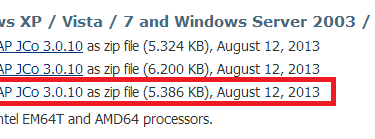
This blog can be used as a very simple tutorial for those guys who are new to JCO. It demonstrates how to download and configure the JCO in your laptop. A simple consumption from JAVA side to call a remote function module in ABAP side is done to verify the correctness of the configuration.

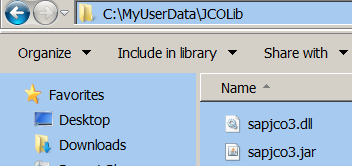
1. download [SAP](http://www.salixd.com/tag/sap) JCO3.0 from [service marketplace](http://www.salixd.com/redirect/aHR0cHM6Ly93ZWJzbXAxMDMuc2FwLWFnLmRlL35mb3JtL2hhbmRsZXI/X0FQUD0wMDIwMDY4MjUwMDAwMDAwMjY3MiZhbXA7X0VWRU5UPURJU1BMQVkmYW1wO19TQ0VOQVJJTz0wMTEwMDAzNTg3MDAwMDAwMDEyMiZhbXA7X0hJRVJfS0VZPTUwMTEwMDAzNTg3MDAwMDAxODUyNyZhbXA7X0hJRVJfS0VZPTYwMTEwMDAzNTg3MDAwMDIyNTQxNCZhbXA7X0hJRVJfS0VZPTYwMTEwMDAzNTg3MDAwMDI1NTk4MiZhbXA7)

[](http://www.salixd.com/wp-content/uploads/2016/08/1471082620-2060-clipboard1.png)

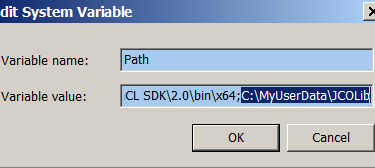
Choose the proper file according to your OS to download:

[](http://www.salixd.com/wp-content/uploads/2016/08/1471082620-1279-clipboard2.png)

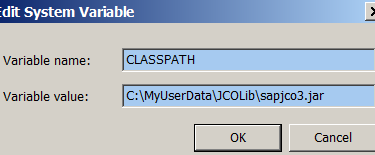
Unzip the downloaded file, create a new folder and put the dll and jar file into the folder.

[](http://www.salixd.com/wp-content/uploads/2016/08/1471082620-3853-clipboard3.png)

add your local folder path to PATH environment variable:

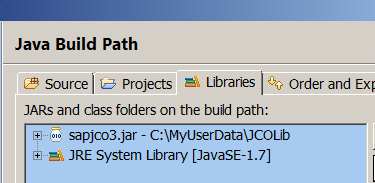
[](http://www.salixd.com/wp-content/uploads/2016/08/1471082619-5764-clipboard4.png)

maintain the absolute path for jar file as value for CLASSPATH:

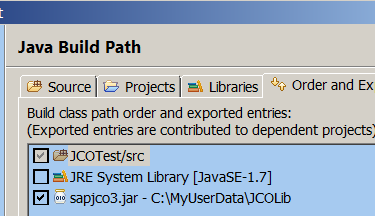
[](http://www.salixd.com/wp-content/uploads/2016/08/1471082620-8632-clipboard5.png)

Note: there is a folder javadoc after downloaded file is unzipped and it contains detailed information about JCO like installation guide, API documentation and sample code etc.

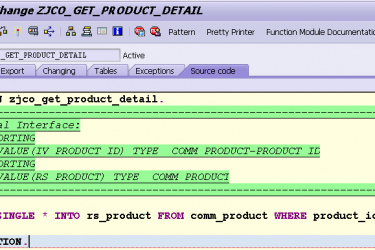
2. Create a new Java project. Right click on project and click "Property".

[](http://www.salixd.com/wp-content/uploads/2016/08/1471082620-9649-clipboard6.png)

Add the JCO jar file to the project by clicking button "Add External JARs".

[](http://www.salixd.com/wp-content/uploads/2016/08/1471082623-7650-clipboard7.png)

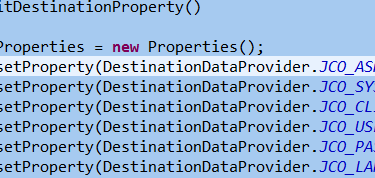
3. Create a simple remote function module in ABAP. The function module just accepts the product ID passed from JAVA side, and query against it in database table COMM\_PRODUCT, and return the query result which will be fetched via JCO in JAVA side. Never forget to mark the "Remote enabled Function module" flag.

[](http://www.salixd.com/wp-content/uploads/2016/08/1471082624-2864-clipboard9.png)

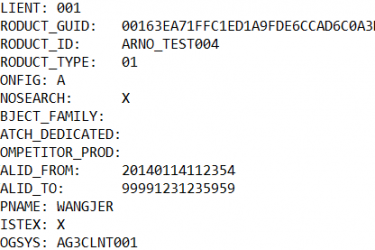
4. copy the attached JAVA source code to your project.

For demonstration purpose, the sample code just use a local file to store ABAP destination configuration which would lead to security issues and should not be used in productive code.

Just specify your own ABAP system information and your user credentials in the sample code:

[](http://www.salixd.com/wp-content/uploads/2016/08/1471082627-9581-clipboard9.png)

in the test code, we hard code the product ID as ARNO\_TEST004. Here below is the execution result, the detail information of this product is returned to JAVA side:

[](http://www.salixd.com/wp-content/uploads/2016/08/1471082623-6764-clipboard10.png)

We can compare it with the information in SE16, and they are exactly the same: