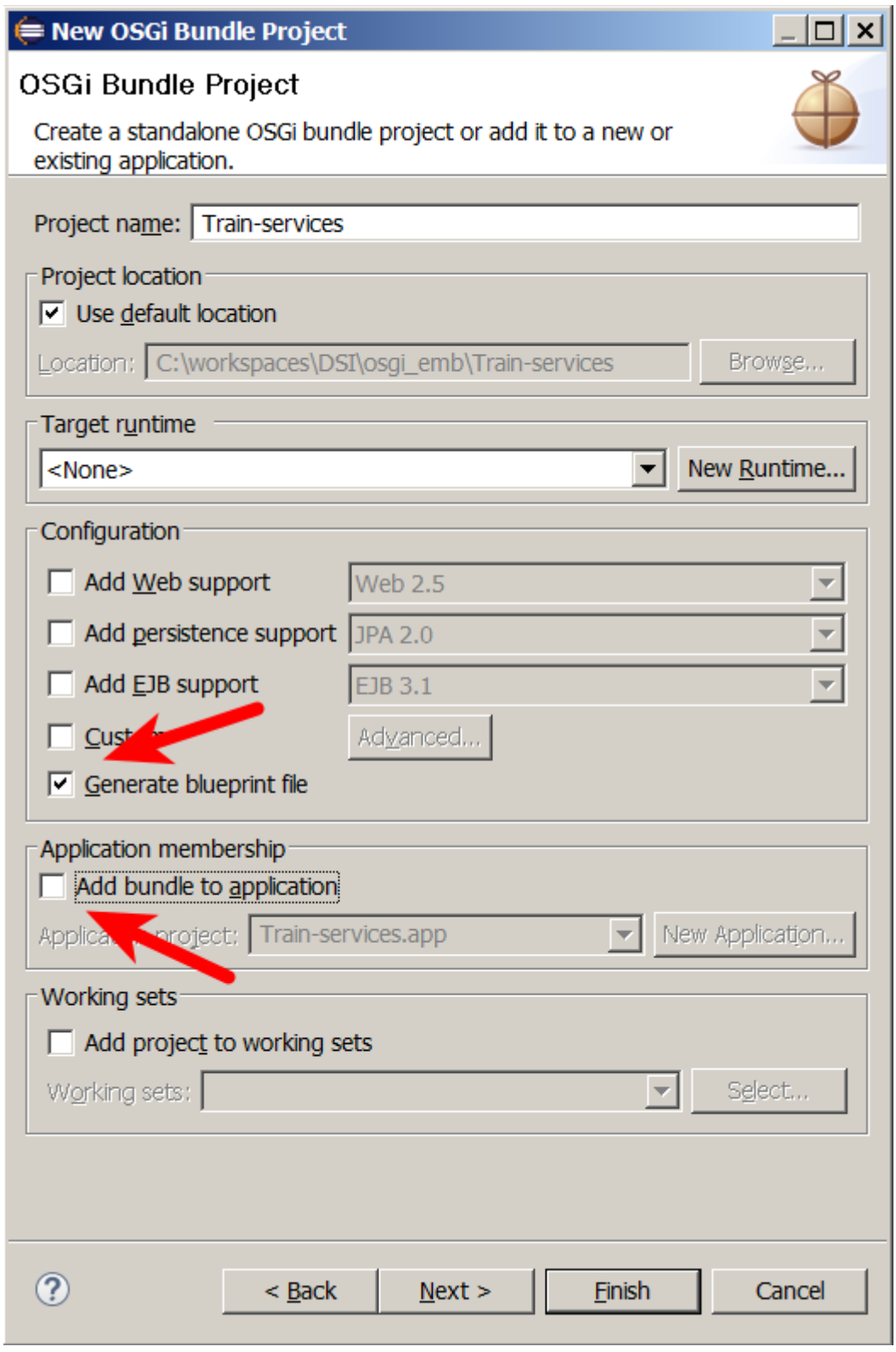
OSGi in 8.8

Creating OSGi bundles

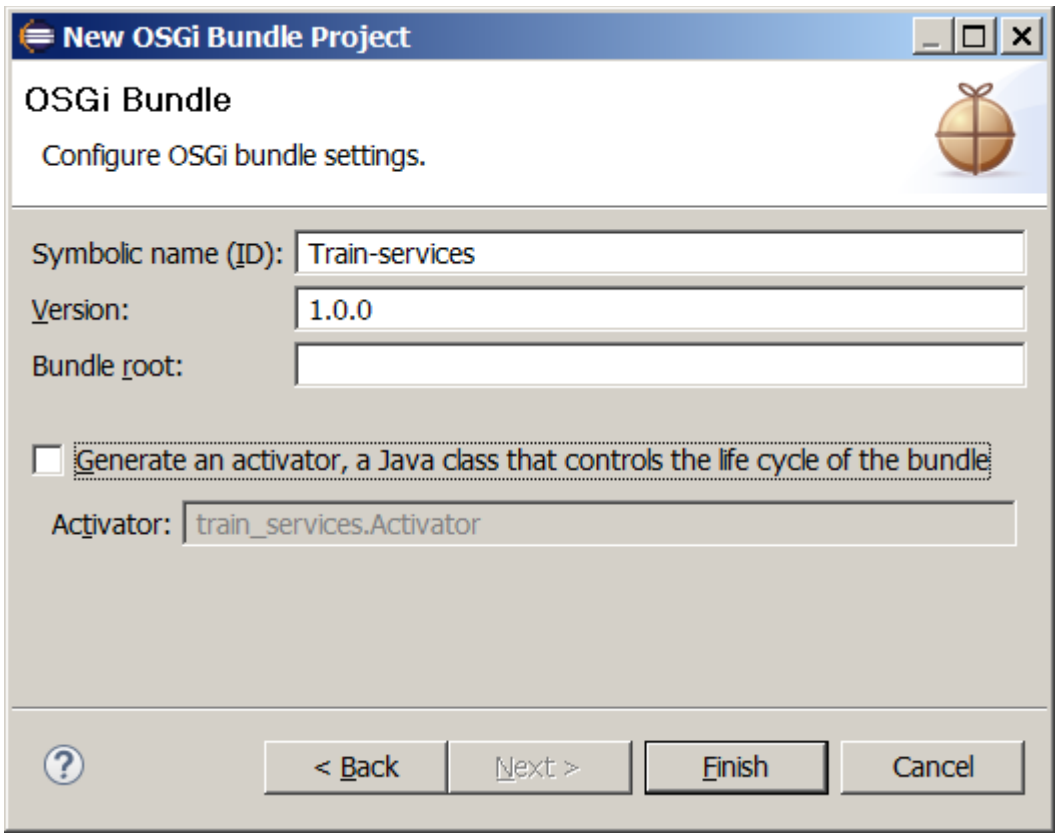
The first step of integrating an OSGi service is to create an OSGi bundle in Eclipse. You then create a Java™ interface and a Java class implementation, and you define the dependencies within the service.

Procedure

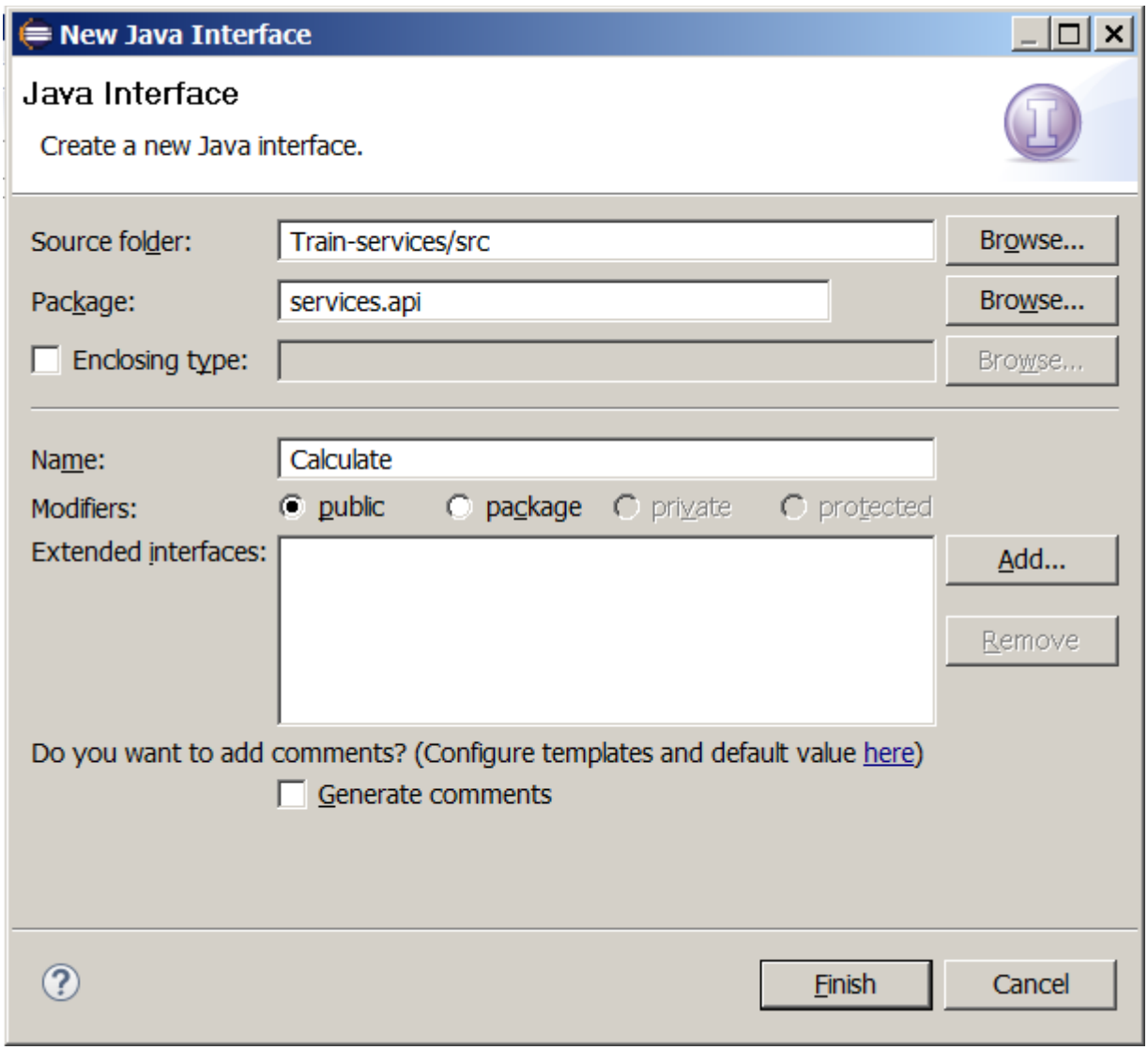
1. Click **File > New > Other > OSGi > OSGi Bundle Project**, and then click Next.
2. In the wizard, specify the name of your project.
3. Clear the **Add bundle to application** option, and click **Next** twice.



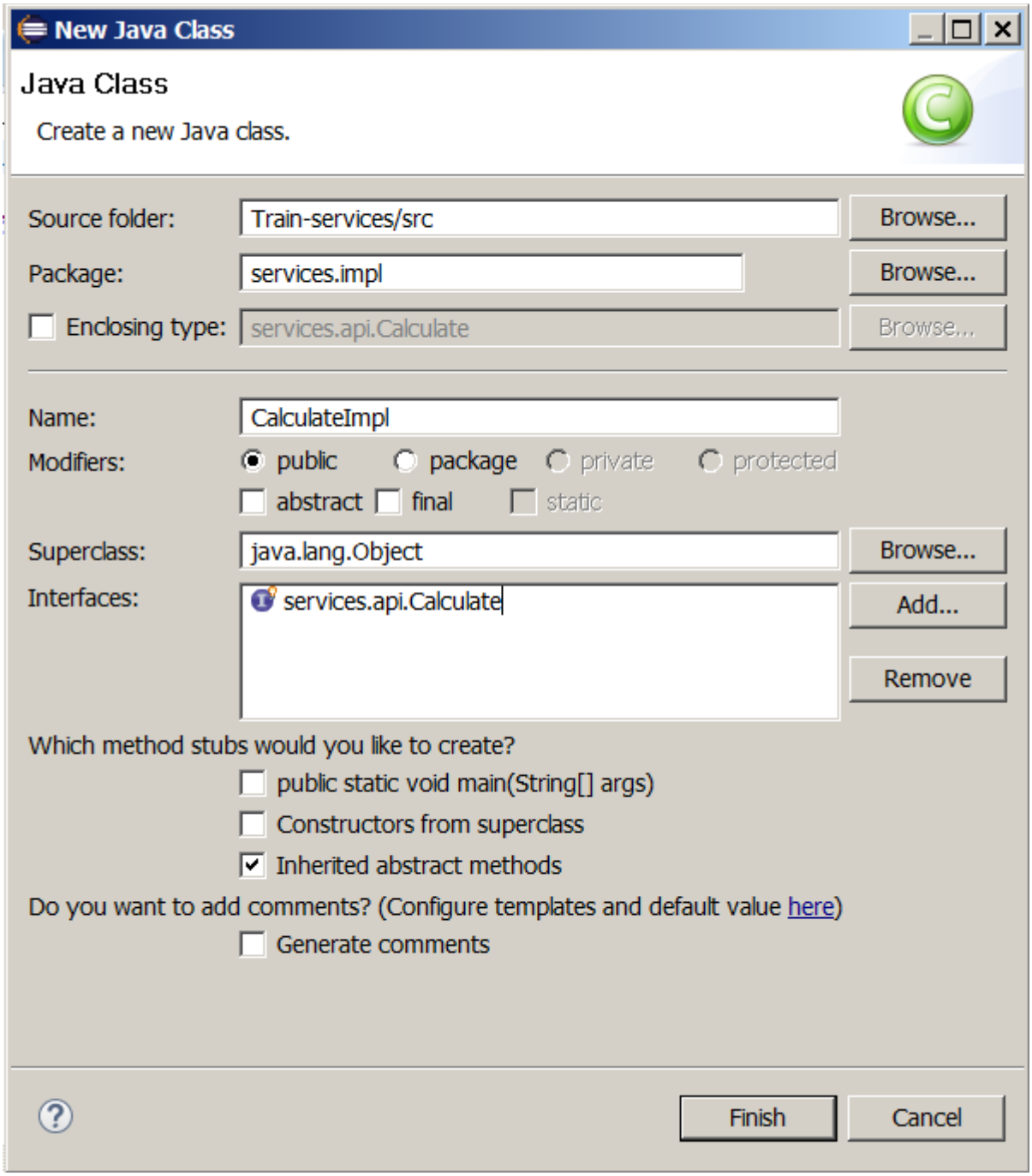
4. To build and deploy the service outside of the Plug-in Development Environment build, **remove** .qualifier from the version name, and **check** the Generate an activator option.
5. Clear **Bundle root**



6. Click **Finish**.
7. Open blueprint.xml
8. Click Add, select Service then OK
9. Click **New** for Service Interface and follow the wizard to define a new inteface for the service



10. Click **New** for Bean Reference, then again **New** for Bean Class. Define the implementation class and add the previously defined interface.



11. Click **OK**, then **OK** to complete the definition. **Close and save t**he blueprint.xml

12. Open the MANIFEST,MF file

13. In Dependencies tab, **add packages** needed in your service. If you need reference the Solution Business Object Model add here (e.g. train)

14. In Runtime tab, in the export section click Add.. to add the packages that contains the service interface. (e.g. services.api)

15. Add the Java logic.

Prepare the OSGi build

Since this OSGi service is linked to the solution model, it should be embedded to the Solution build.

In order to ensure that the project is correctly built, you need to change the build properties:

1. Open build.properties file and change the content with the following text:

```
source.. = src/
output.. = bin/
bin.includes = META-INF/,\
               .,\
               OSGI-INF/
```

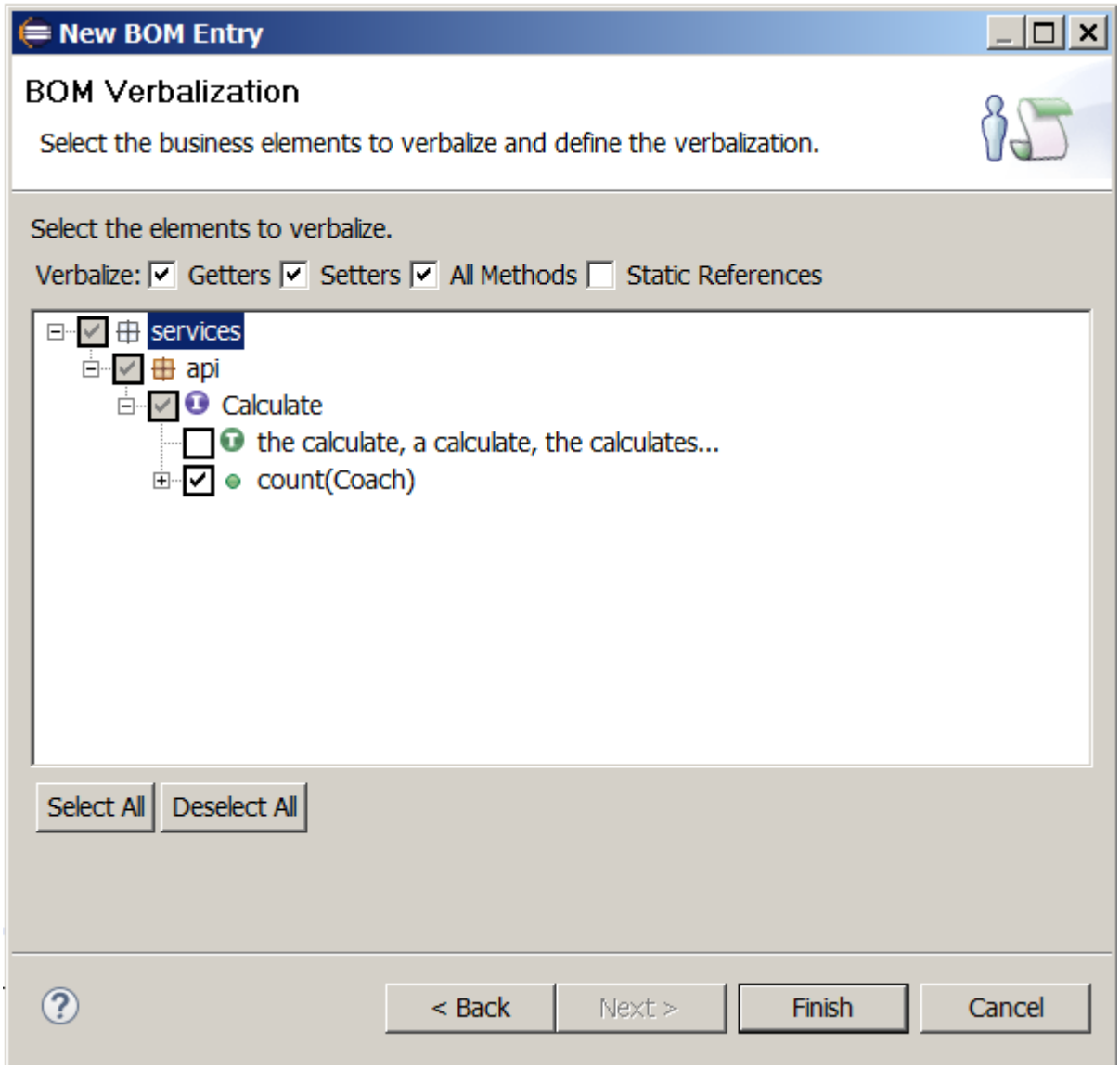
Consuming the OSGi services

Create a rule project

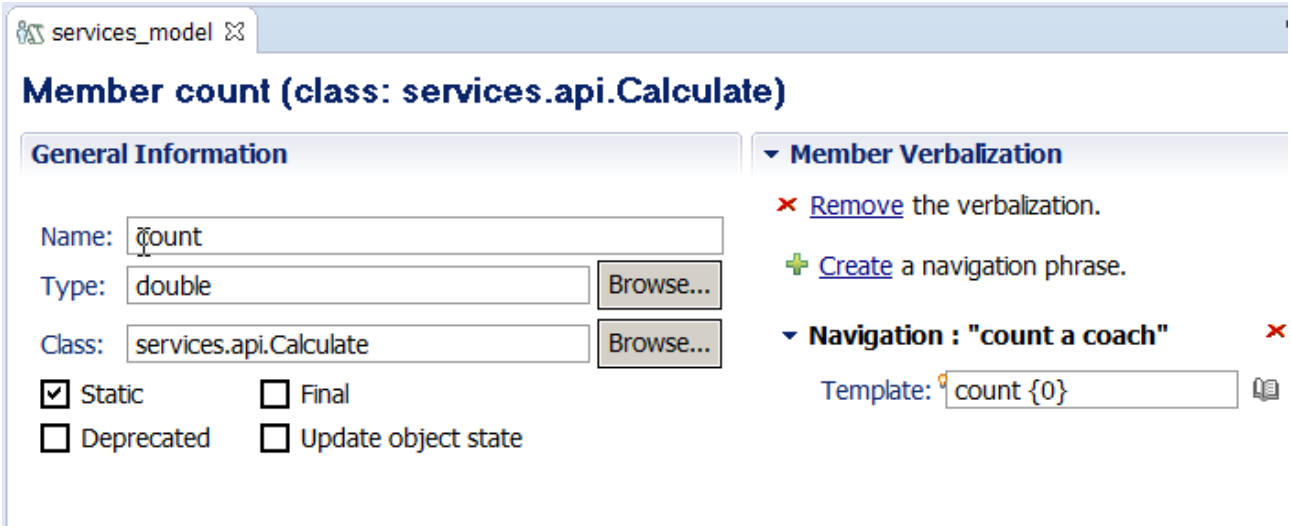
1. Click **File > New > Other > Insight Designer > Rule Project**.
2. Follow the wizard to create a standard rule project.
3. On the Rule Project References page, add the .
4. On the **Rule Project XOM Settings** page, select the Java project that contains your service, and click Finish. This setting is used at export time to check the validity of the OSGi service API.
5. In the Solution Explorer, right-click the new rule project and click Properties.
6. Click Rule Engine, select Decision Engine, and click OK. (The Classic Rule Engine is not supported in Decision Server Insights.)

Create a BOM entry

1. Right-click the new rule project, and click New > Other > Insight Designer > BOM Entry, and then click Next.
2. Enter a name for the BOM entry, select **Create a BOM entry from a XOM**
3. Browse XOM and select the services project.
4. Select the interface then **Next**.
5. Select all methods and remove the term verbalization



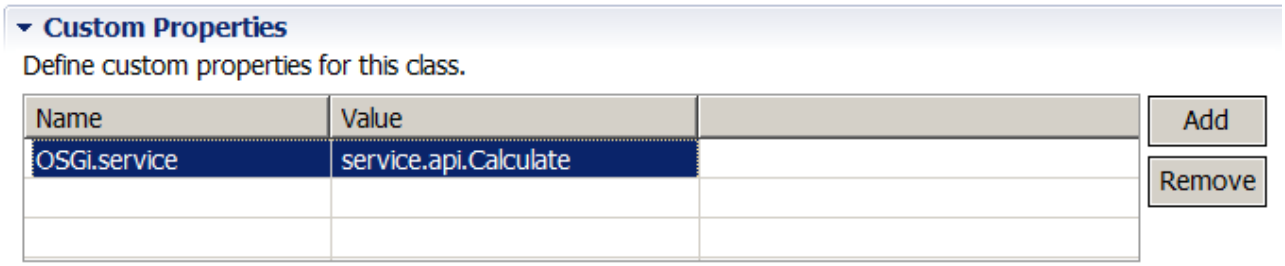
6. Click **Finish** and Open the BOM Entry to perform some adjustment:
    1. Open the method make it **static** and choose the preferred verbalization.



    2. In the **Class** tab, expand the Custom Properties section and click Add. You might have to

scroll down in the BOM editor to view the Custom properties section.

3. Enter **OSGi.service** as the name of the property, and the fully qualified name of the Java type for your OSGi service as the value. You can also add an **OSGi.service.version** property to refer to a specific version of the service. If you don't specify a version, the most recent version of the service available on the server is used.



7. Save the changes.
8. In the rule agent project, add a reference to the rule project that contains the service BOM:
    1. Right-click the agent project, and click Properties.
    2. Click Project References, and select the rule project where you created the service BOM. Click OK.
9. **Add the Project Reference t**o the services project (not the BOM) in the solution