STEPS

This document will talk about downloading the source code, install them and also talk about how to use our tool to develop a supporting dynamic update tuscany application. If you just want to run our example illustrated in our example, please reference ReadMe in distribution folder.

## Import and run our examples in eclipse

1. Checkout from svn

In terminal, type “svn checkout <https://conup.googlecode.com/svn/tags/2.0-DU/>”, download all the modules , samples and distribution.

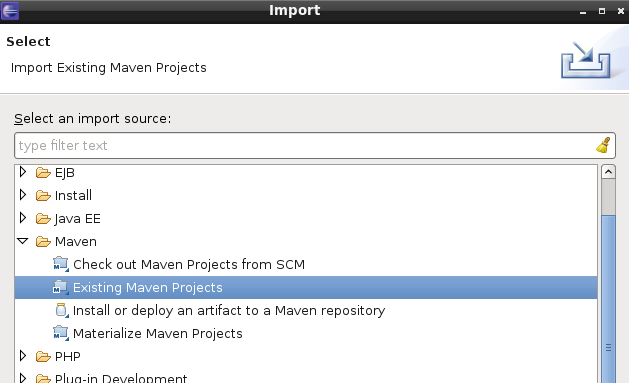
1. Compile and install modules and samples

Change directory to the folder where you checkout source code, we need to install all these modules and samples into local repository

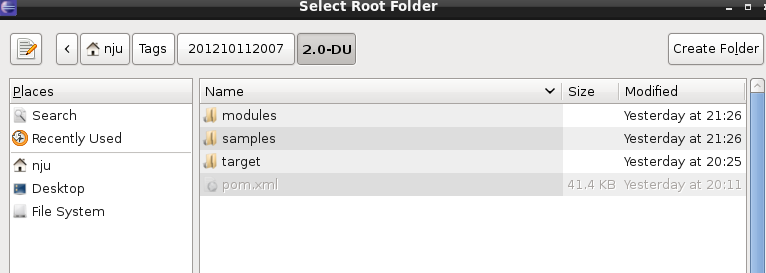
Please run the following command in your terminal: mvn install

1. Import into eclipse

After we have finished previous work, we can import all these modules and samples into eclipse.

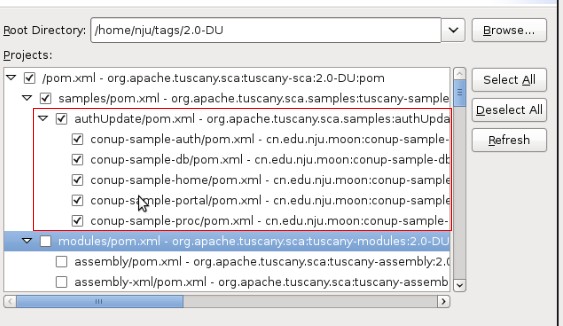


Select root folder.

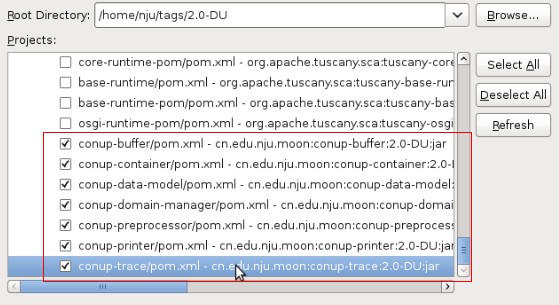


After searching, we should select the following projects into eclipse:

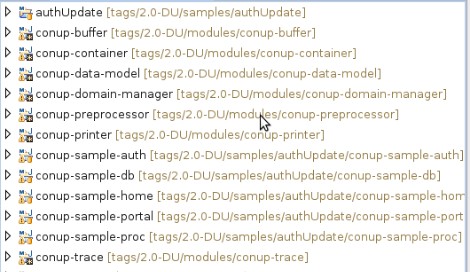
* Projects in samples folder:



* All conup-\* projects:



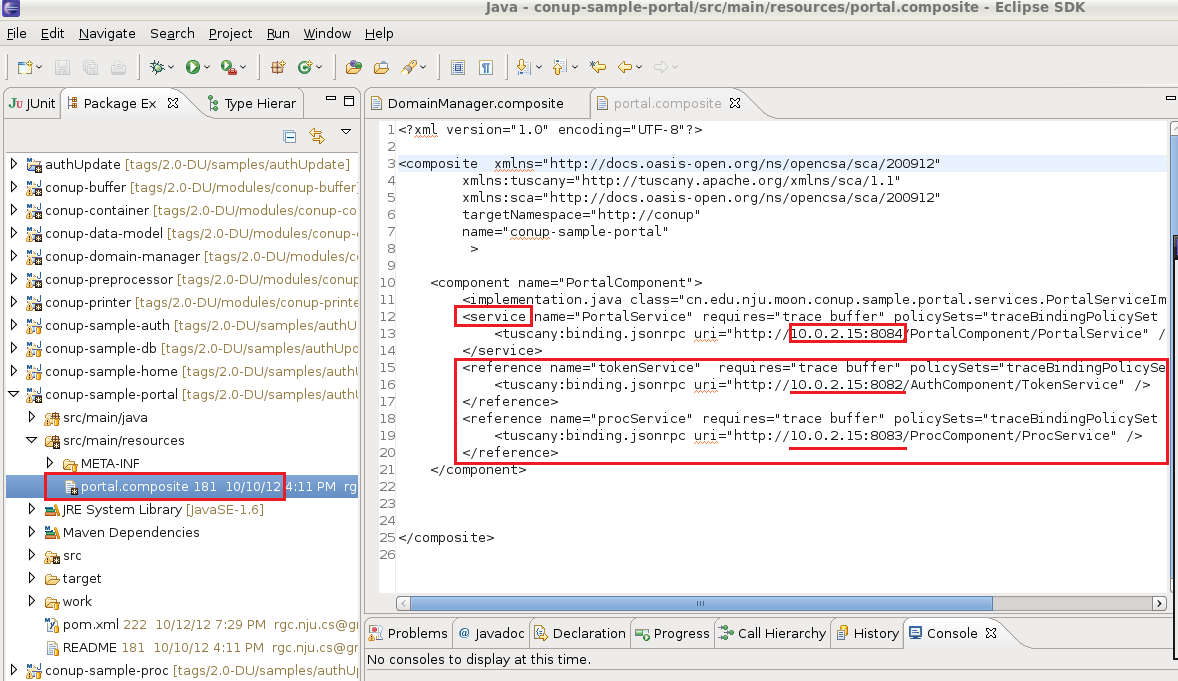
After you finished importing, all these imported projects will be displayed as the following figure:



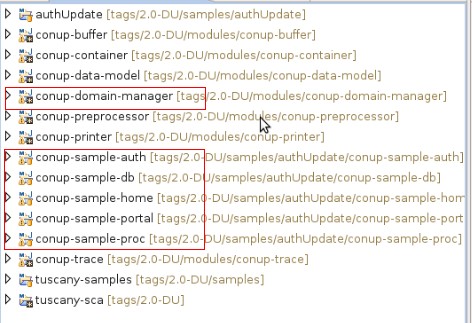
Here you have to pay attention, in order to simplify our example, we assign all the service binding to the specific ip address(10.0.2.15). If your environment satisfy this requirement, you can skip the following steps to launch nodes(step 4).

If you want to run our example in different computers, you should manually change the ip address like the following figure:

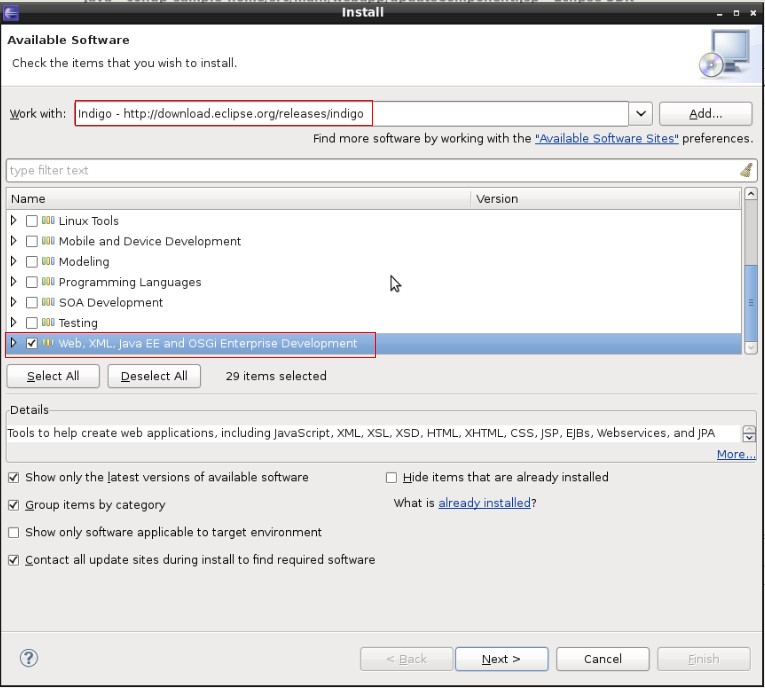
* Open the composite file, here we use conup-sample-portal as an example. So we need to modify portal.composite file
* In service tag, we need to change 10.0.2.15:8084 to PortalComponent host ip. (If your portalService’s host pc’s ip is 114.212.85.3, you should change the binding uri to 114.212.85.3:8084)
* In reference tags, for example in PortalComponent, it will reference TokenService from AuthComponent and ProcService from PortalComponent. We need to change them to their specific ip. (If AuthComponent’s TokenService is running in host pc with ip: 114.212.85.2:8082, you should change reference tag named with tokenService 10.0.2.15:8082-->114.212.85.2:8082. You should change another reference tag’s ip and port to its service provider’s binding)



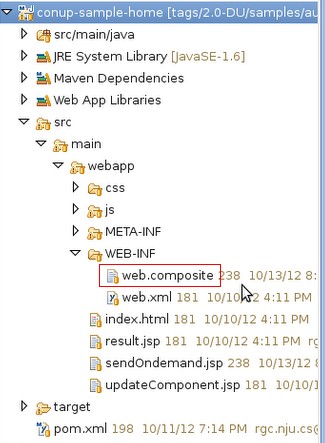
* You should change the ip in the red box to their host computer’s ip.



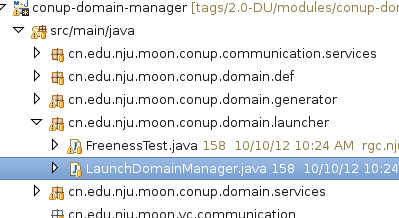
* If you want to run conup-sample-home, you should install the plugin(WEB, XML, Java EE and OSGi Enterprise Development). You can find it in default available site, if your eclipse version is 3.7 indigo, you can select it from the list.



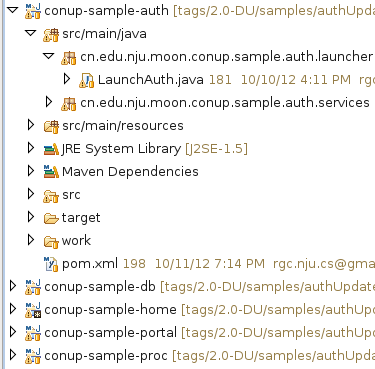
When you want to access service by using conup-sample-home project, you should change the reference in web.composite in this project



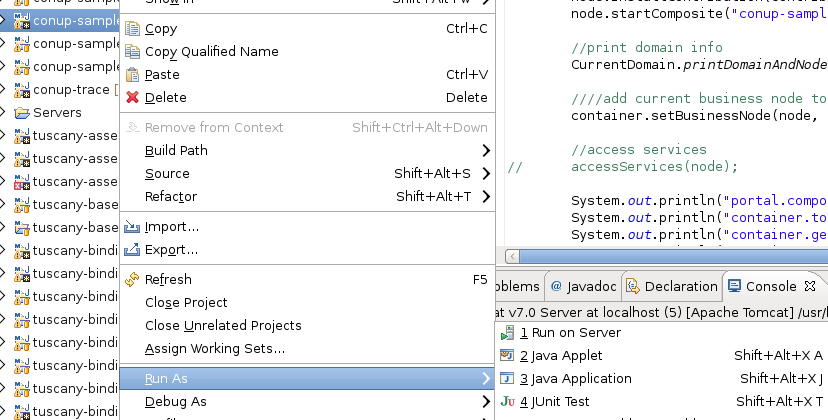
1. Launch nodes
2. First we need to launch domain manager



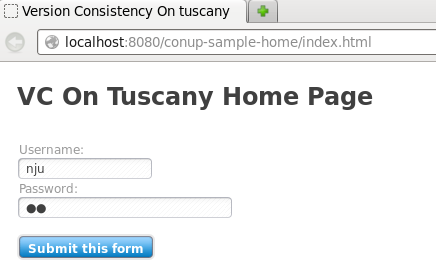
1. We launch all other four node in eclipse.



1. After we finish starting all four nodes, we need to start conup-sample-home web application to visit all these service.



And visit in your browser, input username: nju, password: cs.

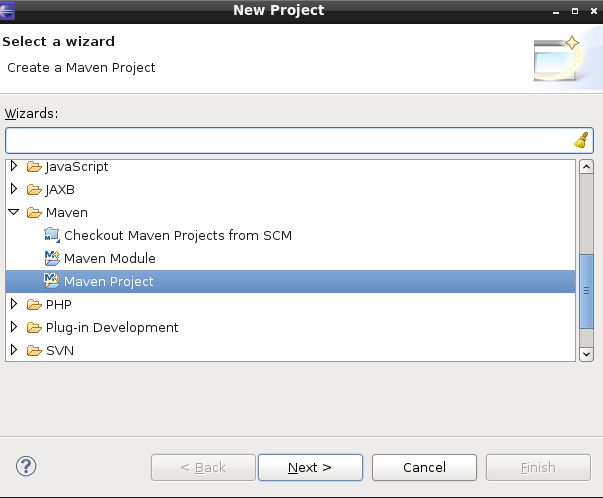
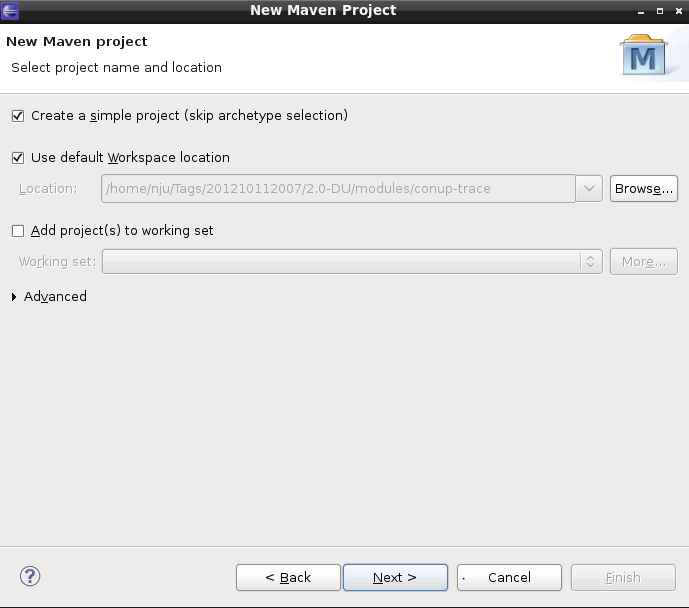


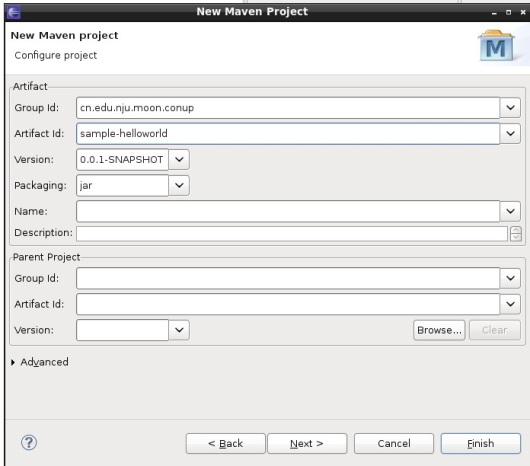
This is just a demo, you can use samples in distribution folder where you can send ondemandSetUp, Update, and query currentStatus etc.

## Develop an application using our tools

In this section, we will show you how to develop a Tuscany application by using our tools.

1. we need to create a new maven project

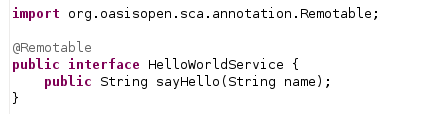
 



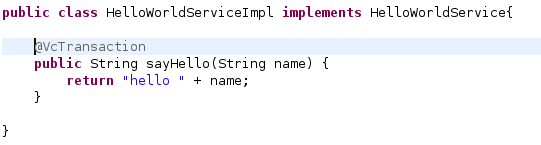
1. Create a service interface and a service implementation.

In here, you just create a service interface and service implementation like in Tuscany.

HelloworldService.java:

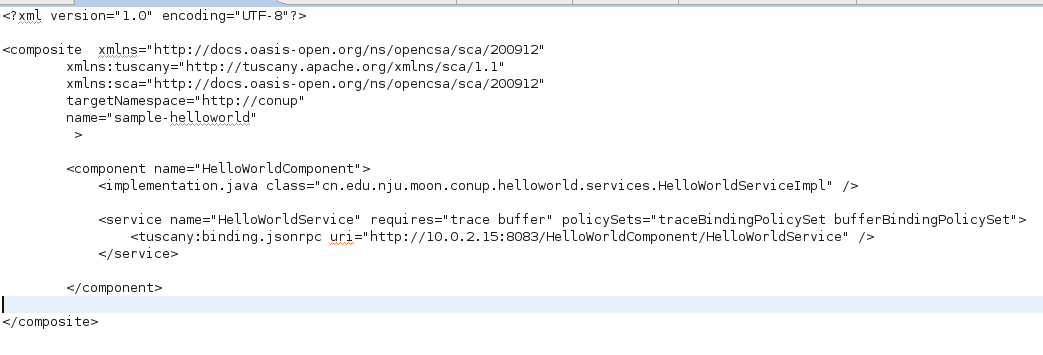


HelloworldServiceImpl.java(here we need to add VcTransaction to annotate this function as a transaction)

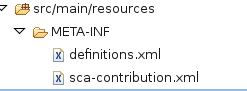


1. Create composite file

We create composite file like other Tuscany application, we will use two custom interceptors in our platform, so we need to add the intents and policySets in service definition



1. Add definition.xml and sca-contribution.xml



Because we use intents and policySets in composite file, so we need to define them in META-INF/definition.xml. In definition.xml, we define two policySets



We also need to add a sca-contribution.xml to tell Tuscany which composite will be deployed.



1. Create a launcher

Due to our conup-container has a defect, after you have finished all the steps aforementioned, you still need to create a launcher for your application, namely create a class with *public static void main(String [])*. In order to create a launcher, you can simply copy the code from class *LaunchAuth* in our conup-sample-auth which is available in our repository, and modify several variables for it. In detail, you should replace the following variables with your own:

* replace “auth.composite” with your own .composite file name;
* replace “conup-sample-auth” with your own .composite file name;
* replace “AuthComponent” with your own component’s name which is described in your .composite file.

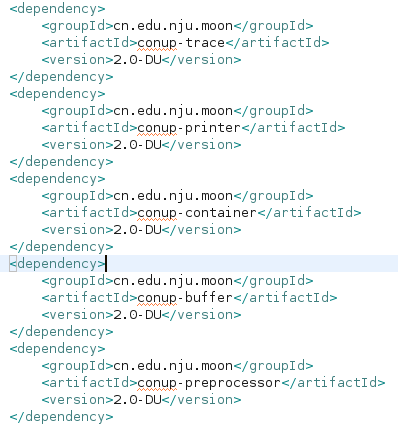
1. Deploy application

It takes several steps for you to deploy your application:

* after you have created your launcher, you MUST run it, so that our conup-container can generate a communication module for your application which is located in conup-sample-container directory
* run click on your conup-container project, and execute “mvn package”. (Attention: don’t execute “mvn clean”, otherwise the communication module you just generated will be deleted)
* find the conup-container-\*.jar which is under target/ directory of the conup-container project, and rename it to conup-container.jar
* copy the conup-container.jar to the directory ${TUSCANY\_HOME}/modules/conup-container/
* right click on your own project in eclipse, and execute “mvn package”, the copy the .jar file under target/ to any directory where you want to place your application (Attention: you can only deploy your application in a .jar file temporally)
* run your application with shell/console.

1. Add dependencies to pom.xml

Unlike other Tuscany applications, we also need to add the following dependencies to make sure our code works fine.



These dependences which are required in Tuscany applications should also be added into pom.xml.