

Installation Guide of the OSS Common API Reference Implementation (JSR 144)

OSS through Java™ Initiative

Vincent Perrot, Sun Microsystems Inc.

COM-API-Ri_Installation_guide.1.4.doc

Copyright © 2002-2006 The Members of the OSS through Java™ Initiative. All Rights Reserved. Use is subject to license terms.

Sun, Sun Microsystems, the Sun Logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Executive Summary

The Common API offers interfaces and classes, which are common across all OSS API defined under OSS J Initiative. This document describes how to install and use the Reference Implementation (RI).

Table of Contents

Executive Summary.....	2
Table of Contents.....	3
Preface	4
<i>Objectives.....</i>	<i>4</i>
<i>Audience.....</i>	<i>4</i>
<i>Related Information</i>	<i>4</i>
<i>Revision History:</i>	<i>4</i>
1 Introduction.....	5
2 Extendable part.....	6
3 Examples demonstrating the 3 integration profiles	7
<i>3.1 Installation instructions</i>	<i>7</i>
3.1.1 Load the project	8
3.1.2 Configure the application server.....	8
3.1.3 Deploy the application:.....	13
3.1.4 Exercise the Web Services profile	14
3.1.4.1 Common WS profile	14
3.1.4.2 VPN Web Service (common_ex_vpn_WS project)	16
Appendix A: Glossary and References	19
<i>References</i>	<i>19</i>

Preface

Objectives

Installation and usage description of the OSS/J Common Reference Implementation

Audience

The target audiences are

- Developers who seek information about how the Common API can be implemented
- Developers of other OSS/J API Reference Implementers
- Developers who want to make use of these API and extend its implementations

Related Information

Prerequisite

« Java EE Tools Bundle Beta », containing JSE 5, JEE, netbeans 5.5, and OpenESB. You can download it at:

<http://java.sun.com/javaee/downloads/index.jsp>



Revision History:

Date	Version	Author	State	Comments
September 2006	1.4	Vincent Perrot Sun Microsystems Inc	Maintenance Release 4	Change all the content according to the new RI design

1 Introduction

This document describe how to install, run and verify the OSS Common API Reference implementation (RI).

The RI is divided into two main parts:

- The extendable implementation of CBE and based objects
- The example demonstrating the usage and extension of the API for the three integration profiles.

In all the following

- the given paths are relative to the **<installation path>\oss_common_j2eesdk-1_4-src-ri**.
- the operating system is Windows XP
- the application server bin directory is in the execution path of the environment

2 Extendable part

This part of the RI is designed to be reused as much as possible into the other OSS through Java (OSS/J) APIs.

In the src directory you will find the java code of the classes and a build .xml file that allow to compile and create the java archive of the Common API implementation.

This part of the RI the JavaEE or J2EE SDK to be installed.

Sun Java System Application Server Platform Edition 9 is recommended.

The common java archives of the RI are already delivered compiled into the lib directory (one jar per package name).

In order to re-compile and re-create the jar the following command could be executed:

```
> cd src\ri  
> asant
```

The newly recompiled archives are placed in the lib directory.

3 Examples demonstrating the 3 integration profiles

The part of the RI is delivered as Netbeans projects starting with “common_ex” prefix.

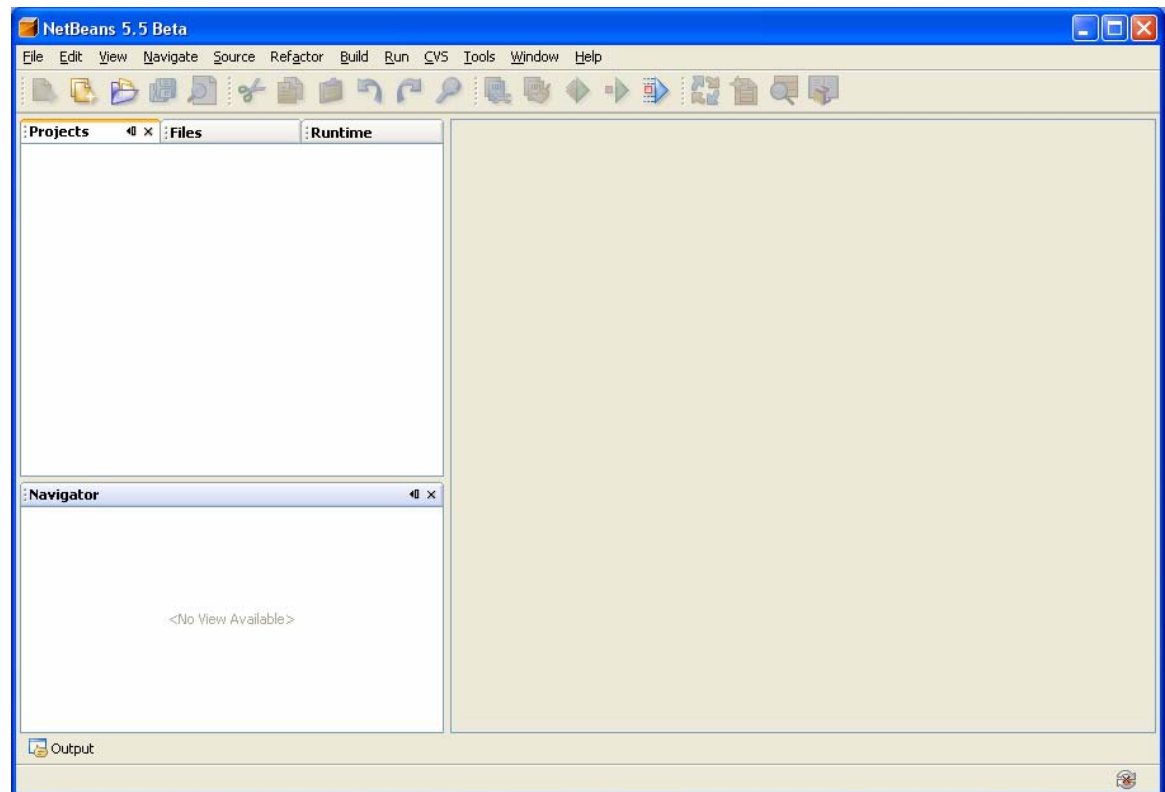
There are 4 projects:

- common_ex contains the deployable application composed of 3 EJB modules (using the NetBeans vocabulary).
- common_ex_EJB contains the Java and XML profiles composed of several EJB
- common_ex_WS contains the Web Services (WS) representing the Common WS integration profile
- common_ex_vpn_WS contains the WS representing a simplified vpn service extending the Common interface.

3.1 Installation instructions

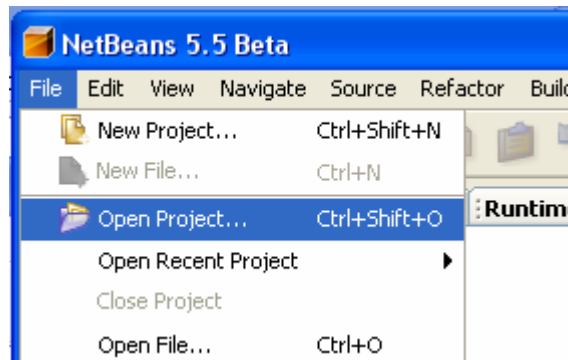
All the installation steps are performed using NetBeans.

Start Netbeans, the following window appears:

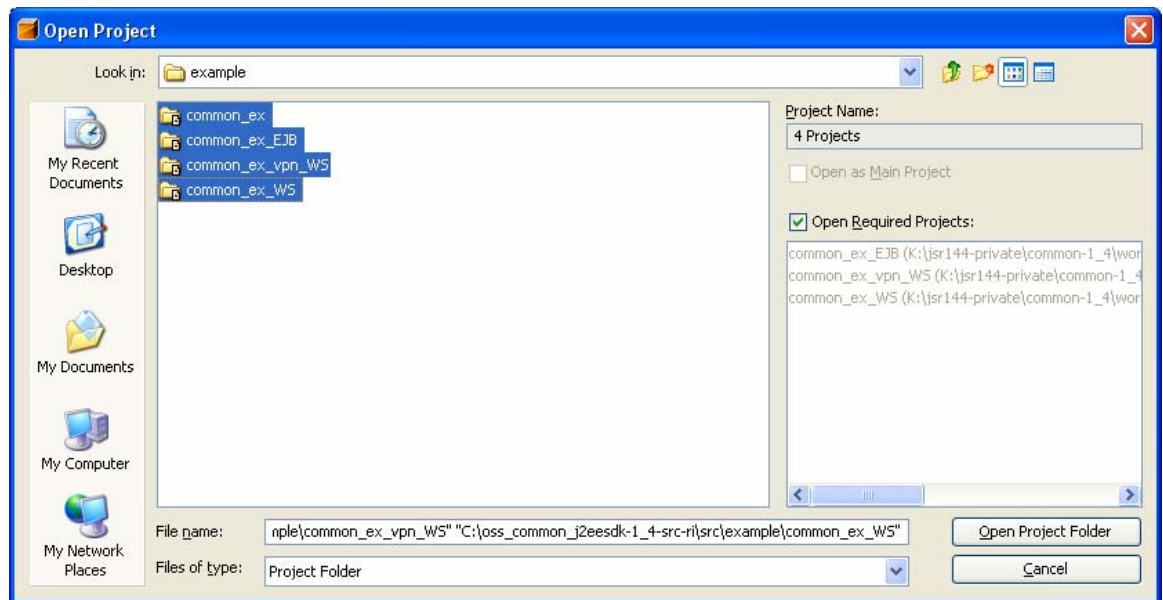


3.1.1 Load the project

Open all the common_ex* projects from the src\example directory



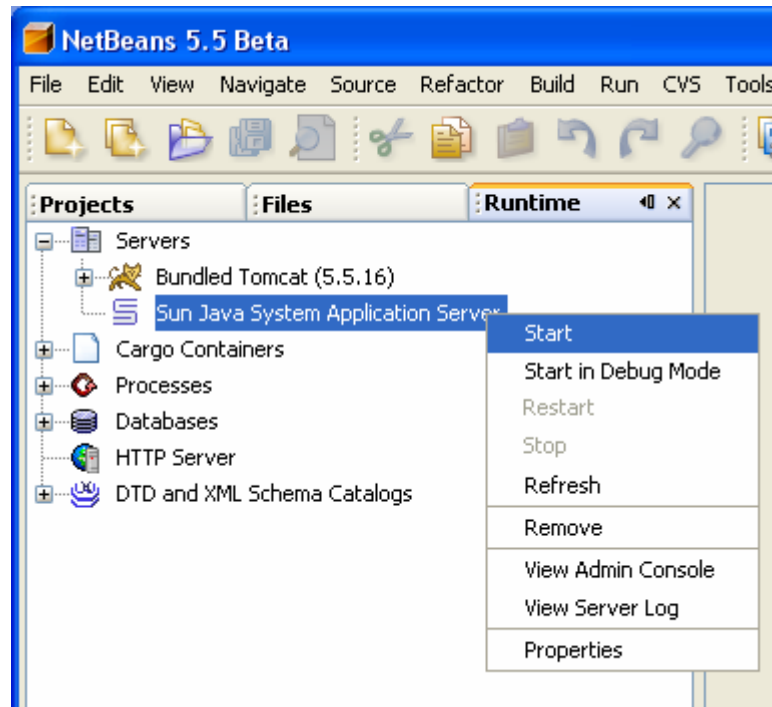
In the wizard window navigate until the example directory and selection the 4 projects:



Solve any reference when missing.

3.1.2 Configure the application server

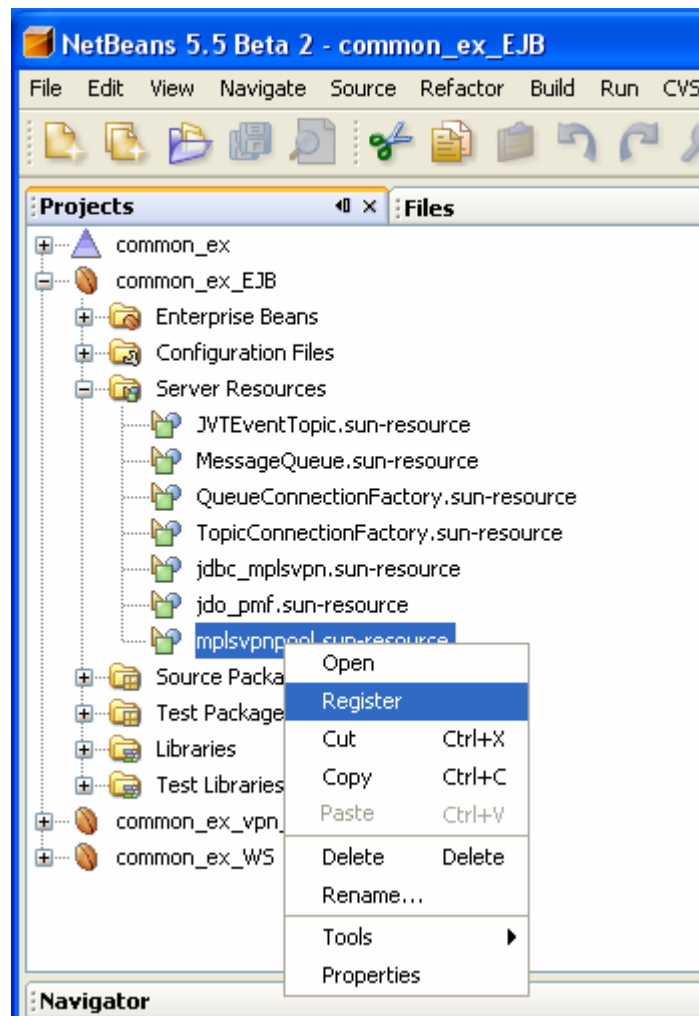
Start the application server:



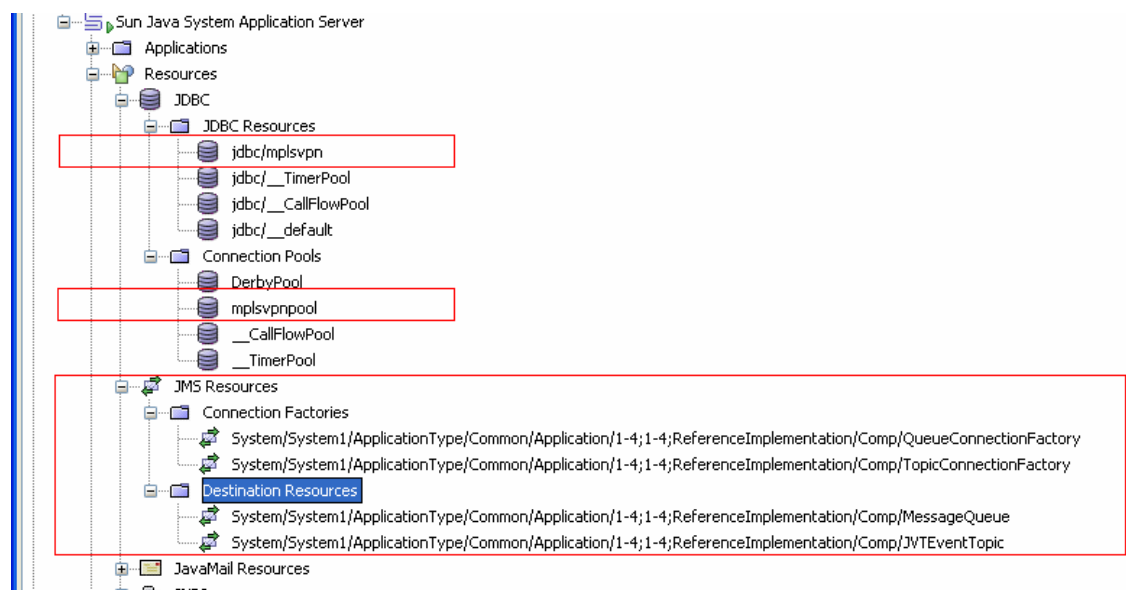
Once started a green arrow appears in front of the name:



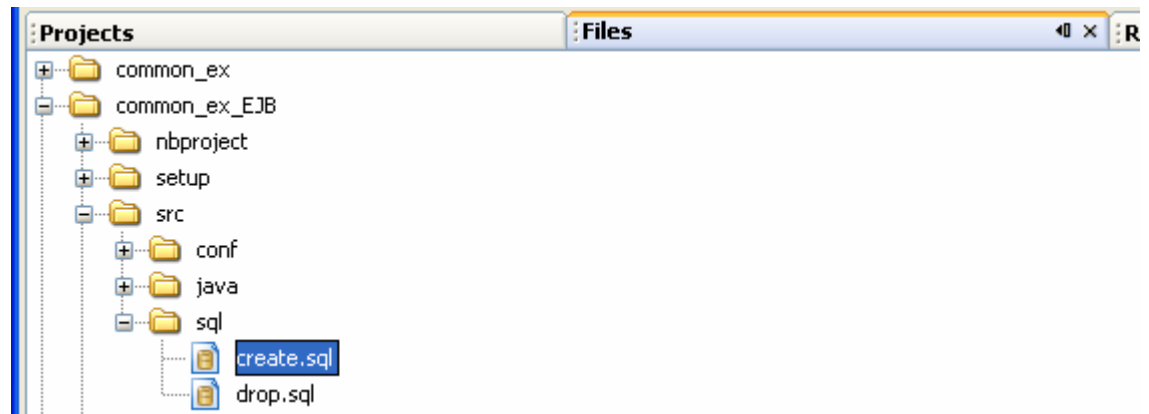
Open the common_ex_EJB project and “register” the resources starting by the mplsvpnpool, etc:



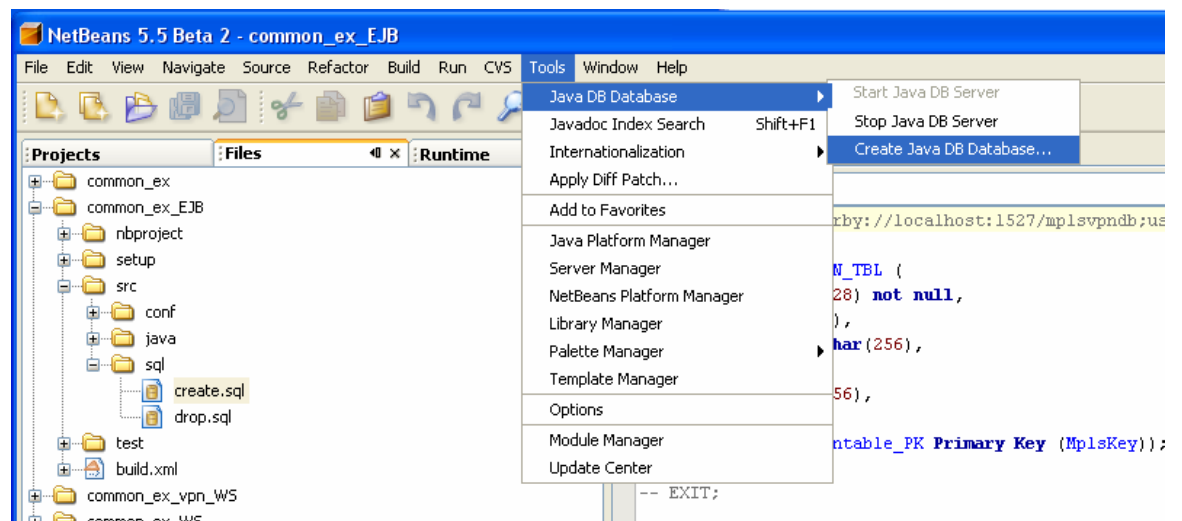
Once all the resources registered, verify the configuration into the “Runtime” tab:



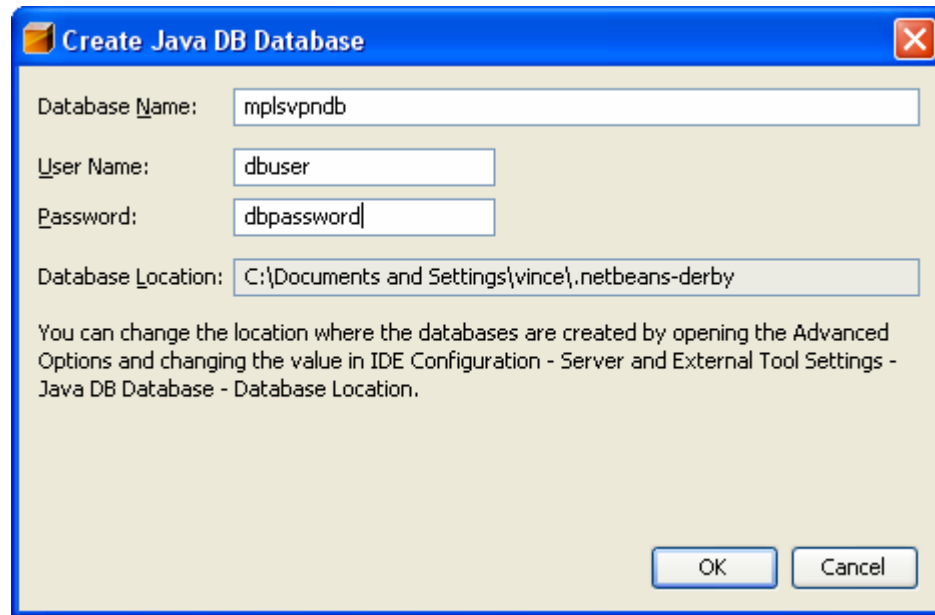
Open and create the table into the database, all needed information is located in the create.sql file of the common_ex_EJB project:



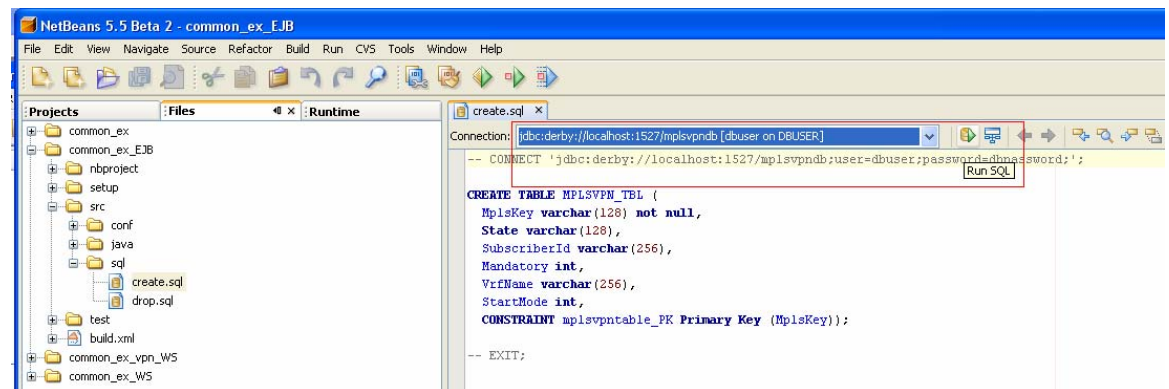
Create a new DB:



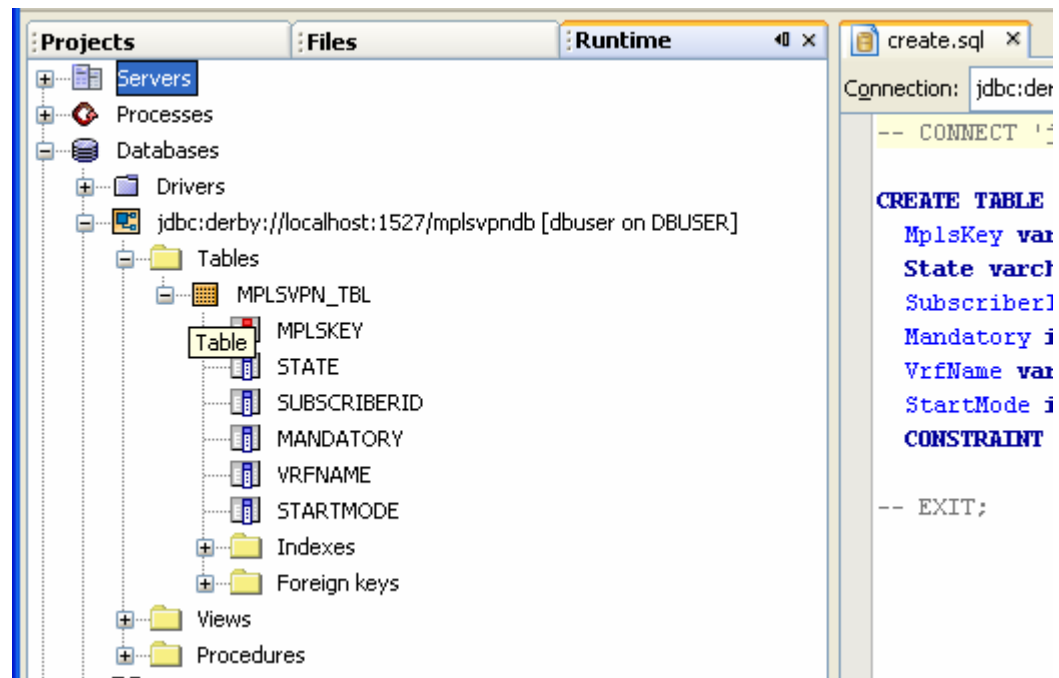
From the wizard:



Once create execute the create.sql from the edition windows once the correct DB selected:

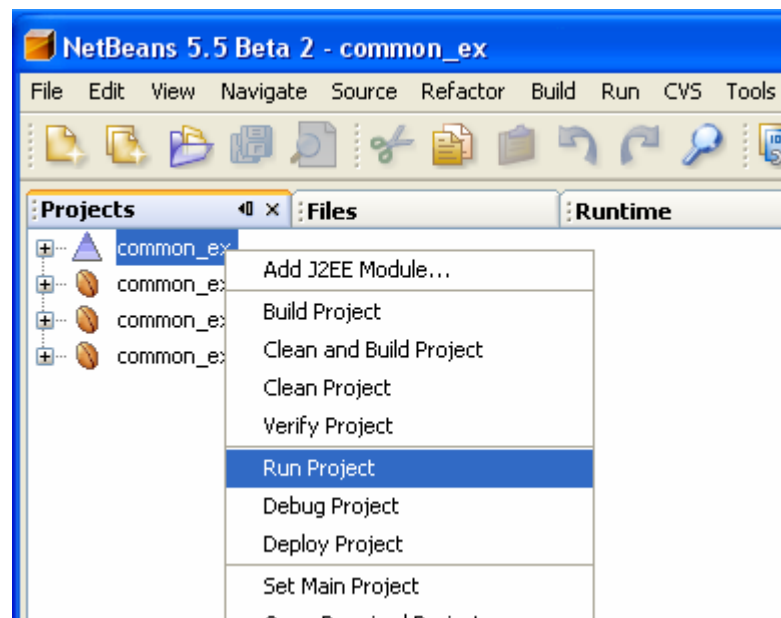


Verify the Table creation from the RunTime tab:

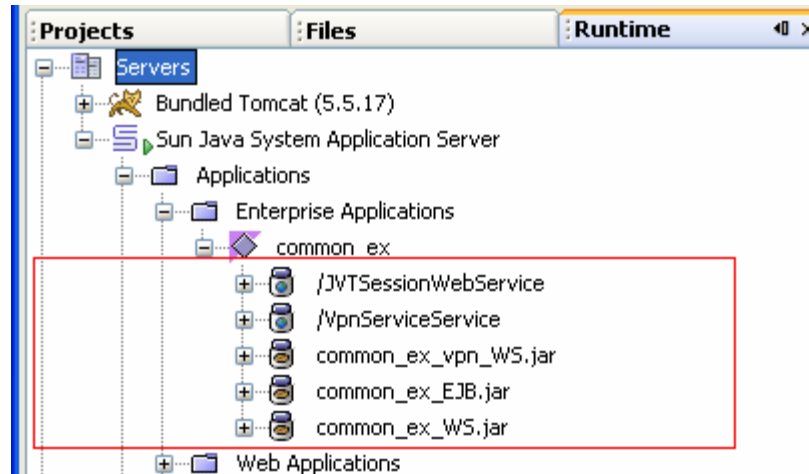


3.1.3 Deploy the application:

From the “Projects” tab, right-click on the common_ex project and select the “Build Project” and then “Run Project” item:



Verify the correct deployment from the “RunTime” tab:



The five lines in the red square indicate that the 2 WS and the 3 jars have been correctly deployed (Incase use “refresh” to actualize the representation).

(If necessary, the traces of the Application server could be checked at the bottom of the NetBeans window)

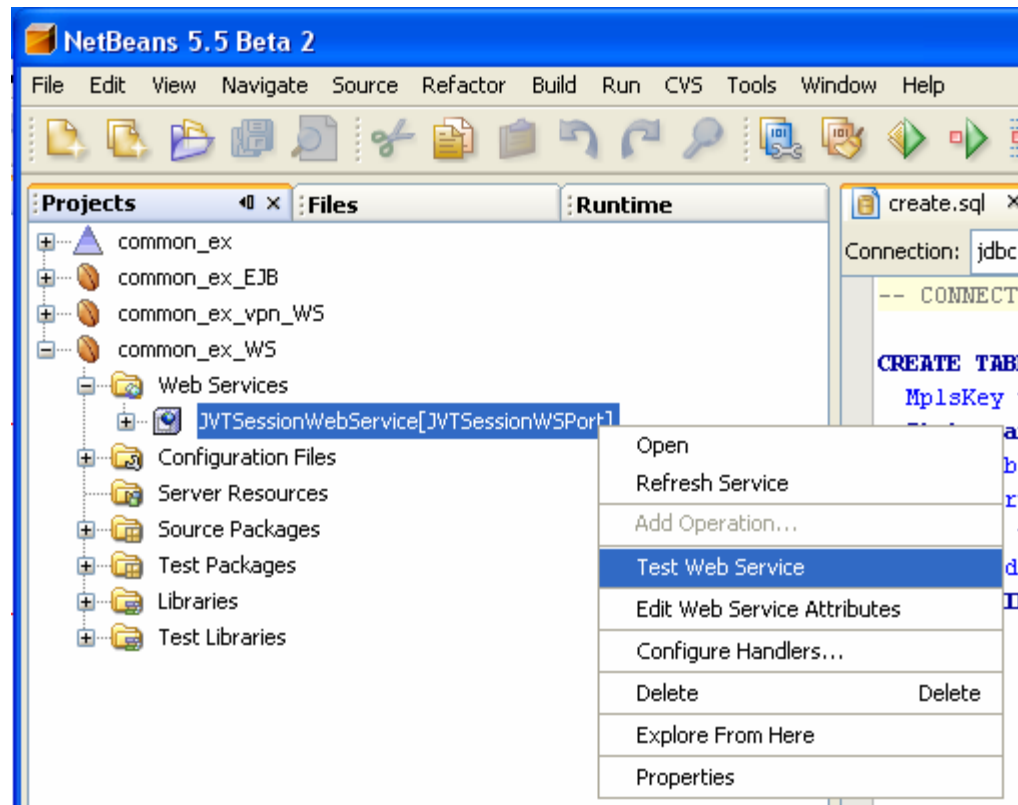
The application is now ready and could be used.

3.1.4 Exercise the Web Services profile

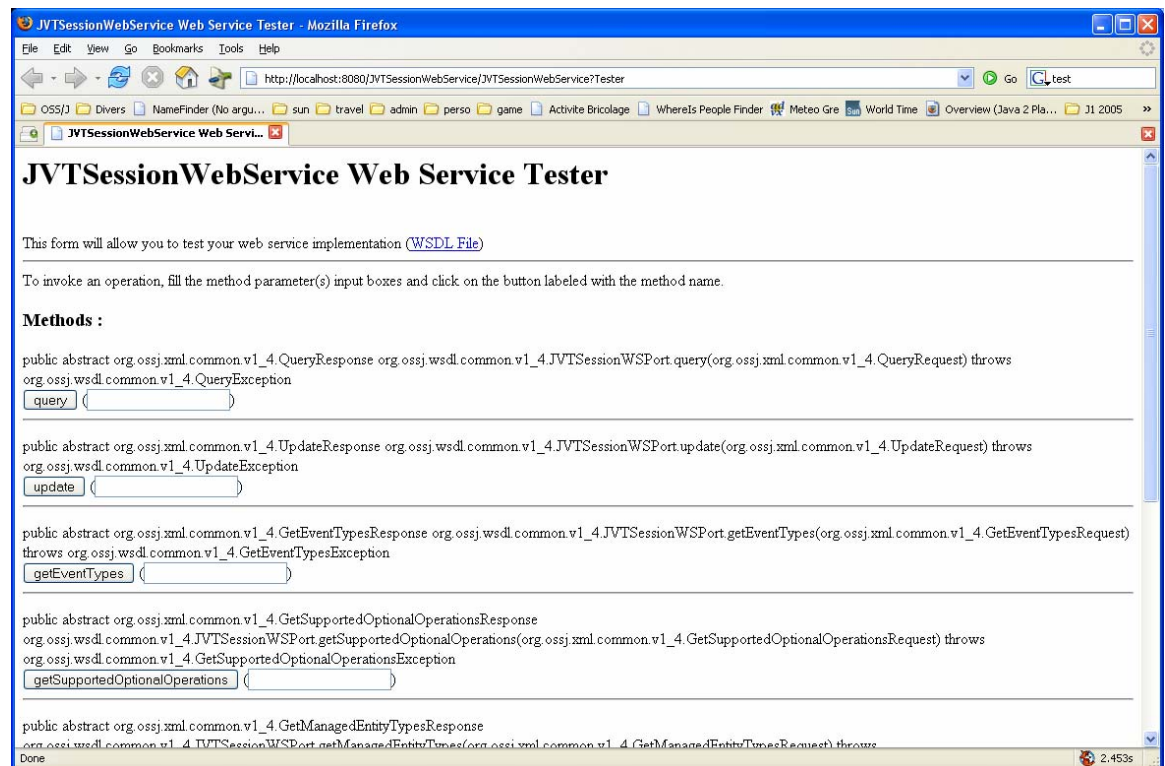
3.1.4.1 Common WS profile

The Common WS profile could be tested from your favorite Web browser.

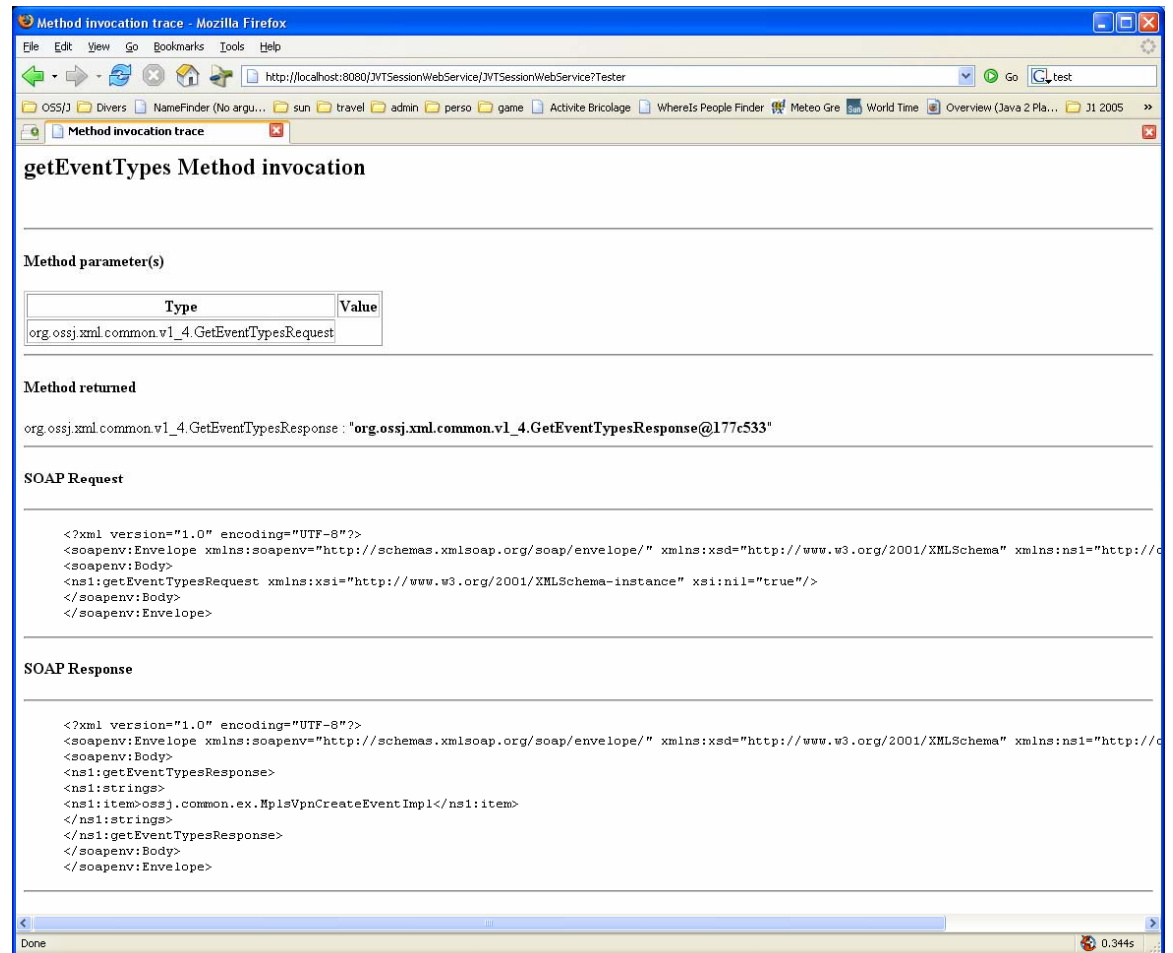
In the common_ex_WS Project section, right-click on the WS name and select “Test Web Service”:



Your favorite web browser will pop up the web services page from where you can exercise one of the OSS Common API feature:



For example, pushing the “getEventTypes” button will return:



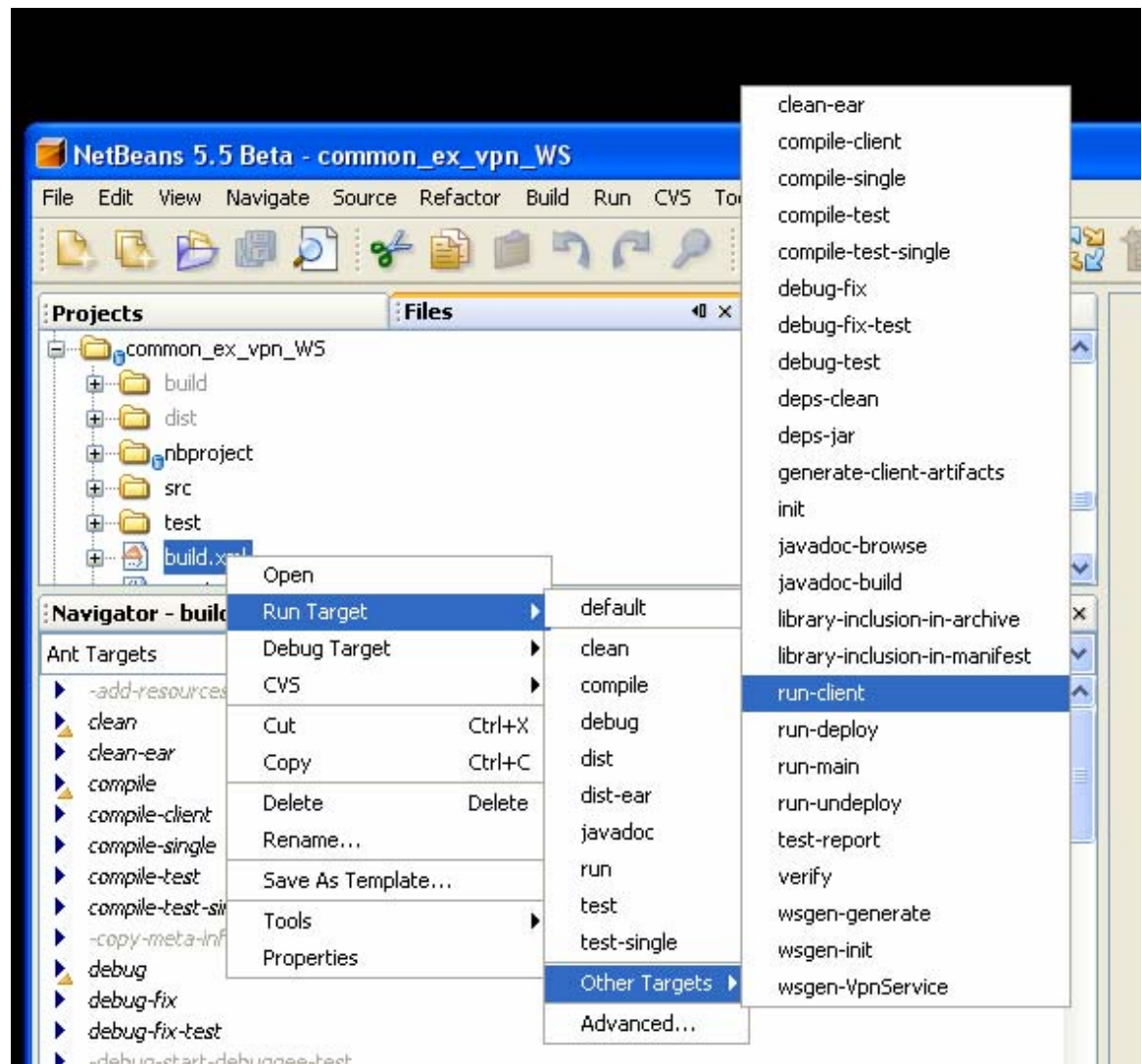
3.1.4.2 VPN Web Service (common_ex_vpn_WS project)

The VPN web service could be exercise using the same method as previously or using the provided client example. The small application benefit from the NetBeans environment to be executed.

In order to compile and execute the client, call the “compile-client” and run-client target from the contextual menu of the project:

In netbeans, select the “Files” section, open the Project directory right-click on the build.xml and select the “compile-client” then the “run-client targets.

Note: the compilation of the client is mandatory before any execution of the client.



At the bottom section of netbeans window, you will see the execution output as follow:

```

Output
Java DB Database Process x build.xml (run-client) x
init:
run-client:
Executing appclient with client class as client.VpnServiceClient
Retrieving port from the service ossj.common.ex.VpnServiceService@19ed7e
Invoking 'createMplsVpnService' operation on the Servant port
Result, MplsVpnServiceKey returned for the new created:
    primaryKey: 1158151709343
    type       : com.infomodel.MplsVpnServiceKeyImpl
    VrfName    : myVrfName
Invoking 'getMplsVpnService' with 1158151709343
Result, MplsVpnServiceKey returned for the new created:
    primaryKey    : 1158151709343
    SubscriberId  : mySubscriberId
    VrfName       : myVrfName
Invoking 'setMplsVpnServiceByValue'
Invoking 'getMplsVpnService' with 1158151709343
Result, MplsVpnServiceKey returned for the new created:
    primaryKey    : 1158151709343
    SubscriberId  : mySubscriberId
    VrfName       : updated VrfName: [1158151709343]
BUILD SUCCESSFUL (total time: 3 seconds)

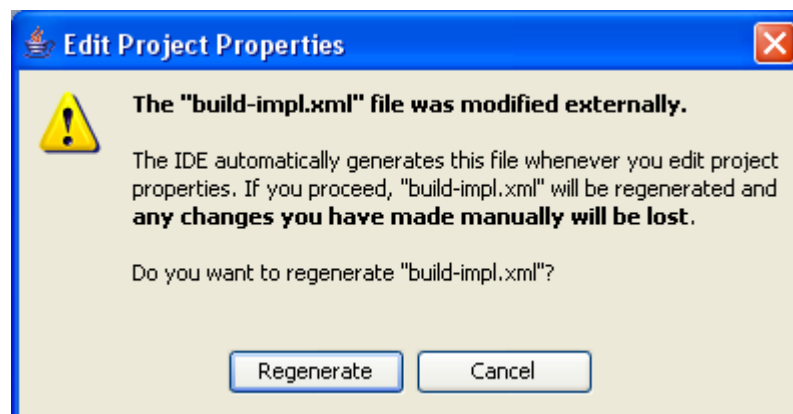
```

Output

Finished building build.xml (run-client).

WARNING:

Netbeans may detect that its internal projects file have been modified (see design document), and will pop up the following message:



Always select "Cancel" to preserve the necessary modifications.

Appendix A: Glossary and References

References