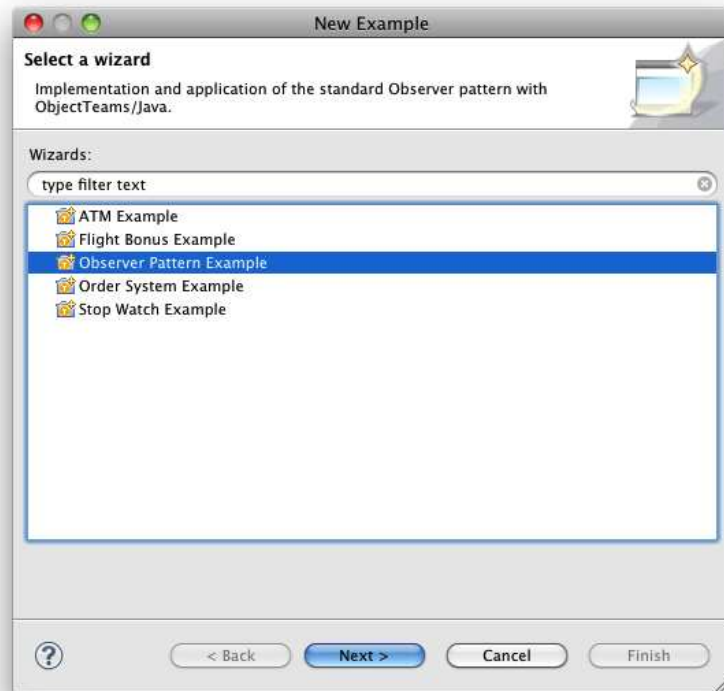
Task 2: The OSGi Framework

2a)

Try out and understand the ObjectTeams Observer pattern example which can be found under File ⇒ New... ⇒ Examples...

³<http://trac.objectteams.org/ot/wiki/OtEquinox/HowtoCreateOTEquinoxAppFromOTApp> Section 1

⁴<http://trac.objectteams.org/ot/wiki/OtEquinox>



The ObjectTeams Language Reference⁵ provides detailed explanation on roles in ObjectTeams.

2b)

In this subtask we are going to extend the example of Task 1. Create an ObjectTeams Equinox Plug-in project `dpf.osgi.ext` (without generating an Activator).

Import the exported package of `dpf.osgi.base` at `dpf.osgi.ext` and the packages of `dpf.osgi.base` and `dpf.osgi.ext` at `dpf.osgi`.

Now integrate the Team below in the new plug-in and change the implementation of `IApplication` in the `dpf.osgi` plug-in.

```
public team class University {

    public void register(Person as Student ersti, int matrikel) {
        ersti.matrikel = matrikel;
    }

    public class Student playedBy Person {

        studentIdentify <- before identify;

        public int matrikel;

        public void studentIdentify() {
            System.out.println("Matrikel: "+matrikel);
        }
    }
}
```

⁵<http://objectteams.org/def/1.2/OTJLDv1.2-current.pdf>

```

import org.eclipse.equinox.app.IApplication;
import org.eclipse.equinox.app.IApplicationContext;

public class Apparat implements IApplication {

    @Override
    public Object start(IApplicationContext context) throws Exception {
        Person hans = new Person("Hans");
        Person karl = new Person("Karl");

        System.out.println("--No context--");
        hans.identify();
        karl.identify();

        University u = new University();
        u.activate();

        u.register(hans, 123);
        u.register(karl, 345);

        System.out.println("--Uni active-----");
        hans.identify();
        karl.identify();

        u.deactivate();

        System.out.println("--Uni inactive---");
        hans.identify();
        karl.identify();

        return null;
    }

    @Override
    public void stop() { }
}

```

To start this application some changes need to be integrated in the Eclipse run configuration.

First, register the extension point `org.objectteams.otequinox.aspectBindings` in the `plugin.xml`.⁶ It should look like this:

```

<extension point="org.objectteams.otequinox.aspectBindings">
  <aspectBinding
    icon="platform:/plugin/org.objectteams.otdt.ui/icons/ot/calloutbinding_obj.gif">
    <basePlugin
      icon="platform:/plugin/org.eclipse.pde.ui/icons/obj16/plugin_obj.gif"
      id="dpf.osgi.base">
    </basePlugin>
    <team
      activation="ALL_THREADS"
      class="dpf.osgi.ext.University"
      icon="platform:/plugin/org.objectteams.otdt.ui/icons/ot/team_obj.gif">
    </team>
  </aspectBinding>
</extension>

```

Additionally, add these parameters to the VM arguments:

```

-Declipse.ignoreApp=true
-Dosgi.noShutdown=true
-XX:+UnlockDiagnosticVMOptions
-XX:+UnsyncloadClass
-Dot.equinox=0
-Dosgi.hook.configurators.include=org.objectteams.eclipse.transformer.hook.HookConfigurator
-Dosgi.framework.extensions=org.objectteams.eclipse.transformer.hook

```

⁶<http://trac.objectteams.org/ot/wiki/OtEquinox/AspectBinding>

2c)

Extend the previous solutions with the role `professor`. A professor can be identified through his group membership.

```
---Uni inactive---  
...  
---Uni active-----  
Matrikel: 123  
Hans  
Matrikel: 345  
Anja  
Peter is professor of the database group  
---Uni inactive---  
...
```

2d)

Extend the solution with lectures, which are held by professors and attended by students. Implement a method which returns all lectures attended by a student.