

Getting Started with the SAS® Drug Development Java API

Since you are reading this document, you've already expanded the client distribution file (sdd-java-api-client-<release>.zip) into some directory. In that directory is now an "sdd-java-api-client-<release>" subdirectory which will be referred to as *HOME*.

In your *HOME* directory you should see:

- README.txt
- docs [a directory containing javadoc and documentation on the use of the API]
- samples [a directory with sample programs using the API]
- lib [a directory of jars the API needs to run]
- Setcp.bat [a command to set your classpath in a windows command line session]

This document assumes that the server-side portion of the API has already been installed on your application server. Speak to your system administrator if this has not been done.

The API requires that you have a version 6 or greater Java JDK installed. At your command prompt you can type "java -version" to verify that you have the appropriate JDK installed on your system.

Verifying Your Client Installation

To run one of the sample programs or verify your client installation:

1. From a command prompt, go to the *HOME* directory.
2. Run the Setcp command to set your classpath.
3. Run the following command:

```
java sample.SASDrugDevCommand -help
```

You should see a list of commands that this sample program provides. This indicates the client install has been done correctly.

To verify that the remote api has been properly installed on your application server, you can run the command:

```
java sample.SASDrugDevCommand -u userid -p password -s https://yourmachinename:443 -test
```

You should see a success message indicating that the client and server portions of the API match and could both be accessed.

Finally, to verify your credentials and access to the system, you can run this command:

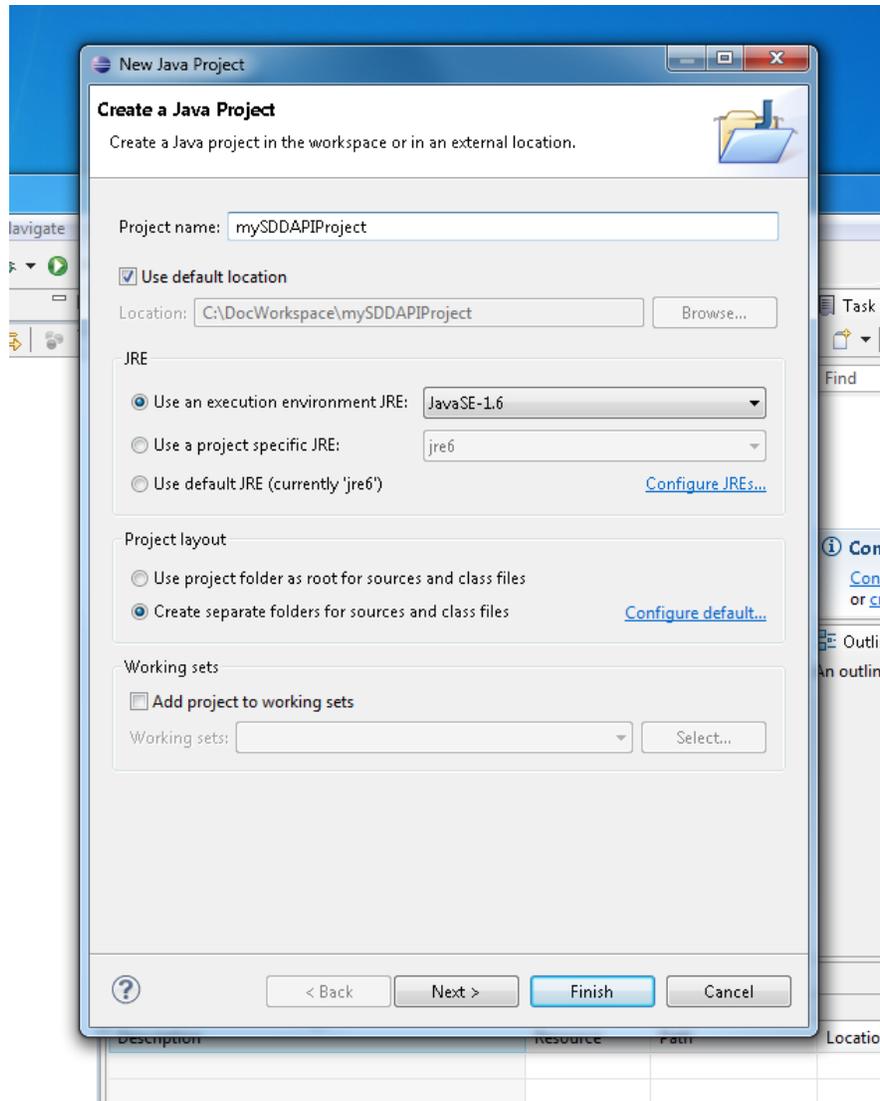
```
java sample.SASDrugDevCommand -u userid -p password -s https://yourmachinename:443 -  
repository_list / -showChildren
```

You should see a listing that includes your Organization, and a “Users” folder.

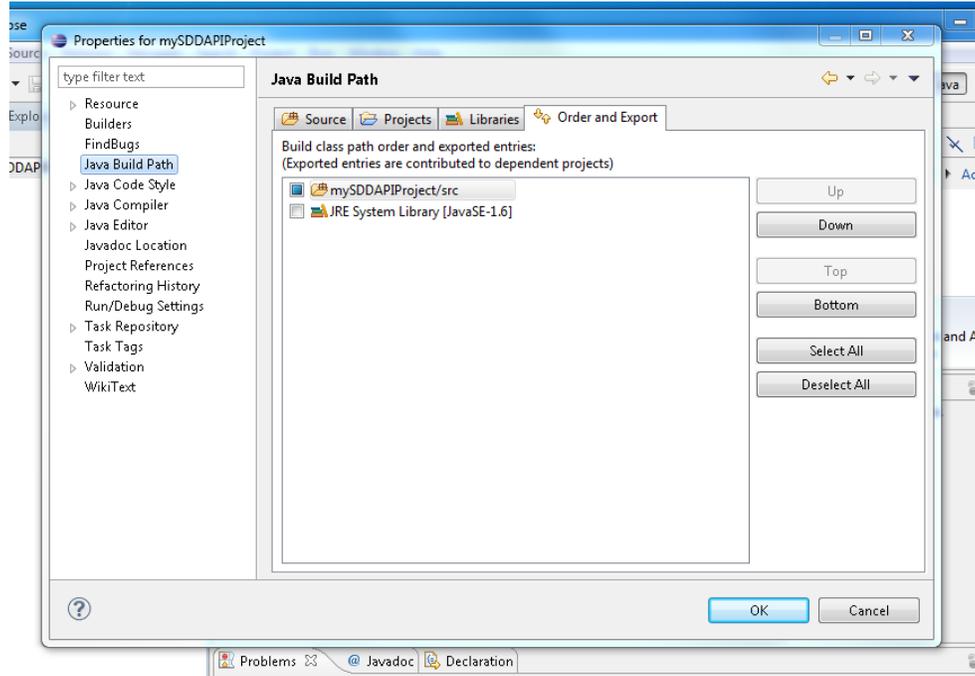
Setting up a Development Environment

The SAS Drug Development API requires the use of Java 6, so you must have a Java 6 JDK installed along with your IDE. For this document we'll assume you are using eclipse.

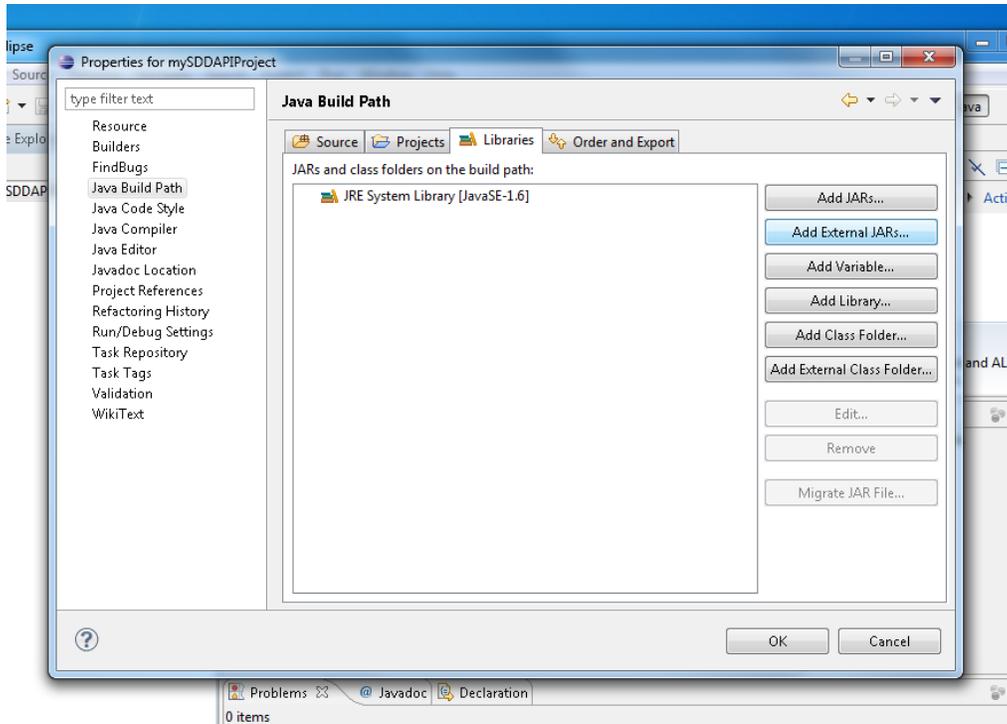
- 1) In eclipse, create a new java project, specifying a Java 6 execution environment:



2) Once the project is made, open the project properties and select the java build path setting:



3) Select the libraries tab and click the “Add External JARs...” button.



- 5) Finally you can create your main program, starting with one of the sample programs, or with something like the following:

```
import java.net.MalformedURLException;
import java.net.URL;

import com.sas.hls.client.*;
import com.sas.hls.drug.client.DrugServiceManager;
import com.sas.hls.resource.repository.RepositoryResourceService;
import com.sas.hls.resource.repository.exception.RepositoryResourceCheckoutException;
import com.sas.hls.resource.repository.exception.RepositoryResourceNotFoundException;
import com.sas.hls.security.authentication.exception.AuthenticationException;

public class MyProg {

public static void main(String[] argv) throws RepositoryResourceNotFoundException,
                                             RepositoryResourceCheckoutException {

    try {
        Session session = SessionFactory.newSession(new URL("http://yourSDDMachine.domain.com"),
                                                    "mySDDuid",
                                                    "mySDDpwd");

        RepositoryResourceService repoService = null;
        repoService = ((DrugServiceManager) session.getServiceManager()).getRepositoryResourceService();

        //some command
        repoService.checkoutFileToWorkspace("/YOURORG/Files/Folder/someProgram.sas");
        session.invalidate(); //logout

    } catch (AuthenticationException e) {
        //you typed your password wrong above
    } catch (MalformedURLException e) {
        // you typed your url wrong above
    } catch (RepositoryResourceNotFoundException e) {
        // The file you tried to check out doesn't exist
    } catch (RepositoryResourceCheckoutException e) {
        // the file you tried to check out was already checked out
    }

}

} //end main
} //end class
```

See the API documentation in your docs directory for information on the services and methods provided.

Several example programs that use the API can be found in the samples directory, including a simple command line application that exposes many of the functions in the services currently provided.